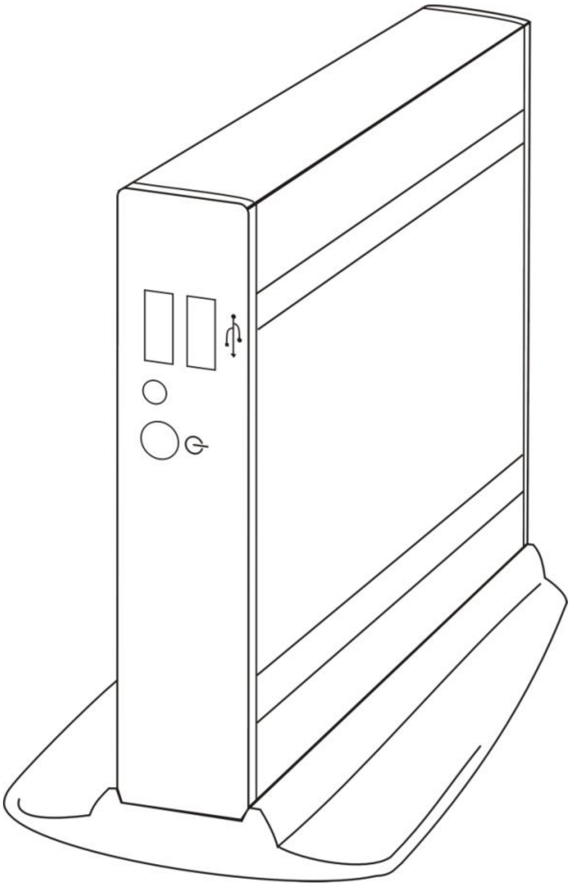
TEMPEST DSM USER MANUAL



Manual Version 1.0a

Nov 2006
©Tempest DSM Installation Guide:®
Tempest DSM Version 1.0a

Acknowledgements:
Microsoft and Windows are registered to Microsoft Corporation
Hewlett Packard
Jet direct

E&OE

Software Licence and Copyright
The Software is copyrighted and all rights reserved by William Matthew Limited, 22 Haxey Lane, Haxey, Doncaster, South Yorkshire, DN9 2NE. UK. Email Info@wmltd.co.uk

IMPORTANT: Before using the Tempest DSM Software please read the licence agreement carefully, the right to use this software is granted only if the Customer agrees to the terms of this Licence. If you do not agree to the terms of this Licence, please return the product unused.

William Matthew Limited considers that installation of the Software is your acceptance of our terms and conditions of the license.

1. The Software has been developed for use with the Tempest DSM Product range; The Licence is to be used without any time limit and maybe subject to change.

2. The Software may not be duplicated, copied, resold, or re-branded without the written consent of William Matthew Limited.

3. It is the responsibility of any user, reseller, Distributor, OEM and other third party to protect William Matthew Limited rights against violation of this Software Licence.

4. William Matthew will consider any attempt to duplicate the functionality of any application provided by the software will constitute a breach of software licence, and may instigate legal proceedings to protect the intellectual property rights of the Tempest DSM and William Matthew Limited.

5. This Licence and the Software product may be transferred to a third party provided the third party agrees to all the terms of this licence and that the transferring party retains no copies of the Software.

6. William Matthew reserves the right to terminate this Licence upon breach.

7. In the event of a termination, all copies of the software must be returned to William Matthew Ltd

8. Any disputes In respect of the Software Licence will be considered a breach of copyright and is subject to English Law.

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

Introduction to Tempest DSM Print Appliance

The Tempest DSM is a multiport print server/network bridge with the capability of manipulating print data streams and embedding forms and overlays to produce and distribute print ready output to any laser on the network or connected to its local USB ports. The device will allow the conversion of standard PCL5 escape sequences to print the most commonly used bar codes. The Tempest DSM's on board software generates these fonts internally. The unit is designed to be code compatible with the more commonly used application that supports PCL5 bar code printing. The on board software also includes PreScribe, Epson FX100, IBM Proprinter and Printronix emulations so output which was normally printed using these matrix printer drivers can now be printed to a standard PCL5 laser printer.

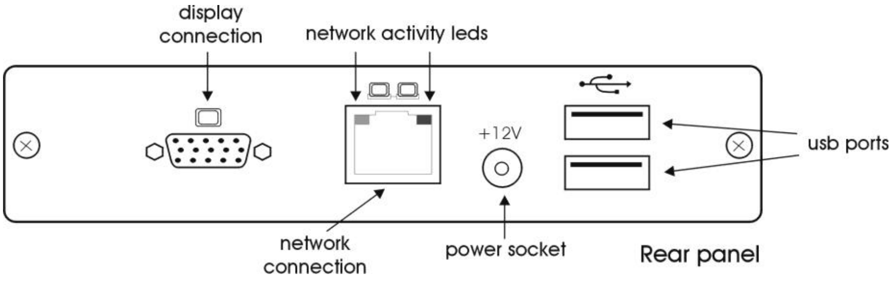
User defined fonts forms and overlays can be stored on the unit and can be associated with each output port with a simple to use web interface. The device can be licensed to print to a maximum of 4 local printers and up to 50 Network ready printers.

Connection to the Network

The Tempest DSM can be connected to any 10/100 Ethernet network that supports TCP/IP Raw port or LPR printing.

The following configuration information is for Windows XP but most Windows systems operate in a similar manner.

You will need one Ethernet RJ45 cable of the right type for your network configuration, also one or more USB printer cables. If you would like to connect to a parallel printer conversion cables USB/Parallel are readily available.

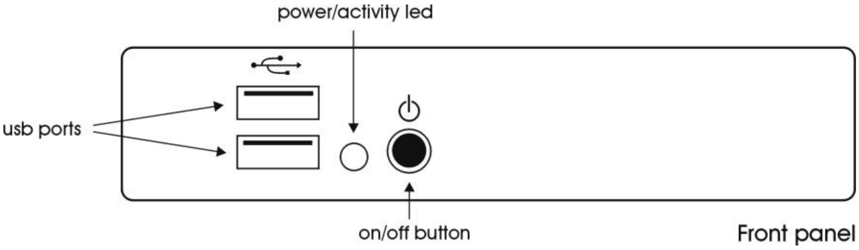


Connect the box to your network by plugging the relevant RJ45 cable into the network port on the back of the unit.

Plug in the power supply marked on the back of the unit. Turn the box on using the button on the front of the box. You should see the blue indicator light come on.

Test and configuration page

Note: After you have connected the unit to a USB printer the Test and configuration page will be printed to usb1 when the IP address of the unit is changed. Alternatively you can switch any USB printer connected to the box off and on and a Test and configuration page will be printed on that printer.



The unit comes with four USB output ports for this particular model. You can use one or all simultaneously depending how the box is licensed.

USB Port Layout

USB Port 1	Rear Panel Top
USB Port 2	Rear Panel Bottom
USB Port 3	Front Panel Top
USB Port 4	Front Panel Bottom

Factory Default Settings

Default Password = password

The box can be configured from a built in Web page this can be accessed from a web browser on a network client.

Note: The device will automatically search for a DHCP server for 30 seconds to obtain a network IP address, if it finds a DHCP server it will apply the supplied IP address and print a status page to USB1. If the unit does not find a DHCP server it will default to the static IP address of 192.0.0.192. This process repeats every time the box is powered on until a valid IP address is stored.

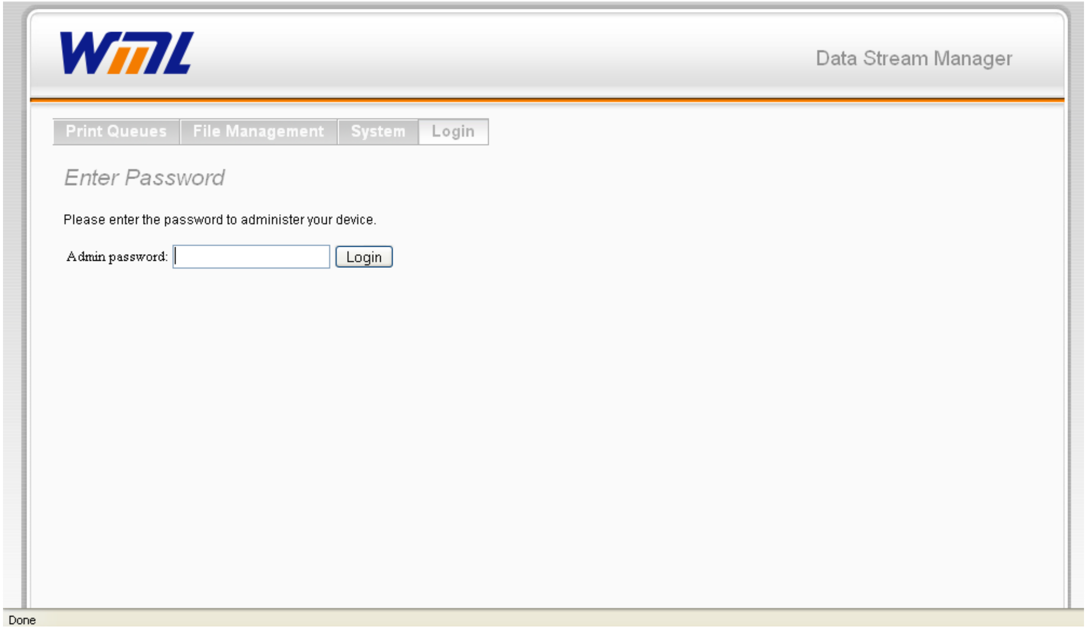
To setup a static IP address the first time you use the Tempest DSM, initially connect to the box from a network client within the IP range 192.0.0.??? Once a valid IP address for your network has been set the unit will be available on the network.

Alternatively connect a monitor to the VGA port and plug in a USB keyboard into any of the USB Ports. After the Tempest DSM Unit has booted the User prompt will be displayed on the monitor. Log in to the unit with the username “admin” the default password is “password”. You will be prompted to set DHCP or input a static IP address and subnet mask. When the IP address is changed a status page is always printed on USB1.

Web Interface Administration

To access the Administration Web page of the unit type `http://192.0.0.192` this will display the password page of the web administration.

The default Admin password is **“password”**
Type **password** into the Admin Password field and click on **Login**.



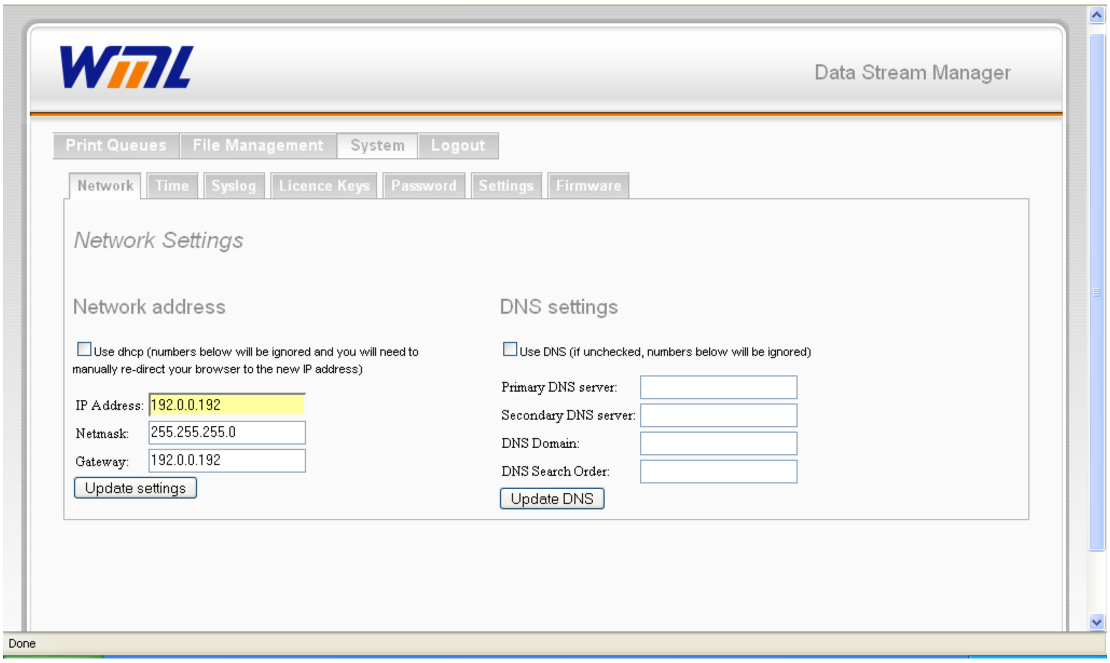
Enter Password

Please enter the password to administer your device.

Admin password:

Done

The Network setup Screen will be displayed when clicking the **System/Networ**k Tabs.



Network Settings

Network address

□ Use dhcp (numbers below will be ignored and you will need to manually re-direct your browser to the new IP address)

IP Address: 192.0.0.192

Netmask: 255.255.255.0

Gateway: 192.0.0.192

DNS settings

□ Use DNS (if unchecked, numbers below will be ignored)

Primary DNS server:

Secondary DNS server:

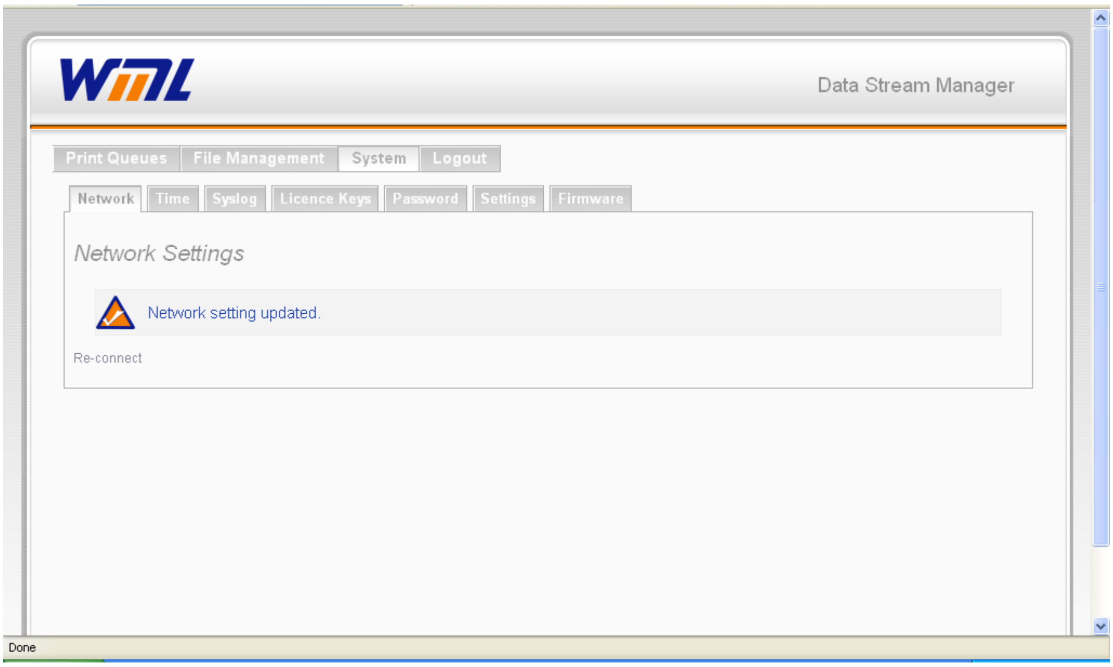
DNS Domain:

DNS Search Order:

This screen will allow you to set the units IP address, Netmask and gateway where necessary. If you have a DNS server and wish to use printer names on your network this can be set from this page also.

Note: If DHCP is selected and no DHCP server is found the unit will default back to the last configured static IP address.

When you are happy with the settings, click the Update setting button. The following page will be displayed and a status page will be printed on the printer connected to USB1.



Network Settings

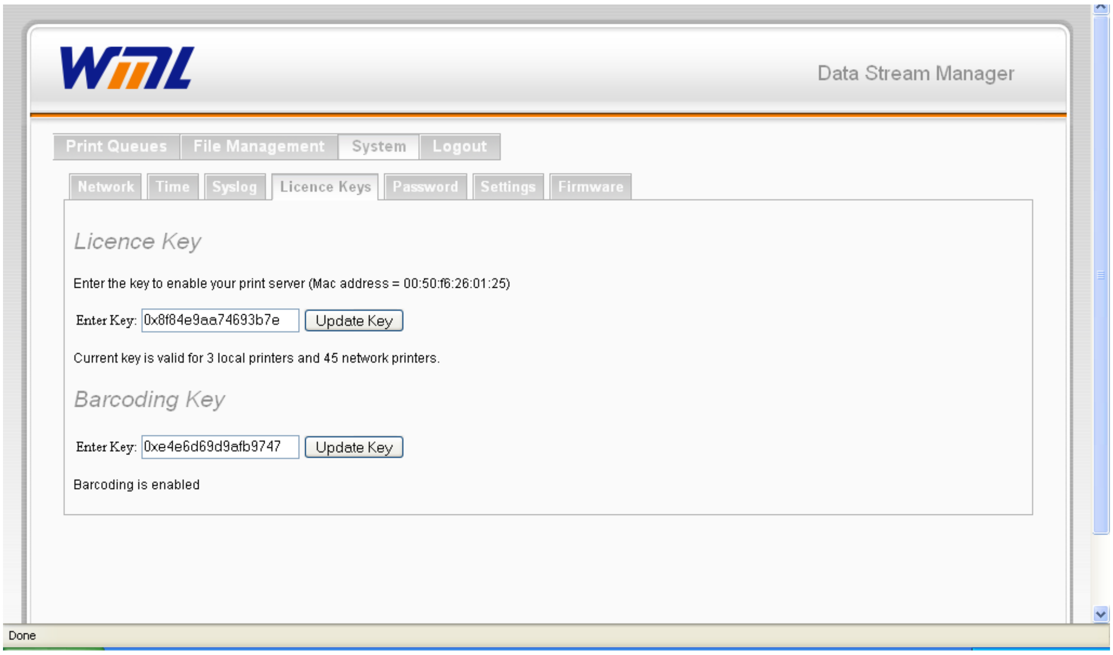
Network setting updated.

Re-connect

Note: connecting a monitor to the display connection and a USB Keyboard to any USB port will allow you to login to the unit and set a static IP address or set the DHCP default.

Login: admin
Password: password

Licence Key Page



Licence Key

Enter the key to enable your print server (Mac address = 00:50:f6:26:01:25)

Enter Key: 0x8f84e9aa74693b7e

Current key is valid for 3 local printers and 45 network printers.

Barcoding Key

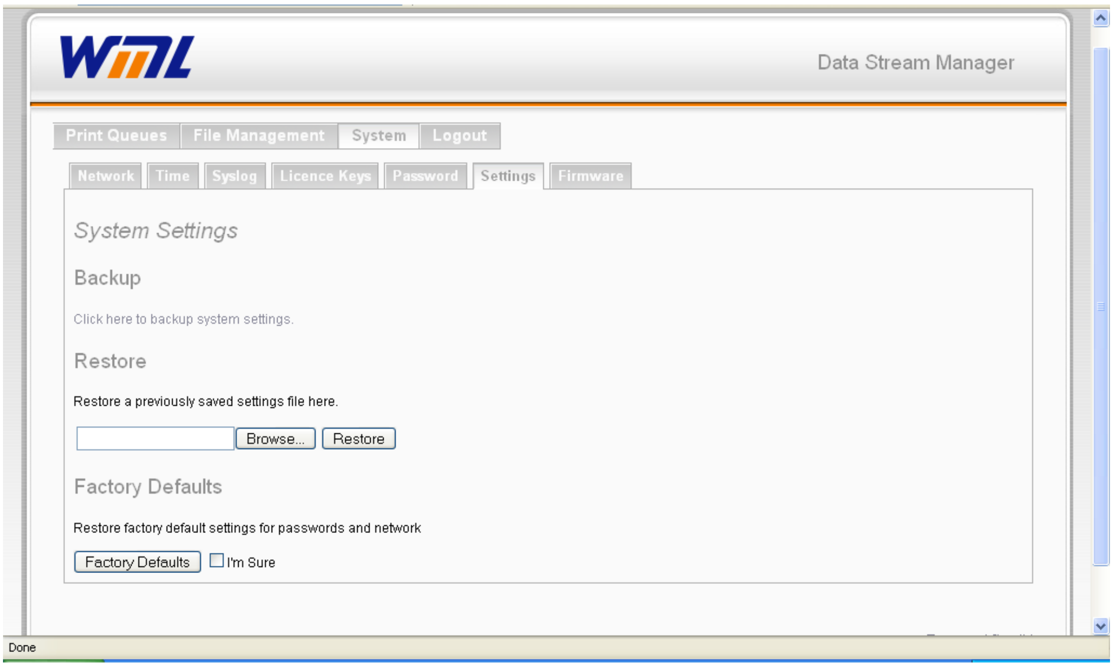
Enter Key: 0xe4e6d69d9afb9747

Barcoding is enabled

From this page a user can define and update configuration Keys for the number of enabled ports and enable the licence for barcode printing. The number of printers enabled will be displayed in this case is 3 local USB ports and 45 network printers and Barcoding are enabled. If you need to add extra printers or the Barcoding function please contact WML or your distributor for a relevant valid Licence key.

Note: Only install a valid Key that has been provided from WML or you may permanently disable the unit and invalidate any warranty.

Backing up the configuration: System/Settings



System Settings

Backup

Click here to backup system settings.

Restore

Restore a previously saved settings file here.

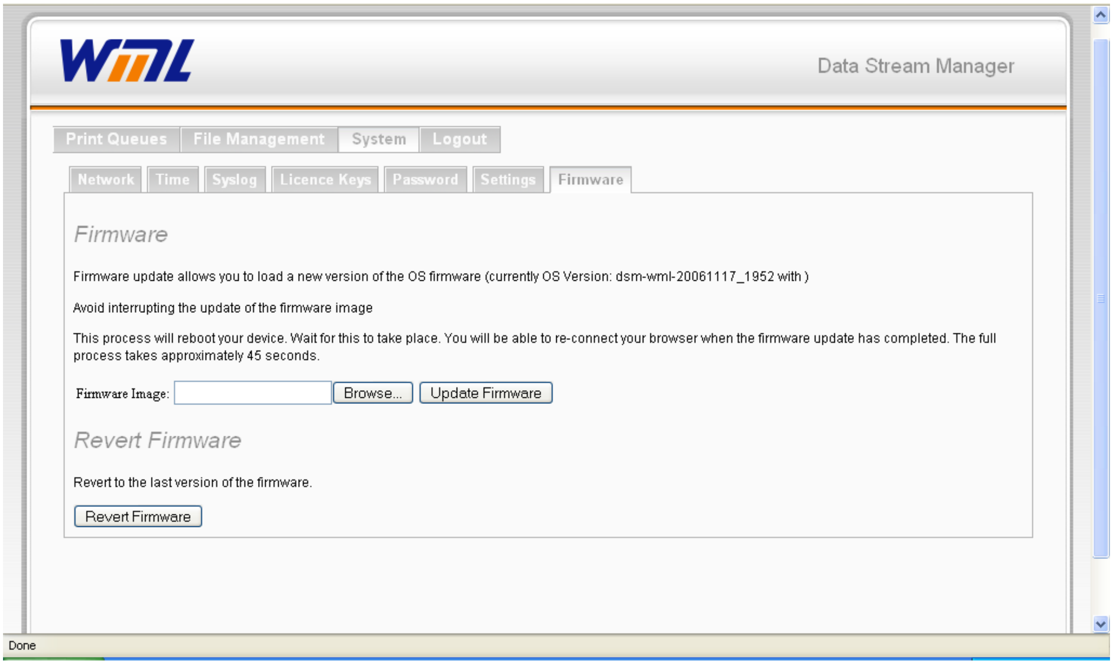
Factory Defaults

Restore factory default settings for passwords and network

From this page you can backup and restore the units queue configuration including all fonts, Forms, Macro information and ini files. It is always worth keeping a valid configuration backup so you can restore the exact configuration setup on another unit if required.

Note: Your key will be backed up, but will not be restored if you restore to a completely new unit. It is always worth while to cut and paste the key into say a word document and keep this document safe in case you over type or destroy your key.

Firmware Upload: System/Firmware



Firmware

Firmware update allows you to load a new version of the OS firmware (currently OS Version: dsm-wml-20061117_1952 with)

Avoid interrupting the update of the firmware image

This process will reboot your device. Wait for this to take place. You will be able to re-connect your browser when the firmware update has completed. The full process takes approximately 45 seconds.

Firmware Image:

Revert Firmware

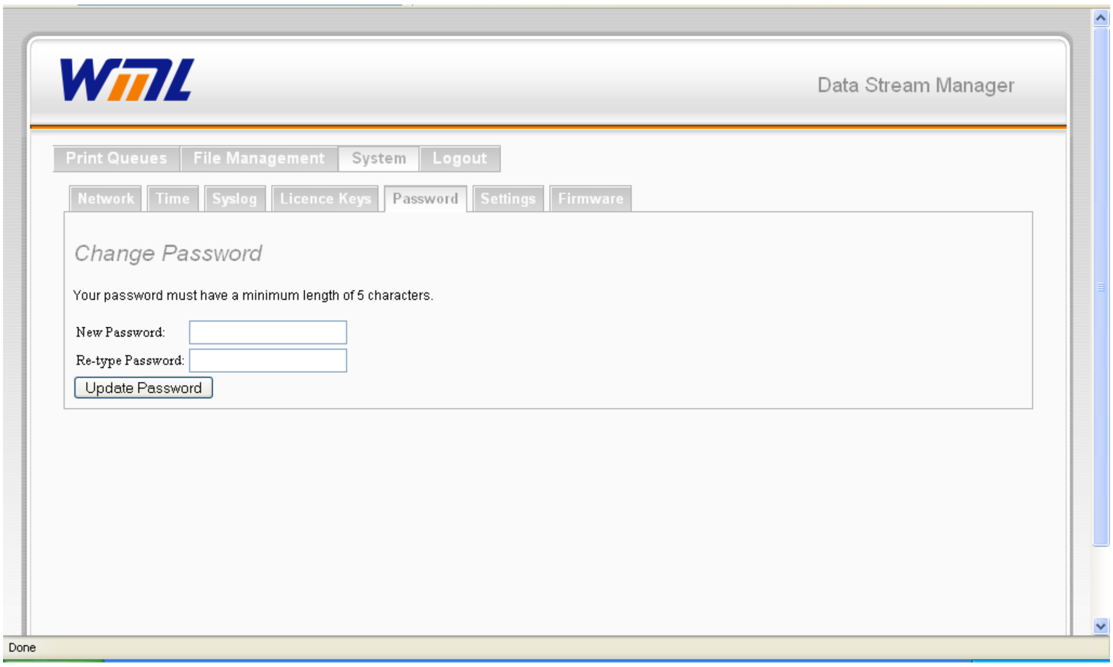
Revert to the last version of the firmware.

From time to time WML will release upgrades or later revisions of the firmware. This can be updated from this page. The user can browse the network and open the required update file, much as you would do from any Windows system. Press the Update firmware button and the box will automatically update and reboot. This may take up to 45 seconds and you may have to log into the Web page after an update has been completed.

Also on this page there is a Revert Firmware button which will allow the user to Revert to the last know working Firmware.

Note: Uploading the same Firmware twice will mean that you will loose the previous version of the Firmware.

Changing the Admin System password: System/Password



Change Password

Your password must have a minimum length of 5 characters.

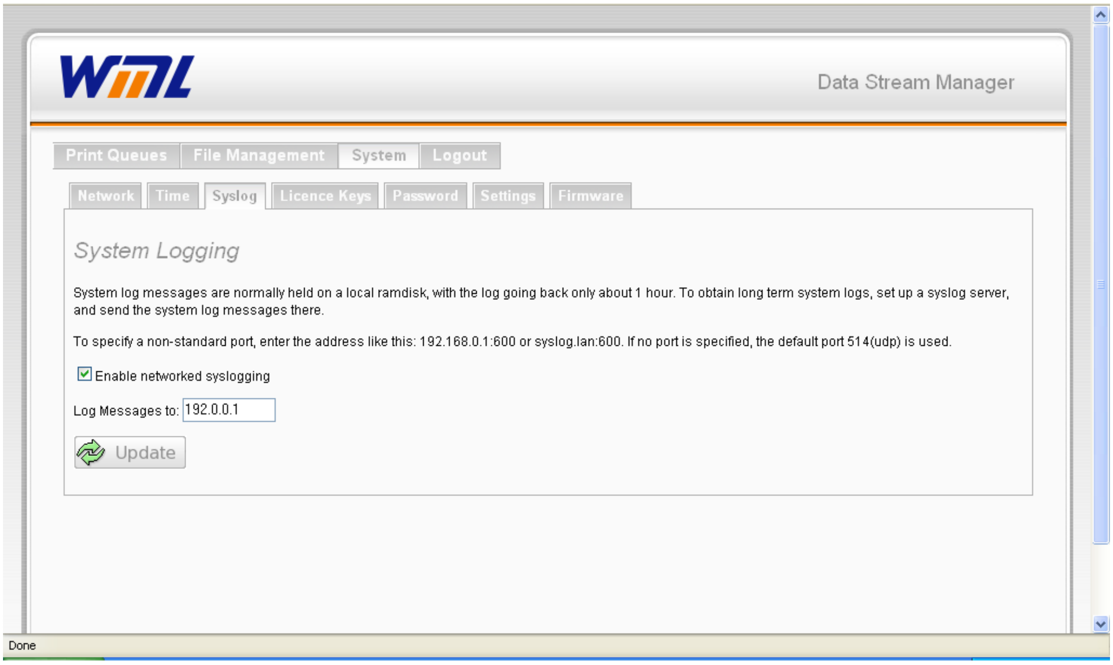
New Password:

Re-type Password:

To change the default password (Password) to something more secure use the update fields on the System/passwords page. Clicking Update password will store the new password so **make a note of it for future use**.

Time and Syslog : System/Time , System/Syslog

You may need a complete log of all the processed jobs that have passed through the Tempest DSM in order to do this you will need a syslog server on your network. To point the syslog data generated from the unit to this server input the servers IP address and enable logging. Click Update to save the settings.



System Logging

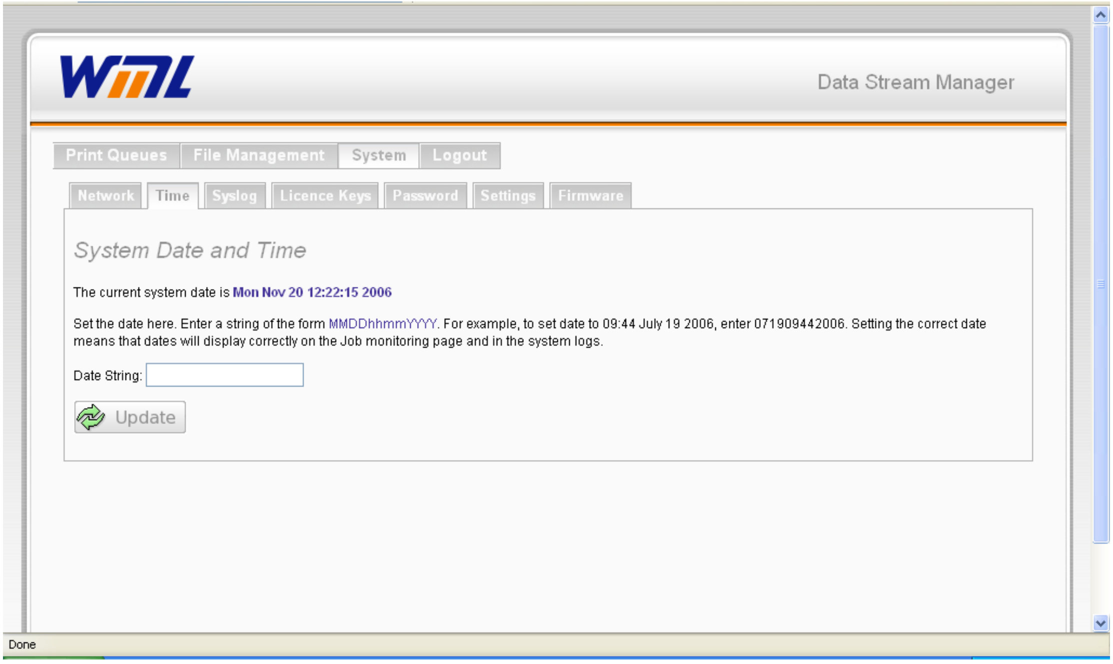
System log messages are normally held on a local ramdisk, with the log going back only about 1 hour. To obtain long term system logs, set up a syslog server, and send the system log messages there.

To specify a non-standard port, enter the address like this: 192.168.0.1:600 or syslog.lan:600. If no port is specified, the default port 514(udp) is used.

☑ Enable networked syslogging

Log Messages to: 192.0.0.1

To correctly time stamp the Syslog data set the Date and Time on the System/Time page. Click update to save the setting.



System Date and Time

The current system date is Mon Nov 20 12:22:15 2006

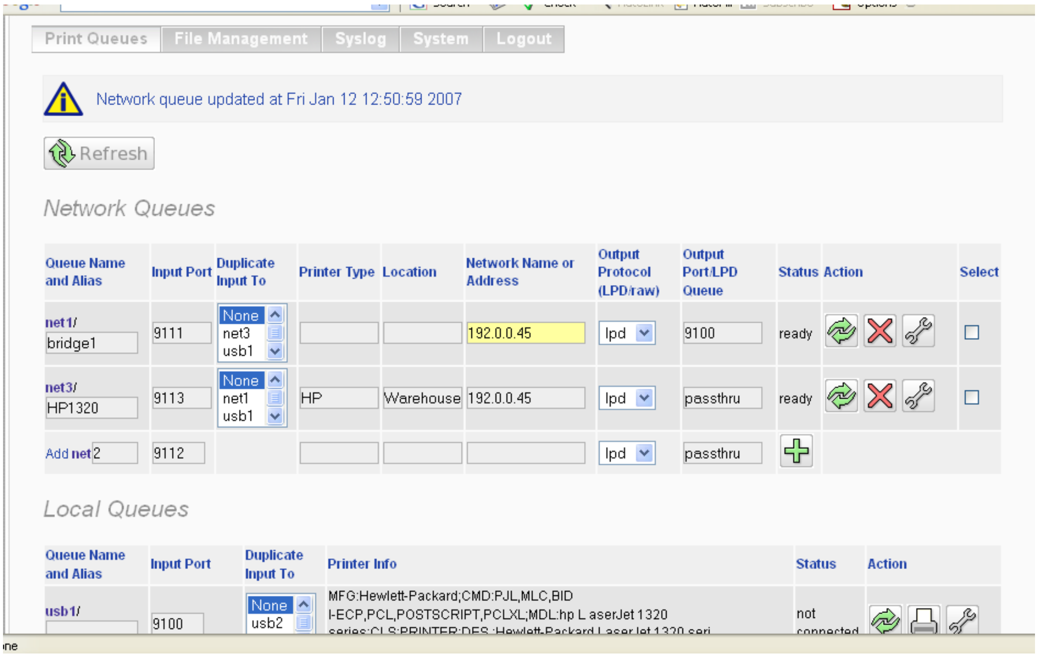
Set the date here. Enter a string of the form MMDDhhmmYYYY. For example, to set date to 09:44 July 19 2006, enter 071909442006. Setting the correct date means that dates will display correctly on the Job monitoring page and in the system logs.

Date String:

Setting up Local and Network Printers: Print Queues

The Tempest DSM depending on how the unit is licensed can manipulate print data, serve stored fonts, forms, overlays and Bar codes to up to 50 network ready PCL printers and 4 locally attached USB printers.

Select the Print Queues page on the web interface, from this page you can manage which data stream or system queue will be pointed to each network or local USB printer.



Network queue updated at Fri Jan 12 12:50:59 2007

Network Queues

Queue Name and Alias	Input Port	Duplicate Input To	Printer Type	Location	Network Name or Address	Output Protocol (LPD/raw)	Output Port/LPD Queue	Status	Action	Select
net1/ bridge1	9111	None net3 usb1			192.0.0.45	lpd	9100	ready		□
net3/ HP1320	9113	None net1 usb1	HP	Warehouse	192.0.0.45	lpd	passthru	ready		□
Add net 2	9112					lpd	passthru			

Local Queues

Queue Name and Alias	Input Port	Duplicate Input To	Printer Info	Status	Action
usb1/	9100	None usb2	MFG:Hewlett-Packard;CMD:PJL,MLC,BID I-ECP,PCL,POSTSCRIPT,PCLXL;MDL:hp LaserJet 1320 series:CLS:PRINTER:DES:Hewlett-Packard LaserJet 1320 seri	not connected	

Queue Name and Alias Queue Name: If you are inputting data from a LPR/LPD queue on your server the default queue names for each enabled input port on the Tempest DSM follow the format:

The first local USB port queue name is usb1, USB 2 queue name is usb2 the third usb3 and the forth is usb4.

The same convention is used for the bridged network printers, the first being net1 the second being net2 and so on.

Editing the field provided can set an alias Queue Name. For example net1/bridge1 are the default LPR/LPD queue name and Alias name for the first network port. This port will always respond to the queue name net1 but you can also define the Alias by editing the bridge1 field to another name.

Input Port: If you are inputting data from a RAW queue on you server the input port addresses are user configurable but the box defaults to the following port numbers.

Local Ports USB1= port 9100, USB2= port 9101, USB3= port 9102 and USB4= port 9104.
Network bridged ports net1= port 9111, net2=9112, net3=9113, net4=9114 and so on up to net50=9160.
If you need to output to another port from your server change the input port field for the relevant printer and click the update

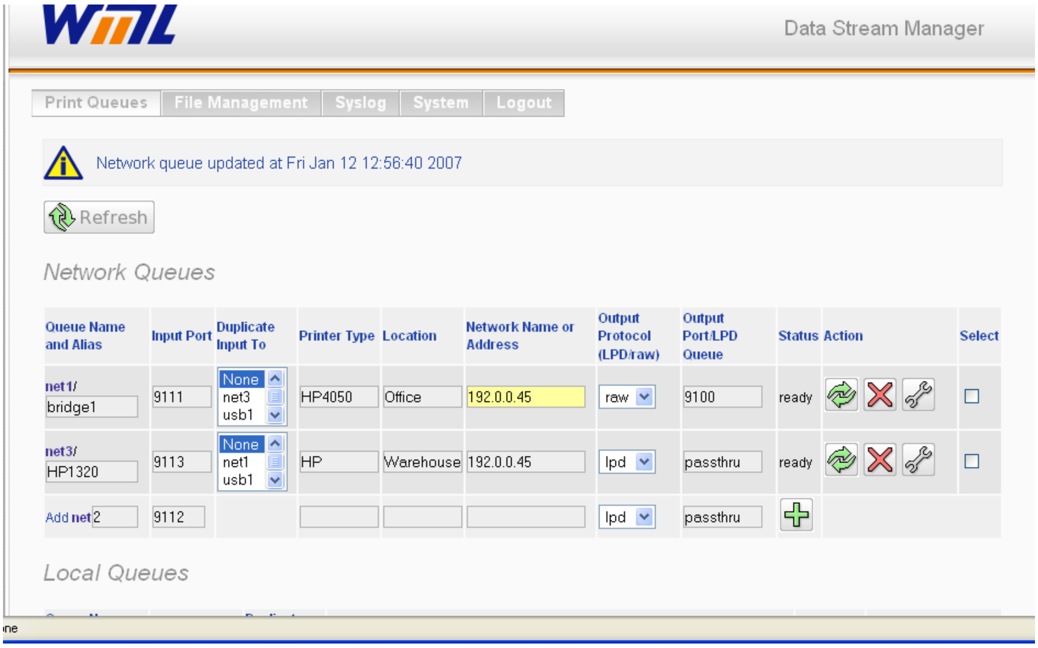
Printer Type and Location fields: These fields are purely descriptive and can be used for information on where the printer is located for example: **HP Laser Jet 1320, Admin Office.** Use the button to store any changes to this page.

Network Name or Address: In this field either tell the Tempest DSM box which network ready printer you want the output to go to: If the printer has a network name that is resolved with a DNS/DHCP server or if the device has a know IP address Example: If you set 192.0.0.45 in this field all output data for that queue will go to the network printer with the IP address of 192.0.0.45. If this field were set to HP1320N all out put data would be sent to the printer with the network name HP1320N. Again click the 🔄 button to store the settings.

Output Protocol (LPD/RAW): You will need to tell the Tempest DSM which type of data protocol you wish to output to the printer select LPD or RAW from the drop down menu. Click the update button to store 🔄.

Output Port LPD Queue: you will need to either input the LPD queue name or the RAW port number for the output printer you are sending the data to. Most network ready printers will accept the LPD queue name Passthru which is the default setting, like wise most network ready printers will accept data on RAW port 9100. Click the update button to store the settings 🔄.

Example: Settings for Net1 queue may look like this.



Network queue updated at Fri Jan 12 12:56:40 2007

Network Queues

Queue Name and Alias	Input Port	Duplicate Input To	Printer Type	Location	Network Name or Address	Output Protocol (LPD/raw)	Output Port/LPD Queue	Status	Action	Select
net1/ bridge1	9111	None net3 usb1	HP4050	Office	192.0.0.45	raw	9100	ready	🔄 ✗ 🔧	□
net3/ HP1320	9113	None net1 usb1	HP	Warehouse	192.0.0.45	lpd	passthru	ready	🔄 ✗ 🔧	□
Add net 2	9112					lpd	passthru	+		

Local Queues

ne

Duplicate Input To: This function will allow you to copy the input to any queue to any of the other queues on the input side of the Tempest DSM. The Default setting is None by holding down Ctrl or Shift you can select multiple queues.

Example if you hold down ctrl on your keyboard and configure queue net1 high light net2 and net3 all data going to queue net1 will also be put onto queues net2 and net3. Click the update as usual to store the setting.

To delete this setting select **None** on the first Queue and click update. 🔄

Note: Master Data will be the lowest net queue number. You cannot copy input data from queue net2 to queue net1 and net3 you will need to copy the data to a higher number say net3 and net4. Be careful not to set up a data loop, if this happens you will get a constant data stream until the printer runs out of paper.

This function is very powerful for example sending a copy of a document to multiple departments, also you will see later in the manual how you can use this function with variables and different overlays on each queue to produce different outputs on multiple printers using one data stream.

✗ Button will allow you delete the relevant queue or function you will see this symbol on a number of pages in the interface.

🔧 Button allows you drill down to set up specific feature of a particular queue.

Select

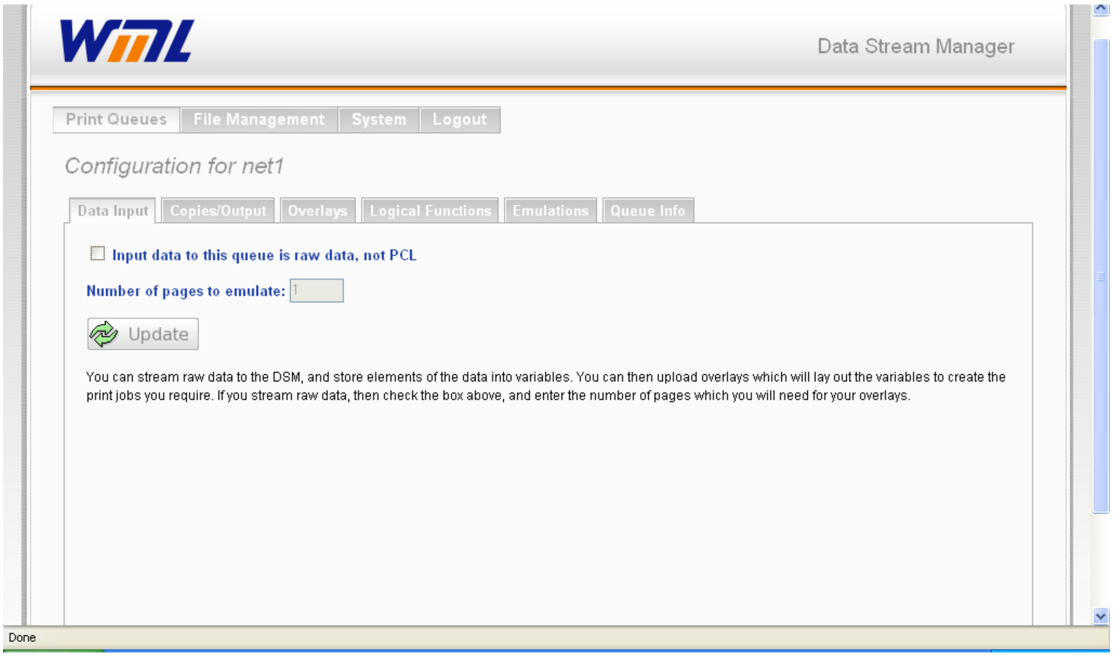
□ Checking this field on multiple queues will allow you to copy all functions of a number of queues or delete a number of queues with one operation. When Select is checked ✖ and 🔧 will appear below queue selection list use ✖ to delete selection and use 🔧 to copy your selection.

➕ Use this button to add a new queue.

🖨 Use the printer symbol to print a Tempest DSM status page this can only be done to a USB port.

Programming Individual Queues.

If you click on 🔧 field on any queue on Print Queues page the following setup page will be displayed.



Configuration for net1

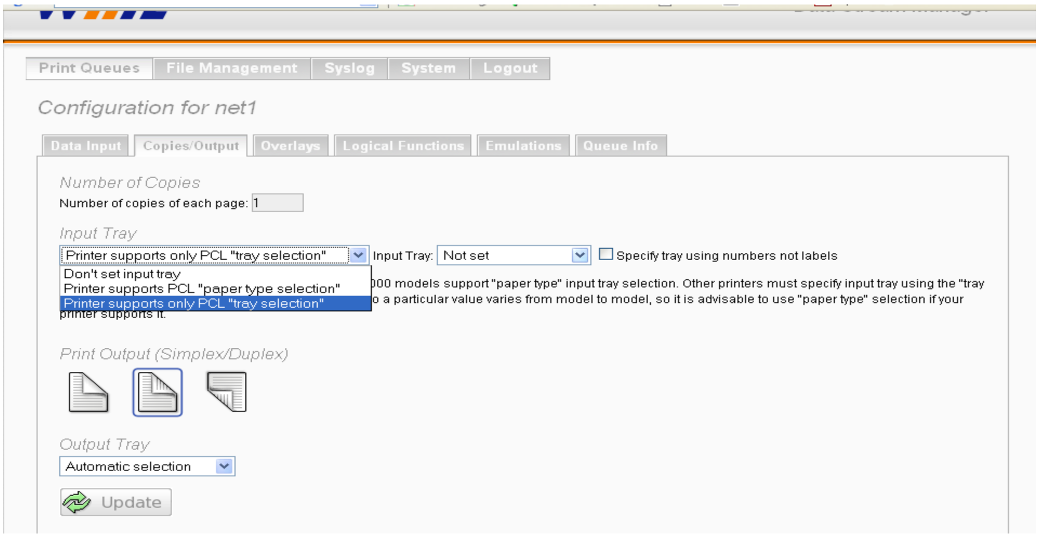
□ Input data to this queue is raw data, not PCL

Number of pages to emulate: 1

You can stream raw data to the DSM, and store elements of the data into variables. You can then upload overlays which will lay out the variables to create the print jobs you require. If you stream raw data, then check the box above, and enter the number of pages which you will need for your overlays.

If you will be sending PCL 5 data to that queue no changes need to be made to the settings. If you will be sending RAW unformatted data that has not been created by a PCL5 printer driver you will need to check the not PCL box. By checking this box the Tempest DSM will automatically put the required PCL headers into the data stream. You will need to tell the unit as well if you expect to print more then one page of RAW data.
Example: if you want to produce a document with 3 pages of RAW data you will need to put the number 3 in the number of pages to emulate field. **Press update to save the settings.**

Copies/Output Page



Configuration for net1

Number of Copies

Number of copies of each page: 1

Input Tray

Printer supports only PCL "tray selection" | Input Tray: Not set | □ Specify tray using numbers not labels

Don't set input tray
Printer supports PCL "paper type selection"
Printer supports only PCL "tray selection"

)00 models support "paper type" input tray selection. Other printers must specify input tray using the "tray
o a particular value varies from model to model, so it is advisable to use "paper type" selection if your
printer supports it.

Print Output (Simplex/Duplex)

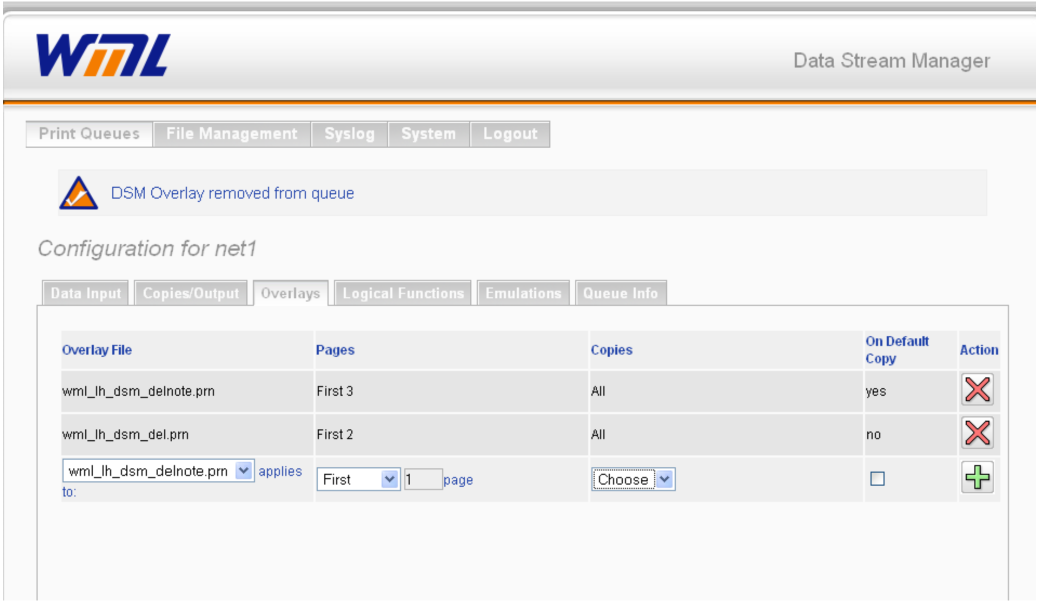
Output Tray

Automatic selection

You can select how many copies of a document you require by updating the field. Also you can select single sided, duplex or reverse duplex. **Click on Update to save the settings.**

Input and Output Tray selection can also be selected from this page. Tray numbering can differ on different printers, so some experimentation maybe required if using Tray numbers. On some printers Paper type selection for each Tray can be set from the printer's front panel. So if you setup an input tray from the printers front panel to be say Letterhead you can tell DSM to print to Tray Letterhead. Output Tray selection can be made from the drop down menu.

Overlays page



DSM Overlay removed from queue

Configuration for net1

Overlay File	Pages	Copies	On Default Copy	Action
wml_lh_dsm_delnote.prn	First 3	All	yes	☒
wml_lh_dsm_del.prn	First 2	All	no	☒
wml_lh_dsm_delnote.prn applies to:	First 1 page	Choose	□	+

Later in this manual it will be explained how to store a form, overlays or font in the Tempest DSM but associating any number of forms to any queue is a simple process. All available forms that have been downloaded to the Tempest DSM can be viewed in the first drop down menu called Overlay file, high light the required file. The next field pages the drop down menu will allow you to select if you want this file to apply to the first, only the first X pages or All pages. The copies field again will allow you to select how many of the copies the file applies to, select first, last, only X or All. Click the + to save the changes and display the setting. This process can be repeated to build up exactly which forms or overlay you want printed on each page of a document and each of the copies of that document.

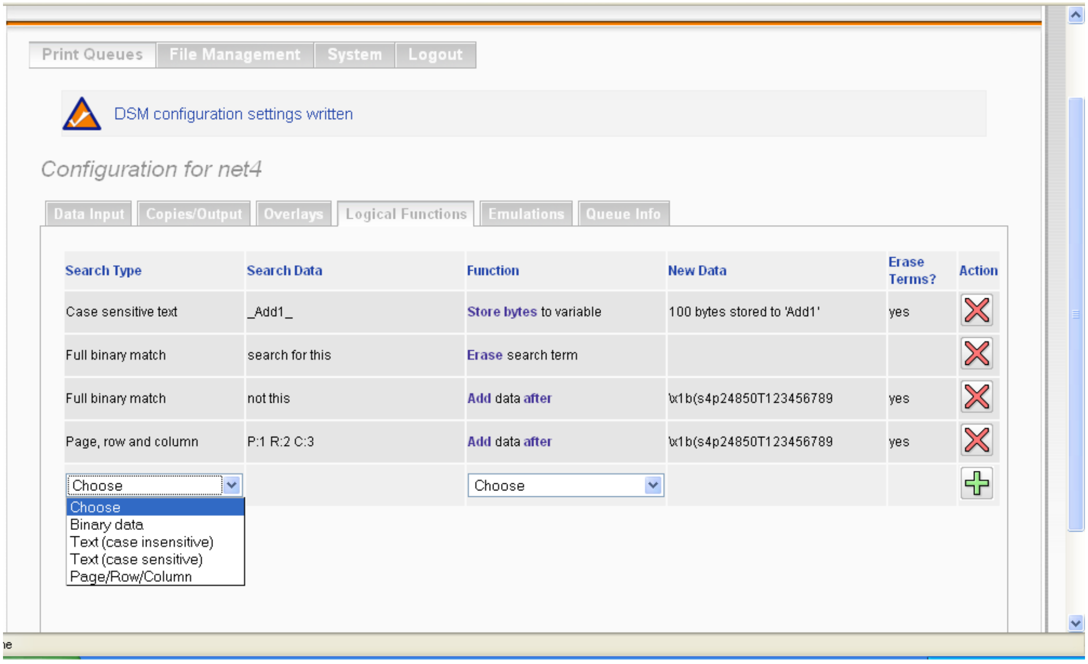
If On Default Copy is checked when setting up an overlay, the selected overlay will be used as the default if no overlay is defined for a particular page. (DSM D command)

You can edit any line to change the saved settings.

Logical Functions

Some of the more generally used logical functions that are available can be set from this page. More complex function need to setup manually with the on board .ini file.

Note: These functions work best on raw data if you are trying to manipulate PCL data streams great care must be taken to ensure the PCL driver you are using does not insert characters that may cause your data searches to fail. You may have to analyse the PCL data stream to find the correct search term. See section on Search_Binary.

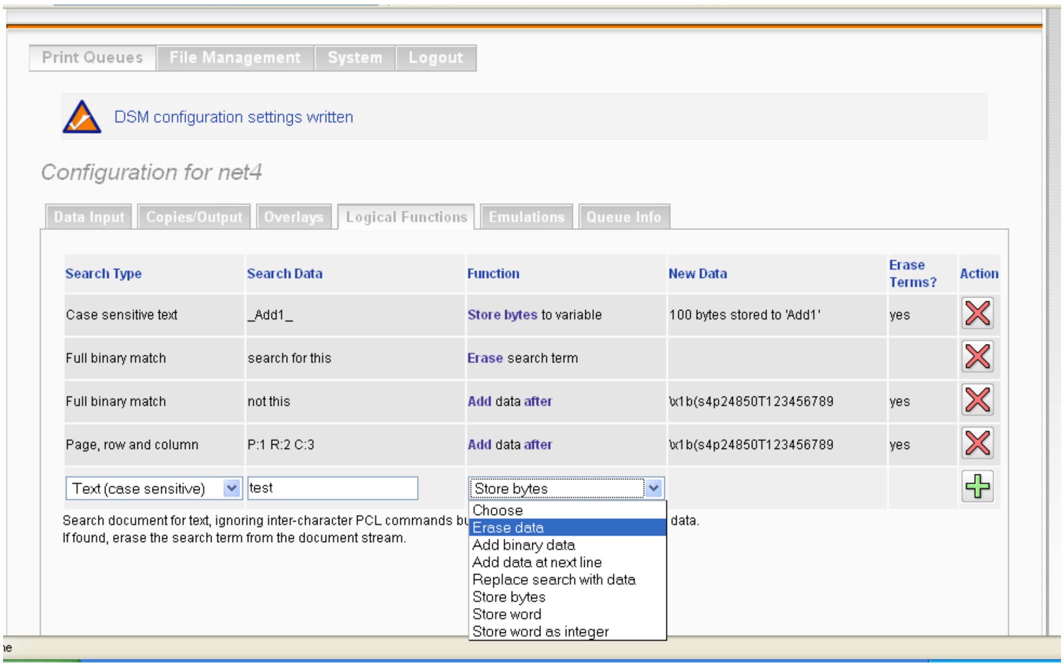


DSM configuration settings written

Configuration for net4

Search Type	Search Data	Function	New Data	Erase Terms?	Action
Case sensitive text	_Add1_	Store bytes to variable	100 bytes stored to 'Add1'	yes	☒
Full binary match	search for this	Erase search term			☒
Full binary match	not this	Add data after	\x1b(s4p24850T123456789	yes	☒
Page, row and column	P:1 R:2 C:3	Add data after	\x1b(s4p24850T123456789	yes	☒
Choose		Choose			+

In the first input field called Search type you can choose from the drop down menu the type of search you wish to perform on the input PCL or RAW data stream, these include Binary data, Text (case insensitive), Text (case sensitive) and Page/Row/Column.



DSM configuration settings written

Configuration for net4

Search Type	Search Data	Function	New Data	Erase Terms?	Action
Case sensitive text	_Add1_	Store bytes to variable	100 bytes stored to 'Add1'	yes	☒
Full binary match	search for this	Erase search term			☒
Full binary match	not this	Add data after	\x1b(s4p24850T123456789	yes	☒
Page, row and column	P:1 R:2 C:3	Add data after	\x1b(s4p24850T123456789	yes	☒
Text (case sensitive)	test	Store bytes			⊞

Search document for text, ignoring inter-character PCL commands bu data.
If found, erase the search term from the document stream.

The Second field will allow you to attach a function to the search trigger these include, Erase data, Add Binary Data, Add data at next line, Replace search with data, Store bytes, Store word and Store word as integer.

Note: The Tempest DSM will recognise \x1b as an escape character when you add binary data.

The Erase Terms function when checked will delete the search term from the final output. Example If you use a case sensitive text search for TEST and Store bytes 100 into variable HERE and erase terms. The Tempest DSM will erase the word TEST from the data stream place the next 100 bytes or till it finds a carriage return line feed into a variable called HERE. Now you have a variable with data stored in it called HERE, if you have a form or overlay associated with the queue with #HERE# embedded the stored data will be positioned at this point on the form or overlay.

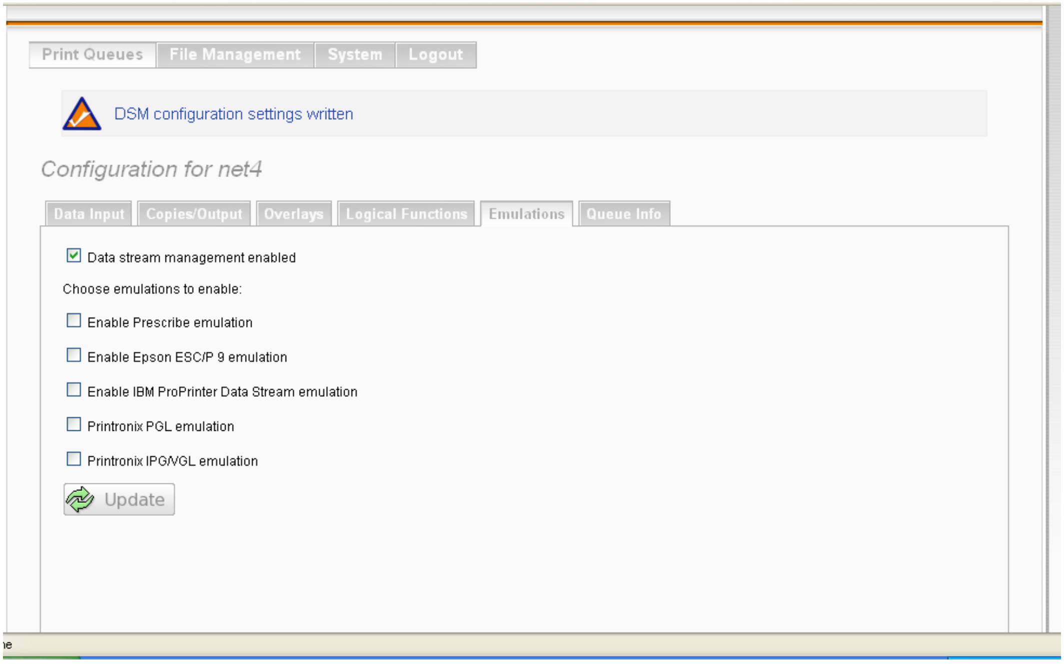
Multiple logical functions can be carried out to each queue. The ☒ and ⊞ will delete and add functions respectively.

You can edit any line to change the saved settings.

Emulations

Using the Emulations page you can convert input data streams from Kyocera PreScribe, Epson FX100, IBM ProPrinter, Printronix PGL and Printronix IPG/VGL to PCL5.

Note: The emulations are not perfect but the more common emulation commands should work. Contact WML if you experience errors.



DSM configuration settings written

Configuration for net4

☑ Data stream management enabled

Choose emulations to enable:

☐ Enable Prescribe emulation

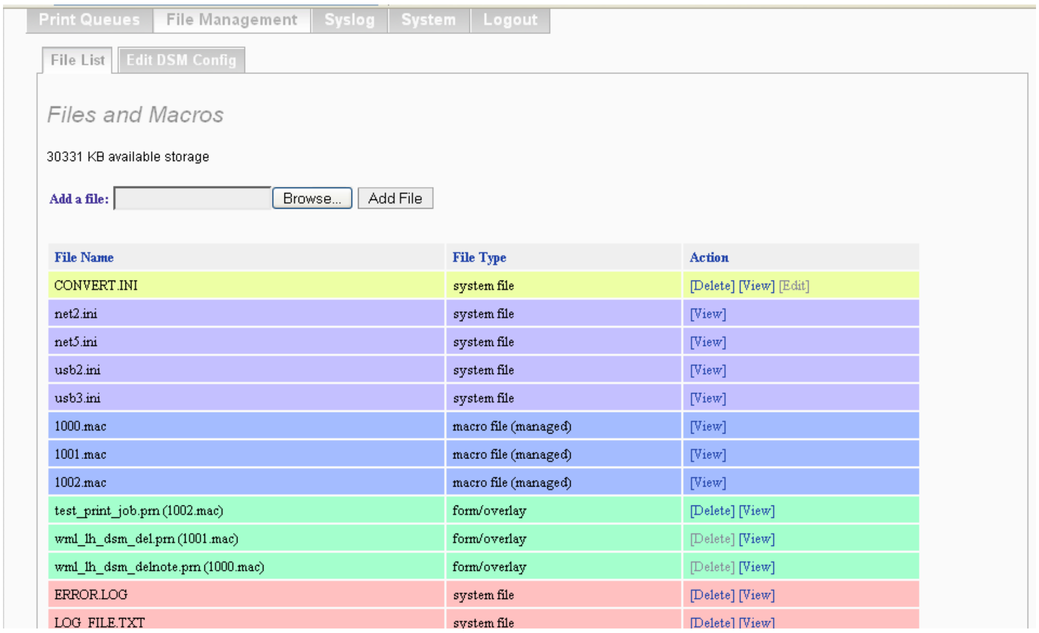
☐ Enable Epson ESC/P 9 emulation

☐ Enable IBM ProPrinter Data Stream emulation

☐ Printronix PGL emulation

☐ Printronix IPG/VGL emulation

File Management



Files and Macros

30331 KB available storage

Add a file:

File Name	File Type	Action
CONVERT.INI	system file	[Delete] [View] [Edit]
net2.ini	system file	[View]
net5.ini	system file	[View]
usb2.ini	system file	[View]
usb3.ini	system file	[View]
1000.mac	macro file (managed)	[View]
1001.mac	macro file (managed)	[View]
1002.mac	macro file (managed)	[View]
test_print_job.prn (1002.mac)	form/overlay	[Delete] [View]
wml_lh_dsm_del.prn (1001.mac)	form/overlay	[Delete] [View]
wml_lh_dsm_delnote.prn (1000.mac)	form/overlay	[Delete] [View]
ERROR.LOG	system file	[Delete] [View]
LOG_FILE.TXT	system file	[Delete] [View]

The File Management page is used to add and view and edit stored Forms/Overlays, Fonts and .ini files you can also enable the diagnostic from this page.

As you can see each section of the page is associated with a colour code.

Green section:- Displays the Forms/Overlays/Fonts stored on the Tempest DSM allowed file extensions include .mac, .ovl., .ovly, .prn, .frm and .form.

A simple way to create a form or overlay is to use a commonly used word processor such as Microsoft Word. Lay out your form and print the file to a file using a HP LJ4plus PCL5 printer driver, the resultant file will have a file extension of .prn. This file can be uploaded to the Tempest DSM using the add file button on this page. The files when uploaded will automatically assigned an internal MAC number the first file being 1000.MAC. This File is now ready and available to be used with any print queue. You can view the data in the file or delete it at any time.

Blue Section:- This displays the internal MAC numbers of the files in the Green Section and internally generated .ini files associated with each queue. You can view the contents of these files.

Mauve Section:- If you enable diagnostics and click the set button the Tempest DSM will produce a Prefilter.out and Postfilter.out file for the last print job. Viewing these files will help in problem solving as the Prefilter is the input data and the postfilter is the output to the printer. You will need to close the page and reopen it to display the Mauve section after the first enable of the diagnostic function. The files can be viewed.

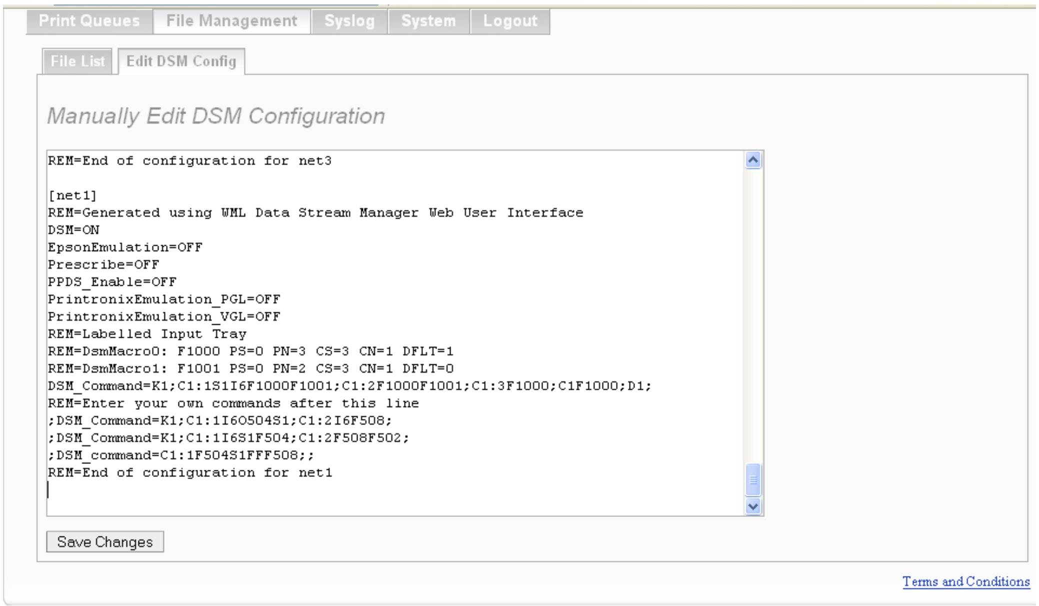
Yellow Section:- contains the main Convert.ini file which is editable.

User Defined FONTS

If you require a user defined font to be downloaded to the printer before each print job you must create a file named job_init.mac. This file should contain all the required fonts you wish to be down loaded to the printer. <u>NOTE: it is the users responsibility to make sure all fonts that are included within the job_init.mac file are licenced for use.</u>

Convert.ini

When you edit the on board Convert.ini or create your own .ini file for upload to the Tempest DSM you must follow some simple rules.



```
REM=End of configuration for net3

[net1]
REM=Generated using WML Data Stream Manager Web User Interface
DSM=ON
EpsonEmulation=OFF
Prescribe=OFF
PPDS_Enable=OFF
PrintronixEmulation_PGL=OFF
PrintronixEmulation_VGL=OFF
REM=Labelled Input Tray
REM=DsmMacro0: F1000 PS=0 PN=3 CS=3 CN=1 DFLT=1
REM=DsmMacro1: F1001 PS=0 PN=2 CS=3 CN=1 DFLT=0
DSM_Command=K1;C1:1S1I6F1000F1001;C1:2F1000F1001;C1:3F1000;C1F1000;D1;
REM=Enter your own commands after this line
;DSM_Command=K1;C1:1I6O504S1;C1:2I6F508;
;DSM_Command=K1;C1:1I6S1F504;C1:2F508F502;
;DSM_command=C1:1F504S1FFF508;;
REM=End of configuration for net1
```

The Convert.ini file is divided into sections, each new section starts with a description within [] square brackets.

[GLOBAL] section: all commands in this section will act on every enabled queue. USB1 – USB4 or net1- net50.

[net1] to [net50] sections: If you create a section which starts [net1] the commands in this section will only act on data streams that are processed by queue net1.

[usb1] to [usb4] sections: If you create a section which start [usb1] the commands in this section will only act on data streams that are processed by queue usb1.

Only the most commonly used DSM commands are created within the web interface command generator, but by creating a queue and enabling some of the functions you will see that the relevant section is setup, this can be can be added to with more complex commands or sections. If you wish to add a command to any port enter the command in the space provided.

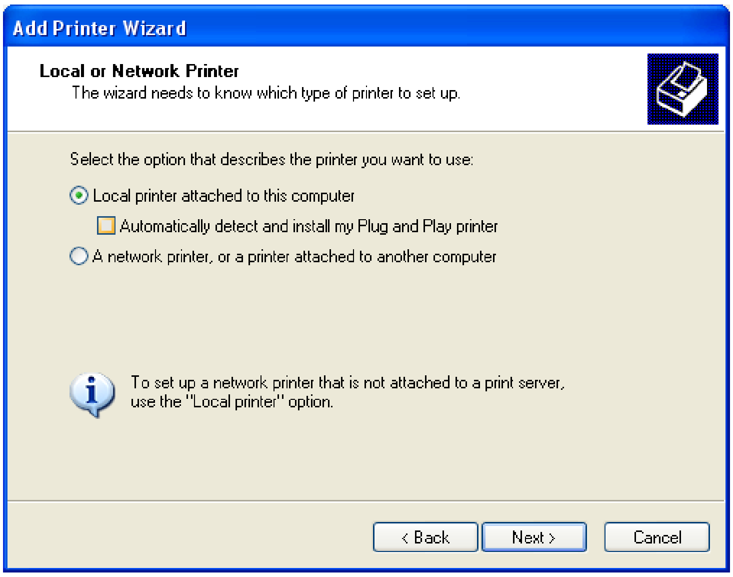
```
REM=Enter your own commands after this line
DSM_Command=K1;C1:1I6O504S1;C1:2I6F508;
;DSM_command=C1:1F504S1FFF508;
[new Section]
;DSM_Command=K1;C1:1I6S1F504;C1:2F508F502;
REM=End of configuration for net1
```

If you edit or change the Convert.ini you must click the save button to save the changes.

Windows Printer Setup

Although you can print from most operating systems through the DSM appliance we have enclosed as an example the setup that could be used to enable a print queue from Windows XP.

From the **Control Panel** window select **Printers and Faxes**,
from Printer tasks menu select **Add Printer.**



Add Printer Wizard

Local or Network Printer
The wizard needs to know which type of printer to set up.

Select the option that describes the printer you want to use:

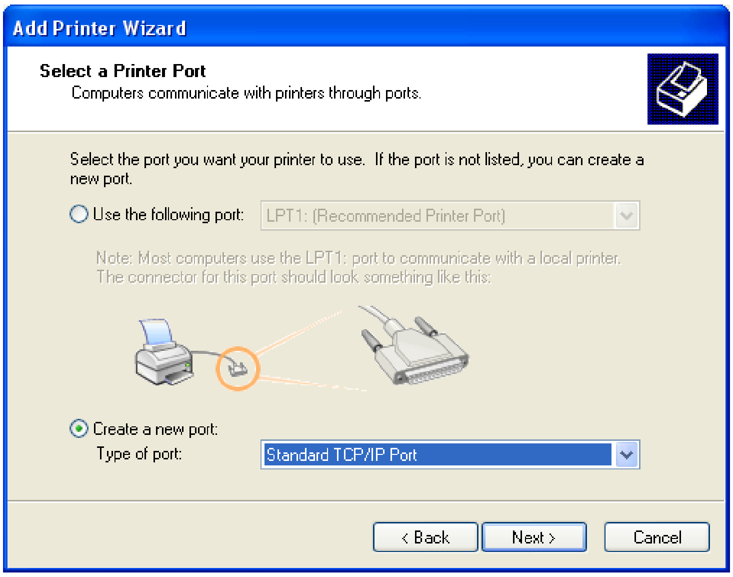
● Local printer attached to this computer

□ Automatically detect and install my Plug and Play printer

○ A network printer, or a printer attached to another computer

To set up a network printer that is not attached to a print server, use the "Local printer" option.

Select **Local Printer** and click **Next**



Add Printer Wizard

Select a Printer Port
Computers communicate with printers through ports.

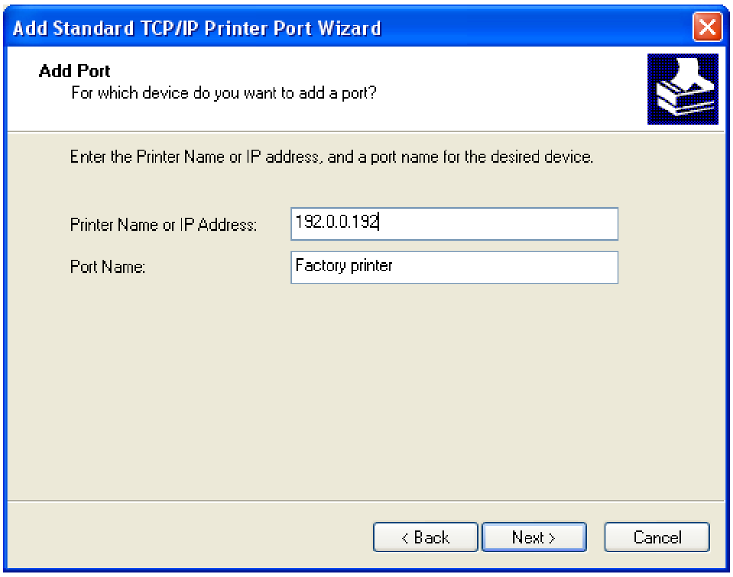
Select the port you want your printer to use. If the port is not listed, you can create a new port.

○ Use the following port: LPT1: (Recommended Printer Port)

Note: Most computers use the LPT1: port to communicate with a local printer. The connector for this port should look something like this:

● Create a new port:
Type of port: Standard TCP/IP Port

Click **Create a new port** and select **Standard TCP/IP Port** from the drop down menu



Add Standard TCP/IP Printer Port Wizard

Add Port

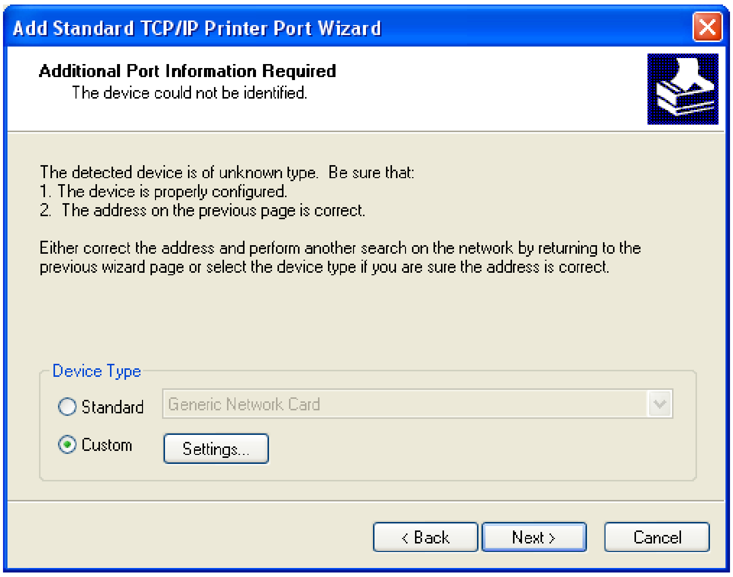
For which device do you want to add a port?

Enter the Printer Name or IP address, and a port name for the desired device.

Printer Name or IP Address: 192.0.0.192

Port Name: Factory printer

Input the Units IP address or Printer name that you have previously selected. Any descriptive printer name can be used. Click **Next**



Add Standard TCP/IP Printer Port Wizard

Additional Port Information Required

The device could not be identified.

The detected device is of unknown type. Be sure that:
1. The device is properly configured.
2. The address on the previous page is correct.

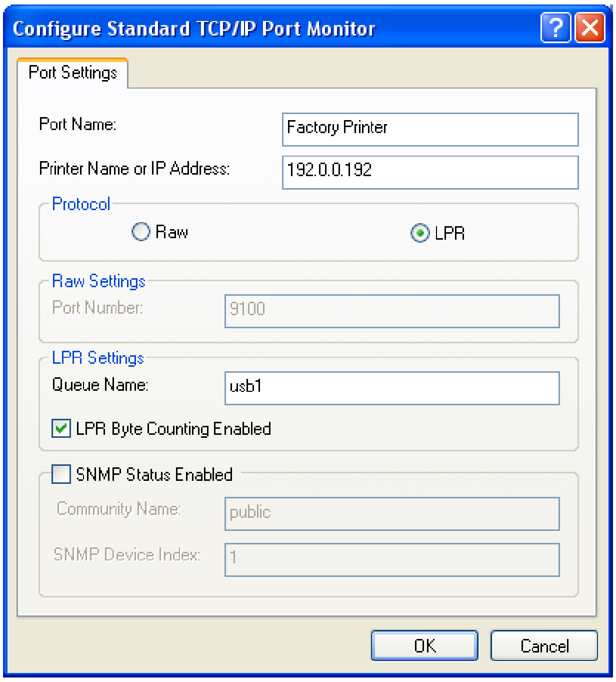
Either correct the address and perform another search on the network by returning to the previous wizard page or select the device type if you are sure the address is correct.

Device Type

○ Standard Generic Network Card

● Custom

Select **Custom** and Click **Settings**
Click **Next**



Configure Standard TCP/IP Port Monitor

Port Settings

Port Name: Factory Printer

Printer Name or IP Address: 192.0.0.192

Protocol: ○ Raw ● LPR

Raw Settings

Port Number: 9100

LPR Settings

Queue Name: usb1

☑ LPR Byte Counting Enabled

□ SNMP Status Enabled

Community Name: public

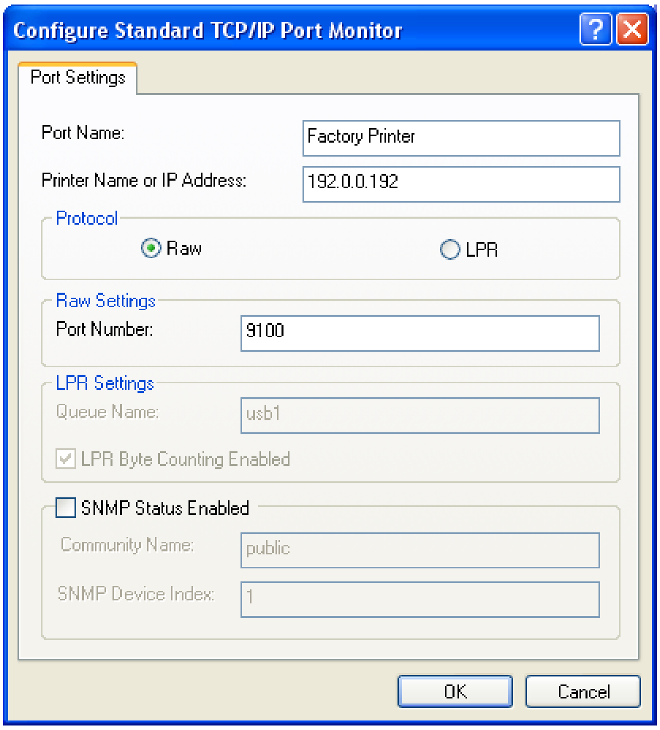
SNMP Device Index: 1

The DSM Appliance can support RAW port or LPR

If you select LPR each port on the unit is defined with a different queue name as follows:

First USB queue name is **usb1**
Second USB queue name is **usb2**
Third USB queue name is **usb3**
Fourth USB queue name is **usb4**
First network printer queue name is net1**.**

Select **LPR byte counting** and Click **OK**



Configure Standard TCP/IP Port Monitor

Port Settings

Port Name: Factory Printer

Printer Name or IP Address: 192.0.0.192

Protocol: ● Raw ○ LPR

Raw Settings

Port Number: 9100

LPR Settings

Queue Name: usb1

☑ LPR Byte Counting Enabled

□ SNMP Status Enabled

Community Name: public

SNMP Device Index: 1

If you whish to use raw port printing Select **RAW**

The following default Port numbers can be used to print from each physical output port or network printer set up on the unit. Alternatively you set none standard port number of your choice but make sure you use the same port numbering convention when setting up the queues on the DSM Appliance.

9100 = USB1
9101 = USB2
9102 = USB3
9103 = USB4
9111 – 9160 = network printers 1 – 50

e.g

Network printer 1 = port 9111
Network printer 5 = port 9115
Network printer 45 = port 9155

This completes the port setup, you can select your printer type and driver as normal.

DSM *command short reference*

Apart from the web interface command generator the full list of extra commands are listed below, these can be manually edited into the convert.ini file. You can also write an ini file of any name with the file extension .ini

Example: create myini.ini and upload it from the file manager page. To process this. myini.ini file after it has been uploaded just insert the following command into any section of the Command.ini at the point you wish to execute it.
IniFile=/etc/persistent/files/myini.ini

Full list of commands are:

C#[:#]…;	What to do on a specific copy and optional on a specific page
C#:L | R…;	What to do on the left | right pages of a specific copy
A#	Select a specific page size, like A4 or Legal etc.
B#	Select a specific page while duplex printing
D#	Select print direction, portrait, landscape, reverse ….
F[-]#	Print macro/form number # on page/copy
FF	Insert FormFeed
FN	Insert FormFeed, but DSM will not count and recognize the FormFeed
G#	Print macro/form number # on page/copy, but at the beginning of the page
H#	Draw paper from tray using PCL Sequence Esc&l#H for page/copy
I#	Draw paper from tray using PCL Sequence paper type for page/copy
J	Use Job offset in out tray
L	Add the report lines on the page
M#	Send paper to out tray #
N	Do not print page, only used with C#:#..; command
O#	Print macro/form number # on this and following pages as overlay
PL#	Long edge margin
PS#	Short edge margin
PX	Set page margins to 0
Q#	Static reprint factor per page
R	Resets the page number to 1. The next page will be treated as C1…
S#	Simplex, Duplex
X#	Don`t use this page definition on print page number #.
D#;	Default copy definition if no copy definition for actual copy is defined
EA$$$;	Inserts a full C#-Commands right in front of the first printing data
EB#;	Send form/data at beginning of job, start with PCL UEL command (CoverSheet)
EE#;	Send form/data at end of job
E3|6;	Set print data to 300 or 600 dpi., EB command is also required!
ET;	Wrap around the data stream a PJL Job.
EX$$$;	Inserts a full C#-Commands right at the end of the data
F[-]#;	Generates on the fly a new mac file
K#;	Print # copies, order 123, 123 … Use C#.. *DSM* commands if given
N1;	Passthrough data, no changes are made
N2;	No data is printed, but archiving and print to port functions still work
Q#;	Print # copies, order 111, 222 … C#.. *DSM* commands are NOT used
R#;	Print # copies, order 111, 222 … C#.. *DSM* commands are used
T#[:#];	Staple job
X;	Print *DSM* command error on console

DSM *command syntax*

To enable a DSM command you must insert the relevant command within the Convert.ini file by using the DSM appliances on board editor or send the command within the print datastream.

There are two principle methods to use a *DSM* command

1. In the data stream, the command will look like this:

   [2 TriggerStartSigns]   Command[CommandOption];[....]   [2 TriggerStopSigns]

   example using the default trigger signs: <<X;K2;D2;C1F1000;C2F2000;>>

2. Within the convert.ini file trigger signs are not required:

   Command[CommandOption];[....]

   DSM example for the usage in a command file convert.ini:
   DSM_COMMAND=X;K2;D2;C1F1000;C2F2000;

Every command starts with a command character and must be ended with a semicolon. There is no required order for the specification of a single command in the full command line.

There can be *DSM* variables used within the DSM_Command_Add and DSM_Command, but:
- The variables must be available, when reading the command.
- The commands need to edited directly in the ini file,

C#[:# | :L | :R | :X | :$-$]Command; - What to do per page

The first number # after the C defines, which printed copy of the incoming data stream the command is valid for.

As an example

C1 for the first copy, the original, C2 for the second and so on.

As an option you can add:

C#:#	Command definition #:# for the copy number # and page number # of the copy
C#:$-$	Command definition for all pages between page $ and $ for copy #
C#:L	Additional command definition for left hand pages
C#:R	Additional command definition for right hand pages
C#:X	See below.

You may define within a copy rule specific page numbers. The page number is inserted with a colon plus number: # right after the copy number definition.

Examples:

C1command;	The command is valid for each page of the first copy
C2: 1command;	The command is valid only for the first page of the second copy

Important:

Additional pages, which are inserted through called forms, **are** counted. So if you use an overlay command to add terms and conditions on the back of the page, you must add a form feed in the form data of the terms and conditions data stream. This page will be counted! If this is done on page one, the next data page will be page 3!   Page 1 data stream, Page 2 Form (back of page 1), Page 3 data stream.

If for the actual printing page a C#:# command is found, The commands C# or C#:$-$ will be not executed.

C#:L and/or C#:R definitions are always executed after the C# or C#:# command.

The Default Copy Command D#; has got the lowest priority. It will be performed on every page on a particular copy, which is not addressed with a C# and/or C#:# command;.

C#command; Is the general command, which describes which specific copy of the data stream is to be printed.

The command will always be executed for any page of the # copy, unless it is overwritten by the C#:#command; .

C#:#command; command has the highest priority, because the copy number and page number is defined.

For example C1:1command;: This command is only valid for the first page of the first copy. It will not be used on any other page on any copy with one exception! Which is, when *DSM* prints the first page of a copy, where the command line does not define a C# or C#:# command **AND** the default copy command is set to D1;

Example:
K3;**C1:1**F1000;**C1**F2000;**C2**F2000F3000;**D1**;   (note K3 denotes 3 copies of the whole document)

The first copy page one is different to all other pages of copy one. The second copy again is different, But because of the D1; command the third copy is treated the same as copy 1.

C#:$-$command;   If the Comand C#:# is not provided, then the C#:$-$ command has the second highest priority. This command is used when the copy number # is valid and the actual page number is equal or between the defined page limits.

Example:
C1:10-29S1;**C1**S0;

All pages of the first copy from page number 10 up to 29 will be printed in Duplex mode. All the other pages will be printed simplex.

C#:Lcommand; Will be executed only for left pages after the standard C#Comamnd This command will force DSM to print a form on left handed pages only.

C#:Rcommand; Same as C#:L, but for right pages

C#:Xcommand; This is a very special command. It is only executed, if an EX; (C-Command at the end of the job) or EA; (C-Command at the beginning of the job) is defined. In this case the given commands are executed directly before the EA command, so at the beginning or at the end of the print data stream.

This command is used to give the EA; and EX; command some copy specific settings.

It is also possible to use variables within DSM_Commands. One major usage of this functionality is to generate dynamic DSM_commands. The following advanced example demonstrates how some pages within a larger job can be drawn out of different paper tray, when a special word is found on that particular page:

```
[GLOBAL]
; Generate an own page counter using the Variable
; #myPageCounter# and initialize it to 1 (first page)
variable=myPageCounter:1
; Normally all pages are drwn out of tray NORMAL
DSM_COMMAND=C1I1;

[search for next page and increase page counter]
; search for formfeed and next sequenze(s) to find formfeeds
search_binary=\x0C\x1B&l1X
; As now the next page is read, increase the counter
```

```
counter=#myPageCounter#
; Speed up, do not read all keys
ReadOnlySearchKeys=ON


[Rule for pages drawn form other tray]
; There must be a special trigger text on that page
Search_Windows=Paymentform page
; in this example the tigger needs to be erased
erase_binary=1
; Any Variables in DSM_COMMANDS[_ADD] are immediately replaced
; when the command is read form the ini-file
DSM_Replace_Variables=ON
; Add to the current DSM_Command the exeption command for that
; particular page. The variable #myPageCounter# is directly
; replaced by reading the section.
DSM_command_add=c1:#myPageCounter#I9;
```

Result: If the trigger word "Paymentform page" was found on page 2, 8 and 12 then the final DSM_Command is: C1I1; C2:2I9; C2:8I9; C2:12I9;
The limitation is the maximum length of the DSM_Command with 2048 Bytes. Or with the above routine a document with about 220 payment forms.

C-Command A#; Page size selection

Larger than zero is valid. Here is a list of some paper sizes:

26	A4	27	A3	25	A5
24	A6	90	DL envelope	91	C5 envelope
100	B5 envelope	1	Executive	2	Legal
3	Letter	101	Custom		

This command is used for the paper size selection, the PCL commands ←&l##A. Any entered value

C-Command B#; Back page selection

This command is used in conjunction with the Duplex command. The following values are supported:
B0; Continue on next printing side
B1; Continue on next front side
B2; Continue on next back side

Important notes:

- This command may perform a form feed, if it does, the page will not be counted, as *DSM* counts FormFeeds! In this case no next page definitions will be performed.
- This command should always be used in conjunction with the duplex command, which needs to be defined first in the *DSM* command.

C-Command D#; Printing Direction

The following variables are valid:
D0; Portrait
D1; Landscape
D2; Reverse Portrait
D3; Reverse Landscape

C-Command F[-]#[x[,y[,d]]]; Call macro

The definition F# calls a macro/form with the ID number # on the C command defined page. Any number between 1 and 32000 is valid. But especially for SAP users, it is strongly recommended not to use any macro number below 1000.

The macros can be generated from any PCL5 .PRN file, or on the fly using the *DSM* Software.

First DSM inserts the PCL 5 command for a macro select and call. If the download function is active, and the macro file (*.mac) is installed in the device, *DSM* will automatically download the macro to the printer. If the macro cannot be found in the device and the macro is not stored in the printer, the command has no effect.

Sometimes you may wish to position the macros on the page, for example a signature. The position variable follows the macro number separated with commas F1000x,y. The first number is the x position and the second one the y position. This command remain active until changed by either using other offsets in the next F# command, or in the general PL and PS or PX commands, if valid on the next page.

Finally you may add a print direction to the macro, 0 is the default, 90, 180 or 270 degrees.   0 (normal print direction)

Notes:

- In order to work correctly, this command needs a FormFeed in the as the page separator. If you use for example if on the last page EscE instead of a FormFeed is present, there will be no forms added to that last page!

- As many F statements as you like can be added to a C command. Again note, the maximum length of the total *DSM* command is limited to 1023 bytes.

Examples:

C1:1F100F200;C1F200; *DSM* inserts on page 1 of copy 1 the forms 100 and 200, on all other pages of copy 1 the Form 200. Form 100 could be a letter head, Form 200 maybe a logo, which needs to be printed on all pages.

If the form number is provided as a negative value, *DSM* searches for the positive form, but at the end of the job, the mac/form file will be deleted! Use this method to delete temporary form files.

Forms with their ID between 25000 and 30000 will be loaded every time to the printer.

C-Command FF; FormFeed

This command inserts a FormFeed into the data stream. For example this function can be used to print a terms and condition form on the reverse side of a selected page. All forms/macros are inserted as the last function on any page. See below C-Command S for an example.

Important:

Be careful using this parameter, because a recursive situation happens very easily. So C1F2000FFF3000; is a invalid command. Form 2000 is printed on the first page, then a FormFeed, and F3000 on the second page. But in this case the second page may be interpreted as the first page by subsequent commands.

There are 3 ways to get around this:

1. If the page should only be inserted once for example on the back of the first page; then use this command: C1:1F2000FF;C1:2F3000;C1F3000;

2. The page should be inserted specifically for right hand and left hand pages: C1:RF2000FF;C1:LF3000;So on every right sited page a FormFeed is inserted, which results in a F3000 being inserted on the left hand page.

3. See next command

C-Command FN; FormFeed using next page command

This command is much the same as the formfeed, (FF command). The printer will move the cursor to the next page. The difference is, that DSM will not recognize, that the page change happened.

Coming back to the FF command example the third solution is: C1F2000FN3000;

C-Command G#; Call macro

Very much like the F command, but called at the beginning of the page. This function is similar to the F-Command, but has the benefit, that coloured forms are printed underneath the printing data.

The disadvantage is, that an additional page may be printed at the data stream end, because the command is inserted right at the beginning of the page, where it is still unknown if there is additional data to be printed. So it is wise to add a page number to the C# command as an example: C1:1G8000;

C-Command H#; Input paper tray selection

This command should only be used, if the designated printer is not able to handle the I#; command which is described in the next section.

This command is used for the paper source selection where the old PCL commands ←&l##H is applicable.

The disadvantage is, that there is no clear numbering convention of input paper trays across all PCL printers. In most cases it is different between each printer model. Usually the numbering follows the format:

1 Main paper source 2 Manual tray
3 Envelope feeder manual 5 Tray 3
6 Tray 4

If there is no printer manual available, then use MS Word and a PCL 5 printer driver and follow these steps:

1. Open a new blank document
2. File -> Document settings -> Select the proper paper input tray
3. print the empty document to a file: File -> Print -> Mark Print to File -> Ok
4. Load the print file into MS Word
5. Search for &l (lower case L) and look if there is a h or H in the Escape Sequence. The number before this character is the searched H-value for this command.

C-Command I#; Input paper tray selection

This command defines the input paper tray instead of using the PCL sequences to select a paper tray, which are unfortunately different form printer to printer, the digit following the I command defines the requested paper type. This Command is only supported by HP LaserJet Series 4000, 2100 and newer.

I# 1 : Plain 2 : Bond
3 : Color 4 : Labels
5 : Recycled 6 : Letterhead
7 : Cardstock 8 : Prepunched
9 : Preprinted 10: Transparency
11: Rough

The benefits of specifying paper types instead of tray numbers are:

- You may use any tray for a special type of paper. This gives the user the benefit, that he can load the printer with paper according to his needs. Different paper types may be required within different departments of any organisation. Example production area or Accounts department.

- If the application requires a special type of paper, you have either have different kinds of papers loaded in the printer trays, or the user is notified through the front panel of the printer, to load a special paper.
  In both cases, the printer needs to stop if the tray is empty. In order to achieve this, each tray needs to be assigned to one of the above listed paper types. If this is done the programmer doesn't need to

know, how the printer is loaded.

- The name of the paper does not need to match the real type of paper, which you want to print on. It is just a name, nothing else.

- If DSM receives the paper type selection command, the converter will erase all tray selection commands, Esc&l_H, Esc&n_Wxxxxxx and Esc&l_M.

- The Esc&l0H remains unchanged in the data stream (eject page).

Example: C1:1I2;C1I3;C2I3F1000;
Page 1 of copy 1 is drawn out of tray bond. The rest of the pages of copy 1 and all pages from copy 2 are drawn out of tray Colour. In addition C2 prints watermark macro 1000 on every page.

C-Command J; Job Offset

A J in any C-command enables the Job Offset mode. There are no arguments needed.

If the command is not given, the data stream will not be changed. If the command is given, the converter will erase all PCL 5 job offset commands, Esc&l_T and turn the Job Offset on!

It is enough to send this command only once on any page.

Please note, that not all printers are able to offset print jobs.

C-Command L; Report Lines Generator

Inserts the usually grey report line using the build in generator.

C-Command M#; Paper out tray selection

M# Output (Media) Bin Selection
0 : Automatic selection
1 : Upper bin
2 : Left or Rear bin
3 to 15 selects the destination bin of the mailbox or stapler bin.

If the C-command is not given, the data stream will not be changed. If the command is given, the *DSM* print processor will erase all paper out tray selection commands, Esc&l_G

C#:#-Command N; No Page print

The N command will advise *DSM* to not print the page. The following notes are important for this command:

- Use the command only on a special page of a copy. If it is the first page of the first copy, then use C1:1N;
- This command does not work for the last page of a job. However, in this case, instead of the data, a blank page will be printed.
- Always use the N command on a specific page. The command C1N; does not make sense, because there will be only a blank page printed for the first copy.
- This function is not supported in conjunction with the R command.

C#:#-Command N1; Ignore FormFeed

The N1 command can be used to combine 2 pages to one page.

C-Command O[-]#; Overlay

In principle the overlay function has the same functionality as the C-command F#, but:

- The form will be automatically called on every page, the printer feeds.

- The value of 0 disables the actual enabled overlay macro.

- Only one Overlay command can be used in any C command. If you want to call more then one, generate by hand a new macro, which calls the other ones.

If the form number is provided as a negative value, *DSM* searches for the positive form, but at the end of the job, the mac/form file will be deleted! Use this method to delete temporary form files, generated with the F# command

C-Command PL#; Long edge positioning

Any positive or negative digit will move the long-edge origin of the page coordinating system to the new value.

A value of 0 sets the default value.

Negative values move the origin out of the page. But note that printing information can be lost, when the origin is positioned outside the unprintable area.

Positive values move the origin to the right. Printing information can be lost on the other side of the page, when printed outside the unprintable area.

If the command is not given, the data stream will not be changed. If the command is recieved, the converter will erase all related sequences in the data stream.

Example: C1PL300;
Printing for example a left bounded ASCII data stream, this command will move the whole text on an A4 portrait page 300 units to the right.

C-Command PS#; Short edge positioning

Same functionality as PL#, but on the short edge of the paper.

C-Command PX; sets PL#; and PS#; to 0,0

If this command is recieved, the previous defined PL# and PS# moves will be set to 0,0 right before the formfeed and before any macro using C-command F# is called.

If this command is used on its own, any Esc&lU or Esc&lZ in the data stream will not be erased!

C-Command Q#; Single page reprint factor

The reprint factor must be a positive digit larger then 0.

This command reprints a single page # times, as it is defined 1:1, no additional forms or other DSM_Commands are issued.

The command is normally not used in duplex mode, as the pages are simply copied by the printer. For a page counter, those additional pages are NOT counted!

The commands should be used very carefully in conjunction with the page copy commands K#; Q#; and R#;

Example: K1;C1:1Q2;C1:2Q3;
The complete document is printed once K1; but the first page will be printed 2 times and the second (and all following pages) 3 times.

C-Command R; Reset Page Count

This command resets the page counter. It can be used to create a page number based recursive DSM Command actions. But note: If accounting is enabled, this command will lead to wrong accounting results.

Example: A data stream is endless. It contains n times 3 pages, and the pages have to be drawn out of different paper trays.

```
[GLOBAL]
; The 3P^rdP page will reset the page counter to 1.
DSM_COMMAND=C1:1H4;C1:2H1;C1:3H6R;
```

The result of the R command in the definition of the 3P^{rdP} page is, that the 4P^{thP} page is performed as it is would be the first page, the 6P^{thP} also again as the first etc.

This method only works for jobs which have constantly the same amount of pages. If this is not the case, then the new *DSM* Escape Sequence <27>)s-995Z can also be used on the last page of a section. Whenever this sequence is found, DSM will then reset the page counter on the next page to 1.

Example:
Invoice printing by a huge data stream with a lot of invoices: On the first page form 2000 is requested on all the other pages form 3000. If the text “Payment period 30 days” is found on the page, the next page will be again a page 1:

```
[GLOBAL]
DSM_COMMAND=C1:1F2000;C1F3000;

[Search for last page of invoice]
Search_Binary=Payment period 30 days
Add_Binary=\x1B)s-995Z
ReadOnlySearchKeys=ON
```

Doing it this way, DSM will loose the total page counter.

C-Command S#; Simplex / Duplex

This C-command sets the printer in the simplex or duplex mode. The following parameters are possible:

#	0 : Simplex
	1 : Duplex long side
	2 : Duplex short side

Notes:

- Any other # number apart from 0,1,2 are ignored.
- Once this command is used, all PCL commands for simplex/duplex in the data stream are erased. This means, if you want to set the complete job in duplex, then simply turn it on the first page C1:1S2.
- A duplex command needs to be always sent on the front page.
- Turn off duplex printing also on the front page!
- You may use C-Command B to select the next printing page

Example: C1:1F1100S1FFF1000;C1:3S0;

This example demonstrates how to print terms and conditions on the first page only. Copy 1 page 1 starts printing in duplex S1 before the page form feed, the macro 1100 is inserted, then the FormFeed FF and on the new page 2 the macro 1000. After that point the original FormFeed from page 1 is performed. So the next page is number 3 and the duplex mode is turned off.

C-Command X#; Don`t use this C Command on print page

This C-command is used to suppress the command on a specific page. It is important to use the command right after the C-definition.

Example: C1:1S1F5000;C1:RX1S1F50001;

The purpose of this example is to print a duplex job. On the first page the letterhead form 5000 needs to be printed. No forms are printed on the back pages. Finally a different logo needs to be printed on each following front page.

C1:1S1F5000;   Page 1 is duplex and Form 5000 used
C1:RX1S1F50001; All right pages, except page 1, are duplex and use form 50001

D#; - Default Copy Command

The command value # defines, which C# command definition will be selected by default, when there is no C-command defined for the actual copy number.

Example (without the C-command descriptions): <<K4;C1...;C2...;D2;>>
This command will print the data stream 4 times K4. For the first copy the definition of C1 is used, for the second C2. The D2; command now defines, that if there is no page copy description, the description of the number of the D command is used. In our example, the copies 3 and 4 are treated the same way, as copy 2.

EA$$$$; - Send a C-Command right before the first print data

This command is used for initializing the printer. The $$$$ can be any combinations of C# commands, like macros, simplex/duplex commands etc.

Example 1: EAS2B2F4711;
This command will set the data stream to duplex printing, and selects the reverse side as the first printing page. Also on this back page the form 4711 is printed. If in the data stream after this command no PCL command which will perform a FormFeed is found, the following data starts printing on the reverse of the page.

Example 1: EAS2B2F4711FF;C1:2S0;
The same command as the previous example, but the form will be printed on the first page only, maybe the forms are using the same variables for printing the user name and print time, but then the form feed tells the printer to feed the page. As the B2 command is not performing a formfeed, the page counter is set to 2 after the FF command. The C1:2 command will set the rest of the data stream to simplex printing.

The following command does the same, but as 2 FormFeeds are sent, the empty first page is counted:
EAS2FFF4711FF;C1:3S0;

Notes:

- The major difference compared to the EB command is, the EB loads a form in front of the data stream; The EA command can then set the printer within the data stream to any initialisation; The K-copy command does even reprint this initialisation, which is not the case for the EB command.
- Any page initialisation of C1 or C1:1 command will be sent out before the EA commands.
- In order to enable copy specific tray selections, the Command C#:X is searched and if defined executed right before the EX commands.
- You may use several EA statements in a command line.

These commands set the printer to a specific resolution, 300 or 600 dpi.

E3; - Select 300 dpi resolution

E6; - Select 600 dpi resolution

The commands can only be used in conjunction with an EB#; If you do not want to call a macro right at the beginning of the data stream, please enter a unused macro number, like for example 9999.

EB#; - Send a form/job at the beginning of complete job

EE#; - Send a form/job after the end of the complete job

The # in the command is the macro/form ID number, which is called right before = EB and right after = EE the provided data stream (See F command).

In most cases, the EB command is used to initialise the real data stream, like the job_ini.mac, or downloading forms or setting the printer into a specific state. But it can also be used as a cover sheet, in this case do not convert a print file into a macro, as the job itself is the macro including all initialisations, so print your form to file and store it with the File management page.

Example: EB1001;   Inserts the form 1001 at the beginning of the print job.

Notes for PCL data streams:

- The EB and EE command can be used several times in the DSM_Command. Example: The intension is to have several items at the beginning of the data stream specific fonts, like normal or bold, depending on the real adressed fonts.
- The macro ID number must be larger then 0 and less then 32000.
- There is no other *DSM* Command performed on the 2 inserted jobs. This means, the first *DSM* page is the first page of the data stream. The EE pages do not count.
- The Search Binary and Windows definitions in the ini-file are performed on these macros, as well as the variables replacements.
- There will be an UEL inserted right at beginning of the data stream.
- If the data stream itself contains an UEL (Esc%-12345X) or a PCL Reset (EscE) statement, this method can’t be used to set the printer.

EI#$$$; - Performs a DSM-Command at the beginning of the # copy

This command is performed right before the # reprint copy of the job. This command is only performed if the K#; command is provided.

Example: K300;EI50I9FF;
There are 300 copies in order 123… 123… 123… printed. After every 50P^{thP} copy an empty sheet is drawn form paper try with media type preprinted.

ET[J]; - Wrap in PJL JOB statements

This command should be used, if the data stream provides only ASCII characters, and doesn’t initialise the job correctly: No BEGIN and no END-OF-JOB Sequences.

ET; wraps the data stream into well defined PCL 5 PJL commands. As the result the printer treats the output stream as one job.

ETJ;K#; The parameter J is only usable in conjunction with the K#; parameter. If both are used, *DSM* does wrap each copy job independently. Maybe used to staple each job together, if the job itself is not well bounded.

EX$$$$; - Send a C-Commands right at the end of print data

This command is used to add forms at the end of the data stream. The syntax of the command is similar to the EA command. $$$$ Can be any combinations of C# commands, like macros, simplex/duplex commands etc.

However you must always use C-Command FF for ending the last page.

Notes:

- This command is only activated before the reset sequence EscE or before the UEL commands Esc%-12345X, which ever comes first.
- In order to enable copy specific tray selections, the Command C#:X is searched and if defined executed right before the EX commands.
- You may use several EX statements in a command line, if they are combined during the processing with DSM_COMAMND_ADD.
- You may also combine several pages even from different trays using duplex.
  Example: EXS2I9F1000FFF1001FFF1002FFF1003FF.
  The macros 1000 to 1003 are added at the end of the job, seperated by FF, so each macro is printing on one page, using long edge duplex printing and finally the paper is drawn from paper tray with media type preprinted.

F#; - Generate a new mac file on the fly

This command will store the actual data stream up to the first form feed into a macro file named #.mac. You may use this command to generate new forms on the fly, without using DSM File management page. It can also be used to select printing pages and combine them to one print job.

So for example F4711; will generate the file 4711.mac in the working directory

Notes:

- The print file is only stored up to the first form feed in the printing file.
- Like the DSM macro conversion on the web interface, all escape sequences, which may cause form feeds are erased, even when they are defined in the *DSM*-Command! Example 1: F4711;C1S2I5; file 4711.mac is created, but all C1 commands are ignored.

- Example 2: F4712;C1:F1999; file 4711.mac is created, and at the end of the new form, the form 1999 is called. Note: PCL allows only 2 levels of nesting.
- Previous stored files are overwritten, without any warning!
- If this printing file needs to be called by another print file, then use simply the newly predefined number.

K#; - Document reprint factor

The positive number # tells the system how often the complete data stream has to be printed, original + reprints. The minimum value is 1 for the original. If there is no K,Q or R factor in the *DSM* command defined, K1 is assumed.

Example:
The page printing order of a document with 3 pages using the K4; command is like this: 123 123 123 123

Notes:

- If the copy number is more then 1, the total size of the data stream is limited to the heap memory available in the system. By default the size is limited to 1 Mbytes.
- If the 1MB is not enough, then use *DSM*_MAX_HEAP key in the *DSM* ini File to redefine the maximum available heap to any size.
- The K#; should only be used once in the *DSM* command line.
- If any valid number is given, the PCL copy command Esc&l_X is ignored
- As each page is stored as a macro, do not define any additional macro within the data stream.
  For unix: when JetMobile™ barcodes are used on the page, therefore in definit.h remark out the definition: TEXT_WITH_MACRO. It is not enough to set the definition value to 0.
  For MS Windows: If JetMobile™ barcodes are needed, do not print the barcode with readable text, or ask your local distributor to send you a system with the TEXT_WITHMACRO turned off.

- The value can be overwritten using those keys: DSM_SET_COPYFACTOR and DSM_USE_QUATITY_FACTOR

N1; - Passthrough

This command will switch the *DSM* functionality to passthrough. There will be no changes made on the data stream beside to present the content of the file job_ini.mac, if the file exists.

N2; - NoPrinting

DSM will not print any data. This *DSM* command can be used to avoid the print out, but in conjunction with archiving the data can still be sent to another directory or port, like fax. (these features will be added to DSM at a later date)

Q#; - Pages reprint factor

The positive number # tells the system how often the complete data stream has to be printed. The minimum and default value is 1. This command should not be used in conjunction with the R#; or K#; command.

K factor can be overwritten by a copy factor defined in the print file. Another way to set the value is to use the Key: DSM_USE_QUATITY_FACTOR

Compared to K#; parameter, this Q#; parameter works totally differently. The document will be copied using the printers copy function. This means, that the data stream is sent only once to the printer.

Example:
The page printing order of a document with 3 pages using the K4; command is like this: 1111 2222 3333

Note: Only the C1[:#]commands can be used, all other C commands for copies higher then 1 are ignored!

R#; - Pages reprint factor, with changeable *DSM* content

The R#; command does produce the same page print order as the Q#; command. But on each printed page/copy you may select any C#[:#] command.

K factor can be overwritten by a copy factor defined in the print file.

Another way to set the Value is to use the Key: DSM_USE_QUATITY_FACTOR

T#[:#]; - Staple print job with stapler/stacker

The T#; command is used to staple a print job. Once used the PJL commands for stapling are inserted or deleted.

Value	Finicher	Description
-1	HP Standard	One staple is sticked top left of paper with an angle
0	HP Standard	Do not staple job
1	HP Standard	One staple is stuck on top left of paper
2	HP Standard	Two staples are stuck into the paper
3	HP Standard	Three staples are stuck into the paper

6	HP Standard	Six staples are stuck into the paper
100	HP Multifunc.	Left bin Face up
101	HP Multifunc.	Left bin Face down
102	HP Multifunc.	One staple stuck at top left
103	HP Multifunc.	Two staples stuck at left border.
104	HP Multifunc.	Three staples stuck at left border.
105	HP Multifunc.	Angled stapled top left corner
106	HP Multifunc.	Custom stapling
107	HP Multifunc.	Six staples stuck at left border.
108	HP Multifunc.	Front Staple
109	HP Multifunc.	Rear Staple
110	HP Multifunc.	Angled forced
111	HP Multifunc.	Booklet maker
200	Ricoh	One staple is stuck on top left of paper
201	Ricoh	One staple is stuck on top right of paper
202	Ricoh	Two staples stuck at left border.
203	Ricoh	Two staples stuck at top border.
204	Ricoh	Two staples stuck at right border.
205	Ricoh	No punching holes
206	Ricoh	2 US punches left at left border
207	Ricoh	2 US punches left at top border
208	Ricoh	2 US punches left at right border
209	Ricoh	4 US punches left at left border
210	Ricoh	4 US punches left at top border
211	Ricoh	4 US punches left at right border
212	Ricoh	2 JP punches left at left border
213	Ricoh	2 JP punches left at top border
214	Ricoh	2 JP punches left at right border
215	Ricoh	4 EU punches left at left border
216	Ricoh	4 EU punches left at top border
217	Ricoh	4 EU punches left at right border
218	Ricoh	4 NEU punches left at left border
219	Ricoh	4 NEU punches left at top border
219	Ricoh	4 NEU punches left at right border

Be sure, that the outbin escape sequence in the data stream is set to the correct outbin! If there is a *DSM* command, which changes the outbin to a non-stapling bin, the T command will have no effect.

"Split files " for how to split an ASCII like data stream into several, print jobs, for example to staple each job (invoice).

For mail merge support, the option: # can be added. This static option defines how many pages should be stapled. Usually it is the amount of pages per mail. If you intend to print mail merge stapled document with Microsoft word, then please have a look at the Microsoft Word chapter in the installation manual.

Notes:

1. In order to work correctly, a print job needs to begin with an UEL command (Esc%-12345X). If you are not sure, then load the print file into any editor. If this command is missing, use the ET; *DSM* command.

2. Keep in mind that a defined *DSM* command in the data stream is sent with every printed record, but *DSM* looks only for the first true command. In order to not print the commands on the second and following records, either print them in white colour, put the print command into a contraction: “IF first record then print command, else do nothing”, or use the IniFilemethod.

3. Mail merge with variable page numbers is NOT supported! Each record needs to print Tx:# pages.

4. There should be no additional UEL statements in the middle of the data stream.

5. SPEED: In order to enhance speed:
   A) Use macros for standard parts of mail merge form

B) from the programmer standpoint, some speed tuning could be made, but this needs to be weighed up against cost.

Example: Your MS Word mail merge document contains 3 pages, and you want to staple all 3 pages together, then your command may look like this: T-1:3;

Programmers hints:

You can find a short PCL 5 command sequence listing at www.hp.com.

Once defined, some of the DSM page commands remain active until they are changed, turned off, or the PCL reset command is performed.

For example: Once set to A4 the paper size. It remains A4 until the end of job, or it is changed. For a pure ASCII data stream this page size can survive all copies. This may become critical, when your print has an odd amount of pages and needs to be printed in duplex. So the first page of the second copy will be on a back page. In this case enter the command ET; which will add a separator between the copies.

As a second example the single page reprint factor will also be set for the actual page, but remain active on all following pages.

On the first page it is a bit tricky to insert commands like simplex, duplex, tray selections etc. Inserted at the wrong place, the commands may cause the printing of blank pages or printing one page on 2 pages. In order to avoid this, we have listed the search methods for the insertion of those commands.

1. The program first looks for a PCL page definition command. They all start with <Esc>&l. If such a sequence is found, the *DSM* software will integrate the requested sequences right before the first appearance of this sequence.

2. The second best place will be right after the first appearance of the reset sequence <Esc>E. This sequence should be right at the beginning of the data, maybe after the PJL commands. If there is data before the reset statement, they will be printed and not recognized by the software!

3. Right after the PJL or UEL commands.

4. Direct in front of the data stream, ASCII like data stream is assumed

5. If some commands are missing, you may use the ini file keys DSM_Init_PCL and/or DSM_PageInit_PCL in order to insert any valid text / command at the beginning of the data stream or page.

6. Also if for example a FormFeed is missing at the end, use the DSM_Exit_PCL key to add any requested PCL command.

Get copy factors from print file

There are two ways to change or set the *DSM* Commands K, R or Q from the definition value to a value, which is defined in the data stream.

1. Some drivers and applications let the printer perform automatically the copies by sending down an PJL Command called: @PJL SET QTY=#.

First check if the command is in the data stream, by printing to file and loading the file into any texteditor. At the very beginning of the print file all PJL commands are listed. Check if the SER QTY command is used.

If this is the case, enable this function by using key *DSM*_USE_QTY_FACTOR in the ini-file in the sections Global, PrinterName, PrinterDriverName or UserName.

Once this key is provided, *DSM* checks in the PJL commands for an entry @PJL SET QTY=#. The copy factor K, R or Q is then replaced by the # factor. If no copy factor is provided, the K factor is used.

Note: If there is a later *DSM* command change, the value can be overwritten again.

2. Most windows driver uses the PCL command Escape-sign&l#X, where the copy number is coded in the # before X.

In order to trigger on those commands the search keys can be used:

```
[Search_Copy for 2 copies]
Search_Binary=\x1B&l2X
DSM_COMMAND=K2;C1I9;C2I1;

[Search_Copy for 3 copies]
Search_Binary=\x1B&l3X
DSM_COMMAND=K3;C1I9;C2I1;D2;
…
```

If other conditions set different DSM_Comamnds, please use variables to set the copy factor. Example:

```
[Global]
DSM=ON
; Preset the variable to 1 copy
Variable=MyCopyFactor=1

[Search_Copy for 2 copies]
Search_Binary=\x1B&l2X
Variable=MyCopyFactor=2

[Search_Copy for 3 copies]
Search_Binary=\x1B&l3X
Variable=MyCopyFactor=3
…

[Trigger Invoice]
Trigger_Windows=Invocie
DSM_COMMAND=K#MyCopyFactor#;C1I9F500;C2I1F500F200;D2
```

Known problems

In the case of an environment variable being set in the middle of the data stream in a way that the data before this sequence is treated differently to data after this sequence, the reprint of this page may fail.

Example: You start with Carriage Return and Line Feed printing with the default setting. Some CR, LF do appear, then you switch to CR=CR+LF, for the rest of the page to be printed. For the reprint of the page, this function is still enabled. So there is an additional line feed at the beginning of the document for each CR, this would be wrong.

PCL 5c = Printing Watermarks on Colour LaserJet is currently not supported if you overlay user text. For the first page there is an exception. See G-Command.

Newer implementations of Mircosoft Excel do not use printer-internal fonts anymore (only Truetype fonts). Therefore for Excel no data stream triggering for printer terms can take place, if you require this feature, then it should be printed on a special dedicated queue, in which the *DSM* command is located in the section linked with the driver name.

It is not possible to print from an Adobe Acrobat PDF Writer or Distiller Port!

DSM examples

<<C1:1O2000;>>

On the very first printed page C1:1 an overlay 2000 is invoked. So the form 2000 is printed on every page of the print job.

<<X;K2;D1;C1:1F100F200O10S0I8M10;C1F11S1JPS-12;>>

X: will display any error
K2: reprints the document twice
D1: If more then one page is provided, the second and following pages are treated according to the C1 command.
As C2 is not provided, the second copy will have the same attributes like the first one.
C1:1: First copy first page is defined with Form 100 and 2000 Overlay 10 Simplex Input Tray PrePunched and Output Tray 10.
Because of the D-command, this C-command is executed on every first page of every copy.
C1: All remaining Pages: Form 11, Duplex is turned on, Job Offset and short edge offset of minus 12.

Rule based printing, used in the *DSM* Ini command file

Direct after the start of the *DSM* Server Print Processor, the software tries to read the content of the provided IniFile, which is supposed to hold all instruction on what to do. The file can also be part of the data stream, this feature is more for programmers, see further below in this chapter.

General structure and syntax of ini file.

The *DSM* ini file is build upon the same syntax as any windows ini file. The different components are:

[section] Every project is coded as a section, wrapped in [and]. Every Inifile should at least define the [GLOBAL] section. This section is performed right at the beginning of the *DSM* process. Here you can define all common actions.
Blanks are allowed in the section name.

; Any text If a command line starts with a semicolon, the line is ignored. This statement can be used for documentation.

Key=Argument The content of a section are a list of predefined keys, which are examined by the *DSM* server.

Notes:
Some keys like trigger can be used in one section several times.
As leading and following blanks are cut off by the ini handler software, wrap the arguments in "".

Examples:

```
SEARCH_BINARY= SearchData
; With additional blanks
TRIGGER_BINARY=" SearchData "
; With FormFeed
ADD_BINARY=EndOfFile\0x0C
ADD_BINARY=" EndOfFile\0x0C"
```

The section [GLOBAL]

This section is first read, direct after the program call. It lists all global commands for the *DSM* server application. These command arguments can be changed in any other sections, if those sections have trigger or search entries, and their arguments are found in the data stream

As all settings of the section [GLOBAL] are performed at program start, do not insert a Search or Trigger key in this section.
The valid keys are all listed in the table in the next section of the manual. Do not use PASSTHROUGH in the GLOBAL section.

The section [PrinterName]

This section is read/executed by the *DSM* software right after the program start, and after the commands in the GLOBAL section.

The commands in this [PRINTERNAME] section are handled like any other command! This means, that any Search and/or Trigger commands are examined and if found in the data stream, the section is executed.

So there are three kinds of keys, which go typically into this section:

1. Special ON/OFF settings, which may overwrite the GLOBAL settings.
2. Direct commands like *DSM*_COMMAND=K2;. Which need to be performed everytime a data stream is sent through the queue.
3. The Key INFILE=....
   With this command, the active ini-File is changed to a new file, its GLOBAL section is read, and all Search and Trigger Keys are activated. The old keys from the first ini-File are not valid anymore.
   Please have a look on the next page for further key information.

Notes:
- Do not use any Search and Trigger keys in this section
- If your commands are not recognized, then check the section name. If you are not sure about the printer name, then
  - view last LogFile

The section [UserName]

This is the 3PPP^{rdPPP} section that *DSM* looks after right at the beginning of job process. Actually it does the same as the previous section, but based upon the user logon name.

- DSM only processes commands if the user name is found in the datastream.

Example:

```
[Greta Garbos XP]
DSM_COMMAND=C1:1O1000;
; Prints on all pages form 1000 as an overlay
```

The section [Printer Driver Name]

This is the 4th section which *DSM* looks during the beginning of job process. Again, it does the same as the previous section, but based upon the name of the [windows] printer driver.

To find the printer driver name, open the printer settings go to the enhanced tab, the driver name is displayed.

Example:

```
[hp LaserJet 1300 PCL5e]
DSM_COMMAND=C1:1O1000;
; Prints on all pages form 1000 as an overlay]
```

Ini File key: SEARCH_BINARY

While reading the incoming data stream, this command searches using a binary comparison for the argument. Once found, one the following keys, listed in their priorities, are performed:

Replace_Binary	The search argument is replaced by the argument
Insert_Binary	Inserts the argument before the search argument
Erase_Binary and	Deletes the search argument from the data stream
Add_Binary	Adds its argument after the search argument
ReplaceNextTerm	Is used to search for a string only once and replace it with the provided second string
StoreNextWordToVariable	reads the next word into a provided variable
StoreNextWordToIntVariable	reads the next word into a provided integer variable
SkipUntilEnd	Will ignore all following bytes, until data stream end.
	Plus all available keys in the rule section.

Notes:

- If your ini file does have several sections with SEARCH keys, then be very careful when using the REPLACE, INSERT and ADD keys, in order to prevent a recursion.
- The processing of the argument is handled case sensitive.
- You may add several SEARCH_BINARY key statements within one section. The *DSM* server will search for all definitions. If only one of them is found, all keys of the section are performed. This means, that several SEARCH_BINARY commands in one section are OR combined, and that only one REPLACE_BINARY or INSERT_BINARY ADD_BINARY and ERASE_BINARY is allowed per section.
- SEARCH_BINARY is after the initialisation (read ini file) the first routine performed by the *DSM* Server.
- Do not use this key in the section GLOBAL. This is meaningless.
- Use before REPLACE and the other text editing keys are performed, *DSM* does scan the active section for other action keys, like sending the data to different ports, archive etc. (not implement in this version of DSM)
- For MS windows driver generated data streams use SEARCH_WINDOWS
- It is not recommended to use the command also for information in macro files.

Examples:

```
[Section_SEARCH_AND_PERFORM1]
SEARCH_BINARY="Invoice Number:  "
DSM_COMMAND=K3;C1F1000;C2F1100;D2
```

If the length of the actual data stream is less than 1MB, and it contains the text Invoice Number: followed with a leading blank, the *DSM* command file is performed.

```
[Section_SEARCH_AND_PERFORM2]
SEARCH_BINARY=Stra~e
FREE_ESCAPE=OFF
```

This section turns Free-Escape off, when data stream looks like using IS0-7 bit German symbol set.

Important: If the size of the print data is larger, the *DSM* command is performed partly. Actual this is an indefinite program state and should be avoided.

Notes:
Unfortunately newer Microsoft™ Windows drivers print in a way, that it is very difficult if not impossible to find any reasonable text. (eg. If everything is printed using graphic formats). DSM has several mechanisms for getting text from single positioned characters together into their words. (In these types of Files words do not exist as contiguous characters) (See search_windows or search_text keys). If DSM is still not triggering on your searched data, then you may use for the key temporarily for further analysis PRINTONLYASCII in the GLOBAL section of the ini-File.

More about the way to find the search item / trigger in the data stream, see next page.

DSM will try to not print any PCL sequences or binary data. Also some formatting will be performed for better readability, but the text may still not be searchable.

In order to increase the process speed you may limit the evaluation of DSM activation keys to Add, Insert, Replace, Erase and StoreNextWordToVariable or StoreNextWordToIntVariable by setting the key ReadOnlySearchKeys to ON. In this case none of the other keys like DSM_COMMAND, Archive etc. are read.

If the searched section should only be performed one time, actually the first time the searched item is found, then add the key Search_Only_Once to the same section and turn it ON.

Ini File key: SEARCH_TEXT

This command performs the same functionality as the SEARCH_BINARY key with two differences:

1. The comparison between the searched text and the data stream text is performed case insensitive.

   Example: SEARCH_TEXT=Greta Garbo will be true for all following data stream contents: greta garbo; Greta Garbo; Greta GARBO etc.

2. In addition this key ignores any positioning Escape Sequences (Esc*p#X or Esc*p#Y), the windows driver might generate between each printed character. Here are some tips to make the trigger findable in the data stream, especially if you are not able to interpret a PCL data stream:

   - Do not set the driver to print Truetype as graphics.
   - The trigger word(s) should be formatted if possible with a printer internal font.

If *DSM* does not finding your trigger/search words at all, the reason could be that newer HP windows drivers are installed or drivers provided by Microsoft, as they do not use the same characters in their ASCII codes, so this is not easy to scan them for a unique string.

Ini File key: SEARCH_WINDOWS

This key performs like SEARCH_TEXT, exception, that the search string is case sensitive to the key value.

Ini File key: SEARCH_ROWNO

The Syntax for this key is:

Search_RowNo=RowNumber[,ColumNumber[,PageNumber[,Length, VarName]]]

This command is used as the other search keys usually in a stand-alone section, but only the key ADD_BINARY or the store to variable option is interpreted.

It should only be used for pure ASCII print files, because it counts the print rows by searching for LineFeeds. The counter is set to 1 at the beginning of each page, found by a FormFeed character Ascii 12.

Important information:
A) If a Search_RowNo key is used the first time in a second ini-file, AND
B) This ini-file is called by the main inifile AND
C) In the main ini-file there is no Search_RowNo AND
D) The new ini-file is called by a Search_XXX rule,

THEN the row and column and page counters are reset to 1. In other words, the actual processed row which performed the rule search_Binary -> ini-file reading -> new Search_Row key will be row Nr. 1 at Col Nr. 1 on page 1. (Actually it is not at the actual position, it is at the beginning of the actual processed data stream.
In order to avoid that, simply insert in the main ini-file a dummy section, with a dummy search_rowno command.

Once *DSM* begins to evaluate a specific row, indicated by the first operand of the keys data entry, this happens:

Only RowNumber is provided and NO ColumnNumber is defined

The ADD_Binary ini-file key is searched in the provided section and performed right at the beginning of the row.

A RowNumber is provided and the ColumnNumber is larger then 0

The ADD_Binary ini-file key is searched in the provided section and performed right at that column.
A ColumnNumber of 0 will lead to ignore the whole command.
If the row does have fewer columns, then the command is ignored as well.

Like above, but also a page number is provided

If the page number is 0, the ADD_Binary ini-file key is searched in the provided section and performed right at that column for every page. If the PageNumber is greater then 0, then the Add_Binary is only performed on the provided page number.

Like above, page number, length and variable Name is provided

The next length bytes of the print data stream is stored into the provided *DSM* variable. This variable can later be used and positioned in forms, or for searches in assigned databases.
The Bytes are only read until the provided length is reached, or any byte below ASCII blank is found.
The key Erase_Binary=1 can be used to blank the length Bytes out.
The key Add_Binary should not be used, as his entry would be inserted after the first and not after the length byte.
Any blanks at the end of the term are not stored into the variable.

Note: The command can't be used at the beginning of a Row, which is also modified with key ADD_AtNextLine!

The following example will print line 8 of the document in 14pt Universe bold, and switches back to the standard font on the next line. In order not to screw up the line spacing, 6lpi is maintained.

```
[PrintLine8Bold]
SEARCH_ROWNO=8
ADD_BINARY=\x=E\x1B)s1p14vs3b4148T\x1B&l6D
ReadOnlySearchKeys=ON

[PrintLine9StandardFont]
SEARCH_ROWNO=9
ADD_BINARY=\x0F
ReadOnlySearchKeys= ON
```

The next example reads the bytes 20 to 35 out of the data stream on page 1 in row 11 int the variable #test#:

```
[Get Adress]
SEARCH_ROWNO=11,20,1,15,#test#
```

Ini File key: ERASE_BINARY

This command is performed, if the value of the SEARCH_xxxx key in the same section is found in the data stream.

Valid arguments are

1 Erase the searched information from the data stream
0 Leave the searched information in the data stream, (default)

Example:

```
[Section_SEARCH_AND_DELETE]
SEARCH_BINARY=!K TRIGGER!
ERASE_BINARY=1
DSM_COMMAND=K3;C1F1000;C2F1100;D2
```

The data stream is searched for "!K TRIGGER". Once the trigger is found, it is erased and the *DSM*_COMMAND performed.

Ini File key: ADD_BINARY

If the one Search_Binary, Search_Text, Search_Windows, Search_RowNo, OutSearch_Binary, OutSearch_Text or OutSearch_Windows key is found in the data stream, and this key is provided in the same section, the argument of the ADD_BINARY key is inserted into the data stream right after the searched argument.

Examples:

```
[Section_SEARCH_AND_ADD]
;\x1B is the hex representation of ASCII 27 = Escape Sign
SEARCH_BINARY=\x1BE
;Add direct after the PCL reset command a paper tray command.
ADD_BINARY=\&n13WdTransparency
ReadOnlySearchKeys=ON

[Section_SEARCH_AND_ADD_AFTER]
SEARCH_TEXT="Rechnung No: "
;If the print data contains the searched text, DSM needs to put ;on all pages the same
;Letter head form, form number 100 using DSM_COMMANDS in Data
;Stream -> highest priority
ADD_BINARY=<< C1:1O100; >>
ReadOnlySearchKeys=ON
```

The same functionality can be achieved using the direct PCL command for macro Overlay form 100: Add_Binary=\x1B&f100y4X, or DSM_Command=C1:1O100;. But for the second way the ReadOnlySearchKeys=ON key has to be turned OF or simply not used.

Note: Instead of using the ERASE_BINARY and ADD_BINARY keys, please use REPLACE_BINARY.

Ini File key: ADD_ATNEXTLINE

If the one Search_Binary, Search_Text, Search_Windows or Search_RowNo key is found in the data stream,the argument of the ADD_ATNEXTLINE key is inserted into the data stream right after the next found LineFeed.

Notes:

- Use this key only for processing ASCII files!
- This and the key Search_RowNow=Linenumber can't be used in the same row!

Ini File key: REPLACE_BINARY

This command key is performed, if one Search_Binary, Search_Text or Search_Windows argument is located in the same ini-File section and the keys value is found in the data stream. The argument of this command will replace the argument of the SEARCH_xxxx command.

If this command is found in a section beside the SEARCH_BINARY, all other binary commands are ignored.

Example:

```
[Section_SEARCH_AND_CHANGE]
;\x1B is the hex representation of ASCII 27 = Escape Sign
SEARCH_BINARY=\x1B&l5H
REPLACE_BINARY=\&n13WdTransparency
ReadOnlySearchKeys=ON
```

Ini File key: ReplaceNextTerm

This key is used to insert an additional search for a specific term, but only one time. This searched new term will be replaced by the key argument, or deleted if there was no argument provided. The syntax of the keys argument is

SEARCHTEXT:REPLACETEXT

As the column is used to divide the search and replace arguments it is not allowed to be used in the search text. After DSM has found the search text beginning right after the SEARCH_... searched item it will replaced the found item will destroy itself. Therefore the command is only searched/executed once.

For example, the key can be used to set parameters for instance colour at the end of a line. So the following example searches in a row

```
[Invoice csutomer order number german/englich print version]
; If one of the two items can be found in the data stream
SEARCH_BINARY="Bestellung           "
SEARCH_BINARY="Your Order           "
; switch to bold and red color
INSERT_BINARY \x1B(s3B\x1B*r3U\x1B*v1S
; at line end; return to black and white
ReplaceNextTerm=\x0D\x0A:\x0D\x0A\x1B*r1U\x1B*v1sT
ReadOnlySearchKeys=ON
```

Note, that included in the search text is a CR-LF and that both characters are also added in the argument. So the amount of printed lines remains the same.

Ini File key: INSERT_BINARY

The argument of this command is inserted right before the found Search_Binary, Search_Text or Search_Windows argument.

If a section is activated by a found SEARCH_BINARY key, and the Replace_Binary command is not used, the INSERT_BINARY has the second highest execution level, and the others are ignored.

Note: The length of both arguments of SEARCH_ and INSERT_BINARY may not exceed the size of 255 characters!

Example:

```
[Section_SEARCH_AND_INSERT]
;\x1B is the hex representation of ASCII 27 = Escape Sign
SEARCH_BINARY=Invoice.
INSERT_BINARY=\x1B(s20vs3b4148T
ReadOnlySearchKeys=ON
```

This example demonstrates how to print the text "Invoice." Very large with 20pt, bold and in a Universe font.

Ini File key: OutSearch_Binary, OutSearch_Windows, OutSearch_Text

Those keys performs exact the same like the SEARCH_xxxx keys, but

- The search is based on the outgoing data.
- The only 4 executed action keys are:
  - StoreNextWordToVariable
  - StoreNextWordToIntVariable
  - Add_Binary.
  - Counter
  - all Call keys

The keys are only used in conjunction with splitting the data stream into X archived print files, using an archive file name based upon a value found in the data stream, like a invoice number. Not implemented in this version of DSM if you would like to use this function please contact WML.

Example,

```
[HP LaserJet 4200 PCL5e]
; Archive and not print the data stream of this printer
OutArchiveDir=d:\archive\Invoices\#InvoiceNo#.prn
; Turn the splitting function on
OutArchiveSplitFiles=ON
; Do not print anything
NoPrinting=ON

[Split after the next formfeed]
; Search for a trigger expression on the last page of an
; invoice
Search_Binary=Total amount incl. vat
; Insert the split command
Add_Binary=\x1B)s-997Z
; You may enter the ADD command direct in the print stream.
; In this case this section is not needed

[Search last Invocienumber]
; The invoice number is the next word after the search text
OutSearch_Windows=Invoice No:
StoreNextWordToVariable=#InvocieNo#
```

Ini File key: StoreNextWordToVariable

Like all the other keys above, the StoreNextWordToVariable key is activated after a search key was successfully found. The argument of the key is a value name.

Without an optional length indication *DSM* will store the next following word after the searched term into the provided value name. This variable can be used later in any form.

If the searched section requests to delete the searched term using the Erase_Binary key, the found next word will also be erased.

It is recommended to start and end the variable name with the # sign, as otherwise you may loose a significant processing speed.

The optional length does indicate the requested minimum length in bytes of the stored item. Requested means, that if the data ends before with an escape sequence (Search_Binary mode) or any other character below Ascii 32, the parser stops reading into variable. In Search_Windows mode the positioning sequences are skipped.

If the length ends in the middle of a word, DSM will continue reading until to the end of that word.

Example:

```
[Global]
; init variable
Variable=OrderName:Nothing found

[Search for address Line 1]
; After this text in the printed document
Search_Windows=Ordered by:
; The name of the person who has ordered the parts,
; We need to store the name to a variable, which may read
; over all blanks for minium 20 characters
StoreNextWordToVariable=#OrderName#;20
; Should be only read on time
Search_Only_Once=ON
; Speed up process
ReadOnlySearchBytes=ON
```

Ini File key: StoreNextWordToIntVariable

Like all the keys mentioned in the previous descriptions, but only for the next word is read, the value is converted to a signed long integer. Length option is not allowed!
If the next word is 0123 the value will be 123. 12AB34 will be 12 and the result of A123 will be 0 (zero).

Ini File key: SkipUntilEnd

All remaining bytes of the data stream, including the search item will be ignoed by DSM. No additional search_ keys are performed on the remaining data stream, nor any possible Trigger items in that skipped part are searched.

Use this key for example to avoid additional pages from spoolers.

Primary Trigger key: TRIGGER_BINARY

This command is pretty much the same as SEARCH_BINARY, but

- The ADD_BINARY; INSERT_BINARY; REPLACE and ERASE_BINARY commands of the same section are ignored.

- Several TRIGGER_BINARY keys in one section are analysed as an AND operation. In other words, all triggers needed to be found in the data stream, before the further control commands of this section are performed.

- If all triggers of a section are found, additional analysis like Resolution page size and orientation and be checked before the further control commands of this section are performed. See below.

- The Server verifies and performs the found section only, if all arguments are found in the first bytes of the data stream.

- Do not use this key in the section [GLOBAL].

Example:

```
[Section_TRIGGER_SAMPLE]
TRIGGER_BINARY=Invoice
TRIGGER_BINARY=Delivery:
DSM_COMMAND=K3;C1F1000;C2F1100;D2
```

The *DSM* command is performed, if both trigger arguments are found in the data stream.

Primary Trigger key: TRIGGER_TEXT

This command performs the same functionality as the TRIGGER_BINARY key, but with 2 enhancements;

1. The comparison between the searched text and the data stream text is performed case insensitive.

   Example: TRIGGER_TEXT= Greta Garbo will be true for all following data stream contents: Greta Garbo; greta garbo; Greta GARBO etc.

2. In addition this key ignores any positioning Escape Sequences (Esc*p#X or Esc*p#Y), the windows driver might generate between each printed character. Here are some tips to make the trigger findable in the data stream, especially if you are not able to interpret a PCL data stream:

- Do not set the driver to print Truetype as graphics.
- The trigger word(s) should be formatted if possible with a printer internal font.

If *DSM* cannot find your trigger/search word at all, the reason can be that some new HP windows drivers or drivers supplied by Microsoft. Some do not use the same characters in their ASCII codes, hence it is not easy to scan them for a unique string.

However, those drivers do send down to the printer the name of the application and normally the name of the document in the initialisation within the data stream. In order to check this, print the document to file, and reload it into an editor. Right at the beginning of the stream you may find a statement like @PJL SET JOB="xxx". Check if you can use the string xxx for triggering the data stream.

If this doesn’t work DSM, please send the data stream by eMail to your local distributor, and let us know what key you are searching. We will be glad to let you know, how to do implement this function.

Primary Trigger key: TRIGGER_WINDOWS

This command performs the same as the function as the TRIGGER_TEXT key, except the comparison key value to print data value is performed character case sensitive.

Secondary Trigger key: Orientation

Secondary Trigger key: Page_Size

Secondary Trigger key: PaperTray

Secondary Trigger key: Printer_Names

Secondary Trigger key: Resolution

Secondary Trigger key: Simplex_Duplex

Secondary Trigger key: Trigger_Variable

Secondary Trigger key: Trigger_VariableSubString

Secondary Trigger key: User_Names

Secondary Trigger key: OutPrinting

The total key is ignored, if it was not possible to determine the appropriate setting in the data stream. The arguments of those keys are:

Trigger_Variable	Compare any called *DSM* variable to a given value: The value and the variable name is separated by ‘:’, ‘>’ or ‘<’ Notes: You may define the variable in the section GLOBAL, to have a predefined value. Operator ‘:’ does compare strings Operators ‘>’ and ‘<’ are compared as float numbers. Examples: Trigger_Variable=#MyValue#:test Trigger_Variable=#MyCounter#>7 See example in QA document
Trigger_VariableSubString	Like the Trigger_Variable Operator ‘:’. But instead of an equal comparison, the key becomes true, if the right argument can be found in the left string. The search is performed case in-sensitive.
Orientation	PORTRAIT, REVERSEPORTRAIT, LANDSCAPE, REVERSELANDSCAPE
Page_Size	A5, A4, A3, Letter, Legal, or PCL 5e value of the Esc&l#A command.
PaperTray	PCL 5e value of the Esc&l#H command. See “PCL 5 technical reference” or “PCL 5 printer comparism guide” for further details.
Trigger_OutPrinting	Used in conjunction with the database. Trigger a Section after the database inquiry and check if the job is still to be printed or not. This secundary trigger is used for example to react if a job is not printed anymore, but maybe only eMailed of faxed. So new copyfactors or archive rules can be applied. Not implemented in this version of DSM
Printer_Names*	;printer name 1;printer name 2…. Section is only performed, if job arrived through one of the named printers
Resolution	300, 600
Simplex_Duplex	0,1,2

User_Names*	;user name 1;user name 2;… Section is only performed, if job is intiated by one of the named users (logon name)

- User and Printer names are separated with semicolons. The command argument must also begin with a semicolon. The trigger is true, if one of the entries is found. The entries are case sensitive compared against the program call arguments

- Internally in the DSM code –P1Argument (printer_names) and the –P3Argument (user_names). If no command call arguments are provided, the trigger is always true.

Inifile Example:

```
[Printing color is allowed?]
; In every job is a character 1, to the first trigge is always true
Trigger_Binary=1
; Now second trigger, check is the job comed from a user, who is
; allowed to print color
user_names=USER1XP;USER2NT;…
; The color printing is not allowed (command only for HP printers)
PlanesInUse=1
```

Ini File key: ElseTriggerActivateSection

This key is performed if a triggered section does not become true. As the key argument any section name can be used. This section will be evaluated in the same way, as it would have become true using a search or trigger key.

```
[Printing color is allowed?]
Trigger_Binary=1
user_names=USER1XP;USER2NT;…
; Those ueser can print color
ElseTriggerActivateSection=Don’t allow colour printing

[Don`t allow color printing}
; all the other ones not
PlanesInUse=1
```

Ini File key: ActivateSection

In order to generate common functions in one rule/section *DSM* can call from any Searched or triggered section another section, where for example those common settings are performed:

Here an example that stores the delivery notes and invoices into the same archive.

```
[Invoices]
Trigger_Windows=Invoice No:
DSM_Command=K2;C1F500;C2F500F300;
ActivateSection=ArchiveJob

[DeliveryNotes]
Trigger_Windows=Delivery No:
DSM_Command=C1F500;
ActivateSection= ArchiveJob

[ArchiveJob]
OutArchiveDir=path for archive
```

Simplex and Duplex printing on pre-printed or pre-punched paper

If your printer does not have enough paper trays it is difficult to print simplex and duplex jobs from a tray that is loaded with pre-printed or pre-punched paper. The reason is the way the HP printers print duplex. In order to run the printer at full speed, the printer prints first the back page, and then front, back (next page), front, …

The advantage of this method is to run the printer at full speed, the disadvantage is in simplex mode, the front page is on the back page of the form. So the print is unusable unless your Software can handle this problem. Of course there is an easy way doing this with *DSM*. The trick here is to print the page always in duplex, even when a simplex job is launched. An example ini-file for doing this follows:

```
[GLOBAL]
; Function needs DSM to be activated
DSM=ON

[exchange by default]
; Because a 1 is in each data stream the section is always triggered
TRIGGER_BINARY=1
; Do section only if print job is printed out of paper tray 7
PAPERTRAY=7
; Do section only if print job is simplex printed
SIMPLEX_DUPLEX=0
;Set job to duplex and print every job page as a back page
; Also reset Tray 7, in order to erase possible Media Types
DSM_command=C1:1S1B1H7;C1B1;

[Printer Settings]
; Printer default menu settings, needed if datatsream does not
; defined some of those settings
ORIENTATION=PORTRAIT
RESOLUTION=600
PAPER_SIZE=A4
SIMPLEX_DUPLEX=0
PAPERTRAY=1
```

If the tray is selected via the PaperMedia Escape sequence, use search_binary and replace_binary to change the sequence to the dedicated paper tray sequence

DSM Variables

DSM is able to work with variables. Some variable names are predefined; they are listed in the next table.

DSM searches for the names of the variables in the data stream, as well as in all loaded forms/macros. A variable name starts and ends with the character '#'. The name itself is searched case sensitive, in Binary and Windows mode.

Predefined DSM *Variables*

Value-Name	Description
#ACC_EDITTEXT#	Special field generated by the accounting via Projects software. It contains standard editable text which can be placed on the print pages
#ACC_FILE#	Filename usually from ACC_FILE key which contains the file name for the accounting file.
#ACC_PROJECT#	Name of a project, which can be stored into an accounting file.
#ACC_STOREVALUES#	Variable which holds all the variables, whose content should be stored into the accounting database or file.
#ACC_TYPE#	Holds the value if the accounting information is written into a DBASE or CSV file.
#ACCTOTALPAGES#	This is the amount of print pages, provided through the –p5 parameter (In MS Windows automatically done). For PCL data streams the counter is set to 0 and DSM does count all FormFeeds itself. *
#PAGECOUNT#	Use this variable to generate an automatic page counter, the counter starts at 0, *
#PAGEPERCOPYCOUNT#	Same as #PAGECOUNT#, but the page counter counts only within a copy. So with the beginning of every copy the counter is reset to page number 1 With DSM_Command usage of K factor the result is 123… 123… etc. with Q it will be 111… 222… 333…
#COPYCOUNT#	Only usable in conjunction with DSM_Command K for copy generation. Used in a macro with a macro number of larger then 25000, an automatic copy counter can be defined, like Copy 1, Copy 2 etc..
#JOB_COUNTER#	If key enabled, then jobs can be counted.
#PCNAME#	The variable is set to the content of startup –P2 operand, or read with a higher priority from the PJL ATTRIB Commands out of the data stream.
#VAR_OUTFILESIZE#	For accounting, the variable holds the total written amount of bytes. The nearly double volume of data stream is not counted for a reverse printing rule.
#PRINTDOCNAME#	The variable is set to the content of startup –P4 operand
#PRINTERDRIVERNAME#	Also mainly for MS Windows to setup triggered rules which belong for all activated printers, using a special driver.

	(Not supported under pure NT)
#PRINTERNAME#	The variable is set to the content of startup – P1 operand
#PRINTERSHARENAME#	The printer share name is mainly for MS windows integration in order to resend the same job to the same queue using the expression: OutPort=#PCNAME#\#PRINTERSHARENAME# (Not supported under pure NT)
#USERNAME#	The variable is set to the content of startup – P3 operand, or read with a higher priority from the PJL ATTRIB Commands out of the data stream.
#VAR_ORIENTATION#	Stores the print orientation (portrait, landscape, etc) of the data stream, if defined in the data stream.**
#VAR_PAGESIZE#	Stores the page size (A4, Legal, etc) of the data stream, if defined in the data stream.**
#VAR_PAPERTRAY#	Stores the PCL paper tray number of the first page**
#VAR_RESOLUTION#	Stores the resolution (300, 600, etc) of the data stream, if defined in the PJL section of the data stream.**
#VAR_SIMPLEXDUPLEX#	Stores the simplex/duplex function of the data stream, if defined in the data stream.**
#VAR_TCPIP_PORT#	Available variable for MS windows implementation that holds the port address of the queue. (Not supported under pure NT)

* For PCL data streams, the variable is not available at the time the trigger and search events are processed. If the page count is needed, another counter has to be used.

** Variables are evaluated in the first 2KByte of the data stream. Changes within the data stream are not documented.

The variables #PCNAME# and #USERNAME# can also be defined in the data stream itself, using PJL command @PJL SET JOBATTR=”...”. Those sequences in the data stream are treated with a higher priority, then the -P command line operands.

The following table lists all available variables which were defined to cover customer special needs. Usually the variables are defined in the ini command file

In-File Entry	Description
VARIABLE=RESET_PAGECOUNT:0	Default: If an UEL is found in the PCL data stream, the page counter is reset to 1
VARIABLE=RESET_PAGECOUNT:1	If an UEL is found in the PCL data stream and the page counter is even and we are in duplex mode, the counter is incremented by 1. This version should only be used in duplex printing to move to the next odd page number.

The same result could be achieved using the key DSM_Cont_Page_Counting.

Some command line arguments are stored into variables, which can be used to in any form or in conjunction with transferring the data to other programs.

#IN_DSM_FILENAME#	Contains the incoming file name, including the full path for *W_DSM* command line argument.
#IN_DSM_FILENAMEWITHOUTPATH#	Contains only the incoming file name, without path. This variable is not set, if software runs in stdin mode.
#OUT_DSM_FILENAME#	Contains the outgoing file name, including the full path for *W_ DSM* command line argument.
#OUT_DSM_FILENAMEWITHOUTPATH#	Contains only the outgoing file name, without path. This variable is not set, if software runs in stdout mode.
#DSM_FORMS_PATH#	Contains the forms path from the *DSM* command line argument. Backslash or slash (Unix) is added!
#DSM_PROG_PATH#	Path to the DSM Software, In windows to the convert.exe file, maily c:\programme\wDSM
#LAST_ARCHIVE_FILE#	Path and file name of the last created archive file, independent if in or out data are stored.
#RANDOM#	Is available, if the variable #OUT_DSM_FILENAME# exists. The variable will contain a 8 digit long random number, which basically reflects the random filename name without the extension, generated by the print processor.

There are some more variables available, but only for network printers in conjunction with the monitored printing possibilities DSM offers: Not implemented in this version of DSM contact WML if you need to use this function.

#SNMP_TOTALPAGECOUNT#	See chapter Monitored Printing
#SNMP_TOTALDUPLEXPAGECOUNT#	See chapter Monitored Printing
#SNMP_TOTALCOLORPAGECOUNT#	See chapter Monitored Printing
#SNMP_JOBPAGECOUNT#	See chapter Monitored Printing
#SNMP_JOBDUPLEXPAGECOUNT#	See chapter Monitored Printing
#SNMP_JOBCOLORPAGECOUNT#	See chapter Monitored Printing
#SNMP_JOBSTATUS#	See chapter Monitored Printing

Example for automatic time stamp:

1. Purpose: You want to print on every page at a certain position the date stamp.

   Generate a blank form and add at the place of your choice the text #DATETEXT# and or #ACTTIME# formatted with a printer internal font, like Univers and print it to a file using a PCL 5e driver.

   If you use MS Word for forms generation, then please use the print filed in MS Word for setting the position of the stamp, do it like this:

   ```
   Print 14 27 "(10U" 27 "(s1p12vsb4148T#DATETEXT#..." 15
   ```

   Result:

   ```
   {PRINT  14 27 "(10U" 27 "(s1p12vsb4148T#DATETEXT#..." 15 \* MERGEFORMAT}
   ```

2. Add the macro using the DSM file management page. Say the Macro number is 3333

3. Add an *DSM* command for example into the Global Section: *DSM*_COMMAND=C1:1F3333; This command will call any time, unless there is no other command in a section or in the printfile the Macro 3333 on the very first printed page. As in the macro file there is only the time variable defined, the time will be printed.

4. If the position of the time changes from portrait to landscape and/or from A3 to A4, which is certainly the case, if you want to have the stamp in the lower area of the page. See example later in this manual.

5. Available the time related variables:

   #ACTTIME#,
   #DATENUM#,
   #DATETEXT#, #DATETEXT_D#, #DATETEXT_PL#,
   #DATESERIAL#,
   #MILLISECONDS#,
   #MONTH#, #MONTH_D#, #MONTH_PL#,
   #WEEKDAY#, #WEEKDAY_D#, #WEEKDAY_PL#,
   #TODAYNUM#

Further usable **DSM** *Variables*

The following table lists all variables that can be defined, by some of the ini-file keys or as variables. Not implemented in this version of DSM contact WML if you need to use this function.

In-File Entry	Description
#EMAILSUBJECT#	Holds the subject text, when sending out eMails

User defined Variables in Ini-File Sections

Use the ini-file to define any variables. This function is for example a perfect way, to differentiate between several companies. If each company prints fixed to the same queue, the printer sections can define the variables for each company, which are used in the forms.

The syntax in the ini-file is:

VARIABLE=VarName:VarValue

Notes:
Do not encapsulate the variable name with '#', but note that *DSM* does surround the variable name automatically with the sharp character. The searchable variable name of the example above will be: #VarName#.

The character ':' is the differentiator between the variable name and the value. An empty value is defined, if the line ends with the colon.

All read variables are not erased, when a new ini-file is selected (Search and Triggers are erased).

You may encapsulate the variable value in " " in order to use blanks at the beginning or at the end of the value. Also the Hex notation \x## can be used.

Example:
VARIABLE=COMPANY: My Company Ltd.
Form now on the text #COMPANY# is searched in the data stream and is replaced with the "My Company Ltd." company name.

Variable manipiulation functions

Currently *DSM* supports flowing functions for string manipulations. The Syntax of the case sensitive named functions are:

LEFT(#StringVarible#, Length)
LEFT(#StringVarible#,#IntVariable#)
	The functions are only interpreted, if they are set directly in front of a variable and there is a comma directly after the variable followed by the length number and the closing parenthesis. The positive length represents the amount of bytes to be taken out of the provided variable. The amount can be represented by another variable.

RIGHT(#StringVarible#, Length)
RIGHT(#StringVarible#, #IntVariable#)
	In principal the same function, but the amount of bytes are taken from the right side of the evaluated string

LOWER(#StringVariable#)	Turns all ASCII characters from the variable into lower case.
UPPER(#StringVariable#)	Turns all ASCII characters from the variable into upper case.

STRPOS(#Variable#, #SearchVar#)
STRPOS(#Variable#, Text)
STRPOS(#Variable#, “ “)
	Returns the first position minus 1 of the first charater of the #SearchVar# string found in the #Variable# string. If the string could not be found then function returns zero.
	Leading blanks can only be searched if the search text is surrounded be double quotes.
	Examples: The content of the #Variable# is “Hello World”
	STRPOS(#Variable#,” “) -> 5
	STRPOS(#Variable#,”World“) -> 6
	Assume content #SVar# is “World”
	STRPOS(#Variable#, #SVar#) -> 6
	Assume content #SVar# is “ ” (a blank)
	STRPOS(#Variable#, #SVar#) -> 0
	Assume content #SVar# is “ World”
	STRPOS(#Variable#,#SVar#) -> 6

Counter key	Increments an integer variable by default 1 or any other value

Note: Currently DSM dos not support combining several of these function like UPPER(LEFT(#Variable#, 5)). In this case use two command lines:

```
Variable=MyTempVar:LEFT(#Variable#, 5)
Variable=MyTempVar:UPPER(#MyTempVar#)
```

Examples:

Get the actual year, most likely set in the section Global

```
Variable=ActualYear:RIGHT(#DATENUM#, 4)
#DATENUM#  -> 5.2.2004
#ActualYear#    -> 2004
```

Limit the Document name variable to 14 characters

```
Variable=PRINTDOCNAME:LEFT(#PRINTDOCNAME#, 14)
Before: Microsoft Word – TheDocumentname.doc
After:   Microsoft Word
```

Error:Will not work

```
[Named Section as the user logon name]
Variable=FullUserName:UPPER(Mark Brown)
```

Results into the variable content UPPER(Mark Brown). The string Mark Brown is not converted, as it is none variable!

This will do the job:

```
Variable=FullUserName:Mark Brown
Variable=FullUserName:UPPER(#FullUserName#)
```

It can also be used in any file definition, search and replace keys:

```
[Warranty Text printing]
; Find the place to print the info in the data stream
Search_Binary=!WarrentyStatement!
; Replace with text and variable #NextYear#
Replace_binary=This part has warranty up to end of #NextYear#
; Which was calculated from the date code dd.mm.yyyy
variable=ThisYear:RIGHT(#DATENUM#, 4)
; And incremented by one
Counter=#ThisYear#
```

Access Environment Variables

The *DSM* key GetEnvVariable reads any available environment viable into a *DSM* variable. The variable name is set to uppercase and surrounded by the # characters.

GetEnvVariable=os will give a D*SM* Variable #OS# with the content Windows_NT. Use key variables store for a listing of all collected DSM variables.

Type SET in a DOS command window to view the environment settings.

Merging Variables into Forms or Documents

DSM does provide a low level method to insert user-defined variables into a document. This method does work for any kind of data streams, for example Postscript!

The idea behind this is for example storing the prices in an external variable file the format is described in this section, printing this, as an example Pricelist file created in PageMaker, through *DSM*, in order to merge the prices from the external file.

In order to perform this, you may want to generate a new printer queue and link this merging process direct to this dedicated queue, using the key VARIABLE_FILE key. Example:

[PriceListPrinter]
VARIABLE_FILE= pricelist.???
The syntax of the text file is:
SearchText=ReplaceText

The function is very powerful, because you can search for any binary text in the data stream or used forms, and replace them with up to 246 characters, even containing printer commands.

Tips:

Wrap the search variables into special start and stop keys, for example like the standard variable definitions, like #no:1234#. The possible file entry line then could look like this: #no:1234#=90,50 Euro.

If leading blanks are used for right alignment, use double quotes for the value part:
#no:1234#=" 90,50 Euro".

You may format the the values using any predefined font attributes, but also the formatting can be done in the value part, like this:
 #no:1234#=\x0E\x1B)s1p12vs3b4099T   90,50 €\x0F

The filename definition of the Variable File key may contain any previous defines variables. So the file name can be for example user or printer specific.

Example of using #USERNAME#:

The goal is, that on the first page, the user name, who has generated the print out, will be printed somewhere on the page, maybe lower left corner, rotated by 90 degrees.

The first way to implement this request, is simply to add at the print position in the data stream the variable key #USERNAME#.

The second way is, to put the key into a form, which is called on the first page, maybe using *DSM* commands.

Automatic Page Counter

Following steps generate an automatic page counter, which can be placed at any position on the page. If you want to use the given example, then use the provided form 25000 and continue with position 4.

1. Generate a new form with a form number between 25000 and 30000. Remember, forms in this number range are downloaded on every page.

2. Insert on that form the text: #PAGECOUNT#.
   Remember: Format it with a proportional printer font, or use the PRINT field in MS Word.

3. Print the form using a PCL 5 driver to file and convert it to a macro, using File management Macro option.

4. Generate or enhance an *DSM* command, and include the form on every page. The minimum command would be <<C1F25000;D1>> in the data stream, or use Overlay page of the web interface, and alter the ini-file, like for example in the global section: *DSM*_COMMAND= C1F25000;D1.

As the form ID is between 25000 and 30000, the form/macro is loaded on every print page. As in this procedure the variables are inserted on file load time, the actual page number is correctly inserted.

It is possible to reset the Page count to 1:

A) By splitting the files
B) The R - DSM_COMMAND
C) The new Escape Sequence <esc>)s-995Z

Automatic Job Counter

The following steps generate an automatic job counter, which can be placed at any position on the page, or simply be used to count jobs.

One job counter can be placed in every activated rule. So in theory DSM can handle nearly indefinite rules with the same amount of counters. But be careful, as always the same variable is used, the variable content is the counter of the last read counter file.

The variable key JOB_COUNTER points to any regular text file. The file holds in the first line the start up value of the counter, which is incremented and stored for the job into the DSM variable #JOB_COUNTER#.

The variable itself can be used as explained in previous examples in a form or using search and Add or insert.

To install an overall page counter simply add the variable key JOB_COUNTER into the section global, leave the default argument as it is. As long as the variable #JOB_COUNTER# is not used in a form, open the file after some jobs and look at the value.

Another example is to establish a counter how many documents / jobs are printed, containing the word confidential:

```
[Search for Confidential]
Search_Windows=Confidential
Job_Counter= workpath confidential.var      note workpath is the path where the file is stored in the
DSM unit.
; The text file confidential.var must exist, please
; create it for example using notepad.
```

Tipp: The variable can of course be exported into any report / accounting file or can be passed to any external software by command line parameter using one of the CALL keys.

Additional *DSM* functions

Automatic symbol set conversion

DSM is able to convert any incoming characters to ASCII or any other symbol set using the key INSYMBOLSET . As incoming data is converted the key should be used in the sections Global, PrinterName or UserName.

The conversion files are again standard ASCII files and can be changed using an ASCII editor. *DSM* comes with an example file ebcdic.sym, which defines the conversion from EBCDIC to ASCII.

The syntax of the symbol set definition file is pretty easy:
ASCII-Character-Number Min-One-Blank Source-Character-Number

In order to change or generate a new symbolset file, copy the ebcdic.sym file and modify the new file.

HexEmulation

Once set in the GLOBAL or any positive search or triggered section, the output of the *DSM* Server is converted in a readable hex data stream. Use any time the key HEXOUTPUT=ON.

Init file: job_init.mac

If a file named job_init.mac is found in the WorkPath directory, the *DSM* Server will download the content as a Job initialisation. Please note that the data of job_init.mac is not sent through the converter itself.

As always, the file name needs to be lowercase for Unix based operating systems.

This file can also be used for example to download soft fonts to the printer.

If the required initialisation is less then 512 bytes you may also use the DSM_INIT_JOB key.

Exit data file: job_exit.mac

If a file named job_exit.mac is found in the WorkPath directory, the *DSM* Server will download the content as a Job exit initialisation. Please note that the data of job_exit.mac is not sent through the converter itself.

As always, the file name needs to be lowercase for Unix based operating systems.

If the needed initialisation is less then 512 bytes you may also use the DSM_EXIT_PCL key.

*Demo page: Esc**3M*

This additional Escape Sequence generates a demo page, with some barcode samples.

KillJobsWithLessBytes

Some software applications for unknown reasons send several print jobs to the printer, for example the first job contains only one character, e.g. a FormFeed.

For such cases the ini-File key KillJobsWithLessByte=ByteNumber is able to skip a total print job. Skipping means the job is dumped, no searched and no triggers are performed. As the command argument enter any amount of bytes.
This command can only be used in the following startup sections: GLOBAL, PRINTER- and/or USER-NAME.

EraseMacros key

Usually DSM erases all loaded macros from the printer’s memory at the end of the print job. This action is correct, so by default this function is activated even if no key is provided in the ini-file.

As this behaviour may erase also pre-loaded forms, which should still be available after a DSM job is finished, use this key in Mode OFF to turn this function off.

*Print Quality key**

If the print job is PCL 5 and starts with an UEL sequence (Esc%-12345X) this key advises the printer to print the data stream using one of this these quality settings:
1 – Draft
2 – Normal
3 – High

*EconoMode key**

If the print job is PCL 5 and starts with an UEL sequence (Esc%-12345X) this key can advise many types of printers to print in a toner save mode:
1 – ON
2 – OFF

*RenderMode key**

If the print job is PCL 5c and starts with an UEL sequence (Esc%-12345X) this key advises the printer to print the data stream either in color or in grey scale:
1 – Color
2 – Gray scale

- These commands should be used together with the DSM_COMMAND=ET;

PreScribe™ emulation

The emulation was developed mainly to converting PreScribe™ barcode printing to PCL. *DSM* currently supports the following PreScribe commands: .

!R!	Start Prescribe	AMCR	Automatic mcro overlay
ALTF	Set font ID	BARC	Barcoding
BLK	Filled box	BOX	Draw a box
CALL	Call Macro (no parameters!)	CASS	Paper tray selection
COPY	Page copy	CMNT	Remark
CSET	Symbol set	CTXT	Print center positioned text
DAF	Make fonts permanent	DAM	Line to absolute print position
DAP	Delete all macros	DELF	Delete font
DELM	Delete macro	DRP	Line to relative Position
DUPX	Duplex printing	DXPG	Page side selection
DZP	Line to absolute position	ENDM	End macro recording
ENDR	End Raster Graphics	EXIT	End prescribe mode
FONT	Internal Bitmapped fonts	FPAT	Define a pattern
FSET	PCL font selection	JOG	Job offset
LDFC	Define font character	MAP	Move cursor absolute
MCRO	Define macro	MRP	Move Cursor relative
MZP	Move cursor absolute	PAGE	FormFeed
PAT	Select pattern	PDIR	Rotate
PSRC	Paper tray selection	RESL	Resolution
RES	Reset	RGST	Move coordinate system
RPF	Pop last font	RPP	Pop Cursor
RTXT	Print right positioned text	RVRD	Start Raster Graphics
SCF	Pust actual font	SCP	Push Cursor
SCPI	Character per Inch (fixed only)	SEM	Seemulation mode (Ignored)
SETF	Set new font ID	SFA	Set alternate font sequence
SFNT	Set font	SLM	Set left mergin
SLPI	Lines per Inch	SLPP	Lines per Page (Ignored)
SLS	Lines per Inch	SPD	Set Line Dimension
SPL	See lines per inch *1	SPO	Print orientation
SPSZ	Paper size	SRM	Set right margin (Ignored)
STAK	Set outbin stacker	STM	Set top margin
STR	Resolution	TEXT	Print left positioned text
UNIT	Meassure system		

*1 By default ignored, can be turend on with ini-file key PRE_PERFORM_PAGE_COMMANDS

Notes to the commands:
The RES command does currently initialize the printer to the following settings:
Default Tray with media type Plain, 1 copy, portrait, 6 lines per inch with Courier in Roman 8 symbol set and 75 dpi.

The CASS and PSRC commands DON'T select directly the praper trays. When PreScribe selects a tray *DSM* opens the actual ini-file and checks the section [PRESCRIBE] for the keys: PreScribe_CASS1 to 4 or PreScribe_PSRC1 to 6 and 99. If the key is not defined, no selection change is performed. But if it is defined, the key values can be any PCL sequence. Finally the corresponding media type of any available paper tray can be set in the printer's paper handling menu.

So the paper tray selection flow is like this:
CASS / PSRC -> converted to a media type -> send to printer PCL interpreter -> selects finally the correct tray form the printer defined media list.

In order to activate this emulation, add in the global section or in the appropriate printer, user or printer driver section the key: PRESCRIBE and turn it on.

Depending on the printer, HP can set wide printing on and off. We suggest you always use wide printing ON. Especially for landscape printing, it could be that a YMove value needs to be defined.

The following keys, which can be used exclusively in the [PRESCRIBE] section:

Pre_Init — The provided string will be used for initialisation of the printer when Prescribe emulation is activated.
Pre_Init_At_First_!R! — This string will be used for additional initialisation at the place where the first !R! command is found in the data stream.
Pre_Perform_Page_Commands — If turned on, the SPL command is performed
Pre_XScale — Horizontal scaling factor, 100 = 100%, no scaling;
Pre_YScale — Vertical scaling factor, 100 = 100%, no scaling;
Pre_XMove — The definition for the horizontal origin of the coordinate system.Value can be negative;
Pre_YMove — Same for vertical position
Pre_CASS0 to 4, see above
Pre_PSRC0 to 6 an d99, see above

Notes:
This emulation is not perfect for all situations, some modifications may have to be made to make the DSM operate correctly or to add missing commands. Please inform WML of any Promblem you may come across.

Not interpreted PreScribe commands are ignored.

The readable text size may differ to the original also some EAN and UPC barcodes with readable text could look different, according to the barcodes specification, DSM is more accurate.
Paper tray selection can also be done by any DSM_COMMAND.

Epson™ ESC/P (9 Needle)

The emulation was developed for a customer, and does convert about 85 percent of all Epson commands, the remaining other ones are ignored.

In order to activate this emulation, add in the global section section or in the appropriate printer, user or printer driver section the key: EPSONEMULATION and turn it on.

There are 3 more keys, which can be used exclusively in the [EPSON] section:

Epson_Init — This string will be used for initialisation of the printer when EPSON enabled is detected.
Epson_XScale — Horizontal scaling factor, 100 = 100%, no scaling;
Epson_YScale — Verticall scaling factor, 100 = 100%, no scaling;

Note: This emulation not perfect. If it doesn’t match your requirements, please send your data stream to WML or your local distributor for evaluation.

IBM™ ProPrinter Data Stream PPDS Emualtion

The emulation was developed for several customers, and does convert about 70 percent of all PPDS commands; the remaining ones lead to an abort of the emulation.

In order to activate this emulation, add in the global section or in the appropriate printer or user section the key: PPDS_ENABLE and turn it on.

The *DSM* iniFile you may use the following additional Section with following entries:

```
[PPDS]
; initialisation of PCL converted data stream
PPDS_INIT=\x1BE\x1B&l1O
; Set scaling factors
PPDS_XSCALE=95
PPDS_YSCALE=100
```

Note: The name of the section is fixed. If the PPDS_ keys are listed in any other section, they will be ignored!

Note: This emulation is not perfect. If it doesn’t match your requirements, please send your data stream to WML for evaluation.

Paper tray handling commands “1B 19 #” where # is the character 0 (hex 30) to maximum 9 (hex 39).

The paper source can be set by the key PPDS_PaperTray# searched in the section PPDS in the actual ini-file. Any valid escape sequences are allowed, so the direct paper tray sequence Esc&l#H can be used, as well as the media type sequences.

Printronix IPG™/VGL and PGL emulation

This emulation was also developed for a customer. The following commands are supported: Comma, Minus, Star, Plus, A, B (Code 3of9), F IPEXP, IFONT,S, J, K, LB, LS, M, O, P R, SM, T, X

The following commands are ignored: , IF1XX0, IC, IOP, IMH, IMV

All other commands will generate an error and exit the emulation.

In order to activate the emulations use the ini-file keys:

Printronixemulation_vgl	Activate VGL Emualtion
Printronixemulation_pgl	Activate PGL Emualtion

In order to activate this emulation, add in the global section or in the appropriate printer section the key PrintronixEmulation and turn it on.

All the following keys can ONLY be used in the section: PRINTRONIX, which needs to be inserted in to the ini.file manually.

PR_CC_TextChar	Sets the default Printronix command character, usually Tilde
PR_HorScale	Definition of the horizontal scaling factor as an integer value. Enter 100 (= 100%) for no scaling
PR_Init	The predefined string will be used for initialisation of the printer when a VGL data stream is detected. You can set here all page definitions, PCL coordinates, etc.
PR_PageInit	Same like PR_init, but the PCL commands will be sended as an initialisation on every page.
PR_ScaleMode	Defines how the positioning commands are processed 1 (default) the values are defined in dots 0 the values are used in CD.DP positioning
PR_VerScale	Definition of the vertical scaling factor as an integer value. Enter 100 (= 100%) for no scaling
SM#####	For every used font id #####, a PCL font can be assigned. Example: SM00005=\x1B(s1p60v1s3b4168T
VGL_STD_FontStroke	Definition for standard Font stroke in VGL emulation: 0 = Normal 3 = Bold.
VGL_STD_FontType	Definition for standard Font in VGL emulation only: 4099 = Courier 4102 = Letter Gothic.

The *DSM* iniFile you may use the following additional Section with following entries:

```
[PRINTRONIX]
; initialisation of PCL converted data stream
PR_INIT=\x1BE\x1B&l1O
; Set scaling factors
PR_HORSCALE=95
PR_VERSCALE=100
; Set SM font definitions
SM00091=\x1B(s1p8.5v4sb4148T
SM…
```

Note: The name of the section is fixed. If the PR_ keys are listed in any other section, they will be ignored!

If any SM commands are found in the data stream, the font selection can be done in the ini file.

Note: This emulation is not perfect. If it doesn't match your requirements, please send your data stream to WML.

Softfont loading on demand:

Font: Automatic OCR A1 Loading

Font: Automatic OCR B Loading

DSM traces the data stream for the original HP Escape Sequence for OCR A1 (check printing) and OCR B (Swiss, Austrian address printing). Once the correct sequence is found, the font is automatically loaded to the printer and activated.

This function may save a lot of network bandwidth, because there is no need to download the fonts together with every data stream. It is only loaded, when needed.

Due to licencing reasons, the fonts are NOT provided in the standard *DSM* Print appliance.

General methods for loading Softfonts

There are several ways using *DSM* to automate the softfont download into the printer's memory, before they are used in the data stream.

The PCL 5 Technical Reference manual from Hewlett-Packard explains how Softfonts are built. In order to work properly with *DSM*, a downloadable softfont needs to have a proper ID sequence in front, ←*c###D and at the end of the font file, the font resident sequence ←*c5F, which makes the loaded font resident against any software reset.

After the font is loaded, it can be addressed either by the ID or using the full escape sequence.

Here are some examples how to download softfonts.

1. Use the predownload file **pjl_init.mac**. This file needs to have all the required fonts stored in it, for all possible print jobs. So the disadvantage is that this file is maybe quite large and as it is sent together with every print job. It may cause a lot of additional and unneeded network traffic. See beginning of this chapter for further information.

2. Name your downloadable font file like a macro: AnyNumber.mac and use the ***DSM* command EEAnyNummer;**. The benefit of this method is, that you can download the fonts only on request, they will be loaded only once within a datastream to the printer.
   The request itself can be triggered with one of the following and all previously described trigger methods, where ## stands for the filename ###.mac which is stored in the Overlay directory

2.1 The most elegant way is to use the BINARY_SEARCH key, search for the first font selection escape string. Then use the key INSERT_BINARY add the macro call ←&f##Y.

```
[OCR-B download]
search_binary=\x1B(1O\x1B(sp10h12v0s0b110T
insert_binary=\x1B&f9999Y
ReadOnlySearchKeys=ON
```

2.2 Search in the data stream, using TRIGGER_BINARY for any appropriate trigger word. If found add to the section the command: *DSM*_Command=EB##; (Sending a form at the beginning of the job).

The benefit here is, no data stream changes are needed.

```
[Trigger Cyrillic]
; Code Page 866
  Trigger_binary=\x1B3R
  ; Search for a trigger word.
Search_binary=<<Cyrillic>>
  Erase_Binar=1
  ; The trigger command
  DSM_Command=EB9999;
```

In this example two trigger methods are used within one section. The search looks for the provided text word, erases it and sets the command. Alternatively if the Escape Sequence for Russian PC866 code page is found, the command is set as well.

2.3 If there are no Triggers in the data stream at all put the same DSM_Command into the [GLOBAL] section. If there is no printer initialisation in the data stream, add the proper font selection sequence at the end of the file, l←(##X. The number ## is the same as is used in ←*c##D command at the beginning of the file.

2.4 If the file does have some initialisation sequences, which reset the font selection, then use the DSM_Command=C1:1G9999; and insert the font right after the PCl initialisation.

2.4 Use the DSM command <<EB999;>> directly in the print data stream.

PCL3GUI HP Business InkJet support

DSM partly supports the HP Business InkJet Printers. The *DSM* commands that can be used are:

EE#; EB#; C#[,#]H#; D#; pjl_init.mac

Warning: *DSM* does not verify any inserted data stream (macros) or unsupported commands within a PCL3GUI data stream.

Rescaling the print page

This functionality enables *DSM* to resize the page independently to any factor in X and Y directions. For example a page designed for 80 characters per line at 10 cpi can be resized horizontally to fill printing of 80 characters at 10.1 cpi.

In order to do this, the key PCL_X_SCALING and/or PCL_Y_SCALING needs to be added to the ini-file section. The scaling factor is provided in percentage. Therefore the entry needs to be larger than 0. An entry of 100 means no scaling at all. An entry of 50 will reduce the page to 50% of its original size, still printed from the upper left corner. Refer to *DSM* commands for repositioning the data on the page.

Notes:
DSM does **not** rescale any graphical data.
DSM does **not** rescale any HP GL/2 data. As PCL picture frames are rescaled, the Graphical information may be scaled correctly.

Barcode printing and readability considerations

The creation of printable barcodes, including the print quality, clarity of readability is extremely important. The Tempest DSM generates barcodes from the escape sequences sent to the device. A barcode is constructed from a combination of black (Stripes) and white bars (Spaces). The separated distances (width) between them determine determines the barcode type and data format.

When printing your Barcode output the following important factors must be considered to ensure easy and reliable readability.

1. The Output print resolution and density should be sent at the highest quality.
2. The Toner Quality should be good to ensure a solid, clear and strong output.
3. The Print Media selection is important to prevent excessive toner bleed and contrasts of output and subsequent print and readability issues.
4. The Printer that you choose to print your barcode output should preferable be a laser technology based solutions. Inkjet printers can be used but it is recommended that the final print output is scanned for readability before releasing the labels or documents produced.

The Tempest DSM supports scalable barcode printing applications, It is important the barcodes produced are not scaled to small due to definition and potential readability issues. We recommend that the barcode be produced in “normal” output size and tested.

In addition to this it is important that magnification and expansion of your barcode output be clear and non pixelated in output. The Tempest DSM supports fixed sized fonts and scaling them to large will caused print quality problems.

You should test the output, and adjust settings, with all the components, to produce output that will work with your requirements.

It is advisable to test the output with the Scanning equipment intended to be used on the project.

William Matthew Ltd. does not warrant that all the Barcodes and fonts contained with the Tempest DSM product are readable by all devices.

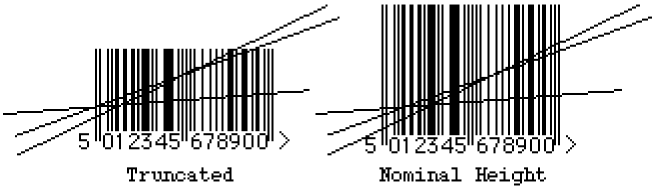
Bar thickness can be adjusted to overcome specific issues by the Tempest DSM.

Getting a Good Scan

The data contained within the barcode is read by moving a spot of light across the barcode, starting in the white space before the first bar and moving through all the bars and spaces to the white area at the other end of the barcode.

Therefore, when printing a bar code, the method of scanning must be considered when choosing the bar code dimensions. In applications where the barcode is scanned using a wand a barcode which has been shortened or truncated will still read as long as the wand can be easily passed over the full width of the barcode without going of the edge.

Most retailer checkouts use moving beam scanners, where a beam of light moves over the barcode. There is no way of controlling the items path over the beam sufficiently to ensure that the beam will move horizontally across the barcode. The wider you print the barcode the greater the chances of the scanner beam crossing the top and bottom of the barcode. It soon becomes obvious that by increasing the barcode height, that the chances of a good scan are increased (See example below).



Scanning and the issues of printable contrast.

Another area to consider is the contrast of the printed output. It is important that the contrast between the bars and spaces are totally different. All the bars should reflect no light and all the spaces should reflect 100%. In practice this never happens, but barcodes will still read providing the lightness of the bars and the darkness of the space do not exceed limits.

Barcode Information

In This section we have included information on the Formats for a few of the most commonly used Barcode formats namely UPC and EAN 8 and 13 formats. This information should be used a background reference in assisting with production of the data format.

UPC Barcode Structure

The Universal Product Code symbology is used throughout the U.S. and Canadian grocery industry and has expanded into general retail as well.

In its standard version (UPC-A), the bar code consists of a five digit manufacturer number and a five digit product number. In addition there is a 1 digit number system identifier at the start of the code. The number system digit denotes the use of one of ten number systems defined by UPC:

0, 6 and 7 are for regular UPC codes.

2 is for random weight items, e.g. meat, marked in-store.

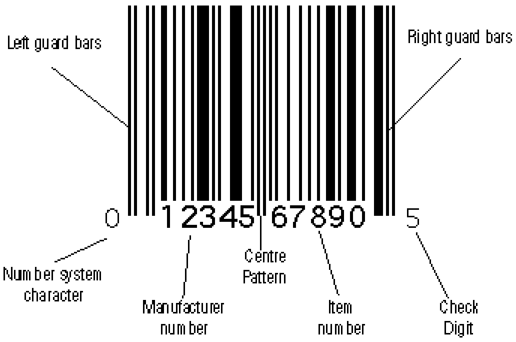
3 is for National Drug Code and National Health Related Items.

4 is for in-store marking of non-food items.

5 is for coupon use.

1,8 and 9 are not used at present.

The UPC symbol also has a check digit which is the last digit of the code and is calculated according to the algorithm used for EAN.



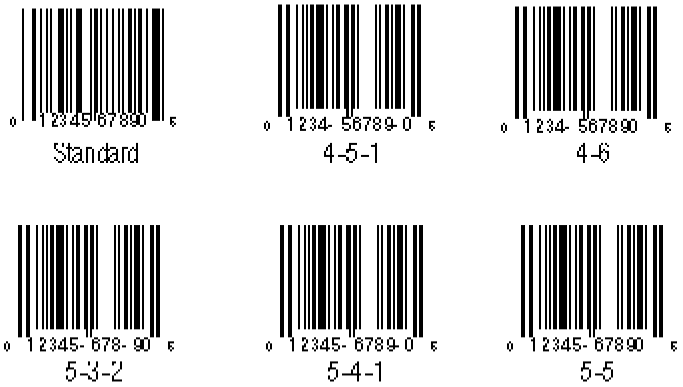
As with EAN, UPC is described in terms of magnification factor as a percentage.

The 100% code is the same size as a 100% EAN.

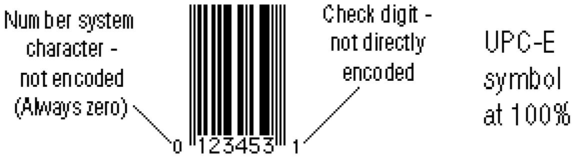
The human-readable characters for the number system character and the check digit are printed at a smaller size and separately from the other digits.

There are a number of human readable character layouts in use currently. The HRC layout does not affect the encodation of the bar code, but does assist in visual identification and the grouping of human readable characters can aid in key entry if scanning is not used.

The following are common layouts:



As well as the standard UPC-A, there is another UPC symbol in common use - the UPC-E or zero-suppressed symbol.


Number system character - not encoded (Always zero)
0
123453
1
Check digit - not directly encoded
UPC-E symbol at 100%

The UPC-E is used on packages which worepresents a full UPC code and the full coduld otherwise be too small for a bar code symbol. It actually e can be re-constructed from the shortened form. Only UPC codes using number set 0 and with sufficient zeros can be shortened to UPC-E codes. The form and number of zeros in the manufacturer numbers dictates how many product numbers are available.

The rules are as follows:

1. If a manufacturer number ends in 000, 100 or 200, then three digits are allowed for the product code and only the first three digits of the manufacturer number are used. In such a case only the product numbers 00000-00999 could be shortened to UPC-E. The six digits are constructed from the first two digits of the manufacturers number followed by the last three digits of the item number, followed by the third digit of the manufacturers number.

e.g. 12100-00745 becomes 127451

2. If a manufacturer number ends in 300, 400, 500, 600, 700, 800 or 900 then two digits are allowed for the product code. The first three digits of the manufacturer number are used and a "3" is inserted after the product code.

In such a case only the product numbers 00000-00099 could be shortened to

UPC-E.

e.g. 12500-00081 becomes 125813

3. If a manufacturer number ends in 10, 20, 30, 40, 50, 60, 70, 80 or 90 then one digit is allowed for the product code. The first four digits of the manufacturer number are used and a "4" is inserted after the product code.

In such a case only the product numbers 00000-0009 could be shortened to

UPC-E.

e.g. 12390-00007 becomes 123974

4. If the manufacturer number does not end in zero all five digits of the manufacturer number are used followed by one digit of product code. Only numbers 5 - 9 may be used. Therefore the manufacturer only has 5 UPC-E codes available.

e.g. 12345-00008 becomes 123458

The check digit for the UPC-E is calculated from the full length UPC code and is not directly encoded by a bar/space pattern. Instead it is derived from the parity of the rest of the bar code.

Each of the numbers 0 - 9 can be represented by a bar/space pattern with a even number of dark bars or one with an odd number of dark bars according to an encodation table.

The pattern of even parity and odd parity patterns tells the scanner which check digit has been used according to the following table: (character positions 1-6 are numbered across the top)

Check Character	1	2	3	4	5	6
0	E	E	E	O	O	O
1	E	E	O	E	O	O

2	E	E	O	O	E	O
3	E	E	O	O	O	E
4	E	O	E	E	O	O
5	E	O	O	E	E	O
6	E	O	O	O	E	E
7	E	O	E	O	E	O
8	E	O	E	O	O	E
9	E	O	O	E	O	E

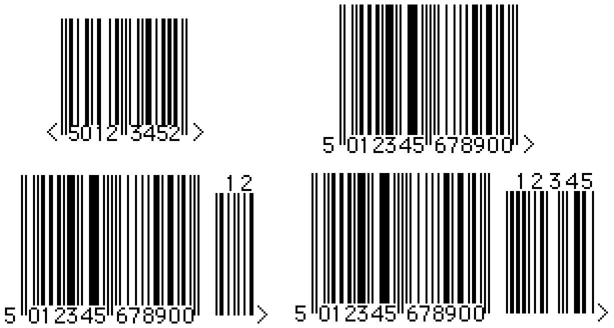
The UPC symbology allows for supplementary codes to be added to the main code. These add-ons may be 2 or 5 digits. They do not include a check character and have no bearing on the check calculation of the main code

The EAN symbology

The barcode symbology most commonly seen in Europe (and most of the rest of the world) is EAN. It is a numeric only bar code system used for identification of retail products. Unique EAN numbers are allocated to each separate retail product, not just by product brand but by variation (weight, colour, flavour, etc..). Also separate numbers are required when the product changes (except when the price changes).

The EAN symbol has two basic formats, the 8 and 13 digit variants.

The 13 digit code is more common. The 8 digit code is generally used where space is restricted.



There are two other variants of EAN which have supplementary "add-on" codes: 13+2 and 13+5:

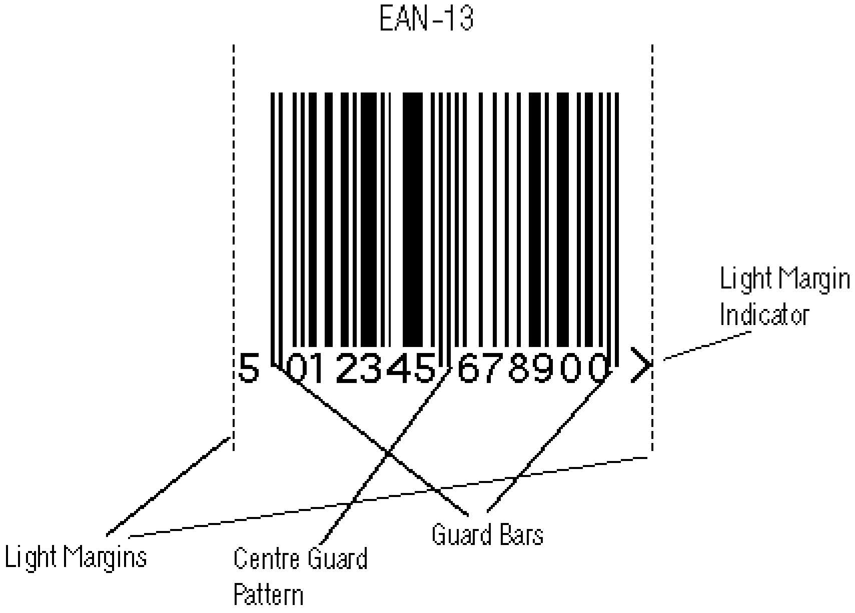
Although not often used in practice, the add-on codes are available for encoding supplementary information.

The EAN symbology is intended as a world wide standard (although some countries use other systems), therefore, no two retail products may have the same EAN number. To ease the administration of number allocation, Each country using EAN has a country identifier at the start of the barcode. For the U.K. the digits 50 identify U.K. manufacture. Other countries have 2 or 3 digit prefixes, (93 for Australia, 773 for Uruguay, etc..)

The rest of the EAN13 code is divided into the Manufacturer Number, the Item Reference Number and the check digit. In the U.K. the Manufacturer Number is usually 5 digits, the Item Reference Number 5 digits and the check digit is the last number.

Article Numbering Associations in each country assign Manufacturer Numbers. The Item Reference Number is decided by the manufacturers, who are free allocate the available digits as they wish.

Although the UPC system preceded the use of EAN codes, EAN was made retrospectively compatible. In theory all scanners designed to read UPC should also read EAN, and vice-versa. In practice, all EAN scanners will read UPC but some scanners sold in the USA will not read EAN, . Such scanners are no longer being sold and it is hoped that in time the two systems will become totally compatible.



Like UPC, the EAN symbol is described by magnification, the allowable limits being 80% to 200%. For each magnification there is a recommended (or nominal) height. This figure is recommended to ensure symbol readability when read by a multi-directional scanner, therefore any reduction in height (truncation) should only be attempted if absolutely necessary.

In common with most other bar code implementations EAN8 and EAN13 have a check digit which is the last number on the right. It is used to check for an error in scanning or data entry. The most common error found with the transcribing or keying of data is that of transposition (reversing the order of two digits). Therefore, the following system is used:

1. Starting from the right of the number and excluding the check digit, add each alternate digit.

2. Multiply the result of 1. by 3

3. Add all remaining digits.

4. Add result 2. to result 3.

5. The check digit is the smallest number which when added to the result of step 4. produces an exact multiple of 10.

For example, the check digit for the number 401234567890 is calculated as follows:

0+8+6+4+2+0= 20

20 x 3 = 60

9+7+5+3+1+4 = 29

60 + 29 = 89

89 + 1 = 90 - therefore the check digit is 1.

In both the UPC-A and EAN13 bar code symbologies 12 digits are encoded by the bar/space pattern above the human readable character. The major difference with EAN13 is that the 13th character (the one to the left of the code) is encoded by the variable parity of the left hand side of the code. The EAN number system uses 3 differing character sets (A,B and C). In each character set the bar/space pattern for a given digit is different. The right half of the code uses only character set C whereas the left half can use a mixture of sets A and B. It is the pattern of the mixture which determines the 13th character. For example if the left hand half of the code used ABBAAB then the 13th digit would be 5.

In common with other symbologies, EAN has a set of allowed tolerances for the quality of the printed code. These measurements relate to the maximum variation in the width of a single bar or space and vary according to magnification factor:

Magnification Factor	Tolerance µ (microns)
80%	±35µ
85%	±51µ
90%	±69µ
95%	±85µ
100%	±101µ
105%	±108µ
110%	±115µ
115%	±124µ
120%	±132µ
125%	±140µ
130%	±147µ
135%	±152µ
140%	±163µ

As the above table shows, the tolerances allowed reduce rapidly for magnification factors below 100%. For this reason the final print method should be considered carefully when choosing a magnification factor. For example, some printing processes are too variable to consistently print small (e.g. 80%) EAN codes.

Barcodes creation and programming.

To enable the printing of Barcodes to be created and embedded into a Printable document, The Tempest DSM uses a defined set of PCL compliant Escape sequences. The Tempest DSM will interpret a series of special font commands which are called within the PCL5 escape sequences.

These escape sequences follow a standard command set format and structure. An understanding of how PCL compliant printers and the PCL control code are interpreted would be beneficial.

When an alternative PCL barcode escape sequence is created the code does not process the command by printing a combination of Fonts and barcodes it calls a sequence of special font calls, and defines the barcode type, format and size of the output. This is achieved by a mathematically produced image.

Basic Control Sequence Commands

At the start of any barcode command sequence an <Esc> character will be defined, This is represented in an ASCII character map as ASCII 27 or as a Hex Code format as Hex1B.

See ACII Code Table Appendix 1.

Other Termination codes can be used based upon the data formats as listed below.

Numeric codes use either:	Space/CR/LF/FF/ESC
Alphanumeric codes use either:	CR/LF/FF/ESC

A type face selection sequence in PCL uses a typeface “T” number. Tempest DSM uses a reserved set of “T” numbers in the range (24580-24900) to initiate the barcode, and uses differing termination sequences depending on the barcode type.

The Special Call Escape sequence format

The PCL special font call command sequence follows similar rules to normal PCL escape codes.

The special font calls take the form:

<Esc>(s#p#h#v#b#s#T

The escape code command calls format are defined as follows:

<Esc>(s	required
#p	Print text method
#h	Human readable text font
#v	Vertical Sizing
#b	Barcode Widths
#s	Space Widths
#T	Type of Barcode
#	represents special parameters to call options within each code attribute:

Example PCL Escape sequence.

A valid PCL sequence to select a barcode is as follows:

<Esc>(s5p101h50v10,30b15,40s24600T

<Esc>	The Escape character
(s	start sequence
5p	Set Readable text Above the bars, without the Start/Stop Chars(*)
101h	Set Text Font to Letter gothic Regular
50v	Set Bar Height to 50 Point
10,30b	Set Thin Bars at 10 dots wide, thick bars at 30 dots wide
15,40s	Set Thin spaces at 15 dots wide, thick spaces at 40 dots wide
24600T	Select barcode type UPC-A

Barcode Types are supported by the “T” sequence.

Validation of Checksums

Checksums are additional characters appended to bar codes to guarantee good reads. Checksums are necessary on some bar codes that are prone to errors. For example, Interleaved 2 of 5 is a very dense, numeric-only bar code, but it is prone to substitution errors. You should always use a checksum on this code. Other codes, such as Code 128 and Code 39, are self-checking and seldom require a checksum. The will automatically perform these checksum calculations where needed. If a checksum is provided on a fixed length code such as EAN13, it is ignored.

Barcode Sizing and point size range.

Not all barcodes can be resized. If a request to scale a barcode is not allowed within the barcode format standard the printable output will create a default configuration. This also applies to any parameters that have not been set within the Escape code sequence. The Tempest DSM once again will use the default settings. Scaling of single dimension (1D) barcodes via changes to the X and Y axis is supported. The Tempest DSM supports point sizes from 3 to 9600. in addition some 2D barcodes can be resized by columns/rows or by defining a ratio.

The current cursor position is at the bottom left corner of the leftmost black bar.

Example: if the height parameter is not provided the default size is used.

If required the Tempest DSM will print the data as centred text with the bars, either under, half-embedded, full embedded or above the bars. Checksum and flag characters are automatically placed in the right place for certain codes, according to standards. Automatic sizing limits the embedded text point size to 15 (there is no limit for text above or below the barcode).

After the barcode is printed, the cursor is set to the bottom right corner of the rightmost black bar.
If another barcode needs to be printed, you can move the cursor to a new location and send new barcode data.

Barcode Escape Sequence Structure examples

As previously explained to make the unit generate a barcode as part of a print job an escape sequence needs to be embedded in the data sent to the box.

As an example you would like to produce a default barcode for EAN13 with a code number of 012345678901.

In this case the barcode number has to be 12 digits because EAN13 requires that for a complete barcode. Below is the escape sequence:

<ESC>(sphvbs24630T012345678901

The word <ESC> is in place of the actual character escape, which is 27 decimal, or 1B hexadecimal. The complete print job you would have the following:

<ESC>E<ESC>(sphvbs24630T012345678901**<ESC>E**

This sequence only produces a default size and style of EAN13 barcode.

It is possible for the system to change how the barcode is printed. Below is the same escape sequence but it shows where the numeric parameters can be inserted.

<ESC>(s**0**p**0**h**0**v**0**,**0**b**0**,**0**s24630T012345678901

This will still produce a default EAN13 barcode. But if we change the first parameter, which controls text printing to say 5, we get the text above the barcode instead:

<ESC>(s**5**p0h0v0,0b0,0s24630T012345678901

This p parameter has the following other option:

default for the given barcode = 0
don’t print human readable barcode data = 1
print human readable barcode data embedded = 2
print human readable barcode data half-embedded = 3
print human readable barcode data below = 4
print human readable barcode data above = 5

The parameter h, controls text style for the human readable barcode data. It is actually broken down into three parts. If we change it to 304 then we get bold CG Times used for the human readable barcode data. So now the sequence looks like this:

<ESC>(s0p**304**h0v0,0b0,0s24630T012345678901

The 3 in the 304 selects bold, the 0 is reserved and therefore is always 0, and the 4 select the CG Times font below are the other useable codes:

First digit codes:

default for given barcode = 0
use regular = 1
use italics = 2
use bold = 3
use bold & italics = 4

Middle digit codes:

Always zero and any other value reserved

Last digit codes:

default for given barcode = 0
selected letter gothic font for human readable barcode data = 1
select Univers font for human readable barcode data = 2
select Univers condense for human readable barcode data = 3
use CG Times font for human readable barcode data = 4
use OCR-B font for human readable barcode data (may not be supported on some printers) = 5

The v parameter affects the size of the barcode in terms of height, the default for EAN13 is 62 point (or 1/72 of an inch). If you wish to convert this to 600 dpi size then divide the point size by 0.12 therefore 62/0.12 gives you 516 dots height. If you want to work back from a given height e.g. two inches which is 1200 dots then multiply by 0.12 1200*0.12 given you 144 point.

So the escape sequence would look like:

<ESC>(s0p0h**144v**0,0b0,0s24630T012345678901

The next two parameter b’s control bar widths. The first number before the comma is the narrow bar width, which is then followed by the wide bar width. Both values are in terms of 600 dpi. The default for EAN13 is 8 dots for the narrow bar and 16 for the wide bar. The following sequence would not change your output bar code from the default:

<ESC>(s0p0h0v8,**16b0,0s**24630T012345678901

It is best not change the bar width without adjusting the space widths, which work in a similar way except the gaps between bars. Again the default is 8 for narrow space and 16 for wide space so therefore this sequence does not change the output from default:

<ESC>(s0p0h0v0,0b8,16s24630T012345678901

So let’s say we want the barcode for EAN13 to be twice the size of the default. We would increase the height from 62 points to 124 point and double both the narrow bar and space size from 8 to 16 then double the wide bar and space from 16 to 32. Therefore the sequence would look like this:

<ESC>(s0p0h124v16,32b16,32s24630T012345678901

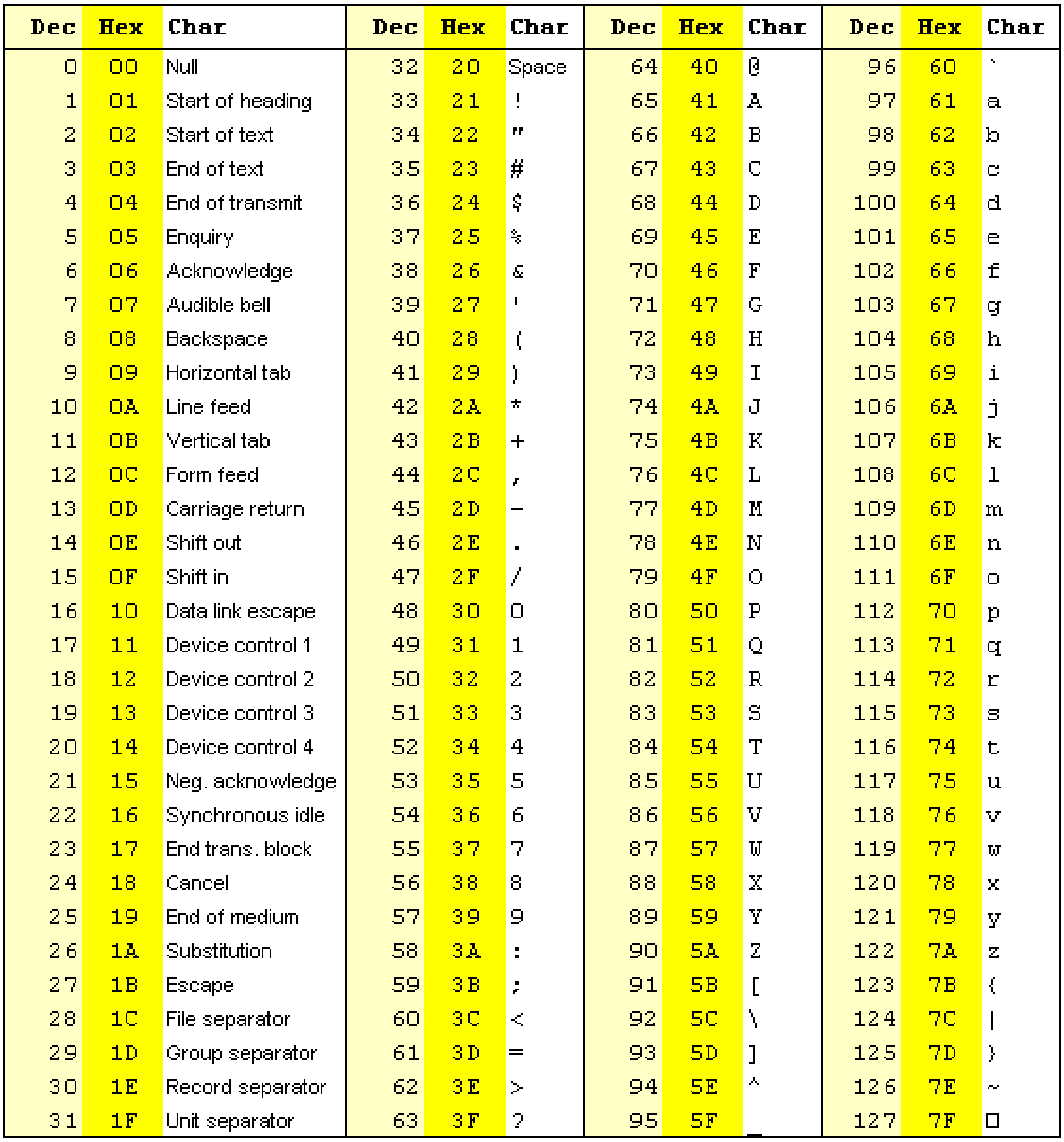
The last parameter T controls what kind of barcode you wish to use. If you wanted to use UPC-A instead, you would change the 24630 to 24600 like this:

<ESC>(s0p0h0v0,0b0,0s**24600T**01234567890

Note that UPC-A only requires 11 digits for the correct bar coding not 12.

Appendix 1

ASCII Code Table



Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Appendix 2

ASCII Code Set extended Characters

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

Glossary of Terms

Article Numbering Association (A.N.A.) The co-ordinating body in the U.K. for article numbering using the EAN, ITF-14 and EAN-128 standards. Now renamed the e-Centre.

AIM Automatic Identification Manufacturers Association. This organisation has branches in most countries or regions. It is supported by manufacturers and suppliers of automatic ID equipment and services.

Alphanumeric A character set including numbers and letters.

ASCII The American Standard Code for Information Interchange. This character set is used for **standard communication between electronic devices such as computers and scanners.**

Bar Width Reduction The thinning of bars within a bar code which compensates for print gain.

Bearer Bars The protective printed lines or box around a bar code which hDSM to support the printing plate and reduce the squashing effect on the bar code image.

Code128 A full alphanumeric bar code type which can encode the entire ASCII character set.

Code39 An alphanumeric bar code type which can encode capital letters, numbers and 7 special characters.

Codabar A bar code type which may encode 0-9 and six special characters. Four different start and stop patterns are used to signify different uses.

CPI Characters per inch.

EAN European Article Number. The international standard for coding retail goods.

Font The design or typestyle of a complete character set of printers type.

Guard Bars The bar space patterns at the start, centre and end of EAN and UPC codes. Used as a reference point for the scanning device.

HIBC The Health Industry Bar Code. The format for coding goods within the U.S. health industry using conventional symbologies.

Interleaved 2 of 5 A numeric only bar code symbology. Characters are encoded in pairs with the bars representing the first character and the spaces representing the second.

ISBN International Standard Book Number. Represented by the ISBN/EAN bar code on books.

ISSN International Standard Serial Number. Represented on magazines and periodical publications by the ISSN/EAN bar code.

Magnification The method of describing bar code width and height in some symbologies. 100% magnification is nominal size.

Micron One thousandth of a millimetre.

Narrow Bar The width of a single module or narrowest bar in some symbologies.

Nominal Size The standard size for a bar code of a given symbology.

Numeric Representation using only the digits 1,2,3,4,5,6,7,8,9 and 0.

OCR-A and OCR-B Character sets or fonts both of which may be machine or human read. Used for the human readable information in bar coding.

Parity A way of encoding bar/space patterns using an odd or even number of modules bars or spaces. The parity may be calculated by the scanner to provide an error check.

PostScript The high-quality image description language developed by Adobe Systems Inc. which uses vectors to describe graphical elements.

Printer Compensation A method of adjusting bar width to compensate for a bias on a print device.

Quiet Zone The light space at either side of a bar code which is needed by the scanner to detect the start and end of the code.

Raster A series of dots or cells arranged in a regular grid.

Redundant - See **Film Master**

Reflectance The ability of a surface or printed area to reflect light.

Start Character The specific bar/space pattern which indicates the start of a bar code.

Stop Character The specific bar/space pattern which indicates the end of a bar code.

Truncation The reduction in bar height below nominal height or that normally specified.

Uniform Code Council (U.C.C.) The U.S. organisation which administers the UPC and other retail specifications.

Universal Product Code (U.P.C.) The standard bar code type for retail products in the U.S.

Verifier A device used to measure the bars and spaces of a bar code and report on their accuracy in relation to the specified tolerances.

Wide/Narrow Ratio The ratio between the narrow bar and the wide bar widths in certain bar code symbologies.

X dimension The nominal width of a narrow bar