Océ|User manual

PRISMAproduction Server

Technical Reference Manual

...and Training?

For this product we also offer seminars at our Training Center in Poing

Information:
Phone+49 8121 72-3940
Fax+49 8121 72-3950

Océ Printing Systems GmbH
ITC
Postfach 1260
85581 Poing
Germany

December 2008 Edition

A29247-X100-X-7-7672

Release 4.00.02

Contents

1 Print Parameter

1.1 Available Tuning Sets

Predefined tuning sets allow you to easily modify the print parameters of AFP/Linedata printers e.g. to enable traces or to drive non Océ printers.

A tuning set is a file, stored in the directory:
`/u/prismapro/cfg/printers/.system/afp.opt/.`
The files refer to sps or ftp-type printers.

The administrator or service engineer can put additional tuning sets into this directory.

The format of such files is the usual ini-file-format. The contents depends on the printer device type (described in the "print parameters" tab). The name of the tuning set, which appears in the list, can be specified with a 'Name' statement for each language before the 1st section.

At the moment the following tuning sets are available:

Name= **Enable all File Infopages**
[HEADER_PAGE_PROCESSING]
HDRP_ENABLING=1
[MESSAGE_PAGE_PROCESSING]
MSGP_ENABLING=5
[SEPARATOR_PAGE_PROCESSING]
SEPP_ENABLING=2
[TRAILER_PAGE_PROCESSING]
TRLP_ENABLING=1

Name= **Disable all Job Infopages**
[GROUP_HEADER_PAGE_PROCESSING]
GHDRP_ENABLING=0
[GROUP_MESSAGE_PAGE_PROCESSING]
GMSGP_ENABLING=0
[GROUP_SEPARATOR_PAGE_PROCESSING]

GSEPP_ENABLING=0
[GROUP_TRAILER_PAGE_PROCESSING]
GTRLP_ENABLING=0

Name= **Enable all Job Infopages**
[GROUP_HEADER_PAGE_PROCESSING]
GHDRP_ENABLING=1
[GROUP_MESSAGE_PAGE_PROCESSING]
GMSGP_ENABLING=1
[GROUP_SEPARATOR_PAGE_PROCESSING]
GSEPP_ENABLING=1
[GROUP_TRAILER_PAGE_PROCESSING]
GTRLP_ENABLING=1

Name= **Disable all File Infopages**
[HEADER_PAGE_PROCESSING]
HDRP_ENABLING=0
[MESSAGE_PAGE_PROCESSING]
MSGP_ENABLING=0
[SEPARATOR_PAGE_PROCESSING]
SEPP_ENABLING=0
[TRAILER_PAGE_PROCESSING]
TRLP_ENABLING=0

Name= **Activate media name mismatch (MMT) message**
[PRINT_PROCESSING]
PDSFLG7=0x04
PDSISUB=0

Name= **Deactivate MMT adjustments**
[PRINT_PROCESSING]
PDSFLG7=0x10
PDSISUB=1

Name= **Activate media attribute mismatch (MMT) message**
[PRINT_PROCESSING]
PDSFLG7=0x02
PDSISUB=0

Name= **Disable all traces**
[TRACE]
TRACEFILE_LIMIT=10240
TRACEFLAGS=0x00000000

Name= **Enable full traces**
[TRACE]
TRACEFILE_LIMIT=1900000
TRACEFLAGS=0xffffffff

Tuning Sets for "Non Océ" Printers

For printing on "non-OCÈ" Printers one additional tuning set has been introduced.

This means that there are now 3 tuning sets, setting small printer memory values (the same for all 3 tuning sets) and different LUPUB values:

Name= **Non Oce Printers (Host) and APA controller adjustment**
[CONFIGURATION]
OBUFSIZE=0x2000
[PRINT_PROCESSING]
PRTUPUB=0x0960

Name=**Non Oce Printers (300 DPI e.g. LCDS -> Xerox)**
[CONFIGURATION]
OBUFSIZE=0x2000
[PRINT_PROCESSING]
PRTUPUB=0x0bb8

Name=**Non Oce Printers (600 DPI / MRM e.g. -> IBM)**
[CONFIGURATION]
OBUFSIZE=0x2000
[PRINT_PROCESSING]
PRTUPUB=0x3840

For performance reasons, the settings may be changed manually in respect to e.g. actual memory size of the respective printer.

1.2 AFP2IPDS Backend

1.2.1 Introduction

The AFP2IPDS backend, also called SPS (Smart Print Subsystem), receives data in line data or structured field format. It composes data for printing according to the parameters defined for the job. (Parameter description see ...). This backend is especially designed to support the enhanced functionality of the Océ high volume printers, which is partially outside the scope of "standard" IPDS.

The basic functions of the backend are:

- Together with print resources data is converted to the printer-specific IPDS format.
- Print resources can be embedded in print data; if not, they are retrieved from one or more external resource libraries. (Resource handling see ...)
- The backend also determines the alignment of the print pages on the physical medium: depending on how parameters are set, two consecutive or identical pages can be printed side by side, for instance, and printing can be on either one or both sides of the medium. If specified in the print parameters, the backend also inserts information pages (header, separator, message and trailer pages).
- The backend reacts directly and efficiently to problems that occur during execution of print jobs.
- It detects and distinguishes device, data and user errors and reports them.

1.2.2 AFP2IPDS Output Backend and PRISMAproduction Server

The mission of the AFP2IPDS backend in the server context is to enable the implementation of an AFP2IPDS capable output management system.

The AFP2IPDS backend is running under the control of the PRISMAproduction Output Distribution System (ODS). It is fully integrated in the PRISMAproduction Server workflow. Océ proprietary interfaces are used to allocate the jobs and access the print data which is stored in the PRISMAproduction spool. Different types of printer attachments are supported (SCSI, TCP/IP, /370 channel). Multiple instances of the AFP2IPDS backend may be running in parallel. The number of them is only restricted by the resource limitations of the system.

1.2.3 Parameter Description

1.2.3.1 CONFIGURATION

HIGHWATERMARK	Defines the amount of maximum pages in the internal pagebuffer. If the maximum is reached page generation will be stopped. HighWaterMark must be higher than LowWaterMark, if not for both values the intial values will be used. Default = 64
LOWWATERMARK	Defines the amount of minimum pages in the internal pagebuffer. If the minimum is reached new pages will be generated. LowWaterMark must be lower than HighWaterMark, if not for both values the intial values will be used. Default = 8
MEMLIMIT	MEMLIMIT defines the maximum amount of memory in [MB], which the backend can use. If the limit should be reached, then the backend is terminated immediately and a core dump is created for analysis. Default = 200
NPRO_FILE	Filename of the printfile for NPRO-Pages. Default = NPROPAGES.AFP
NPRO_FORMDEF	Filename of the NPRO-Formdef. Default = F1D0101
NPRO_PAGEDEF	Filename of the NPRO-Pagedef.
OBUFSIZE	This size is limited to max. 0xfe00. Size of the output buffers to be used for sending data to the printer. This size can have a significant influence on performance if too small. For A-Twin printers, this parameter specifies the number of output buffers used for each physical printer. For values greater than 0x2000(8K) please verify whether your printer and channel attachment (like channel extenders) support it. This value should be 0x8100 for printers with SRA I controller (and 0x2000 for printers with APA controller). 0x2000 is used by several compatible IBM printers (e. g. IBM-Infoprint 21) Default = 0xFA00

PRTACKR	Number of physical pages that AFP2IPDS backend will attempt to send to the printer before requesting an Aknowledgment Reply with updated page counter information. Specifying zeroes means that AFP2IPDS backend will use its internal default. Increasing this parameter may be used to improve the performance for TCP/IP connected printers. ATTENTION: This parameter is evaluated only, as long as position check is deactivated and undefined character codes are ignored (see parameter "PDSDCK=0x03" and parameter "CKPTTIME=0"). Default = 0
PSINUFON	Limits the number of fonts that will be downloaded to the printer. 0 use the printers limit. 30000 This value should be used for DS8090 printers. 128 default value A value between 1 and 99999 will be checked at the end of every physical page. When the value is reached, all fonts will be cleared from the printer and the printer will be loaded with the resources needed for the next page. If the value is exceeded again by the number of fonts for that page, the job will be requeued as unprintable.
PSINUOVE	Limits the number of overlays that will be downloaded to the printer. 0 use the printers limit. 100 default value A value between 1 and 99999 will be checked at the end of every physical page. When the value is reached, all overlays will be cleared from the printer and the printer will be loaded with the resources needed for the next page. If the value is exceeded again by the number of overlays for that page, the job will requeued as unprintable.

PSINUPSE	Limits the number of pagesegments that will be downloaded to the printer. 0 use the printers limit. 100 default value A value between 1 and 99999 will be checked at the end of every physical page. When the value is reached, all pagesegments will be cleared from the printer and the printer will be loaded with the resources needed for the next page. If the value is exceeded again by the number of pagesegments for that page, the job will requeued as unprintable.
TWINPAGEBUFFERSIZE	Is used to avoid PageBufferFull-Situations on the 2nd printer of an ATWIN, respectivly a TRIPLEX. Default = 64

1.2.3.2 DUMMYPRINTER

In dummy printing mode, the AFP2IPDS backend internally simulates all printer communication procedures and is able to process any print job as if there would be a physical printer present. In dummy mode the print datastream passes the complete AFP2IPDS backend process chain until the generated pages would be sent to the printer. Generated pages will not be sent but rather discarded at end of chain. To get rid of particular communication channel problems pages will be discarded before entering PAL, SCSI or S370 modules.

In dummy mode AFP2IPDS backend will still be able to run in multiple instances on the same machine. No further executable is required to run AFP2IPDS backend in dummy mode. You are able to switch to dummy printing directly.

DMY_LOAD_PRINTER	DMY_LOAD_PRINTER= string Load predefined or former saved printer profiles (with SAVE_PRINTER) Predefined printer keywords are: – VARIOPRINT5000 – VARIOSTREAM7000 – VARIOSTREAM7000_ATWIN – VARIOSTREAM9000 – IBM_IP32 – IBM_IP6400 – CUTSHEET as an alias of VARIOPRINT5000 – FANFOLD as an alias of VARIOSTREAM7000 – ATWIN as an alias of VARIOSTREAM7000_ATWIN 6400 as an alias of IBM_IP6400END_LOAD_PRINTER
DMY_SAVE_PRINTER	DMY_SAVE_PRINTER= string Save printer profile to file. This parameter allows to save a real printer as a printer profile during printing session. This parameter works in dummy mode (DUMMYDEV=1) as well in non dummy mode (DUMMYDEV=0). In dummy mode it may be used to save predefined printer profiles. In non dummy mode it may be used to save a profile of a real printer for later usage in dummy mode.
DMY_SPEED	Limits dummy speed on a per printer base (400ppm on a twin results in 800ppm overall speed). 0 No speed limit is set.
DUMMYDEV	Enable or disable the internal printer simulator. 0 Printer simulator disabled 1 Printer simulator enabled

1.2.3.3 MESSAGE_PAGE_PROCESSING, HEADER_PAGE_PROCESSING, TRAILER_PAGE_PROCESSING, SEPARATOR_PAGE_PROCESSING, GROUP_MESSAGE_PAGE_PROCESSING, GROUP_HEADER_PAGE_PROCESSING, GROUP_TRAILER_PAGE_PROCESSING, GROUP_SEPARATOR_PAGE_PROCESSING

Introduction

The AFP2IPDS backend is able to generate and print job information sheets.

There are two types of information pages:

- Group Information Pages
- Information Pages

The Group Information Pages will be printed, if enabled, between jobs and/or job copies, and are used by the operator to recognize the beginning and end of a job and/or of job copies.

The Information Pages will be printed, if enabled, between files and/or file copies, and are used by the operator to recognize the beginning and end of a file and/or of file copies.

In special cases, the AFP2IPDS backend also insert sheets containing text information (messages) describing special conditions which occurred while processing the job respectively the file (e.g. when automatic error recovery occurs).

AFP2IPDS has four types of system Information Pages for files: **Message**, **Header**, **Trailer** and **Separator** pages. These pages are normally printed between files and/or file copies, and may be selectively suppressed by the AFP2IPDS backend administrator, end user, operator, or systems programmer.

AFP2IPDS has four types of system Group Information Pages for jobs: **Group_Message**, **Group_Header**, **Group_Trailer** and **Group_Separator** pages. These pages are normally printed between jobs and/or job copies, and may be selectively suppressed by the AFP2IPDS backend administrator, end user, operator, or systems programmer.

The AFP2IPDS backend uses a facility available in the IPDS data stream to inform the printer that a system generated (as opposed to user data) information page is being printed. Please refer to your Océ printer documentation for more information about the handling provided for this kind of pages. This handling may include: automatic generation and printing of synchronisation bar codes (A-Twin and C-twin printers), automatic waste-bin routing (UP³I capable system), etc.

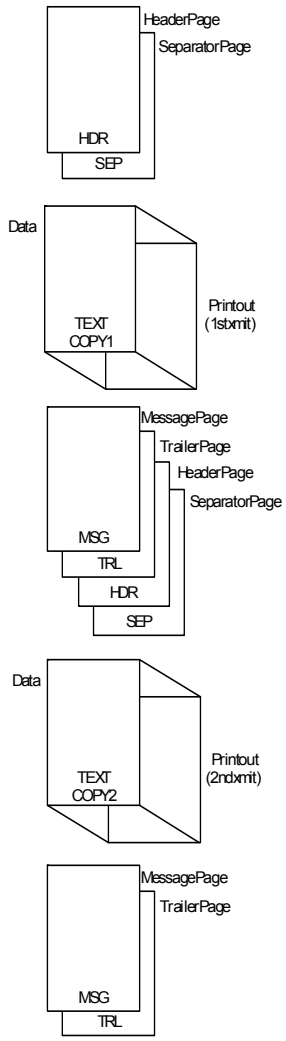
Customize Information Pages

It is possible to customize the PRISMAproduction-generated separator pages in two ways:

- You can influence if a specific separator page (either Trailer, Message, Separator or Header Page) is printed.
- You can completely customize the layout and content of these pages

Information when a separator page is to be printed is kept in the print parameter <xxx>_ENABLING (xxx stands for the information page type) which is located in the following sections:

- MESSAGE_PAGE_PROCESSING
- HEADER_PAGE_PROCESSING
- TRAILER_PAGE_PROCESSING
- SEPARATOR_PAGE_PROCESSING
- GROUP_MESSAGE_PAGE_PROCESSING
- GROUP_HEADER_PAGE_PROCESSING
- GROUP_TRAILER_PAGE_PROCESSING
- GROUP_SEPARATOR_PAGE_PROCESSING



Arrangement of header, separator, data, message and trailer pages.

Message pages are printed only if messages for the current print job have occurred. The messages can be error messages, warnings and repositioning messages.

To change the layout and content of a specific information page, it is necessary to change the corresponding Print Parameter which are described in the following table.

Custom specific infopages and resources for configured printers

The system automatically creates new runtime printer directories(in **/u/prismapro/data/prt**), which contain printer specific files, e.g. resources for infopages (**Please do not add any files here!**). This is done when an new printer is configured AND always on startup of the system. During this procedure, the contents of the old printer runtime directories is deleted, if they exist.

To use your own user-defined ressources in the printer, you have to set up the following printer runtime directories. These are built and filled in 5 steps:

1. Every configured printer leads to a sub-directory in `/u/prismapro/cfg/printers/` where the name of the subdirectory is the name of the printer. Exception: Names, starting with a dot "." are not valid printer names and are not printer directories.
2. The content of a template directory in `/u/prismapro/cfg/printers/` **`.system`**`/<XXX.YYY>` is copied into the created directory.
   Please do not modify this template directory, as it is not saved through updates!
3. The same procedure is repeated for the directories `u/prismapro/cfg/printers/` **`.custom`**`/<XXX.YYY>` and `u/prismapro/cfg/printers/` **`.custom`**`/<PRTNAME>`, but only if they exist.
   These directories are still supported for backward compatibility.
4. If exists, then the content of the directory **`/u/prismapro/cfg/printers/<XXX>.<YYY>`** is copied into the working directory.
   This is the directory to place the customer specific info page resources to be used for all printers.
5. If exists, then the files of the directory **`/u/prismapro/cfg/printers/<PRTNAME>`** are copied into the working directory.
   This is the directory to place the customer specific info page resources to be used for an individual printer.

Description of XXX, YYY and PRTNAME

"XXX" is the internal name of the general backend type, e.g. "afp" for AFP2IPDS-Backend:

XXX	Backend
afp	AFP2IPDS-Backend (IPDS Driver)
gen	Generic Backend
ops	PostScript Backend (Océ printers)
pjl	PJL Backend
xdp	Xerox DocuPrint Backend

"YYY" represents the supported resolutions:

YYY	Resolution
144	144 dpi for Infoprint_6400 and Infoprint_6500 printers
240	240 dpi
300	300 dpi
600	600 dpi and MRM-Printer

"PRTNAME" has to be the name of the created printer.

Parameter Description

MSGP_ENABLING HDRP_ENABLING TRLP_ENABLING SEPP_ENABLING GMSGP_ENABLING GHDRP_ENABLING GTRLP_ENABLING GSEPP_ENABLING	Defines printing of MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP pages. 0 Suppress MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP page 1 Enable MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP pages. MSG | HDR | TRL | SEP: After last file copy 2 MSG | HDR | TRL | SEP: After all file copies 4 MSG | HDR | TRL | SEP::At repositioning 8 MSG | HDR | TRL | SEP: Mark form after last file copy 16 NPRO before MSG page
MSGP_TEMPLATE HDRP_TEMPLATE TRLP_TEMPLATE SEPP_TEMPLATE GMSGP_TEMPLATE GHDRP_TEMPLATE GTRLP_TEMPLATE GSEPP_TEMPLATE	Filename of the MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP page template file.
MSGP_FORMDEF HDRP_FORMDEF TRLP_FORMDEF SEPP_FORMDEF GMSGP_FORMDEF GHDRP_FORMDEF GTRLP_FORMDEF GSEPP_FORMDEF	MSGP | HDRP | TRLP | SEPP | GMSGP | GHDRP | GTRLP | GSEPP_FORMDEF=F1INFOPG Defines the MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP page Formdef.
MSGP_PAGEDEF HDRP_PAGEDEF TRLP_PAGEDEF SEPP_PAGEDEF GMSGP_PAGEDEF GHDRP_PAGEDEF GTRLP_PAGEDEF GSEPP_PAGEDEF	MSGP | HDRP | TRLP | SEPP | GMSGP | GHDRP | GTRLP | GSEPP_PAGEDEF=P1INFOPG Defines the MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP page Pagedef.

MSGP_DCK HDRP_DCK TRLP_DCK SEPP_DCK GMSGP_DCK GHDRP_DCKH GTRLP_DCK GSEPP_DCK	MSGP | HDRP | TRLP | SEPP | GMSGP | GHDRP | GTRLP | GSEPP_DCK **READONLY** Bitmask to disable data checks: 0x00 All data checks are reported by the printer. 0x01 Printer ignores printing outside of the printable area. 0x02 Printer ignores character codes that are undefined in the code page of the font and prints the default character instead. 0x03 Is a combination of 0x01 and 0x02 (Default) **ATTENTION**: This parameter can't be changed, because no data- and/or position-checks may appear at a MSG | HDR | TRL | SEP | GMSG | GHDR | GTRL | GSEP page!
MSGP_X2UP HDRP_X2UP TRLP_X2UP SEPP_X2UP GMSGP_X2UP GHDRP_X2UP GTRLP_X2UP GSEPP_X2UP	Specifies default Two-up processing: 0 No Two-up 1 Normal Two-up (left/right) 2 Identical copies 4 Inversed Two-up (right/left) No value means, use information from FORMDEF
MSGP_INPUT_BIN HDRP_INPUT_BIN TRLP_INPUT_BIN SEPP_INPUT_BIN GMSGP_INPUT_BIN GHDRP_INPUT_BIN GTRLP_INPUT_BIN GSEPP_INPUT_BIN	Specifies default Input Media source code (for cutsheet printers only): 0 Input Bin A 1 Input Bin B 2 Input Bin C 3 Input Bin D No value means, use information from FORMDEF
MSGP_OUTPUT_BIN HDRP_OUTPUT_BIN TRLP_OUTPUT_BIN SEPP_OUTPUT_BIN GMSGP_OUTPUT_BIN GHDRP_OUTPUT_BIN GTRLP_OUTPUT_BIN GSEPP_OUTPUT_BIN	Specifies the Output Media destination (for cutsheet printers only): 0 use the default output tray of the printer 1 Output tray A is selected 2 Output tray B is selected **Note**: In this case remote tray selection must be enabled at operator panel of the printer as well !

MSGP_DUPLEX HDRP_DUPLEX TRLP_DUPLEX SEPP_DUPLEX GMSGP_DUPLEX GHDRP_DUPLEX GTRLP_DUPLEX GSEPP_DUPLEX	Specifies default Simplex/duplex information: 0 Simplex 1 Normal duplex 2 Tumble duplex No value means, use information from FORMDEF
MSGP_FLG2 HDRP_FLG2 TRLP_FLG2 SEPP_FLG2 GMSGP_FLG2 GHDRP_FLG2 GTRLP_FLG2 GSEPP_FLG2	Bitmask for following options: 0x02 Burst stacking request 0x04 EOT mark/Offset stacking request at Begin of 1. copy of file 0x20 EOT mark/Offset stacking request at Begin of each copy 0x40 Swapped printing request (A-Twin printers only, front- and backpages will be swapped).
MSGP_SOSI HDRP_SOSI TRLP_SOSI SEPP_SOSI GMSGP_SOSI GHDRP_SOSI GTRLP_SOSI GSEPP_SOSI	Defines double byte mode: 0 Shift-out/shift-in codes are ignored. 1 Shift-out/shift-in codes are recognized and replaced with spaces. 2 Shift-out/shift-in codes are recognized

MSGP_PRINT_DIRECTION HDRP_PRINT_DIRECTION TRLP_PRINT_DIRECTION SEPP_PRINT_DIRECTION GMSGP_PRINT_DIRECTION GHDRP_PRINT_DIRECTION GTRLP_PRINT_DIRECTION GSEPP_PRINT_DIRECTION	Information about the print direction: 0 PDSDUPLX field (if specified) overrides the duplex and orientation information of the FORMDEF. 1 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is used unchanged. 2 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is force to be PORTRAIT. 3 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE. 4 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is force to be PORTRAIT 90. 5 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE 90. 6 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is force to be PORTRAIT 180. 7 PDSDUPLX field (if specified) overrides the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE 180.
MSGP_COLMAP HDRP_COLMAP TRLP_COLMAP SEPP_COLMAP GMSGP_COLMAP GHDRP_COLMAP GTRLP_COLMAP GSEPP_COLMAP	Defines a colormap resource.

MSGP_FRONTSIDEOVLS HDRP_FRONTSIDEOVLS TRLP_FRONTSIDEOVLS SEPP_FRONTSIDEOVLS GMSGP_FRONTSIDEOVLS GHDRP_FRONTSIDEOVLS GTRLP_FRONTSIDEOVLS GSEPP_FRONTSIDEOVLS	List of overlays to be placed on the front side of each sheet. List delimiter is ';'
MSGP_BACKSIDEOVLS HDRP_BACKSIDEOVLS TRLP_BACKSIDEOVLS SEPP_BACKSIDEOVLS GMSGP_BACKSIDEOVLS GHDRP_BACKSIDEOVLS GTRLP_BACKSIDEOVLS GSEPP_BACKSIDEOVLS	List of overlays to be placed on the back side of each sheet. List delimiter is ';'
MSGP_XOFFXB HDRP_XOFFXB TRLP_XOFFXB SEPP_XOFFXB GMSGP_XOFFXB GHDRP_XOFFXB GTRLP_XOFFXB GSEPP_XOFFXB	X-offset on the reverse side of the sheet: Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also parameter PRTUPUB.
MSGP_XOFFXF HDRP_XOFFXF TRLP_XOFFXF SEPP_XOFFXF GMSGP_XOFFXF GHDRP_XOFFXF GTRLP_XOFFXF GSEPP_XOFFXF	X-offset on the front side of the sheet: Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also parameter PRTUPUB.
MSGP_XOFFYB HDRP_XOFFYB TRLP_XOFFYB SEPP_XOFFYB GMSGP_XOFFYB GHDRP_XOFFYB GTRLP_XOFFYB GSEPP_XOFFYB	Y-offset on the reverse side of the sheet: Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also parameter PRTUPUB.

MSGP_XOFFYF HDRP_XOFFYF TRLP_XOFFYF SEPP_XOFFYF GMSGP_XOFFYF GHDRP_XOFFYF GTRLP_XOFFYF GSEPP_XOFFYF	Y-offset on the front side of the sheet: Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also parameter PRTUPUB.
MSGP_CC HDRP_CC TRLP_CC SEPP_CC GMSGP_CC GHDRP_CC GTRLP_CC GSEPP_CC	Specifies if carriage control characters are availabe (1) or not (0). Only ASA carriage control characters are supported. Allowed ASCII values are: hex representation Blank (0x020) 0 - 9 (0x30 -0x39) A (0x41) B (0x42) C (0x43) / (0x2F) + (0x2B)
MSGP_TRC HDRP_TRC TRLP_TRC SEPP_TRC GMSGP_TRC GHDRP_TRC GTRLP_TRC GSEPP_TRC	Specifies if table reference characters are availabe (1) or not (0). Allowed ASCII values are: hex representation 0 - 9 (0x30 -0x39) If more than 10 TRCs are requested, the user has to define the requested fonts by his PAGEDEF

MESSAGE

PIMSG	<value>	Maximal number of error messages before cancelling job. When this threshold is reached the job is cancelled and a message 17002229 is written.
	0	No limit of warning messages. No extra messages are printed unless there is an error that forces termination of the printing of the data set.

1.2.3.4 PRINT_PROCESSING

ACTIVATE_MEDIA_SIZE	A Media Description MDD in the formdef specifies the size used in the IPDS command XOH SMS Set Media Size. 0 disable the control of variable media size 1 enable the control of variable media size
BACKSIDEOVLS	List of Medium Overlays to be placed on the back side of every sheet. Each entry in the list has the format: C8 Overlay name. The standard end-of list indicator is used (';'). PS uses this field when the PDS2OVLB bit is set. Up to 8 names (1 to 8 character) specifying the medium overlays to be placed on the Back side of each sheet, in addition to overlays from other sources.
CKPTPAGE	Specifies the number of logical pages to be printed or transmitted before AFP2IPDS backend takes a checkpoint.. Zero means that no page controlled ckpt is to be done Default: 100
CKPTTIME	Specifies how many seconds of printing are to elapse between each checkpoint of this printed data set. In units of 1/100 seconds. ATTENTION: Changing this parameter to a value != "0" will cause that parameter "PRTACKR" to be ignored! Default: 0
COLOR_SETUP_NAME	Name of the "Color Setup File" that will be activated on the printer. This function is only supported for the VarioPrint and VarioStream printers which have the functionality to activate the printer setup.
COMP38MO	3800 compatible media origin 0 off (Default) 1 on
CSE_EJECT	Controls additional sheet ejects between jobs, sections and info pages, when printing in Cut Sheet Emulation mode. 0 No additional sheet ejects (save paper) 1 Additional sheet ejects (Start at defined position)
DEFAULT_CHARACTER	Defines the default character if a character is not available in the code page.
FONTS	List of the default font, if no other fonts are specified.
FORM	Form identifier of physical form to be used. PS uses this id to request operator intervention when a form mount is required.

FORMDEF	Defines a default Form definition (FORMDEF) name. For jobs using inline resources specify DUMMY. A 1 to 8 character name of the form definition to be used in printing the print data stream. The complete name must be specified. No prefix is added by AFP2IPDS backend. Defines default Formdef (normally not used). For jobs using inline resources specify DUMMY. If there is no Formdef defined in the job (e.g. PJM-Ticket), then this defined Formdef is used. Exceptions: If on the other hand the Formdef "DUMMY" is defined in the job (e.g. PJM-Ticket), first it is checked for an Inline Formdef, if the inline is not present, then this defined Formdef is used instead. F10101 for cutsheet and F10101LA for fanfold printers are available in the default printer resource path. For all other Formdefs the resource path must be specified e.g. by using the parameter LIB_FORMDEF or by creating a new customer printer resource path (/u/prismapro/cfg/printers/.custom/...).
FORMLEN	Value of the FORMLEN Parameter (in 11440 inches) Overrights the information of the FORMDEF. **Note**: The value of the printer must be greater. Default: 0
FRONTSIDEOVLS	List of Medium Overlays to be placed on the front side of every sheet. Each entry in the list has the format: C8 Overlay name The standard end-of list indicator is used. PS uses this field when the PDS2OVLF bit is set. Up to 8 names (1 to 8 character) specifying the medium overlays to be placed on the Front side of each sheet, in addition to overlays from other sources.
LINEMERGE	Determines shich kind of line merging is to be used to convert the input data 0 Specifies that the 3800 compatible line merging is disabled (Default). The standard AFP line merging (which overprints characters) is used. 1 Specifies that the 3800 compatible line merging is to be used. In this case, the characters contained in two or more input lines are merged in order to produce one single output line. The merge process is controlled by the carriage control.

LONG_JOB_NAME	Defines the length of the Jobname which is shown on the operator control panel of the printer. This also applies to a TrueProof Printer. NO: 8 EBCDIC characters are shown (MVS format) YES: Maximum of 64 ASCII character are shown (AIX/NT format) **Note**: The actual Jobname length is dependent of the configuration and/or maximum length of the operator control panel. Each printer can have different limits here.
NO_EDGEMARKS	Controls the printing of edgemarks: 0 edgemarks are printed 1 edgemarks are suppressed
PAGEDEF	Page definition (PAGEDEF) name. A 1 to 8 character name of the page definition to be used while converting S/370 Line format data. The complete name must be specified. Defines default Pagedef (normally not used). For jobs using inline resources specify DUMMY. If there is no Pagedef defined in the job (e.g. PJM-Ticket), then this defined Pagedef is used. Exceptions: If on the other hand the Pagedef "DUMMY" is defined in the job (e.g. PJM-Ticket), first it is checked for an Inline Pagedef, if the inline is not present, then this defined Pagedef is used instead. P1V06483 for cutsheet and P1STD3 for fanfold printers are avaibable in the default printer resource path. For all other Pagedefs the resource path must be specified e.g. by using the parameter LIB_PAGEDEF or by creating a new customer printer resource path (/u/prismapro/cfg/printers/.custom/...).
PDSCC	Specifies default carriage control character. PDSCCA 0x01 ASA control characters PDSCCI 0x02 IBM control characters PDSCCS 0x04 OCE(SIEMENS) control characters PDSCCX 0x08 XEROX data stream PDSCCSI 0x10 force single line spacing (16) PDSCCDO 0x20 force double line spacing (32) PDSCCTR 0x40 force triple line spacing (64)

PDSDCK	Bitmask to disable data checks 0x00 All data checks are reported by the printer PDSDCKPO 0x01 block invalid-position data checks (printing outside of the printable area)BLKPOS PDSDCKCH 0x02 block invalid-character data checks BLKCHAR (character codes that are undefined in the code page of the font and prints the default character instead) 0x03 Is a combination of 0x01 and 0x02 PDSDCKFL 0x04 block invalid-length FCB checks ATTENTION: Changing this parameter to a value != "0x03" will cause that parameter "PRTACKR" to be ignored!
PDSDUPLX	Specifies default Simplex/duplex information: -1 use information from FORMDEF 0 Simplex 1 Normal duplex 2 Tumble duplex No value or -1 means, use information from Formdef.
PDSFLG2	Bitmask for following options 0x01 Start every document at upperside page (fan-fold) 0x02 burst stacking request 0x04 EOT mark/Offset stacking requested at the beginning of the first transmission. 0x20 EOT mark/Offset stacking requested at the beginning of all transmissions. (Default) 0x40 Swapped printing requested (A-Twin printers only). The front and back sides are swapped. 0x80 Used whenPDS3FONT is set. If both are on, the font list specified in PDSFONT is ignored for an inline PAGEDEF.

PDSFLG7	PS processing dialog PDS7X2OF 0x01 Activate the X2UP-OFFSET Option may be used in case of RSTACK/ROFFSET processing for report separation and cut sheet application to be printed X2-up in continuous printing environment with inline split - merge post processing equipment. This option will break and insert blank pages on the right side of a 2-Up sheet. This allows proper offsetting with new reports starting on the left side of a 2-Up sheet. PDS7MMTD 0x02 Dialog if MMT-Attribute not matched PDS7MMTN 0x04 Dialog if MMT-Name not matched PDS7MMTO 0x08 Dialog if MMT-OID not matched PDS7USMT 0x10 Activate the Backend part of AIMS (Attribute-based Media selection) PDS7USPS 0x20 use printer setup file with printer alarm
	The combination of the parameters PDSISUB and PDSFLG7 are designed to enable / disable termination and errormessages. PDSISUB 1 and PDSFLG7 0x00 In case of a mismatch the printer use the default input bin. PDSISUB 0 and PDSFLG7 0x00 The printout will be terminated. PDSISUB 0 and PDSFLG7 0x02 Enable an operator dialog in case of a mismatch of a requested FORM withAttributes in the MMT, the attributes will be compared with the attributes of the UP3I input media SF of printers OPC PDSISUB 0 and PDSFLG7 0x04 Enable an operator dialog in case of a mismatch of requested FORM (XEROX notation stock-name: papername, AFP notation is MEDIANAME) PDSISUB 0 and PDSFLG7 0x08 Enable an operator dialog in case of a mismatch of requested FORM with an OID PDSISUB 0 and PDSFLG7 0x10 Enable an operator dialog in case of a mismatch of requested FORM (Activates the Backend part of AIMS (Attribute-based Media selection))

PDSGFLG	Grouping processing flag Bitmask for following options: 0x01 CANCEL KEY effects on all group members 0x08 EOT mark/Offset stacking request at the beginning of the first member of a group 0x10 Suppress delete resources between members
PDSIBIN	Specifies default Input Media source code (for CutSheet printers only) If this parameter is set, all other input bin selection is ignored. -1 default, use information from FORMDEF 0 Input Bin A 1 Input Bin B 2 Input Bin C 3 Input Bin D No value or -1 means, use information from FORMDEF
PDSISUB	ontrols the handling of error situations on the input side. 0 Terminate the print process if the selected input tray is not present in the printer. 1 Perform input tray substitution (like IBM PSF) i.e. if the requested input tray is not present in the printer, the default input tray is selected.
PDSOBIN	Specifies the Output Media destination (for cutsheet printers only): -1 use information from FORMDEF 0 use the default output tray of the printer 1 Output tray A is selected 2 Output tray B is selected The selection is done by physical media destination, and overrides all media destinations specified in the Form definition. **Note**: In this case remote tray selection must be enabled at operator panel of the printer as well!
PDSOFFXB	X-offset on the reverse side of the sheet: Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also Section PRTINFO parameter PRTUPUB. This field is used in connection with PDS5OFXB.

PDSOFFXF	X-offset on frontside of the sheet: Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also Section PRTINFO parameter PRTUPUB. This field is used in connection with PDS5OFXF.
PDSOFFYB	Y-offset on the reverse side of the sheet: Defines a new y-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 l-units/unitbase], the unitbase is given by the printer HW. See also Section PRTINFO parameter PRTUPUB. This field is used in connection with PDS5OFYB.
PDSOFFYF	Y-offset on frontside of the sheet: Defines a new y-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unitbase], the unitbase is given by the printer HW. See also Section PRTINFO parameter PRTUPUB. This field is used in connection with PDS5OFYF.

PDSPDIR	Information about thePrint direction PDSPD00 0 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex and orientation information of the FORMDEF. PDSPLEA 1 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is used unchanged. PDSPPOR 2 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is force to be PORTRAIT. PDSPLAN 3 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE. PDSPP90 4 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is force to be PORTRAIT 90. PDSPL90 5 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE 90. PDSPP180 6 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE 180. PDSPL180 7 PDSDUPLX/PDS3DUPX fields (if specified) override the duplex information but not the orientation information of the FORMDEF, which is force to be LANDSCAPE 180. 8 don't use orientation from FORMDEF, orientation is forced to be Portrait (only cutsheet printers). 9 use orientation from Formdef (only cutsheet printers). **Note**: If 8 or 9 is used, the setting of the MDDflg (by PPFA or SLE) is ignored. If N-up presentation is active, the orientation of the FORMDEF will always be passed to the cutsheet printer

PDSSOSI	Double-byte mode control PDSSOSI0 0 Shift-out/Shift-in codes are ignored PDSSOSI1 1 SO-SI codes are replaced with a blank character (X'40') and used to switch modes PDSSOSI2 2 SO-SI codes are just used to switch modes
PDSTRC	Translate table reference character PDSTRCN 0x00 No TRC present PDSTRCI 0x01 IBM TRC present PDSTRCS 0x02 SIEMENS TRC present PDSTRCX 0x03 XEROX TRC present The PAGEDEF determines whether a TRC should be processed in AFPDS mode, where 7 bits (IBM TRC) or 8 bits (OPS TRC) are used, or compatibility mode, where TRC validation takes place on the last 4 bits (bits 4-7) of the TRC byte, using the following rules: If the value of these bits is between 0 and 3 then they are used as TRC, other values (from 4 to 15) cause TRC 0 to be used. Xerox TRC indicates that the last 4 bits (bits 4-7) of the TRC are to be used to select up to 16 fonts, ignoring the first 4 bits.
PDSX2UP	Specifies whether the two up feature should be used or not. This feature is only available on Océ printers. Do not specify it unless the printer to be used support it. -1 default, (X2UP specified via resource files(FORMDEF) should be left unchanged). (ASIS) PDSX2OF 0 two up off (OFF) PDSX2NO 1 two up normal (left-right sequence) (ON/LEFT) PDSX2IC 2 indentical copy(two copies of each input page on each physical page) (ICOPIES) PDSX2RL 4 two up inversed (right-left sequence)(RIGHT) **Note**: For all values != -1 also the PDS3X2UP Bit in the PDSFLG3 field is set!

PRINTER_SETUP_NAME	Name of the "Printer Setup File" that will be activated on the printer. This function is only supported for the VarioPrint printers which have the functionality to activate the printer setup.
PRTANPRO	Specifies the number of additional sheets that AFP2IPDS backend should eject when executing a software driven NPRO. Valid for continous form printers only. 0 The ANPRO option is inactive. 1-999 Number of extra NPRO sheets to be ejected during the normal NPRO processing. Should only be used in connection with old printer controllers where the SPC (Stacked Page Counter) cannot be configured to include the post-processing device(s). Please check your printer documentation.
PRTUPUB	Number of l-units-per-unit-base support by the printer for a unit base of 10 inches. Values from X'0000' to X'7FFF' are allowed. 0x0000 (0): Whatever value is found in the AFP data is sent unchanged to the printer. Any other valid value will cause PS to convert the l-units-per-unit-base value in the input data to the value specified in this field. 0x0960 (2400) is used by several compatible IBM printers (e. g. IBM-Infoprint 21) 0x0BB8 (3000) for 300dpi datastreams e. g. LCDS->XEROX 0x3840 (14400) for 300dpi and 600dpi e. g. LCDS->IBM
SETUPIDS	List of setup verification ID's. These ID'S are referenced in the formdef and are checked to be present in the printer hardware. Delimiter is ';'.
XOHPODHD	Controls the resource management of the DS8090 printer in POD-Module configurations. 0 hard disk of the printer is not used for storage of the pagesegments. 1 hard disk of the printer is used for storage of the pagesegments.

1.2.3.5 Resource_Processing

The AFP2IPDS backend is able to detect, find and include the external AFP resources requested in the print data. The external resource library(ies) defined in the job ticket are used for this dynamic process. Context specific security rules are taken into consideration so that a print job may be restricted as to which resource libraries it is allowed to use.

Resource processing may be very time consuming as it is I/O intensive. This process is to be optimized by means of a **Resource Manager** which avoids searching and including resources which are already known to the printer.

The use of parallel processing (i.e. multiple execution threads running concurrently) is evaluated in order to shorten the elapsed time required to find and include the resources used by a job or page. Parallel processing is also used to convert resource formats in case they are not supported by the target device. This process involves decompressing and recompressing raster images (e.g. TIFF -> Fax G4).

Location of Resources

for host resources:

- **System libraries** (including security libraries) can be a concatenation of partitioned data sets, containing one or more members for one or more kinds of resources.
- **Private user libraries** resource library, that has an individual owner and has to be authorized by the job submitter.
- **Print file** can contain inline resources.

for resident resources:

- Printer (intermediate caching device) can store resident resource

Sequence of search for a resource

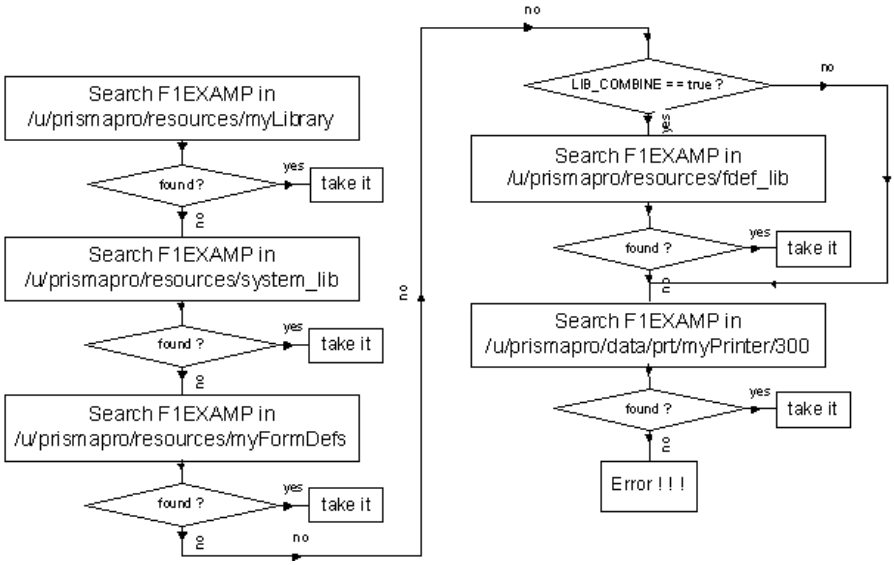
Sequence of search for a resource:

1. Inline resources in print file
2. Private user libraries
3. System libraries

Printer-resident resources are used only after a marked host resource is found.

Resource Search Procedure

```
Example searching the FORMDEF F1EXAMPL:

     Job-Definitions are:     LIB_COMMON =/u/prismapro/resources/myLibrary
                              LIB_FORMDEF=/u/prismapro/resources/myFormDefs
     Printer-Definitions are: LIB_COMMON =/u/prismapro/resources/system_lib
                              LIB_FORMDEF=/u/prismapro/resources/fdef_lib
                              SYSTEMDATA =/u/prismapro/data/prt/myPrinter/300
```



DSCASECTRL	Controls the spelling of resource names and extensions before searching the file in the attached resource libraries 0 read the resource name and the extensions in lower case 1 read the resource name and the extensions in upper case 2 do not convert the spelling of the resource name and the extensions 3 at first the resource name and the extendions are reading in upper case and afterwards in lower case (combination of "0" and "1") Please note that directory names (specified with the library parameters LIB_COMMON, etc.) are always handled without converting the spelling.

EXT_CODED_FONT	List of individual Resource File Name Extensions of coded fonts. Specifies one or more suffixes to be appended to an AFP CodedFont resource name while searching for the corresponding file. extension A variable length character string starting with a dot. Extensions are free definable and separated by ";". The default is no suffix. No extension is written as ;;. The search order is the same as the extensions are written from left to right. The first matching resource will be used. Example: EXT_CODED_FONT=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name X0BASE: X0BASE.600 X0BASE.300 X0BASE.240 X0BASE
EXT_CODE_PAGE	List of individual Resource File Name Extensions of code pages. Specifies one or more suffixes to be appended to an AFP CodePage resource name while searching for the corresponding file. extension A variable length character string starting with a dot. . Extensions are free definable and separated by ";". The default is no suffix. No extension is written as ;;. The search order is the same as the extensions are written from left to right. The first matching resource will be used. Example: EXT_CODE_PAGE=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name T1BASE: T1BASE.600 T1BASE.300 T1BASE.240 T1BASE

EXT_FONT_CHARSET	List of individual Resource File Name Extensions of font character sets. Specifies one or more suffixes to be appended to an AFP FontCharacterSet resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The extensions are free definable and separated by ";". The default is no suffix. No extension is written as ;;. The search order is the same as the extensions are written from left to right. The first matching resource will be used. Example: EXT_FONT_CHARSET=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name C0BASE: C0BASE.600 C0BASE.300 C0BASE.240 C0BASE
EXT_FORMDEF	Specifies one or more suffixes to be appended to an AFP FormDef resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The default is no suffix. No extension is written as ;;. Example: EXT_FORMDEF=.600;.frm;.fd;; The above example causes AFP2IPDS backend to search for the following resource with name F1BASE: F1BASE.600 F1BASE.frm F1BASE.fd F1BASE

EXT_OBJECT_CONTAINER	Specifies one or more suffixes to be appended to an AFP ObjectContainer resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The extensions are free definable and separated by ";". The default is no suffix. No extension is written as ;;. The search order is the same as the extensions are written from left to right. The first matching resource will be used. Example: EXT_OBJECT_CONTAINER=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name H1BASE: H1BASE.600 H1BASE.300 H1BASE.240 H1BASE
EXT_OFF	0: The defined Resource File Name Extensions will be used. If the Extension is defined in <backend>.ini and in the job ticket, then the job ticket is overwriting the <backend>.ini definition. 1: Resource File Name Extensions will not be used. **Attention**: This parameter is typically set in a APA job ticket which overwrites this adjustment.
EXT_OUTLINE_FONT	Specifies one or more suffixes to be appended to an AFP OutlineFont resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The extensions are free definable and separated by ";". The default is no suffix. No extension is written as ;;. The search order is the same as the extensions are written from left to right. The first matching resource will be used. Example: EXT_OUTLINE_FONT=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name XZBASE: XZBASE.600 XZBASE.300 XZBASE.240 XZBASE

EXT_OVERLAY	Specifies one or more suffixes to be appended to an AFP Overlay resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The default is no suffix. No extension is written as ;;. Example: EXT_OVERLAY=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name O1BASE: O1BASE.600 O1BASE.300 O1BASE.240 O1BASE
EXT_PAGEDEF	Specifies one or more suffixes to be appended to an AFP PageDef resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The default is no suffix. No extension is written as ;;. Example: EXT_PAGEDEF=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name P1BASE: P1BASE.600 P1BASE.300 P1BASE.240 P1BASE
EXT_PAGESEG	Specifies one or more suffixes to be appended to an AFP PageSegment resource name while searching for the corresponding file. extension A variable length character string starting with a dot. The default is no suffix. No extension is written as ;;. Example: EXT_PAGESEG=.600;.300;.240;; The above example causes AFP2IPDS backend to search for the following resource with name S1BASE: S1BASE.600 S1BASE.300 S1BASE.240 S1BASE

LIB_COMBINE	Specifies the individual resource search paths. 0: The job ticket definitions will overwrite the <backend>.ini settings. 1: The parameters defined in <backend>.ini will be added to the parameters specified in the job ticket (search order: job ticket parameters are searched first). Search in the job ticket and <backend>.ini. **Attention**: This parameter is typically set in a APA job ticket which overwrites this adjustment.
LIB_COMMON	List of individual resource paths. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the resource specific paths. The first matching resource will be used. Example: LIB_COMMON: /u/prismapro/resources/system/240dpi; LIB_FORMDEF: /u/prismapro/resources/local/new/frm; /u/prismapro/resources/local/mylib/forms; /u/prismapro/resources/local/mylib/fdef Searching the formdef resource "f1test" would result in the following search order: /u/prismapro/resources/system/240dpi/f1test /u/prismapro/resources/local/new/frm/f1test /u/prismapro/resources/local/mylib/forms/f1test /u/prismapro/resources/local/mylib/fdef/f1test

LIB_FONT	List of individual resource paths of fonts. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the font specific paths. The first matching resource will be used. Example: LIB_COMMON: /u/prismapro/resources/system/240dpi; LIB_FONT: /u/prismapro/resources/local/mylib/font Searching the font resource "X0GT10Y" would result in the following search order: /u/prismapro/resources/system/240dpi/X0GT10Y /u/prismapro/resources/local/mylib/font/X0GT10Y
LIB_FORMDEF	List of individual resource paths of formdefs. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the formdef specific paths. The first matching resource will be used. Example: LIB_COMMON: /u/prismapro/resources/system/240dpi; LIB_FORMDEF:/u/prismapro/resources/local/new/frm; /u/prismapro/resources/local/mylib/forms; /u/prismapro/resources/local/mylib/fdef Searching the formdef resource "f1test" would result in the following search order: /u/prismapro/resources/system/240dpi/f1test /u/prismapro/resources/local/new/frm/f1test /u/prismapro/resources/local/mylib/forms/f1test /u/prismapro/resources/local/mylib/fdef/f1test

LIB_OBJECT_ CONTAINER	List of individual resource paths of object libraries containers. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the object libraries container specific paths. The first matching resource will be used. Example: see LIB_FONT
LIB_OVERLAY	List of individual resource paths of overlays. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the overlay specific paths. The first matching resource will be used. Example: LIB_COMMON: /u/prismapro/resources/system/240dpi; LIB_FORMDEF: /u/prismapro/resources/local/new/frm; /u/prismapro/resources/local/mylib/forms; /u/prismapro/resources/local/mylib/fdef LIB_PAGEDEF: /u/prismapro/resources/local/new/pdef; /u/prismapro/resources/local/mylib/pagedef LIB_OVERLAY: /u/prismapro/resources/local/mylib/overlay Searching the overlay resource "o1lese12" would result in the following search order: /u/prismapro/resources/system/240dpi/o1lese12 /u/prismapro/resources/local/mylib/overlay/o1lese12

LIB_PAGEDEF	List of individual resource paths of pagedefs. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the pagedef specific paths. The first matching resource will be used. Example: LIB_COMMON: /u/prismapro/resources/system/240dpi; LIB_FORMDEF: /u/prismapro/resources/local/new/frm; /u/prismapro/resources/local/mylib/forms; /u/prismapro/resources/local/mylib/fdef LIB_PAGEDEF: /u/prismapro/resources/local/new/pdef; /u/prismapro/resources/local/mylib/pagedef Searching the pagedef resource "p1test" would result in the following search order: /u/prismapro/resources/system/240dpi/p1test /u/prismapro/resources/local/new/pdef/p1test /u/prismapro/resources/local/mylib/pagedef/p1test
LIB_PAGESEG	List of individual resource paths of pagesegments. The paths are free definable and separated by ";". The search order is the same as the directories are written from left to right. First the resource is searched in the paths specified at "LIB_COMMON". Second in the pagesegment specific paths. The first matching resource will be used. Example: see LIB_FONT
MULTIPLE_RESOURCE_GROUPS	0 Multiple inlineresource groups will be ignored. 1 Multiple inlineresource groups are supported.

PDSCOLMAP	Color Map table name; Defines a default colormap resource (usually not used). The name can be 1 to 8 alphanumeric characters, including the two-character prefix, if there is one. Specifying DUMMY requires the print file to contain at least one inline Color Mapping table. Specifies the default COLORMAP name that will be used for this printer session when no colormap is specified. A hardware colormap reset is performed when this parameter is not specified.
PDSFOMAP	(fontMap) Specifies the default FONTMAP name that will be used for this printer session. fontMap specifies the name of a Font Mapping Object Container which is loaded from the Object Container library. The default, when this parameter is not specified is that all font mapping operations are disabled. A Font Mapping Table FMT maps a Global resource ID GRID to an external font resource name. The name can be 1 to 8 alphanumeric characters, including the two character prefix, if there is one. Specifying DUMMY requires the print file to contain at least one inline Font Mapping table.
RESIDENTFONTTABLE	RESIDENTFONTTABLE= string Filename of the ResidentFontTable for IP6400 or IP6500 printers RESIDENTFONTTABLE=/u/prismapro/cfg/printers/.system/afp.all/ResidentFontTable.txt (default)
SYS_EXT_OFF	0: The defined Resource File Name Extensions in the job ticket will be also used for the system resources. H I D D E N E.g. for the Formdef or Pagedef of the Infopage- Resources (Headerpage, Messagepage ....) 1: Resource File Name Extensions will not be used for the system resources. **Attention**: This parameter is only working by EXT_OFF=0 - EXT_OFF is typically set in a APA job ticket which overwrites the printparameter adjustment.

1.2.3.6 TRACE

TRACECOM-PRESS	TRACECOMPRESS specifies compression of Tracefile 0 trace file is not compressed (Default) 1 trace file is compress by AFP2IPDS and written in GZIP file format Note that trace file compression is a cpu intensive task and should only be activated in cases where huge trace files raise problems. Also note that TRACELIMIT is deactivated when TRACECOM-PRESS is activated.
TRACEFILE	Filename of the trace file. TRACEFILE=/u/prismapro/diag/printers/<Printer name>.trc The name must be unique for each configured printer (Printer name is the name of the logical printer).
TRACEFILE_LIMIT	Specifies maximum size of trace file in kBytes. The trace file is a "wrap around file". If set to 0 then the trace file will be unlimited. Suggested for TRACELEVEL=2, transfer of compressed file with email should be possible 10240 (10 MB) default file size the problem happens on startup 20480 (20 MB) the problem happens after some pages 51200 (50 MB) the problem happens sporadically pre-history may be interesting Suggested for TRACELEVEL=4, transfer of compressed file with ftp should be possible 102400 (100 MB) suggested for most of the problems 204800 (200 MB) suggested, if it's easy to transfer the data easy to view 512000 (500 MB) suggested, if it's easy to transfer the data viewing becomes a problem Suggested for 'FULL' traces, transfer of compressed file with CD or DVD may be necessary 1000000 (ca 1.0 GB) capture a problem with all data and a long prehistory 1900000 (ca 1.9 GB) maximum file size allowed

TRACEFLAGS	Specifies the internal Trace Set to be used. The internal Trace Set is a 64-bit field specifying the components to be traced. Up to 8 pairs of 2 hex characters may be entered. Unspecified values default to x'FF'. Use this option when a problem is being diagnosed, following the directions of your Océ Service Representative. **NOTE**: Tracing inflicts performance! All combinations are allowed. TRC_NONE will be overruled by any other flag. Enumeration values: TF_NONE 0x00000000 Tracing disabled. TF_ALL 0xFFFFFFFF Tracing enabled completely. TF_ITEM_MASK 0x00FFFFFF Masks trace items (layers, etc.) TF_CTRL_MASK 0xFF000000 Masks control flags (dump, X-flags, etc.) TF_TRACE 0x00000001 Trace tracing enabled TF_INPUT 0x00000002 Input tracing enabled, contains all Stream classes TF_IPQ 0x00000004 IPQ tracing enabled TF_AFPOBJPROC 0x00000008 AFP-Object-Processor tracing anabled TF_IPDSTRACER 0x00000010 IPDSTRACER tracing enabled, contains: IPDSCmdDGB IPDSCmdGenerator IPDSCmdLCC IPDSSeqGenerator IPDSTracer TF_DISPATCHING 0x00000020 Layer4-Dispatcher trace enabled TF_PAL 0x00000040 Printer Access Library trace enabled PrinterInterface TF_IPDSDUMMY 0x00000080 DummyPrinter tracing enabled TF_SPOOL 0x00000100 Following Classes will be traced:PRISMA Spool IP Spool SpoolInterface

	TF_OPCSTM	0x00000200	Following Data will be traced: OPC and STM (switch also 0x01 on) Following Functionality will be traced: Binselection OperatorMessage MountForm PrinterSetup ColorSetup
	TF_PRTCMD	0x00000400	PrtCmd and PrtCmdGroup tracing endabled: FinishingControl MediaType MediumDescriptor ModificationControl PresentationEnvironment
	TF_LINEDATA	0x00000800	LineDataFormatter tracing enabled
	TF_MESSAGER	0x00001000	Messager and EnvironmentMessages tracing enabled
	TF_RESOURCE	0x00002000	Resource tracing enabled
	TF_ACCOUNTING	0x00004000	Accounting trace enabled Recordformatting
	TF_XML	0x00008000	XML tracing enabled
	TF_AFPCONTROL	0x00010000	AFPControl tracing enabled CheckpointHandler and all CheckPoint from Layer 2 RestartContextHandler InfoPageProcessor
	TF_BASICCONTROL	0x00020000	BasicControl tracing enabled Interval Timer tracing enabled
	TF_FLOWCONTROL	0x00040000	FlowControl tracing enabled
	TF_INPUTCONTROL	0x00080000	InputControl tracing enabled, contains: InputControl Job JobGenerator
	TF_MAIN	0x00100000	Main tracing enabled
	TF_OUTPUTCONTROL	0x00200000	OutputControl tracing enabled

	TF_PAGECONTROL	0x00400000	PageControl tracing enabled
	TF_DUMPIPDS	0x08000000	Enable tracing of ipds sequences
	Control flags:		
	TF_DUMP	0x01000000	Enable tracing of data (dump)
	TF_DUMPRES	0x02000000	Enable tracing of resource data
	TF_DUMPINPUT	0x04000000	Enable tracing of input data
	Special, undocumented control flags:		
	TF_X_INVALID	0xF0000000	All X-flags are invalid
	TF_X_VS6_IDE	0x10000000	Enable tracing to Visual Studio
	TF_X_CERR	0x20000000	Enable tracing to cerr
	TF_X_COUT	0x40000000	Enable tracing to cout
	TF_X_NO_FILEOUT	0x80000000	Supress tracing to file

1.2.4 Workflow: Generation of InfoPages

This document describes how InfoPages can be used under PRISMAproduction Server. As example a header page will be used.

If all kinds of InfoPages have the same parameter, the parameter name is written as XXX_<parametername>.

An InfoPage consists of:

- template
- formdef
- pagedef
- parameters

1.2.4.1 Template

To use InfoPages in PRISMAproduction Server a template has to be built. A template can consist of:

- free text
- keywords framed with $-signs
- Carriage-Control-Characters
- TRC-Characters
- BigLetter escape sequence

Free Text

A user defined text.

Keywords: Parameter-Keywords

Each parameter which comes with the backend.ini-, job- or session-parameter file can be used as keyword, by embracing the parameter name with $-signs.

Example:

Entry In the job-parameter file:
`JOBNAME=0010kb_1000pg_test`

To get the same line on an InfoPage you have to write:
`JOBNAME=$JOBNAME$`

Keywords: Accounting-Keywords

Accounting-Keywords are used to bring accounting information on an InfoPage, usually for the TrailerPage.

Also the old keywords from SPS4 can be used on the InfoPages.

For using an accounting-keyword the keyword has to be embraced with $-characters.

Example:

To get the amount of generated Pages you have to write:
`Generated Pages=$ACCT_DATA_PAGES_GENERATED$`

Following Accounting-Keywords are available:

Page/sheet accounting INPUT

ACCT_DATA_PAGES_GENERATED
ACCT_DATA_SHEETS_GENERATED
ACCT_INFO_PAGES_GENERATED
ACCT_INFO_SHEETS_GENERATED

Resource accounting - INPUT

ACCT_NUM_FORMDEFS
ACCT_NUM_PAGEDEFS
ACCT_NUM_REQUESTED_FONTS
ACCT_NUM_REQUESTED_OVERLAYS
ACCT_NUM_REQUESTED_PAGESEGS

Page/sheet accounting - OUTPUT

ACCT_DATA_PAGES_STACKED
ACCT_DATA_SHEETS_STACKED
ACCT_INFO_PAGES_STACKED
ACCT_INFO_SHEETS_STACKED

Resource accounting - OUTPUT

ACCT_NUM_LOADED_FONTS
ACCT_NUM_LOADED_OVERLAYS
ACCT_NUM_LOADED_PAGESEGS

General accounting

ACCT_DUPLEX
ACCT_INPUT_BIN_LIST
ACCT_OUTPUT_BIN
ACCT_XMITS

General information

DATE
MSG_LIST
PAGENAME
PAGESTATUS
TIME
USERMESSAGE
VERSION
VERSIONDATE
VERSIONTIME
OPERATOR_NAME
OPERATOR_LOG_IN

Job monitoring

CURRENT_COPY
STEPS
CURRENT_STEP
CURRENT_FILECOPY

Carriage-Control-Characters

Only ASA carriage control characters are supported.

Allowed ASCII values are:

ascii value	hex value
Blank	0x20
0 – 9	0x30 - 0x39
A	0x41
B	0x42
C	0x43
/	0x2F
+	0x2B

If CC-Characters are available the user has to set the XXX_CC parameter to 1.

TRC-Characters

Allowed ASCII values are:

ascii value	hex value
Blank	0x20
0 – 9	0x30 - 0x39

If more than 10 TRCs are requested, the user has to define the requested fonts by his PAGEDEF.

If TRC-Characters are available the user has to set the XXX_TRC parameter to 1.

BigLetter Escape Sequence

The user can choose between a normal character size and a big letter character size. To switch between these sizes the user has to use the following escape sequence:
`\b`

Text which should be printed in big letters must be framed by \b.

Example:

Entry in the job-parameter file:
`JOBNAME=pg_test`

To get the job name in big letters on an InfoPage you have to write:
`\b$JOBNAME$\b`

This will create the following text:

```
PPPPPPPP    GGGGGGG              TTTTTTTT   EEEEEEEEE   SSSSSSS   TTTTTTTT
PPPPPPPPP  GGGGGGGGG             TTTTTTTT   EEEEEEEEE  SSSSSSSSS  TTTTTTTT
PP      PP GG     GG                TT      EE         SS      SS     TT
PP      PP GG                       TT      EE         SS             TT
PPPPPPPPP  GG  GGGGG                TT      EEEEEE     SSSSSSSS       TT
PPPPPPPP   GG  GGGGG                TT      EEEEEE      SSSSSSSS      TT
PP         GG     GG                TT      EE                SS     TT
PP         GG     GG                TT      EE         SS     SS     TT
PP         GGGGGGGGG  _________      TT      EEEEEEEEE  SSSSSSSSS     TT
PP          GGGGGGG   _________      TT      EEEEEEEEE   SSSSSSS      TT
```

Note: Characters are converted to uppercase when using big letters.

1.2.4.2 Formdef

For printing the InfoPage a formdef is necessary. With PRISMAproduction Server a formdef for the standard InfoPages is delivered: F1INFOPG

Following parameters are set:

parameter	value
orientation	portrait
one up	one up
OffsetStack/EdgeMarkChange	no
MediaSourceID	0

DuplexControl	simplex
ConstantFormsControl	inactive

1.2.4.3 Pagedef

For printing the InfoPage a pagedef is necessary. With PRISMAproduction Server a pagedef for the standard InfoPages is delivered: P1STD3

Following parameters are set:

parameter	value
size	A4
units	300dpi
font	X000203Y
LND's	59

1.2.4.4 Parameter

Following parameters are available for all InfoPages, their names consist of <InfoPageType>_<parametername>, here we only mention the parameter name, as they are valid for all InfoPageTypes:

BACKSIDEOVLS
CC
COLMAP
DCK
DUPLEX
ENABLING
FLG2
FONTS
FORMDEF
FRONTSIDEOVLS
INPUT_BIN
OFFXB
OFFXF
OFFYB
OFFYF
OUTPUT_BIN
PAGEDEF
PRINT_DIRECTION
SOSI
TEMPLATE
TRC
X2UP

BackSideOvls

List of overlays to be placed on the reverse side of each sheet. List delimiter is ';'

CC

Only ASA carriage control characters are supported.

Allowed ASCII values are:

ascii value	hex value
Blank	0x20
0 – 9	0x30 - 0x39
A	0x41
B	0x42
C	0x43
/	0x2F
+	0x2B

ColMap

Defines a colormap resource.

DCK

Bitmask to disable data checks:

value	description
0x00	All data checks are reported by the printer.
0x01	Printer ignores printing outside of the printable area.
0x02	Printer ignores character codes that are undefined in the code page of the font and prints the default character instead.
0x03	Is a combination of 0x01 and 0x02

ATTENTION: This parameter can't be changed, because no data- and/or position-checks may appear at a header page!

DUPLEX

Specifies default Simplex/duplex information:

value	description
	No value means, use information from FORMDEF
0	Simplex
1	Normal Duplex

2	Tumble Duplex

Enabling

Defines printing of InfoPages. The parameter setting depends on the type of InfoPage. Following types are available:

- Group InfoPages

  GroupHeaderPage
  GroupSeparatorPage
  GroupMessagePage
  GroupTrailerPage

- File InfoPages

  HeaderPage
  SeparatorPage
  MessagePage
  TrailerPage

Group InfoPages

Following settings for enabling are possible:

value	description
0	Suppress Group InfoPage
1	Enable Group InfoPage

File InfoPages

- HeaderPage and SeparatorPage

Following settings for enabling are possible.

value	description
0	Suppress Header- or Separator-Page
1	Before first file copy
2	Before all file copies
4	At repositioning
8	Mark form before first file copy **Attention**: Mark Form instructs the printer to print black bars on the fold, so the operator can better recognize where an InfoPage was printed. In the printer control panel the user can configure what number of marked pages should be printed. Is this value > 1, so Mark Form leads to copies of the InfoPages. If a printer doesn't support this function, the setting of this value will be ignored.

All these values may be combined.

Example: HDRP_ENABLING=7

HeaderPage will be printed:

1 Before first file copy

+2 Before all file copies

+4 At repositioning

- MessagePage

Following settings for enabling are possible.

value	description
0	Suppress MessagePage
1	After last file copy
2	After all file copies
4	At repositioning
8	Mark form after last file copy **Attention**: Mark Form instructs the printer to print black bars on the fold, so the operator can better recognize where an InfoPage was printed. In the printer control panel the user can configure what number of marked pages should be printed. Is this value > 1, so Mark Form leads to copies of the InfoPages. If a printer doesn't support this function, the setting of this value will be ignored.
16	NPRO before MessagePage

All these values may be combined.

Example: MSGP_ENABLING=7

MessagePage will be printed:

1 After first file copy

+2 After all file copies

+4 At repositioning

- TrailerPage

Following settings for enabling are possible.

value	description
0	Suppress MessagePage
1	After last file copy
2	After all file copies

4	At repositioning
8	Mark form after last file copy **Attention**: Mark Form instructs the printer to print black bars on the fold, so the operator can better recognize where an InfoPage was printed. In the printer control panel the user can configure what number of marked pages should be printed. Is this value > 1, so Mark Form leads to copies of the InfoPages. If a printer doesn't support this function, the setting of this value will be ignored.

All these values may be combined.

Example: TRLP_ENABLING=7

MessagePage will be printed:

1 After first file copy

+2 After all file copies

+4 At repositioning

Enabling Example

To understand the insertion of InfoPages, following definitions have to be clarified:

Job
Section

A Job consists of Job-Parameters and one or more sections, in the upper description they are called files.

- Job with one Section, no Section copies

  All GroupInfoPages are enabled.

  All FileInfoPages are enabled with value 1 (Before first file copy / After last file copy).

The following output will be generated:

```
Group Header Page
Group Separator Page
         Header Page
         Separator Page
                  Print Data
         <Message Page>
      Trailer Page
<Group Message Page>
Group Trailer Page
```

- Job with one Section, 3 Section copies

  All GroupInfoPages are enabled.

  All FileInfoPages are enabled with value 1 (Before first file copy / After last file copy).

The following output will be generated:

```
Group Header Page
Group Separator Page
      Header Page
      Separator Page
            Print Data copy 1
            Print Data copy 2
            Print Data copy 3
      <Message Page>
      Trailer Page
<Group Message Page>
Group Trailer Page
```

All FileInfoPages are enabled with value 2 (Before all file copies / After all file copies).

The following output will be generated:

```
Group Header Page
Group Separator Page
      Header Page
      Separator Page
            Print Data copy 1
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data copy 2
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data copy 3
      <Message Page>
      Trailer Page
<Group Message Page>
Group Trailer Page
```

- Job with 3 Sections, no copies

All GroupInfoPages are enabled.

All FileInfoPages are enabled with value 1 (Before first file copy / After last file copy).

The following output will be generated:

```
Group Header Page
Group Separator Page
      Header Page
      Separator Page
            Print Data section 1
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 2
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 3
      <Message Page>
      Trailer Page
<Group Message Page>
Group Trailer Page
```

- Job with 3 Sections, first Section 2 copies, second Section 3 copies, third Section 2 copies

All FileInfoPages are enabled with value 1 (Before first file copy / After last file copy).

The following output will be generated:

```
Group Header Page
Group Separator Page
      Header Page
      Separator Page
            Print Data section 1 copy 1
            Print Data section 1 copy 2
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 2 copy 1
```

```
            Print Data section 2 copy 2
            Print Data section 2 copy 3
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 3 copy 1
            Print Data section 3 copy 2
      <Message Page>
      Trailer Page
<Group Message Page>
Group Trailer Page
```

All FileInfoPages are enabled with value 2 (Before all file copies / After all file copies).

The following output will be generated:

```
Group Header Page
Group Separator Page
      Header Page
      Separator Page
            Print Data section 1 copy 1
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 1 copy 2
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 2 copy 1
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 2 copy 2
      <Message Page>
      Trailer Page
      Header Page
      Separator Page
            Print Data section 2 copy 3
      <Message Page>
      Trailer Page
```

```
        Header Page
        Separator Page
              Print Data section 3 copy 1
        <Message Page>
        Trailer Page
        Header Page
        Separator Page
              Print Data section 3 copy 2
        <Message Page>
        Trailer Page
  <Group Message Page>
  Group Trailer Page
```

FLG2

Bitmask for following options:

Value	Description
0x02	Burst stacking request
0x04	EOT mark/Offset stacking request at Begin of 1. copy of file
0x20	EOT mark/Offset stacking request at Begin of each copy
0x40	Swapped printing request (A-Twin printers only, front and back pages will be swapped).

All these values may be combined.

Fonts

The content of this parameter will be used, if TRC=ON and a PageDef with no fonts inside is used. Up to 10 font names can be given separated by semicolon. For further information see chapter 7.3

FormDef

Defines the Formdef for the InfoPage, the default is F1INFOPG.

FrontSideOvls

List of overlays to be placed on the front side of each sheet. List delimiter is the semicolon (';')

Input_Bin

Specifies default Input Media source code (for cutsheet printers only):

value	description
	No value means, use information from FORMDEF
0	Input Bin A
1	Input Bin B
2	Input Bin C
3	Input Bin D

OFFXB

X-offset on the reverse side of the sheet:

Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unit base], the unit base is given by the printer HW. See also Section PRINT_PROCESSING parameter PRTUPUB.

OFFXF

X-offset on front side of the sheet:

Defines a new x-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unit base], the unit base is given by the printer HW. See also Section PRINT_PROCESSING parameter PRTUPUB.

OFFYB

Y-offset on the reverse side of the sheet:

Defines a new y-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 l-units/unit base], the unit base is given by the printer HW. See also Section PRINT_PROCESSING parameter PRTUPUB.

OFFYF

Y-offset on front side of the sheet:

Defines a new y-offset of the logical page position with respect to the origin of the medium presentation space, or when N-up is selected with respect to one of the N-up partition origins. The number is in [14400 units/unit base], the unit base is given by the printer HW. See also Section PRINT_PROCESSING parameter PRTUPUB.

Output_Bin

Specifies the Output Media destination (for cutsheet printers only):

value	description
0	use the default output tray of the printer
1	Output tray A is selected
2	Output tray B is selected

Note: In case the value is not zero, remote tray selection must be enabled at the operator panel of the printer as well!

PageDef

Defines the Pagedef for the InfoPage, the default is P1INFOPG.

Print_Direction

Information about the print direction:

value	description
1	the orientation of the FORMDEF is used unchanged.
2	the orientation is forced to be PORTRAIT
3	the orientation is forced to be LANDSCAPE.
4	the orientation is forced to be PORTRAIT 90.
5	the orientation is forced to be LANDSCAPE 90.
6	the orientation is forced to be PORTRAIT 180.
7	the orientation is forced to be LANDSCAPE 180.
8	don't use orientation from FORMDEF, orientation is forced to be Portrait (only cutsheet printers).
9	use orientation from Formdef (only cutsheet printers).

Note: If 8 or 9 is used, the setting of the MDDflg (by PPFA or SLE) is ignored. If N-up presentation is active, the orientation of the FORMDEF will always be passed to the cutsheet printer.

SOSI

Defines double byte mode:

value	description
0	Shift-out/shift-in codes are ignored.
1	Shift-out/shift-in codes are recognized and replaced with spaces.
2	Shift-out/shift-in codes are recognized

Template

Filename of the header page template file. PRISMAproduction Server delivers following templates:

- hdrpage
- infopage
- msgpage
- trlpage

The customer can define his own templates, which must be placed for example in /u/prismapro/cfg/printers/<printername> (please see step 5 in 'Custom specific infopages and resources for configured printers' on page 12).

TRC

Specifies if table reference characters are availabe (1) or not (0).

Allowed ASCII values are:

ascii value	hex value
Blank	0x20
0 – 9	0x30 - 0x39

If more than 10 TRCs are requested, the user has to define the requested fonts by his PAGEDEF.

X2UP

Specifies default Two-up processing:

value	description
	No value means, use information from FORMDEF
0	No Two-up
1	Normal Two-up (left/right)
2	Identical copies
3	Inversed Two-up (right/left)

1.2.4.5 Tracing

To see which Keywords were not found, following TraceFlag must be set: 0x00010000

1.2.4.6 Example

These examples show how to use the InfoPage-Parameters. We will use the standard hdrpage template and then change it step by step to demonstrate how the parameters work.

Standard HeaderPage

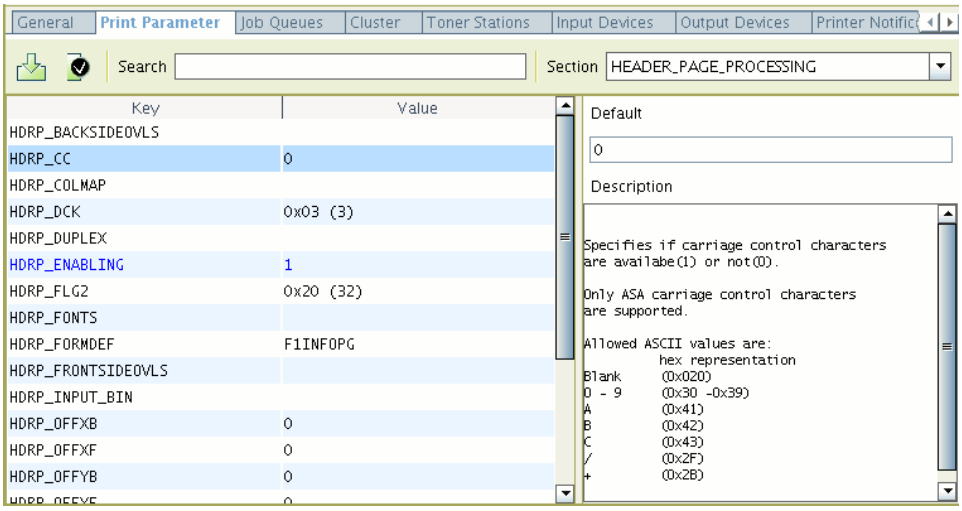
Template:

```
**********************************************************************
$TITLE$
OPS PRISMA SNIPDS Backend Version $VERSION$ created on $VERSIONDATE$
$VERSIONTIME$
$PAGENAME$
**********************************************************************
USER    NAME:     $OWNER$
JOB     NAME:     $JOBNAME$
FILE    NAME:     $FILENAME$
HOST    NAME:     $CLIENT$
SPOOL   ID:       $JOBID$
SPOOL   TIME:     $SPOOLDATE$   $SPOOLTIME$
PRINT   TIME:     $DATE$    $TIME$
PRINTER ID:       $PRINTER$
STATUS:           $PAGESTATUS$    (START NEW_XMIT OR CONT)
**********************************************************************
NAME:             $REC_NAME$ $DEPARTMENT$
ROOM:             $ROOM$
BUILDING:         $BUILDING$
ADDRESS:          $ADDRESS1$ $ADDRESS2$ $ADDRESS3$ $ADDRESS4$
**********************************************************************
$OWNER$
**********************************************************************
$USERMESSAGE$
```

Printout:

```
**********************************************************************************


OPS PRISMA SNIPDS Backend Version 5.00.019xx3 created on 2008-02-07 14:00:00
Header Page

**********************************************************************************

USER  NAME:     root
JOB   NAME:     sps5u-AFP_001_SIM_E00_OK.0001
FILE  NAME:     /spscis/pool/data/SLEIB.SPS5/DAT1SPS5
HOST  NAME:     1x011530vm3.ops.oce.net
SPOOL   ID:     00000140
SPOOL TIME:     07.02.2008  12:23:18
PRINT TIME:     07.02.2008  09:18:35
PRINTER ID:     vs_9250§1x011530vm3.ops.oce.net
STATUS:         START       (START NEW_XMIT OR CONT)

**********************************************************************************

NAME:           SWD11
ROOM:           14.342
BUILDING:       14
ADDRESS:        Siemensallee 2, 85586 Poing

**********************************************************************************

root

**********************************************************************************
```

Parameter Settings:



HeaderPage with CC

Now the same HeaderPage Output should be printed, but with a template without empty lines and with Carriage-Control (CC) Characters.

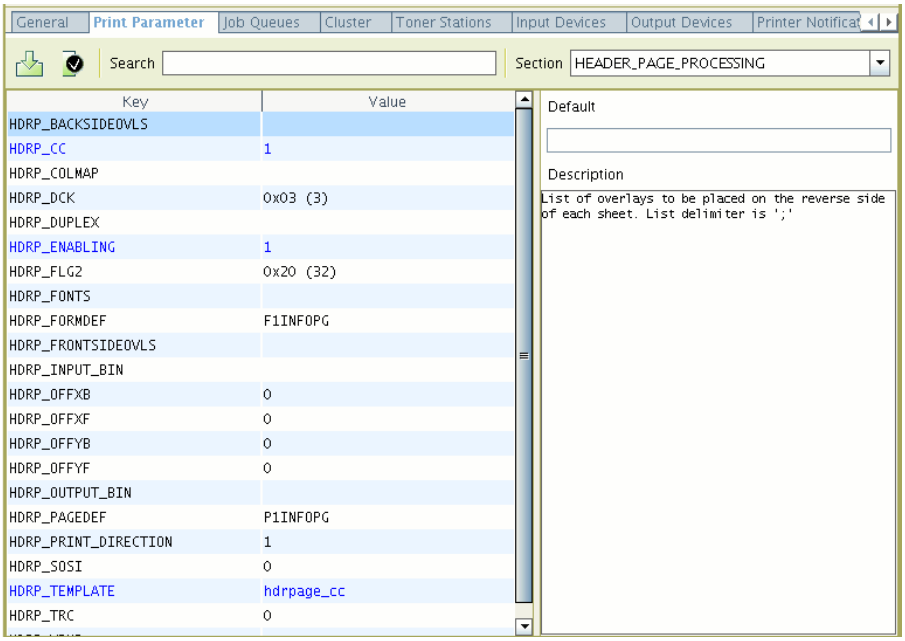
Template:

```
**********************************************************************
$TITLE$
OPS PRISMA SNIPDS Backend Version $VERSION$ created on $VERSIONDATE$
$VERSIONTIME$
$PAGENAME$
**********************************************************************
USER  NAME:      $OWNER$
JOB   NAME:      $JOBNAME$
FILE  NAME:      $FILENAME$
HOST  NAME:      $CLIENT$
SPOOL   ID:      $JOBID$
SPOOL TIME:      $SPOOLDATE$  $SPOOLTIME$
PRINT TIME:      $DATE$  $TIME$
PRINTER ID:      $PRINTER$
STATUS:          $PAGESTATUS$       (START NEW_XMIT OR CONT)
**********************************************************************
NAME:            $REC_NAME$ $DEPARTMENT$
ROOM:            $ROOM$
BUILDING:        $BUILDING$
ADDRESS:         $ADDRESS1$ $ADDRESS2$ $ADDRESS3$ $ADDRESS4$
**********************************************************************
$OWNER$
**********************************************************************
$USERMESSAGE$
```

Printout:

```
**********************************************************************************


OPS PRISMA SNIPDS Backend Version 5.00.019 created on 2008-02-07 10:30:00
Header Page

**********************************************************************************

USER  NAME:      root
JOB   NAME:      sps5u-AFP_001_SIM_E00_OK.0001
FILE  NAME:      /spscis/pool/data/SLEIB.SPS5/DAT1SPS5
HOST  NAME:      1x011530vm3.ops.oce.net
SPOOL   ID:      00000177
SPOOL TIME:      07.02.2008  13:25:11
PRINT TIME:      07.02.2008  13:30:02
PRINTER ID:      vs_9250§1x011530vm3.ops.oce.net
STATUS:          START       (START NEW_XMIT OR CONT)

**********************************************************************************

NAME:            SWD11
ROOM:            14.342
BUILDING:        14
ADDRESS:         Siemensallee 2, 85586 Poing

**********************************************************************************

root

**********************************************************************************
```

Parameter Settings:



HeaderPage with CC and TRC

There are two ways to use TRC's for InfoPages:

- PAGEDEF
- XXXX_FONTS

Using the PAGEDEF we need a PAGEDEF, which defines the fonts to use with the InfoPage. As an example the PAGEDEF with name P1CC_TRC can be used.

ATTENTION: All used fonts must be copied to the following directory: /u/prismapro/cfg/printers/<printername> (for more details please see step 5 in 'Custom specific infopages and resources for configured printers' on page 12).

Following fonts are available and can be assigned by TRC:

Font	TRC-Value
X1GB10	0
X1GB12	1
X1GI12	2
X1GT10	3
X1GT12	4
X1GT13	5
X1GT15	6
X1GT18	7

The second way is to use the parameter XXXX_FONTS. All fonts given in this parameter must be available in the directory `/u/prismapro/cfg/printers/<printername>`. They will be used in the order they are given in the parameter.

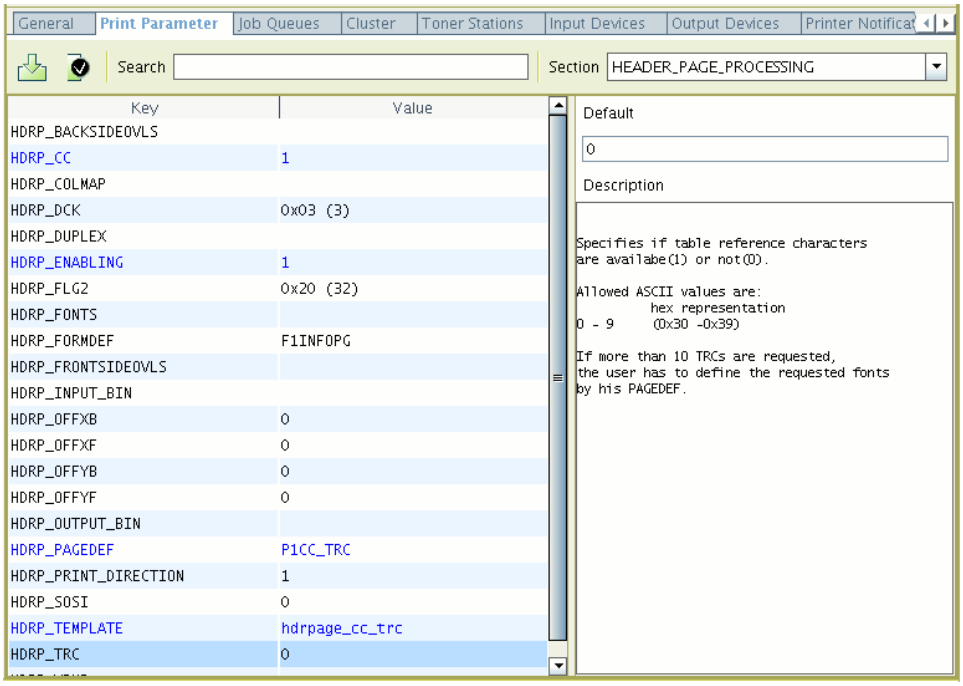
Template:

```
**********************************************************************
$TITLE$
OPS PRISMA SNIPDS Backend Version $VERSION$ created on $VERSIONDATE$
$VERSIONTIME$
$PAGENAME$
**********************************************************************
USER  NAME:     $OWNER$
JOB   NAME:     $JOBNAME$
FILE  NAME:     $FILENAME$
HOST  NAME:     $CLIENT$
SPOOL   ID:     $JOBID$
SPOOL TIME:     $SPOOLDATE$  $SPOOLTIME$
PRINT TIME:     $DATE$  $TIME$
PRINTER ID:     $PRINTER$
STATUS:         $PAGESTATUS$       (START NEW_XMIT OR CONT)
**********************************************************************
NAME:           $REC_NAME$ $DEPARTMENT$
ROOM:           $ROOM$
BUILDING:       $BUILDING$
ADDRESS:        $ADDRESS1$ $ADDRESS2$ $ADDRESS3$ $ADDRESS4$
**********************************************************************
$OWNER$
**********************************************************************
$USERMESSAGE$
```

Printout:

```
********************************************************************************


OPS PRISMA SNIPDS Backend Version 5.00.019 created on 2008-02-07 10:30:00
Header Page

********************************************************************************

USER  NAME:     root
JOB   NAME:     sps5u-AFP_001_SIM_E00_OK.0001
FILE  NAME:     /spscis/pool/data/SLEIB.SPS5/DAT1SPS5
HOST  NAME:     lx011530vm3.ops.oce.net
SPOOL   ID:     00000177
SPOOL TIME:     07.02.2008  13:25:11
PRINT TIME:     07.02.2008  14:01:27
PRINTER ID:     vs_9250@lx011530vm3.ops.oce.net
STATUS:         START       (START NEW_XMIT OR CONT)

********************************************************************************

NAME:           SWD11
ROOM:           14.342
BUILDING:       14
ADDRESS:        Siemensallee 2, 85586 Poing

********************************************************************************

root

********************************************************************************
```

Parameter Settings



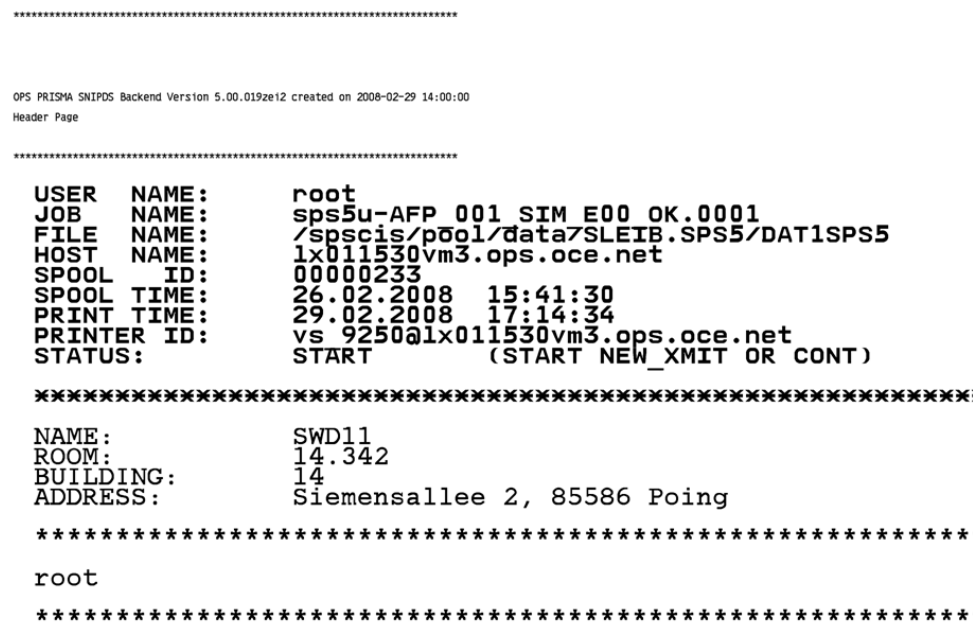
HeaderPage with CC using different TRC's

Template:

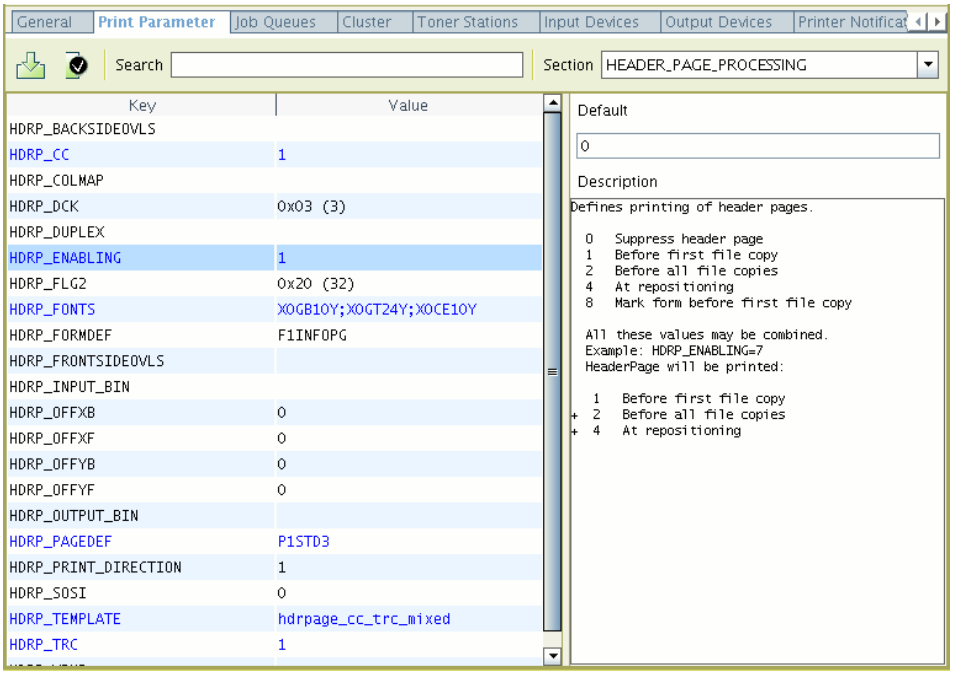
```
************************************************************************

$TITLE$

OPS PRISMA SNIPDS Backend Version $VERSION$ created on $VERSIONDATE$
$VERSIONTIME$

$PAGENAME$
************************************************************************

USER  NAME:      $OWNER$

JOB   NAME:      $JOBNAME$

FILE  NAME:      $FILENAME$

HOST  NAME:      $CLIENT$

SPOOL   ID:      $JOBID$

SPOOL TIME:      $SPOOLDATE$  $SPOOLTIME$

PRINT TIME:      $DATE$  $TIME$
```

```
PRINTER ID:      $PRINTER$
STATUS:          $PAGESTATUS$       (START NEW_XMIT OR CONT)
*****************************************************************
NAME:            $REC_NAME$ $DEPARTMENT$
ROOM:            $ROOM$
BUILDING:        $BUILDING$
ADDRESS:         $ADDRESS1$ $ADDRESS2$ $ADDRESS3$ $ADDRESS4$
*****************************************************************
$OWNER$
*****************************************************************
```

Printout:



Parameter Settings:



HeaderPage with FrontSideOverlay

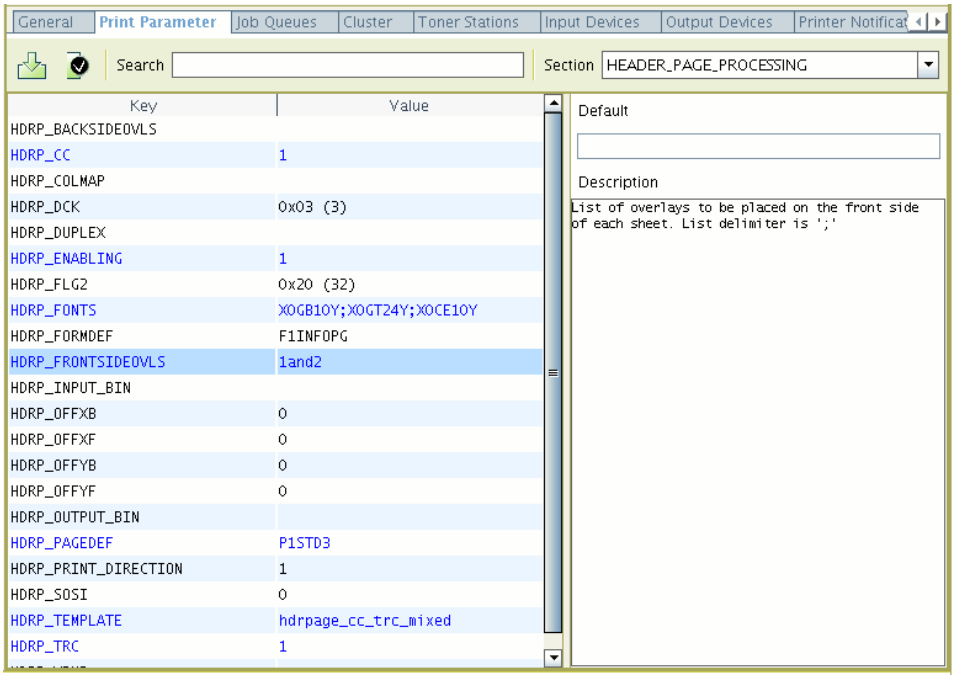
Template:

Is the same as in 'HeaderPage with CC using different TRC's'

Printout:

```
**********************************************************************

OPS PRISMA SNIPDS Backend Version 5.00.010rel2 created on 2008-02-29 14:00:00
Header Page

**********************************************************************

 USER   NAME:      root
 JOB    NAME:      sps5u-AFP_001_SIM_E00_OK.0001
 FILE   NAME:      /spscis/pool/data7SLEIB.SPS5/DAT1SPS5
 HOST   NAME:      lx011530vm3.ops.oce.net
 SPOOL    ID:      00000233
 SPOOL TIME:       26.02.2008  15:41:30
 PRINT TIME:       29.02.2008  17:31:11
 PRINTER ID:       vs_9250@lx011530vm3.ops.oce.net
 STATUS:           START       (START NEW_XMIT OR CONT)

**********************************************************************

NAME:              SWD11
ROOM:              14.342
BUILDING:          14
ADDRESS:           Siemensallee 2, 85586 Poing

**********************************************************************

root

**********************************************************************
```

Parameter Settings:



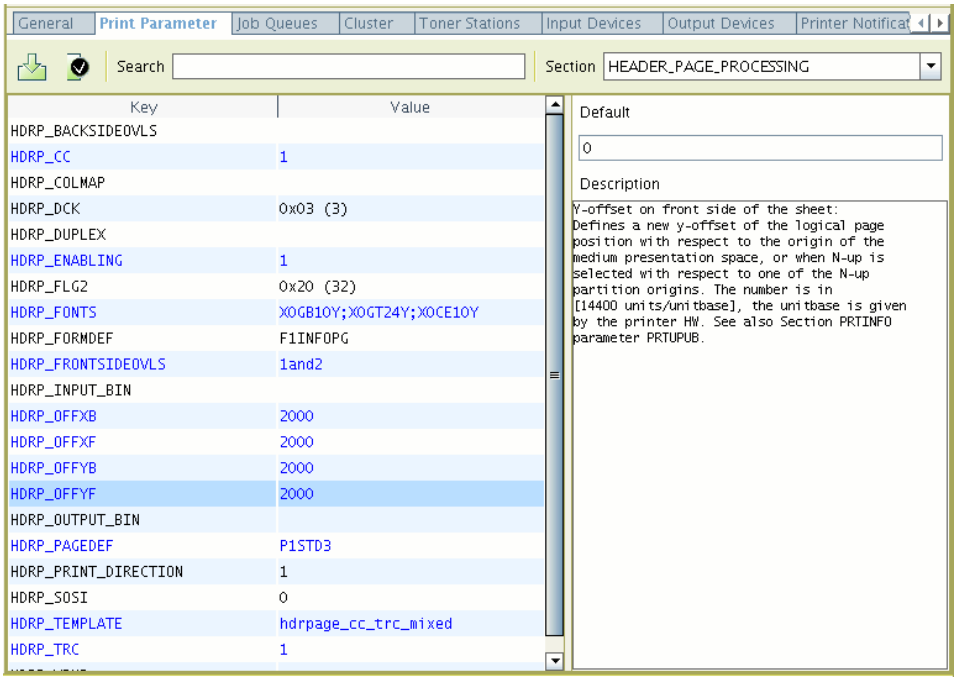
HeaderPage with OffSets

Template: The same as in ’HeaderPage with CC using different TRC's’

Printout:

```
**********

OPS PRISMA SNIPDS Backend Version 5.00.010e12 created on 2008-02-29 14:00:00
Header Page

**********

USER  NAME:      root
JOB   NAME:      sps5u-AFP_001_SIM_E00_OK.0001
FILE  NAME:      /spscis/pool/data7SLEI8.SPS5/DAT1SPS5
HOST  NAME:      1x011530vm3.ops.oce.net
SPOOL    ID:     00000233
SPOOL TIME:      26.02.2008  15:41:30
PRINT TIME:      29.02.2008  17:47:00
PRINTER ID:      vs_9250@1x011530vm3.ops.oce.net
STATUS:          START       (START NEW_XMIT OR CONT)

**********************************************************************************

NAME:            SWD11
ROOM:            14.342
BUILDING:        14
ADDRESS:         Siemensallee 2, 85586 Poing

**********************************************************************************

root

**********************************************************************************
```

Parameter Settings:



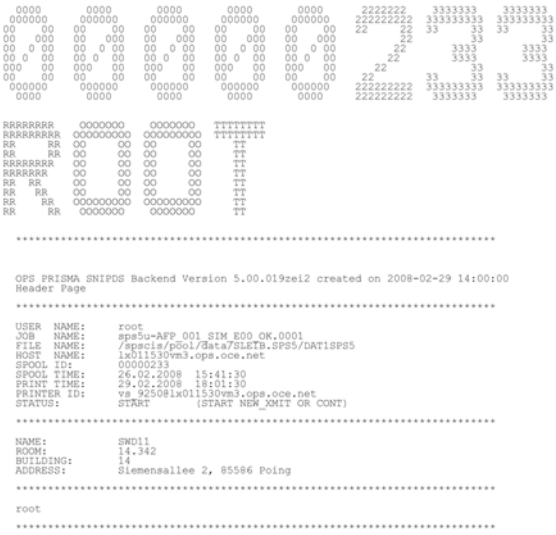
HeaderPage with Big Letters

Template

```
\b  $JOBID$\b
\b  $OWNER$\b
          ******************************************************

       $TITLE$

       OPS PRISMA SNIPDS Backend Version $VERSION$ created on
        $VERSIONDATE$ $VERSIONTIME$ $PAGENAME$

       ******************************************************

          USER  NAME:      $OWNER$

          JOB   NAME:      $JOBNAME$

          FILE  NAME:      $FILENAME$

          HOST  NAME:      $CLIENT$

          SPOOL ID:        $JOBID$

          SPOOL TIME:      $SPOOLDATE$  $SPOOLTIME$
```

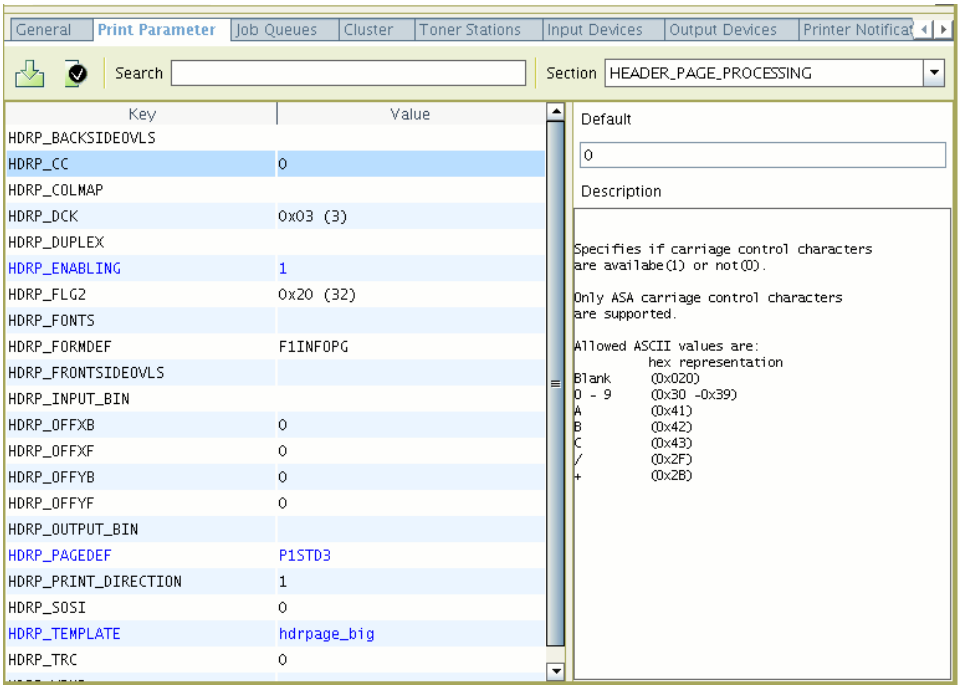
```
                PRINT TIME:      $DATE$  $TIME$
                PRINTER ID:      $PRINTER$
                STATUS:          $PAGESTATUS$ (START NEW_XMIT OR CONT)

*****************************************************************
                NAME:            $REC_NAME$ $DEPARTMENT$
                ROOM:            $ROOM$
                BUILDING:        $BUILDING$
                ADDRESS:         $ADDRESS1$ $ADDRESS2$ $ADDRESS3$
                $ADDRESS4$
                ***************************************************
                $OWNER$
                ***************************************************
                $USERMESSAGE$
```

Printout:



```
*****************************************************************

OPS PRISMA SNIPDS Backend Version 5.00.019zei2 created on 2008-02-29 14:00:00
Header Page
*****************************************************************

USER  NAME:      root
JOB   NAME:      sps5u-AFP_001_SIM_E00_OK.0001
FILE  NAME:      /spscis/pGol/data7SLEIB.SPS5/DAT1SPS5
HOST  NAME:      lx011530vm3.ops.oce.net
SPOOL ID:        00000233
SPOOL TIME:      26.02.2008  15:41:30
PRINT TIME:      29.02.2008  18:01:30
PRINTER ID:      vs_92508lx011530vm3.ops.oce.net
STATUS:          START       (START NEW_XMIT OR CONT)
*****************************************************************

NAME:            SWD11
ROOM:            14.342
BUILDING:        14
ADDRESS:         Siemensallee 2, 85586 Poing
*****************************************************************

root
*****************************************************************
```

Parameter Settings:



1.2.4.7 Differences between SPS4 and AFP2IPDS_BE

Trailer page Accounting

In AFP2IPDS_BE the resource-counters contain only resources which are requested and/or loaded for the print file. In SPS4 the requested and/or loaded resources for the InfoPages were counted also.

1.3 Functions of the Océ PostScript Backends

(for Océ VP21xx, Océ31xx, Océ61xx, Océ VP62xx, Océ CPS700-900, Xerox 6180, Canon 150)

1.3.1 Purpose of the PostScript Backends

- Printing of a given PostScript file.
- Conversion of a POD file into PostScript. Number or media based tray selection will be translated to PostScript accordingly.
- Accounting records will be written for the media transferred.
- Some backends (VP2110, Xerox 6180, CPS900) are able to monitor the print process.

The printers can be driven by two input data streams:

- By PostScript prepared for the printers: These files have to be spooled in using the so called PJM RAW format.
- By POD-AFP. In this case the backend converts the POD-AFP into PostScript. All parameters of the Professional Document Composer are preserved and transferred to PostScript.

The backends are by default configured as AFP backends, because they receive POD-AFP files from the spool. These POD files are converted into PostScript. The converted file is then transferred using the lp protocol. The CPS printers are an exception. These printers are driven via the external interface of the efi controller.

1.3.2 Configuration of the Backends

The backend is ready-to-print after setting up the IP-address of the printer and the name of remote queue on the 'General' tab in the 'Printer-Configuration'.

Océ PostScript printer (not CPS):
Here the queue will not be analyzed. By default LP is set as queue.

Xerox DocuTech: The queue will be analyzed. You have to define a new print queue on the Xerox DocuTech or you have to set an existing queue and to activate this queue on the printer. The insert the name of this queue in the general backend parameters. Otherwise the printing will be aborted with an LP error.

CPS700/900: The queue will be analyzed. Each queue has its own properties. By default the queues "print" and "hold" exist. They almost correspond to the settings of the so called mailbox.
Queue print: A job which is sent to this queue will be printed immediately. In this case 'printed' means that the job has been ripped and sent to the CPS by the EFI controller. The EFI controller then reports, that the job has been printed.
Queue hold: In this case the job remains in the EFI controller and has to be activated separately.

1.3.3 Tray numbers and Media attributes

PRISMAproduction POD-Module is based on the IOCA-AFP data stream, which is a subset of AFP. In this data stream the definition of the input material is possible either via input bin numbers or via media attributes. The control via input bin numbers is the older method, which defines the tray numbers from which the input material is to be drawn. It is not coded in the data stream on which paper material (medium) the files actually have to be printed. The newer method specifies the media exactly in the data stream. The media attributes describe color, weight, size, hole count and order count (with ordered material like tab sets) of the paper as well as the type. Additionally it is coded if a medium may be fused or not. So called "Cold Inserts" manly come from offset printing and must not be fused.

If there are media attributes in the data stream, they will be analyzed. Otherwise the tray numbers are analyzed. Consequently media attributes overrule tray numbers. An exception are printers which are not capable of working with media attributes (e.g. the VP2050). Here only the tray numbers will be analyzed. If a printer only works with media attributes and the data stream does not contain any attributes, the

backend creates generic media attributes from the trays as MediaType trayxx (xx = tray number). All other media attributes are omitted or replaced by default values (as for Xerox). This enabled you to also print tray number with these printers.

At the moment you can set **Cold Inserts** only with the PDC (via 'Blank Pages'). If the media attributes vary before and after the blank page and if there is no printed page with the media attributes of the blank page in the data stream, the blank page will be regarded as a cold insert. On the other hand cold inserts can be output directly via DocWorks (see the print room project). On a VP2110 cold inserts are displayed by a special icon. On a Xerox DocuTech they are visible as blank page inserts in the media attributes of a job.

Tabs can be output directly via the PJM or PDC. With the VP21xx/VP61xx a corresponding icon is displayed on the printer.

Tab printing (Xerox)

Tab printing is supported by Xerox. The names of the tabs in the data stream are interpreted and the size and order count is set accordingly on the Xerox machine. Other tab values like color and weight can be set defining a name of a valid paper attribute with the DEV_TAB_NAME parameter in the LI section of the backend.

Media Attributes (Xerox 6180)

With Xerox all media attributes have to be set. If they are not present, the backend replaces them by 0-values. These have to be adjusted in the printer. The calculates a default material from the material which is most often used. For this material not all attributes have to be set. The queue, in which the backend print the jobs, can be configured to replace the respective attributes.

Media Attributes (VP2110)

With the VP2110 not all attributes have to be set. They will not be analyzed by the VP2110 GUI. Tabs and inserts will be displayed on the printer GUI by a special icon.

Media Attributes (Oce_31x5, VP2070,VP10x5,VP2075)

The Oce_31x5/VP2070/VP10x5/VP2075 can only be driven by tray numbers. You can either set a tray, which is valid for the whole job, via the PJM job ticket or you can define different tray numbers per ranges using the PDC. Media attributes will be ignored with these printers.

Media Attributes (CPS700-900)

The CPS700-900 printers can handle up to 6 different media at the same time. These are: 'Normal', 'Special 1', ...'Special n' (depending on the printer 'n' is in a range from 4 to 1000). If the data stream contains tray numbers, these will be transferred to the types 'Normal', 'Special 1', ...'Special n'. If the data stream contains media attributes, these will be first transferred to tray numbers an then to the attributes 'Normal', 'Special 1', ...'Special n'. This is done by the so called extended AIMS (**A**ttribute based **I**nput **M**edia **S**election) concept. This concept replaces media attributes with tray numbers. If a medium is requested, a dialog appears in PRISMAproduction, which asks the user to select the tray, where the requested material has to be placed. The attributes of the media which are in the trays are kept in PRISMAproduction (and not on the printer). In this way all needed media can be defined. In case of a media request a dialog pops up where you can assign the media to tray 1...n = special1...specialn. Only then the job can be printed. Media which have already been loaded with a previous job are not offered anymore in this dialog.

Example: If a first job needs Media1 and Media2 (Mediax={MediaSize, Mediatype, MediaWeight, MediaColor, OrderCount, HoleCount}), you can assign Media1 to Special1 and Media2 to Special2. The next job wants Media1 and Media3. In this case you only have to assign Media3 to Special2...Specialn. The backend remembers that Media1 has already been defined for Special1. If a third job needs Media4 and Media5, you can use Special1...Specialn again.

1.3.4 Post Processing Commands

The following sources are available for postprocessing commands:

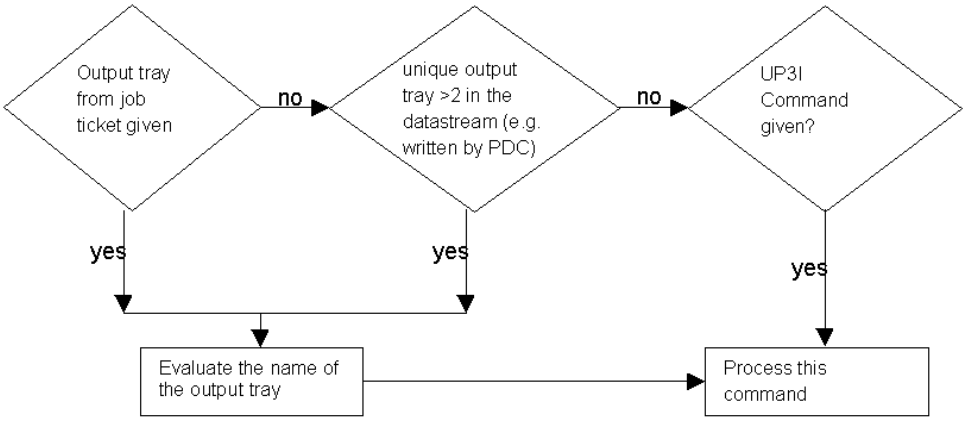
Constraints:
For historical reasons the VP5xxx family uses in IPDS mode tray 240 as SampleTray.

Output Tray Names:
The names of the output trays define the postprocessing in PRISMAproduction. Therefore the output tray, which is defined in the job ticket is analyzed. If no tray has been defined, the data stream will be analyzed. If all pages are sent to the same tray, the tray name will be analyzed and leads to a postprocessing device.

UP3i Commands:
If there are UP3i commands in the data stream, these will be analyzed. With UP3i trays, the PDC writes the respective command into the data stream. These commands are then analyzed again by the backend.

Post Processing from DocWorks:
Postprocessing commands which come from DocWorks are exchanged between Unity and the backend via an 'own channel'.

Order of the Evaluation:
1. Post processings from DocWorks
2. Constraints
3. Output tray names
4. UP3i commands



If UP3i commands have to be coded, a new output tray of the type UP3i has to be defined. This tray must then be connected to the respective control file.

Some post processing can only be addressed by output tray name, others only by UP3i commands. Some post processings can be addressed with both possibilities. The reason is because also in the case of pure PostScript post processings can be triggered by output commands. The following table lists all post processing and the way they are addressed (VP2100 is compatible to VP2110, VP6150,VP6200 are compatible to VP6250, Oce_3165 and VP2075 VP10x5 are compatible to VP2070):

Post Preocessing	Xerox	VP2110	VP6250	VP2070	Canon150
Booklet Advanced	XU[4]	OEM[3]	ED		XU[4]
Internal Bookletmaker	X[4]	X[2]	X		X[4]
Binding	XU	OEM	U	U	XU
BindingRight	X	-	-		-
Stacking	XU	XU	XU		XU
Single portrait left stitch	XU	U	U		U
Single landscape left stitch	XU	U	U		U
Single portrait right stitch	X	-	-		-
Single landscape right stitch	XU	-	-		-
Dual landscape left stitch	XU	U	U		X
Dual landscape right stitch	XU	-	-		-
Dual portrait stitch	XU	U	U		-
HighCapacityStacker	ED “HCS”	OEM			ED
SampleTray	XU	XU		XU	XU
DocWorks: “Other”	ED[1]	OEM	ED[1]		ED[1]
Subset stapling	X	X	X		X
User defined output tray	ED <otn>	OEM	ED <otn>	-	ED

X: is addressed by output tray names
U: is addressed by UP3i commands
ED: external device which can be addressed by output tray name
OEM: the output is sent to an external post processing device, which has no separate name
[1]: the tray number has to be defined in the LI parameters
[2]: the tray "IBLM" has to be defined manually, if the printer obtains an internal Bookletmaker
[3]: if an IBLM tray is defined, the ’external Bookletmaker’ also becomes an ’internal Bookletmaker’. If no IBLM tray is defined, the output will be sent to the external device.
[4]: both are mapped on the Bookletmaker

External Post Processings:
With Xerox DocuTech and with VP6250 external post processings are addressed by the tray name. If you want to print to an external post processing device, you have to create an output tray with a respective name and to select this for printing.

UP³I Finishing Names for DocuTech Printer:

In the POD Workflow of PRISMAproduction, the UP³I standard is used to define the

finishing options of a printer. This is true for both the printer configuration and the standardized AFP data stream, which is spooled in. In the context of standardization 2 new UP³I finishing options (XML UP3I Configuration files), currently only possible for DocuTech 6180, have been introduced. These are:

- Binding (DT6180_bind)
- Bookletmaker (DT6180_BookletMaker).

They define the PRISMAproduction representation and not necessarily the technical implementation or the printer hardware. Please configure these options, if they are available, which is especially useful for the DPconnect DocuTech Driver workflow. These new XML files are installed only, if the DocuTech_Driver license is activated. (If license was already deactivated, please reactivate).

With all other printer models, the post processings have no name. You can print on them by creating an output tray. The name of the tray is in this case not relevant.

External Post Processings via DocWorks:
You can now drive an external finisher by creating an output tray. The name of that output tray is the name of the external finisher. If you want to address an external finisher by DocWorks you have to set the variable OTHER of the LI section with this output tray number. When the backend finds an UP3I Command “special handling”, the “OTHER”-Variable is also evaluated. If you select the high capacity stacker from DocWorks, the name HCS is used for the external finisher. You can configure the name via the variable HCS_NAME.

Subset Stapling:
For subset stapling the respective UP3i commands must be in the data stream (e.g. jobs from DocWorks can contain subset stapling).

Stitch commands are now selectable via xml-file
The backend now ships an xml file (VP_2110_stitcher) to select the stitch commands (single/dual portrait/landscape stitch)

1.3.5 Background Process

A job will be regarded as printed successfully if the printing via the lpr protocol was successful. For some jobs background control of the print process is possible. The following printers support this feature: VP2110, VP61xx (planned), Xerox DocuTech and the CPS. For all other printers background control cannot be activated.
If this function is active, the status after submitting the job changes to ’output control’. The printing status is then analyzed periodically. Only if the job is printed suc-

cessfully, the spool is notified accordingly. To enable this function, the backend parameter IP has to be set to 1. For the DocuTech the rsh, rcp commands have to be activated.

Background check of DocuTech print status

If a print job was processed and sent to a DocuTech, the backend is ready to accept and process the next print job. Using the default setting, the backend will not check the status of the print job transmitted to the DocuTech. Usually, the operator wants the backend to check the status of the print process of the jobs transmitted to the DocuTech. In this case (interleaved processing) the backend will notice necessary operator interventions and issue a corresponding message; e.g. to load paper with new paper attributes.

In order to check the print status, the backend must be able to make a remote login on the DocuTech. In this case it is required to add the PRISMA print server as "trusted hosts" in the ".rhosts" file in the home directory of the user name root. This means, the server name and the IP address of the print server must be added to the file `/etc/hosts` and the entry "<server_name> root" must be added to the file ".rhosts". located in the home directory of user root. Both files are located on the DocuSP server.

In this case, the backend will use a different login name than root; this name may be entered as parameter rsh_user in section LI. E.g.:

`/etc/hosts`

```
  ...
   printserver <ip-address>
  ....
```

or

```
/root/.rhosts                    /<username>/.rhosts
...                              ...
printserver root                 printserver root
...                              ...
```

The user name can be specified in the LI-Parameter RSH_USER.

After print file conversion and successful transmission to the DocuSP server, the spool status of the print job is switched to "output control" in the "interleaved processing" case and to "successfully completed" in the default case.

In the interleaved processing case the print status is periodically checked on the DocuTech by a background process. If a operator intervention is necessary, this background control issues a corresponding message to the print server.

The background process will issue the PRISMA spool states "finished" or "error", if all print files of the corresponding ODS job have been printed with the DocuSP states "Successfully Completed", "Completed with Errors", or "Faulted".

Under certain circumstances various print files of one ODS print job may be printed (e.g. PJM job ticket or header pages, if switched on by the PJM.

If the DocuSP status of all printed files for one ODS job is "Successfully Completed", the ODS job will be set to spool status "finished", in all other cases spool status "error".

If a print job is transmitted to the DocuTech via LP command, the backend will convert the next print job, while the background process controls up to 10000 jobs on the DocuSP server.

The operator should delete the completed print jobs on the DocuSP server to allow the background to empty its internal list.

If a job is deleted on the DocuSP server, which was not completely printed, the PRISMA spool status is set to "interrupted".

The DocuTech driver now supports a default paper

The docutech driver now counts the pages for every media. The media with most of the pages is the default medium. For that you must not specify the weight nor the color. They are set to printer default on the printer.

1.3.6 Parameters of the Océ PostScript Backend

All Backends (Parameters which are not supported by a printer model are not displayed in the Printer Configuration)

ACCOUNT	Identifies the name of an account to which costs associated with this job accrue. You can also use the VarioPJM account field. If no value for ACCOUNT is set in the LI section of the backend the VarioPJM-value is taken.
AFP2PSFLAG (default=0)	0x00: POD Datastream is supported 0x01: use external afp2ps (license protected) + 0x04: convert tray numbers to media attributes according to JDE JDL mapping file + 0x08: choose converter according to afphead information (this costs performance) if the header was written by the Unity or PDC, use the POD converter + 0x10: Don't force a back page between a duplex front - duplex front page sequence
AUTO_DB_UPDATE (default=0)	0: the backend reads the paper tables from the database every time it's started. 1: the backend reads the paper tables from the database every time a job is started.
BANNER_PAPER_NAME	Paper name for the banner page. Empty string means the backend generates no additional banner pages. Enter a 1 to print a banner page without specifying the paper format.
CONFIGFLAGS (default=0)	Defines the configuration. 0x01: Instead of taking the ODS-Id as the sender, the sender name of the (Vario) PJM is taken (-> IP=1 won't work).
DUMMYDEV=0	Internal use only.
IGNORE_JOG (default=0)	Some finishing devices have problems with jog commands. So you can suppress this command: 1: Ignore jogging commands, which are specified in the data stream 2: Ignore jogging commands, which are specified in the job ticket 3: Ignore jogging commands, which are specified either in the data stream or in the job ticket

IP (default=0)	This parameter handles remote print status control IP=0: Default setting of this parameter is 0, because in this case, no special settings have to be done on the DocuSP server. In this case the backend regards all transmitted jobs as successfully printed and switches the spool status to "finished complete". The operator has to check the messages on the DocuSP on his own. IP=1: If the LI switch IP "interleaved processing" is set to 1, transmitted jobs get the spool status "output control" and are periodically checked by the backend. For Docutech backends, which send their jobs to a DocuSp version 2.x controller see 'Xerox PostScript Printers (Docutech)' on page 89.
IP_TIMEOUT (default=20)	Timeout of getting the job status information from the printer.
ITERATE_JOB_ COPIES (default=0)	Jobcopies and filecopies are multiplied by default (0). With this setting (1) the backend is forced to send each job copy separately.
ITRAYMAP	File under `/u/prismapro/lib/itraymap` containing a mapping table which maps the input trays to media attributes.
JOBDELAY (default=0)	Delay time in ms after job submission. After this delay time the backend reports ready after submitting a print job. This can avoid that a backend receives more than one print job in a big cluster.
JOBTICKET_ REPLACEMENT (default=0)	Replaces the job ticket in "raw"-mode.
LPR_OUT_PORT (default=1024)	TCP/IP-PORT used for lp-submission.
LPR_PORT	Portnumber
MEDIAMODE (default=3)	MediaName MediaNames are not supported by all PostScript printers. Therefore it may occur that a job, which uses paper selection via MediaName cannot be printed on such a printer. 0: the medianame is passed 1: the mediatype is passed 2: the color is passed 3: the color together with the type and weight are passed
OTHER (default=0)	Determines the output tray for the finishing operation "other" coming from DocWorks.
OUTPUT_PATH	Path for storing the converted file (only PDF out).
PDSFLG2 (default=0x20)	0: don't jog copies 0x20: jog copies
TRACEFILE	Internal Trace-File

TRACEFLAGS (default=0)	0x4: Tell why a postprocess was processed 0x10: Enable tracing for the lpr
TRACEMODE (default=0)	Internal Trace-Mode 1: Create trace files while printing and store them in the directory `diag/printers` 2: Like 1. but the backend doesn't send the print file to the printer.

Océ PostScript Printers

GROUPNAME	Groupname specifies the name of the department or group of which the user is a member. This entry is used to indicate on which group or department to account this job. This entry is logged in the job record of the account log. If the group name is not specified in the ticket, the controller will treat it as an empty string.
JOBTYPE (default=1)	The jobtype specifies whether the print job will be stored in the digital mailbox (INTERACTIVE) or whether the print job will be appended to the Automatic print queue (AUTOMATIC) for automatic printing. 1: interactive: the job will be saved in the mailbox. This allows you to modify the job attributes later or to make reprints. 2: automatic: the job is printed immediately.
JOB_TICKET_REPLACEMENT (default=0)	Replaces the most important job ticket information of raw spool in PostScript files, if a job ticket is present.
RAW_SOCKET_PORT (default=9100)	Portnumber of the RAW_SOCKET_PORT.
RAW_SOCKET_STREAMING (default=no)	"yes" enables sending files via socket. This allows you to send files > 2 GB. Generally files which are send by socket are not spooled on the target printer. Therefore it is not possible to print copies. If you want to print copies, the job has to be sent as many times as the desired number of copies. The port has to be activated on the target printer.
PRINTQUALITY (default=2)	Defines the print Quality for the printer type Océ 31x5. 0 : 8x8 dithermatrix 1 : 10x10 ditheratrix 2 : 12x12 dithermatrix 3 : 14x14 dithermatrix (Ab DAC R4.1) 4 : 16x16 dithermatrix (Ab DAC R4.1)
PRINTMODE (default=0)	Defines the print mode for the CPS700-900. 1: Direct printing: File is printed immediately 2: Print file will be sent to the mailbox

Xerox PostScript Printers (Docutech)

ACCOUNT_TIC (default=-)	Identifies the name of the account to which costs associated with tickets accrue.
CONTROLLER_SOFTWARE	You can set the Software Version.
DEF_PAPER_NAME	Default paper name.
DEF_TAB_NAME	The paper name determines the name of the paper in the database from which the color and weight of the inserted tabs are taken.
DPCONNECT_SUPPORT	DPconnect compatibility mode. Default. Finishing Option defined in AFP file (e.g. overwritten by DPconnect ticket translation rules) is taken. 1 The original finishing operations, as defined in the DigiPath XRX Ticket, are taken. 2 Not only finishing option, but also the DigiPath ticket parameter “XRXsenderName” is taken as job identifier on DT6180. The IP_MODE has to be set to 0. For DPCONNECT_SUPPORT 1 and 2, please do not use print slaves, but normal slave. Master servers are obviously also possible. **Workflow example**: DigiPath Job with Bind option. This job is send to PRISMAproduction/DPconnect. DPconnect should import this job, to be printed with the same spooled file on a VP5000 or a DT6180 via rerouting. The “bind” option should be replaced on the VP5000 with the stitch option. This requires the DPconnect translation rule “Bind” “Stitch” and the job is spooled in as a stitching job. Via the DPCONNECT_SUPPORT switch, you can define, if the job should also be stitched on the DT6180 (DPCONNECT_SUPPORT=0) or if the job should be printed on the DT6180 with the original XRX binding command (DPCONNECT_SUPPORT=1)
FONT_TIC	Determines the font used to print job tickets.
HCS_NAME	System name of the High Capacity Stacker. The HCS can be addressed by DocWorks (for the DocuTech this is an external post processing).
CONTROLLER_SOFTWARE	Set the DocuSP version. Version 3.0 supports right subset stapling. 0: The DocuSP version you are printing to is < V3.0. 1: The DocuSP version you are printing to is equal or greater V3.0

IP (default=0)	This parameter handles remote print status control IP=0: Default setting of this parameter is 0, because in this case, no special settings have to be done on the DocuSP server. In this case the backend regards all transmitted jobs as successfully printed and switches the spool status to "finished complete". The operator has to check the messages on the DocuSP on his own. IP=1: If the LI switch IP "interleaved processing" is set to 1, transmitted jobs get the spool status "output control" and are periodically checked by the backend. To allow this check via remote shell command, "trusted host" settings have to be made in the '.rhosts' and 'hosts' file on the DocuSP server (the command rsh <docutech_ip> [-l <rsh_user> listjob has to be working on the PRISMAproduction server, where the backend runs). See section background process for further details. If the print job is actually printed on the DocuTech, the PRISMA spool status is set to "finished complete" (or "error" in an error case). It is important to point out that the backend will not be able to start if this IP parameter is set and the ".rhosts" settings are not correct.
RSH_USER (default=root)	The backend perodically logs on to the DocuTech to check the print status. Usually the user, which is currently active in PRISMAproduction is used. Here you can define another user for the remote login on the DocuTech.

Behaviour of the PostScript Backends for DocuTech and Vario 21xx Printers with normal POD IOCA jobs (e.g. via Unity) and "Raw" PS jobs:

	AFP / IOCA Jobs	**"Raw" PS Jobs without any ticket**	**"Raw" PS Jobs with XRX Ticket**	**"Raw" PS Jobs with OJT Ticket**	**"Raw" PS Jobs with JEC Ticket**
DocuTech xdtbe	PS with XRX Ticket	PS with XRX ticket	Ticket Merge *	not supported, **	not supported, **
oce PS Backend driving 2110	PS with OJT ticket	PS with OJT ticket	not supported, **	Ticket Merge *	not supported, **
oce PS Backend driving 3165	PS with old JEC ticket	PS with old JEC ticket	not supported, **	not supported, **.	Ticket Merge *

Additional Information : Ticket Information for AFP / IOCA jobs are extracted from both PJM Tic and AFP.

"PJM-Ticket" information for jobs, spooled in Postscript are extracted from PJM, only. Naturally ticket information, based on AFP data (Medium Map AIMS etc.) cannot be extracted.

*) Ticket Merge: Original ticket is kept, but the part of the ticket, which has equivalent PJM TIC parameters is overwritten (merge of PJM and target ticket). Example: Copycount.

**) PostScript files with ticket of type A, are blocked for drivers, using type "B" (A != B) because important ticket information, available in type A, but not in type B, may result in unpredictable behaviour. (A/B = XRX/OJT/JEC). An error message is issued.

1.4 Print Parameters of the Linedata Printer Driver (for Xerox Docuprint)

The license LCDS_Driver 4.0 is required to use this driver. With this package you can receive LCDS line data/DJDE/Metacode jobs (typically from an IBM host via TCP/IP download) and pass them through to a Xerox DocuPrint (via TCP/IP or /370 channel).

The functionality is similar to IBM's "Job Entry System":

- insert an FCB record,
- prepend 2 header and append 2 trailer banner pages,
- convert any type of carriage controls (ANSI, IBM3211, NONE) into IBM3211,
- drive the physical printer.

(No ifilter, xfilter, AFP, IPDS is involved in this.)

Printer Configuration

Open ’Configuration’ -> ’Printers’ and create a new printer (select ’Docuprint’as ’Printer Model’). The ’Linedata Printer Driver’ is assigned automatically to this printer.

When you choose ’DocuPrint’ as the printer name, you will have the advantage to run the demo-job (see below) and MVS-download-jobs without any modification of a job-ticket.

After clicking ’Save’, the ’Data Format’ ’LCDS’ will be predefined in the general printer settings.

Complete the configuration by setting the /370 channel parameters:

- ’/370 channel unit’ must match the hardware equipment of your Linux server
- ’/370 address’ must match the /370 address of your Xerox destination printer

or by setting the TCP/IP parameters, respectively:

- ’Host’ must match the IP-address or host name of your Xerox destination printer
- ’Port’ must match the port number where your Xerox printer listens

Like in most other printer types the print parameter "DUMMYDEV" is supported:

- DUMMYDEV 0 = sends output to the physical printer
- DUMMYDEV 1 = discards output, which may be useful for testing

FCB configuration

You can configure this item with an own applet, which you select from the 'Configuration' node of the PRISMAproduction Explorer.

Testing with the Demo-Job

After having configured an LCDS-Printer and before receiving MVS-Download jobs, we recommend to print the demo job. This demo job contains 4 pages of EBCDIC-coded data with ANSI carriage controls. The jesfilter will add an FCB ("FCB1"), add 2 header banner pages, convert ANSI to IBM3211 carriage controls, and add 2 trailer banner pages before submitting the job to the spool:

1. Open the PJM with an LCDS job ticket. Open the job ticket `$DEMO/jes/jes-demo.TIC`.
2. Under 'Printer' select the LCDS printer you created before in the printer configuration.
3. Activate 'Print' and 'Header/Trailer' in the 'Job Attributes' section of the PJM.

Finally submit the job with the 'Submit' button.

Job-Queue

Linedata Printer Driver-Jobs appear in the Job-List with 'Type'='LCDS". This assures that the spool will direct LCDS-jobs to an LCDS-printer, and not to an AFP-printer. Other jobs usually appear here with 'Type'='AFP' (including ordinary standard LCDS-jobs converted by the xfilter).

Restrictions:

- Page number (see column "Pages" above) is not known exactly. This is because PRISMA cannot know the page structure, as the break-down of the job into pages would require JSL-information (VFU, PDE, ...) which is available on the Xerox-destination-printer only. Therefore a simple-minded page-counting has been implemented: Each advance across channel-1 is counted as one page.
- Printing page ranges (see column "Print range") is not supported (same reason as above).
- Check-pointing is supported, but the only check point is at job begin (same reason as above).
- Accounting records are written. But the accounted page-numbers have the same restriction as mentioned above.

Printers configured as "XEROX LCDS Printers" appear in the printer pool. You can operate (activate, deactivate, reset, interrupt, assign to job classes) them in the usual way.

Successful operation of LCDS jobs will produce messages in the messages window.

Download Configuration

1. Open the "Download Configuration" and create a new Download port.
2. Enter the port number which matches the TCP/IP-printer in MVS.
3. Click ’OK’. Then you have to change the following two settings:
   - Set "AddCRLF" to false
   - Set "DefaultTicket" to ".LCDS_Driver.TIC" (note the leading dot)

This job ticket sets the job type to "LCDS", switches banner generation on and selects the printer "DocuPrint".

Without these settings the JES-filter will probably complain because of nonsense record lengths.

- Dependent on the selected printer type, the tab ’JES’ appears in the PJM ’File Properties’ dialog. Here you can select the default carriage control character.

On MVS-host side you will need:

- a properly configured TCP/IP-printer (with the IP-Address of your Linux-server and port number of your download daemon)

some print-jobs (with ANSI, IBM3211, or without carriage controls).

1.4.1 Print Parameters

CONTINUE_ CONFIRMATION	0 Print without confirmation (default) > 0 Popup a confirmation message after printing/sending N channel 1 PCC's. The answer option for this confirmation message contain 3 options: - "Continue printing" = just a confirmation - "Restart job from beginning" = reprint current job from begin - "Cancel print job" = cancel and hold the job This dialog appears for every print job, when N Channel 1's counts have been processed. **Example**: Jes Demo Job with header page and therefore 7 channel 1 send to a (dummy)linedata printer and CONTINUE_CONFIRMATION set to “2”. -> Message occurred after process of 2 channel 1 -> In job queue window : Page position 2, (accumulated) printed pages: 2 -> Operator checks loaded form on line printer and may adjust form "Restart" option chosen. -> Job is restarted and 2 channel 1's are again processed/transferred. -> Again Message with spool entry: Page position 2 and printed pages 4. -> Operator now confirms form adjust "Continue" option chosen: Rest of 5 channel 1's are processed -> After print Queue entry display: Page position 7 and printed pages 9. **Important** : The Spool Page counter for Line-data jobs is based on channel 1's. To use this feature, channel 1 PCC's must be present in the data, downloaded with the Download input module either as ANSI PCC's (0xf1) or IBM3211 PCC's (0x89/8b) including the respective Ticket entry defining the PCC Type.
DUMMYDEV=0	Enable or disable the internal printer simulator. 0 Printer simulator disabled 1 Printer simulator enabled
ERROR_COUNT	Set count of printer errors per job, which has to be reached, before the job is aborted. 0 = abort after first error.
IOSLEEP	Set i/o sleep time in msec. 0 No sleep -> sync execution >0 Sleep time -> async execution

MONITOR=0	Loglevel for internal traces. All logs are written into a file, specified in TRACEFILE. 0x0000 No logs written 0x0001 Some few logs 0x0002 Diagnostic logs. 0x0004 LPT-internal logs.
NOP_LENGTH	Set the data length in bytes for the noop command on socket connection. If set to 0, max. data length is used.
NOT_READY (default=2)	Set the delay in seconds, after that not-ready status is displayed for the printer (if no data can be transmitted).
ONLINE_CHECK (default=15)	Set the interval in seconds, after that a printer is checked for being online (only if printer is idling).
POSTFILECMD=	Command to call after file transmission is completed. A possibly specified variable $NAME is replaced with the job name.
POSTJOBCMD=	Command to call after job transmission is completed. A possibly available variable $NAME is replaced with the file name.
PROGRESS (default=250000)	Update print progress information. =0 Update only only at end of file or copy changes. >0 Amount of transmitted bytes, after which progress is updated.
SIGNALS (default=10:12)	Set two ':' separated signal numbers, which should be used for signalling: - end of an i/o request (def. 10 = SIGUSR1) - not-ready to ready interrupts (def. 18 = SIGUSR2) when print data is sent via 370 channel. 0 Signal should not be used >0 Signal to use Available signals: enter the command 'kill -l' to get the list of available signals.
TIMEOUT	Set i/o timeout in sec. -1 No timeout at all. 0 No timeout specified, use defaults. >0 Timeout in seconds.
TRACEFILE=/u/prismapro/diag/printers/<Printer name>.trc	Filename of the trace file. The name must be unique for each configured printer (Printer name is the name of the logical printer).

1.5 Print Parameter for Network-Connection

1.5.1 FTP

ABORT (default=0)	If set to 1, the ftp-backend sends an ABORT command to the ftp-server, if a running data transmission must be cancelled due to an user request to cancel or interrupt active print job(s). This parameter can be enabled, if the ftp-server supports async abort commands. In the default setting, a running upload is canceled simply by stopping data transfer and deleting the incompletely uploaded file.
BUFSIZE (default=0)	Buffer size for reading data blocks from the print file. The backend reads always this amount of bytes and sends as much as possible of them to the ftp server. The unsent data is sent in following send commands: = 0 Use default buffer size (recommended) > 0 Use this size in bytes (max. 60 Kbyte).
FS_DELAY (default=15)	Set delay time in seconds, which is used to wait , if destination filesystem becomes full.
FS_RETRY	Specify a retry counter, after which the backend terminates, if a file-system full problem is reported n times for the same print file. For infinite retrys set the counter to 0.
IOSLEEP (default=100)	Set ftp sleep time in msec. 0 No sleep >0 Sleep time if nothing to do
KEEPALIVE (default=60)	Keeps the ftp connection alive by sending noop requests to the server. This happens only if the backend idles. 0 No alive commands specified. >0 Alife intervall in seconds.
NOT_READY (default=2)	Set the delay in seconds, after that not-ready status is displayed for the printer (if no data can be transmitted).
PASSIVE (default=0)	Use passive mode for data transmission (not supported by all ftp-servers) 0 = passive mode disabled (highest compatibility) 1 = passive mode enabled (better, can be useful with firewalls)
PORT	Optional parameter to set the ftp port number, which is expected by the ftp server.

POSTOCTCMD	Command to call after new local oct file is generated, but before it is transmitted. Some possibly available variables in the command line parameters are replaced: $ID is replaced with the job id. $PRINTER is replaced with the printer name. $NAME is replaced with the oct file name.
PREFILECMD	Command to call before file transmission is started. The allowed variables in the command line parameters are replaced: $ID is replaced with the job id. $PRINTER is replaced with the printer name. $NAME is replaced with the file name (this file can be modified in this action).
PROGRESS (default=250000)	Update print progress information. =0 Update only at end of file or copy changes. >0 Amount of transmitted bytes, after which progress is updated.
RENAME=1	If set to 1, the ftp-backend sends the oct file with an invalid name and renames it to the valid name after successful transfer only. If set to 2, the ftp-backend tries to delete a possibly existing oct-file before the rename command is executed.
RETRY=0	If set, the ftp-backend automatically retries to send the data again, in case of a network error. 0x0001 retry after time-outs. 0x0002 retry after minor errors. 0x0004 retry after severe errors. 0x0007 retry after all errors.
TIMEOUT=-1	Set ftp time-out in seconds. -1 No time-out at all. 0 No time-out specified, use defaults. >0 Time-out in seconds.

Additional parameter transmission with FTP backend

If the currently implemented parameters, transmitted with the FTP backend to a target system are not sufficient, a new interface of the FTP backend can be used. This interface gives the opportunity, to change the OCE custom ticket before it is transferred to the target system.

An example script is delivered and can be used directly. This script copies the parameters from the original ticket to OCE custom ticket. To configure it, please open the Printer Configuration window and set the POSTOCTCMD parameter of your FTP-Backend-Printer to the value:

```
/u/prismapro/bin/addOCTinfos $ID $NAME $PRINTER
```

Additionally the original job ticket must be catched, if this job is not printed directly, but creating new print jobs (different job IDs to the original job). Therefore another script is delivered to be executed via the Job-ANI interface. To configure it, please open the Printer Configuration Window and select `collectReferenceJobInfos` in the Job-ANI-Dialog

Restrictions:

- Job grouping is not supported with the delivered script files.
- The scripts are created for the usage of the FTP backend on the master system only. All configuration must be done on the master system too.
- LCDS-Printing: the OCT will get Type LCDS, but the generated and transmitted print file is in AFP-Format. Therefore the job will not be spoolable directly and must be manipulated on the target system, e.g. with TicketRules or with another script, or the delivered "addOCTinfos" script can be adjusted.
- POD and OLDS-Jobs can cause similar problems as LCDS.

1.5.2 LPR

PORT (defaulr=515)	Port number of the LP service on remote print system.
TRACEFLAGS (default=0xffff00f8)	Define the trace flags. See also the 'Trace Level' parameter. Common used flags sets: 0xffff1cf9: Trace are written with performace information, no print data files are collected (Standard trace). 0xffff1cfb:Trace as well as data files are collected (Data trace).
TRACELEVEL (default=0)	Define the detail level of the trace file. When trace level is 0 error traces are still being written. Traces are written into one of file: `/u/prismapro/diag/lp/lprbe.log`. Default value 0. 0 - Only error messages are written 1 - Additional section traces are written 2 - Additional method traces are written 3 - Additional information traces are written 4 - Additional detail traces are written

1.5.3 LI-Section

DUMMYDEV=0	Enable or disable the internal printer simulator. 0 Printer simulator disabled 1 Printer simulator enabled
INITCMD	Command to execute, during backend startup. The allowed variables in the command line parameters are replaced: $PRINTER is replaced with the printer name. $INIFILE is replaced with the backend start-parameter file name.
MODE (default=0)	File processing mode for: 'Executable Program' connection type: 0: Process all print files separately 1: Process all print files at once
MONITOR (default=0)	Loglevel for internal traces. All logs are written into a file, specified in TRACEFILE. 0x0000 No logs written 0x0001 Some few logs 0x0002 Diagnostic logs. 0x0003 FTP-internal logs.
POSTFILECMD	Command to call after file transmission is completed. The allowed variables in the command line parameters are replaced: $ID is replaced with the job id. $PRINTER is replaced with the printer name. $NAME is replaced with the file name.
POSTJOBCMD	Command to call after job transmission is completed. The allowed variables in the command line parameters are replaced: $ID is replaced with the job id. $PRINTER is replaced with the printer name. $NAME is replaced with the job name.
TRACEFILE (default=`/u/prismapro/diag/printers/<Printer name>.trc`)	Filename of the trace file. The name must be unique for each configured printer (Printer name is the name of the logical printer).

2 Input-Filter

2.1 Introduction

The Input-Filter reads data from input media and passes these depending on the output format to the communicator (Process chain) or directly to an output file. During this process it takes care of the following tasks:

- Hide the device dependent handling such as multi-volume tape handing, tape label consistency check, Xerox floppy format read.
- Check the consistency of the data without interpreting the data itself in detail. Consistency of block and record structures (if available) in data are checked.
- Where needed, modify the printer control codes if these consist of multiple bytes.
- Convert, when requested, packed data (Honeywell, Radix 50, Univac) or character sets (EBCDIC or H6BCD) to ASCII.

The Input-Filter supports several input media. Since the data is in a different format dependent of the input media, the data type/media combination must be supported for a correct functioning. At the moment the following input devices are supported: 3,5" & 5.25" floppies, Cartridge (18/36-Track), HD, QIC and Tape (9-Track). See the Supported features chapter to see which data type and host/label combination for what input media is supported.

The Input-Filter now also can write Xerox Resource files to tape (Cartridge, QIC, Tape) and floppy (3.½" 1.2 MB or 5¼" 720KB, Xerox Format) devices.

Note: See also ’Installation of Tape Drives’ on page 33 for BIOS changes needed for the installation of SCSI tape devices.

2.2 New Features

Input-Filter version 03.10.45 or higher.

1. General: The Input-Filter trace filenames are no longer identical each time the program runs. It now has the following syntax: jobid<jobid>_pid<process id>_ifilter.log.
   <jobid> is the 8 digit number of the PRISMAproduction process identifier or '-------' if it is not specified. The <process id> is the linux process id of the input-filter. The first 2 fields are also used for the other trace files (jobid<jobid>_pid<process id>_).
2. LCDS-Module: /370-Channel only - Input-Filter will read latest FCB from HD and send this, before data is received, to the X-Filter as if it was received over the /370-channel.
3. LCDS-Module: Print While Spooling in between Input-Filter and X-Filter is handled without using the communicator. The difference in handling is specified in the JCF-file with the following (new) keywords: OutFormat GENERATE_GCI, OutType, NextProcess and NextOptions.
4. LCDS-Module: When resource files are processed and the OutSubcat parameter is set to the value AUTO the Input-Filter will create these sub-directories if they do not exist yet.

/370 Channel device: ChanStopTime causes the I-Filter to close the channel after this time has passed and no data is received. This feature is only valid when the /370-Channel device is used and the channel timeout value (JCF keyword ChanTimeout) is set to 0. When the SIGUSR1 (Signal No. 10) is send to the I-Filter it sets the channel timeout value to the ChanStopTime JCFkeyword value. The process Identifier (PID) is stored in a file in the `/u/prismapro/data/jobinput/ifilter` directory with the name PID_<Channel device>_<Channel address>.

2.3 Hardware Requirements

The Linux operating system determines the tape device names (`/dev/(n)st0-`....) during the system startup. When devices are not switched on they are not found and do not have a device name. This also means that it is not possible to guarantee that a device has always the same device name.

The Input-Filter uses the JCF-Keyword DeviceName to distinguish between Cartridge, QIC and Tape. It assumes that the SCSI Id's are however set as is described in the table below. It ignores the SCSI host adapter, so the tapes can be connected to different host adapters.

Dependency	Supported formats	SCSI ID
Cartridge	3480/3490 18/36-track	3
Tape	Reel Tape, Density 1600/6250 bpi	4
QIC	1.2, 8, or 100 GB QIC tape drive	2

2.4 Functionality

2.4.1 Output Format

Output Format	JCF Parameter OutFormat	Description
Device output	DEVICE	Output is written to a device specified in the OutputDevice section. At the moment this feature is limited to Xerox Resource files as input and tape or Xerox floppy (3½" and 5¼") devices as output media.
File output	CRLF	Output into a file with 0x0d0a as record separator, or when no record structure exists after each block. If these bytes already exists in the record data, they are not added. This is checked for each record separately.
File output	CRLF_FORCED	Output into a file with 0x0d0a as record separator, or when no record structure exists after each block. These are always added, also when record/block ends with this byte combination
GCI output	GCI	Output via GCI.
GCI output	GENERATE_GCI	Generate GCI output format.

2.4.2 APA-Module

2.4.2.1 Input Media/Data Type Combinations

A quick overview of the supported input media/data type combinations:

Media	Supported data type(s)	Description
3.5” Floppy	AFP Resources	AFP resource files in DOS format
HD	AFPDS, Pure AFPDS	AFP print data written into a file.
Cartridge, Tape, QIC or as image file	AFPDS, AFP Resources, AFP MVS libraries	AFP print data AFP resource files AFP libraries in IEBCOPY format

2.4.2.2 Data types

Description of the supported Input-Filter data types and which keyword needs to be used:

Media	JCF-Parameter Infile-Type	Description
AFPDS	AFPDS	The data itself can be Structured Fields (SF), linedata and mixed data (Structure fields and line data). The data structure can be: Fixed record, Variable record, Variable blocked record, Delimiter, or pure Structured fields.
AFP MVS libraries	AFP_Resource	Multiple resource files written in the IEB-COPY MVS format.
AFP Resources	AFP_Resource	AFPDS Resource (fonts, overlays, pagedefs, ..). Format pure Structured Fields (SF) only (MODCA's).
Pure AFPDS	Pure_AFPDS	Pure Structured Fields (SF), no linedata allowed

Data of type ”Pure_AFPDS” are written in FORMAT_BLOCK and give a higher performance compared to ”AFPDS” data which is written in FORMAT_RECORD.

2.4.2.3 Data Structures

There are several data structures supported for HD AFP files as well as for Labeled and non-labeled AFP tapes. For the labeled tapes the data structure description is documented in the tape labels. For the HD as well as the non-labeled data this in-

formation is not available and must be specified by the user. The information the user has to specify by a certain data structure can be determined by the following table as is valid for the AFPDS and Pure AFPDS data types.

Record structure	**JCF-Parameter JDE_Name**	**Data structure description**
HD mixed data	ATXT_LF	Mixed data and line data each record must be ended by a <CR><LF> (0x0d0a). This is not required for Structured Fields (SF) unless the JFC-Parameter AfpSfTrim is used..
Fixed length RecordLength= <Value>	FIXED	Fixed length Structured fields, Linedata or mixed data with fixed record length. This means that no record or block length fields do exits in the data file (besides the AFP record length fields) and each record has the same size. Record length must be specified using the JCF_Parameter RecordLength.
Variable length (IBM RDW)	VAR	Structured fields, Linedata or mixed data with IBM binary formatted record length field (2 bytes), followed by 2 reserved bytes (RDW's). No Block length (BDW's) bytes are allowed.
Variable length, blocked (IBM BDW, RDW)	VARBLK	Structured fields, Linedata or mixed data with IBM binary formatted record length field (2 bytes), followed by 2 reserved bytes (RDW's). Block length bytes (BDW's) do exist too with the same format as the RDW's.
Undefined length (Delimiter) RecordConstant= 0x0D0A	UNDEF	Structured fields, Linedata or mixed data which use a byte sequence (delimiter) to indicate the end of a record. The delimiter must be specified using the JCF-Parameter RecordConstant and can be up to 4 bytes.

2.4.2.4 3.5" Floppy media

The 3,5" floppy media is only supported for the AFP resource files.
The DOS floppy is mounted and the files are then accessed as normal HD files. The floppy is mounted with the path '`/mnt/floppy`'.

2.4.2.5 Hard disk media

JCF-Parameter InFileType	JCF-Parameter JDE_Name	Restrictions/Requirements
AFPDS Pure_AFPDS	ATXT_LF	Mixed data and line data each record must be ended by a <CR><LF> (0x0d0a). This is not required for Structured Fields (SF) unless the JFC-Parameter 'AfpSfTrim' is used.
AFPDS	FIXED	Fixed length AFPDS data records.
AFPDS	VAR	Variable length AFPDS records.
AFPDS	VARBLK	Variable length AFPDS records with block structure.
AFPDS	UNDEF	Data which uses a delimiter as record end identifier.
AFP_Resource	ATXT_LF	Only single file resources.
LINEDATA	ATXT_LF BLOCK	ASCII text as input for the 'linedata' filter can be read from hard disk using JDE-Name ATXT_LF. The records must be separated by <LF> (0x0a) characters. This data type can also be used to read unstructured data (like PCL). In this case use JDE_Name BLOCK which will result in output in FORMAT_BLOCK with 32000 bytes/block.

2.4.2.6 Tape media

The tape label format which is used internally as HOST=ANSI and LABEL=ANSI has been extended to handle also the IBM(D)OS label differences. The new extensions are detailed in the table below.

The tape support can also handle non-labeled tapes, were the user must specify additional parameters, which describe the data format. No Multi-volume support exists for non-labeled tapes. Only non-labeled resource tapes using IEBCOPY format are supported.

JCF-Parameter InFileType	JCF-Parameter JDE-Name	HOST/LABEL (Internally used)	Description
AFPDS	ANSI01	ANSI/ANSI	Used for all ANSI like labeled tapes which have at least VOL1/ HDR1/HDR2 labels. This includes all STANDARD MVS- and BS2000 tapes.
AFPDS	FIXED	UNDEF/NONE	Used for all non-labeled print data tapes with fixed length AFPDS data records

JCF-Parameter InFileType	JCF-Parameter JDE-Name	HOST/LABEL (Internally used)	Description
AFPDS	VAR	UNDEF/NONE	Used for all non-labeled print data tapes with variable length AFPDS records.
AFPDS	VARBLK	UNDEF/NONE	Used for all non-labeled print data tapes with variable length AFPDS records with block structure
AFPDS	UNDEF	UNDEF/NONE	Used for all non-labeled print data tapes which uses a delimiter as record end identifier.
AFP_Resource (single)	ANSI01	ANSI/ANSI	Used for all ANSI labeled tapes where each resource file is in a separate file in binary format.
AFP_Resource (IEBCOPY)	RESLIB	IEBCOPY/ ANSI	Used for AFP MVS libraries, no filenames can be specified. The directory listing is only supported for the library names, not the library contents. A single library can then be selected to be read.
AFP_Resource (IEBCOPY)	RESLIBNONE	IEBCOPY/ NONE	Used for non-labeled resource tapes.

2.4.3 LCDS-Module

A quick overview of the supported **input** media/data type combinations:

Media	Supported data type(s)	Description
3.5" Floppy	Xerox Resource	Supports only 1.2 Mbytes Xerox resource floppies
HD	Xerox Print Data Xerox Resource	Xerox print data (LCDS) Xerox resources with 128 bytes Xerox LPS header
Cartridge, Tape, or as image file	Xerox Print Data Xerox Resource	Xerox printout (LCDS) Xerox resources with 128 bytes Xerox LPS header
online, 370-channel	Xerox Print Data	The following printer emulations are supported: 3202, 3211 and 6262. See also JCF-Keyword 'ChanEmulation'.

Also the following output media/data type combinations are supported:

Media	Supported data type(s)	Description
3.5" Floppy	Xerox Resource	Supports only 1.2 Mbytes Xerox resource floppies
Cartridge, Tape, or as image file	Xerox Resource	Xerox resources with 128 bytes Xerox LPS header

To process Xerox Print Data without using the communicator the following setting should be used: OutFormat GENERATE_GCI and OutType NamedPipe or PrintWhileSpooling. This setting must be the same as the setting of the parameter InType of Xfilter. To start the next process (X-Filter) the NextProcess/NextOptions parameter should be set correctly. See the description of the JCF-parameter for more details.

To process Xerox resources without using the communicator, the OutFormat should be set to LF and the OutPath variable should be used to define were the resource files/sub-directories should be created. When the OutSubcat variable is set to AUTO the sub-directories will be created in lower case.

2.4.3.1 Data types

Description of the supported Input-Filter data types and which keyword needs to be used:

Media	JCF-Parameter InFileType	Description
Xerox Resource	XRX_Resource	Xerox resources of type CMD, FNT, FRM, IMG, JSL, LGO are supported, others are ignored using the GCI environment. It is possible to load all Xerox resource types, see LCDS description (XDL/DJDE Reference Guide) for more details.
Xerox Print Data	XRX_DATA	Print data formatted using block / record structure Xerox print data (LCDS)

2.4.3.2 Xerox Floppy media

The Xerox floppy media are only supported for Xerox resource floppies 5.25" formatted for 720 Kbytes or 1.2 Mbytes and 3.5" floppies formatted for 1.2 Mbytes.

Media	JCF_Parameter DeviceName	Description
3.5" Floppy	35_FLOPPY_XRX	Xerox resource floppy, only 1.2 Mb format supported
5.25" Floppy	FLOPPY	Xerox resource floppy, 720 Kbyte and 1.2 Mb format supported

2.4.3.3 HD media

Xerox Resource

Xerox Resources of type CMD, FNT, FRM, IMG, JSL, LGO can be read from hard disk, other types are ignored. These files must start with the 128 byte Xerox LPS header! Use JDE TAPE05 for these files.

Xerox Print Data

Xerox Print Data is supported for three different data types. The xpar/JDE parameters are only partly supported depending on the data type. See also chapter 'JDE parameter'.

Also the new feature of the Input-Filter to make data traces can be used to see how the program interprets the data (Level 4, Flags 0xFFFFFFFa. See chapters 'Writing a trace file', 'Special trace flags' as well as 'Data trace flags' for more information.

The following data types are supported:

- Burroughs Large files with fixed blocked or variable record length, specify HOST=B6700 and when needed the block size (BLOCK=<size>).
- Block (optional) and Record format (BarrPC). This should be used when data which is not blocked is used. Also the JCF-Keywords ByteOrder and SkipBytes are only supported when HOST=BARRPC.
- Fixed blocked data with special images (HOST=XSD)
- Fixed blocked data
- Variable blocked data
- Variable record length.

Burroughs large files

Burroughs Large files can be read if they have a fixed block format or when they have only a variable records structure. The format which must be specified in the JDE using the VOLUME HOST=6700 and when a block structure exists the BLOCK LENGTH=<block size>. The other JDE keywords are ignored.

Block and Record format

This refers to print data from hard disk with block (optional) and record format similar as the data written on tape. When data is blocked and the host system is known, specify the HOST=<host name> and set the block and record parameters to conform to the data. See example 3 for more details.

If the host system is unknown, if bytes have to be skipped at the beginning of the data or if no block structure exist, it should be read using the VOLUME HOST=BARRPC settings. The format must be specified in the JDE using the VOLUME HOST=BARRPC, LABEL=NONE with the correct values for block (if used) and record length fields. If no block structure is found in the data BLOCK LTHFLD=0 must be specified. The input filter will then read each record as a separate block.

Some header files have additional "header bytes" at the beginning of the file which can be discarded. Use the JCF Action parameter SkipBytes to specify the number of bytes which should be skipped. This is supported only, if VOLUME HOST=BARRPC.

To specify the length field byte order the JCF Action parameter ByteOrder must be used. Little Endian byte ordering is used on the Intel CPUs, PDP-11 and VAX family. Big Endian ordering is used on the IBM 370 family, PDP-10, Motorola CPUs and

several RISC CPUs. Some RISC CPUs like the PowerPC and the SPARC can be switched between Little and Big Endian byte ordering!! Use the ByteOrder JCF keyword to define Big Endian byte ordering.

Supported only if VOLUME HOST=BARRPC.

See also examples 1 and 2 for more details.

Fixed blocked data with special images

When the data has fixed length blocks (where length is 512), variable length records, and images where the length field is missing, it must be read with HOST=XSD. Otherwise see the chapter 'Fixed block length'. See also example 5 for more information about this special data type.

- VOLUMEHOST=XSD, LABEL=NONE
- BLOCKLENGTH=<x>(x <= 32760 && x = n * 512, best performance with 32256)
- RECORD LENGTH=512, STRUCTURE=F

See example 5 for more details.

Fixed blocked data

- The data has fix length blocks with records. All blocks and record have the same structure, otherwise see the chapter 'Fixed block data with special images'. This can be read using following parameter:
- VOLUMEHOST=<Host-Type>, LABEL=NONE
- BLOCKLENGTH=<x>(x <= 32760)
- RECORD LENGTH=<y>, STRUCTURE=F (y <= block length)

See example 6 for more details.

Variable record length with delimiter and no length field

Print data from hard disk with a variable record length can be read using the RECORD CONSTANT parameter. For UNIX formatted ASCII text the parameter CONSTANT should be set to X '0A' ! When DOS formatted ASCII text is used, the parameter CONSTANT should be set to X '0D0A'. These are just 2 examples, up to 4 bytes can be specified. The following set of parameters also must be specified:

- VOLUME:HOST=UNDEF, LABEL=NONE, CODE=ASCII
- RECORD:STRUCTURE=U, CONSTANT=X'0A' (or CONSTANT=X'0D0A')

Be careful when using this data format in following cases:

- Data is transferred over the network - During (ASCII) data transfer the delimiter can be modified, therefore use the binary transfer mode.
- Inline resources exist in the data - These use the above described delimiter which can cause an unusable print file.

See example 4 for more details.

Examples

When the value from JDE-Parameter is the same as the default value, these are not specified in the examples. In the description however these are marked as being the default value. See also chapter 'JDE parameter'.

Note: Be aware that this section describes only the Input-Filter side of the problem. Look also in the X-Filter chapter for the parameters and settings this program needs.

Example 1: No block structure, big endian byte order

When the HD input data is as follows (the rectangles are the record length fields):

```
00000000 00 01 8B 00 02 01 40 00 4F 01 00 5C C3 E2 E2 5C  ..A... .|..*CSS*
00000010 40 C6 D6 D9 D4 E2 7E D5 D6 D5 C5 6B C2 C6 D6 D9   FORMS=NONE,BFOR
00000020 D4 7E 4D D5 D6 D5 C5 5D 6B D1 C4 C5 7E D1 F2 6B M=(NONE),JDE=J2,
00000030 D1 C4 D3 7E E3 C5 E2 E3 6B C6 D6 D5 E3 C9 D5 C4 JDL=TEST,FONTIND
00000040 C5 E7 7E 4D F0 6B E9 C5 D9 D6 6B F7 5D 6B C6 C5 EX=(0,ZERO,7),FE
00000050 C5 C4 7E C1 E4 E7 6B 5E 00 68 01 00 5C C3 E2 E2 ED=AUX,;....*CSS
00000060 5C 40 C6 D6 D5 E3 E2 7E 4D 4D E9 F1 E4 D4 D9 D7 * FONTS=((Z1UMRP
00000070 6B F1 F0 40 C4 D6 E3 E2 5D 6B C2 C1 D9 D9 C5 D7 ,10 DOTS),BARREP
00000080 6B C3 E2 C6 D3 F0 F3 6B C3 E2 C6 D7 F0 F3 6B C3 ,CSFL03,CSFP03,C
00000090 E2 F7 F0 F3 F0 6B C3 E2 F6 F0 F3 F0 6B C3 E2 F5 S7030,CS6030,CS5
000000A0 F0 F3 F0 6B C3 E2 F4 F0 F3 F0 6B C3 E2 F3 F0 F3 030,CS4030,CS303
000000B0 F0 6B C3 E2 F2 F0 F3 F0 6B C3 E2 F1 F0 F3 F0 5D 0,CS2030,CS1030)
000000C0 6B 5E 00 0C 01 00 5C C3 E2 E2 5C 40 C5 D5 C4 5E ,;....*CSS* END;
```

The data can be read by using following parameters:

JCF-File:	ByteOrder	big endian
	SkipBytes	0
JDE-File:	VOLUME	HOST=BARRPC, LABEL=NONE
	RECORD	ADJUST=2, LENGTH=182, LTHFLD=2, PREAMBLE=2, STRUCTURE=V

No block structure exist in data (BLOCK LTHFLD=0 =>DEFAULT). The record length field starts directly at the beginning of the record (RECORD OFFSET=0 =>DEFAULT) and uses 2 bytes (RECORD LTHFLD=2) which don't have to be exchanged in position (ByteOrder Big Endian). The length field is not part of the record

user data so skip this (RECORD PREAMBLE=2). The record length does not include the length field (RECORD ADJUST=2). The records have variable length without block structure (RECORD STRUCTURE=V).

The data is then extracted as follows:

No	Start position	Record length	Record Preamble	User data
1	0x0000	0x0001	0001	8B
2	0x0003	0x0002	0002	0140
3	0x0007	0x004F	004F	0100 5CC3 E2E2 5C40 C6D6 D9D4 E27E D5 D6D5 C56B C2C6 D6D9 D47E 4DD5 D6D5 C55D 6BD1 C4C5 7ED1 F26B D1C4 D37E E3C5 E2E3 6BC6 D6D5 E3C9 D5C4 C5E7 7E4D F06B E9C5 D9D6 6BF7 5D6B C6C5 C5C4 7EC1 E4E7 6B5E
4	0x0058	0x0068	0068	0100 5CC3 E2E2 5C40 C6D6 D5E3 E27E 4D4D E9F1 E4D4 D9D7 6BF1 F040 C4D6 E3E2 5D6B C2C1 D9D9 C5D7 6BC3 E2C6 D3F0 F36B C3E2 C6D7 F0F3 6BC3 E2F7 F0F3 F06B C3E2 F6F0 F3F0 6BC3 E2F5 F0F3 F06B C3E2 F4F0 F3F0 6BC3 E2F3 F0F3 F06B C3E2 F2F0 F3F0 6BC3 E2F1 F0F3 F05D 6B5E
5	0x00C2	0x000C	000C	0100 5CC3 E2E2 5C40 C5D5 C45E
6	0x00D0			

Example 2: Byte order Little endian with record pre- and postamble (BARRPC)

When the HD input data is as follows (the rectangles are the record length fields):

```
00000000 76 1A FF FF 0B 00 01 40 13 13 13 13 13 13 13 13  ....... ........
00000010 01 0B 00 20 00 01 40 5B C4 D1 C4 C5 5B 40 40 40  ...... $DJDE$
00000020 D1 C4 D3 7E F1 F0 F6 C1 F0 F1 6B D1 C4 C5 7E F0  JDL=106A01,JDE=0
00000030 F4 6B 5E 40 01 20 00 6F 00 01 40 5B C4 D1 C4 C5  4,; ...?.. $DJDE
00000040 5B 40 40 40 C6 D6 D5 E3 E2 7E 4D E9 F0 F4 D5 F6  $   FONTS=(Z04N6
00000050 D7 6B D4 C2 F0 F2 F1 D7 6B E9 F0 F5 C6 F5 D7 6B  P,MB021P,Z05F5P,
00000060 E9 F0 F5 C6 F7 D7 6B E9 F0 F5 C6 F6 D7 6B E9 F0  Z05F7P,Z05F6P,Z0
00000070 F5 C6 F4 D7 6B F6 F0 F2 F6 C4 D7 6B F6 F2 F8 F4  5F4P,6026DP,6284
00000080 D8 D7 6B E9 F0 F4 D5 C3 D7 6B F6 F0 F2 F6 C1 D7  QP,Z04NCP,6026AP
00000090 6B F6 F2 F8 F4 D4 D7 6B F6 F0 F2 F5 E8 D7 6B C8  ,6284MP,6025YP,H
000000A0 F1 F0 F0 D5 D7 6B 40 01 6F 00 71 00 01 40 5B C4  100NP, .?.... $D
000000B0 D1 C4 C5 5B 40 40 40 E9 F0 F5 C6 F2 D7 6B E9 F0  JDE$   Z05F2P,Z0
000000C0 C4 E6 D4 D7 6B E9 F0 F4 D5 C1 D7 6B E9 F0 F5 C6  DWMP,Z04NAP,Z05F
000000D0 F1 D7 6B E9 F0 F4 D5 F9 D7 6B F6 F0 F2 F5 D8 D7  1P,Z04N9P,6025QP
000000E0 6B E9 F0 C6 C7 F4 D7 6B E9 F0 F6 F6 E2 D7 6B E9  ,Z0FG4P,Z066SP,Z
000000F0 F0 F5 C6 F0 D7 6B C6 D9 F4 F1 F2 D7 6B E9 F0 F4  05F0P,FR412P,Z04
00000100 D5 F8 D7 6B E9 F0 F5 C5 E9 D7 6B E9 F0 F5 C5 E8  N8P,Z05EZP,Z05EY
00000110 D7 6B C6 D6 D9 D4 E2 E7 5D 6B 5E 40 01 71 00 2E  P,FORMSX),; ....
00000120 00 01 40 5B C4 D1 C4 C5 5B 40 40 40 D6 E3 C5 E7  .. $DJDE$   OTEX
00000130 E3 7E 7D C3 D6 D4 D7 E4 E2 C5 E3 40 D1 D6 C2 40  T='COMPUSET JOB
00000140 F0 F4 6B F1 F0 F6 C1 F0 F1 7D 6B 40 5E 40 01 2E  04,106A01', ; ..
00000150 00 19 00 01 40 5B C4 D1 C4 C5 5B 40 40 40 C4 E4  .... $DJDE$   DU
00000160 D7 D3 C5 E7 7E E8 C5 E2 6B 5E 40 01 19 00 1C 00  PLEX=YES,; .....
00000170 89 40 5B C4 D1 C4 C5 5B 40 40 40 C6 C5 C5 C4 7E  i $DJDE$   FEED=
00000180 E3 D9 C1 E8 F2 6B C5 D5 C4 5E 40 01 1C 00 0D 00  TRAY2,END; .....
```

The data can be read by using following parameters:

JCF-File:	ByteOrder	Little Endian
	SkipBytes	4
JDE-File:	VOLUME	HOST=BARRPC, LABEL=NONE
	RECORD	ADJUST=4, LENGTH=182, LTHFLD=2, POSTAMBLE=2, PREAMBLE=2, STRUCTURE=V

No block structure exist in data (BLOCK LTHFLD=0 =>DEFAULT). The first 4 bytes needs to be skipped (SkipBytes 4). Then the record length field follows (RECORD OFFSET=0 =>DEFAULT) and uses 2 bytes (RECORD LTHFLD=2), which have to be exchanged in position (ByteOrder Little Endian). The length field is not part of the record user data so skip this (RECORD PREAMBLE=2). At the end of the record the record length field is repeated (RECORD POSTAMBLE=2). Both record pre- and post-amble are not included in the record length (RECORD ADJUST=PREAMBLE+POSTAMBLE=4). The records have variable length without block structure (RECORD STRUCTURE=V). These parameters are used for the BARRPC data structure.

The data is then extracted as follows:

No	Start Position	(Record) Length	Record pre-amble	User data	Record post-amble
	0x0000	0x0004		761AFFFF (Skipped bytes)	
1	0x0004	0x000B	0B00	0140 1313131313131313 01	0B00
2	0x0013	0x0020	2000	01405B C4D1C4C55B404040 D1C4D37EF1F0F6C1 F0F16BD1C4C57EF0 F46B5E4001	2000
3	0x0037	0x006F	6F00	0001405BC4D1C4C5 5B404040C6D6D5E3 E27E4DE9F0F4D5F6 D76BD4C2F0F2F1D7 6BE9F0F5C6F5D76B E9F0F5C6F7D76BE9 F0F5C6F6D76BE9F0 F5C6F4D76BF6F0F2 F6C4D76BF6F2F8F4 D8D76BE9F0F4D5C3 D76BF6F0F2F6C1D7 6BF6F2F8F4D4D76B F6F0F2F5E8D76BC8 F1F0F0D5D76B4001	6F00
4	0x00AA	0x0071	7100	0001405BC4 D1C4C55B404040E9 F0F5C6F2D76BE9F0 C4E6D4D76BE9F0F4 D5C1D76BE9F0F5C6 F1D76BE9F0F4D5F9 D76BF6F0F2F5D8D7 6BE9F0C6C7F4D76B E9F0F6F6E2D76BE9 F0F5C6F0D76BC6D9 F4F1F2D76BE9F0F4 D5F8D76BE9F0F5C5 E9D76BE9F0F5C5E8 D76BC6D6D9D4E2E7 5D6B5E4001	7100
5	0x011F	0x002E	2E00	01405BC4D1C4C5 5B404040D6E3C5E7 E37E7DC3D6D4D7E4 E2C5E340D1D6C240 F0F46BF1F0F6C1F0 F17D6B405E4001	2E00
6	0x0151	0x0019	1900	01405BC4D1 C4C55B404040C4E4 D7D3C5E77EE8C5E2 6B5E4001	1900
7	0x016E	0x001C	1C00	89405BC4D1C4C55B 404040C6C5C5C47E E3D9C1E8F26BC5D5 C45E4001	1C00
8	0x018E	0x000D	0D00	..	

Example 3: Variable blocked data

When the HD input data is as follows (the rectangles are the block or record length fields):

```
00000000 00 00 26 D8 26 D4 00 00 00 24 00 00 2B 24 44 4A  ..&Ø&Ô...$..+$DJ
00000010 44 45 24 20 20 20 4A 44 4C 3D 58 58 58 58 58 58  DE$   JDL=XXXXXX
00000020 2C 4A 44 45 3D 44 46 4C 54 2C 3B 20 00 96 00 00  ,JDE=DFLT,;  .▌..
00000030 2B 24 44 4A 44 45 24 20 20 20 46 4F 4E 54 53 3D  +$DJDE$   FONTS=
00000040 28 55 4E 31 30 34 42 2C 44 42 31 38 4E 50 2C 44  (UN104B,DB18NP,D
00000050 42 31 34 4E 50 2C 48 45 31 32 42 50 2C 48 45 31  B14NP,HE12BP,HE1
00000060 31 42 50 2C 48 45 31 30 42 50 2C 48 45 31 30 4E  1BP,HE10BP,HE10N
00000070 50 2C 44 42 30 38 57 50 2C 48 45 30 39 42 50 2C  P,DB08WP,HE09BP,
00000080 48 45 30 39 4E 50 2C 48 45 30 38 42 50 2C 48 45  HE09NP,HE08BP,HE
00000090 30 38 4E 50 2C 48 45 30 37 42 50 2C 48 45 30 37  08NP,HE07BP,HE07
000000A0 4E 50 2C 48 45 30 36 4E 50 2C 50 30 38 4F 41 41  NP,HE06NP,P08OAA
000000B0 2C 4D 42 30 34 35 50 2C 46 4F 52 4D 53 58 29 2C  ,MB045P,FORMSX),
000000C0 3B 20 00 32 00 00 2B 24 44 4A 44 45 24 20 20 20  ;  .2..+$DJDE$
000000D0 4F 54 45 58 54 3D 27 58 58 58 58 58 58 58 58 20  OTEXT='XXXXXXXX
000000E0 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 27  XXXXXXXXXXXXXXX'
000000F0 2C 20 3B 20 00 26 00 00 2B 24 44 4A 44 45 24 20  ,  ;  .&..+$DJDE$
00000100 20 20 44 55 50 4C 45 58 3D 4E 4F 2C 43 4F 4C 4C    DUPLEX=NO,COLL
00000110 41 54 45 3D 59 45 53 2C 3B 20 00 13 00 00 2B 24  ATE=YES,;  ....+$
00000120 44 4A 44 45 24 20 20 20 45 4E 44 3B 20 00 10 00  DJDE$   END;  ...
```

Data continued just before first block ends and second block starts:

```
00002610 02 41 32 31 00 24 00 00 2B 58 58 58 58 58 58 58  .A21.$..+XXXXXXX
00002620 58 58 58 58 58 06 93 01 58 58 58 58 06 0E 02 58  XXXXX.▌.XXXX...X
00002630 58 58 06 58 02 58 58 58 00 40 00 00 2B 06 17 01  XX.X.XXX.@..+...
00002640 04 10 0A 00 06 58 58 58 58 58 58 58 58 58 58 58  .....XXXXXXXXXX
00002650 06 58 02 58 58 06 8A 02 58 58 58 58 31 06 58 05  .X.XX.▌.XXXX1.X.
00002660 04 B7 0A 58 58 58 58 06 58 06 58 58 58 58 58 58  .·.XXXX.X.XXXXXX
00002670 06 1B 07 58 58 58 58 58 00 1E 00 00 2B 06 58 05  .▌.XXXXX.▌..+.X.
00002680 04 8D 0A 00 06 58 58 58 2E 06 58 58 58 58 58 06  .▌...XXX..XXXXX.
00002690 58 06 58 58 58 58 58 00 27 00 00 2B 06 58 05 04  X.XXXX.'..+.X..X
000026A0 0A 00 06 58 58 58 58 58 58 58 2C 06 58 06 58 58  ...XXXXXXX,.X.XX
000026B0 06 05 07 58 58 58 58 58 58 58 58 58 58 00 1B 00  ...XXXXXXXXXX.▌.
000026C0 00 2B 06 58 58 04 58 09 00 10 58 58 58 58 58 58  .+.XX.X...XXXXXX
000026D0 58 58 58 58 58 58 58 58 00 00 26 D4 00 00 26 FC  XXXXXXXX..&Ô..&ü
000026E0 26 F8 00 00 00 93 00 00 2B 06 2C 02 04 58 09 00  &ø...▌..+.,..X..
000026F0 0A 58 06 58 02 58 58 58 58 06 90 02 58 58 58 06  .X.X.XXXX.▌.XXX.
00002700 58 02 58 58 06 58 02 58 58 2E 58 58 06 58 05 58  X.XX.X.XX.XX.X.X
00002710 58 58 58 06 8E 03 58 58 06 58 03 58 58 58 58 58  XXX.▌.XX.X.XXXXX
00002720 58 58 06 3B 04 58 58 05 58 04 58 58 58 06 A6 04  XX.;.XX.X.XXX.¦.
00002730 58 58 58 58 58 58 06 27 05 58 58 58 06 58 58 58  XXXXXX.'.XXX.XXX
00002740 58 58 58 58 58 58 58 06 58 05 58 58 58 58 06 58  XXXXXXX.X.XXXX.X
00002750 58 58 06 60 06 58 58 58 06 58 58 58 58 06 58 06  XX.`.XXX.XXXX.X.
00002760 58 58 58 06 07 07 58 58 58 58 58 58 58 06 85 07  XXX...XXXXXXX.▌.
00002770 58 58 58 58 58 58 2E 00 58 00 00 2B 06 0D 04 04  XXXXXX..X..+....
00002780 19 09 00 07 58 06 58 04 00 0E 58 58 58 58 58 58  ....X.X...XXXXXX
00002790 06 91 05 58 58 58 58 58 04 58 58 58 58 58 58 06  .'.XXXXX.XXXXXX.
000027A0 58 58 58 58 58 58 06 58 58 58 58 58 58 58 58 58  XXXXXX.XXXXXXXXX
000027B0 06 03 06 00 07 58 06 58 08 00 0E 58 58 58 58 58  .....X.X...XXXXX
000027C0 58 58 06 58 09 58 06 58 09 58 58 58 58 58 58 00  XX.X.X.X.XXXXXX.
000027D0 59 00 00 2B 06 60 00 04 11 09 00 11 09 1A 1A 1A  Y..+.`.........
```

The data can be read by using following parameters:

JCF-File: ByteOrder and SkipBytes are not interpreted since HOST=IBMOS is specified.

JDE-File:

VOLUME	HOST=IBMOS, LABEL=NONE,
BLOCK	ADJUST=4, LENGTH=12000, LTHFLD=2, OFFSET=2, POSTAMBLE=4, PREAMBLE=8
RECORD	LENGTH=11996, LTHFLD=2, PREAMBLE=4, STRUCTURE=VB

Alternative the following parameters will work too (NOT used here):

JCF-File: ByteOrder and SkipBytes are not interpreted since HOST=IBMOS.

JDE-File:

VOLUME	HOST=IBMOS, LABEL=NONE,
BLOCK	ADJUST=8, LENGTH=12000, LTHFLD=2, OFFSET=4, POSTAMBLE=4, PREAMBLE=8
RECORD	LENGTH=11996, LTHFLD=2, PREAMBLE=4, STRUCTURE=VB

The value of the host must be the name of host system, where the data is generated (VOLUME HOST=IBMOS).

The block length exists twice in the data (BLOCK ADJUST=4, BLOCK OFFSET=2 or BLOCK ADJUST=8, BLOCK OFFSET=4). The block preamble is 8 bytes (BLOCK PREAMBLE=8) and at the end of the block the postamble (BLOCK POSTAMBLE=4). The block adjust depends, if the block length fields are included in the specified block length.

The record length field is included in the specified record length (RECORD ADJUST=0 => DEFAULT) and the length field starts with no offset (RECORD OFFSET=0 => DEFAULT). The first 4 bytes, which is the record length field, are not part of the record user data (RECORD PREAMBLE=4). The records have variable length with block structure (RECORD STRUCTURE=VB).

The data is then extracted as follows:

No	Position	Type	Length	Preamble	User data	Postamble
B1	0x0000	Block	0x26D8	000026D826D40000		
1	0x0008	Record	0x0024	00240000	2B24444A 4445242020204A44 4C3D585858585858 2C4A44453D44464C 542C3B20	
2	0x002C	Record	0x0096	00960000	2B24444A44452420 2020464F4E54533D 28554E3130344422C 444231384E502C44 4231344E502C4845 313242502C484531 3142502C48453130 42502C484531304E 502C444230385750 2C4845303942502C 484530394E502C48 45303842502C4845 30384E502C484530 3742502C48453037 4E502C4845303 64E 502C5030384F4141 2C4D42303435502C 464F524D5358292C 3B20	
3	0x00C2	Record	0x0032	00320000	2B24 444A444524202020 4F544558543D2758 5858585858585858 5858585858585858 5858585858585827 2C203B20	
4	0x00F4	Record	0x0026	00260000	2B24444A44452420 20204455504C4558 3D4E4F2C434F4C4C 4154453D5945532C 3B20	
5	0x011A	Record	0x0013	00130000	2B24 444A444524202020 454E443B20	
6	0x012D	Record	0x0010	00100000		

Now continued just before the first data block ends and the second block starts:

No	Position	Type	Length	Preamble	User data	Postamble
n+1	0x2610	Record	0x0024	00240000	2B58585858585858 5858585858069301 5858585060E0241 5858065802585858	
n+2	0x2638	Record	0x0040	00400000	2B061701 04100A0006585858 5858585858585858 0658025858068A02 5858585831065805 04B70A5858585806 5806585858585858 061B075858585858	
n+3	0x2678	Record	0x001E	001E0000	2B065805 048D0A0006585858 2E06585858585806 5806585858585858	
n+4	0x2696	Record	0x0027	0027 0000	2B0658050458 0A00065858585858 58582C0658065858 0605075858585858 5858585858	
n+5	0x26BD	Record	0x001B	001B0000	2B0658580458090010585858585858 5858585858585858	
B1	0x26D8	Block				000026D4
B2	0x26DC	Block	0x26FC	000026FC26F80000		
n+6	0x26E4	Record	0x0093	00930000	2B062C0204580900 0A58065802585858 5806900258585806 5802585806580258 582E585806580558 585858068E035858 0658035858585858 5858063B04585805 5804585858 06A604 5858585858580627 0558585806585858 5858585858585806 5805585858580658 5858066006585858 0658585858065806 5858580607075858 5858585858068507 5858585858582E	
n+7	0x2777	Record	0x0058	00580000	2B060D0404 1909000758065804 000E585858585858 0691055858585858 0458585858585806 5858585858580658 5858585858585858 0603060007580658 08000E5858585858 5858065809580658 09585858585858	
n+8	0x27CF	Record	0x0059	00590000	2B06600004 11090011091A1A1A	

Example 4: Using record parameter constant

When the HD input data is as follows (the rectangles are the record delimiters):

```
00000000 31 5F 44 4A 44 45 5F 00 4A 44 45 3D 58 58 58 58 1_DJDE_.JDE=XXXX
00000010 58 2C 3B 20 20 20 20 20 20 20 20 20 20 20 20 20 X,;
00000020 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000030 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000040 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000050 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000060 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000070 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000080 20 20 20 20 20 20 0D 0A 20 5F 44 4A 44 45 5F 00        .. _DJDE_.
00000090 4A 44 4C 3D 44 46 41 55 4C 54 2C 3B 20 20 20 20 JDL=DFAULT,;
000000A0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000B0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000C0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000D0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000F0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000100 20 20 20 20 20 20 20 20 20 20 20 20 20 0D 0A 20                .
00000110 5F 44 4A 44 45 5F 00 44 55 50 4C 45 58 3D 4E 4F _DJDE_.DUPLEX=NO
00000120 2C 3B 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ,;
00000130 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000140 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000150 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000160 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000170 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000180 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000190 20 20 20 20 0D 0A 20 5F 44 4A 44 45 5F 00 43 4F      .. _DJDE_.CO
000001A0 50 49 45 53 3D 30 30 30 30 31 2C 3B 20 20 20 20 PIES=00001,;
```

The data can be read by using following parameters:

JCF-File: ByteOrder and SkipBytes are not interpreted since HOST=UNDEF is specified.

JDE-File:

	VOLUME	HOST=UNDEF, LABEL=NONE,
	BLOCK	LENGTH=32768
	RECORD	CONSTANT=X'0D0A', LENGTH=133, STRUCTURE=U

No host system is known (VOLUME HOST=UNDEF).

No block structure exists (BLOCK LTHFLD=0 => DEFAULT).

Also no record length field exists (RECORD LTHFLD=0 => DEFAULT), but record delimiter exists (RECORD CONSTANT=X'0D0A'). The delimiters themselves are automatically defined as record postamble, so this keyword is not needed. To enable the use of the record delimiter the structure must be defined as undefined (RECORD STRUCTURE=U).

The data is then extracted as follows:

No	Start Position	(Record) Length	Record user data	Record postam-ble
	0x0000	0x0087	315F444A44455F00 4A44453D58585858 582C3B2020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020	0D0A
	0x0088	0x0087	20 5F444A44455F004A 444C3D444641554C 542C3B2020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 20202020	0D0A
	0x010E	0x0087	205F 444A44455F004455 504C45583D4E4F2C 3B2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 2020202020202020 202020	0D0A
	0x0195	...	205F44 4A44455F00434F50	

Example 5: Using HOST=XSD using fixed records

The data has fixed blocks and records with length field. The images, which are inside the data, however don't have these records length fields. Since the Input-Filter does only check the data structure and not interpret the data itself it can't read the data using this approach. To read the data HOST=XSD must be used and read with fixed block length.

The data can be recognized while the following sequences exist:

0x2B64	IMAGE=(<Name>,<Position>,<Position>,D)
0x2BAB	GRAPHIC=DUMMY

As also shown in the data below:

```
00002B60  45 20 30 30 49 4D 41 47 45 3D 28 41 30 31 2C 30  E 00IMAGE=(A01,0
00002B70  34 2E 30 30 49 4E 2C 30 30 2E 32 32 49 4E 2C 44  4.00IN,00.22IN,D
00002B80  29 2C 45 4E 44 3B 01 00 15 00 00 42 32 4E 20 20  ),END;.....B2N
00002B90  20 20 20 20 30 30 20 20 20 20 20 01 00 30 00 00      00      ..0..
00002BA0  42 33 4E 20 44 4A 44 45 20 30 30 47 52 41 50 48  B3N DJDE 00GRAPH
00002BB0  49 43 3D 44 55 4D 4D 59 2C 45 4E 44 3B 20 20 20  IC=DUMMY,END;
```

The data can be read by using following parameters:

JCF-File: ByteOrder and SkipBytes are not interpreted since HOST=XSD is specified.

JDE-File:

VOLUME	HOST=XSD, LABEL=NONE,
BLOCK	LENGTH=32256
RECORD	LENGTH=512, STRUCTURE=F

The block length must be set to a multiple of the record size (BLOCK LENGTH=32256), which is in this case 512 and smaller then 32768. Be ware that per block / record 8 additional bytes are written and the maximum block size, including these additional bytes, is 32 Kbyte (32768). This mean the term for the maximum specified block length is as follows: Record_size * n + ((n + 1) * 8) < 32768 (32 Kbyte), were n = max. records in block. The data is passed through the Input-Filter by specifying the fixed record size of 512 bytes (RECORD LENGTH=512, RECORD STRUCTURE=F).

```
00000000 00 50 00 00 42 32 4E 20 44 4A 44 45 20 30 30 4F .P..B2N DJDE 00O
00000010 54 45 58 54 3D 28 27 4E 45 57 20 50 41 50 45 52 TEXT=('NEW PAPER
00000020 20 54 59 50 45 20 4E 45 45 44 45 44 2E 20 20 4E  TYPE NEEDED.  N
00000030 45 57 20 50 41 50 45 52 3D 53 54 41 4E 44 41 52 EW PAPER=STANDAR
00000040 44 27 2C 57 41 49 54 29 20 2C 20 45 4E 44 3B 00 D',WAIT) , END;.
00000050 00 12 00 00 42 32 F1 20 20 20 20 00 00 00 00 01 ....B2ñ     .....
00000060 01 01 00 50 00 00 42 32 4E 20 44 4A 44 45 20 30 ...P..B2N DJDE 0
00000070 30 4F 54 45 58 54 3D 28 27 56 45 52 49 46 59 20 0OTEXT=('VERIFY
00000080 50 41 50 45 52 20 54 59 50 45 20 4E 4F 57 20 20 PAPER TYPE NOW
00000090 20 4E 45 57 20 50 41 50 45 52 3D 53 54 41 4E 44  NEW PAPER=STAND
000000A0 41 52 44 27 2C 57 41 49 54 29 20 2C 20 45 4E 44 ARD',WAIT) , END
000000B0 3B 00 00 12 00 00 42 32 F1 20 20 20 20 00 00 00 ;.....B2ñ    ...
000000C0 00 01 01 01 00 2C 00 00 42 32 F1 20 44 4A 44 45 .....,..B2ñ DJDE
000000D0 20 30 30 4A 44 45 3D 54 45 53 54 2C 45 4E 44 3B  00JDE=TEST,END;
000000E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000F0 00 15 00 00 42 32 4E 20 20 20 20 20 20 30 30 20 ....B2N      00
00000100 20 20 20 20 01 00 76 00 00 42 32 F1 20 44 4A 44     ..v..B2ñ DJD
00000110 45 20 30 30 46 4F 4E 54 53 3D 28 52 41 31 32 42 E 00FONTS=(RA12B
00000120 50 2C 52 42 31 32 42 50 2C 52 36 31 32 42 50 2C P,RB12BP,R612BP,
00000130 52 36 42 4F 42 50 2C 52 38 4F 42 44 50 2C 52 4B R6BOBP,R8OBDP,RK
00000140 31 36 42 50 2C 52 4B 31 38 42 50 2C 52 4B 32 38 16BP,RK18BP,RK28
00000150 42 50 2C 52 4B 32 42 45 50 2C 42 43 58 31 43 50 BP,RK2BEP,BCX1CP
00000160 2C 42 4F 58 33 30 50 2C 52 41 31 32 58 58 2C 42 ,BOX30P,RA12XX,B
00000170 43 30 31 43 50 29 2C 45 4E 44 3B 00 68 00 00 42 C01CP),END;.h..B
00000180 32 F1 20 20 20 20 20 20 30 30 00 03 04 60 09 06 2ñ      00...`..
00000190 2C 01 54 54 54 54 54 54 54 54 20 20 45 45 45 45 ,.TTTTTTTT  EEEE
000001A0 45 45 45 45 20 20 20 53 53 53 53 53 53 53 20 20 EEEE   SSSSSSS
000001B0 54 54 54 54 54 54 54 54 20 20 20 20 20 20 20 4A TTTTTTTT       J
000001C0 4A 20 20 20 20 4F 4F 4F 4F 4F 4F 20 20 20 42 42 J    OOOOOO   BB
000001D0 42 42 42 42 20 20 20 20 42 42 42 42 42 42 20 20 BBBB    BBBBBB
000001E0 20 20 01 00 00 00 00 00 00 00 00 00 00 00 00 00   ..............
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000200 00 68 00 00 42 32 4E 20 20 20 20 20 20 30 30 00 .h..B2N      00.
00000210 03 04 24 09 06 2C 01 54 54 54 54 54 54 54 54 20 ..$..,.TTTTTTTT
00000220 20 45 45 45 45 45 45 45 45 20 20 53 53 53 53 53  EEEEEEEE  SSSSS
00000230 53 53 53 20 20 54 54 54 54 54 54 54 54 20 20 20 SSS  TTTTTTTT
```

The data is then extracted as follows:

No	Start Position	(Record) Length	Record user data
1	0x0000	0x0200	0050000042324E2044...
2	0x0200	0x0200	0068000042324E2020...

Example 6: Fixed block structure, variable record structure

This example uses the same data file a shown in example 5. When this special processing because of the images is NOT needed the data can also be read with the following parameter:

JCF-File: ByteOrder Big Endian
SkipBytes 0

JDE-File: VOLUME HOST=BARRPC, LABEL=NONE
BLOCK LENGTH=512, ZERO=YES
RECORD LENGTH=500, LTHFLD=2, PREAMBLE=4, STRUCTURE=VB

The blocks have fixed length of 512 bytes (BLOCK LENGTH=512, LTHFLD=0 => DEFAULT). End of block is specified by a zero record length field (ZERO=YES). The record length field are 2 bytes (RECORD LTHFLD=2) and starts at the beginning of the record (OFFSET=0 => DEFAULT). The user data starts at the 4th byte of the record (PREAMBLE=4). The structure is variable blocked (STRUCTURE=VB).

```
00000000 00 50 00 00 42 32 4E 20 44 4A 44 45 20 30 30 4F .P..B2N DJDE 00O
00000010 54 45 58 54 3D 28 27 4E 45 57 20 50 41 50 45 52 TEXT=('NEW PAPER
00000020 20 54 59 50 45 20 4E 45 45 44 45 44 2E 20 20 4E  TYPE NEEDED.  N
00000030 45 57 20 50 41 50 45 52 3D 53 54 41 4E 44 41 52 EW PAPER=STANDAR
00000040 44 27 2C 57 41 49 54 29 20 2C 20 45 4E 44 3B 00 D',WAIT) , END;.
00000050 00 12 00 00 42 32 F1 20 20 20 20 00 00 00 00 01 ....B2ñ     .....
00000060 01 01 00 50 00 00 42 32 4E 20 44 4A 44 45 20 30 ...P..B2N DJDE 0
00000070 30 4F 54 45 58 54 3D 28 27 56 45 52 49 46 59 20 0OTEXT=('VERIFY
00000080 50 41 50 45 52 20 54 59 50 45 20 4E 4F 57 20 20 PAPER TYPE NOW
00000090 20 4E 45 57 20 50 41 50 45 52 3D 53 54 41 4E 44  NEW PAPER=STAND
000000A0 41 52 44 27 2C 57 41 49 54 29 20 2C 20 45 4E 44 ARD',WAIT) , END
000000B0 3B 00 00 12 00 00 42 32 F1 20 20 20 20 00 00 00 ;.....B2ñ     ...
000000C0 00 01 01 01 00 2C 00 00 42 32 F1 20 44 4A 44 45 .....,..B2ñ DJDE
000000D0 20 30 30 4A 44 45 3D 54 45 53 54 2C 45 4E 44 3B  00JDE=TEST,END;
000000E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000F0 00 15 00 00 42 32 4E 20 20 20 20 20 20 30 30 20 ....B2N      00
00000100 20 20 20 20 01 00 76 00 00 42 32 F1 20 44 4A 44     ..v..B2ñ DJD
00000110 45 20 30 30 46 4F 4E 54 53 3D 28 52 41 31 32 42 E 00FONTS=(RA12B
00000120 50 2C 52 42 31 32 42 50 2C 52 36 31 32 42 50 2C P,RB12BP,R612BP,
00000130 52 36 42 4F 42 50 2C 52 38 4F 42 44 50 2C 52 4B R6BOBP,R8OBDP,RK
00000140 31 36 42 50 2C 52 4B 31 38 42 50 2C 52 4B 32 38 16BP,RK18BP,RK28
00000150 42 50 2C 52 4B 32 42 45 50 2C 42 43 58 31 43 50 BP,RK2BEP,BCX1CP
00000160 2C 42 4F 58 33 30 50 2C 52 41 31 32 58 58 2C 42 ,BOX30P,RA12XX,B
00000170 43 30 31 43 50 29 2C 45 4E 44 3B 00 68 00 00 42 C01CP),END;.h..B
00000180 32 F1 20 20 20 20 20 20 30 30 00 03 04 60 09 06 2ñ      00...`..
00000190 2C 01 54 54 54 54 54 54 54 54 20 20 45 45 45 45 ,.TTTTTTTT  EEEE
000001A0 45 45 45 45 20 20 20 53 53 53 53 53 53 53 20 20 EEEE   SSSSSSS
000001B0 54 54 54 54 54 54 54 54 20 20 20 20 20 20 20 4A TTTTTTTT       J
000001C0 4A 20 20 20 20 4F 4F 4F 4F 4F 4F 20 20 20 42 42 J    OOOOOO   BB
000001D0 42 42 42 42 20 20 20 20 42 42 42 42 42 42 20 20 BBBB    BBBBBB
000001E0 20 20 01 00 00 00 00 00 00 00 00 00 00 00 00 00   ..............
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000200 00 68 00 00 42 32 4E 20 20 20 20 20 20 30 30 00 .h..B2N      00.
00000210 03 04 24 09 06 2C 01 54 54 54 54 54 54 54 54 20 ..$..,.TTTTTTTT
00000220 20 45 45 45 45 45 45 45 45 20 20 53 53 53 53 53  EEEEEEEE  SSSSS
00000230 53 53 53 20 20 54 54 54 54 54 54 54 54 20 20 20 SSS  TTTTTTTT
```

The data is then extracted as follows:

No	Position	Type	Length	Preamble	User data
B1	0x0000	Block	0x0200		
1	0x0000	Record	0x0050	00500000	42324e20 444a44452030304f 544558543d28274e 4557205041504552 2054595045204e45 454445442e20204e 4557205041504552 3d5354414e444152 44272c5741495429 202c20454e443b00
2	0x0050	Record	0x0012	00120000	4232f120 2020200000000001 0101
3	0x0062	Record	0x0050	00500000	4232 4e20444a44452030 304f544558543d28 2756455249465920 5041504552205459 5045204e4f572020 204e455720504150 45523d5354414e44 415244272c574149 5429202c20454e44 3b00
4	0x00B2	Record	0x0050	00120000	4232 f1202020200000000 00010101
5	0x00C4	Record	0x002C	002C0000	4232f120444a4445 2030304a44453d54 4553542c454e443b 2020202020202020 2020202020202020
6	0x00F0	Record	0x0015	00150000	42324e20 2020202020303020 2020202001
7	0x0105	Record	0x0076	00760000	4232f120444a44 45203030464f4e54 533d285241313242 502c524231324250 2c5236313242502c 5236424f42502c52 384f4244502c524b 313642502c524b31 3842502c524b3238 42502c524b324245 502c424358314350 2c424f583330502c 5241313258582c42 4330314350292c45 4e443b
8	0x017B	Record	0x0068	00680000	42 32f1202020202020 3030000304600906 2c01545454545454 5454202045454545 4545454520202053 5353535353532020 5454545454545454 202020202020204a 4a202020204f4f4f 4f4f4f2020204242 4242424220202020 4242424242422020 202001
(9)	0x01e3	Record	0x0000		
B2	0x0200	Block	0x0200		
9	0x0200	Record	0x0068	00680000	42324e20 2020202020303000 03042409062c0154 5454545454545420 2045454545454545 4520205353535353 5353532020545454 5454545454202020 ...

2.4.3.4 Tape media

Xerox Resource

The supported Xerox Resource tape media should be formatted with the HOST=UNDEF and LABEL=NONE settings.

Xerox Print Data

The Xerox Print Data tape media type are defined by the HOST/LABEL combination in the InputFilter parameter file.

The following combinations are at the moment supported:

Host	Label	Used when tape …
ANSI	ANSI	is generated to conform the American National Standards Institute (ANSI) standard.
ANSI	NONE	is an unlabeled tape with record and block structure. No Multi-volume support.
BS2000	ANSI	is generated to conform the ANSI standard on an BS2000 host.
BS2000	STANDARD	is generated to conform the ANSI standard on an BS2000 host.
BS2000	NONE	is an unlabeled tape with BS2000 host record/block structure. **No Multi-volume support.**
B2500, B2700, B3500, B3700, B4700	ANSI	is generated as a non-printer backup ANSI on an Medium Burroughs system
B2500, B2700, B3500, B3700, B4700	STANDARD	is generated as a printer backup on an Medium Burroughs system
BURLAR	STANDARD	is generated to conform the ANSI standard on an B6700 Burroughs large host.
IBMDOS	ANSI or UNDEF	is generated to conform ANSI standard.
IBMDOS	STANDARD	is similar to the ANSI standard with special IBM DOS/360 host features enabled.
IBMDOS	NONE	is an unlabeled tape with IBM DOS/360 host record/block structure. **No Multi-volume support.**

IBMOS	ANSI or UNDEF	is generated to conform the ANSI standard on an IBM OS/360 host.
IBMOS	STANDARD	is similar to the ANSI standard with special IBM OS/360 host features enabled.
IBMOS	NONE	is an unlabeled tape with IBM OS/360 host record/block structure. **No Multi-volume support.**
ICL2900	STANDARD	is similar to the ANSI standard on an ICL 2900 host.
NCR	STANDARD	is generated to conform the ANSI standard on an IBM OS/360 host.
NCR	NONE	is an unlabeled tape with NCR host record/block structure. **No Multi-volume support.**
UNDEF	NONE	is an unlabeled tape with undefined format. No Multi-volume support.
UNIVAC	ANSI or STANDARD	is similar to the ANSI standard created on an UNIVAC host.
UNIVAC	NONE	is an unlabeled tape with Univac host record/block structure. **No Multi-volume support.**
US70	STANDARD	is similar to the ANSI standard with special Univac Series 70 host features enabled
US70	NONE	is an unlabeled tape with Univac Series 70 host record/block structure. **No Multi-volume support.**

370-channel

Host	Label	Description
EXOTIC		May be used when data is received via /370-channel from BS2000 without so called IBR-BOX to specify that one certain special rule concerning linespacing is processed in the LCDS converter.
IBMONL		Used when data is received over 370-channel

2.4.4 OLDS-Module

Input Media/Data Type Combinations

A quick overview of the supported input media/data type combinations:

Media	Supported data type(s)	Description
HD	SPP (OLDS) Data SPP (OLDS) Resource	SPP printdata (OLDS) SPP resources (OLDS)
Cartridge, Tape, QIC or as image file	SPP (OLDS) Data SPP (OLDS) Resource	SPP printdata (OLDS) SPP resources (OLDS)

Data types

Description of the supported Input-Filter data types and which keyword needs to be used:

Media	JCF-Parameter InFileType	Description
SPP (OLDS) Data	SPP_Data	SPP print data (OLDS)
SPP (OLDS) Resource	SPP_Resource	OMSLIB resource data

Hard disk media

The following hard disk formats are supported:

Input-filter data type	JCF-Parameter JDE-Name	Restrictions/Requirements
SPP_Data	*	For JDE-Name see chapter: SPP Print Data
SPP_Resource	OMSLIB	OMSLIB, record length fields must be added to the input-filter input data stream

Tape media

SPP Resource

The following hard disk formats are supported:

JCF-Parameter JDE-Name	Host	Label	Description
OMSLIB	OLDS	STANDARD	Used to read OMSLIB libraries.

SPP Print Data

The SPP Print Data tape media type are defined by the Overlay name (= JCF-Parameter: JDE-Name) which then selects the correct section in the InputFilter parameter file (olds_params.if) with correct HOST/LABEL settings.

The following overlays are at the moment supported:

Overlay	Host	Label	Description
S1	ANSI	ANSI	Processes VOL1, HDR1, EOF1 and EOV1 labels. Labels use EBCDIC-Code.
S3	ANSI	ANSI	Processes VOL1, HDR1, EOF1 and EOV1 labels. Labels use ASCII-Code.
NO	IB-MOS	NONE	Unlabeled tape. **No Multi-volume support.**
LT	UN-DEF	UN-DEF	Unlabeled tape were first 'label' is a tapemark (TM). **No Multi-volume support.**
H1	H6000	STANDARD	Processes standard 80 or 120 (COBOL) character labels, Basic Tape Labels (BTL), partial BTL EOF and EOR labels. Labels use H6BCD-Code.
U1	UNI-VAC	ANSI	Processes VOL1, HDR1, EOF1 and EOV1 labels. Labels use ASCII-Code. First data part O50 in Fielddata coded.

2.4.5 Job Control File Parameter

The Job Control File (JCF) parameter is used to define a process chain. The Input-filter is the first part of the chain which reads the data from the hardware device. The data is then passed via the communicator (GCI) to the next process in the chain or written to hard disk. The JCF file is divided in several sections, the following parts are used by the Input-Filter:

JCF-Section	Sub-Section	Description
Header		General part read by all processes as well as communicator.
I-Filter		General Input-Filter parameter
I-Filter	Device	Specifies which data source (device) is used and its parameters.
I-Filter	Outputdevice	Specifies which output device is used and its parameters
I-Filter	SubcatInfo	Defines all used subcat types
I-Filter	Action	Defines action to perform together with all needed parameters.

2.4.5.1 Header section

The header section is mandatory for each Job Control File (JCF). It contains general used parameters as well as the setup of the process chain, specifying names and sequences of the dynamic processes. This part can be read by all dynamic processes as well as by the communicator.

Example:

```
Begin-Header
       Description PrintSPDS.JCF
       BufferSize  32
       Catalog     DEFAULT
       AccountId   12345678
       Begin-ProcessChain
          First I-Filter
          I-Filter SpoolJob
       End-ProcessChain
End-Header
```

Keyword	Variable Type	Description
AccountId	String	AccountId which is requested by the ui_manager from the account process.
BufferSize	Value	Defines the size of the write/read buffer in kilobytes (= 1024 bytes) for communication to/form the communicator (GCI). This value is fixed set to 32!!!!!
Catalog	String	Defines the catalog name which all processes use. All dynamic processes have to use this catalog name for input and output. Besides this catalog only the job catalog can be used for the JCF.
Description	String	Short description of the job, blanks are not allowed, max. 32 characters.
First	String	Defines the first process in the process chain which is most of the time the Input-Filter.

2.4.5.2 Input-Filter Section

Input-Filter section, which has general parameter and special Input-filter sections.

Example:

```
Begin-I-Filter
    Begin-ComInfo
#       …
    End-ComInfo
    TraceLevel          0
    TraceFlags          ffffffff8
    TraceMaxFileSize    0
    Begin-Device
#       …
    End-Device
    Begin-SubcatInfo
#       …
    End-SubcatInfo
    ActionCount         1
    Begin-Action
#       …
    End-Action
End-I-Filter
```

Keyword	Variable Type	Description
ActionCount	Value	Number of action sections which follow. Maximum action sections 10 are allowed.
DataTraceStart	Value (int/hex)	Data position from which data trace is written in detail. Default 0 which means start at the beginning.
DataTraceStop	Value (int/hex)	Data position from which data trace not written in detail anymore. Default 0xffffffff which means whole file.
TraceFlags	Value (hex.)	Contains the trace flags for a more exact differentiation of the traces in the Input-Filter. See chapter 'Trace level and flags'.
TraceLevel	Value	Defines the trace level. Allowed values are 0 (no traces) to 4 (maximum trace).
TraceMaxFile-Size	Value	Sets the maximum trace file size in bytes. When the file gets larger a warning is printed into the file and all following trace messages are NOT written. To disable this feature set the value to 0. Default value is 1 GB.

2.4.5.3 ComInfo Section

The ComInfo section contains information on each process for the communicator and has to be included in each dynamic process once. All following parameters must be defined

Example:

```
Begin-ComInfo
    ModuleName     ifilter
    ModuleVersion  01.01.18
    HostLink       local
    ProcessType    dynamic
    WaitType       send_finished
                   TimeoutStart1
                   TimeoutTime300
                   TraceLevel0
    TraceFlags     FF00000000000000
    TraceFile      /u/spool/diag/ifilter.trc
End-ComInfo
```

Keyword	Variable Type	Description
ModuleName	String	Name of the process (ifilter)
ModuleVersion	String	Defines the version of the program. It has to correspond to the version sent to the commnicator by the dynamic process after starting with send_init.
HostLink	String	
ProcessType	String	Has to be 'dynamic'.
WaitType	String	Specifies when the communicator has to start the process. The start is dependent of the predecessor process, which can submit either a 'data_ready' or a 'send_finished'. For the first process of a chain the parameter has no meaning, the process is started immediately.
TimeoutStart	Value	
TimeoutTime	Value	
TraceLevel	Value	Communicator interface trace (usertrace) level
TraceFlags	Value	Communcator interface trace flags
TraceFile	String	Path/filename of communicator trace file, in binary, needs to be interpreted with the 'usrtrac.exe'/'usrtrc' command.

2.4.5.4 Device Section

The device section contains information on the data source of the input-filter.

Example:

```
Begin-Device
    DeviceType     CARTRIDGE
    TapeRewind     REWIND
    MagazineType   SINGLE
    MinDiskspace   10000
End-Device
```

General device section

The JCF-Keywords described in the table below are used for all devices.

Keyword	Variable Type	Description
DeleteInput	Value	Delete Input file: 0 - File is NOT deleted (default), 1 - File is deleted. Only valid for HD & 3.5" floppies.
DeviceName	String	Device name. Normally these have following settings when the DeviceType has following settings: /370 /dev/ikgX (EISA, SCO Unix) /dev/ikpX (PCI, SCO Unix) /dev/i370/chanX/cua (PCI/EIA,Linux) hostname or ip-address (for receiving data from socket) 35_FLOPPY 35_FLOPPY_XRX Is ignored by Unix, dependent of floppy format CARTRIDGE, QIC, `/dev/nrStpX,` were X = 0…2 (SCO), or TAPE is ignored (Linux) FLOPPY Is ignored by Unix (5¼" floppy) HD <Not used>
DeviceType	String	Category of the data source. permitted entries are / 370, 35_FLOPPY, 35_FLOPPY_XRX, TAPE, CARTRIDGE, FLOPPY, HD.
ImageName	String	TAPE: Path to tape image data file created by tapman (DeviceType = CARTRIDGE, QIC or TAPE). Xerox Floppy: Path to image file created with dd (DeviceType = 35_FLOPPY_XRX or FLOPPY). This will force the program to read directly from the file instead of from the tape/floppy device. This parameter is only used in special cases.
MinDiskspace	Value	Minimum free disk space in kilobytes. If free space becomes smaller InputFilter will wait until more space is available (Default: 10 Mb).

/370 Channel device section

The JCF-Keywords described in the table below are used only for the /370-channel device.

Keyword	Variable Type	Description
ChanAddress	Hex	Specifies channel address, value between 0 - 255, or (when in the DeviceName the host-name/IP-address is set) the port number.
ChanBandID	Hex	Specifies band ID of the emulated printer. Default value is 0xf9f2f0f2 which is in EBCDIC 2029.
ChanBufSize	Value	Specifies channel buffer size and can be used in combination with ChanDEDelay. The delay is executed after the buffer is filled with data. Smaller buffer means more frequent delays resulting is less data throughput for input filter.
ChanChkPos	Value	Specifies byte position for length bytes and enables /370 data checking if value > 0
ChanCondCode	String	Specifies channel status when program exits. Possible options: OFFLINE - Channel is closed ONLINE - Channel is kept open
ChanDEDelay	Value	Delay time in milliseconds between the end of the input filter processing a received CCW and the DeviceEnd send. Set to value > 0 is recommended when channel is opened in daisy-chain mode and other processing are also using the channel.
ChanDumpMode	Value	Channel dump mode : 0 - Off, 1 = Read, 2 = Write (Only write CCw's),3 = Write (All CCw's) Output file is defined by 'InFile' keyword in Action Section
ChanEmulation	String	Only valid when DeviceType = /370. Specifies which printer type will be emulated on the channel. The following printer emulations are supported: 3203, 3211, 4245 and 6262
ChanSenseId	String	Sense-Id-Data for retruning on Sense-ID ccws: default for 3203 <empty> default for 3211 <empty> default for 4245 FF4245010000 default for 6262 FF4248010000

ChanStartTime	Value	Time in seconds before timeout event is generated when no data is received from the 370 channel. Disabled when set to 0, minimum value is 20 seconds
ChanStopTime	Value	Time in seconds before channel is closed when no data is received from the 370 channel. This value is taken when a SIGUSR1 (Signal No. 10) is sent to the I-Filter, the /370 channel is used and 'ChanTimeout' is set to 0. Default value is 20 seconds.
ChanTimeout	Value	Time in seconds after which a timeout event is generated when no data is received from the / 370 channel. Disabled when set to 0, minimum value is 20 seconds.
ChanTimeoutAction	String	Automatic channel timeout action response. Possible options: CLOSE_CHAN: Channel will be closed, program terminates. DIALOG: Dialog is shown where user has to selects action. (Default) SEND_REND: Record End command is send. Channel stays opened.
ChanTransMode	Value	Transfer mode. Possible options: BYTE_MUX, BLOCK_MUX, HIGHSPEED, DATASTREAM_150, DATASTREAM_225, DATASTREAM_300, DATA_STREAM_450
ChanTrcFile	String	Output file for 370 driver traces.
FCBIgnore	Value	When set (value = 0) online print-data will be checked with the FCB, otherwise (value > 0) it is not checked.
MaxWriteLen	Value	Maximum Channel record length. Setting a large value will negatively impact performance. Should always be smaller than the ChanBufSize value. Record Length (JSL) | MaxWriteLen | Used max. record length --- > 0 | -1 | Record length JSL <= 0 | -1 | Default value (1024) any | >= 0 | MaxWriteLen any | not defined | Default value (1024)

Tape device section

Keyword	Variable Type	Description
MagazineType	String	Only valid when DeviceType = CARTRIDGE. When a single cartridge drive is used this parameter should be SINGLE. MULTI is used for tape drives which can work sequentially through multiple magazines. JUKE is used for tape drives which individually addressable slots.
SCSIadapter	Value	SCSI host adapter number as Operating System counts them. The value of -1 specifies to check all adapters for device. Default value: -1
SCSIid	Value	SCSI device identifier as is set on the HW device itself. The value of -1 specifies to use the default SCSI id which is dependent of the specified DeviceType. Current defaults: CARTRIDGE: 3, QIC: 2, TAPE: 4. Default value: -1.
SCSIlun	Value	Reserved for future use.
TapeRewind	String	Tape command to be executed after input processing is finished. Possible options: NOREWIND, REWIND and UNLOAD. Default when parameter is not specified is NOREWIND.

Action Section

The JCF action section is to define the actual action which the InputFilter is supposed to do. Several actions can be defined after each other. Specific parameters can be defined for each action seperately.

General

Keyword	Variable Type	Description
ActionName	String	Program accepted actions. Possible values: MAKE_DIR : Make directory/tape listing POSITION : Set device on specific position READ_FILE : Read data from device
JDE-Name	String	Entry in parameter file to be used for this job.
ParamFile	String	Parameter file with VOLUME, BLOCK and RECORD parameters.
ParamSubcat	String	Sub catalog (sub-directory) were parameter file can be found.

Input data control

Keyword	Variable Type	Description
ByteOrder	String	Defines byte ordering from CPU/Host only supported for block / record length fields and HOST=BARRPC. Possible values: BigEndian: MSB LSB ordering (Default, IBM 370, PDP-10, Motorola, RISC) LittleEndian: LSB MSB ordering (INTEL, PDP-11, VAX)
InFile	String	Name of input file, wildcard '*' or a combination of character and wildcard. Wildcard can be used in the middle of characters, only wildcard is allowed. Also used when 370-channel dump is read or written for the dump output filename.
InFileFirst	Value	First file to print. Possible values: 0 - Prints from first file or all files (when InFileLast = 0) > 0 - Prints from file no. until InFileLast value
InFileLast	Value	Last report to print. Possible values: > 0 - Last file to print (InFileFirst <= InFileLast) 0 - Prints until last file (InFileFirst >= 0)
InFileType	Value	Input file format type. Possible values: AFP_Resource - AFP resource files AFPDS - AFP print data with linedata Pure_AFPDS - AFP print data SPP_Data - OLDS print data SPP_Resource - OLDS resource files XRX_Data - Xerox print data XRX_Resource - Xerox resource files
PositDir	Character	Positioning direction. Possible values: '+' - Position forward (Toward End of Tape/File) '-' - Position backward (Towards Beginning of Tape/File)
PositMedium		Positioning type. Possible values: BLOCK Positioning by specifying the number of blocks. DEVICE Device dependent positioning, for example by tapes using the tape marks (TM). FILE Specify the n-th file of a media.
PositUnit		Position unit. Possible values: BOT - Beginning of tape (PositMedium = DEVICE) value -
SkipBytes	Value	BARRPC only: Skip n-bytes at the beginning of the input file

Output data control

Keyword	Variable Type	Description
AddSequenceNo	Binary	Defines that file sequence number is written as first part of the output filename followed by a dot '.' and the input filename when data is read. When the directory listing command is used the file sequence number is added too, with a colon as separator. Is only active when AFP print data is processed and the 'OutFile' parameter is set to identical. Possible values: Yes - Add file sequence number to output filename/directory listing. No - Create output filename as usual, is default.
AfpSfTrim	Binary	Removes padding bytes after Structured field in AFP-Print data until next CR/LF. CAREFULL can cause data loss when CR/LF are not added after Structured Fields! Yes - Removes padding bytes (Default) No - Nothing changed.
DeleteOutput	Value	When error occurs output file is deleted. Possible values: 0 - Output file is never deleted (Default) 1 - Output file is deleted when error occurs
MergeFiles	String	Merge output files. Possible values: Yes - Merge all output files to one output file No - Write multiple output files
ModifyFilename	String	Output filename modifications. Possible values: LowerCase - Change output filename to lower case. UpperCase - Change output filename to upper case.
NextProcess	String	Next process which should be started including the full path, without the options needed to start the program (See NextOptions, OutType)
NextOptions	String	Option for the next process to be started, blanks are allowed (see NextProcess).
OutFile	String	Name of output file or keyword 'IDENTICAL' to have the output filename the same as the input filename

OutFormat	String	Output file format. Possible values: CRLF — Add CR/LF to records which do not end with these bytes. CRLF_FORCED — Add always CR/LF to records. LF — Add LF to records which do not end with these bytes for ASCII formatted data. DEVICE — When output is written to an output device. GCI — Communicator output format. NO_GCI — Write data without and messages with Communicator. GENERATE_GCI — Generate identical output as GCI.
OutPath		Path where to write output file
OutSubcat		Sub catalog (sub-directory) where output file(s) is/are written
OutSubName		
OutType	String	Defines the type of the output file. **PrintWhileSpooling** - Special handling to be able to read file when it's still being written. Reading process mustalso be modified to handle this feature (Xfilter with the parameter InType)!! **NamedPipe** - A linux name pipe (or FIFO) is used to write thew output data. This also means that the process must be started with the NextProcess keyword. This is needed because the named pipe blocks until the reading process has also opend the pipe. **Normal** - Create a normal output file.
PJM-Forced	Binary	Defines if tape filename listing is only read from tape when tape is different as last PJM listed tape, or also when tape is the same. Only valid when PJM-Remote has value YES. Possible values: Yes - Whole tape listing is always made. NO - Tape listing only made when tape is different, is default.
PJM-Remote	Binary	Specifies if PJM process is local or not. This is used when MAKE_DIR command is used to write file listing in output file specified by 'OutFile', instead of directly to the communicator. Possible values: Yes - PJM is remote, write MAKE_DIR list in file No - Normal behavior.
WriteTrailer	String	Enables trailer page info for LCDS. Possible values: Yes - Write trailer page info in output file No - No Trailer page info is written

2.4.5.5 Device Output Section

This section is added to describe the output device for the Input-Filter. This is only needed if the program needs to write to a device, or write a tape/floppy image file. Presently, the only supported mode is to write Xerox HD image files to Xerox floppy or tape. The other possibilities are to be released in the future.

```
Begin-OutputDevice
    DeviceTypeCARTRIDGE
    TapeRewindREWIND
End-OutputDevice
```

General Output Device Section

The JCF-Keywords described in the table below are used for all devices.

Keyword	Variable Type	Description
DeviceName	String	Device name. Normally these have following settings when the DeviceType has following settings: 35_FLOPPY_XRX Is ignored by Unix, dependent of floppy format CARTRIDGE, QIC, /dev/nrStpX, were X = 0…2 (SCO), or TAPE is ignored (Linux) FLOPPY Is ignored by Unix (5¼" floppy) HD <Not used>
DeviceType	String	Category of the data target. permitted entries are 35_FLOPPY_XRX, TAPE, CARTRIDGE, FLOPPY, HD.
FloppyFormat	String	Specifies if floppy media has to be formatted previously before data is written to it. Floppy is formatted to following format, dependent of floppy type for 5¼" is 720 KB, for 3½" it is 1.2 MB. Yes – Floppy is formatted. No - Floppy is not formatted (Default)
ImageName	String	TAPE: Filename/Path to output image data file (DeviceType = CARTRIDGE, QIC or TAPE). Xerox Floppy: Filename/Path to output image file (DeviceType = 35_FLOPPY_XRX or FLOPPY).

Tape Output Device Section

Keyword	Variable Type	Description
MagazineType	String	Only valid when DeviceType = CARTRIDGE. When a single cartridge drive is used this parameter should be SINGLE. MULTI is used for tape drives which can work sequentially through multiple magazines. JUKE is used for tape drives which individually addressable slots.
SCSIadapter	Value	SCSI host adapter number as Operating System counts them. The value of -1 specifies to check all adapters for device. Default value: -1
SCSIid	Value	SCSI device identifier as is set on the HW device itself. The value of -1 specifies to use the default SCSI id which is dependent of the specified DeviceType. Current defaults: CARTRIDGE: 3, QIC: 2, TAPE: 4. Default value: -1.
SCSIlun	Value	Reserved for future use.
TapeRewind	String	Tape command to be executed after input processing is finished. Possible options: NOREWIND, REWIND and UNLOAD. Default when parameter is not specified is NOREWIND.

SubcatInfo section

The SubcatInfo section is used to define all sub-catalog types used by the Input-Filter, together with the maximum number of members that may be open at one time. In the current version this number is always 1. The subcat types JOB and XPAR always have to be used, the others must be specified as needed. When a subcat is not defined, no printer data/resources files are generated (works as filter).

Some Xerox Resource subcats are NOT supported by the GCI, these are marked as such in the table below.

```
Begin-SubcatInfo
JOB             1
                XPAR1
                RDS2401
End-SubcatInfo
```

Supported Subcat types:

Sub catalog	GCI Supported	Used for Data type	Description
CFG	Yes		Configuration file ID.
CMD	Yes	Xerox Resource	Command file.
CME	Yes	Xerox Resource	Copy modification entry file.

Sub catalog	GCI Supported	Used for Data type	Description
CPX	Yes	Xerox Data	Print data files (comparex).
DAT		Xerox Resource	System data file.
DATA	Yes		ID for general data file.
FIS		Xerox Resource	Font interchange standard.
FNT	Yes	Xerox Resource	Font file.
FN6	Yes	Xerox Resource	Font file (600 dpi)
FRM	Yes	Xerox Resource	Form object file.
FR6	Yes	Xerox Resource	Form object file (600 dpi)
FSL	Yes	Xerox Resource	Form source language file.
ICT	Yes	Xerox Resource	Ink catalog.
IDR	Yes	Xerox Resource	Ink directory.
IMG	Yes	Xerox Resource	Digitized image file.
IM6	Yes	Xerox Resource	Digitized image file (600 dpi)
IPF		Xerox Resource	Interpress fragment file.
IPM		Xerox Resource	Interpress font mapping file.
ISL		Xerox Resource	
JDL	Yes	Xerox Resource	Job descriptor object file.
JOB	Yes	Job Control File	General job description file. **Always needed !!!!!!**
JSL	Yes	Xerox Resource	Job source language file.
LGO	Yes	Xerox Resource	Logo file.
LG6	Yes	Xerox Resource	Logo file (600 dpi)
LIB		Xerox Resource	
LOG		Xerox Resource	System integrity log file.
MSC		Xerox Resource	Temporary user file.
OLDS	Yes	SPP Data	OLDS print data files
OMSLIB	Yes	SPP resource	OLDS resource files
OSD		Xerox Resource	OSDS diagnostic file.
PCH		Xerox Resource	System patch file.
PDE	Yes	Xerox Resource	Page descriptor entry file.
PDS	Yes	AFPDS	AFPDS print file library.
PRIV	Yes	Xerox Resource	Temporary converted resources (X-Filter)
RDS	Yes	Xerox Resource	Xerox resource files
RDS240	Yes	AFP Resource	AFP resource library, 240 DPI
RDS300	Yes	AFP Resource	AFP resource library, 300 DPI
RDS600	Yes	AFP Resource	AFP resource library, 600 DPI
SAF		Xerox Resource	SAFES diagnostic file.
STK	Yes	Xerox Resource	Stockset file.

Sub catalog	GCI Supported	Used for Data type	Description
SYS		Xerox Resource	System control file.
TAPE	Yes		Input-Filter file ID
TMP	Yes	Xerox Resource	Temporary user file.
TPF		Xerox Resource	Temporary patch file.
TSK		Xerox Resource	System task image file.
TST	Yes	Xerox Resource	RTEXT object file.
XPAR	Yes	Parameter file	Input-/X-Filter host specific parameter file, compiled from Xerox JSL file. **Always needed !!!!!!**
XRX	Yes	Xerox Data	LCDS print file library
XCS		Xerox Resource	Interpress document files.

Action Section

Keyword	Variable Type	Description
ActionName	String	Program accepted actions. Possible values: DEV_OUT_COPY: Read from input and write to output can be used to create image files from tape DEV_OUT_WRITE: Read files from HD (LCDS Resource only, format …) and write this to a Xerox Floppy or tape. MAKE_DIR: Make directory/tape listing POSITION: Set device on specific position READ_FILE: Read data from device
CheckDataP-rocess	String	Activates check if the data needs to be processed by the Input-Filter, this is the case when the input data can be read by the next process. At the moment only valid for AFPDS data type, use OutFormatFile filename to write file which specifies the output format of the data file. Possible values: Yes - Activate check. No - Always process data (Default).
JDE-Name	String	Entry in parameter file to be used for this job.
ParamFile	String	Parameter file with VOLUME, BLOCK and RECORD parameters.
ParamSubcat	String	Sub catalog (sub-directory) were parameter file can be found.

Input Data Control

Keyword	Variable Type	Description
ByteOrder	String	Defines byte ordering from CPU/Host only supported for block / record length fields and HOST=BARRPC. Possible values: BigEndian: MSB LSB ordering (Default, IBM 370, PDP-10, Motorola, RISC) LittleEndian: LSB MSB ordering (INTEL, PDP-11, VAX)

Keyword	Variable Type	Description
InFile	String	Name of input file, wildcard '*' or a combination of character(s) and a wildcard. Wildcard can be used in the middle of characters, only one wildcard is allowed. When 'InFileType' is AFPDS or Pure_AFPDS and the data is read from tape the filename may contain the file sequence number followed by a colon, this enables file selection by tape sequence number instead on the filename itself (Examples: '1:', '1:Name', '1:*'). Be aware when both filename and file sequence numbers are used, both need to be correct. Also used when 370-channel dump is read or written for the dump output filename.
InFileFirst	Value	First file to print. Possible values: 0 - Prints from first file or all files (when InFileLast = 0) > 0 - Prints from file no. until InFileLast value
InFileLast	Value	Last report to print. Possible values: > 0 - Last file to print (InFileFirst <= InFileLast) 0 - Prints until last file (InFileFirst >= 0)
InFileType	Value	Input file format type. Possible values: AFP_Resource - AFP resource files AFPDS - AFP print data with linedata Pure_AFPDS - AFP print data SPP_Data - OLDS print data SPP_Resource - OLDS resource files XRX_Data - Xerox print data XRX_Resource - Xerox resource files
InFormat	Value	Define data input format. Possible values ….
PositDir	Character	Positioning direction. Possible values: '+' - Position forward (Toward End of Tape/File) '-' - Position backward (Towards Beginning of Tape/File)
PositMedium		Positioning type. Possible values: BLOCK - Positioning by specifying the number of blocks. DEVICE - Device dependent positioning, for example by tapes using the tape marks (TM). FILE - Specify the n-th file of a media.
PositUnit		Position unit. Possible values: BOT - Beginning of tape (PositMedium = DEVICE) Value -
RecordConstant	Value, characters	Used when AFPDS record delimiter is used and data is read from HD or from tape. Can be specified in hexadecimal (0x…) or characters with upto 4 bytes. Default value is the CRLF-delimiter (0x0D0A).

Keyword	Variable Type	Description
RecordLength	Value	Used when AFPDS Fixed records are used and read from HD or from non-labeled tapes (Range is from 12 up to 32760 bytes).
SkipBytes	Value	BARRPC only: Skip n-bytes at the beginning of the input file

Output Data Control

Keyword	Variable Type	Description
AddSequence-No	Binary	Defines that file sequence number is written as first part of the output filename followed by a dot '.' and the input filename when data is read. When the directory listing command is used the file sequence number is added too, with a colon as separator. Is only active when AFP print data is processed and the 'OutFile' parameter is set to identical. Possible values: Yes - Add file sequence number to output filename/directory listing. No - Create output filename as usual, is default.
AfpSfTrim	Binary	Removes padding bytes after Structured field in AFP-Print data until next CR/LF. CAREFULL can cause data loss when CR/LF are not added after SF!!!! Yes - Removes padding bytes No - All data is passed.
DeleteOutput	Value	When error occurs output file is deleted. Possible values: 0 - Output file is never deleted (Default) 1 - Output file is deleted when error occurs
MergeFiles	String	Merge output files. Possible values: Yes - Merge all output files to one output file No - Write multiple output files
ModifyFilename	String	Output filename modifications. Possible values: LowerCase - Change output filename to lower case. UpperCase - Change output filename to upper case.
OutFile	String	Name of output file or keyword 'IDENTICAL' to have the output filename the same as the input filename

Keyword	Variable Type	Description
OutFormat	String	Output file format. Possible values: CRLF Add CR/LF to record which do not end with these bytes. CRLF_FORCED Add always CR/LF to records. DEVICE When output is written to an output device. GCI Communicator output format. NO_GCI Write data without and messages with Communicator.
OutFormatFile	String	Path/filename in which this process writes the output format the file is written in. At the moment the process which follows is the backend which needs following keyword: CRLF delimiter - EXTINPUT=0, RDW – EXTINPUT=2.
OutPath	String	Path were to write output file
OutSubcat	String	Sub catalog (sub-directory) were output file(s) is/are written
OutSubName		
PJM-Forced	Binary	Defines if tape filename listing is only read from tape when tape is different as last PJM listed tape, or also when tape is the same. Only valid when PJM-Remote has value YES. Possible values: Yes - Whole tape listing is always made. No - Tape listing only made when tape is different, is default.
PJM-Remote	Binary	Specifies if PJM process is local or not. This is used when MAKE_DIR command is used to write file listing in output file specified by 'OutFile', instead of directly to the communicator. Possible values: Yes - PJM is remote, write MAKE_DIR list in file No - Normal behavior.
WriteTrailer	String	Enables trailer page info for LCDS. Possible values: Yes - Write trailer page info in output file No - No Trailer page info is written

2.5 JDE Parameter

Only the JDE Parameter which the Input-Filter supports and the allowed values are described here. A few ot the parameters have Input-Filter specific extensions!

2.5.1 Volume structure parameter

Keyword	Variable Type	Description
BMULT	Value Default: 1	Specifies a multiplication factor which is applied to the block length extracted from a tape label in order to determine the true block length. The value is an integer in the range of 1 to 15.
CODE	ASCII, EBCDIC, H6BCD, NONE, UNIVAC Default: EBCDIC	Specifies which code translation table is being used in interpreting the input data. The keyword NONE is used when data is not to be translated by input processing. This will be used when the job contains metacodes.
HOST	Default: IBMOS	Specifies the computer and/or host operating system that generated the input data file. See APA, LCDS and OLDS specific chapters for supported HOST/ LABEL values.
LABEL	ANSI, NONE STANDARD, UNDEF Default: STANDARD	Specifies the type of label on the input tape. The value NONE means no labels are written on tape. See APA, LCDS and OLDS specific chapters for supported HOST/LABEL values.
LCODE	ASCII, EBCDIC, H6BCD, NONE, UNIVAC, UNDEF Default: EBCDIC	Specifies which code translation table is used when interpreting the input label. When the value is set to UNDEF and ANSI or ANSI like labels are used the label code translation table is detected automatically.
MAXLAB	Default: 81	Is used internally when the rules are used. Not read from external parameter files.
MINLAB	Default: 80	Is used internally when the rules are used. Not read from external parameter files.
PLABEL	YES, NO Default: NO	Specifies whether the tape labels are to be printed. If YES is specified, all tape labels (except those encountered during a volume change) are printed on an output page and delivered to the sample print tray. Labels are truncated if they exceed the line width limits of the page. If NO is specified no tape label are printed.

Keyword	Variable Type	Description
RMULT	Value Default: 1	Specifies a multiplication factor to be applied to the record length extracted from a tape label in order to determine the true record length. A value is an integer in the range of 1 to 15.
UNPACK	Default: None	Is used internally when the rules are used. Not read from external parameter files.

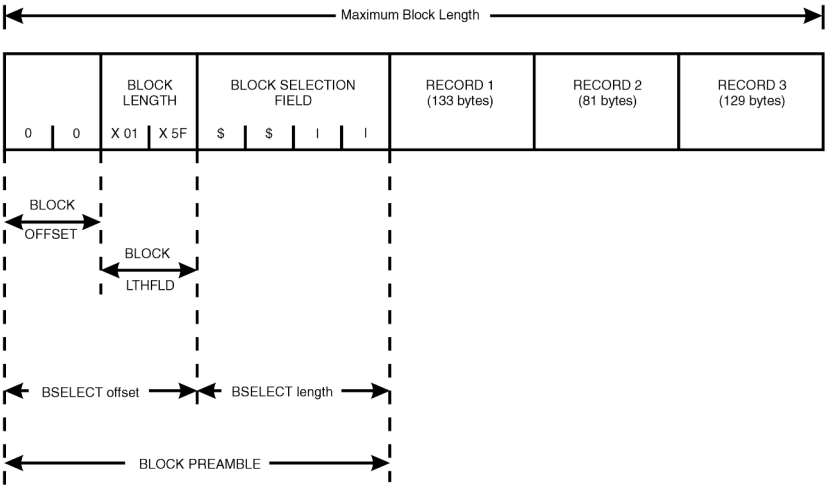
2.5.2 Block structure parameter

Keyword	Variable Type	Description
ADJUST	Value Default: 0	Specifies a block adjustment value which is added to or subtracted from the contents of the block length field to determine the true block length. A value specifies the block adjustment length. This length is a constant integer added to or subtracted from the value in the block length field of every tape block. The resulting value is the true block length. The range is from -127 to 127 and must be less than the block length option (LENGTH). The character plus (+) or minus (-) may be used to specify a positive or negative adjustment.
CONSTANT	String No Default	Specifies that the block delimiter constant string and all data following it are ignored until the end of the block is reached. The string may be a hexadecimal, octal or character constant. The length of the constant may be from one to four bytes.
FORMAT	BIN, DEC (, PACK, PKSG) Default: BIN	Specifies the recording mode of the block length field. Available type options are: BIN (binary), DEC (decimal). The following options are accepted, but not interpreted: PACK (packed with no sign), KKSG (packed with sign).
LENGTH	Value Default: 1330	Specifies the longest physical block being processed. A value specifies the length, in bytes, of the longest physical block (an integer in the range from 12 to 24576). For off-line processing the tape label contents may override a coded LENGTH command, but this length is still limited by the above maximum value.
LMULT	Value Default: 1	Specifies a multiplication factor being applied to the contents of the block length field to determine the true block length. A value is multiplied by the value in the length field. (refer to LENGTH command) to compute the number of bytes in the block. A value is an integer in the range of 1 to 15.

Keyword	Variable Type	Description
LTHFLD	Value Default: 0	Specifies the length of the field containing the block length. The size specifies the length in bytes of the field containing the block LENGTH specified above. The size is an integer in the range of 0 to 5. If size is set to zero, the block length field is not considered to be part of the block, and the length of a block, on tape, is the actual block length.
OFFSET	Value Default: 0	Specifies the location of the block length field. A value specifies the block length field offset. This offset is the number of bytes from the first byte of a block to the block length field. A value is an integer in the range of 0 to block length.
POSTAMBLE	Value Default: 0	Specifies the length in bytes of the extraneous data at the end of each tape block: i.e. it is an offset from the end of a block backwards to the end of the last logical record. The value is an integer in the range of 0 to block length.
PREAMBLE	Value Default: 0	Specifies the length of the operating system portion of the block, i.e. the byte offset from the first byte of a tape block to the first byte of the first logical record. The length is an integer in the range of 0 to the block length.
ZERO	YES,NO Default: NO	Specifies the end of block indicator. YES specifies that the end of a tape block is indicated by a value of zero in the record length field (before applying the record length adjustment). Data which follows the record is ignored up through the end of the block. NO indicates that the end of a tape block is not indicated by a value of 0 in the record length field.

Example BLOCK statement usage

BLOCK LENGTH=351, OFFSET=2, ADJUST=0, LTHFLD=2, PREAMBLE=8, FORMAT=BIN



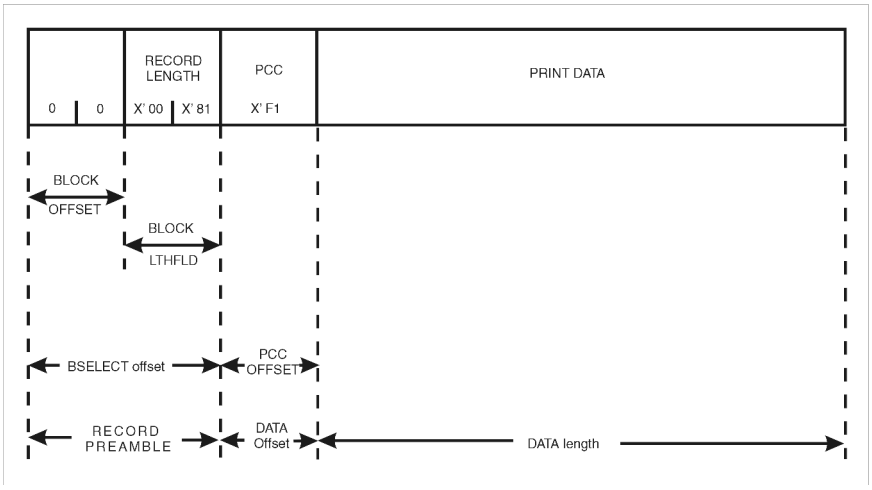
2.5.3 Record structure parameter

Keyword	Variable Type	Description
ADJUST	Value Default: 0	Specifies an adjustment value added to or subtracted from the contents of the record length field to determine the true record length. A value specifies the record adjustment length. It is a constant integer added to or subtracted from the value in the length field of every record. The range of a value is -127 to 127 and must be smaller or equal then the record length. The first character may be the sign plus (+) or minus (-).
CONSTANT	String No default.	Specifies a constant string used to signal the end of a record. This record delimiter constant string signals the end of a record, but it is not include in the print line. The string format can be an hexadecimal, octal or alpha constant. The length of the constant may be from 1 to 4.
FORMAT	BIN, DEC (, PACK, PKSG) Default: BIN	Specifies the format of the record length field. Available type options are: BIN (binary), DEC (decimal). The option PACK (packed with no sign) and PKSG (packed with sign) are accepted, but not used internally.
LENGTH	Value Default: 133 (off-line), 150 (on-line)	Specifies the length of the longest logical record. A value specifies the length, in bytes, of the longest logical record. A value is an integer in the range 1 to 310 for off-line. Maximum value for on-line without print position indexing is 214 (optimize mode) or 2140 (non-optimize mode). For off-line, the tape label contents may override the value.
LMULT	Value Default: 1	Specifies a multiplication factor applied to the contents of the record length field in order to determine the true record length. A value specifies the multiplication factor. The value specified is by the value in the length field to compute the number of bytes in the record. A value is an integer in the range 1 to 15.
LTHFLD	Value Default: 0	Specifies the length of the field containing the record length. The size specifies, in bytes, the record length field length. It must be an integer in the range 0 to 5. If the size is set equal to zero, record length are not contained in the records and the record length is the maximum length (LENGTH) for each record.
OFFSET	Value Default: 0	Specifies the location of the record length field. A value specifies the record length field offset. This offset is the byte offset from the first byte of the record to the record length field. A value must be an integer in the range 0 to LENGTH-LTHFLD-1.

Keyword	Variable Type	Description
POSTAM-BLE	Value Default: 0	Specifies the length of any extraneous data at the end of the record. The value specifies the length in bytes. The value must be an integer in the range 0 to record length.
PREAMBLE	Value Default: 0	Specifies the offset to the user portion of the record (i.e. the record preamble length). The value is the byte offset from the first byte of the record to the first byte of the user's portion of the record. The value must be an integer in the range 0 to record length.
STRUC-TURE	F, FB, V, VB, U, UB Default: FB	Specifies the general record structure of the input data. The string may be any of the following: F (Fixed length), FB (fixed length blocked), V (Variable length), VB (Variable length blocked), U (Undefined length) or UB (undefined length blocked). For off-line processing, the tape label contents may override this option.

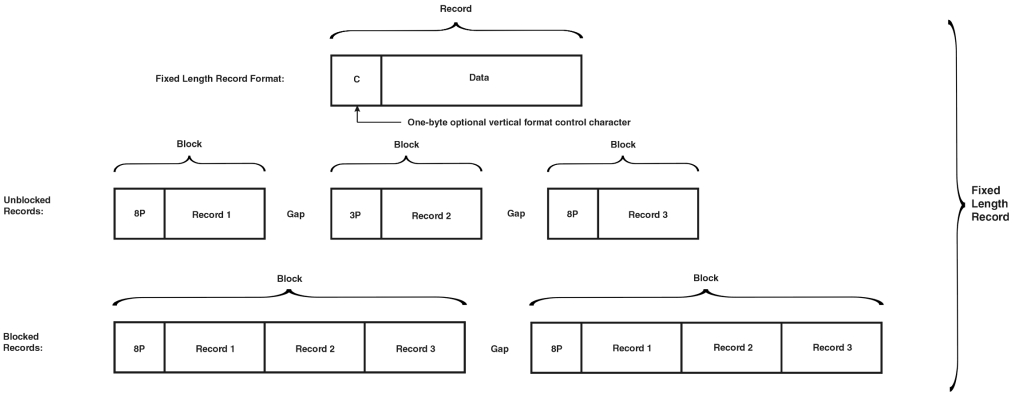
Example RECORD statement usage

RECORD LENGTH=133, OFFSET=2, LTHFLD=2, PREAMBLE=4, ADJUST=4, FORMAT=BIN

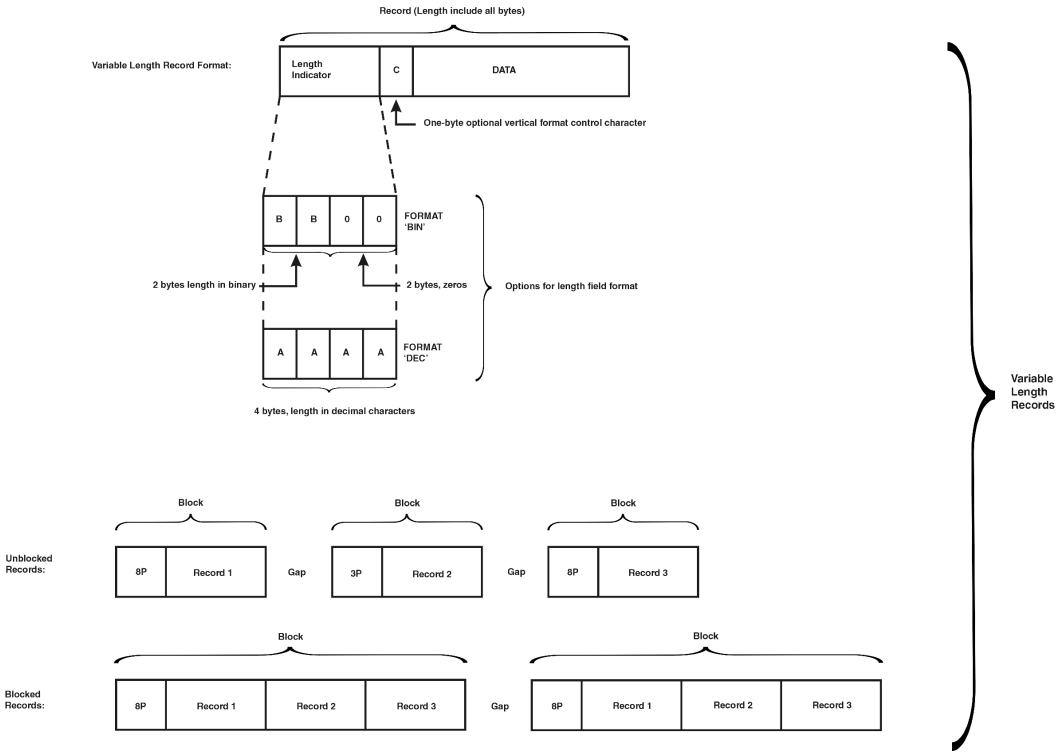


Generally are the following record format types used:

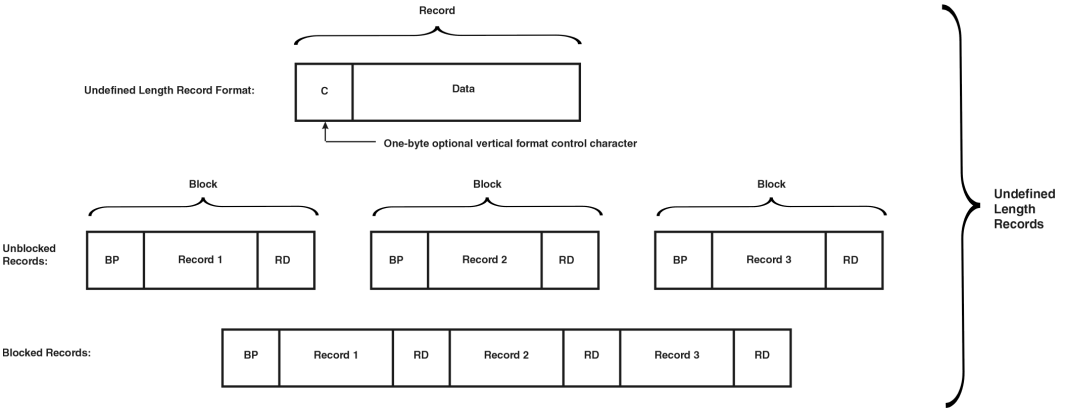
Fixed length records:



Variable length records:



Undefined length records:



2.6 Troubleshooting

2.6.1 Writing a trace file

This file is generated by the Input-Filter and shows the function call sequences as well as main values from variables which are used. This can be used as a first indication where the error is located in the I-Filter. To create this trace file the following actions are needed:

1. Edit the following Keyword values in correct `/u/prismapro/lib/uimng/jcf/*.JCF` file.
2. Set the trace level to 3 or 4.
3. Set trace flags to

Normal trace	FFFFFFF8	Full trace with time information
Full Speed trace	FFFFFFFC	Only when Input-Filter detects internal error trace is written. Has not always enough information to solve problem but allows full speed trace.
Compare trace	FFFFFFF0	Full trace without time information, can be used to be able to compare 2 trace files
Data trace	FFFFFFFA	Data trace files are written, be aware that these are written besides the log-file, can use a lot of disk-space dependent of data size.

Be aware that trace level 4 generates large trace files, at the moment the maximum size is 1 GB then the Input-Filter stops writing to the file. This behavior can be changed by using the JCF-Parameter 'TraceMaxFileSize' in chapter 'Input-Filter section'. For more details about the meaning of the trace level and flags see chapter 'Trace level and flags'.

```
Begin-I-Filter
     Begin-ComInfo
         ....
         TraceLevel        0
         TraceFlags        FF00000000000000
         TraceFile         /u/spool/diag/ifilter.trc
     End-ComInfo
     TraceLevel     4                                          <-- I-Filter trace level
     TraceFlags     FFFFFFF0, FFFFFFF8, FFFFFFFA or FFFFFFFC<-- I-Filter trace flags
....
End-I-Filter
```

4. Reproduce the error
5. Copy the `/u/prismapro/diag/ifilter/*` to save the traces, these files are overwritten every time the Input-Filter is started.
6. Repeat point 1), 2) and 3) as often as traces are needed, when the problem occurs sometimes a correct and an incorrect trace can help.
7. Set Trace level back to original value.

2.6.2 Trace Level and Flags

The Input-Filter has a trace level and (normal/special) flags which control the behavior on how the trace file is written. It is possible by using the level and flags to control the amount of data and when the data is written. This is important when print by spooling is used and the speed may not be reduced too much.

The flag are used bit wise which means each trace flag mode can be set separately. The modes are divided into two sections called special flags and normal flags. The special flags use the lower part of the flags (0x000000nn), whereas the normal flags use the upper part (0xnnnnnn00).

Normally the TraceLevel should be set to 3 or 4 to get enough trace details, this produces then a large trace file up to 1 GB (default). The maximum trace files size can be reduced by using the JCF-file keyword 'TraceMaxFileSize'. The trace flags should be usually set to 0xfffffffc for normal tracing, to 0xfffffffe to write only a trace when an I-Filter error occurs.

Trace level

The trace level can be set from 0 to 4. Each level higher produces more detailed information and includes all the messages from the lower levels.

Level	Level Name	Description
0	Error messages	When errors occur.
1	Section messages	Main section enter/exit info.
2	Function messages	Function enter/exit w. return code info
3	Info messages	Important variables/information
4	Detail messages	Detailed variables and other information (Large Trace File!!!!!)

Scheme of program process for Single-Thread-Mode:

Read data -> **Input buffer** -> Process data -> **Output buffer** -> Write data

Depending on the buffer segment type (part of buffer) the trace lines are written in ASCII or binary together with conversion to ASCII/EBCDIC/H6BCD format. The size of the file can be reduced by using the DataTraceBegin and/or DataTraceEnd keywords, these define at which data position the data trace is written and when it stops writing it.

By using the 'Normal trace flags' the data trace can be selected individually to be written or not. The files are written to directory `'/u/prismapro/diag/ifilter'` with one of the following names:

Data trace filename	Status	Description
jobid*_pid*_bt_inp.in	Enabled	Input buffer trace triggered when data is written to buffer segment and released to be read.
jobid*_pid*_bt_outp.in	Enabled	Output buffer trace triggered when data is written to buffer segment and released to be read.
jobid*_pid*_bt_outp2.in	Enabled	Generate GCI output buffer trace triggered when data is written to buffer segment and released to be read.
jobid*_pid*_bt_lbl.in	Enabled	Label buffer trace triggered when data is written to buffer segment and released to be read.

Error only flag

Only when the Input-Filter detects internally an error the trace file with the history (what happened just before the error) is written. Also a snapshot of all the internal class variables is written.

Time flag

Write time information per line into the trace file. The first number is the hour/minute/seconds the second number is the time in milliseconds since the Input-Filter is running.

Section flag

Write Input-Filter section information (see also 'Normal Trace flags') and a number by which thread the method is called from.

Buffer trace flag

When set the trace messages are internally buffered. When cleared the traces are written line by line to the output.

Normal trace flags

The normal trace flags are used to select trace messages from one or more Input-Filter sections. Normally all normal trace flags will be set to get all trace messages. If only traces are needed from a special program section these can be selected by using the following flags:

Special Flags	Name	Description
0x00000100	Init section flag	Write program initialization (command line interpretation, reading parameter files) trace messages
0x00000200	Input section flag	Write program input (device input) trace messages
0x00000400	Input buffer section flag	Write program input buffer trace messages
0x00000800	Process section flag	Write program label / data processing trace messages
0x00001000	Output buffer section flag	Write program output buffer trace messages
0x00002000	Output section flag	Write program output (GCI / no GCI) trace messages
0x00010000	Label input data trace	Write label input trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00020000	Label output data trace (Disabled by SW)	Write label output trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00040000	Input buffer input data trace	Write input buffer input trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00080000	Input buffer output data trace (Disabled by SW)	Write input buffer output trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00100000	Output buffer input data trace	Write output buffer input trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00200000	Output buffer output data trace (Disabled by SW)	Write output buffer output trace (Only when ' Data trace flag' is set, see 'Special trace flags')

Special trace flags

The special flags are used to control the behavior of the trace file/Input-Filter. The following special trace mode are available:

Special Flags	Name	Description
0x00000001	Copy flag	Creates a copy of the output file in the current directory.
0x00000002	Data trace flag	Special buffer traces are written as ASCII file of the data which is passed through the internal Input (called ifilter.inp)/Output (called ifilter.out) buffers
0x00000004	Error only flag	Only when error occurs trace info is written into the trace file. This includes also traces which happen just before the error occurs. The trace file limit is disabled to enable writing this trace info.
0x00000008	Time flag	Time info is added for each trace line.
0x00000010	Section flag	Program section and source code id./class are added for each trace line.
0x00000020	Buffer trace flag	The trace info is buffered until the buffer is full and then written to the trace file. By NOT setting this bit the trace information is written line by line to the trace file.

The Copy flag and the Data trace flag are not often used because they reduce the performance quite a lot.

Copy flag

A copy of the Input-Filter output files is created in the current directory of the program.

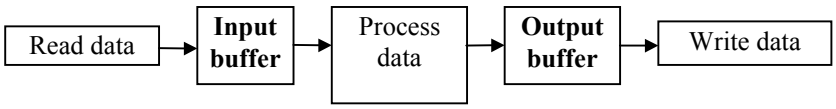
Data trace flag

All data which is passed through the Input-Filter buffers can be written to output files. At the moment 3 buffers are used:

Input buffer — When data is read from device it is written into the Input Buffer. When data is processed it is read from the Input buffer.

Output buffer — When data is processed it is written into the Output buffer. The data which is written as output is read from the output buffer.

Label buffer — This buffer is used to recognized if the PJM listing is from the tape as the last one. If so the Input-Filter stops reading the tape and the previous PJM listing is shown.



Scheme of program process for Single-Thread-Mode

Depending on the buffer segment type (part of buffer) the trace lines are written in ASCII or binary together with conversion to ASCII/EBCDIC/H6BCD format. The size of the file can be reduced by using the DataTraceBegin and/or DataTraceEnd keywords, these define at which data position the data trace is written and when it stops writing it.

By using the 'Normal trace flags' the data trace can be selected individually to be written or not. The files are written to directory `'/u/prismapro/diag/ifilter'` with one of the following names:

Data trace filename	Status	Description
jobid*_pid*_bt_inp.in	Enabled	Input buffer trace triggered when data is written to buffer segment and released to be read.
jobid*_pid*_bt_outp.in	Enabled	Output buffer trace triggered when data is written to buffer segment and released to be read.
jobid*_pid*_bt_outp2.in	Enabled	Generate GCI output buffer trace triggered when data is written to buffer segment and released to be read.
jobid*_pid*_bt_lbl.in	Enabled	Label buffer trace triggered when data is written to buffer segment and released to be read.

Error only flag

Only when the Input-Filter detects internally an error the trace file with the history (what happened just before the error) is written. Also a snapshot of all the internal class variables is written.

Time flag

Write time information per line into the trace file. The first number is the hour/minute/seconds the second number is the time in milliseconds since the Input-Filter is running.

Section flag

Write Input-Filter section information (see also 'Normal Trace flags') and a number by which thread the method is called from.

Buffer trace flag

When set the trace messages are internally buffered. When cleared the traces are written line by line to the output.

Normal trace flags

The normal trace flags are used to select trace messages from one or more Input-Filter sections. Normally all normal trace flags will be set to get all trace messages. If only traces are needed from a special program section these can be selected by using the following flags:

Special Flags	Name	Description
0x00000100	Init section flag	Write program initialization (command line interpretation, reading parameter files) trace messages
0x00000200	Input section flag	Write program input (device input) trace messages
0x00000400	Input buffer section flag	Write program input buffer trace messages
0x00000800	Process section flag	Write program label / data processing trace messages
0x00001000	Output buffer section flag	Write program output buffer trace messages
0x00002000	Output section flag	Write program output (GCI / no GCI) trace messages
0x00010000	Label input data trace	Write label input trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00020000	Label output data trace (Disabled by SW)	Write label output trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00040000	Input buffer input data trace	Write input buffer input trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00080000	Input buffer output data trace (Disabled by SW)	Write input buffer output trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00100000	Output buffer input data trace	Write output buffer input trace (Only when ' Data trace flag' is set, see 'Special trace flags')
0x00200000	Output buffer output data trace (Disabled by SW)	Write output buffer output trace (Only when ' Data trace flag' is set, see 'Special trace flags')

3 Accounting

3.1 Basic Accounting Information on the Trailer Pages

Trailer pages which are (optionally) printed after each print job contain information like the number of printed pages and sheets for this specific job. The layout and content of the header page (hdrpage), the separator page (infopage), the message page (infopage) and the trailer page (trlpage) can be customized to meet customer requirements. However, the information that can be put on the trailer page is limited to a fixed list of system-provided information shown below and user-defined values supplied with the job ticket at job submission time.

```
Title:                                  $TITLE$
Software version:                       $VERSION$
Software build date:                    $VERSIONDATE$
Software build time:                    $VERSIONTIME$
Type of separator page:                 $PAGENAME$
User name:                              $OWNER$
Job name:                               $JOBNAME$
Printfile name:                         $FILENAME$
Host name:                              $CLIENT$
Spool ID:                               $SPLID$
Hostname:                               $CLIENT$
Print time:                             $TIME$
Print date:                             $DATE$
Printer ID:                             $PRINTER$
Status(START NEW_XMIT OR CONT):         $PAGESTATUS$
Name:                                   $REC_NAME$ $DEPARTMENT$
Room:                                   $ROOM$
Building:                               $BUILDING$
Address:                                $ADDRESS1$ $ADDRESS2$ $ADDRESS3$
$ADDRESS4$

Total XMITS :                           $ACCT_XMITS$
Total Sheets:                           $ACCT_PAGES$
Total Pages:                            $ACCT_PAGESIDES$
Extra Sheets:                           $ACCT_EXTRA_PAGES$
Number of PAGEDEFS:                     $ACCT_NUM_PAGEDEFS$
Number of FORMDEFS:                     $ACCT_NUM_FORMDEFS$
Number of requested fonts:              $ACCT_NUM_FONTS$
Number of loaded fonts:                 $ACCT_NUM_LOADED_FONTS$
Number of requested overlays:           $ACCT_NUM_OVERLAYS$
Number of loaded overlays:              $ACCT_NUM_LOADED_OVERLAYS$
Number of requ. page segmemts:          $ACCT_NUM_PAGESEGS$
Number of loaded page segmemts:         $ACCT_NUM_LOADED_PAGESEGS$
Duplex/Simplex flag                     $ACCT_DUPLEX$
Used Output Bins flag                   $ACCT_OUTPUT_BIN$
Input Bin 1 Page length (1/6î)          $ACCT_PAGE_LEN1$
            Page width  (1/6î)          $ACCT_PAGE_WIDTH1$
            Page count                  $ACCT_PAGES1$
Input Bin 2 Page length (1/6î)          $ACCT_PAGE_LEN2$
            Page width  (1/6î)          $ACCT_PAGE_WIDTH2$
            Page count                  $ACCT_PAGES2$
Input Bin 3 Page length (1/6î)          $ACCT_PAGE_LEN3$
            Page width  (1/6î)          $ACCT_PAGE_WIDTH3$
            Page count                  $ACCT_PAGES3$
Input Bin 4 Page length (1/6î)          $ACCT_PAGE_LEN4$
            Page width  (1/6î)          $ACCT_PAGE_WIDTH4$
            Page count $ACCT_PAGES4$
```

This is the default layout for the LCDS trailer page:

```
20
20 OPS PRISMAproduction LCDS
20 Customer title
20
20
20 DATE:                  $DATE$     AT      $TIME$
20 DEPARTMENT:            $XDEPT$
20 JOB ID:                $SPLID$  REPORT NO.$XREP$
20 FILE ID:               $IFILEID$
20
20 INPUT                  PROCESSING TIME:   $ITIME$
20 CONVERTER              PROCESSING TIME:   $XTIME$
20 OUTPUT                 PROCESSING TIME:   $PTIME$
20
20 PAGES CONVERTED:                          $PAGES$
20 LOGICAL PAGES CONVERTED:                  $XPAG$
20
20 GRAPHIC PAGES PRINTED:                    $XGRA_PAG$
20 GRAPHIC IMAGES READ:                      $XIMG_READ$
20 LINES PRINTED:                            $XLINES$
20 TAPE MOUNTS:                              $ITAPMNT$
20 BLOCKS READ:                              $XBLOCK$
20 RECORDS READ:                             $XREC$
20 DJDE RECORDS READ:                        $XDJDE$
20 MAXIMUM COPY COUNT:                       $XCOPY$
20 OVERPRINTS:                               $XOVERP$
20 COLLATE:                                  $XCOLL$
20 SIMPLEX/DUPLEX:                           $ACCT_DUPLEX$
20 JDE,JDL USED:                             $XJDE$, $XJDL$
20 ACCTINFO:                                 $XACCT$
20 JOBNAME:                                  $JOBNAME$
20
20 INITIAL FONT LIST:                        $XFNT1$
20                                           $XFNT2$
20                                           $XFNT3$
20 $XMFNT$
20 INITIAL FORM LIST:                        $XFRM1$
20                                           $XFRM2$
20                                           $XFRM3$
20 $XMFRM$
20 INITIAL CME  LIST:                        $XCME1$
20                                           $XCME2$
20                                           $XCME3$
20 $XMCME$
20 INITIAL INK LIST:                         $XINK1$
20                                           $XINK2$
20                                           $XINK3$
20 $XMINK$
20
20
20 OCE Printing Systems GmbH
20 OPS PRISMAproduction LCDS
20
```

3.2 Enhanced trailer page accounting

The general idea to print accounting information on the trailer page has proven its use and versatility. In order to enhance this feature, only some small changes have to be implemented.

Attention: Every name in capital letters enclosed a pair of "$" signs is treated as a parameter and replaced with its value. If the parameter is not found, it is replaced with a blank sign " ". To prevent the interpretation of "$" signs in constant text parts, they have to be quoted with a "\" sign.

- **Make it possible to provide information to be printed on the trailer page after the job has started printing.**
  This is important for data extracted from the data stream and/or collected by filter and converter programs like X-Filter in print-while-spooling mode. This requires changes in the implementation of Spool (to accept updates for print parameters after the job has been submitted) and the LI (to re-read the job parameters after printout and before preparing the trailer page content).

Some examples (for the LCDS-Module) for additional data available on the trailer page:

```
LINES PRINTED :                  $Xlines$
BLOCKS READ :                    $Xblock$
BLOCKS SKIPPED :                 $XblockS$
RECORDS READ :                   $Xrec$
DJDE RECORDS READ :              $Xdjde$
GRAPHIC PAGES PRINTED:           $Xgra_pag$
GRAPHIC IMAGES READ:             $Ximg_read$
MAXIMUM COPY COUNT:              $Xcopy$
OVERPRINTS :                     $Xoverp$
COLLATE :                        $Xcoll$
SF/MF :                          $Xsfmf$
SIMPLEX/DUPLEX:                  $ACCT_DUPLEX$
JDE,JDL USED :                   $Xjde$, $Xjdl$
ACCTINFO:                        $Xacct$
INITIAL FONT LIST :              $Xfont$
```

3.3 Administration of Accounting Data

Accounting data is collected in two ways: the information is stored in a database and also in text-based legacy accounting files. Both possibilities can be configured in a separate administration application. See 'Accounting Administration' in the 'Configuration' chapter of the PRISMAproduction user's guides.

3.4 Legacy Accounting Files

- All accounting information based on legacy files is kept in the directory `/u/prismapro/account/`. This directory name cannot be changed.
- Each accounting file contains the data collected for one day. The name of the accounting file contains the date: **<YYYYMMDD>.acc**
- Accounting files are automatically generated by the account server program.
- The account files are written in text format, in the representation of the operating system on which the account server program is running.
- Legacy Account file handling is available via GUI for the administrator ('Configuration' -> 'Accounting'). The following functions are available: Deleting accounting records (after a defined period of time or directly) and copying accounting files (e.g. to DOS-formatted floppies).

Record format

- Each accounting record can be up to 255 characters long and is stored as one line in a text file. This means the record ends with either a <LF> (0x0a) on Unix platforms or <CR><LF> on WindowsNT platforms.
- Each accounting record begins in column 1 with a statement of the record type in 4 digits followed by a semicolon.
- The first two digits of the record type are product-specific, the third digit is program-specific. The last digit allows programs to write more than one record types. The next chapter contains a list of defined product code numbers.
- The second item of information in every accounting record is the accounting ID. This consists of a main account-id with 8 digits and an optional sub-id with 4 digits, separated by a point (aaaaaaaa.ssss). The optional sub-id is used for jobs which are split up during processing (see chapter 2.2.3). The accounting ID is also followed by a semicolon.

- All following items of information are separated by a semicolon. The separator is included even if a particular item is missing, so the number of items per record is fixed.
- The items in a record must be either of type TEXT (all printable characters with the exception of the semicolon, which is used as field separator), or NUM (digits 0-9, characters + - and . are allowed).
- Since job grouping (multiple print data files per job) is supported from PRISMA, an additional record with a sub ID becomes relevant. Records without an accounting-sub-id belong to the whole job. They include the job specified parameters like job name and job copies. Some records types will write additional records including an accounting-sub-id for each data file of the job. Here the specific data file name as well as the specific copies for each file (file copies) is stored. The sub-id number is incremented for every additional file. E.g.: A job with ID 12345678 containing 2 files will generate the sub-ID 12345678.0001 for the first file, and 12345678.0002 for the second file.

3.4.1 List of product code numbers

The first two digits of the record type are product-specific. Here are the currently defined numbers:

Product code	Product name
10	PRISMAproduction Common Components
11	PRISMAproduction Input Modules
12	PRISMAproduction LCDS-Module
13	PRISMAproduction POD/PCL-Module
14	PRISMAproduction OLDS-Module
16	PRISMAproduction Backend V3.02 and higher
17	PRISMAproduction Backend V3.02 and higher
20	EPSS (Namur)
21	PRISMAarchive Document Manager (Namur)
4xxx	DPS (Venlo)
5xxx	DPS (Venlo)
6xxx	WFPS (Venlo)
7xxx	WFPS (Venlo)
99	User-specific records. The code 99 is reserved for user-defined extensions of the standard PRISMApro supplied accounting data.

3.5 Description of specific record types in PRISMAproduction CC

3.5.1 Input-Filter

Record Type 1000

This record type is written only once for each job if the Input-Filter is used. It contains information about the input device.

Parameter	Type	Length (max.)	Description
1000	NUM	4	record type
account_id	NUM	8.4	accounting-ID
device	TEXT	32	type of input device (e.g. Tape, Harddisk, /370-Channel)
filename	TEXT	128	original filename on the input device (may be empty if no filename is available) For multifile input only the first filename is given.

Example: `1000;12345678;Tape;DATA123`

3.5.2 Print Job Manager

Record Type 1010

This record is written once for each job which is submitted using the Print Job Manager. It contains the global job parameters.

Parameter	Type	Length (max.)	Description
1010	NUM	4	record type
account_id	NUM	8.4	accounting-ID
client	TEXT	64	network client ID (hostname or IP address) of the computer on which the job was submitted
user	TEXT	32	user-ID from which the job was submitted
num_files	NUM	3	number of print files in the list of files
printer	TEXT	32	name of printer for which the job was submitted
jobclass	NUM	3	job class
resolution	NUM	3	printing resolution in dpi (240, 300 or 600)
proofprint	TEXT	3	proof print yes/no
storeprint	TEXT	3	storeprint yes/no
cust_acc	TEXT	16	customer account id
date	TEXT	10	submission date (format dd.mm.yyyy)
time	TEXT	8	submission time (format hh:mm:ss)
redirection	NUM	8.4	not used in V3
form	TEXT	8	Content of the FORM parameter (not used = empty)

Example: `1010;12345678;server2.ops.de;miller;1;PS700 P22;3;300;yes;no;08/15 R2D2; 23.08.1999;14:42:55;12345677;STD`

Record Type 1011

This record type will appear once for each entry in the list of files in a job ticket. It contains file-specific parameters. This record type is not used for the redirection mode.

Parameter	Type	Length (max.)	Description
1011	NUM	4	record type
account_id	NUM	8.4	accounting-ID
filetype	TEXT	8	data type of input file (e.g. PS, PCL, AFP)
filename	TEXT	128	original filename on the client (including path)

Example: `1011;12345678;AFP;/usr/data/very/long/pathname/file24`

3.5.3 ODS

Record Type 1012

This record is always written for each job disappeared from the All Jobs List in the explorer.

Parameter	Type	Length (max.)	Description
1012	NUM	4	record type
account_id	NUM	8.4	accounting-ID
user name	TEXT	32	user of the explorer
endestatus	NUM	1	0...normal end of the job (after end of the "Hold Finished Jobs" time) 2...deleted by the operator
date	TEXT	10	end date (format tt.mm.yyyy)
time	TEXT	8	end time (format hh:mm:ss)

Example: `1012;12345678;service;0;23.08.2002;14:42:55`

3.5.4 UI-Manager

Record Type 1020

This record type is written once for each job started from the UI-Manager. This is internally done by submitting a JCF to the communicator. The record contains the global information about the submitted job.

Parameter	Type	Length (max.)	Description
1020	NUM	4	record type
account_id	NUM	8.4	accounting-ID
user name	TEXT	32	user-ID from which the job was submitted
JCF	TEXT	32	filename of the job control file that was submitted
date	TEXT	10	submission date (format dd.mm.yyyy)
time	TEXT	8	submission time (format hh:mm:ss)

Example: `1020;12345678;applic;Line_std.JCF;23.06.1999;`
`14:42:55`

3.5.5 Consumables

Record Type 1026

This record is used to post a forecast of used colors for one job-copy of the print job. Because it's not known, if the job would be printed simplex or duplex, all counts are calculated as pages, instead of sheets. This record is created when the job becomes "Ready to Print".

Parameter	Type	Description
1026	NUM	record type: forecast colors
account_id	NUM	accounting-ID without sub-id: Record belongs to the whole JOB only
Pages	TEXT	Pages in the job for per job-copy
Colors	TEXT	Count of colors used on this page count (usually 1 – 5)

Example: `1026;00001305;237;2`

The job with the jobID 00001305 contains 237 pages with 2 colors.

There can be multiple entries for the same jobID if there are also pages with only 1 or more colors.

3.5.6 Spool

Record Type 1030

This record is **always** written for every job and contains data which the spool reads from the job parameter file. If a parameter does not contain a correct value, the spool has no possibility to know and correct it. Therefore the submitting process is responsible for providing the proper information with the job.

Parameter	Type	Length (max.)	Description
1030	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
user name	TEXT	32	user-ID from which the job was submitted
filetype	TEXT	8	data type of input file (e.g. PS; PCL; AFP)
jobname	TEXT		JOB: jobname
filename	TEXT	128	FILE: original filename on the client (including path)
printer	TEXT	32	name of printer for which the job was submitted

Parameter	Type	Length (max.)	Description
jobqueue	NUM	3	jobqueue
resolution	NUM	3	printing resolution in dpi (240; 300 or 600)
date	TEXT	10	submission date (format dd.mm.yyyy)
time	TEXT	8	submission time (format hh:mm:ss)
job-copies	NUM	5	JOB: copies
file-copies	NUM	5	FILE: copies

Example (with job grouping):

```
1030;12345678;service;AFP;pages;;1;600;10.12.2003;14:42:55;1

1030;12345678.0001;service;AFP;/u/prismapro/demo/afp/pages/
pages.txt;;1;600;10.12.2003;14:42:55;1
```

Record Type 1031

This record is written for every job after it has been finished or deleted from the queue.

Parameter	Type	Length (max.)	Description
1031	NUM	4	record type
account_id	NUM NUM	8 8.4	accounting-ID without sub-id: Record belongs to the whole JOB accounting-ID with sub-id: Record belongs to one data FILE
user name	TEXT	32	user-ID of the operator who printed or deleted the file
endstatus	NUM	1	end status of the job: 0...printed successfully 1...printed successfully and kept with status “locked” 2...deleted by the operator 3...deleted automatically 4...reset active printer 5...PS Abend 6...Job was interrupted after "interrupt job" request. 7...Job was aborted because of bad print data or resources 8...cancel+hold 9...canvel+delete
printer	TEXT	32	name of printer on which the job was printed (if endstatus is not 2 or 3)
jobqueue	NUM	3	last used jobqueue for this job
range	TEXT	14	last used print range for this job
form	TEXT	8	Content of the FORM parameter
jobname filename	TEXT TEXT	32 128	JOB: jobname FILE: original filename on the client (including path)
startdate	TEXT	10	date (format dd.mm.yyyy) when job was received by spool
starttime	TEXT	8	time (format hh:mm:ss) when job was received by spool
enddate	TEXT	10	end date (format dd.mm.yyyy) when job was removed from the spool queue or locked
endtime	TEXT	8	end time (format hh:mm:ss) when job was removed from the spool queue or locked
jobcopies filecopies	NUM NUM	5 5	JOB: copies FILE: copies
printcount	NUM	5	count, how many times, the job has been printed (start with 1). 0 means: Job has never printed.

Parameter	Type	Length (max.)	Description
PPM	NUM	5	printing speed in pages per minute (ppm), based on A4 pagesize. 0 means: unknown ppm value.

Example (with job grouping):

```
1031;12345678;admin;0;PS880 P15;7;1101-2000;A4;Department 08/
15; 23.08.1999;14:42:55;24.08.1999;06:12:59;1;5;222

1031;12345678.0001;admin;0;PS880 P15;7;1101-2000;A4;/u/ser-
vice/
test.afp;23.08.1999;14:42:55;24.08.1999;06:12:59;3;5;333
```

Record Type 1032

This record is always written for every job (the value of the parameter useracc and the parameter itself is independent).

It is a container for user-defined and user-supplied data which should be written into the accounting database for a print job. This record type is written on the same occasion as record type 1031.

Parameter	Type	Length (max.)	Description
1032	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
useracc	NUM	3	value of the parameter useracc
u_acc1	TEXT	32	content of parameter u_acc1
u_acc2	TEXT	32	content of parameter u_acc2
u_acc3	TEXT	32	content of parameter u_acc3
u_acc4	TEXT	32	content of parameter u_acc4
u_acc5	TEXT	32	content of parameter u_acc5
u_acc6	TEXT	32	content of parameter u_acc6

Examples for using OCT custom tickets (only for PRISMA ≥ V3.02.15):

To write user accounting information in Rec 1032, the following has to be done:

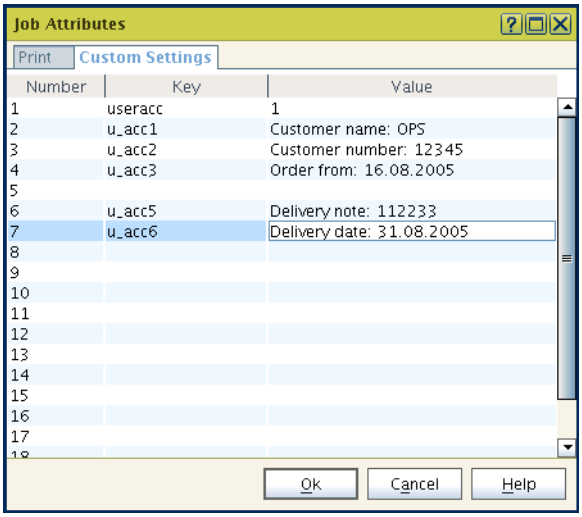
For use with a POD or a APA job ticket an OCT must be created with at least one of the following sections:

[job]
Job_Info1=u_acc1;job-usertext for acc1
Job_Info2=u_acc2;job-usertext for acc2
Job_Info3=u_acc3;job-usertext for acc3 ... etc.

[files]
Info1=u_acc1;usertext for acc1 related to file 1
Info2=u_acc2;usertext for acc2 related to file 1 ...etc

[files]
Info1=u_acc1;usertext for acc1 related to file 2 ...etc until
Info6=u_acc6;usertext for acc6 related to file 2

As an **example** for getting this record type, you can use the Custom settings tab in the 'Job Attributes' section of the PJM for each file:



Result of this example (with job grouping):

```
1032;12345678;1;Customer name: OPS;Customer number: 12345;Or-
der from 16.08.2005;;Delivery note: 112233;Delivery date:
31.08.2005
```

```
1032;12345678.0001;1;Custom name: OPS;Customer number:
12345;Order from 16.08.2005;;Delivery note: 112233;Delivery
date: 31.08.2005
```

3.5.7 AFP2IPDS Backend

Record Type 1040

This record is written by the AFP2IPDS Backend for every job and contains the basic information about the actually printed pages and the time it took to print the job.

Parameter	Type	Length (max.)	Description
1040	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
startdate	TEXT	10	date (format dd.mm.yyyy) when printout or job was started
starttime	TEXT	8	time (format hh:mm:ss) when printout or job was started
enddate	TEXT	10	date (format dd.mm.yyyy) when printout or job was ended
endtime	TEXT	8	time (format hh:mm:ss) when printout or job was ended
simplex	NUM	1	simplex mode was used 0...no 1...yes (e. g. simplex and rsimplex)
duplex	NUM	1	duplex mode was used 0...no 1...yes (e. g. normal duplex, tumble duplex and rzigzag normal)
colormode	NUM	1	reserved
endstatus	NUM	1	status after printout was ended 0...job completed without errors 1...job completed with errors 2...job interrupted by operator 3...job processing interrupted by server error
pages	NUM	8	JOB: summary of the number of pages printed (incl. info pages) / n-up-mode FILE: number of pages printed for all file copies and one job copy (incl. info pages) / n-up-mode
opages	NUM	8	JOB: summary of the original job size in pages (if known) FILE: original job size in pages (if known) for one file- and job copy
fpages	NUM	8	JOB: summary of the number of front pages FILE: number of front pages for all file copies and one job copy

Parameter	Type	Length (max.)	Description
bpages	NUM	8	JOB: summary of the number of back pages FILE: number of back pages for all file copies and one job copy
sheets	NUM	8	JOB: summary of the number of sheets (incl. info pages) FILE: number of sheets (incl. info pages) for all file copies
ipages	NUM	8	JOB: summary of the number of info pages printed FILE: number of info pages printed for all file copies and one job copy
isheets	NUM	8	JOB: summary of the number of info sheets printed FILE: number of info sheets printed for all file copies and one job copy
offsets	NUM	8	JOB: summary of the number of offsets in the output tray FILE: number of offsets in the output tray
feet	NUM	8	JOB: summary of the length of paper in feet printed (fanfold only) FILE: length of paper in feet printed (fanfold only)
printcount	NUM	5	count, how many times, the job has been printed (start with 1). 0 means: Job has never printed.

Example (with job grouping):

```
1040;00001541.0001;16.12.2003;16:31:38;16.12.2003;16:31:55;
1;0;;0;300;300;300;0;300;0;0;;300;1
1040;00001541.0002;16.12.2003;16:31:48;16.12.2003;16:32:05;
1;0;;0;300;300;300;0;300;0;0;;300;1
1040;00001541.0003;16.12.2003;16:31:58;16.12.2003;16:32:11;
1;0;;0;300;300;300;0;300;0;0;;300;1
1040;00001541;16.12.2003;16:31:38;16.12.2003;16:32:11;1;0;;0
;900;900;900;0;900;0;0;;900;1
```

Example for the pagecounter:

Infopages=3 logical pages=300

pages=303 for oneup (300 + 3)
pages=153 for 2up (150 + 3)
pages=78 for 4up (75 + 3)

For SIMPLEX: fpages=303 bpages=0
For DUPLEX: fpages=153 bpages=150

Record Type 1041

This record is written for every job and indicates the used input and output bins.

Parameter	Type	Length (max.)	Description
1041	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
inum	NUM	2	number of used input bins
onum	NUM	2	number of used output bins

Example (with job grouping):

```
1041;00001541.0001;1;1
1041;00001541.0002;1;1
1041;00001541.0003;1;1
1041;00001541;1;1
```

Record Type 1042

This record is written for every used input bin and gives detailed information about it.

Parameter	Type	Length (max.)	Description
1042	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
ibin	NUM	5	input bin number
sheets	NUM	8	number of sheets from this input bin
swid	NUM	4	width of page (1/6 inch)
slen	NUM	4	length of page (1/6 inch)

Example (with job grouping):

```
1042;00001541.0001;1;300;88;72
1042;00001541.0002;1;300;88;72
1042;00001541.0003;1;300;88;72
1042;00001541;1;900;88;72
```

Record Type 1043

This record is written for every used output bin and gives detailed information about it. For each output bin a new record is written.

Parameter	Type	Length (max.)	Description
1043	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
obin	NUM	5	output bin number
sheets	NUM	8	number of sheets for this output bin (not used at the moment)

Example (with job grouping):

```
1043;00001541.0001;1;
1043;00001541.0002;1;
1043;00001541.0003;1;
1043;00001541;1;
```

3.5.8 PJL/PCL Backend

Record Type 1052

This record is written by the PJL/PCL Backend for every job and contains the basic information about the actually printed pages and the time it took to print the job.

Parameter	Type	Length (max.)	Description
1052	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
startdate	TEXT	10	data (format dd.mm.yyyy) when printout or job was started
starttime	TEXT	8	time (format hh:mm:ss) when printout or job was started
enddate	TEXT	10	data (format dd.mm.yyyy) when printout or job was ended
endtime	TEXT	8	time (format hh:mm:ss) when printout or job was ended

simplex	NUM	1	reserved
duplex	NUM	1	reserved
color	NUM	1	reserved
endstatus	NUM	1	status after printout was ended 0…job competed without errors 1… job competed with errors 2…job interrupted by operator 3…job processing interrupted by server error
pages	NUM	8	JOB: summary of the number of printed pages (incl. info pages) FILE: summary of the number of printed pages (incl. info pages)

Example for a job:

```
1052;00001431;02.02.2004;17:33:58;02.02.2004;17:34:34;;;;1;1
53;
```

Example for a file:

```
1052;00001431.0001;02.02.2004;17:33:58;02.02.2004;17:34:34;;
;;1;153;
```

3.6 PRISMAproduction Input Module

3.6.1 Host Download

Record Type 1110

This record is written after a finished transmission of a Host Download print job.

Parameter	Type	Length (max.)	Description
1110	NUM	4	record type
account_id	NUM	8.4	accounting-ID
client	TEXT	64	hostname or IP of the server to which the job was transferred
port	TEXT	17	Connection Port
default_ticket	TEXT	128	Default Ticket for job submission
afp_trim	TEXT	5	trimming of AFP records enabled (true | false)
pws	TEXT	5	print while spooling enabled (true | false)
program	TEXT	128	Program name for job submission
jobname	TEXT	8	Host Jobname
jobqueue	TEXT	1	Host Jobqueue
Jobid	TEXT	8	Host Jobid
mapped_queue	NUM	3	mapped PRISMA jobqueue if enabled
formname	TEXT	8	Spool Formname
byte_count	NUM	12	Spooled Bytes of this printjob
startdate	TEXT	10	date (format dd.mm.yyyy) when receive job was started
starttime	TEXT	8	time (format hh:mm:ss) when receive job was started
enddate	TEXT	10	date (format dd.mm.yyyy) when receive job was ended
endtime	TEXT	8	time (format hh:mm:ss) when receive job was ended
status	TEXT	12	Status of received job e.g. successful, unsuccessful
host_ip	TEXT	15	IP-Address from host

Example: `1110;00000514;lx008571.ops.de;5401;Download.TIC;false;false;/u/prismapro/cfg/jobinput/bin/pjm;$#§FOE2;G;JOB00607;7;$#§FOE2;155533;30.07.2002;15:58:23;30.07.2002;15:58:23;successful;192.168.5.1`

3.6.2 LP

Record Type 1120

This record is written after a finished transmission of a LP print job.

Parameter	Type	Length (max.)	Description
1120	NUM	4	record type
account_id	NUM	8.4	accounting-ID
client	TEXT	64	hostname or IP of the PRISMA server, where the job was received.
port	TEXT	17	Connection Port
default_ticket	TEXT	128	Default Ticket for job submission
pws	TEXT	5	print while spooling enabled (true | false)
program	TEXT	128	Program name for job submission
jobname	TEXT	8	Host Jobname
jobclass	TEXT	1	Host Jobclass
Jobid	TEXT	8	Host Jobid
formname	TEXT	8	Spool Formname
byte_count	NUM	12	Spooled Bytes of this printjob
startdate	TEXT	10	date (format dd.mm.yyyy) when receive job was started
starttime	TEXT	8	time (format hh:mm:ss) when receive job was started
enddate	TEXT	10	date (format dd.mm.yyyy) when receive job was ended
endtime	TEXT	8	time (format hh:mm:ss) when receive job was ended
status	TEXT	12	Status of received job e.g. successful, unsuccessful
host_ip	TEXT	15	IP-Address from host

Example:

```
1120;12345678;lx001234.ops.de;1234;LP.TIC;false;/u/pris-
mapro/cfg/jobinput/bin/
pjm;Jobname1;3;JOB00607;STD;155533;30.07.2002;
15:58:23;30.07.2002;15:58:23;successful;192.168.5.1
```

3.6.3 HotDir

Record Type 1130

This record is written after a finished transmission of a Hot Directory print job.

Parameter	Type	Length (max.)	Description
1130	NUM	4	record type
account_id	NUM	8.4	accounting-ID
client	TEXT	64	hostname or IP of the server to which the job was transferred
name	TEXT	17	Hot Directory name
default_ticket	TEXT	128	Default Ticket for job submission
program	TEXT	128	Program name for job submission
filename	TEXT	128	received file name
byte_count	NUM	12	Spooled Bytes of this print job
submit_date	TEXT	10	date (format dd.mm.yyyy) when job was submitted
submit_time	TEXT	8	time (format hh:mm:ss) when job was submitted
status	TEXT	12	Status of received job e.g. successful, unsuccessful

Example:

```
1130;12345678;lx001234.ops.de;oce_germany;Hotdir.TIC;/u/
prismapro/cfg/jobinput/bin/
pjm;Jobname1;155533;30.07.2002;15:58:23;successful
```

3.7 LCDS-Module Specific Records

For the LCDS-Module following additional record types are currently being defined. LCDS record types always start with the id 12, so 1200 would be a valid record type for e.g. the X-Filter

3.7.1 Xfilter

.Record Type 1200 (LCDS report information)

This record type logs the general information for the report. This record is written for each report at the end of the report. The last department specified by DJDE DEPT is valid for the whole report. The record 1200 has a maximum length of 155 characters.

Parameter	Type	Length (max.)	Description
1200	NUM	4	record type
account_id	NUM	8	accounting-ID for this report (identical to ODS- and Spool-ID)
start_jde	TEXT	6	JDE name at the time of creation of the job (starting JDE)
start_jdl	TEXT	6	JDL name at the time of creation of the job (starting JDL)
collate	TEXT	1	Y/N: COLLATE start setting
hrptna	TEXT	16	Host Report Name from BANNER/RSTACK statement, may be empty e.g. if no HRPTNA is specified in JDE.
jde	TEXT	6	JDE name at the end of this report (last DJDE JDE)
jdl	TEXT	6	JDL name at the end of this report (last DJDE JDL)
dept	TEXT	31	department valid for this report (last DJDE DEPT); if no DEPT is given, the default is "<jdl>:JDL"
inrec	NUM	10	Number of counted input records of the report
indjde	NUM	10	Number of counted DJDE-records of the report
dpsh	NUM	9	Number of sheets converted for department
dppa	NUM	9	Number of pages converted for department

Parameter	Type	Length (max.)	Description
pagesize_1	NUM	2.2	1st edge in inches inclusive edge-marking, Long-Edge-Feed: 2nd edge is longer; Short-Edge-Feed for longer edge > 14.33 inch: 1st edge is longer
pagesize_2	NUM	2.2	2nd edge in inches inclusive edge-marking, Long-Edge-Feed: 2nd edge is longer; Short-Edge-Feed for longer edge > 14.33 inch: 1st edge is longer
job_account_id	NUM	8	Accounting-ID used by the Ifilter for this job.

Example:

DIN A4:

```
1200;12345679;DFLT;DFAULT;Y;Report Name 8/15;BILLS;ONLINE;
Department 4711;12001;745;120;195;8.45;11.85;12345678
```

DIN A3 with default department

```
1200;12345680;DFLT;DFAULT;Y;;BILLS;ONLIN;
ONLIN:JDL;12001;745;120;195;16.61;12.02;12345678
```

Record Type 1201 (additional report information)

This record type logs the information extracted from the input file. This record is written for each report at the end of report, if the PrintLCDS.JCF parameter Jobname Cfg is set to a value that does not begin with a 0 (zero).

The record 1201 has a maximum length of 151 characters.

Parameter	Type	Length (max.)	Description
1201	NUM	4	record type
account_id	NUM	8	accounting-ID for this report (identical to ODS- and Spool-ID)
hjobno	TEXT	6	Host job number from BANNER statement, may be empty e.g. if no HJOBNO is specified in JDE.
acctinfo	TEXT	64	account info from RSTACK statement, may be empty e.g. if no ACCTINFO is specified in JDE.
file_id	TEXT	65	File ID from the Tape Header1 label, or Hard disk file name as filled in PrintLCDS window (without directory path). Semicolons in the file_id will be replaced by underscores.

Example:

Hard disk file:

```
1201;12345679;123456;Accounted to 08/15;testdata.dat
```

Tape file:

```
1201;12345679;123456;Accounted to 08/15;TESTDATA
```

Record Type 1202 (additional job information)

This record type logs the information extracted from the input file. This record is written for each job at the end of the Xfilter process, if the job is printing a Hard Disk File, and if the PrintLCDS.JCF parameter Jobname Cfg is set to a value not beginning with 0 (zero).

The record 1202 has a maximum length of 255 characters.

Parameter	Type	Length (max.)	Description
1202	NUM	4	record type
account_id	NUM	8	accounting-ID identical to job_account_id of record type 1200
HD_fullname	TEXT	241	Hard disk file name as filled in PrintLCDS window (with directory path)

Example:

```
1202;12345678;/u/prismapro/data/custom/customer_a/alit-
tlebitlongname/testdata.dat
```

3.7.2 LCDS-Driver (jesfilter)

Record Type 1210

For ≥ V3.04: A record of this type is written by the jesfilter each time a job step is converted.

Parameter	Type	Length (max.)	Description
1210	NUM	4	record type
account_id	NUM	8.4	accounting-ID
pages	NUM	10	approx. number of pages (number of skip-to-channel-1's)

Example: `1210;12345678.0001;9876`

3.8 PRISMAproduction Backends (≥ V3.02)

3.8.1 FTP-Backend

Record Type 1600

This record is written after job is finished (third column is the field length, x=unlimited).

Parameter	Type	Length (max.)	Description
1600	NUM	4	record type
account ID	NUM	8	account ID (= Job-ID)
startdate	TEXT	10	date (format dd.mm.yyyy) when printout was started
starttime	TEXT	8	time (format hh:mm:ss) when printout was started
enddate	TEXT	10	date (format dd.mm.yyyy) when printout was ended
endtime	TEXT	8	time (format hh:mm:ss) when printout was ended
endstatus	NUM	1	status after printout was ended 0...job completed without errors 1...job completed with errors 2...job interrupted by operator 3...job processing interrupted by server error
jobname	TEXT	x	jobname
jobopies	NUM	x	jobcopies
sendsize	NUM	x	size of the transferred job in bytes
rate	TEXT	x.3	transmission rate for the job in kbytes

Example:

```
1600;00002844;20.02.2004;12:18:05;20.02.2004;12:18:06;0;WHBA
C030;1;292105;409.111
```

Record Type 1601

This record is written after job is finished (third column is the field length, x=unlimited)

Parameter	Type	Length (max.)	Description
1601	NUM	4	record type
account ID	NUM	8	account ID (= Job-ID)
hostname	TEXT	x	destination hostname
directory	TEXT	x	destination directory
ftp-server	TEXT	x	ftp server response
ftp-os	TEXT	x	ftp server operating system

Example:

```
1601;00002844;localhost;/u/tmp;ProFTPD 1.2.6 Server (ProFTPD
Default Installation) [lx001234.ops.de];UNIX Type: L8
```

Record Type 1602

This record is written after transmission of each print file (third column is the field length, x=unlimited)

Parameter	Type	Length (max.)	Description
1602	NUM	4	record type
account ID	NUM	8.4	account ID (= Job-ID)
startdate	TEXT	10	date (format dd.mm.yyyy) when printout was started
starttime	TEXT	8	time (format hh:mm:ss) when printout was started
enddate	TEXT	10	date (format dd.mm.yyyy) when printout was ended
endtime	TEXT	8	time (format hh:mm:ss) when printout was ended
endstatus	NUM	1	status after printout was ended 0...job completed without errors 1...job completed with errors 2...job interrupted by operator 3...job processing interrupted by server error
filename	TEXT	x	Destination file name
filecopies	NUM	x	filecopies
filepages	NUM	x	original job size in pages (if known) for one file- and jobcopy (opages)
filesize	NUM	x	size of the file in bytes
sendsize	NUM	x	amount of the transferred bytes
rate	TEXT	x.3	transmission rate in kbytes per second (3 fractional digits)

Example:

```
1602;00002844.0001;20.02.2004;12:18:05;20.02.2004;12:18:06;
0;00002844/0001_WHBAC030.JOB07045;1;506;291818;291818;
1274.314
```

Record Type 1603

This record is written after transmission of a generated oct-file (third column is the field length, x=unlimited)

Parameter	Type	Length (max.)	Description
1603	NUM	4	record type
account ID	NUM	8	account ID (= Job-ID)
startdate	TEXT	10	date (format dd.mm.yyyy) when printout was started
starttime	TEXT	8	time (format hh:mm:ss) when printout was started
enddate	TEXT	10	date (format dd.mm.yyyy) when printout was ended
endtime	TEXT	8	time (format hh:mm:ss) when printout was ended
endstatus	NUM	1	status after printout was ended 0...job completed without errors 1...job completed with errors 2...job interrupted by operator 3...job processing interrupted by server error
filename	TEXT	x	name of the generated and transmitted oct-file.
filecopies	NUM	x	filecopies
filepages	NUM	1	always 0
filesize	NUM	x	size of the file in bytes
sendsize	NUM	x	amount of the transferred bytes
rate	TEXT	x.3	transmission rate in kbytes per second (3 fractional digits)

Example:

```
1603;00002844;20.02.2004;12:18:05;20.02.2004;12:18:06;0;0000
2844.oct;1;0;287;287;143.500
```

3.8.2 Postscript Backends

For ≥ V3.10.01: The record 162x is written by the Xrx DocuTech Driver and/or by the Oce Postscript Driver.

Record Type 1620

This record is written for every job and contains the basic information about the actually printed files. (Third column is the field length, x=unlimited)

Parameter	Type	Length (max.)	Description
1620	NUM	4	record type
account_id	NUM	8	accounting-ID without sub-id: Record belongs to the whole JOB
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
printer	TEXT	x	name of printer on which the job was printed
startdate	TEXT	10	date (format dd.mm.yyyy) when printout or job was started
starttime	TEXT	8	time (format hh:mm:ss) when printout or job was started
enddate	TEXT	10	date (format dd.mm.yyyy) when printout or job was ended
endtime	TEXT	8	time (format hh:mm:ss) when printout or job was ended
simplex	NUM	1	simplex mode was used 0...no 1...yes (e. g. simplex and rsimplex)
duplex	NUM	1	duplex mode was used 0...no 1...yes (e. g. normal duplex, tumble duplex and rzigzag normal)
colormode	NUM	1	0 …without color, 1 … with color in the datafile
endstatus	NUM	1	status after printout was ended 0...job completed without errors 1...job completed with errors 2...job interrupted by operator 3...job processing interrupted by server error
pages	NUM	x	JOB: summary of the number of pagesides printed FILE: number of pagesides printed for all filecopies and one jobcopy
opages	NUM	x	JOB: summary of the original job size in pages FILE: original job size in pages for one file- and jobcopy
fpages	NUM	x	JOB: summary of the number of front pages FILE: number of front pages for all filecopies and one jobcopy

bpages	NUM	x	JOB: summary of the number of back pages FILE: number of back pages for all filecopies and one jobcopy
sheets	NUM	x	JOB: summary of the number of sheets (incl. interposer) FILE: number of sheets (incl. interposer) for all filecopies
interposer-sheets	NUM	x	JOB: summary of interposer sheets. FILE: number of interposer sheets for all filecopies
copy	NUM	x	JOB: jobcopy FILE: filecopy
offsets	NUM	x	JOB: summary of the number of offsets in the output tray FILE: number of offsets in the output tray
printcount	NUM	x	count, how many time times, the job has been printed (start with 1).
medianum	NUM	x	JOB: summary of the number of used medias FILE: number of used medias

Example:

```
1620;00000344.0001;My2110;08.03.2006;11:19:00;08.03.2006;11:
19:08;1;0;0;0;93;31;93;0;93;0;3;0;27;1;
1620;00000344.0001;My2110;08.03.2006;11:19:00;08.03.2006;11:
19:08;1;0;0;0;93;31;93;0;93;0;3;0;27;1;
1620;00000344.0002;My2110;08.03.2006;11:19:05;08.03.2006;11:
19:08;1;0;0;0;155;31;155;0;155;0;5;0;27;1;
1620;00000344.0002;My2110;08.03.2006;11:19:05;08.03.2006;11:
19:08;1;0;0;0;155;31;155;0;155;0;5;0;27;1;
1620;00000344;My2110;08.03.2006;11:18:52;08.03.2006;11:19:08
;1;0;0;0;496;124;496;0;496;0;2;0;27;2;
```

Record Type 1621

This record is written for every used media. (Third column is the field length, x=unlimited):

Parameter	Type	Length (max.)	Description
1621	NUM	4	record type
account_id	NUM	8.4	accounting-ID with sub-id: Record belongs to one data FILE
sheets	NUM	x	summary of the number of sheets for this media (for one jobcopy)
name	TEXT	x	medianame
weight	TEXT	x	weight of the page in gram

color	TEXT	x	color of the page
size_x	TEXT	x	x-size of the page in mm
size_y	TEXT	x	y-size of the page in mm
order_count	TEXT	x	number of the order_count
perforation	TEXT	x	number of the perforation of a page
type	TEXT	x	Type as string
interposer	NUM	x	0 ... normal media, 1 ... interposer media

Example:

```
1621;00000344.0001;93;A4g;80;Green;210;297;;0;Plain;0;
1621;00000344.0001;93;A4g;80;Green;210;297;;0;Plain;0;
1621;00000344.0002;155;A4g;80;Green;210;297;;0;Plain;0;
1621;00000344.0002;155;A4g;80;Green;210;297;;0;Plain;0;
```

3.9 User-specific Extensions

Record Types 99xx

The record types beginning with 9900 are reserved for user-specific extensions. These may be used to write user-specific account data from user-specific programs and/or shell scripts. The record structure is freely definable as long as the basic rules specified in this document are followed

Parameter	Type	Length (max.)	Description
9900	NUM	4	record type
account_id	NUM	8.4	accounting-ID
...	...	...	...
...	...	...	...

3.10 PRISMAproduction internal accounting workflow

3.10.1 Accounting-IDs

Print jobs submitted from the PJM

The PJM requests the accounting-ID from the account server and writes it into the job ticket. Thus each subsequent processing module has access to a valid ID.

The program which actually submits the print job to the spool must supply this accounting-ID as a parameter for the spool.

Print jobs submitted from the UI-Manager

Print jobs which are started from the UI-Manager using a start_job-call to the communicator carry their accounting-ID in the JCF. The UI-Manager is responsible for requesting the accounting-ID from the account server and writing it into the "Header"-section of the JCF before actually starting the job. All processes in the process chain must read the accounting-ID from the JCF if they have to write accounting information. The parameter name in the JCF is "AccountID".

The program which actually submits the print job to the spool (usually the program "spooljob") must supply this accounting-ID as a parameter for the spool.

All other Print jobs

All jobs which are accepted by the spool without an accounting-ID will get this ID from the spool itself. The spool requests the ID from the account server and writes it into the list of job parameters. Thus the printing process (AFP2IPDS Backend) can use this ID for writing its account information. For each of these jobs the record type 1030 is written which contains the basic submission information which is otherwise provided by the initial process (e.g. PJM).

3.10.2 Accounting Command Line Interface

The accounting command line utility is placed in the directory `/u/prismapro/bin/` with the name "ppacc". This command is a full replacement for the old "accnum" and "accmsg" utilities.

ppacc -i get a new accounting ID from the server
This is a replacement for **accnum** command, but **accnum** stays available as an alias for **ppacc -i**.

ppacc -a write one accounting record
This is a replacement for **accmsg** command, but **accmsg** stays available as an alias for **ppacc -a**

This command line interface is also the officially supported external interface for writing accounting data from external (non-PRISMA) programs or scripts.

Command line parameters:

Usage: `ppacc [options] parameters`

General options:

`-h`		print help information	
`-v`		print version number	
`-s`	`SRV`	server, dealing with accounting	(def.: auto-detected)
`-un`	`USERNAME`	name of the user	(def.: current user)
`-up`	`PASSWD`	user password	
`-o`	`TIMEOUT`	set timeout for the command in seconds	
`-y`		print error information on stderr	

Get new ID usage: `ppacc <option>`

`-i`		return new PRISMA job id

Send record usage: `ppacc <option> RECORDS`

`-a`	`RECORD(s)`	send new RECORD(s) to accounting server

Delete usage: `ppacc <option> DAYS`

`-r`		remove outdated entries and files, which are older than DAYS
	`DAYS`	day count (0: delete all)

4 Command Line Interface

The components

- Job Editor
- Print Job Manager
- Spsprt-To-Job Ticket Converter
- Printer Pool
- Administrator
- Status Information
- Message Information
- Accounting

comprise the PRISMAproduction command line interface. With this interface, you can control and manage ongoing job processing and obtain details on the status and results of processing from remote computers which do not support the X Window System PRISMA UI. The command line interface is also useful for automation tasks using shell scripts.

A few special functions for spool controlling are only accessible via command line.

4.1 Job Editor

The job editor is started with the following command:

```
spsprt [{Option} {Parameter}] {Filename}
```

This command submits the specified file for printing, using the options and parameters you enter. The following sequence of actions is triggered when you submit a print job:

- The print file is copied to the spool directory;
- A job file with the selected job parameters is generated and entered in the spool directory;

- The spool system is informed that a new print job is pending;
- A message containing the job's spool ID is issued to inform the user that the job has been submitted successfully. This ID can be used to track the job as it is processed and query its status.

If you enter the command without any options, you are shown general information on the spsprt command.

Usage: 'spsprt [-<option> <param>] <print file>'

Possible options:

```
[-C                path for the configuration directory, e.g. <u/prismapro/cfg/
spsprt>]
[-option           <options file>]
[-file             <printfile>],  max. length: 128
[-erase            no | yes]
[-filetype         sps | pcl]
[-title            <edit-operand>],  max. length: 80
[-copies           <digital edit operand>],  max. length: 5
[-range            <edit-operand>],  max. length: 15
[-header           no | yes]
[-dest             | single 300]
[-class            1 | 2 | 3]
[-type             255 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 10]
[-x2up             255 | 0 | 1 | 2 | 4]
[-spslib           /u/prismapro/resource/system/240dpi]
[-spslib           /u/prismapro/resource/system/300dpi]
[-spslib           /u/prismapro/resource/system/600dpi]
[-form             std/std1]
[-formdef          f10101 | f10101la | f10101pa | f1dpage | f1dpage1]
[-pagedef          p1std3 | dummy]
[-fonts            x0gs10y]
[-fonts            x0gs12y]
[-fonts            x0gs15y]
[-fonts            x0gt10y]
[-fonts            x0gt12y]
[-fonts            x0gt15y]
[-fonts            x0cr10y]
[-fonts            x0ce10y]
[-fonts            x0ce12y]
[-fonts            x0ci10y]
[-usrlib           <edit-operand>],  max. length: 256
[-dpi              0 | 240 | 300 | 600]
[-formdef          <edit-operand>],  max. length: 8
```

[-pdscc	01 | 02 | 04]
[-pagedef	<edit-operand>], max. length: 8
[-pdstrc	00 | 01 | 02]
[-pages	<digital edit operand>], max. length: 8

Overview of options and parameters

Command	Option	Parameter	File name	
spsprt	-C	<directory>		Directory containing config files.
spsprt	-option		<options file>	Name of parameter file
spsprt	-jobname	<freefor text <32chars>		User-supplied name for this job
spsprt	-file		<print file>	Name of print file (do not use - see note on previous page 204)
spsprt	-filetype	afp (or 'sps') pcl pdf ps		AFP printfile type PCL printfile type PDF printfile type Postscript printfile type
spsprt	-header	yes no		Print information pages Do not print information pages
spsprt	-class	1 – 999		Job queue
spsprt	-copies	<numeric edit operand>		Number of copies (max. 32767)
spsprt	-range	X-Y		Printing range
spsprt	-dest	NAME		Name of the printer
spsprt	-dpi	0 240 300 600		Default resolution 240 dpi 300 dpi 600 dpi
spsprt	-pdscc	01 02 04		ASA carriage control IBM carriage control OCE carriage control
spsprt	-pdstrc	00 01 02		Interpretation of TRC byte: No TRC IBM OCE

Command	Option	Parameter	File name	
spsprt	-type	255 0 1 2 3 4 5 6 7 8 9 10		Setting from FormDef Default Simplex Normal duplex Tumble duplex Rotated simplex Rotated normal duplex Rotated tumble duplex Normal zigzag Tumble zigzag Rotated normal zigzag Rotated tumble zigzag
spsprt	-x2up	255 0 1 2 4		Default Off Normal two-up, sequence 1-2, 3-4, ... Identical copies Invers two-up: Two diff. Pages side-by-side, sequence 2-1, 4-3, ...
spsprt	-spslib			List of library directories
spsprt	-form			Form name. The operator will be notified to supply this form when the print job is started.
spsprt	-formdef			Standard form definitions (can be added as required)
spsprt	-pagedef			Standard page definitions
spsprt	-fonts			Standard fonts that can be loaded into the printer prior to job
spsprt	-pagesf			Count of pages in the job.
spsprt	-title	<edit-oper-and>		Header page title

Example

for submitting a print file with default print settings

```
spsprt   printfile
```

Note: Using the –file option instead of the normal file name parameter (last parameter of spsprt command) is discouraged because the standard plausibility check is not executed for the last parameter of the spsprt command. Such a mistake in the last parameter will not be found by spsprt if –file is used. This option is only supported for compatibility reasons.

4.2 Spsprt-to-JobTicket Converter

This program can be used for converting 'old' (x)spsprt job files into job ticket files which can be used by the Print Job Manager.

The conversion is started with the same command line options as the normal spsprt command. The converted job ticket is printed on the screen and can be redirected to a file using the '>' character:

```
job2tic -option  spsprt.jobfile  >  jobticket.TIC
```

4.3 PrinterPool

General Usage

Installation path: `/u/prismapro/bin/prtpool`

This program is a replacement for the old 'spladmin' program. The major enhancement of the new tool is the ability to deal with all printers in the cluster, it is not limited to printers on a certain printserver.

The options of the program can be divided into 4 major function blocks:

- Display or modify printer parameters
- Display or modify spool daemon parameters
- Display all configured form names
- Display all configured job queues or import a class.cfg file from PRISMAproduction version 2.xx installations.

Additionally there are some options, which may be required for every major function.

Common Options

Usage: prtpool -m *ServerName* -n *PortNumber* -un *UserName* -up *UserPassword* ...

Command line option	Description
-h or -help	Print help text
-v or -ver	Print program version information
-m *MasterName*	*MasterName* is the central master of the server cluster. If not specified, the program tries to detect the master automatically (using the `/u/prismapro/bin/getmaster` utility).
-n *PortNumber*	*PortNumber* is the number of the socket port, which is used for the socket communication. If not specified, the port 1207 is used as default.
-un *UserName*	*UserName* is the name of one user, which is known by the PRISMAproduction software (not by the operating system) If not specified, the program uses automatically the name of that user, who executes the program. If specified, the user has to specify the password for that *UserName* too.
-up *UserPassword*	*UserPassword* is the password, which is accepted by the explorer server for the selected *UserName*.

Note: Username and password are usually not required, if the command is executed on a server inside a cluster or from a trusted host.

Jobqueue Options

This option displays the configures jobqueues or import an old jobclass definition file.

Usage: prtpool -q[i] [class.cfg-file]

Command line option	Description
-q	Display all configured jobqueues in the system. The output is displayed on stdout in following format: `QueueNumber <tab> Description <tab> LockStatus` Each queue definition is displayed in a separate line.
-qi *Filename*	Imports the file in class.cfg format from a PRISMAproduction V1.x or V2.x installations.
Examples	prtpool –q -> display all configured queues prtpool –qi ./class.cfg -> import class.cfg file from current directory.

Form Options

This option displays all configured form names in the system.

Usage: prtpool -f

Command line option	Description
-f	Display all configured form names in the system. The output is displayed on stdout, where each line contains one form name.
Examples	prtpool –f -> display all form names

Spool Daemon Options

This option is used for controlling the spool daemons on each print server in the cluster.

Usage:
prtpool -d [Command] - local spool daemon commands
prtpool -s [Server] [Command] - all spool daemons commands
prtpool -l [Server] [LogLevel] - loglevel commands
prtpool -S [Server] [Columns] - scripting options

Command line option	Description
-d	This is a special option for dealing with the spool daemon of the server, where the command is running on. For commands for spool daemons on selectable servers please use the ”-s” option of prtpool command. If a *command-option* is not specified, prtpool returns the status of the local spool daemon. The output contains the information, if the spool daemon is active or not and if the spool and print service of the daemon is active or not.
-d *Command*	The *command-option* has to be specified, if an action should be executed on the local spool daemon. *start* This command enables the spool daemon on local pc. *stop* This command stops the spool daemon on local pc. *restart* This command stops an starts the local spool daemon. *halt* This command halts the spool and print service, without stopping the spool daemon. *continue* This command reenables the spool and print service after the halt command.

Command line option	**Description**
Examples	prtpool –d stop -> stop spool on local machine prtpool –d restart -> stop and start spool daemon on local machine
-s *Server*	This is the general option for dealing with all spool daemons of the whole server cluster. If *server-name* is not specified, prtpool returns the status of all spool daemons in the cluster. The output contains 3 columns per line: hostname daemon-status halt-status The displayed values start always on a dedicated column, empty fields a filled with blanks. The output can be restricted to one server, if the server-name is entered after the '–s' option.
-s *Server* *Command*	If a *command-option* should be used on a special spool daemon, the *server-name* has to be specified before. *start* This command starts the spool daemon on specified pc. *stop* This command stops the spool daemon on specified pc. *restart* This command stops an restarts the daemon on specified pc.
Examples	prtpool –s <servername> restart -> restart spool daemon on dedicated server
-l *Server*	This is the general option for handling traces of all spool daemons in the cluster. If *server-name* is not specified, prtpool returns the status of all spool daemons in the cluster. The output contains 3 columns per line: hostname log-level The displayed values start always on a dedicated column, empty fields a filled with blanks. The output can be restricted to one server, if the server-name is entered after the '-l' option.

Command line option	Description
-l *Server* *Log-Level*	This is the general option for changing traces-levels of all spool daemons in the cluster. If a new log-level should be set, the server-name has to be specified before. The minimum log-level is 0 (no logging, only errors are logged).The maximum log-level is 5. The logs are written into files in the directory ”/u/prismapro/diag/spool/ ” on the server, where the spool daemon is running.
Examples	prtpool –l -> show all loglevels prtpool –l <server-name> 5 -> set spool-loglevel to 5 on <servername>
-S *Server* *Columns*	This option is useful for script programming, where the requested spool information should be processed for further purposes. If server-name is not specified, prtpool displays the requested column infos of all spool daemons in the cluster. The requested infos must be specified by colon (',') separated column names. The output is written always as tab-separated entries for requested columns, in one line per server. Following column names are allowed: "NAME Name of the server. "STATUS Status of the spool daemon. "ENABLED Status of the server. "LOGLEVEL Loglevel of the spool daemon. ***The column names and the result strings, which are displayed as the command output for specified columns, can be changed or extended in future PRISMAproduction releases.***
Examples	prtpool -S NAME,LOGLEVEL -> display the hostname and loglevels of all spool daemons.

Note: The server-names have to be specified in the same format, as it is stored in the PRISMAproduction configuration database.

Printer Options

This is the general option for dealing with all configured printers. Because of the uniqueness of the printer names, no server name has to be specified here.

Usage: prtpool -p [Printer] [Command] [Option] -printer commands
prtpool -P [Printer] [Columns] -scripting options

Command line option	Description
-p Printer	If no other options are specified, prtpool returns the status of all printers in the system. The output consist of one line pre printer and the format looks like the old output from 'spladmin' command: prt-name dpi data-type jobqueues form status The displayed values start always on a dedicated column, empty fields a filled with blanks. The output can be restricted to one printer, if the *printer-name* is entered after the '–p' option.
-p *Printer* **start**	The *start* command activates a printer with the name '*Printer'*.
-p *Printer* **stop**	The *stop* command deactivates a printer with the name '*Printer'*.
-p *Printer* **reset**	The *reset* command stops a printer with the name '*Printer'*, if it not reacts to the normal stop command.
-p *Printer* **jobint** *Option*	The *jobint* command interrupts the current printing job on printer '*Printer'*. The job will stay in the print queue with 'Interrupted' status. Reprints of such job will start on the interrupted position. Following options are allowed: `--` -> interrupt immediately (no option parameter) `fc` -> interrupt after finishing current printfile copy. `jc` -> interrupt after finishing current job copy.
-p *Printer* **jobcan** *Option*	The *jobcan* command interrupts the current printing job on printer '*Printer'*. The job will stay in the print queue with 'Canceled' status. Reprints of such job will always start from the beginning of the job. Following options are allowed: `--` -> cancel immediately (no option parameter) `fc` -> cancel after finishing current printfile copy. `jc` -> cancel after finishing current job copy.
-p *Printer* **jobdel** *Option*	The *jobdel* command cancels the current printing job on printer '*Printer*' and deletes it from the print queue. Following options are allowed: `--` -> cancel immediately (no option parameter) `fc` -> cancel after finishing current printfile copy. `jc` -> cancel after finishing current job copy.

Command line option	Description
-p *Printer* **queue** *Option*	The *queue* command shows, adds, removes or sets the assigned job-queues on a certain printer. `queue` -> show assigned queues `queue`+ number(s) -> add queues `queue`- number(s) -> remove queues `queue` number(s) -> set only this queues The option specifies one or more (',' separated) jobqueue numbers (e.g. 1,5,7) . Queue numbers are restricted to 1-999.
-p *Printer* **color** *Option*	The *color* command shows, adds, removes or sets the assigned color verification ids on a certain printer. `color` -> show assigned color ids `color`+ number(s) -> add color ids `color`- number(s) -> remove color ids `color` number(s) -> set only this color ids The option specifies one or more (',' separated) color ids.
-p *Printer* **npro** *Option*	The *npro* command shows or sets the NPRO time on a certain printer. Without an option the command displays the npro time of the printer. If an option is specified (numeric value in seconds), the new value is set for the printer.
-p *Printer* **form** *Option*	The *form* command shows or sets the form name for job selection on a certain printer. Without an option the command display the current selected form name. If an option is specified, the new form-name is set for the printer.
-p *Printer* **checkform** *Option*	The *checkform* command shows or sets the switch for job selection restriction on form-names for the selected printer. Without an option the command displays the current status for the form-name switch for jobselection: `0/0` -> form check and form sequencing are off. `1/1` -> form check and form sequencing are enabled. `1/0` -> form check enabled, but form sequencing is disabled. Following options are allowed to set: `off` -> form name is not used for jobselection `on` -> form name is used for jobselection `seq` -> form name is used for jobselection, but an automatic switch to a new form is allowed, if there are no more jobs with the same form name.

Command line option	Description
-p *Printer* **checkres** *Option*	The *checkres* command shows or sets the switch for job selection restriction on resolution for the selected printer. Without an option the command displays the current status for the resolution switch for jobselection (0=off, 1=on). Following options are allowed to set: `off` -> resolution is not used for jobselection `on` -> resolution is used for jobselection
-p *Printer* **checkname** *Option*	The *checkname* command shows or sets the switch for job selection restriction on printer-name for the selected printer. Without an option the command displays the current status for the printer-name switch for jobselection (0=off, 1=on). Following options are allowed to set: `off` -> printer-name is not used for jobselection `on` -> printer-name is used for jobselection
-p *Printer* **checkqueue** *Option*	The *checkqueue* command shows or sets the switch for job selection restriction on jobqueues for the selected printer. Without an option the command displays the current status for the jobqueue switch for jobselection (0=off, 1=on). Following options are allowed to set: `off` -> jobqueues are not used for jobselection `on` -> jobqueues are used for jobselection
-p *Printer* **checkcolor** *Option*	The *checkcolor* command shows or sets the switch for job selection restriction on printer-name for the selected printer. Without an option the command displays the current status for the printer-name switch for jobselection (0=off, 1=on). Following options are allowed to set: `off` -> color verification ids are not used for jobselection `on` -> color verification ids are used for jobselection
-p *Printer* **checkdata** *Option*	The *checkdata* command shows or sets the switch for job selection restriction on data-types for the selected printer. Without an option the command displays the current status for the data-types switch for jobselection (0=off, 1=on). Following options are allowed to set: `off` -> data-type is not used for jobselection `on` -> data-type is used for jobselection

Command line option	**Description**
Examples	prtpool –p Dummy start -> start printer 'Dummy' prtpool –p Dummy stop -> stop printer 'Dummy' prtpool –p Dummy queue -> show assigned queues prtpool –p Dummy queue+ 3 -> add 3 to assigned queues list.
-P *Printer Columns*	This option is useful for script programming, where the requested printer information should be processed for further purposes. If printer-name is not specified, prtpool displays the requested column infos of all printers. The requested infos must be specified by colon (',') separated column names. The output is written always as tab-separated entries for requested columns, in one line per printer. Following column names are allowed: NAME Name of the printer STATUS Status of the printer TYPE Basic printer type MODEL Printer model DEVICE Printer output driver FORMAT Data type RESOLUTION Printer resolution SPEED Printer speed DUPLEX Duplex print capability TONERS Toner station count HOST Server name or attached printer JOBID Current job id FORM Current form name QUEUE Selected queues COLORS Selected color verification ids ***The column names and the result strings, which are displayed as the command output for specified columns, can be changed or extended in future PRISMAproduction releases.***
Examples	prtpool -P NAME,STATUS,JOBID,FORM,SPEED -> display the printer name, status jobid, form name and speed of all printers in the cluster.

Possible values for printer STATUS column

ACTIVE	Active
IDLE	Active
INACTIVE	Inactive
STARTING	Starting
DRAINING	Draining
HALTED	Halted
WAITING	Waiting
PRINTING	Printing
PREPARATION	Preparation
PRTDRAIN	Printing + Draining after finishing current job
PRTHALT	Printing + Halt after finishing current job
MESSAGE	Message
NOTREADY	Not Ready
LICENSE	License

Possible values for spool STATUS column

DOWN	Spool inactive
PAUSING	Spool halted
DISABLED	Server inactive
ACTIVE	Active

4.4 Messages

This is the description of all command line parameters of the **ppmsg** utility. It's not a real API, but the command is very useful for shell programming purposes.

4.4.1 General Usage

Installation path: `/u/prismapro/bin/ppmsg`

Required libraries: `/usr/lib/libppmsg.so`

Note: A part of the show_msg program is integrated into the message window.

The options of the program can be divided into 6 major function blocks:

- read message list (default function)
- read details for a special message
- read message counters
- reply open questions
- delete messages
- send messages

Additionally there are some options, which may be required for every major function.

4.4.2 Common Options

Usage: `ppmsg -s` *`ServerName`* `-p` *`PortNumber`* `-un` *`UserName`* `-up` *`UserPassword`* `...`

Command line option	Description
-h or -help	Print help text
-v or -ver	Print program version information
-s *ServerName*	*ServerName* is the name of the server, where the message daemon is running on. If not specified, the program tries to detect the master automatically (using the /u/prismapro/bin/getmaster utility)

Command line option	Description
-p *PortNumber*	*PortNumber* is the number of the socket port, which is used for the socket communication. If not specified, the port 1207 is used as default.
-un *UserName*	*UserName* is the name of one user, which is know by the PRISMA software (not by the operating system) If not specified, the program uses automatically the name of that user, who executes the program. If specified, the user has to specify the password for that *UserName* too.
-up *UserPassword*	*UserPassword* is the password, which is accepted by the explorer server for the selected *UserName*. If not specified, an internal default password is used.
-ul *UserLanguage*	*UserLanguage* is the language, in which message infos are returned. Currently only german ("de") and english ("en") languages are supported. If not specified, the english language is used.
-o *Timeout*	*Timeout* is timeout value in seconds: maximum time to wait on the command reply, before command is aborted.

Note: Username and password are usually not required, if the message is send from a server inside a cluster or from a trusted host.

Examples: `ppmsg -s Master -un root -up .pwroot ...`

4.4.3 Message Counters

Usage: `ppmsg -c[type] [Labels]`

Command line option	Description
-c	This switch returns some general messages counters. 1st counter: MESSAGES: count of received messages in the database. 2nd counter: QUESTIONS: count of open questions in the database. 3rd counter: MSGSBUSY: appears only if the message daemon (msgs) is still reading old messages after restart.
-cm	Returns only the count of messages in the database.
-cq	Returns only the count of open questions.

Command line option	Description	
Labels *(optional, default = 1)*	*Labels*=1:	The output is written in the format: MESSAGES=Value_1 QUESTIONS=Value_2
	Labels=0:	The output is written in the format: Value_1 Value_2

Note: This switch should be the last switch in the command line.

Examples:

`ppmsg -s Master -c`-> print counter with labels
`ppmsg -s Master -cm 0`-> print message counter without labels

Delete certain messages

Usage: `ppmsg -a MessageToken [MessageCount]`

Command line option	Description
-rm *MessageToken*	Use the *MessageToken* as the reference position for an entry in the database. *MessageToken* is an unique identifier for each message entry in the database. *MessageToken* = 0: last message entry *MessageToken* = 1: first message entry
MessageCount (optional, default = 1)	Count of message to delete: = 0 : delete all messages < 0 : delete MessageCount messages before the reference position > 0 : delete MessageCount messages after the reference position

Note: If MessageToken = 0 AND MessageCount = 0 -> all messages are deleted.

Examples:

```
ppmsg -rm 0  -100        -> delete last 100 entries
ppmsg -rm 1   100        -> delete first 100 entries
ppmsg -rm 12345678 50    -> delete first 100 entries starting at token
12345678
```

4.4.4 Send Messages

Usage: `ppmsg -m[w] ModuleName -j[w] JobId MessageId [Insert1] [Insert2] ...`

Command line option	Description
-m[w] *ModuleName* -mt *ModuleName*	*ModuleName* is the name of the module, which will be displayed in the module name column of the message window. If the *MessageId* identifies a message, which requires an answer from the operator, the 'w' switch extension allows the program to return only after the question is answered. Send message and **return the created message token**, which can be used for polling for an answer, if the message is question type entry.
-j[w] *JobId* -jt *JobId*	*JobId* is the PRISMA job identifier, which will be displayed in the jobid column of the message window. If the *MessageId* identifies a message, which requires an answer from the operator, the 'w' switch extension allows the program to return only after the question is answered. Send message and **return the created message token**, which can be used for polling for an answer, if the message is question type entry.
-# `repeatCount`	repeat sending the message repeatCount times.
-z `sleepTime`	sleep sleepTime seconds before repeat to send message again.
MessageId	*MessageId* is the unique identifier, which is used as the reference for the static language dependent text, which is enriched with the supplied inserts.
Insert1 *Insert2* *...*	Enrichment inserts for the static text, which is represented by the *MessageId*.

Note: At least on of the first two options (-m, -j) has to be specified.

Examples:

```
ppmsg -m SendTest 08000001 Insert1 Insert2 ...
ppmsg -j  12345678  99999999 Insert1 Insert2
ppmsg -mw AnswerTest 08000011 Insert1 Insert2 ...
```

4.4.5 Reply An Open Question

Usage: `ppmsg -a[i] MessageToken Answer`

Command line option	Description
-a *MessageToken AnswerText*	Set the answer *AnswerText* for an open question, which is identified by the *MessageToken*. *MessageToken* is an unique identifier for each message entry in the database.
-ai *MessageToken AnswerIndex*	Same as before, but if the question offer a list of predefine allowed answers, the answer can be selected by the index of that answer.

Examples:

```
ppmsg -a 12345678  Continue
```
-> reply with answer string "Continue"

```
ppmsg -ai 12345678  2
```
-> reply with 2nd answer from option list

4.4.6 Read Messages

Usage: ppmsg [-option] [*MessageToken*] [*MessageCount*]

Command line option	Description
-n	Print no header line
-g	Read all entries (no message type filtering)
-e	Get error entries only
-i	Get info entries only
-l	Get log entries only
-w	Get warning entries only
-q	Get open question entries only
-qa	Get answered question entries only
-m ModuleName	return messages from ModuleName only. This switch is valid only after '-g', '-e', '-i', '-l', '-w', '-q' or '-qa' switch.
-j JobId	return messages for job with JobId only. This switch is valid only after '-g', '-e', '-i', '-l', '-w', '-q' or '-qa' switch.
MessageToken (optional, default = 0)	*MessageToken* is the reference position for an entry in the database, it's an unique identifier for each message entry in that database. *MessageToken* = 0: last message entry *MessageToken* = 1: first message entry
MessageCount *(optional, default = -5)*	Count of message to read: = 0 : read all messages < 0 : read MessageCount messages before the reference position > 0 : read MessageCount messages after the reference position

Examples:

```
ppmsg  0  -100
```
-> read last 100 entries

```
ppmsg  -n 1  100
```
-> read first 100 entries and print no header line

```
ppmsg  0  0
```
-> read all entries from last to first

```
ppmsg  1  0          -> read all entries from first to last
ppmsg  -e 0  0       -> read all error entries from first to last
ppmsg                -> read last 5 entries
ppmsg  -g -j 12345678-> read last 5 entries of job 12345678
ppmsg  -g -m test    -> read last 5 entries from module 'test'
```

Each entry is printed in a single line with multiple columns:

Column	Byte positions	Description
Message-Token	001 – 008	Message token as a hex value, e.g. 00000001, 0000000A, 00BC1243, ...
Date	010 – 019	Date in the format: YYYY/MM/DD
Time	021 – 028	Time in the format: HH:MM:SS
User+Hostname	030 – 061	Name of the server, generated the message.
Job-Id	063 – 079	Id of the job, the message belong to.
Message-Id	072 – 079	Id of the text, representing the static message text.
Module-Name	081 – 112	Module name, generated the message.
Message-Type	114 – 114	Message types: L -> Log entry I -> Info entry E -> Error entry W -> Warning entry Q -> Open question A -> answered question U -> User message entry
Description	116 – new line	Short text description

4.4.7 Read Message Details

Usage: `ppmsg -d[option] MessageToken`

Command line option	Description
-d or –dl	get detailed info-text for a single message
-di	get short info for a single message
-do	get operator info for a single message
-dq	get question options for a single message
-da	get operator's answer for a single message return immediately, if answer is not available yet.
-dw	get operator's answer for a single message, wait for operator's answer if not available yet.

MessageToken	*MessageToken* is an unique identifier for each message entry in that database. *MessageToken* = 0: last message entry *MessageToken* = 1: first message entry

Examples:

`ppmsg -d 12345678` -> read detail text for message entry 12345678

`ppmsg -dw 12345678` -> read operator's answer for question 12345678 and wait for answer if required.

4.5 Job Status Information

Monitors the status of print jobs in the teletype environment. There are two commands, which have the same syntactical possibilities: **jobstat** and **splstat**. Both commands differ in the selection of the jobs, to which the command is related, and in the sorting of the result:

jobstat -d (display), jobstat -s (set), jobstat -l (list)

Jobstat relates to all jobs in the job list. The output of the jobs found is in the order of the job numbers.

splstat -d (display, splstst -s (set)

splstat only relates to the jobs, which have at least the 'ready to print' status. Therefore this command only searches for jobs, which are in the print queue. The output is in the print order.

Entering the command without a function or operand displays general information on the command.

Overview of functions and operands:

Note: The description of the **jobstat** command below also applies to **splstat** command for the 'ready to print' jobs in the job list.

Command: **`jobstat -d [Function] [Operand]`** is used for displaying job settings.

Note: With job id/file id you have to enter for instance: 0134/1, i.e. jobid = 0134 and file id = 1. You can get the file id from the PJM 'List of Files' or from the 'Job Details Viewer' in the Job List. 'Release job' means 'continue execution'.

Command	Function	Operands	Explanation

jobstat -d	all	0-999	Displays the following information for each print job in the spool:
			`Job Id of print job`
			`Name of the job`
			`Number of copies`
			`Job class`
			`Form`
			`Status of print job`
			If an operand is specified, only the jobs of that class are displayed.

jobstat -d	first	0-999	Displays the first job id for the given queue.

Command	Function	Operands	Explanation

jobstat -d	job	<jobId>	Displays the following information for the print job with the specified spool Id:
			`Spool-Id: 00000001`
			`Owner: root`
			`Submitted: 12/05/2001 13:08:32`
			
			`Jobname: pages.txt`
			`File type: AFP`
			`Filesize: 794`
			`Pages: 300`
			
			`Job class: 1`
			`Form: STD`
			`Printer:`
			`Resolution: 300`
			`Copies: 1`
			
			`Act. printed file: 0`
			`Act. file copy: 0`
			`Act. job copy: 0`
			`Act. page: 0`
			
			`Status: Ready to print`

jobstat -d	status	<jobId>	Displays the status of the print job with the specified Id.
jobstat -d	copies	<jobId>	Displays the number of copies of the print job with the specified Id.
jobstat -d	jobcl	<jobId>	Displays the job queue of the print job with the specified Id.
jobstat -d	printer	<jobId>	Displays the printer selected for the given job.
jobstat -d	priority	<jobId>	Displays the priority of the print job with the specified id.
jobstat -d	reference	<referenceId>	Displays all jobs for the given reference id.
jobstat -d	filepages	<jobId>/<fileId>	Displays the number of pages for the given file.
jobstat -d	filesize	<jobId>/<fileId>	Displays the size of the given file.
jobstat -d	filecopies	<jobId>/<fileId>	Displays the copies of the given file.

Command: **`jobstat -s {Function} [Operand] [Operand]`** is used for changing settings.

Overview of functions and operands:

Command	Function	Operands	Explanation
jobstat -s	copies	<copies> <jobId>	Sets the copy count for the print job to the specified value.
jobstat -s	jobqueue	<jobcl> <jobId>	Sets the specified job queue for the print job.
jobstat -s	printer	<printer> <jobId>	Change the destination printer for the print job.
jobstat -s	range	<page range> <jobId>	Set the printing range for the print job to the specified page range.
jobstat -s	form	<form> <jobId>	Set a form, which is defined on the system, for the print job.
jobstat -s	deljob	<jobId>	Deletes the print job with the specified job Id.
jobstat -s	release	<jobId>	Sets the status of a job to 'Ready to print'.
jobstat -s	express	<jobId>	Set 'Express' right for the job.
jobstat -s	priority	<jobId>	Set a priority for the specified job.
jobstat -s	filepages	<pages> <jobId/ fileId>	Sets the number of pages for a file.
jobstat -s	filesize	<size> <jobId/ fileId>	Sets the size for a file.
jobstat -s	filecopies	<copies> <jobId/ fileId>	Sets the number of copies for a file.
jobstat -s	posabs	<jobId> <page> <filecopy> <printfile> <jobcopy>	Sets the job to an absolute position.
jobstat -s	posrel	<jobId> <boundary> <reference> <direction>	Sets the job to a relative position. **Note**: boundary: no | filecopy | jobcopy reference: curpage | checkpoint direction: backward | forward

Command: **`jobstat -l [Function] [Operand]`** list format of the jobstat command (not available for splstat).

General function:
Output values for a specific or for all active jobs in the system. The values are returned line by line per job. The single fields are separated by tabstops. The amount of the output can be controlled by indicating the desired parameters or via a parameter file.

Command	Function	Operands	Explanation
jobstat -l			Issue help information
jobstat -l	-crpar	[-p <paramfile>]	Creates a template parameter file in the local directory with the given name.
jobstat -l	-j	all	Displays parameters of all active jobs.
jobstat -l	-j	id	Dislays parameters of the jobs of the given job-id.
jobstat -l	-p <paramfile>		Name of the parameter file which should be used for this command.
jobstat -l	-f	<f1>:<f2>...	Displays single parameters (fields). The fields must be separated by a colon.
jobstat -l	-o	<output file>	Name of the output file. If the name is missing, the output is sent to 'stdout'.
jobstat -l	-h		Also displays header line.
jobstat -l	-sphar		Issues all allowed parameters (fields) for -f.

Contents of the parameter file:
The statement `HeaderLine=yes | no` defines if a descriptive header should be displayed or not. Default is 'no'.

The order of the list of usable parameters defines the order of the output. Lines which start with # are comments, which are not used.
The command jobstat -l -crpar creates a template file, which contains all supported parameters. This template file can be modified in the following ways:

- activate desired parameters by deleting the leading #-characters
- change the order of the parameters

The following parameters can be used:

	Meaning
jobid	
jobname	

jobstatus	ODS-Job status. The following values are possible: – wait – wait-event – transfer – work – ready-to-print – printing – finished – locked – error – interrupted (the job has been interrupted) – interrupt-hold (job waits for release) – interrupt-mirrored – output control
owner	
queue	
printer	
form	
job_copies	
reference_id	
print_start	
print_end	
accept	Time, when the print job has been accepted on the server
range	Print range
pages	Total cont of pages
pages_print	Printed pages
filename	Name of the first file
formdef	Formdef of the first file
pagedef	Pagedef of the first file
userinfo1	
userinfo2	
userinfo3	
userinfo4	
userinfo5	
userinfo6	

Example:
Jobid, status, printer name and job name with header line should be displayed.

Contents of the parameter file:

```
HeaderLine=yes
id
jobstatus
printer.name
name
```

Command:

```
jobstat -l -j all -p <parameterdatei>
```

Output:

```
id     jobstatus   printer.name   name
4711   work  printer1    testjob 1
4712   ready-to-print    printer1    testjob 2
4710   finished    printer1    testjob
```

4.6 PJM

Besides the graphical user interface, the Print Job Manager also provides a command-based interface, which can be called on all systems that have the PJM GUI Client installed. The PJM command line can also be installed separately.

The executable version of the command line is supported on LINUX (`/usr/bin/pjm`) and Windows (`spjm.exe` in folder CD\tools_win). The executable is small and fast on the client, as the PRISMAproduction socket interface is used and most of the work is done on the server.

The Windows user has to install the Windows command line version on the windows system by hand.

The description for the command line is valid for all incarnations.

The PJM command line enables you to execute print jobs by entering a command string. Also a few maintenance functions can be performed especially regarding Print File Libraries. In general some form of a Job Ticket is defined and merged with other command line options. This results in a submission ticket which is transferred to the PRISMAproduction system along with the print data (as far as necessary). The resulting ticket controls the print process on the server site.

In particular you can:

- Specify a PRISMAproduction system on which the print job should run
- Define either a local or reprint Job Ticket as basis for a print job - optional along with a custom ticket OCT for modifications of the original ticket
- Specify a job class, printer, the job name, copy count and form
- Specify a page definition and form definition for APA jobs
- Select a composer set to impose chained files
- Replace local files and/or files from a Print File Library in the list of files already defined in the original ticket
- Delete documents in a Print File Libraries and Resource Libraries
- Display the status of a job or a list of jobs
- Cancel a job or a list of jobs
- Request a jobid and use it in a subsequent job submission

All options regarding Print File Libraries are applicable to PRISMAproduction systems with the language module POD only. Same applies to the composer set name.

The option regarding page and form definition is applicable to PRISMAproduction systems with the language module APA only.

4.6.1 Calling the PJM Command Line:

`pjm` **`parameter...`**

All parameters are optional and can be entered in any order.

Information Parameters	**Meaning**
-? or no parameter	display all parameters available. There is no processing action in this case.
-v	display version number. There is no processing action in this case.
Connection Parameters	**Meaning**
-s <server-name>	server to which the job or command is to be sent Default: the local system
-user <user-name>	user name under which the PJM command line connects to the PRISMAproduction server Default: the current user
Ticket Parameters	**Meaning**
-t <ticket-name>	Path of the local Job Ticket Job ticket containing default parameters. Default: no job ticket
-oct <oct-name>	Path of the Oct Océ Custom Ticket is used to modify the local Job Ticket or the Reprint Job Ticket (or the empty Job Ticket if none has been defined). Default: no oct
-r <reprint-name>	Logical path of reprint ticket/data. The reprint job must be specified as follows ($ can be replaced by @, as the use of the $ character is restricted in some operating systems): `[/$PFL/]Archive/User/Document` for private or `[/$PFL/]Archive/User#Document` for public jobs. Default: no reprint
-tr <ticketrule>	Name of the ticket rule, which is applied to the executed job ticket. This function is not implemented in the PJM user interface. Default: no ticket rule
Print Parameters	**Meaning**
-jn <job-name>	Name of the job. Default: ‘tmp’ if no job name has been defined in the local ticket or custom ticket. The job name in the custom ticket overlays the job name in the local/reprint ticket.

-jc <job-queue>	Number of the jobqueue. Default: 1
-nc <copy-count>	Number of jobcopies Default: 1
-prt <printer-name>	Name of the target printer or clustername. The printer also determines the resolution at which the job will be printed. Instead of a configured printername or clustername you can also insert the keyword <Any printer> or <Any nnn dpi printer> (nnn stands for the resolution). Default: the first printer available.
-form <form>	PRISMA form name which will be displayed to operator before printing starts. Default: STD, i.e. no form message is displayed
-pdef <page-definition>	PRISMA Page Definition Page definition. If more than one file is defined in the list of files, the page definition is set for each of the according print steps.
-fdef <form-definition>	PRISMA Form Definition Form definition. If more than one file is defined in the list of files, the form definition is set for each of the according print steps.
File Parameters	**Meaning**
-f <file-name>…	path of print files(s) Local print files to be printed using the parameters in the job ticket loaded with **-t** or **-r** or/and **-oct**. The file names in the job ticket are replaced one by one by the names specified. **Note**: the replacement in the list of files takes place after the custom ticket has been merged with the local/reprint ticket. If more files are given here than defined in the merged ticket, additional default file entries are created (defaults according to the ticket type APA/LCDS/POD). Default: the filenames defined in the ticket
-pfl <file-name>…	path of pfl(s) PFL print files to be printed using the parameters in the job ticket loaded with **-t** or **-r** or/and **-oct**. The print file names in the job ticket are replaced one by one by the names specified. Files must be specified as follows: `Archive/User/Document` for private or `Archive/User#Document` for public jobs. **Note**: same as for the -f parameter.
-imp <impos-file>	path of imposition file Impositioning set. All files are imposed after they are chained.

PJM control Parameters	Meaning
-type <language module>	specifies the type of the ticket: APA|LCDS|POD Default: POD The **-type** option is useful in combination with the **-f** option only because otherwise the ticket type is part of the job ticket.
-getid	request a job id Returns a new PRISMAproduction jobid to be used in a subsequent job submission (see **-id** option).
-id <job-ident>	Jobid that is to be used in the current submission. It is assumed that the jobid has been requested by a previous PJM call using the -getid option. Default: generated implicitly for current submission
-quiet	run in quiet mode Do not display formatted messages on STDOUT and STDERR. Display a code instead and optional a list of parameters depending on the code. This parameter is supposed to be set when the PJM Command Line is used in shell scripts. Error handling should be much easier when dealing with codes and parameters instead of formatted (language dependent) messages. See more detailed description below. Default: verbose
-wait	wait until the job is ready to print PJM command line does not return any values until the job is 'ready to print' or is aborted with an error or is deleted. Returncode: If the waiting state of the command line is finished because the job is aborted with an error, the return code is 12 otherwise it is 0. Default: do not wait, return after job has been accepted
-force	accept job anyway If you 'force' a job it is submitted anyway. The job appears in the list of jobs despite of missing important options. From there you have to set the missing options for each job via the PJM GUI. Normally also the error situation, which prevented the job from being accepted in the normal way has to be corrected. **Example**: An output tray has been referenced, which is not defined for the given printer. If the job must be printed on the given printer, the referenced tray must be defined for this printer before resubmitting the job.

-link	Creates links on the ODS master to the print files if possible. You can find the respective parameter under PJM -> Menu 'Job' -> 'Transfer Mode...'. Default: Copy print data to the server.
Job control Parameters	**Meaning**
-status <job-id>…	get basic job status Returns the status of a job / a list of jobs. The jobid is the 8-digit number displayed when a job is submitted. The status can be: wait, work, interrupted, error, unknown, canceled, finished. If the status is error, an additional description of the error will be shown, e.g. "error: tiffconv Input file wrong format".
-prio <num>	Priority of the print job. num = 1-255 (no blank character between -prio and <num>!) Example: "pjm -f /u/prismapro/demo/ioca/demo.ioca -prio100". The user must have the permission to modify the priority.
-cancel <job-id>…	cancel job(s) listed Cancels a job / a list of jobs. The jobid is the 8-digit number displayed when a job is submitted.
-rdelete <resource-id>…	name(s) of Reprint/ Resource Delete documents in Print File Libraries and Resource files. You have to add the fully qualified logical path. $ can be replaced by @, as the use of the $ character is restricted in some operating systems. If you are logged in as 'admin', 'root' or 'system' you can replace the userid in the filename by the wildcard *. **Example:** `/$PFL/Archive/User/Document` for private reprints `/@OVERLAYS/User#Document` for public overlays
Mirror Parameters	**Meaning**
-mirror <server-name>	Server, on which the job should be mirrored during submission. The mirror server must be defined under 'Configuration' -> 'System' on the original server. Respective option on the GUI: 'PJM' -> Menu 'Job' -> 'Transfer Mode' -> 'Mirror Jobs during Submission'.and 'Mirror Server'. Default: no mirroring of jobs
-mp <policy>	Mirror policy: optional/mandatory Respective option on the GUI: 'PJM' -> Menu 'Job' -> 'Transfer Mode' -> 'Mirror Jobs during Submission'.and 'Mirror Policy'. Default: The value set in 'Configurtion' -> 'System'.

-mnd	Mirror no data The data must exist on the mirror server to print the job. Respective option on the GUI: This parameter connot be set via the GUI. Default: Transfer of ticket and data.
-mtr <ticket rule>	Ticket Rule which is executed on the mirror server: <ticket rule> | none. 'none' means that no ticket rule is executed on the mirror server. If a ticket rule is set via the tr-parameter, the ticket rule set here is executed on the mirror server additionally to the ticket rule executed on the original server. Respective option on the GUI: 'PJM' -> Menu 'Job' -> 'Transfer Mode' -> 'Mirror Jobs during Submission'.and 'Ticket Rule'. Default: The value set in 'Configurtion' -> 'System'.
-muser <user-name>	User who starts the job on the mirror server. Respective option on the GUI: 'Configuration' -> 'System' -> 'Mirroring' -> 'User Name'. Default: The value set in 'Configurtion' -> 'System'.
-mpw <password>	Password for the user set under 'muser'. Respective option on the GUI: 'Configuration' -> 'System' -> 'Mirroring' -> 'Password'. Default: The value set in 'Configuration' -> 'System'.

Usage of Ticket, Print and File parameter groups:

These parameter groups may be used exclusively or combined (where they supplement each other). The idea behind is to allow to print either without a Job Ticket or with a Job Ticket/Oct and replacing parameters in the ticket.

– *No local Ticket*

for simple print jobs. Normally only the "file parameter" would be used here, maybe with setting a job name and printer by "print parameters".

Example: -jn simple -nc 2 -f test.afp

– *Local Ticket only*

for fully prepared print jobs. All parameters are defined in the Job Ticket.

– *Local Ticket + Print/Files:*

for prepared print jobs. A few parameters in the job ticket must be updated.

Example: -t test.tic -f test.afp -prt myprinter

– *Local Ticket + Oct*

for print jobs which are based on a default ticket and where a lot of parameters must be updated. All dynamic parameters are contained in the Oct.

– *Local Ticket + Oct + Print/Files*

should not be used. Either the dynamic parameters are written in an Oct or the Print/Files group is used for overwriting.

– *Reprint request*

the special request -r, where a job kept in intermediate storage should be reprinted. The Job Ticket for the reprint is supposed to be the ticket of the original job.

If the -f parameter is used together with a Job Ticket, it simply overwrites the file name in the List of Files in the Job Ticket. The file type in the ticket remains valid. If it is used without a Job Ticket, the type of the file depends on the language module which interprets the job on the server. Default file types are 'IOCA' for the language module POD, 'AFP' for the language module APA and 'LCDS Data' for the language module LCDS.

4.6.2 Syntax Examples for Use Cases

Requesting a job id

[`-s` Servername] [`-user` User] `-getid` [`-quiet`]

Displaying the job status

[`-s` Servername] [`-user` User] `-status JobId1` [JobId2 … JobIdN] [`-quiet`]

Deleting resources and reprints

[`-s` Servername] [`-user` User] `-rdelete` ResOrPfl1 [ResOrPfl2 …ResOrPflN]
[`-quiet`]

Cancel and delete job(s)

[`-s` server name] [`-user` User] `-cancel` JobId1 [JobId2 … JobIdN] [`-quiet`]

4.6.3 Return Codes

Return-code	Meaning
0	**Success**
	All information is written to STDOUT. Job Submission Message for each file transferred to the ODS Master and a final success message. The job identification is automatically included as each message starts with ”Job <job identification>: …”. *Quiet mode*: The ODS job identification. Job Id request Message containing the ODS job identification returned. *Quiet mode:* The ODS job identification Status Status of job(s) is written. for detailed information of <job status> see table below. *Quiet mode:* <job identification> -> <job status>*) Resource delete No information is written. *Quiet mode*: No information Cancel Information that job(s) has(have) been canceled is written. *Quiet mode*: No information
1	**Invalid command line option**
	Is set when an invalid option is found in the command line call. Parameter parsing stops on the erroneous option. STDERR: Message indicating the erroneous option *Quiet mode*: 0 → <invalid option>

Return-code	Meaning
3	**No connection error**
	Is set when no connection to the PRISMAproduction server specified by –s option could be established. STDERR: Message describing the connection problem *Quiet mode*: 1:→ <server name> → <server error message>
5	**Ticket handling error**
	Is set when the tickets given in the **-t** or **-r** and/or **-oct** cannot be found/ opened. STDERR: Message describing the open problem *Quiet mode*: 1 → <local ticket open error message> 2 → <reprint ticket open error message> 3 → <custom ticket open error message>
10	**File transfer error**
	Is set when one of the files in the list of files cannot be transferred. Transfer process stops with the file in error, files already transferred to the server are deleted on the server. STDERR: Message describing the file transfer problem *Quiet mode*: 1 → <file name> → <error message>
11	**Negative Ticket check**
	Is set when the combination of the local or reprint and custom ticket with the command line options results in an invalid ticket. All errors found are listed. Note that there are errors which are so basically that no further analysis can be done (for example an invalid printer, as most of the checks are printer related). Analysis stops with such an error. STDERR: List of messages describing the problems, one message per problem *Quiet mode*: <code1> <error text 1> : : <codeN> <error text N>

Return-code	Meaning
12	**Job submission error**
	Is set when the PRISMAproduction server doesn't accept the job. This can happen for two reasons: - A problem in the negotiation process between the PJM client and server which files are to be transferred (not the transfer itself, would raise a 'File Transfer Error') - The job cannot be passed to ODS for execution Returncode 12 is also set if the waiting status of the of the PJM command line is terminated by an error. STDERR: Message describing the job submission problem *Quiet mode*: 1 → <server error message> 2 → <termination error message> (only possible if the -wait option has been specified)
13	**Status display error**
	Is set when the PRISMAproduction server can't return the status of the given job(s). STDERR: Message describing why the status has not been returned *Quiet mode*: 1 → <server error message>
14	**Job cancel error**
	Is set when the PRISMAproduction server can't cancel the given job(s). STDERR: Message describing why the job(s) has(have) not been canceled Quiet mode: 1 → <server error message>
15	**Resource delete error**
	Is set when the PRISMAproduction server can't delete the given resource(s). STDERR: Message describing why the resource(s) has(have) not been deleted *Quiet mode*: 1 → <resource name> 2 → <resource name> <server error message>

<Job Status> is composed of more detailed information in the following format:

`<Main Status> <Detailed Status> [<Additional Status>]`

`<Main Status>` and `<Detailed Status>` are defined explicitly, whereas <Additional Status> is optional and not definitely defined. `<Detailed Status>` must not be used for a logical decision.

Main Status	Detailed Status (Meaning)
processing	the job is busy, no operator intervention required, information only
	transfer files are being transferred
	work a job step is active (converting,…)
	printing the print data is being printed
	prepintr the print has been interrupted but the printer is still printing (for example copies)
wait	the job waits for a resource or an event, no operator intervention required. Job continues when event occurs or resource becomes available
	outputctrl job has been printed but there are still pages in output tray
	print wait until a printer becomes ready to print job
	event wait for an external event to occur
	service wait for a service to become available
	time wait for starting time to occur
exception	the job is in an error or interrupted or hold state, operator intervention is required
	error-service error occurred when processing job step
	intr-service job step has been interrupted by operator
	hold-service job step is in hold
	error-print error occurred while printing
	intr-print print process has been interrupted by operator
	hold-print print process is in hold wait for starting time to occur
finished	the job is finished but still visible in spool, operator intervention is required for reprint
	final finished, job cannot be reprinted
	locked job can still be reprinted

4.7 "prismadiag" Script

The prismadiag feature has been divided into 2 archiving parts:

- Trace archive (volatile data) "pdiag_trc#.tgz"
- Info archive (longer existing data) "pdiag_info.tgz"

Traces containing volatile data should be archived as soon as possible after or during a problem occurs and prior to info! The last 6 trace archives are saved (1 to 6). The info archive contains prisma + system configuration, messages, logs, ...

The resulting ".tgz" files are stored at "/u/prismapro/diag" now. Without any option prismadiag creates the trace + info archive. Please enter "prismadiag -h" to see the available options.

Additionally the capturing speed has been increased as well as additional files are stored including a short "top trace" run of 3 seconds to get an impression of the current cpu and memory usage.

`/u/prismapro/bin/prismadiag` has a the option –V to create a file `/u/prismapro/diag/version/versions.html` which shows the PRISMA versions of all servers of the PRISMA cluster. This file is also automatically integrated into the normal prismadiag compressed info archive file.

4.8 Accounting

See description of the 'Accounting Command Line Interface' on page 197.

4.9 Printer Connection Test (CATEST)

The program CATEST is used to test the connection between the server (LINUX) and the printer. The program was originally developed for -/370 connections, and throughout this description -/370 terminology is used, though SCSI connections are now also supported.

The basic LINUX device operations (OPEN, CLOSE, READ, WRITE and IOCTL) can be executed.

CATEST can access a single printer or a twin system, therefore it supports configuration of 2 physical printers. It can not be used to handle two independent single printers, the second printer PRT2 can only be configured together with PRT1 as a twin system!

Getting Started

CATEST is a debugging tool. Before running it, make sure that no other program is using the printer(s).

Default locations:

Executable program: `/u/prismapro/bin/catest`

Initialization file: `/u/prismapro/cfg/catest/catest.ini`

Initialization File

The initialization file (INIFILE) is a plain ASCII text file which specifies the hardware and program configuration for the test using keyword parameters. Keyword definitions in this file have to start in the first column, otherwise they are ignored. Interpretation of a line stops at a # symbol. Any characters following a # symbol are treated as comments. Each line may contain only one keyword parameter. If no INIFILE is specified at the command line, CATEST looks for a file called 'catest.ini', first in the current directory and, if the file cannot be found there, in the directory `/u/prismapro/cfg/catest`.

The INIFILE has to be customized to match the existing parameters (for example: the address, use of one or two printers).

For SCSI and /370 special ini files `catest.scsi` and `catest.i370` are also provided.

Change to Printer Config Menu [3]

```
1.3 Printer Configuration

config. PRT1..... [1]    config. PRT2..... [2]    scan adress...... [3]
show config...... [s]    toggle PRT 1<->2. [c]    return........... [r]
quit............. [q]
```

To see the devices currently attached, we use function [3] = scan address .

```
SCAN_ADR: entered

SCAN_ADR:No printer found on this channel. Check channel attachment!
```

There is no printer connected to the channel or it is powered off. After attaching a printer or powering it up, we select [3] again and get the message:

```
SCAN_ADR: entered

SCAN_ADR: printer type & model = ff383501 at dev.adr:0X 50

SCAN_ADR: printer type & model = ff383501 at dev.adr:0X 51
```

This means that two printers are connected. To choose one, we start the config printer 1 function [1]:

```
give physical device adr. (hex value) of the printer

or accept 50
```

We agree to select the printer with device address 0x50, and type return:

```
CATEST: def_prt1: device adr. 0x50 for PRT1 accepted
```

To select the other printer as PRT 2, we use function [2]:

A possible response might be as follows:

```
PRT2 not defined! Configuration not possible.
```

There is no PRT2 defined in the INIFILE and you cannot make any changes to this printer.

Return [r] and Change to Basic Functions [1]

To make proper use of the functions in this menu, users should be familiar with the notations of the IBM /370 channel interface, physical device I/O and the IPDS printer language.

Here we give only a very brief introduction to the most common channel command, the command `sense`:

```
1.1 Basic Functions acting on : PRT1

sense............ [1]    sense ext........ [2]    sense id......... [3]
test io.......... [4]    wr.IPDS(noop+arq) [5]    read ACK......... [6]
discard buf. data [7]    send ccw nop..... [8]    status........... [9]
wait dev-end..... [w]    toggle PRT 1<->2. [c]    show config...... [s]
return........... [r]    quit............. [q]
```

We use function [1] = `sense` to find out if the printer is ready:

```
CAT_HDL: ioctl(SENSE) executed!
CAT_HDL: ioctl(CH_STATUS) executed!
usb : 0c
csb : 00
cc  : 00
       0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
Sense: 01 00 0d 01 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CAT_HDL: Printer Presentation Check: NORMAL PRINTER RESTART
```

The sense bytes displayed in the example are the sense information presented by the printer after power on, reboot or a system reset on the channel. We execute the sense command again [1]:

```
CAT_HDL: ioctl(SENSE) executed!
CAT_HDL: ioctl(CH_STATUS) executed!
usb : 0c
csb : 00
cc  : 00
       0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
Sense: 40 00 03 01 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CAT_HDL: Channel/Unit sensed Exception: PRINTER NOT READY
```

The sense bytes "40 00 03....". indicate that the printer is not ready. The printer has to be set to ready on the printer operator panel. The next `sense` command will result in:

```
CAT_HDL: ioctl(SENSE) executed!
CAT_HDL: ioctl(CH_STATUS) executed!
usb : 0c
csb : 00
cc  : 00
       0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
Sense: 00 00 00 81 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CAT_HDL: Channel/Unit sensed Exception: PRINER READY
```

This indicates that the printer is ready.

The following table gives a brief description of the status bytes:

usb = user status byte:	0x01	Unit Exception (UEX)
	0x02	Unit Check (UCK) Error condition detected on the device. Error bytes can be read with the sense function.
	0x04	Device End (DE) Device is ready for I/O operation or I/O operation was executed.
	0x08	Channel End (CE) Subchannel is usable or data transfer between device and channel is complete
csb = channel status byte:	0x01	Chaining check
	0x02	Interface control check
	0x04	Channel control check
	0x08	Channel data check
	0x10	Protection check
	0x20	Program check
	0x40	Incorrect length
	0x80	Program-controlled interrupt
cc = condition code:	0x03	Channel or device is not available

Return [r] and Change to Print Functions [2]

This menu has functions for sending files containing IPDS commands to the printer.

```
1.2 Print Functions printing on : PRT1
print..... [1]    open input..... [2]    file_copy..... [3]
show config..... [s]    return..... [r]    quit..... [q]
input index:
```

Let's assume we would like to print the sample file `u/spslib/applic1/30pages.300dpi.ipds`.

[2] opens a file to be read as input for printing.

```
CATEST: Please enter Name of file to print:
or the string 'quit':
INPUT==> /u/prismapro/demo/afp/pages/30pages.300dpi.ipds
```

[1] will start the output of the file contents to the printer.

```
INPUT==> print.....
RD_WR: rd_wr: 1910 bytes read from input
WAIT_DE: entered!
WAIT_DE: PRT1 device end detected!
WAIT_DE: loop_count = 1
RD_WR: rd_wr: 1910 bytes written
RD_WR: rd_wr: END OF FILE on input reached!
RD_WR: rd_wr: totally 1910 bytes read, 1910 bytes written!
```

At the end of our example we quit the CATEST with [q].

4.10 Tape Utility Program Tapman

Program start

The program tapman is located in `/u/prismapro/bin`. It is a service tool for troubleshooting and diagnostics of problems regarding the SCSI-connected 9-track tapedrives and 18-/36-track cartridge drives. It can be started from the command-line:

```
cd /u/prismapro/bin
```

```
./tapman -c /u/prismapro/cfg/tapman/tape.cfg
```
(for 9-track tapedrive)

or

```
./tapman -c /u/prismapro/cfg/tapman/cartrdge.cfg
```
(for 18-/36-track cartridge drive)

The program can also be started from the KDE UI (‘PRISMAproduction’>’Utilities’>’Tapman’ ’Tapman Tape’ or ‘Tapman Catridge’ or ’Tapman QIC’).

After startup from the UI you have to load the configuration file "tape.cfg" or "cartridge.cfg" (depending on which drive you want to use) using option [1] in the Main menu and [1] in the 1.1 Hardware and Software Configuration menu:

```
1.1 Hardware and Software Configuration

read config file... [1]    write config file.. [2]    show configuration. [3]
edit configuration. [4]    default config..... [5]    return.............. [r]
quit..............  [q]
input index :
```

After returning to the main menu, you have full access to all functions of tapman.

Analyzing unknown tapes

The tapman utility offers three functions to help you analyzing unknown tapes. They are in the 1.7 Tape Analysis menu.

```
1.0 Main Menu

HW and SW config... [1]    tapefile settings.. [2]    tape directory..... [3]
read tapefile...... [4]    write tapefile..... [5]    low-lev. tape acc.. [6]
analyse tape....... [7]    remote tape acc.... [8]    quit............... [q]
input index :
7
INPUT==> analyse tape.......

1.7 Tape Analysis

analyse labels..... [1]    tape structure..... [2]    tape dump.......... [3]
dump tape to file.. [4]    dump file to tape.. [5]    return.............. [r]
quit..............  [q]
input index :
```

The **analyze labels** function [1] reads the beginning of the tape/cartridge and compares the data with known formats. The list of known formats is currently rather limited - tapman knows "Standard labeled" tapes, "Nolabel" and "Xerox Resource" tapes.

More information can be obtained by generating a **tape structure** [2] file. This is a readable text file which contains information about all data blocks, labels and tapemarks found on this tape.

If you want to see the actual content of the tape, you can generate a **tape dump** [3]. This writes the content of the tape into a text file in hexadecimal and text representation. Be careful with this function, because the tape dump file may become very large! You can interrupt and stop the tape dump function with the DEL key.

Generating Image Files

For problem analysis it may be important to transport the original data to a service center or the headquarter. Instead of sending the tape it is possible to generate an image file from a tape and transfer this file via network.

This function is also located in the Tape Analysis menu and is called dump tape to file [4]. The user is then prompted for a filename, and tapman generates a physical copy of the tape into two files: the data file with the given filename, and an index file with the given filename plus the suffix ".idx". Both files together must be transferred via network to their destination in binary format. There the service person can generate an exact copy of the original tape with the function **dump file to tape** [5].

4.11 AFP/IPDS Analyze Tool

The utility program "analyze" is a tool for analyzing data. It can be used to display the structure of AFP files (structured fields), IPDS files and simple hexdumps of any file with EBCDIC or ASCII coding. The program is menu-driven, the input sequence is similar for all 3 modi. You first select the required type of function in the main menu, then specify the coding (EBCDIC or ASCII) in the next menu, then open the input file in the next menu. This is shown below:

```
*****************************************************
*  COPYRIGHT (C) OCE PRINTING SYSTEMS GMBH 2001   *
*            ALL RIGHTS RESERVED                  *
*****************************************************

ANALYZE: V 3.04.00   May 14 2004
--------------------------------
1.0 IPDS- / AFPDS- Analyzer
Analyze - IPDS      [1]
Analyze - AFPDS     [2]
Hexadecimal Dump    [3]
Quit                [q]
Enter function code: 1

ANALYZE: V 3.04.00   May 14 2004
--------------------------------
2.0 IPDS - Analyzer
Analyze (EBCDIC)    [1]
Analyze (ASCII)     [2]
Statistics          [3]
Return              [r]
Quit                [q]
Enter function code: 1

ANALYZE: V 3.04.00   May 14 2004
--------------------------------
3.1 IPDS - Analysis - EBCDIC
Open Input File     [1]
Open Output File    [2]
Analyze all CC      [3]
Statistic Output    [4]
Return              [r]
Quit                [q]
Enter function code:
```

Now you can either start analyzing the input file with option 3 and have the results displayed on the screen, or you can open an output file for the results. The output file can later be browsed using any text editor (e.g. vi).

As an example the screen output is shown below.

```
Output plus hex dump ? <y(def);n> y

Analyze  SF -  Records ; File: print1.dat

001f  BDT   (Begin Document)
            Flag = 00 ; Seq.NR. = 0002
            Token Name <>

000000    001fd3a8 a8000002 40404040 40404040      * ./Lyy...          *
000010    00000518 010c0008 21060080 008000        * ................  *

0008  BPG   (Begin Page)
            Flag = 00 ; Seq.NR. = 0003

000000    0008d3a8 af000003                        * ..Ly....          *

0008  BAG   (Begin Active Envir.Gr.)
            Flag = 00 ; Seq.NR. = 0004


-- continue <enter>; break <b>; resume <r> --
```

After each screen you have the options to continue the display, leave the display with "break" or continue without further stops at the end of each page with "resume".

The "Analyse-IPDS" menu offers EBCDIC and ASCII display, and statistics over the used command codes.

```
ANALYZE: V 3.04.00   May 14 2004
--------------------------------
2.0 IPDS - Analyzer
Analyze (EBCDIC)     [1]
Analyze (ASCII)      [2]
Statistics           [3]
Return               [r]
Quit                 [q]
Enter function code: 1
```

The last available mode is "Hexadecimal dump" which can be used to display the content of any file in either ASCII or EBCDIC representation as well as in hexadecimal.

```
ANALYZE: V 3.04.00   May 14 2004
--------------------------------
2.1 Hexadecimal Dump
Open Input File      [1]
Open Output File     [2]
Hexa-Dump(EBCDIC)    [3]
Hexa-Dump (ASCII)    [4]
Return               [r]
Quit                 [q]
Enter function code:
```

4.12 PRISMAproduction Command Line Tools

The tools are presented here in alphabetical order:

/u/prismapro/bin/add_crlf	Add CR+LF after each structured field record.
/u/prismapro/bin/accana	Uses an accounting file (in /u/prismapro/account/) as input and writes some LCDS relevant account records as human readable text to stdout, to be used for test and analyze purpose, only.
/usr/bin/ana_bat	Uses AFP or IPDS as input and writes the commands into readable text file. Command line version.
/u/prismapro/lib/win32/bin/ana_bat.exe	Uses AFP or IPDS as input and writes the commands into readable text file. Command line version for Windows. Also available on the installation DVD under the path tools_win.
/u/prismapro/bin/ana_nto1	Uses nto1 index files (nto1.idx|nto1.idr|nto1.imm in /u/prismapro/tmp/lcds/) as input and writes the index information in human readable text to stdout.
/u/prismapro/bin/analyze	Uses AFP or IPDS as input and writes the commands into readable text file. Interactive version.
/u/prismapro/lib/win32/bin/analyze.exe	Uses AFP or IPDS as input and writes the commands into readable text file. Interactive version for Windows. Also available on the installation DVD under the path tools_win.
/u/prismapro/bin/chfnm	Change file names to upper - or lower case.
/u/prismapro/bin/dpconnect-set	Sets and displays the log-, trace- and keep-data-settings for the DPconnect executables and shell scripts.
/u/prismapro/bin/dpconnect-ver	Displays the version numbers for the DPconnect executables, shell scripts and the GUI.
/u/prismapro/lib/win32/bin/guitbox.exe	Windows configuration tool for BARR card. Also available on the installation DVD under the path tools_win.
/u/prismapro/bin/iana	Uses ifilter generated files (especial JSL and XRX) as input and writes human readable text to stdout.

/u/prismapro/bin/initlcds <catalog_name>	Sets the LCDS default standard code tables, PCC tables and PDE 's (FMT*) (ldparm). If no catalog name is given, the ldparm is updated for all catalogs.
/u/prismapro/bin/irev	A "reverse ifilter", it uses an ifilter generated file as input and produces an ifilter input file. See the manual pages with 'man irev' .
/u/prismapro/bin/irevjsl	This tool is only to be used for migration of existing PRISMAproduction version 3.04 or older customer catalogs. From version 3.10 on all jsl files are plain ASCII files. This tool uses all JSLs in the actual jsl/dat directory as input and writes human readable text files with the same names to '..' (i.e. jsl/.). These can be used for searching a certain PDE, CME, STO e.a. using grep.
/u/prismapro/bin/ jobclass_import [-s master-server] class.cfg-file	Import a "class.cfg" from PRISMAproduction Version 2.x into Version 3.x
/u/prismapro/bin/job2tic	Convert spsprt job file to ticket
/u/prismapro/bin/jobstat	Status viewer for jobs in PRISMA spool. See 'Job Status Information' on page 222.
/u/prismapro/bin/jpjm	PJM command line as Java application.
/u/prismapro/bin/jview	Displays page images of POD jobs in the spool using pdv or pdv3. Jobs are specified by single job IDs or a range of job IDs.
/usr/bin/lcdsver	Gives version numbers of RPMs and certain modules used by LCDS.
/u/prismapro/bin/modpcname	Change hostnames, IP-address or domain name.
/u/prismapro/bin/octget /u/prismapro/bin/octset	Get (read) and set (overwrite) parameters of OCT custom ticket files. For usage use the “-h” option. These tools replace the former commands ticget / ticset.
/u/prismapro/bin/odsctrl	To stop and start ODS in case of an emergency.
/u/prismapro/bin/odsdiag	ODS diagnostics tool.
/u/prismapro/odsprot	Temporarily overwrite the ODS protocol in case of an emergency
/u/prismapro/odsretinf	Interface between services and ODS.
/u/prismapro/odsselprot	Tool to extract protocol information by time criteria.
/usr/bin/pcla3	Standalone script to convert PCL to AFPDS at 300dpi.

/usr/bin/pcla6	Standalone script to convert PCL to AFPDS at 600dpi
/usr/bin/pclmt3	Standalone script to convert PCL to multiTIFF at 300dpi.
/usr/bin/pclmt6	Standalone script to convert PCL to multiTIFF at 600dpi.
/usr/bin/pclt3	Standalone script to convert PS/PDF to TIFF at 300dpi.
/usr/bin/pclt6	Standalone script to convert PS/PDF to TIFF at 600dpi.
/usr/bin/pdc	Professional Document Composer executable.
/usr/bin/pdv3	Professional Document Viewer executable.
/usr/bin/pdv3convert	converts AFPDS(IOCA) files to PS or PDF (requires PS+PDF output license).
/usr/bin/pfa2font	converts PostScript fonts from binary *.pfa format to Type 1 format for use with the UnityRIP.
/usr/bin/pfb2font	converts PostScript fonts from binary *.pfb format to Type 1 format for use with the UnityRIP.
/usr/bin/pjm	PRISMAproduction Job Submission Command Line.
/usr/bin/ppacc	Accounting command line utility, replacement for the old "accnum" and "accmsg". Utilities: ppacc –i get a new accounting ID from the server. This is a replacement for **accnum** command, but **accnum** stays available as an alias for **ppacc –i** ppacc –a write one accounting record. This is a replacement for **accmsg** command, but **accmsg** stays available as an alias for **ppacc –a**
/u/prismapro/lib/win32/bin/ppacc.exe /u/prismapro/lib/win32/bin/ppacc.dll	Accounting command line utility with library for Windows, function like /usr/bin/ppacc. Also available on the installation DVD under the path tools_win.
/usr/bin/ppmsg	Command line utility to display or submit PRISMA messages.
/u/prismapro/lib/win32/bin/ppmsg.exe /u/prismapro/lib/win32/bin/ppmsg.dll	message command line utility with library for Windows, function like /usr/bin/ppmsg. Also available on the installation DVD under the path tools_win.
/usr/bin/prisma_snmp	Script for user root to set either standard SuSE SNMP or PRISMA SNMP active.

/usr/bin/prismadiag	archives traces and info data (prisma + system config., messages, logs...). Traces contain volatile data and should be archived as soon as possible after or during a problem occures and prior to info! The last 6 trace archives are saved (1 to 6).
/usr/bin/setdefpaper <new_papersize>	Sets the LCDS default paper size (A4, A3, letter, ledger, legal, B4) for all relevant JCFs (Copy, PrintLCDS, Sample) or displays current default paper size (called without parameters).
/usr/bin/setpapersize <new_papersize>	Sets the LCDS default paper size (setdefpaper) and standard code tables, PCC tables and PDE 's (FMT*) (initlcds). Parameters like setdfepaper.
/usr/bin/sf_modi	Utility to modify structured fields.
/usr/bin/spj	Displays detailed properties of POD jobs in the spool. Jobs are specified by single job IDs or by a range of job IDs.
/u/prismapro/bin/spjm	Executable PJM command line.
/u/prismapro/lib/win32/bin/spjm.exe	Executable PJM command line for Windows. Also available on the installation DVD under the path tools_win.
/usr/bin/switch_snmp	Interactive toggle switch for user root to switch from standard SuSE SNMP to PRISMA SNMP and vice versa.
usr/bin/tapman	Tape dump/check utility. Read/write tapes/cartridges to/from hard disk.
/u/prismapro/bin/tarscript_HD	In order to save or restore directories or single files (esp. whole LCDS catalogues) you can use the tarscript command in the following way: /u/prismapro/bin/tarscript_HD <dir_name> <HD-file name> [SAVE|RESTORE] **Example**: /u/prismapro/bin/tarscript_HD /u/prismapro/data/cat/DEFAULT/tmp/examle.tar SAVE This HD file can easily be copied (ftp, etc.) to other computers or USB-, DVD-media.
/usr/bin/tiffconv	TIFF converter executable.
/usr/bin/ua3	Standalone script to convert PS/PDF to AFPDS at 300dpi.
/usr/bin/ua6	Standalone script to convert PS/PDF to AFPDS at 600dpi.

/usr/bin/ut3	Standalone script to convert PS/PDF to TIFF at 300dpi.
/usr/bin/ut6	Standalone script to convert PS/PDF to TIFF at 300dpi.
/usr/bin/xpdc_lib_update	Script to import xpdcresources from PRISMA+POD 2.10.

4.12.1 PRISMAproduction Command Line Tools for Linux on the Installation DVD

/tools_linux/prisma_unlink	This script may be used prior to backup the PRISMA configuration if V3.00.05 or below is installed. It prevents deinstallation of directories in the scanned path of the HotDirectory. It also prevents deletion of customer AFP resources as well as the customer data if V3.00.01 or below is installed. See chapter 'Installation' for details.
/tools_linux/prismaver	It shows version numbers of the installed PRISMAproduction packages including sub-modules as shown in the package list of the release notes.
/tools_linux/testpcname	The tool 'tstpcname' tests the file '/etc/hosts' for correct entries of the hostname for PRISMAproduction.

5 JDF Support

It is assumed that the reader is familiar with the basics of the JDF industry standard. The main entry point to JDF is the CIP4 home page *http://www.cip4.org*.

As a reminder a few basic terms are repeated below.

5.1 Terms

JDF: "Job Definition Format". A comprehensive XML-based file format/proposed industry standard for end-to-end job ticket specifications combined with a message description standard and message interchange protocol.

JMF: "Job Messaging Format". The message description standard and message interchange protocol of JDF.

ICS: "Interoperability Conformance Specification". ICS documents each cover a class of devices, be it equipment or software, such as "desktop digital printers" vs. "professional integrated digital printing." An ICS establishes the minimum level of JDF compliance for a class of devices. The ICS documents are really a subset of JDF, and are not allowed to expand upon JDF.

IDP: "Integrated Digital Printing". The ICS which describes the JDF/JMF subset for digital printing.

5.2 Introduction

In version 3.10 of PRISMAproduction support of JDF/JMF has been introduced. As JDF/JMF is an evolving standard, as the first step the technical basics to be able to process JDF/JMF requests have been implemented.

The company wide JDF Working Group is creating the “Océ Standard JDF Ticket” specification. It is based on the IDP ICS (Integrated Digital Printing). This kind of ICS favors the approach of having most of the workflow steps defined directly in the JDF (in contrast to an “intent based” JDF where the workflow steps evolve during execution).

As such, the specification matches well the PRISMAproduction approach of pre-generated workflows and will be fully supported in future releases of PRISMAproduction.

The Océ extent of the Job Messaging Format (JMF), which is also defined in the “Océ Standard JDF Ticket” specification, is already supported by version 4.00.02 of PRISMAproduction. That means, third party systems can fully communicate with PRISMAproduction using JMF over HTTP.

Regarding JDF Ticket acceptance on the input side, tickets created and submitted by Acrobat7 are supported. This is mainly intended for system engineers / customers to become familiar with the JDF/JMF world in the PRISMAproduction environment.

5.3 The JDF/JMF framework

The framework allows JDF Job Tickets or JDF submission messages (referencing a JDF Ticket) to be sent to a specific URL on the PRISMAproduction master.

PRISMAproduction will process the request and forward it to a JDF Job Submission server. This server will handle the JDF Job Ticket and the data files involved and create a standard PRISMAproduction job.

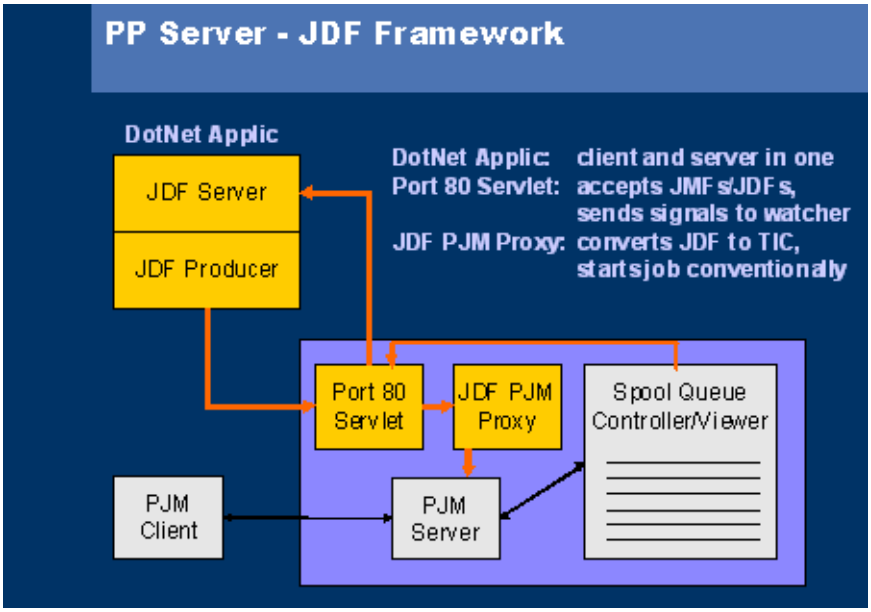
To be able to do that, a PRISMAproduction Job Ticket is created internally, either by just using defaults or by using a template name passed via a JDF parameter. Actual JDF Ticket parameters are merged with in specific fields of the PRISMAproduction ticket.

A JDF/JMF environment allows the submitter of a job to name a location/URL where status messages are to be sent. This could be a one time request to just get back information about the status of the job submission or the request for a "persistent channel". The latter defines that all status changes are to be reported to the defined

location automatically. In addition status information could also be requested by "give me status of job X" messages. The PRISMAproduction JDF/JMF framework supports all described methods of status information passing.

Note: As jobs submitted via the JDF interface end up as standard PRISMAproduction jobs, they can be fully monitored/controlled inside the PRISMA Explorer. But when a status channel has been established, all status changes are also reported to the location defined in JDF/JMF. So the jobs can be seen exactly in the same way by some external JDF based monitoring tool. In addition to job monitoring also the job control operations defined in the “Océ Standard JDF Ticket” specification (for example: abort, hold, remove, ...) are supported.

The following figure shows how existing and new JDF specific components interact with each other:



JDF/JMF Components Overview

Port 80 Servlet: The entry point to the JDF/JMF support in PRISMAproduction. All requests posted to *http://pclife6646/prismapro/jmf/controller.jsp* are forwarded to the PRISMAproduction JDF/JMF Framework

JDF PJM Proxy: The JDF Job Submission Server which interprets the JDF Job Ticket, converts it to a PRISMAproduction Ticket and starts the actual Print Job

DotNet Applic: A Microsoft .Net application developed during the implementation of the JDF/JMF Framework for testing and demo purposes. It can submit prepared jobs and view the status of the submitted jobs, all by using the JMF messaging interface.

Note: The .Net demo application is shown in the picture as third party representative. It's not part of the PRISMAproduction distribution.

5.4 Summary

- The basic technique to support JDF/JFM has been implemented in PRISMAproduction.
- The "Océ Standard JDF Ticket" specification will be supported in future versions of PRISMAproduction and will be the base for all third party systems submitting jobs to PRISMAproduction via JDF/JM.
- The extent of JMF support defined in the "Océ Standard JDF Ticket" specification is already implemented in version 4.00.02 of PRISMAproduction and is the base for all third party systems wanting to monitor/control PRISMAproduction jobs via JMF.
- The Acrobat7 JDF Ticket support is included for demonstration/learning purposes.

Links from PRISMAproduction to other Océ products offering additional functionality (for example: price estimation, …) will be defined and implemented in the future.

6 SNMP Agent

Note: The PRISMAproduction SNMP agent can be switched on and off using the script `u/prismapro/bin/switch_snmp`.

6.1 Accessing Values with SNMP

Accessible Objects after Installation

The following table shows the managed objects accessible via the SNMP Agent.

Object Identifier	Description	ASN.1-Type	ACCESS
svrProductName	Print server product name. The default value is 'unknown product name'.	DisplayString	read-only
svrProductVersion	Print server product version. The default value is 'unknown version'.	DisplayString	read-only
svrServerStatus	The Active (2)/ Inactive (3) status of the print server. When it is active, it can offer service on all user clients. When it is inactive, it cannot offer service on any user clients. The default is Active. The status may also be set with on (4) to active and off (5) to inactive.	INTEGER	read-write
svrServerIpAddr	The IP address of the print server. The default value is '0.0.0.0'.	IpAddress	read-only
svrNumberPrinters Supported	The number of printers supported by this server. The default value is 64.	INTEGER	read-only
svrNumberSystems	The number of systems. For a single system this will always be one. For a cluster of systems this is the number of systems in the cluster. The default value is 0.	INTEGER	read-only

Object Identifier	Description	ASN.1-Type	ACCESS
svrJobManagerStatus	Once the job manager is active (2), the spool looks in its spool directory for printable jobs. When it is active the spool selects any job from the spool directory. When the job manager is inactive (3), the spool directory only contains jobs belonging to the current job class. If the printer has not yet been activated manually, it is activated automatically as soon as the job selection mechanism is started. The default value is unknown (1). The svrJobManagerStatus is normally active, when the svrServerStatus is active, that is the spool is active. If the spool is inactive, the svrJobManagerStatus is normally inactive.	INTEGER	read-only
svrJobAccounting Status	When accounting is active (2), accounting data can be generated for each print job and stored for analysis by users. The default value is unknown (1). The svrJobAccountingStatus is normally active, when the svrServer-Status is active, that is the spool is active. If the spool is inactive (3), the svrJobAccountingStatus is normally inactive.	INTEGER	read-only
svrJobSpoolVolume Name	Current spool volume or directory. This volume provides temporary space for the job files awaiting output. The default value is 'Unknown volume name'.	DisplayString	read-only
svrJobAccounting VolumeName	Current accounting volume or directory. This volume provides the space for the accounting information. The default value is 'Unknown volume name'.	DisplayString	read-only
svrJobRefreshJobList	Performs a refresh of the JobList and returns the Id (A or B) of the actual refreshed table. Has to be called before retrieving JobList values.	INTEGER	read-only
svrJobListA	This value lists the ID's and states of up to 280 jobs. Each job reserves 5 bytes in the list. The first 4 bytes save the ID as long integer value, the fifth a value representing the status.	BITS	read-only
svrJobListB	This value lists the ID's and states of up to 280 jobs. Each job reserves 5 bytes in the list. The first 4 bytes save the ID as long integer value, the fifth a value representing the status.	BITS	read-only

Object Identifier	Description	ASN.1-Type	ACCESS
svrJobIdentifier	A value uniquely identifies a job in the spool system.	DisplayString	read-only
svrJobName	The name of this job. The default value is ''	DisplayString	read-only
svrJobOwner	The owner of this job. The default value is ''	DisplayString	read-only
svrJobQueueIdentifier	The identifier of the Queue to which this job is assigned. The default value is 0. The value may also be set with this entry. The value has to be between 1 and 9999. It cannot be set if the job status is 'printing' or 'output-ctrl'.	INTEGER	read-write
svrJobType	The `type` of this job. The default value is unknown.	INTEGER	read-only
svrJobStatus	The status of this job. The default value is unknown (1). Also to lock, release, express or delete a job.	INTEGER	read-write
svrJobCopies	The number of copies of the job you want to print. The default value is 0. The value may also be set with this entry. The value has to be between 1 and 32767. It cannot be set if the job status is 'printing' or 'output-ctrl'.	INTEGER	read-write
svrJobPercent Completed	The percentage of all copies of this job completed. The default value is 0.	INTEGER	read-only
svrJobPrintedCopies	The amount of printed copies of this job. The default value is 0.	Counter	read-only
svrJobPrintedPages	The number of printed pages of this job in total. The default value is 0.	Counter	read-only
svrPrintFileSize	The size of the print file in bytes. To allow also sizes greater 2 or 4 GB it is displayed as string. The default value is '0'.	DisplayString	read-only
svrUserInfo	Additional user information to this print job. The default value is ''	DisplayString	read-only
svrJobForm	The form of this print job. The default value is ''. The value may also be set with this entry. The value has to be of a valid Form-Id. It cannot be set if the job status is 'printing' or 'output-ctrl'.	DisplayString	read-write
svrJobPages	The number of pages in the print file. The default value is unknown (1).	INTEGER	read-only

Object Identifier	Description	ASN.1-Type	ACCESS
svrJobPrintingRange	The printing range of this job. The default value is ''. The value may also be set with this entry. The value has to be of a valid range. Example (max. 64 chars): 1-50;100;3:1-99;120- Instead of '-' may also '_' can be used. Use ' ' to set no printing range. It cannot be set if the job status is 'printing' or 'output-ctrl'.	DisplayString	read-write
svrJobPrinting Resolution	The resolution of the print job in dots per inch (dpi), e.g. 240, 300 or 600 dpi or mrm(0) for multi resolution mode. The default value is -1.	INTEGER	read-only
svrJobDestination	The destination printer name of this job. The default value is ''. The value may also be set with this entry. It must be a name of a valid existing printer. It cannot be set if the job status is 'printing' or 'output-ctrl'.	DisplayString	read-write
svrJobPosition	The page position of the job. With this entry the absolute or relative page position can be set. The syntax is: posabs-<page>-<filecopy>-<print file>-<jobcopy> for absolute positioning. The print file parameter is the number of the file as indicated in the PJM. posrel-<page>-<boundary>-<reference>-<direction> for relative positioning with boundary: no | filecopy | jobcopy reference: curpage | checkpoint direction: backward | forward When reading this entry the last used positioning command is retrieved. The default value is ''.	DisplayString	read-write
svrJobAuditRequestor	The identifier of the info source of audit jobs. The default value is ''	DisplayString	read-only
svrJobAudit ReferenceId	The unique identifier of an audit job. The default value is ''	DisplayString	read-only
svrJobReferenceId	The id of the reference job of this job as a string with 8 characters. The default value is ''.	DisplayString	read-only
svrJobsFor ReferenceId	All jobs for a given reference index as string with 8 characters for each job. Maximum 175 JobIds are shown. The default value is '.	DisplayString	read-only

Object Identifier	Description	ASN.1-Type	ACCESS
prnPrinterNumber	A value which uniquely identifies an entry in the prnPrinterTable table and in the database of the system. The value is the id used from the database. This number needs not to be the same as the instance-identification of the prnPrinterTable entry and needs not to be in continuous ascending order. The default value is 0.	INTEGER	read-only
prnPrinterName	The name of the printer served by the printer server.	DisplayString	read-only
prnPrinterModel	This attribute identifies the make and model of the printer. The default value is ''.	DisplayString	read-only
prnPrinterType	This attribute identifies the `type` of the printer. It's a combination of the driver and the type, separated by a '_', e.g. SPS_CUTSHEET. The default value is ''.	DisplayString	read-only
prnPrinterStatus	The status of this printer. The status may also be set with this entry. The default value is unknown.	INTEGER	read-write
prnAssignedQueue	The identifiers of the queues assigned to this printer. The default value is '', which means that no queue is assigned. The value may also be set or a queue may be added or removed. The syntax is: <queue>[-addl-rem] The queue has to be a known queue by the spool in the range of 1 to 999.	DisplayString	read-write
prnPrinterNPRO	The duration in seconds until an automatic NPRO. The default value is -1. The value may also be set to a value between 0 and 65535. The maximum NPRO time is 12 hours, this is 43200 seconds. Higher setting values leads to this maximum NPRO time.	INTEGER	read-write
prnPrinterForm	This attribute identifies the form of the actually printed job. The default value is ''. The value may also be set with this entry. The value must be a valid FormId (e.g. STD).	DisplayString	read-write
prnPrinterToner	This attribute identifies the toner of the actually printed job. The default value is ''. The value may also be set or a TonerId may be added or removed. The syntax is: <tonerId>[-addl-rem].	DisplayString	read-write

Object Identifier	Description	ASN.1-Type	ACCESS
prnHostname	This attribute identifies the host which is used as print server for this printer. The default value is ''	DisplayString	read-only
prnPrinterAddress	The address of the printer in system specific way. If a printer supports more than one address, they are separated by comma. The default value is ''	DisplayString	read-only
prnPrinterDeviceNode	The Device node of the printer in system specific way. If a printer supports more than one address, they are separated by comma. The default value is ''.	DisplayString	read-only
prnPrinterResolution	The resolution of the printer in dots per inch (dpi), e.g. 240, 300 or 600 dpi or mrm(0) for multi resolution mode. The default value is -1.	INTEGER	read-only
prnPrinterConnection-Type	The connection type of the printer, e.g. * SCSI (2) * IBM370 Channel via PCI-Channel-Adapter (3) * IBM370 Channel via EISA-Channel-Adapter (4) * Escon Channel via PCI-Channel-Adapter (5) * LAN-Attachment (TCP/IP) (6) The default value is unknown (1).	INTEGER	read-only
prnPrinterCheckForm-SeqStatus	When form sequencing is active, jobs with the same form are printed sequentially. The default value is unknown (1). The form sequence check may be switched on or off sequentially by this entry.	INTEGER	read-write
prnPrinterCheck Resolution	When check resolution is active, jobs with unsuitable resolution for the selected printer are not processed. The default value is unknown (1).The check resolution may also be switched on (2) or off (3)by this entry.	INTEGER	read-write
prnPrinterCheck Printer	When check printer is active, jobs with unsuitable printer selection are not processed on any active printer. The default value is unknown (1). The check printer may also be switched on (2) or off (3) by this entry.	INTEGER	read-write
prnPrinterCheckJob-Queue	When check job queue is active, jobs with not suitable job queue regarding the selected printer are not processed. The default value is unknown (1). The check job queue may also be switched on (2) or off (3) by this entry.	INTEGER	read-write

Object Identifier	Description	ASN.1-Type	ACCESS
prnPrinterCheckToner	When check toner is active, jobs with not suitable toner ID regarding the selected printer are not processed. The default value is unknown (1). The check toner may also be switched on (2) or off (3) by this entry.	INTEGER	read-write
mgmAgentName	The agent's fully qualified product name.	DisplayString	read-only
mgmAgentVersion	The current version of the installed SNMP agent.	DisplayString	read-only
mgmAgentRevision	The current revision code of the installed SNMP agent and the used MIB.	DisplayString	read-only
mgmAgentSerial Number	The agent's serial number.	DisplayString	read-only
mgmAgentManufacturer	The agent's manufacturer.	DisplayString	read-only
mgmAgentDeveloper	The agent's responsible developer.	DisplayString	read-only
mgmAgentControl	This variable is used to control the agent process. Self-test (4) is for internal use only, and should not be set by the user. A warm-start (5) causes the agent to reinitialize the MIB database and the most important internal variables. A cold-start (6) causes the agent process to restart and initialize the MIB data with defaults or system parameters. Agent-hold (7) causes the agent-process to terminate itself. Read-access to this object returns the operation-state value (2). After the occurrence of an internal error, the object returns the value error-state (3). This value will not be automatically set to one of the other states. It may only be changed by the operator.	INTEGER	read-write
mgmAgentLogStatus	The state of the logging mechanism.	INTEGER	read-write
mgmAgentLogFile	The logging file name.	DisplayString	read-write
mgmAgentMemoryUsage	The Memory usage in Percent.	INTEGER	read-only
mgmSWLicenceNumber	A value which uniquely identifies an entry in the mgmSWLicenceTable table. The value is an index beginning with zero.	INTEGER	read-only
mgmSWName	The software's licence product name.	DisplayString	read-only
mgmSWVersion	The software's licence product version.	DisplayString	read-only
mgmSWLicenceCode	The software's licence code.	DisplayString	read-only

Object Identifier	Description	ASN.1-Type	ACCESS
mgmSWLicenceStatus	The software's licence status code. No error: - ok The software product is available for use. Hardware driver error codes: - error-ioctldev(-12) ioctl to the device not successful - error-opendev(-11) open to the device not successful - error-tmpfile(-3) open of the tmp-file not successful - error-netstat(-2) netstat-command not executed - error-mac(-1) MAC-address can not be ascertained Licence system error codes: - error-licencetype(1) Type of keycode not known (Incorrect code version or incorrect encoding key) - error-noentry(2) No entry for program in licence file - error-openerr(3) licence file can't be opened - error-MACaddr(4) MAC-Address doesn't correspond to the licence code - error-badcode(5) The licence code doesn't belong to the Licensed software - error-timeexpired(6) The licence code validity time has expired - error-integrity(7) The licence code has an incorrect checksum	INTEGER	read-only

Object Identifier	Description	ASN.1-Type	ACCESS
mgmSWLicenceDays	Positive values: The number of days without a proper software licence status. Negative values: Occurrence of an error (-1): Can't find the usage logging file (-2): Can't open the usage logging file (-3): The usage logging file has an incorrect checksum (-4): The usage logging file doesn't belong to the licensed software (-5): The date of last modification of the usage logging file is newer than the system date (-6): The date of the last proper usage of the software is newer than the system date	INTEGER	read-only
mgmSWLicenceCodeID	The software's licence code identification.	OCTET STRING	read-only
mgmSWLicenceExtModuleID	The software's licence external module identification.	OCTET STRING	read-only
mgmSWLicenceIntModuleID	The software's licence internal module identification.	OCTET STRING	read-only
mgmSWLicenceHWID	The software's licence hardware identification code, e.g. MAC-address.	OCTET STRING	read-only
mgmSWLicenceOptInfo	The software's licence code optional info.	OCTET STRING	read-only
mgmSWLicence ExpDate	The software's licence expiration date.	DisplayString	read-only
mgmSWLicenceIntegrity	The software's licence integrity. unknown(1), good(2), bad(3)	INTEGER	read-only

For detailed information on any of these objects, refer to the description in the MIB file
(`/u/prismapro/cfg/snmp/mibs/PSMIB.txt`), for example ”svrServerStatus”:

```
svrServerStatus OBJECT-TYPE
       SYNTAX INTEGER {
                           unknown(1),
                           active(2),
                           inactive(3)
                           on (4)
                           off (5)
                           halt (6)
                           continue (7)
                           pausing (8)
                      }
       ACCESS read-write
       STATUS mandatory
       DESCRIPTION
              "The active/inactive status of the print server.
               When it is active it can offer service on all
               user clients. When it is inactive it can not
               offer service on any user clients. The print server
               could also be pausing.
               The default value is unknown.
               The status may also be set with on to active,
               off to inactive, set into pausing state with
               halt or maybe continuedfrom the pausing
               state with continue."
::= { serverGeneral 3 }
```

Supported SNMP Protocol Version

The SNMP agent is based on the public domain NET-SNMP Version 5.1.1. It supports protocol Data Units based on version 1 of the SNMP protocol standard definition.

Examples showing the access to the psmib object information

The following two examples illustrate how the psmib objects are accessed by proprietary SNMP commands. This chapter describes how the simple `snmpget` command is used. The next chapter will demonstrate a more flexible command to retrieve MIB information. The first example demonstrates access to the PRISMA printer status information.

Example: Retrieving the status of a printer managed by the PRISMAproduction system.

The status of a printer is managed by the prnPrinterStatus object within the printing system MIB. This object can assume nine integer values, signifying a specific printer state. The assignments are:

```
1   =  unknown
2   =  active
3   =  inactive
4   =  printing
5   =  stopped
6   =  started
7   =  message
8   =  error
9   =  halted
```

For example, a value of prnPrinterStatus = 2 means that the printer is active and an SPS process is running for this printer.

The process loads the current parameter information from its initialization file and checks whether it can connect to the specified printer. If so, the printer status changes from inactive to active. Once a printer has been activated, SPS is ready to receive jobs from any assigned and active queue.

The printer status information can be retrieved from the SNMP agent using the NET-SNMP `snmpget` command, a management utility that retrieves variables from an SNMP entity.

The `snmpget` command has the following syntax:

```
snmpget [ options... ] entity_addr objectID [ objectID.. ]
```

The command parameters are the entity's address and the object identifier name(s) expressed as either dot-notation or the variable name as it appears in the MIB document.

There are also a lot of options possible, which are normally not needed. To get the full syntax information just type '`snmpget`' and see the help output. It is also possible to retrieve the manual information by typing '`man snmpget`'. The community string for access to the SNMP entity is given with the option -c <community>.

A `snmpget` user can use both the dot-notation form and the name-form of an MIB object to specify the desired value.

The two commands to retrieve the state of printer 1 from the SNMP agent are:

```
snmpget -c public localhost
.1.3.6.1.4.1.1552.102.104.1.3.1.1.5.0
```

```
snmpget -c public localhost prnPrinterStatus.0
```

The `snmpget` application responds to the two program calls with a equivalent printout on the screen:

```
enterprises.oce.prisma.prismaProduction.psmib.printer.prn-
PrinterTable.prnPrinterEntry.prnPrinterStatus.0 = inac-
tive(3)
```

The output shows the printer state in words `inactive` and the integer return code (3). Note that table objects (such as the prnPrinterState-object) require an additional item of information of the respective row, specifying an individual object from a list of several objects that are managed in a similar way. The corresponding `snmpget` command to retrieve status information from the second printer would be:

```
snmpget -c public localhost
.1.3.6.1.4.1.1552.102.104.1.3.1.1.5.1
```

Example: Retrieving the status of a job served by the PRISMAproduction spool system.

The status of a job is managed by the svrJobStatus object within the printing system MIB. This object can assume nine integer values, signifying a specific job state. The assignments are:

```
1   =  unknown
2   =  ready-to-print
3   =  printing
4   =  output-ctrl
5   =  interrupt
6   =  hold
7   =  error
8   =  locked
9   =  final
```

Again, there are two possibilities to call the `snmpget` command. The first call using the dot-notation-form to specify the object identifier of the svrJobStatus object is:

```
snmpget -c public localhost
.1.3.6.1.4.1.1552.102.104.1.1.2.8.1.6.1
```

If you use the name-form to specify the object identifier of svrJobStatus, the following command must be used for requesting the second entry of the job table.

```
snmpget -c public localhost svrJobStatus.1
```

The following output is generated by the SNMP application.

```
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSys-
tem.svrJobTable.svrJobEntry.svrJobStatus.1 = ready-to-print(2)
```

The output shows the state of the corresponding job in text form ready-to-print and also the integer return code (2).

Retrieving MIB Information using the snmpwalk Command

The `snmpwalk` command has a similar syntax to the `snmpget` command and retrieves classes of variables from an SNMP entity instead of single variables.

The syntax is:

```
snmpwalk [ options... ] entity_addr objectID
```

The command parameters are the entity's address and the variable class name(s). The variable class name is expressed as an object identifier in either dot-notation or as the MIB variable from the MIB document.

There are also a lots of options possible, which are normally not needed. The community string for access to the SNMP entity is given with the option -c <community>. To get the full syntax information just type '`snmpwalk`' and see the help output. It is also possible to retrieve the manual information by typing '`man snmpwalk`'.

The `snmpwalk` command retrieves the variable class by first calling the SNMP entity with the variable class name to get the first variable in the class. Utilizing the GET_NEXT() capability, it then calls the entity again using the variable name returned in the previous call to retrieve the next variable in the class.

The following two examples demonstrate the capability of the command.

Example: Retrieving the status of the job table

The entire job information is contained in the svrJobTable. Using the `snmpwalk` application and applying svrJobTable for the varclass parameter as shown below, the PRISMA spool contents will be displayed.

```
snmpwalk -c public localhost svrJobTable
```

```
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSys-
tem.svrJobTable.svrJobEntry.svrJobIdentifier.11= "00000011"
```

```
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSys-
tem.svrJobTable.svrJobEntry.svrJobIdentifier.234 = "00000234"
```

- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobName.11 = pages9
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobName.234 = pages99
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobOwner.11 = "root" Hex: 72 6F 6F 74
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobOwner.234 = "root" Hex: 72 6F 6F 74
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobQueue Identifier.11 = 1
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobQueue Identifier.234= 1
- enterprises.oce.prisma.prismaProduction.psmibenterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobType.11 = afpds(3)
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobType.234 = afpds(3)
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobStatus.11 = ready-to-print(2)
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobStatus.234 = ready-to-print(2)
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobCopies.11 = 1
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobCopies.234 = 1
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPercent Completed.11 = 0
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPercent Completed.234 = 0
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrinted Copies.11 = 0
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrinted Copies.234 = 0
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrintedPages.11 = 0
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrintedPages.234 = 0

- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrPrintFileSize.11 = 13526
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrPrintFileSize.234 = 756980
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrUserInfo.1 = test_job1
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrUserInfo.234 = test_job2
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobForm.11 = "STD" Hex: 53 54 44
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobForm.234 = "STD" Hex: 53 54 44
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPages.11 = 9
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPages.234 = 99
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrinting Range.1 = ""
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrinting Range.234 = ""
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrinting Resolution.11 = mrm(0)
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPrinting Resolution.234 = 300
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobDestination.11 = Printer1
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobDestination.234 = Printer2
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPosition.11 = ""
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobPosition.234 = ""
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobAuditRequestor.11 = ""
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobAuditRequestor.234 = ""
- enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobAuditRequestor.234 = ""

- `enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobAuditReferenceId.11 = ""`
- `enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobAuditReferenceId.234 = ""`
- `enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobReferenceId.11 = "35FA6754"`
- `enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobReferenceId.234 = "87DC3442"`
- `enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobsForReferenceId.11 = ""`
- `enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobsForReferenceId.234 = ""`

The contents of the PRISMAproduction spool system shown below are retrieved by `snmpwalk` in the following order:

Content of the svrJobTable

Identifier	Name	Owner	Queue Identifier	Type	Status	Copies	Percent Completed	Printed Copies	Printed Pages
00000011	pages9	root	1	afpds	ready-to-print	1	0	0	0
00000234	pages99	root	1	afpds	ready-to-print	1	0	0	0

Content of the svrJobTable (continued)

PrintFile Size	UserInfo	JobForm	JobPages	JobPrinting Range	JobPrinting Resolution	Job Destination
13526	test_job1	STD	9	""	mrm(0)	Printer1
756980	test_job2	STD	99	""	300	Printer2

Content of the svrJobTable (continued)

JobPosition	JobAuditRequestor	JobAuditReferenceId	Job ReferenceId	JobsForReferenceId
""	""	""	'35FA6754'	""
""	""	""	'87DC3442'	""

Example: Retrieving the entries of the prnPrinterTable

The information from the printer table can be retrieved from the SNMP agent using the `snmpwalk` command, as described above.

PrnPrinterTable						
Number	Name	Model	Type	Status	Queue	NPRO
1	Printer1	PAGESTREAM 145	SPS_FANFOLD	inactive(3)	1	0
2	Printer2	PAGESTREAM 235	SPS_FANFOLD	inactive(3)	1	0
3	Printer3	PAGESTREAM 470	SPS_TWIN	active(2)	1	0
4	Printer3	PAGESTREAM 470	SPS_TWIN	active(2)	1	0

PrnPrinterTable (continued)						
Form	Toner	Hostname	Address	DeviceNode	Resolution	Connec-tionType
	""	TestSystem.ops.de	0x05	/dev/ikp0	mrm(0)	pci370 (3)
	""	TestSystem.ops.de	0x06	/dev/ikp0	240	pci370 (3)
"STD" Hex 535444	""	TestSystem.ops.de	0x04	dev/ikp0	300	lan (6)
"STD" Hex 535444	""	TestSystem.ops.de	0x04	dev/ikp0	300	lan (6)

PrnPrinterTable (continued)				
CheckFormSeqStatus	CheckResolution	CheckPrinter	CheckJobQueue	CheckToner
inactive(3)	active(2)	active(2)	active(2)	inactive(3)
inactive(3)	active(2)	active(2)	active(2)	inactive(3)
inactive(3)	active(2)	active(2)	active(2)	inactive(3)
inactive(3)	active(2)	active(2)	active(2)	inactive(3)

```
snmpwalk -c public localhost prnPrinterTable
```

Assuming the given prnPrinterTable and `smnpwalk` command are used, the `snmpwalk` command will generate the following output:

```
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNumber.0 = 1
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNumber.1 = 2
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNumber.2 = 3
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNumber.3 = 4
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterName.0 = "Printer1"
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterName.1 = "Printer2"
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterName.2 = "Printer3"
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterName.3 = "Printer4"
```

- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterModel.0 = "PAGESTREAM145"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterModel.1 = "PAGESTREAM235"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterModel.2 = "PAGESTREAM470"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterModel.3 = "PAGESTREAM470"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterType.0 = "SPS_FANFOLD"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterType.1 = "SPS_FANFOLD"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterType.2 = "SPS_TWIN"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterType.3 = "SPS_TWIN"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterStatus.0 = inactive (3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterStatus.1 = inactive (3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterStatus.2 = active (2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterStatus.3 = active (2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnAssignedQueue.0 = "1"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnAssignedQueue.1 = "1"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnAssignedQueue.2 = "1"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnAssignedQueue.3 = "1"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNPRO.0 = 0
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNPRO.1 = 0
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNPRO.2 = 0

- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterNPRO.3 = 0
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterForm.0 = ""
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterForm.1 = ""
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterForm.2 = "STD" Hex: 53 54 44
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterForm.3 = "STD" Hex: 53 54 44
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterToner.0 = ""
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterToner.1 = ""
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterToner.2 = ""
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterToner.3 = ""
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnHostname.0 = "TestSystem.ops.de"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnHostname.1 = "TestSystem.ops.de"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnHostname.2 = "TestSystem.ops.de"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnHostname.3 = "TestSystem.ops.de"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterAddress.0 = "0x05"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterAddress.1 = "0x06"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterAddress.2 = "0x04"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterAddress.3 = "0x04"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterDeviceNode.0 = "/dev/ikp0"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterDeviceNode.1 = "/dev/ikp0"

- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterDeviceNode.2 = "/dev/ikp0"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterDeviceNode.3 = "/dev/ikp0"
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterResolution.0 = mrm(0)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterResolution.1 = 240
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterResolution.2 = 300
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterResolution.3 = 300
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterConnectionType.0 = pci370 (3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterConnectionType.1 = pci370 (3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterConnectionType.2 = lan (6)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterConnectionType.3 = lan (6)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckFormSeqStatus.0 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckFormSeqStatus.1 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckFormSeqStatus.2 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckFormSeqStatus.3 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckResolution.0 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckResolution.1 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckResolution.2 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckResolution.3 = active(2)

- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckPrinter.0 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckPrinter.1 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckPrinter.2 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckPrinter.3 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckJobQueue.0 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckJobQueue.1 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckJobQueue.2 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckJobQueue.3 = active(2)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckToner.0 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckToner.1 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckToner.2 = inactive(3)
- enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrinterTable.prnPrinterEntry.prnPrinterCheckToner.3 = inactive(3)

Setting MIB entries using the snmpset command

The values of all OID's with write access can also be set by the user with the snmpset command. The `snmpset` command has the following syntax:

```
snmpset [options...] entity_addr objectID type value
[objectID type value...]
```

The command parameters are the entity's address, the object identifier name(s) expressed as either dot-notation or the variable name as it appears in the MIB document, a single character indicating the type of the value (like i for integer, s for string) and the value to which the OID shall be set.

There are also a lot of options possible, which are normally not needed. The community string for access to the SNMP entity is given with the option -c <community>. To get the full syntax information just type '`snmpset`' and see the help output. It is also possible to retrieve the manual information bx typing '`man snmpset`'.

A `snmpset` user can use both the dot-notation form and the name-form of an MIB object to specify the desired value. The used community name has to be a community name with write access.

Example: To set the printer status with 'on' to active for Printer2 use the following command:

```
snmpset -c test2 localhost prnPrinterStatus.1 i 10
```

If successful the `snmpset` application responds to the program call with the following printout on the screen:

```
enterprises.oce.prisma.prismaProduction.psmib.printer.prnPrint-
erTable.prnPrinterEntry.prnPrinterStatus.1 = on(10)
```

The status of Printer2 has changed to active. This can also be seen with the `snmpget` command.

Summary

Using the SNMP interface, the objects described above can easily be managed across the network using SNMP management applications residing on different operating system platforms.

For further information on a specific object, refer to the corresponding description in the MIB specification.

6.2 Retrieving or setting Job information

To get or set an OID entry with job specific information a table exists. The only secure and usable index to find a specific job is the Job-Identifier (JobId). So this is used as table index. **Example**: To retrieve the status of a job with Id 00004674 use the OID:

```
.psmib.server.spoolSystem.svrJobTable.svrJobEntry.svrJobSta-
tus.4674
```

or

```
.1.1.2.8.1.6.4674
```

Of course the PRISMAproduction SNMP does not hold all information of all jobs in a big table. The table is only simulated and the agent uses the last number of the OID to retrieve the information for that OID, when a GET request is sent. If a GET-NEXT request is sent the agent searches the next numerical valid JobId and returns the information for that OID. E.g.: After Id 00004674 follows Id 00004756. A GET-NEXT request with the above shown OID shows the status of the Job with JobId

00004756. By that way it is possible to get the status of all jobs with the normal retrieve table functions. With
`'snmpwalk -c public localhost svrJobTable'`
all information of all jobs can be retrieved. Be aware, that this may take a while if thousands of jobs are in the system. Internally the snmp agent retrieves the list of valid JobId's at the beginning of the snmpwalk command and uses this list to get the following job specific information. During retrieving that information it could happen, that a job is deleted and the ID is no longer valid. In that case the default value of the OID as defined in the PSMIB is returned, except for the svrJobStatus which is set to 'deljob(10)', to indicate the Job as deleted. In most cases the default value is an "impossible" value, like 0 for svrJobCopies. To find such a value in a big list of jobs after a snmpwalk command indicates that the ID is no longer valid. Try to retrieve the information for that ID again. If the job has been deleted the OID is no longer valid and this will be shown this time.

The oce-pp-snmp.rpm also installs some get and set commands. Following are some examples which show how these commands can be used for job information. A possible result is shown in italic.

```
/u/prismapro/bin/snmpget -c public localhost svrJobQueueIden-
tifier.4674
```
(shows the queue of job with JobId 00004674)
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-Table.svrJobEntry.svrJobQueueIdentifier.4674 = 1

```
/u/prismapro/bin/snmpset -c test2 localhost svrJobQueueIden-
tifier.4674 i 3
```
(sets the queue of job with JobId 00004674 to 3)
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-Table.svrJobEntry.svrJobQueueIdentifier.4674 = 3

```
/u/prismapro/bin/snmpgetnext -c public localhost svrJobQueue-
Identifier.4674
```
(shows the queue of job with the next JobId following after 00004674)
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-Table.svrJobEntry.svrJobQueueIdentifier.4675 = 1

```
/u/prismapro/bin/snmpgetnext -c public localhost svrJobQueue-
Identifier.0
```
(shows the queue of job with the lowest JobId)
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-Table.svrJobEntry.svrJobQueueIdentifier.28 = 1

```
/u/prismapro/bin/snmpwalk -c public localhost svrJobQueueI-
dentifier
```
(shows the queue of all jobs ordered by the JobId)
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-

Table.svrJobEntry.svrJobQueueIdentifier.28 = 1
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobQueueIdentifier.29 = 1
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobQueueIdentifier.35 = 3
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobQueueIdentifier.36 = 1
...

```
/u/prismapro/bin/snmpget -c public localhost svrJobIdentifi-
er.4674 svrJobName.4674
svrJobOwner.4674 svrJobQueueIdentifier.4674
```

(shows the Id, the name, the owner and the queue of job with JobId 00004674)
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobIdentifier.4674 = "00004674"
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobQueueIdentifier.4674 = 3
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobName.4674 = "linedata.0567.0000.pds"
enterprises.oce.prisma.prismaProduction.psmib.server.spoolSystem.svrJob-
Table.svrJobEntry.svrJobOwner.4674 = "root" Hex: 72 6F 6F 74

To get an actual listing of the actual valid JobId's, see next chapter.

Retrieving present JobIds

The present JobIds and their states are saved in a table with entries with up to 280 JobId's in each entry. Details of the table see above. This table exists two times to allow accesses from different remote systems. To retrieve the present JobId's the user first has to get the Id of the now present table ("A" or "B"). By retrieving this Id the table is filled with the present data. Afterwards all entries of the table have to be retrieved to get the present information at the time of 0.9 seconds. The table keeps valid for 0.9 seconds. If another user requests an Id of the table in less than 0.9 seconds, he will get the same table and the table is not refreshed, to avoid inconsistencies. After 0.9 seconds another user refreshes the second table by his call.

All OID's in this svrJobListTable have the appendix "A" or "B" in it's name, so that all names in the PSMIB are unique. With the UCD-SNMP the commands to get the JobId's would be:

```
/u/prismapro/bin/snmpget -c <community> <hostname> svrJobRe-
freshJobList.0
```

(This may take a while, if thousands of jobs are in the system)
The user gets back "A" or "B". In case of "A" the next call has to be:

`/u/prismapro/bin/snmpwalk -c <community> <hostname> svrJobListA`
to get all present valid JobId's with their states.

6.3 Configuring Access Rights for Different SNMP Communities

The file `/u/prismapro/cfg/snmp/snmpd.conf` contains configuration information about the communities defined for the SNMP agent. The cut-out below shows the important information for the default configuration:

```
# By default, the agent responds to the "public" community for read
# only access, if run out of the box without any configuration file in
# place.  The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access as well.

# The following lines change the access permissions of the agent so
# that the COMMUNITY string provides read-only access to your entire
# NETWORK (EG: 10.10.10.0/24), and read/write access to only the
# localhost (127.0.0.1, not its real ipaddress).

# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.

####
# Setup communities in a simple way
rocommunity test1   default
rwcommunity test2   localhost
rocommunity public  default
rocommunity interop default
rocommunity isc-i88 default

####
```

If other access rights or communities are needed, changes to this file have to be made manually by the system administrator.

6.4 Content of the Distribution Package

The OPS PRISMA SNMP and the corresponding printing system MIB are shipped as an RPM packet.

The table shows the generated file/directory structure after installation of the RPM:

Name of file/directory	Object description
/u/prismapro/autoadmin:	Directory for automatic configuration
oce-pp-snmp.dlg	Delegation script for automatic configuration of the OPS PRISMA SNMP RPM packet
/u/prismapro/bin:	OPS PRISMA executables directory
psadmin	Start/Stop script for the SNMP daemon
prisma_snmp	Script to switch from PRISMA SNMP to net-snmp and back
snmpd	Daemon to respond to SNMP requests
snmpget	Query information from an OID using SNMP GET requests
snmpgetnext	Query information from the lexicographically "next" OID in the MIB using SNMP GET NEXT requests
snmpnetstat	Displays network-related information
snmpset	Set the information on an OID using SNMP SET requests
snmpstatus	Display statistics on a network entity
snmp_switch	Script to toggle between PRISMA SNMP and net-snmp
snmptable	Query information from an SNMP table ID using SNMP GET NEXT requests
snmptest	Monitor and manage information on a network entity in an interactive way
snmptranslate	Translates an SNMP object into another form of information
snmpwalk	Query information from a tree of OID's using SNMP GET NEXT requests
/u/prismapro/cfg/snmp:	OPS PRISMA SNMP configuration directory
snmpd.conf	SNMP configuration file
psmib.ini	OPS PRISMA SNMP initialization file
/u/prismapro/cfg/snmp/mibs:	OPS PRISMA SNMP MIB directory
/u/prismapro/cfg/snmp/man/*	OPS PRISMA SNMP manual pages
/u/prismapro/cfg/snmp/man/*	Several manual pages for OPS PRISMA SNMP commands
PSMIB.txt	Printing System Management Information Base (PSMIB)
...	Other UCD-SNMP standard MIB's
/u/prismapro/diag/snmp:	OPS PRISMA SNMP diagnostic directory
psmib.log<date_timestamp>	Logging file of the SNMP Daemon regarding the PSMIB. Exists after first start of the SNMP Daemon

7 Tips and Tricks

7.1 Printing with AFP2IPDS Backend via TCP/IP Connection

PRISMAproduction supports TCP/IP printing on all currently released VarioStream and VarioPrint printers. Additionally successful tests were done with the Venlo VarioPrint 2110 and 3090 printers.

TCP/IP printing is released also for PRISMAproduction POD-Modules and A-Twins without restrictions.

The following LAN adapter cards for the PRISMA server have been certified:

- 1x Intel LAN (82559) on-board 10/100 in server F250
- Fast Ethernet Adapter 10/100TX Intel Pro 100
- DUAL Fast Ethernet 10/100TX, 2x Port, IPsec:
- Intel Pro 100S 2 Port Server Adapter with IPsec, with 3DES Encryption (1xPCI-Slot)
- Fast Ethernet Adapter 10/100TX: Intel PRO 100 Server Adapter (1 x 32 Bit PCI-Slot)
- Gigabit Ethernet Contr. 1000SX, PCI-X: Intel PRO 1000 XF Server Adapter (1 x PCI-X-Slot or 1 x 64/32 Bit PCI-Slot)
- Gigabit Ethernet Contr. 1000TX, PCI-X: Intel PRO 1000 XT Server Adapter (1 x PCI-X-Slot or 1 x 64/32 Bit PCI-Slot)
- 1x Intel LAN on-board Gigabit Ethernet 10/100/1000 TX (8254OEM) in server TX150

Configuring a TCP/IP printer with the Spool Configuration Window, will set the Printer Parameter "DEVICEFLAGS" for automatic PPD protocol detection (0x200). For special cases the automatic detection can be deactivated and the protocol can be set manual to PPD1 or PPD2.

7.1.1 Performance

The performance using TCP/IP is sufficient to drive fast fanfold printers with all applications. To achieve this, the communication between printer (Functional Code) and server (AFP2IPDS Backend), the PPD protocol version 2 must be used. The printer functional code supports PPD 2 protocol starting with Bundle 3.

Small cutsheet printers, that do not demand any performance may be driven on a shared LAN. If more performance is needed, there should be 1 LAN adapter per printer.

Connection has to be peer to peer, direct, using crossed ethernet cables. The LAN adapter cards and the Printer CSI should be configured to 100Mbit/1 Gbit, full duplex.

Any other intermediate devices like hubs, switches, bridges, etc. are used at your own risk and you have to verify yourself that your configuration runs free of collisions, full speed, without further negotiation of protocols.

In the printer configuration on the server side you have to set the following parameters:

PRTACKR=0	(section PRTINFO) =Default. Please do not change to any other value, as this may cause backend abends!
PDSDCK=3	(section PDSINFO or PrintLCDS.JCF spooljob section for LCDS Jobs)

Hint: PDSDCK=3 means no position and data checking. If data checking is activated (PDSDCK less than 3) then PRTACKR is not evaluated and the performance will decrease.

The "TCP/IP performance switched on" tuning-set is supplied for this purpose.

7.1.2 Multi client functionality

What does 'multi client functionality' mean?

Up to 16 print processes (logical printers on different print servers) may be attached to 1 physical printer at the same time. If one of the print processes gets a job to print, it tries to reserve the physical printer for this purpose. If the reservation succeeds, printing commences. If it failed, the process has to wait for a notification from the physical printer, that the printer has been freed.

After all jobs have been printed the print process checks if other print processes are waiting for this printer by means of an inquiry to the physical printer. If other processes are waiting, it may release the printer, leaving it to serve these other processes, but it stays attached. The physical printer will now notify one of the waiting print processes that it is available now. This process may now reserve the printer for itself and print it's jobs. Only one process may reserve the physical printer at a time, all other processes requesting this printer are in wait state. A process that is in wait state may be canceled any time.

System requirements to use multi client functionality:

- I-mode functional code 2.10.08 or higher (bundle 3), 3.05 or higher (bundle 4).
- Enable multi client mode for this module by setting PPDLIB_MULTICLIENT:1 in sraconf.dat. Standard is: disabled.

In the configuration of the print process (li.ini) DEVICEFLAGS must be set to default (DEVICEFLAGS=0x603) and the NPRO value of all logical printers must be > 0.

7.1.3 Troubleshooting

User action in case of TCP/IP connection problems

All the following helpful hints are related to a single printing system which is connected via a TCP/IP lan cable with a PRISMAproduction server.

A.) Physical problems:

Situation	A wrong cable is used! A crossed patch cable has to be used if you connect the server with the printer directly. If you use a switch, you have to use non crossed cables.
Message	`90010050 Printer single_tcpip initialization failure: TIMEOUT waiting for Printer Assignment`
User action	Check the LED's on the network card if the cable is plugged in: Two LED's should signalize the correct function: 100 MBit or 1 GBit speed LED is on and the Activity LED is on

B.) Configuration problems:

Situation	The printer cannot be reached because the linux network configuration is wrong or the lan cable is plugged in on the wrong network card
Message	`90010050 Printer single_tcpip initialization failure: TIMEOUT waiting for Printer Assignment`
User action	Check the network configuration with 'ifconfig -a'. Network mask and IP-address on the server have to correspond to the printers network configuration. If a 'ping' to the printer doesn't work, check if the cable is connected with the correct network cards (printer and server side) **Note**: The assignment 'ethx' <> 'physical network card' may change after reinstalling linux

Situation	The IP-address is wrong, the port number is configured correctly
Message	`90010050 Printer single_tcpip initialization failure: TIMEOUT waiting for Printer Assignment`
User action	A 'ping' to the printer doesn't work Check the network configuration. If the network configuration is correct but the 'ping' doesn't work again, re-boot the server and the printer

Situation	The port number is wrong, the IP-address is configured correctly
Message	`90010050 Printer single_tcpip initialization failure: TIMEOUT waiting for Printer Assignment`
User action	A 'ping' to the printer is working fine Check the network configuration. If the network configuration is correct but the 'ping' doesn't work again, re-boot the server and the printer

C.) Handling problem:

Situation	IP-address and port number are configured correctly, but the wrong channel is online
Message	`90010050 Printer single_tcpip initialization failure: TIMEOUT waiting for Printer Assignment` **Note**: A 'ping' may work fine, if the TCP/IP channel is attached or not!
User action	set the correct printer channel to ONLINE

D.) 'Active Session' problems:

Situation	IP-address and port number are configured correctly, the correct channel is online but you get an ' initialization failure'
Message	see above
User action	Set the TCP/IP channel to OFFLINE and ONLINE, a hanging session should be closed if a 'ping' is working but you don't get a connect again, reboot the server and the printer

Situation	Another server is connected with this printer and the session is active
Message	no error message! The printer's state is 'waiting'
User action	Is another server connected to this printer? Disconnecting the printers channel should close this session (try to find the other server first)

7.2 Channel Driver Utilities

7.2.1 General /370 Channel Information

The connection to the /370 channel depends on the customer's environment:

- Max. count for supported boards: 4
- Since V3.02.19 it is possible to attach 2 printers to one /370 channel board. Before this was configurable for one twin system only.
- The server operates as a host on the channel (Host System):
  Connect the cable supplied with the card on one side to the first device on the channel and on the other side to the external connector on the board. The terminator resistors are already on the board, so the channel is correctly terminated.
- The server operates as a device on the channel (Control Unit):
  In this case a switch box is required with bus in/out and tag in/out connectors on it and. an additional cable must be used for connecting the switch box to the external connector of the board. The box requires a separate power supply.
  If the box operates as the last device on the channel, the BUS-OUT and the TAG-OUT ports on the box must be terminated.
- No configuration has to be done, because of using different device names for each operation mode (Host or Control-Unit).

Devicenodes for Host operation mode (Printing on a printer):

`/dev/i370/chan#/YYY`	`#`	Board number 0 - 3
	`YYY`	`"hsa" or "hsb"`

hsa: Used for the first printer.

hsb: Used for the 2nd printer.

Devicenodes for Control-Unit operation mode (receiving data from host):

`/dev/i370/chan#/YYY`	`#`	Board number 0 - 3
	`YYY`	`"cua" or "cub"`

cua: This device is used for "Normal" control unit operation mode.

cub: This device is used for "Fast" control unit operation mode.

7.2.2 /370 Channel Driver Control

This program is used to send certain control commands to the driver of the PCI / 370-channel adapter. (The same command is executed during system startup to load the driver, while LINUX enters multiuser mode).

Installation path: `/u/prismapro/bin/i370`

Usage: i370 [-d devicename] cant [parameter]

'-d devicename': Selection of device '`/dev/i370/chan#/YYY`'

#:	Board number 0, 1, 2 or 3
YYY:	hsa, hsb hra, hrb cua, cub

If no device is specified, i370 uses '`/dev/i370/chan0/hsa`' as default.

'cmd' must be one of the following commands. If no [parameter] is specified, some commands will show a short menu to select more values.

Command Description:

start	Start (Load) the /370 channel driver.
stop	Stop (Unload) the /370 channel driver.
restart	Stop and start the /370 channel driver.
status	Display driver status.
ver	Get version string from pci driver.
boards	Show all available /370 channel boards in the system.
reset	Reset and reinitialize the driver
chanstat	Display current channel status
init	Reinitialize driver for selected device.
timeout [sec]	Set the I/O- timeout for host mode to the new value "sec". Switch driver trace on (with special level) or disable trace. Trace appears in the file "/proc/i370" Set new trace buffer size in lines.

Hint: Detailed information about the boards and available firmware version can be read from the file "/proc/i370" while the driver is loaded.

7.2.3 3211 Printer Emulator

A test utility is included in the /370 driver package for testing the online connection (i370prt):

Installation path: `/u/prismapro/bin/i370prt`

Usage: `i370prt [options] [tracefile]`

3203/3211/6262-Printer Emulation Test Utility V 3.00.00

Usage: `i370prt [options] [dumpfile]`

Options:

-h	print this text
-?	print this text
--xX	set trace level (0-9)
-dXXXX	set device name to use, def.: /dev/i370/chan0/cua
-eXXXX	set emulation mode (3203, 3211 or 6262)
-aXX	set channel address to XX (XX = 0x00 - 0xFF or 0 - 255)
-tX	set transmission mode to X (X = 1-7)
-cX	set condition code after exit to X (X = 0 or 3)
-lX	set max. length for a write ccw (ignore overlength)
-fX	set FCB ignore flag to X (X = 0 or 1, 1=ignore)
-qX	specify the position of the 1st length check byte
-wX	set delay in msec before sending DeviceEnd
-bX	set input buffer size for the driver (X = 0-32000)

Channel dump parameters:

	dumpfile	write channel dump file without read ccws
-n	dumpfile	write channel dump file incl. read ccws
-r	dumpfile	read data from channel dump file

Channel trace parameters:

-m	trcfile	generate tracefile
-s	trcfile	analyze trace: Hex output only
-sa	trcfile	analyze trace: Hex + ASCII output
-se	trcfile	analyze trace: Hex + EBCDIC output
		Tracefiles: trcfile.cmd, trcfile.dat

A generated 'ini' file contains descriptions of the above parameters.
If no options are specified, the program looks for a 'i370prt.ini' file in the current directory.
If a trace file is required, the following files will be generated:

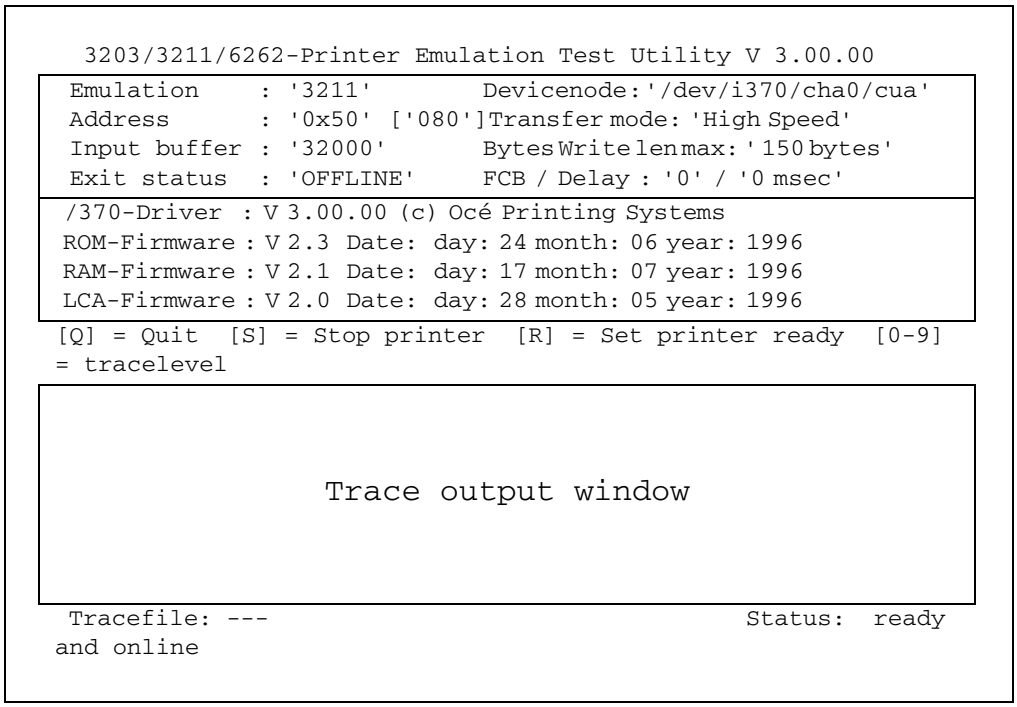
tracefile.cmd - ascii file containing received commands and byte counts
tracefile.dat - bin file containing all received data

The contents of the files depend on the trace level.

When the emulation starts, the program loads an FCB from the file './i370prt.fcb'. If this is not possible, a built-in default FCB is used. If an FCB is received while the emulation is running, it is always saved to the file './i370prt.fcb'.

This utility is fully menu-driven and reacts to the following keys:

'Q'	exits the emulation	
'S'	stops printer (sets printer not ready)	
'R	sets printer ready again	
'0'- '9'	sets trace-	
-> '0'	sets trace-level to 0	No trace
-> '1'	sets trace level to 1 (default)	Display some log infos (trcfile.cmd)
-> '9'	sets trace level to 9	Max. trace



If no command line options are specified and no 'i370prt.ini' file is found, the following defaults are used:

Emulation:	3211
Devicenode:	/dev/i370/chan0/cua
Address:	0x50
Transfer mode:	High speed (=3)
Cond-Code after exit:	3 (= OFFLINE)
Bufsize:	32000 Bytes
Max. Write length:	150 Bytes
FCB ignore:	0 (do not ignore FCB settings)
DE delay:	No delay before sending device end after receiving a block of data.

7.3 Two up Printing

If you are printing with PRISMAproduction, you should never enable the "Hardware Two Up" feature on the printer itself. PRISMA handles all two up printing requests itself, and a mixture of job-defined two up in PRISMA and printer-defined two up will produce all kinds of unpredictable incorrect output and error messages!

7.4 Support of printers with APA II controller

PRISMAproduction – APA Module supports APA II controllers. LCDS and POD Modules can not be used for APA II controllers! The problems using APAII controllers are mainly application related as this controller has limited IPDS command support:

- 600 dpi cannot be used.
- RMF (Relative Metric Fonts) cannot be used.
- All 'OCA' features cannot be used, IOCA, GOCA, BCOCA, ..etc
- DRM and MRM cannot be used.
- N-Up cannot be used.

To be able to use APA II controllers, PRISMAproduction installs the previous used resources, without Relative Metric Fonts into the directories 240dpi.apa, 300dpi.apa and 600dpi.apa. They can be used directly or by changing the links 240dpi, 300dpi, 600dpi.

List of the installed standard resource directories and links:

/u/prismapro/resources/system/240dpi	Link to 240dpi.sra
/u/prismapro/resources/system/300dpi	Link to 300dpi.sra
/u/prismapro/resources/system/600dpi	Link to 600dpi.sra
/u/prismapro/resources/system/240dpi.apa	AFP 240 dpi resources for APA II controller
/u/prismapro/resources/system/300dpi.apa	AFP 300 dpi resources for APA II controller
/u/prismapro/resources/system/600dpi.apa	AFP 600 dpi resources for APA II controller
/u/prismapro/resources/system/240dpi.sra	Standard AFP 240 dpi resources
/u/prismapro/resources/system/300dpi.sra	Standard AFP 300 dpi resources
/u/prismapro/resources/system/600dpi.sra	Standard AFP 600 dpi resources

7.5 System time

Inside a PRISMAproduction cluster the system time of the slaves is synchronized with the master system at boot time and every day change. Therefore system time changes should be done on the master system always.

Index