



**Mercury
User Manual**

Version 10.0

Mercury User Manual

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: Mai 2018

Table of Contents

1	Introduction	10
2	Requirements	10
3	Installation & Licensing	11
4	Principles	13
4.1..	Components	13
4.2..	The Standard Application	13
4.3..	Input Interface In Detail	14
4.4..	Emulation In Detail	14
4.5..	Output Interface In Detail	15
4.6..	Variable Configuration	15
4.7..	Processes And Threads	15
4.8..	Specimen Application	15
5	Program Components	16
5.1..	System Service	16
5.2..	Print Process	16
5.3..	Mercury reference	16
5.3.1	Starting Mercury and logging in	16
	Connecting to a server	17
	Access levels	17
	Logging in	18
5.3.2	The Mercury Client Software	18
	Setup and Configuration	19
	Server	19
	General	19
	Security	21
	Logging	22
	Email Notification	23
	Advanced	24
	FSM Server	25
	Accounting	26
	Print Process Options	27
	Common print process settings	27
	Log levels	28
	Logical Printers	29
	Edit Logical Printer	30
	Test Logical Printer	32
	Run conversion and job monitoring	32
	Job list	32
	Active Jobs	33

Printed Jobs	33
Job Properties	34
Job Properties	34
Advanced Reprint	34
Event view list	35
Menus	35
File	35
Job	36
Setup	37
Help	38
Icon Bar	38
Print Process Bar	38
Watch Print Devices	39

6 Configuration 40

6.1..Introduction	40
6.2..Input Interfaces	40
6.2.1 Overview	40
6.2.2 Channel [ISA]	41
Channel [ISA] (Input Port)	41
6.2.3 Channel [PCI]	42
Channel [PCI] (Input Port)	43
6.2.4 ESCON	44
ESCON (Input)	44
6.2.5 File	44
File (Input)	45
Protocol (Input File Port)	45
6.2.6 IPP	45
Address (IPP Input)	45
Protocol (IPP Input)	46
6.2.7 IPP Server	46
IPP Server	46
TLS/SSL	48
6.2.8 LPD	49
Address (LPD Input)	49
Protocol (LPD Input)	50
6.2.9 Novell	50
Queue Polling (Novell Port)	50
6.2.10 ODBC	50
ODBC	51
6.2.11 Parallel Input	52
Parallel (Input System Device)	52
6.2.12 POP3	52
POP3 Input	53
6.2.13 PPD	55
Address (PPD Input)	55
Protocol (PPD Input)	56
6.2.14 Properties	56
Data (Input Interface)	56
Data Capturing (Input Interface)	56
Data Type	57
Sessions	58
Settings (Input Interface)	59
6.2.15 SNA LU 6.2	60

	Protocol (SNA Port LU 6.2)	60
	SNA LU 6.2 Connection	61
6.2.16	SNA LU1/LU3	62
	Protocol (SNA Port LU1/LU3).....	62
	SNA LU1/LU3 Connection.....	62
6.2.17	System Device	63
	Protocol (Input System Device).....	63
	System Device (Input Port)	63
6.2.18	TCP/IP	65
	Address (TCP/IP Input).....	65
	Protocol (TCP/IP Input).....	66
6.2.19	Twain	66
	Twain (Input Port)	66
6.2.20	XOM	67
	Overview	67
	Requirements.....	69
	Installation	73
	Installation Guidelines	73
	Configuration.....	74
	Configuration Guidelines.....	74
	XOM input.....	85
	Connectivity	85
	Defintions	86
	XOM Command Line Component	88
	Logging	89
	6.3..Emulations	90
6.3.1	Introduction	90
6.3.2	Performance	90
6.3.3	Resources	92
6.3.4	Barcode Label Software	92
	Overview	92
6.3.5	Chained Emulations	92
	Overview	92
6.3.6	Channel Emulation	93
	Overview	93
	Parameters.....	93
	Printer Type.....	93
	Band ID	95
	Translation Table.....	96
	Control Codes	96
	Bold Printing	98
	Download.....	99
	FCB	100
	Job Management	101
	Performance	103
6.3.7	Comparex Line Mode to IPDS/PCL5 Emulation	105
	Overview	105
	Parameters.....	105
	Common	105
	Resources	106
	Job Management	108
6.3.8	Forms Overlay Language	109
	Overview	109
	Parameter.....	110
	Printer Capabilities.....	110

	Paper.....	112
	Paper Assignment	114
	Page Properties.....	115
	Fonts.....	116
	Edit Default Font.....	117
	Performance	118
	Resources	120
6.3.9	ATLAS FD Emulation	120
	Overview.....	120
	Parameters.....	122
	Common	122
	Resources	124
	Performance	126
	Printer settings.....	126
	PDF Export settings.....	128
	Appendix.....	130
	XML Mapping file Specification.....	130
6.3.10	AFP to IPDS Emulation	138
	Overview.....	138
	Parameters.....	138
	Common	138
	Line Data.....	139
	Resources	141
6.3.11	AFP to PCL Emulation	142
	Overview.....	142
6.3.12	IGP to PCL Emulation	143
	Overview.....	143
	Parameters.....	143
	IGP.....	143
	Resources	145
6.3.13	IPDS to PCL5	145
6.3.14	IPDS to PDF	145
	Overview.....	145
	Parameters.....	146
	IPDS to PCL Emulation	146
	Overview.....	146
	Parameters	146
	IPDS Printer.....	146
	Tray Alignment.....	153
	Printer Control.....	154
	PCL Output.....	157
	Input Trays.....	160
	Conversion	163
	Resources	165
	Job Management.....	167
	Miscellaneous.....	173
	IPDS Overlay Supplement.....	177
	Overview.....	177
	Generating Overlay Resources.....	178
	Path to the IPDSOVER.INI file.....	178
	Path of the Resources.....	178
	Structure of the IPDSOVER.INI file.....	178
	Comments.....	179
	Commands.....	179
	Sheet and page counter / Simplex and duplex printing	179

	Definitions.....	180
	Rules.....	181
	Examples for individual functions.....	184
	Example for an application.....	185
	PCL5 to PDF Emulation	186
	Overview.....	186
	Parameters	186
	PDF Common.....	186
	PDF Miscellaneous.....	188
	PDF Security.....	189
	PDF Signatures.....	190
	PCL Defaults.....	190
	Resources.....	192
6.3.15	Job Controller	193
6.3.16	PCL to PDF	194
6.3.17	PCL5 to PostScript® Emulation	194
	Overview.....	194
	Parameters.....	194
	PostScript® Printer.....	194
	Input Bin/Tray Mapping.....	195
	Output Bin/Tray Mapping.....	196
	Job settings.....	196
	Layout.....	197
	Finishing.....	198
	Resources	199
	PCL Defaults	199
6.3.18	PCL5 to X-Metacode	201
	Overview.....	201
	Parameter.....	202
	Emulation.....	202
	Resources	204
	Default Font.....	205
	Xerox Printer.....	206
	Trays.....	207
	Performance	208
	Color.....	210
6.3.19	Prescribe to PCL Emulation	212
	Overview.....	212
	Parameters.....	212
	Prescribe Defaults (FRPO).....	212
	Page Offsets.....	217
	Duplex Offsets.....	218
	Resources	218
	Miscellaneous	218
6.3.20	Program Interface	219
	Overview.....	219
	Parameters.....	219
	Miscellaneous	219
	Performance	221
6.3.21	Raw to Raw	222
	Overview.....	222
	Parameters.....	222
	Common Behavior.....	222
	Performance	223
6.3.22	VIPP Emulation	224

	Overview	224
	Parameters.....	226
	Common	226
	Resources	229
	Global resource path.....	230
	Input media mapping.....	231
	Output bin mapping.....	232
	Staple mapping.....	232
	Jog mapping.....	233
	PS interpreter defaults.....	233
	Postscript overlays.....	236
	Insert specifications	237
	Overlay definitions.....	240
	File overlays	244
	Barcode overlays.....	245
	Text overlays.....	247
	Frame overlays.....	250
	Build in variables.....	251
	Gray GEP Key Examples.....	252
	Postscript textscanner.....	253
	Performance	259
	Output driver settings	259
	VIPP Resources	263
	Finding and Preparing the VIPP resources.....	264
	Standard Fonts	267
	Additional Fonts	267
	Encoding vectors.....	269
	Using CID fonts.....	270
	Inserting additional resource files.....	270
	Default properties and predefined system parameters.....	271
	Replacement of Ghostscript standard fonts.....	273
	Appendix.....	274
	Fontmapping Table.....	275
	Handling of ppd files with non standard keywords.....	276
	Concatenating raw data files with VIPP startup commands.....	279
6.3.23	XES to PCL5	280
	Overview	280
	XES Commands.....	280
	Supported commands	280
	Not supported commands.....	281
	Parameters.....	282
	Common Page Settings.....	282
	Offsets	282
	Scaling.....	285
	Number Of Copies.....	286
	Duplex Settings.....	286
	Page Layout.....	287
	PCL5 Printer Type.....	288
	Tray Mapping	290
	Performance	292
	Emulation	294
	Resources	296
	Common Behavior.....	299
6.3.24	X-Metacode to PCL	300
	Overview	300

	Parameter.....	300
	Start Command.....	300
	System Generation.....	302
	Miscellaneous.....	304
	Resources.....	305
	Messages.....	306
	Finisher Options.....	306
	Color.....	307
	.6.4..Filters	308
6.4.1	Overview	308
6.4.2	Barcode Filter	308
	Application.....	308
	Barcode SIMM Emulation.....	309
	Barcode and Field Scanner.....	313
6.4.3	Codepage Filter	317
	Application.....	317
	NT Language Pack.....	317
	Configuration.....	318
	Patch Table.....	318
6.4.4	Conditional Stringchange Filter	319
	Application.....	319
	Configuration.....	319
	Definition of a trigger.....	320
	Definition of a replacement.....	321
6.4.5	CPCA Remover Filter	321
	Application.....	321
	Configuration.....	322
6.4.6	Field Scanner Filter	322
	Application.....	322
	Configuration.....	322
	Triggered Raw Byte Stream.....	325
	Triggered PCL Job.....	327
	PCL Job Scanned For Text Areas.....	329
	CSV Formatted ASCII Data.....	331
	ASCII Formatted With Fix Column Width.....	332
	Reporting To Later Components.....	333
	Saving To ODBC Database.....	333
	Data Source.....	333
	Data Table.....	334
	Data Fields.....	334
	Writing EASYWARE import files.....	335
	General.....	335
	Data Fields.....	336
6.4.7	Job Copier Filter	336
	Application.....	336
	Configuration.....	336
6.4.8	Job Separator Filter	337
	Application.....	337
	Job Separation.....	337
	Banner Pages.....	339
	Miscellaneous.....	340
6.4.9	Pre-/Post-Job Filter	341
	Application.....	341
	Configuration.....	341
6.4.10	Recordchange Filter	341

	Application	341
	Configuration.....	342
6.4.11	Regular Expressions Filter	353
	Application	353
	Configuration.....	353
	Definition of a replacement	354
6.4.12	Stringchange Filter	354
	Application	354
	Configuration.....	355
	Definition of a replacement	356
6.4.13	Textextractor Filter	356
	Application	356
	Configuration.....	356
	Optimize	356
	Delimiter.....	357
	.6.5..Overlay Manager	359
6.5.1	Overview	359
6.5.2	Configuration	359
	ASCII / PCL Format.....	359
	Profile.....	359
	General	361
	Overlays	362
	Pages without overlay	364
	Fonts	365
	.6.6..Output Ports	365
6.6.1	Overview	365
6.6.2	Properties	366
	Fiery.....	366
	Settings.....	367
	Device Assignment.....	368
	Job Monitoring.....	369
	Data (Output Port).....	371
	Data Capturing (Output Port).....	372
	Spooling	372
	Scheduled printing	375
6.6.3	Channel	375
	Channel Output	375
6.6.4	System Device	375
	System Device (Output Port)	375
6.6.5	File	376
	Filename (Output File Port).....	376
6.6.6	Printers	377
	Printers.....	377
6.6.7	SMTP	377
	SMTP	377
6.6.8	TCP/IP	378
	Address (TCP/IP Output).....	378
	Protocol (TCP/IP)	379
	LPR Output.....	379
6.6.9	IPP Client	380
	IPP Output.....	380
6.6.10	Fiery	382
	Fiery Output	382

Index

383

1 Introduction

Mercury connects printing systems of different performance classes to various LAN and host topologies. The usage of an emulation enables conversion of various data streams into standard data formats (such as PCL and PDF) which can be understood by the printing system.

Mercury therefore secures investments in expensive hardware and specially developed software. Thus high performance printers can be used by non-compatible data sources as well.

Moreover, older printer models can be replaced with more powerful printing systems.

2 Requirements

System requirements for **Mercury**:

Hardware: Intel® or AMD CPU (Core2Duo 6xxx 1,8 GHz or Pentium/Celeron/Athlon/Duron with 2,5 - 3 GHz), 512MB free RAM (for running multiple emulations in parallel, 1GB or more free RAM is recommended). Hard drive capacity according to the amount of spooled files and resources (for the **Mercury** program files additional 250 MB hard drive capacity should be available), monitor, keyboard, mouse, USB port, CD-ROM drive. TCP/IP connectivity over Ethernet or Token Ring.

Operating system: Microsoft® Windows® Vista/7/8/8.1/10 or Windows Server™ 2008/Windows Server™ 2008 R2/Windows Server™ 2012/Windows Server™ 2012 R2/Windows Server™ 2016. Due to Microsoft® workstation operation system limitations in the number of TCP/IP connections, that could be used in parallel, the server versions are recommended.

Microsoft®, MS-DOS™, Windows®, Windows NT®, Windows Server™ are either registered trademarks or trademarks of the Microsoft Corporation in the USA and/or other countries.

Mercury uses Info-ZIP's compression utility. Info-ZIP's software (Zip, UnZip and related utilities) is free and can be obtained as source code or executables from various anonymous-ftp sites, including <ftp.uu.net:/pub/archiving/zip/>.

Xerox® VIPP™ is a registered trademark of Xerox® Corporation.

Adobe®, PostScript®, PostScript® 3™ and the PostScript® logo are trademarks of Adobe Systems.

This software is based in part on the work of the Independent JPEG Group.

Portions of this software are Copyright © 1993-2003, Chad Z. Hower (Kudzu) and the Indy Pit Crew - <http://www.nevrona.com/Indy/> (see also Indy License)

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit - <http://www.openssl.org/> (see also OpenSSL License)

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes cryptographic software copyright (c) 2007-2010 Project RNRT SAPHIR.

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

This software uses the AES implementation of Dr Brian Gladman (see also AES License)

This software uses the Blowfish implementation of Eric Young (see also Blowfish License)

This software uses the PCRE library of Philip Hazel, Copyright © 1997-2011 University of


Cambridge (see also PCRE License)

This software uses the LED Icon Set v1.0 - designed for web designers/developers by Aleksandr Kozmenko, <http://led24.de/iconset/>

This software uses zlib data compression library (c) 1995-2017 Jean-loup Gailly and Mark Adler

All trademarks and tradenames are the sole property of the respective companies.

© Software of the companies named.

 Depending on your software license some of the features described in this document may not be available!

3 Installation & Licensing


Installation of the software

To install [Mercury](#) on your computer, it must be set up using an operating system supported by [Mercury](#). Before installing the software ensure that the hardware components for data communication (network adapter, cables and other devices) have been installed correctly.

The following supplied components are required for the installation:

- The [Mercury](#) installation CD
- Your [Mercury](#) product key

Read the license agreement carefully before you install and use the software. If you use the software you declare your agreement to the terms of the license agreement.

 Follow the listed installation steps to set up [Mercury](#) properly on your computer.

The installation of [Mercury](#) essentially consists of six steps:

- 1st step: Boot your computer. Log on with an account that has administrator rights.
- 2nd step: Insert the supplied installation CD into the CD drive in your computer. If the installation program does not start automatically, start Windows Explorer to install [Mercury](#). Open the CD directory [Mercury/Basis](#) using Explorer and from there start the program [Mercury.exe](#)
- 3rd step: Follow the instructions of the installation program. Enter your [Mercury](#) license number as soon as you are asked to do so. You find the license number on the license card supplied with the software.
- 4th step: Reboot your computer as soon as you are asked to do so by the installation program. Log on again with an account that has administrator rights.
- 5th step: Start the [Mercury](#) system service by using the administration applet for Services from the System Control. Select [Mercury Server \(TCP/IP\)](#) from the list of system services and click on Start. To start [Mercury](#) automatically after a system start, you will have to change the start options of the service from Manual to Automatic.
- 6th step: As soon as the system service has been started, you can set up the print processes you require using the [Mercury Client](#) application for setting up Input Ports, Emulations and Output Ports (see [Configuration](#)).

Licensing of the software

The license for operating your software will be provided as an XML file from your dealer You

can also download this license file during the activation process for the software directly from the software activation website. For licensing your software, open the menu 'Import License File' which is available by right-clicking the Windows tray bar icon of the server part of the software. In the subsequent file selection menu, enter the path and name for the license file. Clicking the 'Open' button imports the XML license file into the software. After licensing 7 days will remain to perform the activation of the software. During this time, the software applies a watermark on each printed page.

Activation process for the software

To activate the software, open the menu 'Software activation', which is also accessible by right-clicking the Windows tray bar icon of the server part of the software. A window appears for entering the software product key. After entering the software product key, an associated hardware key is generated, which is required to activate the software. Then click on the 'open web activation' button to move to the website for product activation.

First, select your language version (English / German) for this page.

Enter then the software product key that you have received from your dealer and your e-mail address into the web form. The e-mail address is used for identification purposes and to send the activation code after successful activation of the software.

Then enter the displayed security code in the appropriate input box and click on 'Next' to continue the activation procedure.

In the following form all (*) marked Fields must be filled with your registration data.

After entering the product key and import of the appropriate license file, which you can download here again by clicking on the 'download license' button, into the software, confirm the installation of the software by activating the "License file is enabled" checkbox. After entering the product key and the import of the license file, a custom specific hardware key is generated from the software, which has to be copied into the appropriate input field of the activation form.

After all fields are filled in, click on the button 'Next' to continue the activation procedure.

After the activation procedure has been finished successfully, a confirmation message will be displayed.

A click on the 'OK' button of this message displays the activation code which has been generated for your software installation.

This activation code must be entered into the input field 'Activation key' of the software for a successful activation. A click on the button 'Activate!' within the software completes the activation of the software.

With a successful activation, at the same time an email is sent out with all activation data to the previously registered email address. In addition the activation data will be printed out for your records by clicking on the 'print all data for your records' entry.

Any change or extension or renewal of an already activated license requires a reactivation of the software by using the new product key which has been provided from the dealer for this change, extension or renewal.

A move of an already activated software to another hardware platform requires a prior getting in contact with your dealer.

4 Principles

4.1 Components

The concept behind [Mercury](#) is based on a modular element system of components. Each component has connections, so-called ports, using which they can be combined with other components.

If we take the aim of [Mercury](#) as a basis, it can be divided as follows:

- Acceptance of print data
- Processing or emulation of print data
- Transfer of the processed print data

This initially leads to a rough distinction of components and splits all the modules of the system into three main groups:

Input Interface

The input interface connects the [Mercury](#) system with the host computer. This component makes the connection to the host computer or a data source using the hardware ports of the [Mercury](#) system.

Possible ports include the following

- **Network** (for example Ethernet, Token Ring, SNA)
- **Hardware options** (for example channel adapters)

Emulation

This is where the print data are prepared for the output device.

Output Interface

When the print data has been processed by the emulation or passed through, they are passed on to the output device using this component. In this case, too, the connection is made using the hardware ports of the [Mercury](#) system.

4.2 The Standard Application

The standard application of [Mercury](#) can be based on one component from each of the three main groups:

- Input Interface
- Emulation
- Output Interface

The example describes the easiest way of outputting print data from a host computer on a printer using [Mercury](#). The following components are used for this purpose.

Input Interface

The input interface connects the [Mercury](#) system with the host computer. This component makes the connection to the host computer or a data source using the hardware ports of the [Mercury](#) system.

Possible ports include the following

- **Network** (for example Ethernet, Token Ring, SNA)
- **Hardware options** (for example channel adapters)

Emulation

This is where the print data from Host and/or network are prepared for the output device (e.g. PCL b/w printers, PDF color printers, Internet, file, etc.).

Output Interface

When the print data have been processed by the emulation, they are passed on to the output device using this component. In this case, too, the connection is made using the hardware ports of the [Mercury](#) system.

4.3 Input Interface In Detail

Depending on the type of port used, an **Input Interface** may be made up of several components. One example of this is an Input port that receives print data with the PPD/PPR protocol via TCP/IP.

In this case the input interface, which we have previously regarded as a single entity, consists of a transport component, namely the TCP/IP network protocol and the communication component, e.g. the PPD/PPR protocol.

What components are offered or supported by the various input interfaces depends mainly on the hardware port of the [Mercury](#) system that is used by the input interface.

Protocols for communication, however, are one of the most important types of components. These are regarded in the [Mercury](#) system as a property of the input interfaces.

4.4 Emulation In Detail

An **Emulation** generally converts print data from one standard language - for example IPDS - into another standard language - for example PDF. To allow as many applications as possible to use this facility, there is an option to process the data using so-called **Filters** before and / or after the actual Emulation.

In other words the emulation component can be split into Pre Emulation Filter (= Input Filter), Emulation and Post Emulation Filter (= Output Filter).

Examples of filter components include

- Stringchange filter
- Conditional Stringchange filter
- Codepage Filter
- Recordchange Filter

Special filters are:

- Barcode Filter
- Overlay Manager

Filters can be linked arbitrarily, in other words Input Filters A and B can be used as follows:

- **A-B or B-A**, the data pass through both filters in sequence
- **A-B-A-B**, it is possible to repeat the filters in the data flow
- **A-B-C-D**-etc., no limit to the number of filters used

The same applies, naturally enough, for Output Filters.

4.5 Output Interface In Detail

Output interfaces may be made up of several components, depending on the type of port used. Mercury knows DOS devices, Files, TCP/IP with/without LPR protocol, SMTP, Printers and Fiery as possible output interfaces.

What components are offered or supported by the various output interfaces, depends, as with the input interface essentially on the hardware port of the Mercury systems that is used by the output interface.

Protocols for communication are also regarded as a property of the output interfaces in the Mercury system.

Another property of output interfaces is spooling. This is similar to the Protocol, and as such represents a component of the output interfaces, which can be enabled or disabled as required.

4.6 Variable Configuration

The simple application is not always sufficient, but the concept of Mercury allows all components to be combined almost arbitrarily.

4.7 Processes And Threads

The efficiency of a Mercury system is primarily dependent on the hardware. Nevertheless the efficiency is also influenced by the configuration. Mercury is not just a program that is active on the system, but it consists of a number of modules, as described in the section Components.

The operating system distinguishes between processes and threads when it allocates processor time. A process corresponds to an application program that has been started, which in turn can start several threads for the parallel processing of data.

All the configurations described above within a Mercury system can be configured in one process, the **Print Process**.

This Print process uses several threads during its processing of print jobs. For example, several print jobs that arrive at the same time are processed in parallel in several threads.

The efficiency of a process falls with the number of threads, however, and therefore, if a large amount of printing has to be completed, it may be beneficial to distribute the whole print control over several processes to achieve greater performance.

4.8 Specimen Application

The Situation

A global company (automobiles) is using IBM IPDS printers and wants to replace them with modern network printers and high performance b/w printers.

The Problem

The new b/w printers process PCL data and must be connected to an Ethernet network. The

host applications, on the other hand, produce IPDS data.

The Solution

Mercury provides the link between the host and the network printer in the form of a bridge. The IPDS to PCL emulation allows the print data to be converted between the two printer languages. As a result of the module structure of **Mercury**, two 110-page b/w printers can be used, and if necessary other printing systems can also be added.

5 Program Components

5.1 System Service

The system service is the core part of the **Mercury**. It is used as an interface between the various program components.

In Windows NT/2000/XP it is permanently available. It receives the commands from the Client application through TCP/IP and sends them on to the print processes which it must start (emulation and input/output interfaces). The system service must also serve the opposite route for status messages, error outputs and dump information.

5.2 Print Process

The Print Process describes the processing of a print job within the **Mercury** print system. A Logical Printer within a Print Process defines three distinctive yet inter-related components as follows:

- **Input interface** receives the print job from the sender (host computer),
- **Emulation** translates the print job from one printer language to another (e.g. IPDS to PCL5),
- **Output interface** transfers the final print job to the receiver (print server, printer, file).

5.3 Mercury reference

5.3.1 Starting Mercury and logging in

To start the Mercury, click the Windows Start button and choose Mercury from the Programs menu; or, if you have created a shortcut to the Mercury application, double-click the Mercury icon on the desktop.


Before you can monitor or administer Mercury, you must log in to the server using the Mercury client software. By default the Mercury client software is installed on the same system where the Mercury server resides. However you can also install and run the client software on a remote Windows workstation (default for the Linux version).

[Connecting to a server](#)
[Access levels](#)
[Logging in](#)

5.3.1.1 Connecting to a server

To connect to the Mercury server

1. Double-click the Mercury icon on your desktop, or click Start, choose Programs, then Mercury, and then Mercury. The Connect to Mercury Server dialog box appears.
2. Select a Mercury Server from the [server list](#).
3. Click on the appropriate access level button.
4. Enter the [password](#) assigned to you by the administrator, and click Connect.
5. The Mercury user interface appears.

 If using the Web based client the default user and password is superadmin / superadmin. Please change the password immediately in the web application.

Server List

Displays the list of the available Mercury servers found in the same LAN (in this particular case in the same IP subnet) and which are running.

Password

Access to a Mercury server can be restricted by means of three passwords. In this way the [access levels](#) to the server are controlled dependent on the password given when logging in.

Connect

Click this button after you have selected a server from the server list.

Add...

Enter DNS name or IP address.

The Add button opens the Add Server dialog box where you can add an additional server to the server list. Enter the domain name or IP address for the server that couldn't be found automatically. See [Connecting to a server](#) for more information.

Remove

Remove selected server from server list.

The button will be disabled if you select a server that is currently connected or has been found automatically.

5.3.1.2 Access levels

The [Mercury](#) server software provides the choice of three access levels: Administrator, Operator and Guest.

Administrator

Administrator access allows full management of the [Mercury](#), including setup and configuration of print processes as well as starting, ending, and monitoring of printing jobs. Administrator access is only available on one computer at a time.

Operator

Operator access allows starting, ending, and monitoring of printing jobs. As Operator, you can

cancel or restart printing jobs.

Guest

Guest access allows monitoring of printing jobs only.

5.3.1.3 Logging in

Connect

Click this button after you have selected a server from the server list.

Add...

Enter DNS name or IP address.

The Add button opens the Add Server dialog box. In this dialog box you can add additional [Mercury](#) server to the server list. If you do not automatically find a server in the list, you can manually add the server DNS name or its IP address. See [Connecting to a server](#) for more information.

Remove

Remove selected server from server list.

The button will be disabled if you select a server that is currently connected or has been found automatically.

5.3.2 The Mercury Client Software

The [Mercury](#) client application main window is displayed the first time a connection to the [Mercury](#) server is established. The main window displays the following sub-windows:

The Upper Window

The upper window displays the logical printers currently configured for this server. The following describes the buttons at the top of the main window:

- **Start Logical Printers** Clicking this button starts the printing process. The Job List window appears. You can not add or remove a logical printer.
- **Stop Logical Printers** When enabled, the Stop Logical Printers icon shows red. Click the button to stop the printing process.
- **Edit Logical Printer** With a printer name highlighted, click this button to access the Edit Logical Printer window, where you can edit properties for each of the characteristics of the printer.
- **Add Logical Printer** Click this button to add a new logical printer.
- **Remove Logical Printer** With a printer name highlighted, click this button to remove the printer.

The Events Window

The lower part of the main window shows a list of events. Initially, the Event view only shows the time that the [Mercury](#) Client has been launched.

As soon as a [connection](#) has been made to the [Mercury](#) Server, any messages from the server and its active logical printers will be shown in the Events window. The message color-coding indicates their order of significance. A difference is made between the following levels:

- Error messages are shown in **red**.
- Warning messages are shown in **blue**.
- Information is issued in **black** lettering.
- Debug messages are issued in **gray**.

5.3.2.1 Setup and Configuration

5.3.2.1.1 Server

From the 'Setup' menu select 'Server...' to access server configuration.

Server properties and options refer to the configuration of the **Mercury** server which is running as system service. You must have [administrator access](#) to use this function.


You can view or modify

- [General](#) server settings
- [Security](#) settings
- [Logging](#) options
- [Email Notification](#) options
- [Advanced](#) options
- [FSM Server](#) options
- [Accounting](#) options
-

General

Connectivity - Server Address

Leave blank if the Print Server which runs **Mercury** server has only one IP address. If multiple IP address's are used, enter the local IP address or DNS name of the network interface which is to be used by **Mercury** server.

 If you enter an IP address for this option and the Print Server's IP address changes at a later time. You will lose communication with the **Mercury** Server. This option will need to be updated if the Print Servers IP address changes.

Connectivity - Computername (Netbios) / Clustername

Normally this field can be left blank. If you want to overwrite the Netbios Computername, especially in a cluster system, it can be defined here.

Connectivity - Port

Currently used TCP/IP port address.

For internal communication the **Mercury** server and client use the TCP port address 49110.

Default: 49110 (not changeable)

Connectivity - Timeout

Set timeout for communication between server and print processes.


If the **Mercury** server is heavily stressed, print processes might be displayed as not responding. Increase this value to resolve this.

Default: 5 seconds

Connectivity - Resolve IP addresses / host names via DNS

If enabled, the Domain Name System (DNS) is used to resolve IP addresses to names.

Enable if DNS is available in the network. All device names and host names must be available via DNS. If not, this feature must not be enabled.

 If enabled although DNS is not available or DNS is not properly configured, system performance will be decreased.

Default: disabled

SNMP interval

SNMP interval in seconds for monitoring output devices

Default: 10 sec

SNMP interval

SNMP timeout in seconds for monitoring output devices

Default: 5 sec

Logical Printers - Run Logical Printers at server start

With this option enabled, as the [Mercury](#) Server starts, it will automatically activate the currently configured Logical Printers.

Default: disabled

Logical Printers - after a delay of

Used to delay the automatic start of Logical Printers by x number of seconds. Delaying the start of Logical Printers might be necessary if more time is needed to load device drivers required by the [Mercury](#).

Default: 0 seconds (no delay)

Logical Printers - Restart not responding Logical Printers

Enabling this option will cause the [Mercury](#) to attempt to recover lost connections. If communication with a print process and its Logical Printers is lost, the [Mercury](#) Server will attempt to re-establish the connection.

Default: disabled

The [Mercury](#) server software provides the choice of three access levels: Administrator, Operator, and Guest.

Database - Data Source

ODBC data source where the [Mercury](#) database tables will be created.

Database - Username and Password

ODBC account for the specified data source.

Database - Job authentication

When checked, the job authentication is activated. Jobs must be authorized, before they are printed. The configuration can be done using the Web-based client. Job authentication is only possible, if the username of the sender is known by [Mercury](#). This is usually the case if an LPD-Input is used.

Security

Security - Administrator Password

Enter password for administrator access.

Administrator access allows full management of the [Mercury](#), including setup and configuration of print processes as well as starting, ending, and monitoring of printing jobs. Administrator access is only available on one computer at a time.

Security - Operator Password

Enter password for operator access.

Operator access allows starting, ending, and monitoring of printing jobs. As Operator, you can cancel or restart printing jobs.

Security - Guest Password

Enter password for monitoring access.

Guest access allows monitoring of printing jobs only.



By default, no passwords are set on the [Mercury](#). If you do not specifically set passwords, all users will have administrator privileges, which include access to important functions such as Setup (including setting passwords), and job control. We strongly recommend that you set at least an Administrator password to protect [Mercury](#) from random or accidental changes to Setup.

Operator Permissions - Operator may start and stop Logical Printers

Check, if the operator may start and stop the Logical Printers.

Default: yes

Operator Permissions - Operator may redirect jobs

Check, if the operator may redirect jobs to another printer

Default: yes

Operator Permissions - Operator may view jobs

Check, if the operator may view job data

Default: no

Operator Permissions - Operator may delete jobs

Check, if the operator may delete jobs

Default: no

Activate TLS/SSL

Activate TLS/SSL encrypted communication (e.g. XMLRPC communication of system service with card readers or PHP backend of webclient).

TLS/SSL Settings (Server) - Certificate

Certificate, which will be used for TLS/SSL encrypted communication. The certificate must be present in the Windows certificate store. The specified text will be search in the subject of the certificate (e.g. CN Name).

TLS/SSL Settings (Server) - Check Certificate...

The specified certificate is searched for in the Windows certificate store. If found, information about the certificate, like serial number or validity date is display. If the certificate was not found, a self-signed certificate can be generated and added to the Windows certificate store. Certificate files in different formats are additionally stored in the SSL subdirectory of the [Mercury-ProgramData](#) directory. If the private key of the generated certificate should be stored in a key file, a passphrase must be specified.

TLS/SSL Settings (Server) - Allow insecure connections

If the option is checked, unencrypted connections are accepted. This is only recommended, if using card readers / printer apps, which do not support TLS/SSL encryption.

Default: no



If the TLS/SSL settings are changed, the client side must be adjusted as well. For adjusting the webclient, please edit the `xmlrpc_ssl`-settings in the file `nps-xmlrpc.host.php` at the webserver. For encrypted communication with card readers / printer applications the appropriate options must be set in the respective configuration.

TLS/SSL Settings (Client) - Trust all certificates

If checked (not recommended), the validity of certificates will not be checked. Use this option only in order to achieve a TLS/SSL encrypted communication without checking the validity of the server certificate.

Default: no

Logging

Log files - Zip log files

If checked, log files will be zipped, if the maximum size is reached. The zip files will be deleted after the specified number of days.

Default: enabled

Log files - Keep one backup

If the maximum log file size is reached, the backup will be renamed ('_BAK' is added) and a new file will be started. This is the fastest method.

Default: disabled

Log files - Shorten log file

If the maximum log file size is reached, the backup will be shortened by 10%.

Default: disabled

Log files - Keep for <x> days

The backup of the log files will be kept for the specified amount of days.

Default: 14

Log files - Size of event log files

Define maximum size of event log file (in MB).

Default: 8 MB

Log files - Debug information

Enable to log debug information of [Mercury](#) server for troubleshooting. Use only for program diagnostics.

Default: disabled

Directory for captured files

Specify the directory where captured data files will be stored.

Email Notification

The [Mercury](#) server software may send a notification email to one or more receivers, if an error occurs.

SMTP server address

IP address or host name of the SMTP mail server. Leave empty to disable the notification feature.

Security

Encryption method for the SMTP connection.

Default: None

SMTP port

IP Port number

Default: 25

Timeout

Timeout for communication with the SMTP server in seconds.

Default: 5

User name

Enter the name of the mail account if the mail server requires authentication.

Password

Enter the password if the mail server requires authentication.

From address

Sender of the mail. Specify a valid email address.

From

Name of the sender. Will be displayed in a receiving mail program.

Email address(es)

Email address to which the email should be sent to. Multiple addresses may be specified separated by semicolons.

Advanced**Web administration - Enabled**

When checked, the [Mercury](#) server extension will be activated. The Server Extension provides additional functionality for the Web based client.

Follow2Print - Enabled

Checking this option will enable [Follow2Print](#). That is all printers which have card readers attached which read the users key cards will automatically print documents associated to the user. For using this feature properly, you have to assign key cards to users and associate card reader terminals to printers.

Stop printing further documents after logout from card reader

If this option is checked, the printing will stop after the current job is printed, if the user has logged out from card reader. If this option is not active, all jobs associated to the user are printed, even if the user has logged out.

Print only a single job on each authentication

If this option is checked, the printing will stop after a single job is printed. To print the next job, a new authentication on the card reader is required.

Newest jobs first (last-in-first-out)

If this option is checked, the newest job will be printed first.

Ignore leading zeros of card reader

If this option is checked, leading zeros delivered by some card reader models are ignored.

Card reader password

Password for the communication between [Mercury](#) and the card readers.

Connection to Index Server

IP Address of the index server. An index server is needed, if a print job should follow to an other [Mercury](#) server.

Time after a print job will be reported to the index server

Time in minutes after a waiting [Follow2Print](#) job will be reported to the index server. From now on the job is available at other [Mercury](#) systems for printing.

TCP port

TCP input port, where print jobs from other Mercury systems will be accepted. An appropriate input port, listening at the specified TCP port, have to be configured.

Separator sheets configuration

This option allows configuration of printing separator pages on specific printers. Separator pages are printed whenever a job was aborted by an error and is re-printed. For each printer the option can be enabled and a template file as also a user defined text constant might be configured.

 The Server Extension, Index Server and Follow2Print has to be licensed.

FSM Server**FSM / Accounting Server****Server IP address**

IP Address or hostname of the FSM / Accounting server

Find automatically

If checked, the FSM / Accounting server will be searched automatically by sending UDP broadcasts. Specify an address if you have more than one accounting server in your network or if the server can not be reached by UDP broadcasts.

 This option is only available if the accounting feature is installed and licensed.

System password

The system password of the FSM / Accounting server

Connect timeout

Timeout in seconds to connect to the FSM / Accounting server

Idle timeout

The connection to the FSM / Accounting server is dropped after the specified number of seconds.

Communication timeout

Timeout in seconds for communication with the FSM / Accounting server

Transfer Settings**Send fleet management statistics to FSM server**

If enabled, a summary of printed pages is calculated at a monthly or weekly base (starting at Mondays) and transferred to the FSM server for further reporting purposes.

Retrieve printers from FSM Server

If checked, printer information will be transferred from the FSM server to the Mercury server.

Additional information like location and properties will be available while configuring printers in Mercury. The FSM server will be polled every few minutes to check, if changes of printers occurred.

Use as printer pool for distributed Mercury servers

When importing printers delivered from a FSM server, usually only printers with matching <PRODUCTNICK> Server ID will be imported. In case you configured distributed <PRODUCTNAME> servers, click this option to import all devices to use them as a printer pool.

Get printers now

Initiating the reception of the printers from the FSM server.

Customer / Dealer

Enter the name of the FSM customer or dealer whose printers shall be retrieved. To get all printers use the name 'superadmin' (without quotes) here.

Data transmission password

Data transmission password of the customer, dealer or superadmin as configured at the FSM server.

Mercury Server ID

The definition of a unique Mercury Server Id (aka "NPS Server Id") is mandatory in some cases. If printers will be imported by a XML file at FSM server side, this server id will be used to assign the printers to several Mercury servers. The server id is also required if Follow2Print jobs located on one Mercury server should be printed by a printer that was assigned to another Mercury server.

Accounting

Enabled

When checked, the Mercury accounting system is activated and all jobs are processed within the accounting system.

Transfer user and terminal information to Accounting database

By enabling this option users (names, pins, card numbers) and terminal information are transferred to the accounting system.

Jobs must be authenticated before printing

By enabling this option all jobs will be checked if their owner is allowed to print them.

User inquiry before printing

By enabling this option the costs of a print job will be presented to the user. The user can decide, if he wants to print the job.

Billing active

By enabling this option, the owner of the print job must enter a billing code, if this feature is

enabled at the [Mercury](#) accounting system.

Notify user when job has been printed


The user will be notified if the job has been printed successfully. This function is only available if the user address is known by [Mercury](#). This depends on the configuration and the input ports used. For displaying the notifications the Accounting Popup Tool must be running on the printing client.

Notify user when printing failed

The user will be notified if the job could not be printed. This function is only available if the user address is known by [Mercury](#). This function is only available if the user address is known by [Mercury](#). This depends on the configuration and the input ports used. For displaying the notifications the Accounting Popup Tool must be running on the printing client.

Retrieve Popup address from DNS

If the username of a print job does not correspond to the account name of the user, the address of the Popup Tool may not be determined. Use this option to resolve the username by DNS. Only use this option, if usernames can be resolved by DNS without errors. Otherwise DNS requests may take some time and decrease the performance of the [Mercury](#) printing system.

 Accounting must be installed separately and has to be licensed.

5.3.2.1.2 Print Process Options

Print process properties and options refer to the configuration of its [common settings](#) and [event logging](#) behavior.

 You must have [administrator access](#) to use this function.

Common print process settings

Name

Define name of the print process.

Max. Concurrent Jobs

Enter maximum number of concurrent jobs.

This value limits the number of print jobs that can be processed at one time by the print process. If Job Splitting is enabled, you might want to set this value to "1" to ensure job integrity. You might also change this value if the print process has to process very large and complicated print jobs or if a very large number of small print jobs occur at the same time.

Default: 0 (unlimited)

Delay job output

Define delay in job output.

If 'Max. Concurrent jobs' is set to 1, it is possible to define a delay in milliseconds which the print process should wait after closing one job and before opening the next.

Default: 0 milliseconds

Notify if there are more than <n> unprinted jobs for a device

If checked, a warning message will be written to the log file, when there are more than the defined count of unprinted jobs queued for a printer or destination address respectively. If you have set the [Email Notification](#) options, the specified email receiver address(es) will additionally be notified by an email.

Default: 0 (Disabled)

Repeat notification after

Defines the time before the next warning message or email will be sent.

If the option 'Notify if there are more than <n> unprinted jobs for a device' is active and continues for a long time, this setting prevents to be get spammed by warning notifications.

Default: 1 hour

Exclude from auto start

Exclude print process from automatically being started at system start.

If checked, the print process will not start at server start, even if the server option '[Run Logical Printers at server start](#)' is checked. The print process will also not be started, when the menu item 'Start all' is selected.

Default: disabled

Log levels

Do not use event logging for this print process

Set if you do not want the print process events to be logged.

Log event entries for

The following options are available for event logging:

- **Error messages:** Logs messages which occur while processing a print job. These are displayed in **red** in the Events Window. Error messages are always logged (default: enabled).
- **Warning notes:** Enable to log warning notes. Warning notes may require intervention and are shown in **blue** (default: enabled).
- **Basic information:** Enable to log basic information. Displays basic job information in the Events Window. These messages are displayed in **black** (default: enabled).
- **Extended information:** Enable to log extended information. Displays extended job information in the Events Window. These messages are displayed in **black**. This information is more detailed than the basic information (default: enabled).
- **Debug information for troubleshooting:** Enable to log debug information for troubleshooting. Use only for program diagnostics (default: disabled).



Depending on how many options you select, the volume of outputs of a log file and in the Events Window will increase or fall. The use of debug information drastically reduces the processing speed of print jobs and may even lead to rejection of print jobs.

5.3.2.1.3 Logical Printers

Logical Printers allow print jobs that arrive at input port interfaces to be distributed to certain emulations and output port interfaces. All defined Logical Printers are listed in the Logical Printers list in the main window of the Mercury client.

The Logical Printers list contains all defined Logical Printers. In the simplest case, the Logical Printer list contains a single Logical Printer which assigns all print jobs from one input interface to one emulation for data conversion and one output to print the data.

Edit logical printer...

To [edit the Logical Printer](#) double click a selected row in the list or click the 'Edit Logical Printer...' button allows.

Add Logical Printer...

Click 'Add Logical Printer' button or select 'Add Logical Printer...' from the 'Setup' menu to add a Logical Printer (a new row) to the list.

Remove Logical Printer...

Click 'Remove Logical Printer' button or select 'Remove Logical Printer...' from the 'Setup' menu to remove a Logical Printer (the highlighted row) from the list.

Input interface

Shows the type and properties of the input port interfaces and also its [description](#). Used to configure the input port for incoming print data.

Input Filter

Shows the input filters. Input filters compensate incompatibilities between the host and destination printing system or incorrectly formatted commands of character strings before the conversion.

Data conversion

Shows the name of the assigned emulation for data conversion (e.g. IPDS to PCL5) and also its [description](#). Each emulation generally has different settings.

Output Filter

Shows the output filters. Output filters compensate incompatibilities between the host and destination printing system or incorrectly formatted commands of character strings after the conversion.

Output interface

Shows the type and further properties of the assigned output port interface and also its [description](#). Used to configure the output for outgoing print data.

Edit Logical Printer

To edit a Logical Printer, double click on a selected row of the Logical Printers list or click the 'Edit Logical Printer...' button above the Logical Printers list.

The 'Edit Logical Printer' window allows to edit the logical printer, consisting of:

- Input Interface
- Condition of usage
- Emulation
- Output Interface

Input Interface

The required input interface for the print job is to be selected here. The list contains all the existing defined by the user.

Condition of usage

In keeping with the selected condition, the required criterion is to be selected or entered here.

• Boolean Expression

Using this condition type you can combine several system and job properties as a condition of usage for a Logical Printer. If the evaluation of such an expression becomes true the Logical Printer will be used to process the job. The expression will be evaluated by a built-in interpreter, whose parsing syntax is described here: [Interpreter Rules](#). Typically an expression will contain a logical combination of job variables and/or data field variables.

Job variables are built in variables, whose contents will be filled in during processing time of a print job. Some variables will be available not until the job has been completely processed. Those "late" variables can only be used for print jobs which have been sent to a post processing print process. The following **job variables** are available for boolean expressions:

\$creationtime
\$destaddress
\$destport
\$inputdatatype
\$jobid
\$jobname
\$jobnamenoext
\$jobserverid
\$pagecount
\$srcaddress
\$starttime
\$userjobid
\$username

Currently there exists just one built-in **system variable**: `$serverid`, which holds the ID of the Mercury server.

Data field variables are the result of a scan of the [Field Scanner](#) filter module. This filter scans the print job data using one of several scanning methods to detect data fields.

Job property variables and data field variables can also be accessed by Lua scripts which can be called by the built-in interpreter.

Emulation

The required emulation for the print job is to be selected here. The list contains all the existing defined by the user.

Output Interface

The required output interface for the print job is to be selected here. The list contains all the existing defined by the user.

Destination/Device

- **Default device as configured in output port**

No specific destination address is to be defined; instead the default destination defined in the output is to be used.

- **Device as delivered by protocol**

Some input protocols supply a destination address in the data stream. If, for example data are received using the LPD/LPR protocol, the data stream contains the name of the destination printer. If this is the actual name of the destination printer, it may be used. The destination device at which the print job is finally printed is therefore not decided until the system is running.


If the destination address supplied by the protocol is a virtual output device that does not really exist, this option should not be selected. In this case, by selecting 'Default device as configured in output port' the default destination defined in the output must be used or another destination chosen by selecting 'Specific Device'.

- **Queue Mapping**

When a queue name is delivered by the input protocol, a destination address may be mapped to it. For a TCP/IP output port the destination address can be specified here by entering an address using the format 'address:port [queue]' (e.g. '172.16.1.100:515 [PRINTER1]') where address can be an ip address or a host name and the [queue] parameter is optional. By writing '*' (an asterisk without apostrophes) as source queue name the associated destination address will be used as default mapping. In addition the destination address can be routed through an other Mercury server. For that you have to enter a *via* address using the 'address:port' format.

There are two mapping definition modes: either you use the mapping table of the Printer/Queue Mapping dialog to enter the definitions as described above or you can edit an external mapping file by using a text editor. The external mapping file uses the INI file format and must contain a section in square brackets named *Queue Mappings*. After the section declaration the mapping definitions follow as INI parameters. Every parameter has a queue name and a destination address delimited by an equals sign. Example:

```
[Queue Mappings]
QUEUENAME1=192.168.1.20:9100
QUEUENAME2=192.168.1.30:515 [auto] via Mercury-pc:9200
QUEUENAME3=192.168.1.40:9100 via Mercury-pc:515
```

 The mapping table will be activated as soon as the OK Button is pressed. No restart of the print process will be needed!

- **Specific Device**

If this rule has been assigned to a specific output device, its name is displayed here. If a print job takes this route, the destination address defined in the output is overwritten and replaced by the name defined here. Depending on the output type you have selected, the appropriate destination address can be defined here.

File port – Definition of a file name

DOS device – Definition of a port, for example COM1:


Printer – Definition of a printer that must be installed in Windows

TCP/IP port – Definition of an IP address and IP port number

If, for example, the output is a file port, the name of the output file can be defined here. The file name defined in the file port is then ignored.

Use as default

If this Logical Printer is to be used as the default printer for all print jobs that do not comply with a criterion, this option is to be activated.

 Only one Logical Printer can be marked as the default.

Test Logical Printer

A test job can be used to check the configuration of an emulation against the real printer. Additionally, by sending a test job to the printer or the attached output port interface, it is possible to test the communication between the output interface and the printer. There is no host system required while printing a test job.

While in [started state](#), select one of the listed test jobs. This job will be processed as soon as you press the button 'Print'.

You can repeat this step as long as you do not leave this dialog using the 'Close' button.

5.3.2.2 Run conversion and job monitoring

Start

Click 'Start' button or select 'Start' from the 'File' menu to start Mercury. After it has been started successfully, Mercury will be able to receive print jobs for data conversion.

 While running in conversion mode, [Logical Printers](#) can only be viewed, but not edited nor added nor removed.

Stop

Click 'Stop' button or select 'Stop' from the 'File' menu to stop Mercury from receiving print jobs.

 In order to edit, add or remove a [Logical Printer](#), the job conversion process must be stopped.

View Logical Printer

Click 'View Logical Printer' to view a [Logical Printer](#). Select one from the list.

5.3.2.2.1 Job list

The Job list window is used to monitor and to manage print jobs. The list includes print jobs which are waiting to print or they already have been printed. For every item in the list further information is available:

- **Name:** shows job name. The name is extracted from the job itself or automatically created by Mercury.
- **Status:** shows job status. Possible entries are: Deleting, Error, Canceled, Printed, Printing (Byte), Held, Spooled, Connecting, Emulating, Receiving. Those entries can be added by: Paused, Delayed, Delayed until (Date, Time), Kept.
- **Converted:** shows number of converted pages.

- **Printed:** shows number of printed pages.
- **Size:** shows size of print job.
- **Created:** shows when print job has been created.

Job management for spooled jobs is available after selection via the context menu or the [job menu](#). It is possible to pause the printing of a job, to continue printing, to abort or re-start the print job.

Detailed information for a selected job are shown in the [job properties](#) frame on the right hand side.

Active Jobs

For every item in the Active Jobs list the following information is available:

- **Name:** shows job name. The name is extracted from the job itself or automatically created by Mercury.
- **Status:** shows job status. Possible entries are: Deleting, Error, Canceled, Printed, Printing (Byte), Held, Spooled, Connecting, Emulating, Receiving. Those entries can be added by: Paused, Delayed, Delayed until (Date, Time), Kept.
- **Converted:** shows number of converted pages.
- **Printed:** shows number of printed pages.
- **Size:** shows size of print job.
- **Created:** shows when print job has been created.

Job management for spooled jobs is available after selection via the context menu or the [job menu](#). It is possible to pause the printing of a job, to continue printing, to abort or re-start the print job.

Detailed information for a selected job are shown in the [job properties](#) frame on the right hand side.

Printed Jobs

For every item in the Printed Jobs list the following information is available:

- **Date/Time:** shows the date and time when the job has been completely printed.
- **Name:** shows job name. The name is extracted from the job itself or automatically created by Mercury.
- **Status:** shows job status. Possible entries are: Deleting, Error, Canceled, Printed, Printing (Byte), Held, Spooled, Connecting, Emulating, Receiving. Those entries can be added by: Paused, Delayed, Delayed until (Date, Time), Kept.
- **Pages:** shows number of printed pages.
- **Printed:** shows number of printed pages.
- **Size:** shows size of print job.
- **User:** shows name of the user (client) who has printed the job.

Detailed information for a selected job are shown in the [job properties](#) frame on the right hand side.

5.3.2.2.2 Job Properties

Job Properties

The job properties window is a dynamic status display of the job currently selected in the [job list](#) frame. Besides the core job information, as its name, type, size, creation time and ownership, there are further dynamically updated information about its state and available actions.

To view and modify the properties of several jobs at once, highlight the respective jobs.

Depending on its status, the following actions are applicable for one or more jobs:

- **Print:** (re-)print the spooled job(s)
- **Advanced Reprint:** [Reprint](#) some or all pages of the spooled job(s) and/or redirect it to other printers
- **View:** viewing of job data
- **Redirect:** redirection of jobs from the last used device to a different device.
- **Schedule:** delay output of a print job to a specific date / time
- **Pause / Undo Pause:** pause or continue processing the job(s)
- **Keep / Undo Keep:** keep or remove the job(s)
- **Printed:** set or remove attribute 'printed' of the job
- **Delete:** delete the job(s)
- **Cancel:** cancel processing of the spooled job(s)
- **Input Trace File:** retrieve the file which contains the captured job data before the data have been converted by the emulation. This action is only available if the Data Monitoring option was active while processing job data.
- **Output Trace File:** retrieve the file which contains the captured job data after the data have been converted by the emulation. This action is only available if the Data Monitoring option was active while processing job data.



Only the actions are shown that can currently be applied.

Advanced Reprint

Define the pages of the job to reprint it or to send it to the original printer or an other destination or printer.

In the page ranges table first define a range starting with **From Page** and ending with **To Page** for the destination or printer. Depending on this range you can select an other printer to print the pages before the range start and to print the pages after the end of the range. Select <Skip> in the **Destination/Printer** column, if you do not want to print a range of pages.



Requirements for Advanced Reprint:

- Output Spooling must be enabled
- Login as Administrator or Operator
- Only available for PCL or ASCII data. Input Data Type Recognition must be enabled in Input Port to detect data as ASCII or PCL
- If using the Raw-Emulation the option Count Pages must be enabled.

5.3.2.3 Event view list

The lower part of the main window shows a list of events. The Event view list initially only shows the start time of the client after starting the application

As soon as a [connection](#) has been made to a [Mercury Server](#), all the messages of the server and its active print processes will be shown in this frame. The message color coding indicates their order of significance. A distinction is made between the following levels:

- **Error** messages are shown in **red**.
- **Warning** messages are shown in **blue**.
- **Information** is issued in **black**.
- **Debug** messages are issued in **gray**.

See also [Log Levels](#). You can control the types of messages to be given through the [properties](#) of the print process.



More messages also mean more system stress.

The event list shows the following criteria:

- **No.:** Incremental number of event
- **Date/Time:** Date and time when the event was logged
- **From:** Name of the component which logged the event
- **Event:** Textual information about the event

5.3.2.4 Menus

The main menu in the [Mercury Client](#) window contains the entries shown below which display menus with further entries. Frequently required menu actions can also be reached directly through the button bar below the main menu.

- [File](#)
- [Job](#)
- [Setup](#)
- [Help](#)

5.3.2.4.1 File

New...

Creates one (more) of the following:

- **Client Window**
Opens new client window. An additional client window might be helpful if you are running more than one print process.
- **Print Process**
Creates new print process. Enter a name for the new print process.
- **Desktop Shortcut**
Creates a shortcut for the current print process on the Windows desktop. A desktop shortcut might be helpful if you are working with several print processes at the time.

Login...

Starts the dialog to build up the connection to the [Mercury Server](#). The [Mercury Server](#) can be monitored or administrated, respectively, as soon as a connection has been made.

Start

Starts the print process with its Logical Printers for job conversion and printing.

Stop

Ends the print process. No further processing of the data is possible. Active print jobs are always processed completely before the print processes are ended. This can lead to delays when ending a print process.

Quit

Closes down [Mercury Client](#), but doesn't stop a previously started print process.

5.3.2.4.2 Job**Print**

Starts sending the selected print job to the output port interface.

As soon as the output is complete, the job will be deleted from the spooler. To prevent deletion, activate the 'Keep' option in the spooling options of this output interface. This is a general setting for all jobs on this output interface.

To prevent deletion of this job only without affecting the general settings, activate this option using the menu command 'Keep'.

Advanced Reprint

By using [Reprint](#) you can print some or all pages of spooled job(s) and/or redirect it to other printers

View

View job data. PCL and ASCII jobs will be viewed as Portable Document Format (PDF) files. This requires add-on software. Adobe® Acrobat® Reader™ is free software that lets you view and print PDF files. It can be downloaded directly from the Adobe® website. None PCL/ASCII jobs will be viewed in a special window containing a hexadecimal and ASCII display region.

Redirect

You can redirect the selected job(s) from the last used device to a different device. Redirection is available for spooled jobs only.

Schedule

The setting Print at offers a way to delay output of the print job to a specific date / time. The job is held until the specified date / time is reached.

By choosing Print as soon as possible, the print job will be delivered when its output port is not occupied.

To print a job manually, the option Hold must be selected. This means that the print job is not output until you select the Print command.

Pause

Suspends processing of the selected print job.

The job will not be sent to output even if the therefore defined time has reached. The job will not be deleted, even if the preset time to keep has passed.

If Pause is reset, the processing of the job is resumed, i.e. the job will be printed if its time for printing has been reached earlier or deleted if the time to keep has already passed.

Keep

Keeps selected print job in the spooler after sending it to the output port, until it will be manually deleted using menu entry 'Delete' or the attribute 'Keep' is reset.

Printed

Selected print jobs will be marked as printed. If a job was scheduled for printing, its time for printing will be reset and the job is kept in the spooler.

If Printed is reset, the job will be marked as 'Held, kept'.

Cancel

Stops processing of the selected print job. The 'Keep' attribute of a spooled job will be set automatically.

Delete

Stops output of the selected print job and deletes it. This function is available for spooled jobs only.

5.3.2.4.3 Setup**Edit Logical Printer...**

Opens the 'Logical Printer' dialog for editing.

See [edit Logical Printer](#).

Add Logical Printer...

Creates a new Logical Printer.

Click 'Add Logical Printer' button or select 'Add Logical Printer...' from the 'Setup' menu to add a Logical Printer (a new row) to the list.

See [add Logical Printer](#).

Remove Logical Printer...

Removes the highlighted Logical Printer(s).

Click 'Remove Logical Printer' button or select 'Remove Logical Printer...' from the 'Setup' menu to remove a logical printer (the highlighted row) from the list. This may be necessary to add a new logical printer component.

Print process

Opens Print process options.

See [print process](#).

 You must have [administrator access](#) to use this function.

Server

Opens Server options.

See [server](#).

Backup

Saves or restores a complete configuration of a [Mercury Server](#).

The backup includes a ZIP compatible archive file of the entire configuration of the server and all log files.

5.3.2.4.4 Help**Index**

Opens the [Mercury](#) online help with the index for information search by keywords.

Info

Displays version and copyright information for the [Mercury](#) application. Click on information window to return to the program.

5.3.2.5 Icon Bar

Gives fast access to the most important [commands](#):

- Start Print Process
- Stop Print Process
- Edit Logical Printer
- Add Logical Printer
- Remove Logical Printer
- View Logical Printers
- Watch Print Devices

5.3.2.6 Print Process Bar

Shows the current print process with its Logical Printers and allows fast access to a print process via pull-down menu.

Choose one print process from the list and click "Go" to go to the chosen print process.

5.3.2.7 Watch Print Devices

The Print Device window is used to monitor printers as output devices, which are used by Mercury to process the print jobs. The required information can be gained by

- SNMP (Simple Network Management Protocol) with continuously polling of the printer or by
- PJI (Printer Job Language) messages send by the printer during the printing of a job.

The way used to gain information from the output devices is set at the Output Port Interfaces configuration.

In the list each printer is displayed as an item with a symbol. The color of the symbol indicates the printers state.

- Green The printer is available and ready
- Yellow Warning, i.g. low of paper
- Red Not available because i.g. offline
- White Unknown or not available

The state of a printer is detected by existing or missing state information delivered by the printer. States like i.g. 'low paper', 'out of paper', 'low toner', 'out of toner', 'door open', 'paper jam' and 'service required' can be signaled by the printer. If multiple conditions are signaled by a printer, the color of the printers symbol is signaling the most critical condition. If a printer i.g. signals 'low paper' (warning) and 'out of toner' (error), the symbol will be displayed in red color. If a printer signals 'low paper' (warning) and 'low toner' (warning), the symbol will be displayed in yellow color, because both conditions are warnings.

By double clicking a list entry, depending on the selected type of device monitoring, further information about a printer will be shown in a separate dialog. Printers monitored by SNMP are responding with a manufacturer dependent subset or all of the following types of information:

Information	Description
IP Address	IP address, assigned to the printer.
MAC Address	MAC address of the printers network interface.
Hostname	Name, assigned to the printer for DNS resolution.
Model Type	Manufacturer dependent model type description.
System Location	Location of the printer.
System Contact	Email contact of the printers administrator.
Impressions	Number of printed impressions during lifetime or since last power on of the printer.
Running since	Time when the printer was powered on.
Input Trays	The amount of paper available in the paper trays. Amount of paper is given in remaining percent as far as the printer is supporting this type of information.
Markers	The amount of toner available at the printer. Amount of toner is given in remaining percent as far as the printer is supporting this type of information.

Printers monitored using PJI do not deliver a unique structure of information. The amount and extent of information depends on manufacturer and model type.

6 Configuration

6.1 Introduction

Set up the Mercury Server

1. In the Main Window of the Client, open 'Setup' menu.
2. Click 'Server...'.
3. In the dialog configure the server options.

Set up a Print process

1. In the Main Window of the Client, open 'Setup' menu.
2. Click 'Print Process...'.
3. In the dialog configure the print process options.

Set up a Logical Printer

Define the Logical Printers using which the dispatcher distributes print jobs received by the Mercury system. The configuration dialog contains several sections that are described in detail in the following:

- Input Interface: allows you to add and set up the required input port interfaces for a Logical Printer within a print process.
- Input Filter: select and configure the input filters.
- Data Conversion: add and set up the required emulation for data conversion.
- Output Filter: select and configure the output filters.
- Output Interface: select and configure the output port interface.

6.2 Input Interfaces

6.2.1 Overview

Input interfaces are used as reception ports for print data. Mercury supports different types of inputs:

- [Channel \[ISA\]](#)
- [Channel \[PCI\]](#)
- [System Devices](#)
- [ESCON](#)
- [Files](#)
- [LPD](#)
- [IPP](#)
- [Novell](#)
- [ODBC](#)
- [POP3](#)
- [PPD](#)
- [SNA LU1/LU3](#)
- [SNA LU 6.2](#)
- [TCP/IP / Local TCP](#)
- [Twain](#)

Parameters that are independent of the input type are described under the point headed Properties. Special parameters are listed under the relevant input type.

To define a new input

To define a new input interface in the system, select Properties in the context menu of a print process and go to the dialog page headed Input interface. On this page you can delete existing input interfaces by selecting Delete or reconfigure them by selecting Configure. By selecting Add, the user can add a new input interface. A list appears with the available input types, from

which the required type must be selected. After this the configuration dialog for the selected interface type appears automatically.

All the inputs defined in this way are automatically activated and can receive data when the print process is started. Through which emulation the data pass and to which output they are finally sent is defined in the Logical Printer editor. In principle there is no direct link between an input and a certain output, but such a link may be created in the Logical Printer editor where print jobs can be assigned to defined outputs on the basis of various criteria.

6.2.2 Channel [ISA]

The selection of a Channel as an input interface opens a dialog with the following tabs:

[Channel Input Device](#)

[Settings](#)


[Sessions](#)

[Data Type](#)

[Data Capturing](#)

You can choose between 2 channel input devices.

6.2.2.1 Channel [ISA] (Input Port)

 A channel input device must be installed on your Mercury server in order to work with this interface. To install device, execute ChannelInput.exe from the Mercury software package.

Options

Printer Emulation

Select a printer emulation from the pull-down menu.

Available printers are: IBM 3211, IBM 3825 (IPDS), IBM 4245, IBM 4248, IPDS Bridge, SNI 3338, SNI 3348, SNI 3365

Default: IBM 3211

This menu allows to select the type of printer which will be emulated by Mercury to the connected host on the channel interface.

The printers IBM 3211, IBM 4245, IBM 4248, SNI 3338/39 and SNI 3348 are fanfold impact printers which are equipped with exchangeable print bands.

The IBM 3825 is an IPDS printer. When using this emulation, the channel card makes a 'normal printer restart' after the start of the channel driver.

The IPDS Bridge works with the IBM 3825 driver but makes no 'normal printer restart' after the start of the channel driver.

The printer SNI 2050/2075/3365 are cut sheet laser printing systems which are able to print simplex/duplex from several paper trays and to stack the paper into different output trays. Besides those printers are able to use different fonts, forms and overlays when the required resources are stored on the printers' hard disk drive before.

Channel Address

Enter channel address in hexadecimal code.

The channel address identifies the byte address of the emulated printer which is used with channel protocol.

Default: AE (hex) for device 1, AF (hex) for device 2

Channel Timeout

Set timeout of the channel input device (in seconds).

Possible values: 0 - 300 sec

Default: 45 seconds

Don't stop channel input service when stopping print process

Enable to keep from stopping the channel input service when stopping a print process.

When enabled, the Windows NT device will not be automatically stopped when closing the input interface. If you are using two Channel Input devices, it is recommended to use this option for starting and stopping the devices independently.

Default: disabled (single device), enabled (two devices)

Ignore suppress out

Enable to ignore the 'suppress out' signal.

With channel connection there is a signal called 'Suppress Out'. The host computer is using this signal to inform the device that it should not activate the 'Request In' signal by itself. The 'Request In' signal usually is used by the device to inform the host computer about a pending state message.

If 'Suppress Out' is set by the host, the host informs about the inconvenience of state messages. Using the 'Ignore suppress out' switch may change the behavior of the Mercury Channel Input concerning the 'Suppress Out' signal set by the host computer.

Default: disabled

Strip channel commands

Enable to strip channel commands.

This switch controls whether PCC bytes (PCC = Printer Control Character), i.e. control commands for line feed and form feed, are passed to the emulation by the Channel Input or not. The Channel Emulation and the X-Metacode to PCL emulation are in need of these PCC Bytes, the IPDS to PCL emulation does not know about them because line feeds and form feeds are already present in the IPDS data.

Default: enabled



see also [Settings](#)

6.2.3 Channel [PCI]

The selection of a Channel as an input interface opens a dialog with the following tabs:

[Channel Input Device](#)

[Settings](#)

[Sessions](#)

[Data Type](#)

[Data Capturing](#)

You can choose between 2 channel input devices.

6.2.3.1 Channel [PCI] (Input Port)



A channel input device must be installed on your Mercury server in order to work with this interface.

Driver settings

Channel Timeout

Set timeout of the channel input device (in seconds).

Possible values: 0 - 300 sec

Default: 45 seconds

Don't stop channel input service when stopping print process

Enable to keep from stopping the channel input service when stopping a print process.

When enabled, the Windows NT device will not be automatically stopped when closing the input interface. If you are using two Channel Input devices, it is recommended to use this option for starting and stopping the devices independently.

Default: disabled (single device), enabled (two devices)

Options

Printer Emulation

Select a printer emulation from the pull-down menu.

Available printers are: IBM 3211, IBM 3203, IBM 3825 (IPDS), IBM 4245, IBM 4248, IPDS Bridge, SNI 3338, SNI 3348, SNI 3365

Default: IBM 3211

This menu allows to select the type of printer which will be emulated by Mercury to the connected host on the channel interface.

The printers IBM 3211, IBM 4245, IBM 4248, SNI 3338/39 and SNI 3348 are fanfold impact printers which are equipped with exchangeable print bands.

The IBM 3825 is an IPDS printer. When using this emulation, the channel card makes a 'normal printer restart' after the start of the channel driver.

The IPDS Bridge works with the IBM 3825 driver but makes no 'normal printer restart' after the start of the channel driver.

The printer SNI 2050/2075/3365 are cut sheet laser printing systems which are able to print simplex/duplex from several paper trays and to stack the paper into different output trays. Besides those printers are able to use different fonts, forms and overlays when the required resources are stored on the printers' hard disk drive before.

Channel Address

Enter channel address in hexadecimal code.

The channel address identifies the byte address of the emulated printer which is used with channel protocol.

Default: AE (hex) for device 1, AF (hex) for device 2

Strip channel commands

Enable to strip channel commands.

This switch controls whether PCC bytes (PCC = Printer Control Character), i.e. control commands for line feed and form feed, are passed to the emulation by the Channel Input or not. The Channel Emulation and the X-Metacode to PCL emulation are in need of these PCC Bytes, the IPDS to PCL emulation does not know about them because line feeds and form feeds are already present in the IPDS data.

Default: enabled

6.2.4 ESCON

The selection of ESCON as input interface opens a dialog with the following tabs:

[ESCON Input Device](#)
[Settings](#)
[Sessions](#)
[Data Type](#)
[Data Capturing](#)

6.2.4.1 ESCON (Input)

Device

Define the ESCON device.

Default: \\.\ESCONIN-0

ESCON Address

Select ESCON address in hexadecimal code.

Default: 1A

Strip channel commands

Enable to strip channel commands.

This switch controls whether PCC bytes (PCC = Printer Control Character), i.e. control commands for line feed and form feed, are passed to the emulation by the ESCON Input or not. The Channel Emulation and the X-Metacode to PCL emulation are in need of these PCC Bytes, the IPDS to PCL emulation does not know about them because line feeds and form feeds are already present in the IPDS data.

Default: enabled



see also [Settings](#)

6.2.5 File

The selection of File as input interface opens a dialog with the following tabs:

[File](#)
[Settings](#)
[Sessions](#)
[Data](#)

[Data Type](#)
[Data Capturing](#)

6.2.5.1 File (Input)

Enter the file name from which the data is read.

Click on the button on the right-hand side to search for an Input Interface (pattern).

The input interface monitors the specified directory and starts to process the file as soon as it is available and not locked by another process.

It is also possible to enter wildcards. For example, if you enter 'C:\Mercury*.BIN', the system will process all files with the extension '.BIN' in the directory 'C:\Mercury'.

Several patterns, separated by semicolon, can be used. For example, 'C:\Mercury*.IN1;*.IN2;D:\INPUT*.IN3;*.IN4' will process all files having extensions '.IN1' and '.IN2' in the directory C:\Mercury and all files having extensions '.IN3' and '.IN4' in the directory 'D:\INPUT'.

Files can be excluded using a pattern starting with '^'. For example, '^*.txt' excludes all files ending with '.txt'.

Mercury remembers the last entry you have made.



see also [Settings](#)

6.2.5.2 Protocol (Input File Port)

No suitable protocols are available for file ports.

6.2.6 IPP

The selection of IPP as input interface opens a dialog with the following tabs:

[IPP](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.6.1 Address (IPP Input)

Address

- **Bind to IP address**

Define the local IP address or DNS name of the network interface that is to receive the data. If the computer has only one local address, the field may remain empty. If there are several local addresses, set the one that should be used by the Input Port.

Default: empty (none)

Normally this field can remain blank.

- **Port**

Define IPP port on which the data is to be received. With IPP connections using no protocol (direct socket) usually port 631 is used, although selection of the port is free.

The URI of a printer is "ipp://servername.domain/printers/printername". If your IPP client doesn't support the "ipp://" URI scheme or you are not using the default port 631, you can also use "http://servername.domain:port/printers/printername". Don't forget to add the tcp port to the firewall.

Default: 631

- **Input Buffer Size**

Define size of input buffer. This is the internal buffer size for storage purposes when data is received. If data arrives more quickly than it can be processed, it is placed in a buffer.

Default: 256 KB



see also [Settings](#)

6.2.6.2 Protocol (IPP Input)

IPP - Internet Printing Protocol

6.2.7 IPP Server

The selection of IPP Server as input interface opens a dialog with the following tabs:

[IPP](#)
[TLS/SSL](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.7.1 IPP Server

- **Local Address**

Define the local IP address or DNS name of the network interface that is to receive the data. If the computer has only one local address, the field may remain empty. If there are several local addresses, set the one that should be used by the Input Port.

Default: empty (none)

- **TCP port**

Define IPP/IPPS port on which the data is to be received. With IPP connections using no protocol (direct socket) usually port 631 is used, although selection of the port is free.

For unencrypted connections the expected URI of a ipp client is "ipp://servername.domain:port/ipp/print/printername". If your IPP client doesn't support the "ipp://" URI scheme, you can also use "http://servername.domain:port/ipp/print/printername". Don't forget to add the tcp port to the firewall. The specification of the *printername* part in the URI is optional. If printername is given the Destination port parameter in the JobInfo will be initialized with the content of this part of the URI. If your IPP client uses the "ipps://" or "https://" URI scheme the connection will be encrypted, using the certificate specified in the

TLS/SSL settings.

Default: 631

- **AirPrint**

Default: Off

Switching this Option to any other setting than 'Off' enables the automatic discovery of this port on Internet Protocol (IP) networks using Bonjour zero configuration networking. The [Mercury](#) Server then acts as a AirPrint capable printer using the name entered in the **Printer name** configuration field as the service name.

To use this option, the **Bonjour Print services for Windows** have to be installed.

To print to this service from Android devices install the MOPRIA plugin on your mobile equipment. For some (older) Versions of the MOPRIA plugin the http basic authentication mode handling, needed for getting the user name from android devices, is different than for IOS devices. To compensate this discrepancy the AirPrint mode should be set to the "Auto detect" option. Beginning with Android 8 The MOPRIA plugin will be part of the system. It seems that for the actual versions of the MOPRIA plugin the handling of the basic http authentication protocol is compatible with IOS devices, so that the IOS option can be used for also for Android devices.

- **Authentication**

Default: Off

Switching this option to any other setting than 'Off' enables basic http authentication for the http layer of the ipp protocol. This option should only be enabled, if the client supports this authentication mode. This option have to be set to 'Password required' to print from AirPrint devices to a [Mercury](#) Follow2Print Queue. Only valid [Mercury](#) user names (Pin, Password, Card, AD) are accepted. If the [Mercury](#) user authentication does not provide a password, only the username will be checked.

- **Printer name**

Default: empty

If **AirPrint** is switched On, the name entered here will be used as the service name displayed for the [Mercury](#) AirPrint printer in the mobile equipment printer list.

For other IPP clients, the name entered here will be used as the default printer name, if it is not supplied with the URI send from the client.

- **Mercury hostname**

Default: empty

If this field is left empty the fully qualified hostname will be automatically resolved by the port. If the [Mercury](#) server contains more than one network interface and implements multiple hostnames, the hostname which shall be used have to be entered here. If the "Server certificate" field in the TLS/SSL settings is left empty, this name (or the automatically resolved name) will be used for searching the server certificate.

- **Job monitoring**

Default: disabled

If disabled, the current job status is reported back to the client immediately after the job data is written to the [Mercury](#) subsystem. In this case it is upon the client to poll the job status e.g. with subsequent get-job-attributes ipp requests until the job is completely processed by [Mercury](#) . If this option is enabled the job status will be reported to the client after [Mercury](#) have processed the job. Effectively the client then is forced to wait until the job is completely

processed. Depending on the processing operations this may be last some time. Be aware that, if the client is configured using a short timeout period, he may cancel the print job.

- **Windows printer**

Default: disabled

Enable this option, if the client is a windows IPP or IPPS printer. Windows IPP printers ports use a slightly different format for the resource part of the URI:

"<http://servername.domain:port/printers/printername>" for unencrypted connections and "<https://servername.domain:port/printers/printername>" TLS/SSL connections.

- **IPv6 support**

Default: On


Disable this option, if the IPP clients should not use the Internet protocol version 6.

- **Generate GPD Information**

Default: Off

Enable this option, if the data send by the client should be treated as if it were generated by the [Mercury](#) GPD printer driver.

Using together with a suitable [Mercury](#) converter (e.g. the Postscript processor emulation), the data can be optionally converted to generic postscript code and then in turn can be processed from a [Mercury](#) GPD Mapper output port.

 Do not enable this option, if the data send from the client was already created using the [Mercury](#) GPD printer driver.

6.2.7.2 TLS/SSL

- **Server certificate, Overwrite global settings**

If the **Overwrite global settings** switch is disabled the **Server certificate** name shown in the associated edit field will be inherited from the global Server configuration settings of the [Mercury](#) server. If the **Overwrite global settings** switch is enabled, a separate certificate name can be specified. In both cases the name entered here will be used as a subject name search term to find a suitable certificate in the local machine personal certificate store. If no suitable certificate is found and the Self signed certificate option is enabled a suitable certificate will be searched and (if necessary) created in the local machine trusted Root Certification Authorities store. If the Server certificate edit field is left empty, the name specified in the Mercury hostname setting will be used for finding a suitable certificate. For using wildcard certificates you can specify a certificate name with the form **.domainname*.

- **Generate self signed certificate**

Default: On

If this option is turned on and no suitable certificate is found in the local machine personal certificate store, the first time a [Mercury](#) IPPS Server Port is used, a self signed certificate will be created and installed in the local machine trusted Root Certification Authorities store on the [Mercury](#) server. Additionally the certificate will be exported as a cer file in the Res\IPP subdirectory of the [Mercury](#) global resource folder. This certificate in turn will be used for all tls enabled transfers from IPP and AirPrint clients.

If the clients are windows clients, self signed certificates have to be manually installed in the trusted Root Certification Authorities store of each client.

If a CA derived server certificate is used, only the CA certificate have to be installed at all used clients.

If a optional **PFX password** is specified the certificate will be exported additionally in pfx format containing the private key file.


If Self signed certificate option is switched off, a suitable certificate have to be installed manually in the local machine personal certification store on the [Mercury](#) server and, if necessary, on all clients machines.

6.2.8 LPD

The selection of LPD as input interface opens a dialog with the following tabs:

[LPD](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.8.1 Address (LPD Input)

 Microsoft Winsock 2 must be installed on the [Mercury](#) server! This is the case with Windows NT 4.0 Service Pack 3 or higher and Windows 98. With Windows 95 an update available from Microsoft is necessary.

Address

- **Bind to IP address**

Define the local IP address or DNS name of the network interface that is to receive the data. If the computer has only one local address, the field may remain empty. If there are several local addresses, set the one that should be used by the Input Port.

Default: empty (none)

Normally this field can remain blank.

- **Port**


Define LPD port on which the data is to be received. With LPD connections using no protocol (direct socket) usually port 515 is used, although selection of the port is free.

Default: 515

Input Buffer Size

Define size of input buffer. This is the internal buffer size for storage purposes when data is received. If data arrives more quickly than it can be processed, it is placed in a buffer.

Default: 256 KB

 see also [Settings](#)

6.2.8.2 Protocol (LPD Input)


LPD/LPR - Line Printer Daemon/Line Printer Remote printer protocol

6.2.9 Novell

The selection of Novell as input interface opens a dialog with the following tabs:

[Novell](#)
[Settings](#)
[Sessions](#)
[Data Type](#)
[Data Capturing](#)


6.2.9.1 Queue Polling (Novell Port)

 The Netware Client 4.11a for Windows NT or higher from [Novell Inc.®](#) must be installed on the [Mercury](#) server system. You can not use the Netware Client from Windows NT!

[Mercury's](#) Novell Input Interface can service queues on a Netware server. Arriving jobs will be processed in a first-in-first-out basis. [Mercury](#) removes the jobs from the queue and sends it to an emulation and/or printer.

At first create the queues on the Netware server, which should be serviced by [Mercury](#).

Then create a print server which is used by [Mercury](#) to access the queues. This print server has to be assigned to all queues which [Mercury](#) should serve. You may not want to assign other print servers to this queues.

 The Print Server has to be configured in Novell® Netware™ bindery compatible mode.

For more specific instructions on creating queues and print servers see your Netware™ documentation.

Queue Polling

- **Tree / Server name**

Enter the name of the Netware server, which is maintaining queues to be served by [Mercury](#).

- **Print server name**

Define the name of the print server that [Mercury](#) is using to access the server queues.

 see also [Settings](#)

6.2.10 ODBC

The selection of ODBC as input interface opens a dialog with the following tabs:

[ODBC](#)
[Sessions](#)

6.2.10.1 ODBC

The source of data records is accessed as an ODBC data source. This has the advantage, that all database applications, which support this interface, may be used to request data. There's no difference if the data source is e.g. a local MS Access™ file or if you access a database server like SQL Server™ in your network. Any database product supporting ODBC may be used.

The only restriction is that all data required for printing of a job has to be available in one table. If your database table is organized with references to other tables, then you have to create a view using a join statement, which will create a logical table containing all required fields from several physical tables. See the documentation of your database application about views and joins.

Data Source

Define the database by selecting a data source name and a table name. A data source must be defined prior using the ODBC Manager. You may either start this applet by using the [ODBC Manager button](#) or using the system control applet for ODBC32. Open this applet and add a system or a user data source by defining the driver e.g. MS Access™ and specify the database file and the name of the data source. This name will be used as database name in the job definition tree.

- **Name**

This is the name of the data source as defined with the ODBC Manager applet.

- **User Name**

Enter the user name of the database account for access to the data source.

- **Password**

Enter the password of the database account for access to the data source.

- **ODBC Manager...**

Open the ODBC Manager to add, change or remove data source definitions.

- **Connect**

Establish a connection to the data source defined by the parameters. This is mandatory to view tables

Data Table

- **Table**

Select an existing table or enter the name of a database table to be accessed. You may also use wildcards as part of the name as e.g. 'JobId_*' or 'Data_Set_???'. The asterisk stands for any number of unknown characters; the question mark represents one single unknown character.

- **Report data to later components**

If enabled, the data records retrieved from the data source are sent as print job data to any emulation configured with this input port. Otherwise, the emulation is only informed about new records and needs to access the data source by itself.

Default: disabled

- **Delete records after processing**

If enabled after successful output the detected data records are deleted from the database table. Otherwise all records are left and the data table is continuously growing.

Default: enabled

- **Delete table**

If enabled after successful output the detected database table is deleted from the data source. Otherwise the table is left and the data records may be optionally deleted.

Default: disabled

- **Wait x seconds before processing new records**

This value defines a time period. As long as no new records are added to matching tables, this input port assumes the end of any inserts and starts processing the newly added records.

Default: 15 sec

6.2.11 Parallel Input

The selection of Parallel as input interface opens a dialog with the following tabs:

[Parallel](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.11.1 Parallel (Input System Device)

Device

Select required [system device](#) or port from the pull-down menu.

The list contains all the known [system devices](#) on the system. If the required system device is not included in the list, it can be entered manually.

Default: \\.\WT12101

For more information on the Parallel Input Device see the documentation of your parallel input card.

6.2.12 POP3

The selection of POP3 as input interface opens a dialog with the following tabs:

[POP3](#)
[Settings](#)
[Sessions](#)
[Data](#)

[Data Type](#)
[Data Capturing](#)

6.2.12.1 POP3 Input

Mercury is able to receive print jobs as emails from a mail server. To do this, a POP3 account has to be prepared on a mail server using POP3 protocol. All the required parameters to access these print jobs are entered on this dialog page:

Email account (POP3)

- **Server Name**

Enter hostname or IP address of the mail server on which an account has been prepared for receiving print jobs. If a hostname is entered here instead of an IP address, resolving via DNS must be guaranteed.

Default: empty (none)

- **Port**

Enter the TCP port which is to be used to communicate with the mail server. Usually port 110 is used for normal access, or port 995 for SSL access.

Default: 110

- **Security**

If the email account you are fetching messages from supports Secure Sockets Layer (SSL) encryption, you can select this option to have all information sent through a secured connection.

- **User name**

Enter user name of the POP3 account at the mail server.

Default: empty (none)

- **Password**

Enter password of the POP3 account at the mail server.

Default: empty (none)

Print job data must be send as attachment in MIME or UUENCODED format. In the mail body definitions for parameters of the print job can be added by using keywords. Each keyword is assigned a value:

<Keyword>=Value

An assignment as shown above must be entered within one line. The following keywords are supported:

JOBNAME

Name of the print job as Mercury should uses.

DESTINATION

Destination where the print job should be sent to. You can uses this parameter for definition of rules with rule type 'Destination Port/Queue'.

USER

Name of the users which the print job should be assigned to.

EXECUTION

Defines the time at which the print job should be printed. The defined value must be given in the following format:

```
Day.Month.Year Hour:Minute[:Second] [UTC]
```

The number of seconds can be dropped. If `UTC` is not added, the time according to the local system time of the `Mercury Servers` will be used.

DELAY

Defines an amount of time by which the output of the print job should be delayed. The value can be specified using the format '`<x> <unit>`', where '`<x>`' is an integer and the unit is defined by one of the keywords 'sec', 'hour', 'hours', 'day', 'days'.

Message receiving options**• Get complete message (Pass through)**

Receive the complete message and pass it through as read from the mail server without any modifications.

• Get message and attachments separately

Select this to process message and attachments of the email as separate print jobs.

• Get message only

Select to process only the message as a print job. Any attachments of the email will be discarded.

• Get attachments only

Select to process only attachments as print jobs. The message itself will be discarded.

This is the default setting.

• Include header information in message

Check this to add email header information in front of the message. This option is only available if 'Get message and attachments separately' or 'Get message only' is activated.

Default: off

• Message in mime format (EML)

Message will be processed in MIME (aka EML) format. This option is only available, if 'Include header information in message' is checked.

Default: off

Message processing**• Leave messages on server**

Enable to keep messages on the mail server after they are processed. After a restart of `Mercury Server` all mails residing on the mail server will be processed again. Therefore mails

should be deleted from the mail server manually or by another software after they are processed.

Default: off

- **Resolve email address to user name**

If the email address of a user is present in the database, it will be resolved to the user name. If switched off, or if the email address is not configured, the user field of the job properties will contain the email address of the sender. This option is only available, if webclient or [Follow2Print](#) is licensed.

Default: on

- **Reply on error**

Send a reply mail to the user on job error.

Default: on

- **Reply settings**

Settings for reply mail on job error.



see also [Settings](#)

6.2.13 PPD

The selection of PPD as input interface opens a dialog with the following tabs:

[PPD](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.13.1 Address (PPD Input)



Microsoft Winsock 2 must be installed on the [Mercury](#) server! This is the case with Windows NT 4.0 Service Pack 3 or higher and Windows 98. With Windows 95 an update available from Microsoft is necessary.

Address

- **Bind to IP address**

Define the local IP address or DNS name of the network interface that is used to receive the data. If the computer has only one local address, the field may remain blank. If there are multiple local addresses, you can limit to a single IP address. Leaving it blank in such a case makes Mercury receive data from all IP addresses.

Default: empty (use all available IP addresses)

- **Port**

Define PPD port on which the data is to be received. With PPD connections using no protocol (direct socket) usually port 5001 is used, although selection of the port is free.

Default: 5001

Input Buffer Size

Define size of input buffer. This is the internal buffer size for storage purposes when data is received. If data arrives more quickly than it can be processed, it is placed in a buffer.

Default: 256 KB



see also [Settings](#)

6.2.13.2 Protocol (PPD Input)

PPD/PPR- AFP/IPDS printer protocol

6.2.14 Properties

6.2.14.1 Data (Input Interface)

Decompress and decrypt data automatically

Enable to let Mercury decode compressed or encrypted data automatically that comes from another Mercury system.

Default: disabled

- **Password**

Enter password used by the other Mercury system.

**Use job information within data stream
(when receiving data from a Mercury output interface or XML)**

Enable to let the input interface interpret additional job information.

If data is received from a Mercury output interface, additional information about a job (e.g. job name) can be transferred before the print data. If this option is enabled, the input interface interprets this information. To use this feature, the corresponding option of the output interface must also be checked. Job information can also be set, if sending an XML header before the job data.

Default: disabled

6.2.14.2 Data Capturing (Input Interface)

Data Capturing

Data capturing allows to write incoming data to a file before the data is converted by the emulation. This feature should only be used for testing purposes because it slows down the system performance.

Active

Enable to log sent data.

Default directory for the log file is [Mercury application path]\DATA\DSPPATHx, default name of the log file is Ixxxx.DAT. Each job is placed in a separate file.

Default: disabled (not active)

- **Without protocol data / With protocol data**

Set if you want the protocol data to be saved to the log file, too.

Default: Without protocol data

- **Maximum files to capture**

Define the maximum number of files to be captured.

 If the specified maximum number of files is reached, older files will be overwritten.

Default: 99

- **Maximum size (MB)**

Define the maximum number of megabytes per job to be captured.

Default: 0 (unlimited)

Maximum hours to keep

Files will be deleted after the specified number of hours.

Default: 720

- **Add job name to captured filename**

Enable if you want the job name to be added to the captured filename.

The job name will be added in parenthesis to the captured filename.

Default: enabled

6.2.14.3 Data Type

Input Data Type Recognition**Active**

Enable to either determine the type of the incoming data automatically or assign it manually to the data stream.

Default: disabled

- **Always Assume**

Set input data type to a certain incoming data stream.

If the type of the incoming data stream is known, there is no need to use the automatic detection. The data type can be set manually to a fix value here.

Default: disabled (only available when Active is enabled)

- **Auto Detect**

Select input data type(s) that shall be detected automatically.

The data type of the incoming data will be recognized automatically. Several types can be selected, for which the incoming data is tested. Because under some circumstances the real data type might not be recognized for sure, unused data types should be turned off to avoid misleading detection.

Default: disabled, no data type selected (only available when Active is enabled)

6.2.14.4 Sessions

Maximum of input sessions

Enter maximum number of input sessions.


When this number of sessions has been reached, further connection requests by a host are rejected until the number falls below the maximum level again. If no value is entered here or if 0 is entered, the number of sessions is unlimited.

Default: 10

Maximum of pending jobs

Enter maximum number of pending jobs.

When the maximum number of pending jobs has been reached, further print jobs will not be processed. If no value or 0 is entered here, there is no limit to the number of pending jobs.

 If a very large number of print jobs is processed at the same time, this may have a major adverse effect on the system performance.

Default: 10

Session Timeout

Enter session timeout interval (in seconds).

If no data are received from a connection (for example a TCP/IP session) for a certain time, Mercury automatically terminates the connection. This time can be defined here (in seconds). No entry or a 0 in this field prevents the automatic termination of a connection at the input by Mercury.

Default: none (empty)

Job Timeout

Enter job timeout interval (in seconds).

If several print jobs arrive at a connection (for example a TCP/IP session) in sequence, Mercury may not recognize these as separate jobs. This may cause problems if, for example, the data are to be output through the PPD/PPR protocol since this protocol must have access to the complete print job before it is passed on to the printer. Therefore it is necessary to define a job timeout. If no data arrive at a connection for a certain period of time, the current print job is terminated. All further data that arrives at this connection will be assigned to a new print job. No entry or a 0 in this box will prevent print jobs from being split automatically.

 Another possibility of splitting a continuous flow of data into individual print jobs is to


simulate the end of the print job after a certain number of pages and to start a new print job with the following pages. This setting is a property of the emulation that must support it. The appropriate possible settings are given in the configuration dialogs of the emulations.

Default: empty (none)

Look for session every ...

Enter interval in milliseconds when input port shall monitor the input for receiving data (polling).

A file port, for example, checks the specified directory at regular intervals for new data so that it can then process them.

 Frequent monitoring increases the workload of the computer. The value should therefore not be made too low. A higher value will only mean that a received job is identified a little later, so it is quite acceptable for this value to be several seconds.

Default: 1000 msec

Job start sound

Define a job start sound.

At arrival of a print job a sound can be played. Therefore the name of a sound defined in the Control Panel, a wave file or '-1' for a short signal can be entered here. Leave this field empty for no sound.

Default: empty (no sound)

6.2.14.5 Settings (Input Interface)

Buffer

• Read block size

Enter block size for read operations in KB.

When data are received, the data are read in blocks of the specified size as long as the input port supports this. With a file port, for example, the data are read and processed in blocks. The larger the block size, the fewer read access operations are required. A large block size, however, increases the memory requirement for job management.

Default: 32 KB


• Direct Read

Enable to let the input data stream not be buffered and not be read directly from the input interface.

In most cases this setting will improve performance. If Channel Input is used in conjunction with very small block sizes, the time needed for storing the data in a buffer would affect performance negatively. In this case be sure that 'Direct Read' is activated. This setting is only available if no protocol is used.

Default: enabled

Timeouts

 These settings are only accessible by the support

- **Read Timeout**

Enter timeout for data reception.

This value allows the time properties of Mercury for read operations to be influenced. This value should only be changed by experienced users.

Default: 50 msec

- **Write Timeout**

Enter timeout for data output.

This value allows the time properties of Mercury for write operations to be influenced. This value should only be changed by experienced users.

Default: 1000 msec

- **Delay, if data received**

Enter the time in milliseconds that a print job may be delayed in receiving data so that other print jobs can be completed fluently.

Print jobs that arrive at the same time are processed in parallel. Each print job is allocated a certain amount of processor time for this purpose, which it must share with other print jobs.

Default: 0 msec

- **Delay, if no data received**

Enter the time in milliseconds that a print job can be delayed if no data are currently being received for this print job.

If no data are currently being received for a print job, the print job may be delayed and the available processor time can be assigned to other print jobs.

Default: 10 msec

6.2.15 SNA LU 6.2


The selection of SNA LU6.2 as input interface opens a dialog with the following tabs:

[SNA LU6.2](#)
[Settings](#)
[Sessions](#)
[Data Type](#)
[Data Capturing](#)

6.2.15.1 Protocol (SNA Port LU 6.2)

SNA LU 6.2 (together with SNA port)

6.2.15.2 SNA LU 6.2 Connection

 The Microsoft DLC protocol must be installed on the [Mercury](#) server!

Host MAC Address

Define MAC address of the host system from which data is accepted.

Ask your network administrator for the correct Host MAC address.

Default: empty (none)

IDBLK

Set the IBM product number (Hex) which defines the type of the control unit and of the attached devices. This value must refer to the VTAM definition at host side. For IBM printers InfoPrint 60, 3130, 3160, 3935 use value 071.

Possible range is 000 to FFF.

Default: 017

IDNUM

Define the unique identification (Hex) of the control unit. It must be in conjunction with IDBLK the XID value and refer to the VTAM definition at host side.

Possible range is 00000 to FFFFF.

Default: empty (none)

Adapter

Set the logical adapter number of the network interface in the [Mercury](#) Server system, which is to be used for communication with the host system. If there is only one network interface, this value is to be set to 0. With several interfaces installed, here you decide which one is to be used by the input port.

Default: 0

Connect timeout

Set the connection timeout (in seconds). After this interval is elapsed without having established a connection, the [Mercury](#) client is no longer searching for a connection. An error message is displayed.

Default: 30 seconds

Retry connect after

Set the time (in seconds) after which the port will try again to get a connection.

Default: 30 seconds

Remote Network Identifier

Define the name of the network in which the printer resides. For AS/400 hosts use 'APPN'.

Default: empty (none)

Remote Control Point

Define the name the host is using for its application control point. For AS/400 hosts set the name of the controller to be configured.

Default: empty (none)



see also [Settings](#)

6.2.16 SNA LU1/LU3

The selection of SNA LU1/LU3 as input interface opens a dialog with the following tabs:

[SNA LU1/LU3](#)

[Settings](#)

[Sessions](#)

[Data Type](#)

[Data Capturing](#)

6.2.16.1 Protocol (SNA Port LU1/LU3)

SNA LU1 / LU3 - Systems Network Architecture Protocol for Logical Units

6.2.16.2 SNA LU1/LU3 Connection



The Microsoft DLC protocol must be installed on the [Mercury](#) server!

SNA Connection

Host MAC Address

Define MAC address of the host system from which data is accepted.

Ask your network administrator for the correct Host MAC address.

Default: empty (none)

IDBLK

Set the IBM product number (Hex) which defines the type of the control unit and of the attached devices. This value must refer to the VTAM definition at host side. For IBM printers InfoPrint 60, 3130, 3160, 3935 use value 071.

Possible range is 000 to FFF

Default: 017

IDNUM

Define the unique identification (Hex) of the control unit. It must be in conjunction with IDBLK the XID value and refer to the VTAM definition at host side.

Possible range is 00000 to FFFFF.

Default: empty (none)

Adapter

Set the logical adapter number of the network interface in the [Mercury Server](#) system, which is to be used for communication with the host system. If there is only one network interface, this value is to be set to 0. With several interfaces installed, here you decide which one is to be used by the input port.

Default: 0

Connect timeout

Set the connection timeout (in seconds). After this interval is elapsed without having established a connection, the [Mercury](#) client is no longer searching for a connection. An error message is displayed.

Default: 30 seconds

Retry connect after

Set the time (in seconds) after which the port will try again to get a connection.

Default: 30 seconds



see also [Settings](#)

6.2.17 System Device

The selection of a System Device as an input interface opens a dialog with the following tabs:

[Device](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.17.1 Protocol (Input System Device)

No suitable protocols are available for system devices.

6.2.17.2 System Device (Input Port)

Device

Select required device or port from the pull-down menu.

The list contains all the known [system devices](#) on the system. If the required system device is not included in the list, it can be entered manually.

Default: no device selected (empty)

Queue sizes

Device-internal buffer size for receiving and sending data. If read errors occur when data are being received, it may be due to the fact that the data cannot be processed quickly enough which causes the device-internal buffer to overflow. In this case the input buffer size should be increased. The buffer size of the output is normally 0.

- **Input**

Enter device-internal buffer size for the Input.

Default: 128 KB

- **Output**


Enter device-internal buffer size for the Output.

Default: 0 KB

Serial Settings

Active

Enable to enter data for a serial connection (i.e. baud rate, number of data bits, parity, number of stop bits, and handshake method).

 If the system device is not a serial device, this setting must not be activated.

- **Baud**

Select a baud rate from the pull-down menu.

Available rates: 4800 - 115200

Default: empty (none)

- **Databits**

Select the number of databits from the pull-down menu.

Available numbers: 5 - 8

Default: 8

- **Parity**

Select the parity from the pull-down menu.

Available values: none, odd, even, mark, space

Default: none

- **Stopbits**

Select the number of stopbits from the pull-down menu.

Available values: 1, 1.5, 2


Default: 1

- **Communication**

Select the handshake method from the pull-down menu.

Available values: none, RTS/CTS, RTS/CTS Handshake, RTS/CTS Toggle, XON/XOFF, unknown printer

Default: RTS/CTS Handshake

 RTS/CTS handshake should be used wherever possible as the handshake method.

- **Communication = XON/XOFF**

If the XON/XOFF method is used, the XON and XOFF character should also be defined.

- XON-Limit

Enter the XON-Limit.

The XON limit defines the maximum number of bytes in the input buffer before the XON character is sent.

Default: 65535

- XOFF-Limit

Enter the XOFF-Limit.

The XOFF limit defines the maximum number of bytes in the input buffer before the XOFF character is sent.

Default: 65535

- XON-Character

Define the XON-Character.

Default: 17

- XOFF-Character

Define the XOFF-Character.

Default: 19


 see also [Settings](#)

6.2.18 TCP/IP

The selection of TCP/IP as input interface opens a dialog with the following tabs:

[TCP/IP](#)
[Settings](#)
[Sessions](#)
[Data](#)
[Data Type](#)
[Data Capturing](#)

6.2.18.1 Address (TCP/IP Input)

 Microsoft Winsock 2 must be installed on the [Mercury](#) server! This is the case with Windows NT 4.0 Service Pack 3 or higher and Windows 98. With Windows 95 an update available from Microsoft is necessary.


Bind to IP address

Define the local IP address of the network interface that is used to receive the data. If the computer has only one local address, the field may remain blank. For a computer with multiple network interfaces or multiple IP addresses, you can limit to a single IP address. Leaving it blank in such a case makes Mercury receive data from all IP addresses.


Default: empty (use all available IP addresses)

Port

Define TCP port on which the data is to be received. With TCP/IP connections using no protocol (direct socket) usually port 9100 is used, although selection of the port is free.

 If the incoming data was sent by a [Mercury Local TCP Output Port](#), be sure to activate the [Decompress and decrypt data automatically](#) option.

Default: 9100

 see also [Settings](#)

6.2.18.2 Protocol (TCP/IP Input)

- **LPD/LPR** - Line Printer Daemon/Line Printer Remote printer protocol
- **IPP** - Internet Printing Protocol
- **PPD/PPR** - AFP/IPDS printer protocol
- **PJL** - Printer Job Language for status readback

6.2.19 Twain

The selection of Twain as an input interface opens a dialog with the following tabs:

[Twain Input](#)
[Settings](#)
[Sessions](#)
[Data Type](#)
[Data Capturing](#)

6.2.19.1 Twain (Input Port)

Twain Source


Selects the Twain Source / scanner.

All scans in a single job

All Twain scans will be collected in a single job. If a is defined, the job will automatically splitted.

If disabled, every Twain scan job will result in a single [Mercury](#) job.

Default: disabled

 A Twain job may contain several pages if supported by the Twain source.

Show user interface of the Twain Source

The user interface of the selected Twain Source will be shown before the scan starts. This user interface is supplied by the manufacturer of the scanner and is not part of [Mercury](#). In this user interface various parameter can be defined. If it is supported by the scanner, the number

of pages to scan can also be defined.

Default: enabled

Maximum number of scan images in buffer

Memory buffer for scanned images. If the buffer is filled, scanning will be delayed until the images are printed.

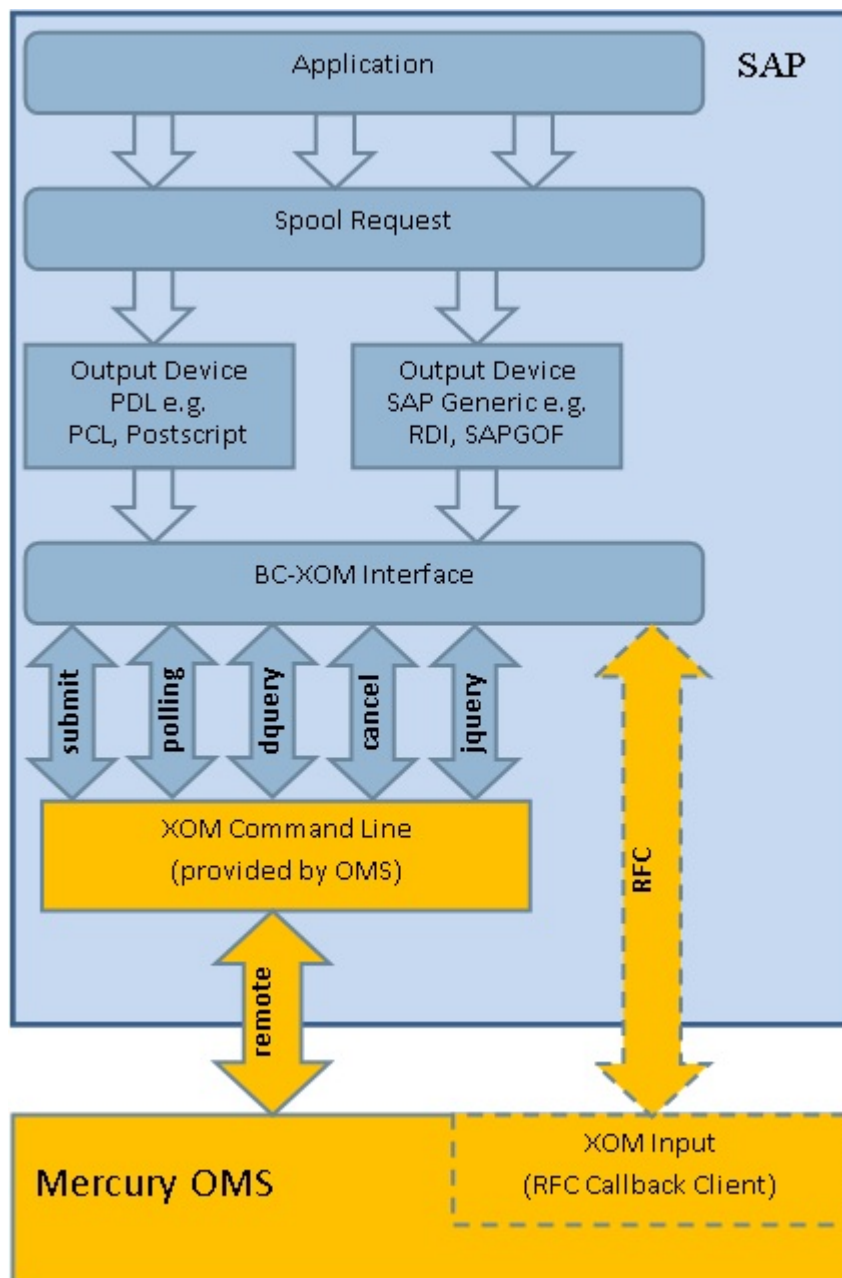
Default: 5

6.2.20 XOM

6.2.20.1 Overview

The SAP® BC-XOM interface (**B**usiness **C**onnecter-**E**xternal **O**utput **M**anagement) allows a tight coupling of the SAP® spool system to external output management systems (OMS). BC-XOM is intended for secure job delivery and reporting of job or device status via polling or RFC callback. RFC is a SAP® term for 'Remote Function Call'. For more information please refer also to [BC-XOM SAP integration scenario](#).

The Mercury XOM input complies the BC-XOM specification and allows a seamless integration with the SAP® ERP system.



The XOM interface provides following interactions :

- Submit (Job submit)
- Polling (Device polling or alternative RFC Callback Client)
- DQuery (Device query, optional)
- Cancel (Job cancel, optional)
- JQuery (Job query, optional)

SAP® and SAP NetWeaver® are registered trademarks of SAP AG in Germany and in several other countries.

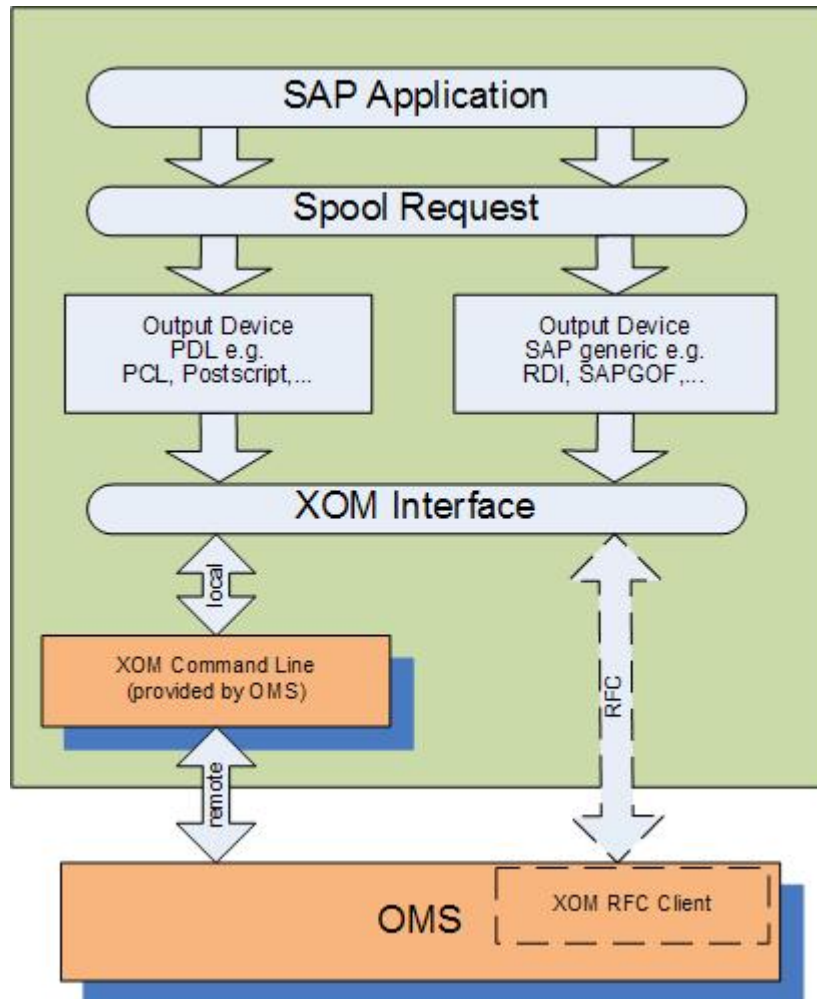
The XOM Command Line Component includes software developed at the Apache Software Foundation (<http://www.apache.org>).

6.2.20.2 Requirements

At the SAP backend**XOM Command Line Component**

On each SAP application server with a spool work process a 'XOM Command Line Component' must be installed (refer to [Installation Guidelines](#)). This component is responsible for job submit via TCP/IP and all other BC-XOM tasks.

⚠ The component needs a Java Runtime Environment (JRE) version ≥ 1.4 . It possibly communicates with the Mercury OMS via [XML-RPC interface](#).

**At the Mercury OMS system**

For the Mercury XOM input a proper license is needed.

RFC functionality

The RFC (Remote Function Call) functionality of XOM input depends on following SAP NetWeaver® RFC runtime libraries:

- sapnwrfc.dll
- libsapucum.dll

- libicudcnumber.dll
- icuuc34.dll
- icuin34.dll
- icudt34.dll

⚠ If the XOM input is used for **job submit** with **polling** only, these DLL's are not needed. Please refer also to [Connectivity->Callback RFC client->Activate](#).

If these DLL's not already exists at the system where **Mercury** is installed, they must be downloaded from SAP marketplace (e.g. package 'NWRFC_6-20004550_SDK711WinServerX86.SAR' for Windows® 32 bit systems) and installed within the system search path or the **Mercury** installation directory.

You need an account to login to [SAP marketplace](#).

Navigate through following menu structure:

- > Software Downloads
- > Search for software downloads
- > Within the search field enter 'SAP NW RFC SDK'
- > Select the most current 32 bit Windows version

⚠ Even if **Mercury** is installed on a Windows® 64 bit system, the SAP NetWeaver® RFC runtime libraries for Windows® 32 bit are needed.

XML-RPC interface

The XML-RPC interface (RPC = Remote Procedure Call) is needed for all XOM actions other than **job submit** and **RFC callback**.

⚠ If the XOM input is used for **job submit** with **callback** only, the XML-RPC interface is not needed (except if the callback client should start with the backend, refer to [ROMS creation](#)).

In order to use the **Mercury** XML-RPC interface refer to [Installation Guidelines](#) and [Configuration Guidelines](#).

⚠ The XML-RPC interface uses port 49111. This port must be opened within a possibly existing firewall at the OMS system.

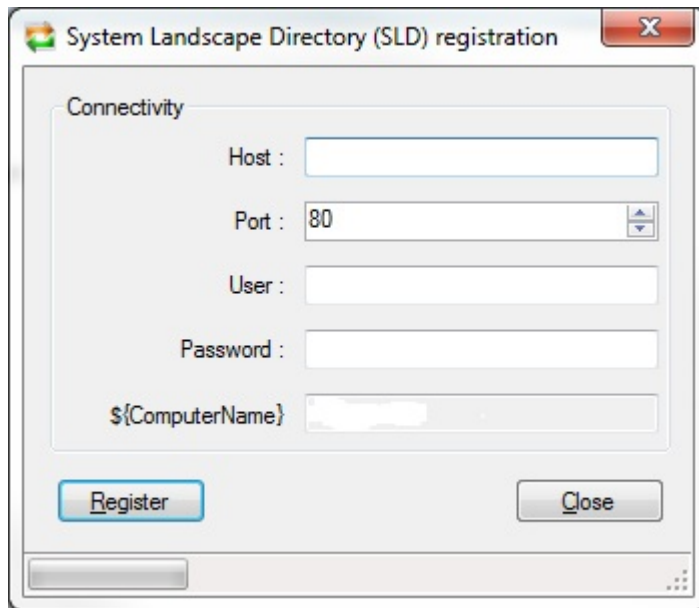
SAP System Landscape Directory (SLD) registration

The XOM input must be registered at the customer SLD. Please refer also to [System Landscape Directory](#).

⚠ The SLD registration process has to be done only once.

If registration has not been done already during product installation setup, invoke the SLD registration tool '**RegisterSLD.exe**' at the **Mercury** server system within the installation path.

⚠ The SLD registration tool needs a Microsoft® .NET Framework version >= 2.0.

**Host**

Enter the IP address or DNS name of the customers System Landscape Directory host.

Port

The TCP/IP port number of the host.

User

User name of the account.

⚠ The user must be assigned at least to the security role '*DataSupplierLD*' and '*LcrInstanceWriterLD*'.

Password

Enter a valid password for the given user.

 `${ComputerName}`

Is only for information. Contains the automatically detected computer name where the tool is running. This name is also submitted during registration.

Press the '*Register*' button. Notice the status line of the bottom if registering was successful.

Please refer also to [Installation Guidelines](#).

XOM Multilingual Message Support (optional, this is not required)

If you prefer to see the reported job and device status messages in the language of the user who is logged on to the SAP system with the SAP GUI, you should upload the message table translations to the SAP backend.

Currently only the languages English and German are available. Contact our support if you wish additional languages.

⚠ The message upload tool needs a Microsoft® .NET Framework version ≥ 2.0 . If not already exists, install the runtime at the [Mercury](#) system before invoking

'UploadXomMsg.exe'.

 The tool needs also the SAP .Net Connector V3.0 (NCo 3.0 for .NET framework 2.0 (32 bit)).

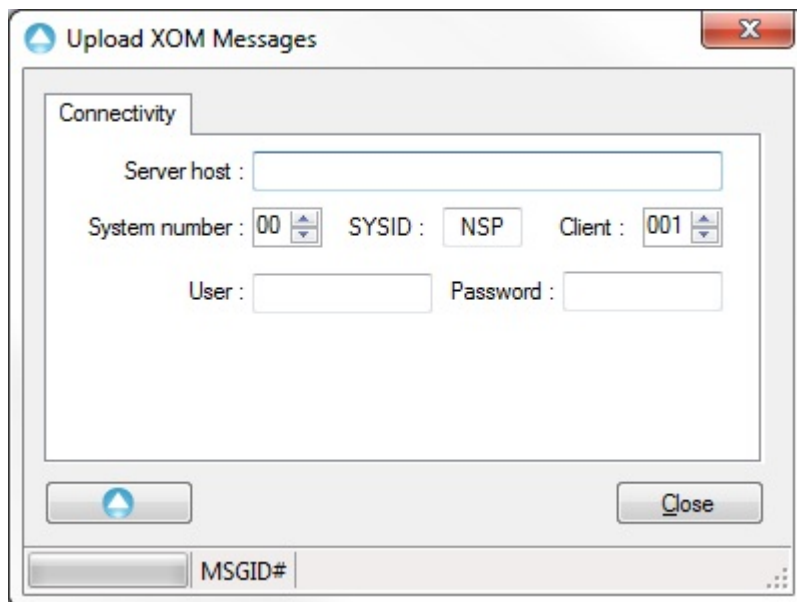
NCo 3.0 must be downloaded from SAP marketplace and installed within the system search path or the [Mercury](#) installation directory.

You need an account to login to [SAP marketplace](#).

Navigate through following menu structure:

- > Software Downloads
- > Support Packages and Patches
- > Browse our Download Catalog
- > SAP Connectors
- > SAP .NET Connector
- > SAP .NET Connector 3.0
- > Windows Server on IA32 32bit
- > Select the most current version for .NET 2.0 framework

To upload the message table, invoke the upload tool **'UploadXomMsg.exe'** at the [Mercury](#) server system within the installation path.

**Host**

Enter the IP address or DNS name of the SAP backend.

System number

The backend SYSNR (SAP System Number, e.g. '00').

System ID

The backend SYSID or R3NAME (System ID, e.g. 'NSP').

Client

The CLIENT number (e.g. '000').

User

User name of the account.

Password

Enter a valid password for the given user.

After successfully uploading the message table, the status bar will show the number of uploaded message ID's (MSGID).

6.2.20.3 Installation

6.2.20.3.1 Installation Guidelines

When the Mercury XOM input is used to couple it with a SAP system, then the 'Register to SAP System Landscape Directory (SLD)' option should be activated during the product installation setup. If this has not been done during setup, it can be performed later by invoking the SLD registration tool at the Mercury system (see below).

In order to use the XOM input, there are at least following additional installation steps necessary:

1. Install the [XOM Command Line Component](#) at each involved SAP backend with a spool work process.
(if not already exists, install a Java Runtime Environment (JRE) version ≥ 1.4)
2. Optional: Install the Mercury 'HTTP and Database server' component at the Mercury system (possibly needed for the [XML-RPC interface](#), see below).
3. Optional: Install the SAP NetWeaver® RFC runtime libraries at the Mercury system (possibly needed for the [RFC functionality](#), see below)
4. Perform SLD registration (only once).
(if not already exists, install a Microsoft® .NET Framework Runtime Version ≥ 2.0 at the Mercury system before invoking '**RegisterSLD.exe**' within the Mercury installation path)

XOM Command Line Component installation

The Xom Command Line Component files resides within the directory '**<installation media>\Support\XomCmdLine**'.

⚠ The procedure is similar for Windows or UNIX/Linux like systems, except that Windows systems with no POSIX compatible subsystem usually using the backslash '\' instead slash '/' as path separator.

To install the component files, all files within the directory '**<installation media>\Support\XomCmdLine**' must be copied to a **writable** directory at the SAP backend (recommended destination directory is /usr/xom). The tool saves per default its configuration in the same directory where it is started, therefore it needs write access rights. For additional information please refer also to [Configure XOM Command Line Component](#).

⚠ On UNIX/Linux like systems you may have to set explicitly the execution flag of the script file **dfxom**, e.g. ...:> **chmod +x dfxom**. Also you may have to set the proper user and group of the copied files (e.g. with **chown**).

HTTP and Database server installation

This optional component provides also the [XML-RPC interface](#) and is usually not installed if you did not select "full installation" during the standard product installation setup.

If not already installed, just invoke the '**<installation media>\Webserver\HttpDbSetup.exe**' and follow the instructions.

⚠ It is needed if XOM input is used with polling.

SAP NetWeaver® RFC runtime library installation

This optional component provides the [RFC functionality](#).

⚠ It is needed if XOM input is used with callback.


Perform SLD registration

Please refer to [SAP System Landscape Directory \(SLD\) registration](#).

6.2.20.4 Configuration

6.2.20.4.1 Configuration Guidelines

In order to use the Mercury XOM input properly, there are some configuration rules and steps necessary, which are described here. First all configuration should be done at the SAP backend, than the corresponding configuration should be done at the Mercury system.

 Generally callback should be preferred in comparison with polling.


It is assumed that the reader of this short description is already familiar with the SAP GUI and the Mercury system.

At the SAP backend


Create at least one ROMS (Real Output Management System), one or more corresponding LOMS (Logical Output Management System) and one or more output devices, i.e. printers (transaction **SPAD**). Usually only one ROMS is necessary, but depending on performance requirements or load balancing of the system, it could make sense to create more than one ROMS. It is recommended to use logical spool servers instead of physical spool servers at the ROMS/LOMS/Printer definitions, because this facilitates the maintaining of the definitions if they must be transferred to multiple SAP systems.

If you want to keep the definitions at one central SAP system, you might decide to create a PAL system (for more information refer to [Printing Assistant for Landscapes](#)). A PAL system benefits the administration if there are many distributed SAP system instances, or a lot of output devices are used.

- The LOMS must be configured with the correct commands or command groups (see step by step example below).
- If multiple different callback targets are configured at the LOMS definitions, the targets must be grouped (transaction **SMLG**)
- The output device must have access method 'E: External output management system'.
- The XOM command line component must be configured (please refer to [Configure XOM Command Line Component](#)).
- In case of RFC callback a proper user account for callback should be created.

 The account needs at least authorization profile 'S_XMI_XOM_A'. When using BC-XOM Extension for Printer Administration, also 'S_ADMI_FCD'.

If your SAP backend has a XOM 0.2 interface (SAP NetWeaver Version \geq 7.02 (SAP Kernel 7.20)) that supports the 'BC-XOM Extension for Printer Administration', the XOM input is able to create the ROMS/LOMS/Printer definitions automatically at the SAP system where it logon. Please refer to [Definitions](#).

 Please keep following in mind to avoid collisions of BC-XOM and PAL managed printers. The following restrictions and rules apply:

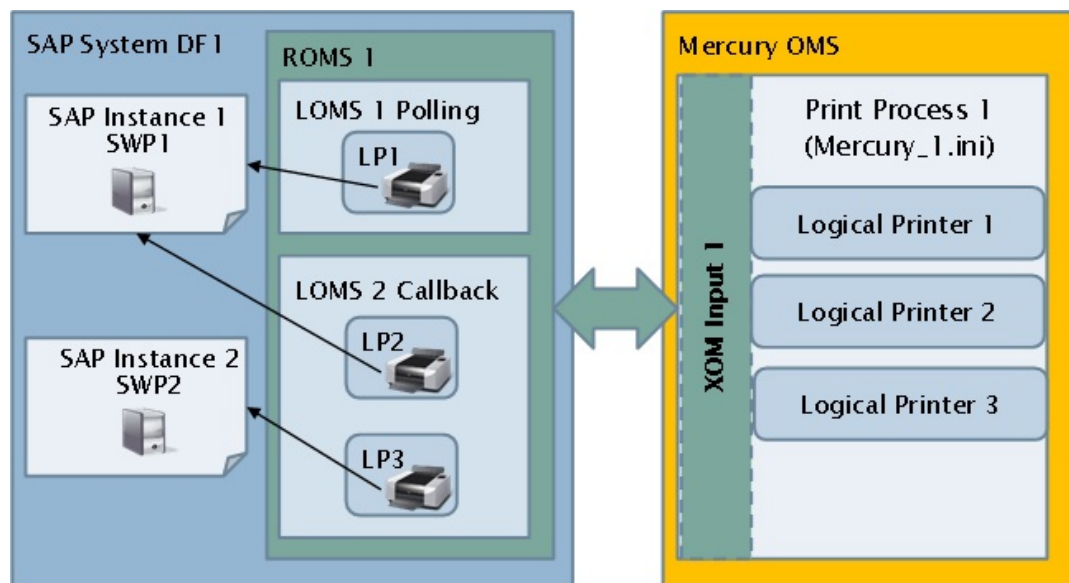
- PAL printers cannot be modified on their target systems via BC-XOM.
- PAL printers can be modified on their central systems via BC-XOM, but deletion is not possible.
- At printer distribution time PAL will not overwrite BC-XOM created printers on a target system.
- BX-XOM created printers cannot be put under PAL control.

At the Mercury OMS system

There are some settings which are very important for correctly using the XOM input.

- For each individual ROMS a separate print process with corresponding XOM input is needed.
- The [XML-RPC interface](#) must be enabled, if needed.
- At the XOM input at least Use job information within data stream must be activated.
- At each Output Interface, coupled to XOM input, Spooling and Job Monitoring and Keep spooled jobs after printing must be activated.
- For each printer added within the SAP backend, a Logical Printer must be added.
- Each SAP system with separate System ID (e.g. 'DF1') needs a separate print process with XOM input.
- If there are multiple different callback targets, XOM input must be configured to use 'Group logon'.
- If multiple XOM input are used on one [Mercury](#) server, each instance needs a other TCP/IP port (default 9100) for job data receive.

⚠ Otherwise no correct job status can be reported to the SAP backend (neither via polling nor via RFC callback).



Picture shows the assigning of SAP output devices (printers) to [Mercury](#) logical printers (NOTE: SWP means Spool Work Process)

A step by step example

Please note that the screenshots itself and the content may vary from system to system depending on the version of the SAP GUI, the version of the [Mercury](#) system and the system environment.

The example configures a command line ROMS with two LOMS (one for polling and one for callback) and two printers assigned to the LOMS.

⚠ The procedure is similar for Windows or UNIX/Linux like systems, except that Windows systems with no POSIX compatible subsystem usually using the backslash '\' instead slash '/' as path separator.

At the SAP backend

ROMS creation

Transaction: **SPAD** (select Extended administration->Output management systems)
Create a ROMS entry with a not already used name, here **DFCMDLINE1**.

The screenshot shows the SAP configuration interface for 'Spool Administration: Real Output Management System'. The 'Real OMS' field is highlighted with a red circle and contains the text 'DFCMDLINE1'. Below it, the 'Description' field is filled with 'docuFORM ROMS command line polling and callback'. The 'OMS Attributes' section is divided into four sub-sections: 'Tasking' (with 'Command line' checked and 'RFC server' unchecked), 'Job Status' (with 'Query', 'Polling', and 'Callback' checked, and 'Deletable' unchecked), 'Device Status' (with 'Queue query' and 'Callback' unchecked), and 'Output Types' (with 'Fax' unchecked). Below these are fields for 'Host Spool Attribute Record' and 'OMS Configuration' (containing 'PP='). The 'SAP Configuration' section at the bottom includes 'Initialization Instance' (virtualxp_NSP_00), 'Initialization Command' (/usr/xom/dfxom -start "&Es" "&Er" "&E3" "&E4"), and 'Reconfiguration Request' (300 Sec) with a checked 'Reconfiguration Required' checkbox.

The 'Initialization Command' must be: **<installation path>/dfxom -start "&Es" "&Er" "&E3" "&E4"**

When the [XOM Command Line Component](#) has not been installed into the recommended destination directory (/usr/xom), adapt the path.

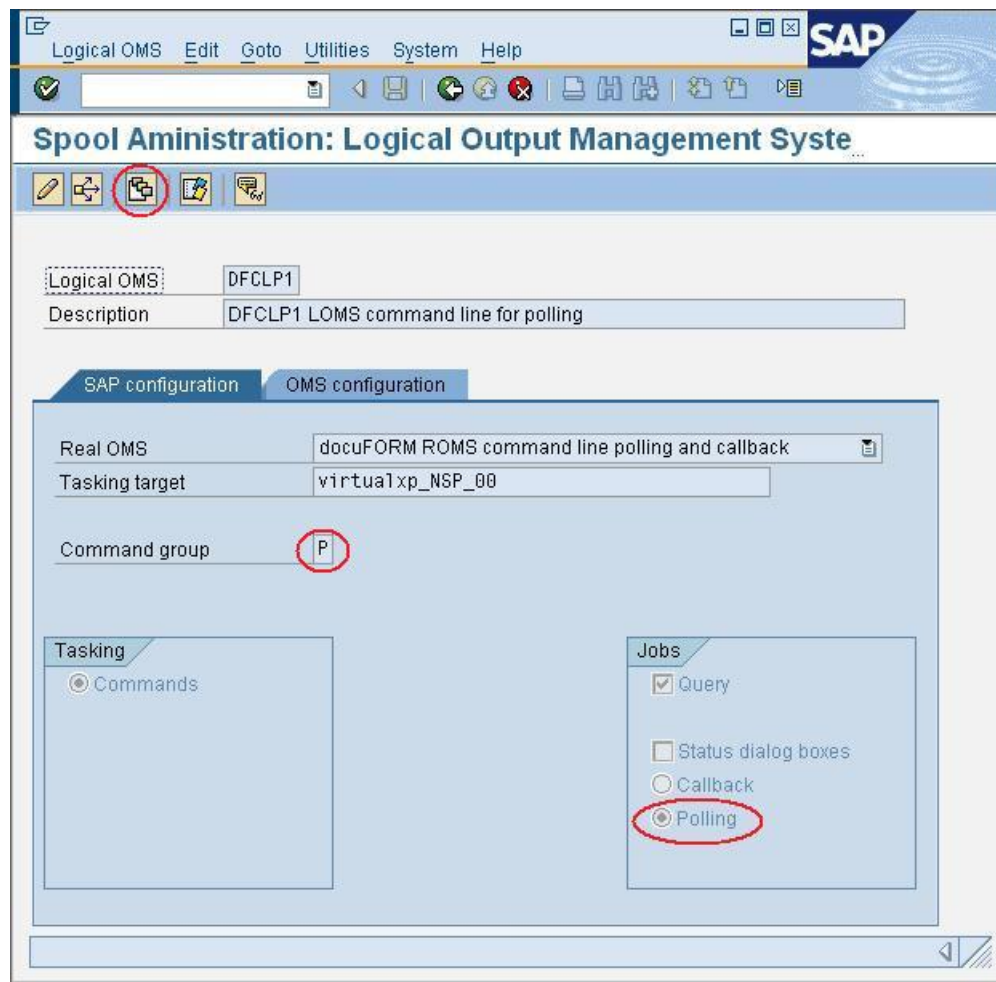
Actually the 'Initialization Command' is needed only, when a callback client at the [Mercury](#) system should start with the backend. In that case also the 'OMS Configuration' field must be set with a corresponding print process entry **PP=<print process name>** (e.g. PP=Mercury_1.INI, or PP=DSPPATH2.INI, etc. depending on the print process where a XOM input is used).

Please refer also to [The SAP parameter meanings](#).

⚠ The print process entry (PP=...) must be also set for polling or other optional XOM actions, e.g. 'Job query', 'Job cancel', 'Device query'.

LOMS creation for polling

Transaction: **SPAD** (select Extended administration->Output management systems)
Create a LOMS entry with a not already used name, here **DFCLP1**.



Add following OMS commands to the LOMS or better create a command group, for example 'P':

Operating systems Edit Goto Utilities System Help

SAP

Spool Administration: Operating System Command

Real OMS: DFCMDLINE1 docuFORM ROMS command line polling and callback
 Command group P
 Operating system Windows NT

Command path
 Path /usr/xom/dfxom

OMS commands

Submit	-submit &EI "&P" &EG "&F" "&T" "&o" &C "&ES"
Polling	-dpoll "&P" &EG "&E4" &EL
Queue query	-dquery "&P" "&E4" &Es
Job cancel	-cancel "&E4" &EL
Job query	-jquery "&E4" &EL

The OMS command sequences must be:

Command path	Sequence
Path	<installation path>/dfxom
Command	Sequence
Submit	-submit &EI "&P" &EG "&F" "&T" "&o" &C "&ES"
Polling	-dpoll "&P" &EG "&E4" &EL
Queue query	-dquery "&P" "&E4" &Es
Job cancel	-cancel "&E4" &EL
Job query	-jquery "&E4" &EL

⚠ A space ' ' must be inserted before each OMS command sequence (e.g. <space>-submit ...), because here the executable name **dfxom** has been added to the command path in order to avoid repeating it for each command. The double quotes must be used as shown. Some commands are not needed depending on the supported and activated features at the ROMS/LOMS, but it is good practice to provide them all.

The SAP parameter meaning is as follows:

Parameter name	Meaning
&EI	SAP spool ID
&P	Destination
&EG	Reply Message Group (RMG)
&F	Print job full file name
&T	SAP job title
&o	SAP name of the user who created the output request
&C	Number of copies
&ES	SAP instance, <hostname>_<system ID>_<system #>
&E3	ROMS R/3 flags
&E4	ROMS configuration parameter
&EL	List of OMS job ID's
&Er	ROMS name
&Es	SAP System ID

LOMS creation for callback

Transaction: **SPAD** (select Extended administration->Output management systems)
Create a LOMS entry with a not already used name, here **DFCLC1**.

Assign the previously created command group '**P**' or create a other one with the same commands.

Printer creation (assigned to polling LOMS)

Transaction: **SPAD** (select Devices / Servers)

Create a output device with proper device type and a not already used name, here **DFP1**.

The screenshot shows the SAP Spool Administration interface for an Output Device. The window title is "Spool Administration: Output Device (Display)". The main content area is divided into tabs: "DeviceAttributes", "Access Method", "Output Attributes", and "Tray Info". The "DeviceAttributes" tab is active, showing the following fields:

Output Device:	DFP1 Printer3 cmd line polling	Shortname	DFP1
Device Type	HP4100 : HP LaserJet 4100 R4.x+		
Spool Server	virtualxp MSP_00		
Server Description			
Host	virtualxp		
Device Class	Standard printer		
Authorization Group			
Model			
Location			
Message			
<input type="checkbox"/> Lock Printer in SAP System			

Change the 'Access Method' to 'E: External output management system' and assign the 'Logical OMS' to the previously created LOMS for polling. The host printer name (here **Printer3**) will be used later in the Mercury system as logical printer at the property Condition of usage->**Destination port/queue**.

The screenshot shows the SAP Spool Administration interface for configuring an output device. The window title is "Spool Administration: Output Device (Display)". The "Output Device" field is set to "DFP1 Printer3 cmd line polling" and the "Short name" is "DFP1". The "Access Method" tab is selected, showing the following configuration:

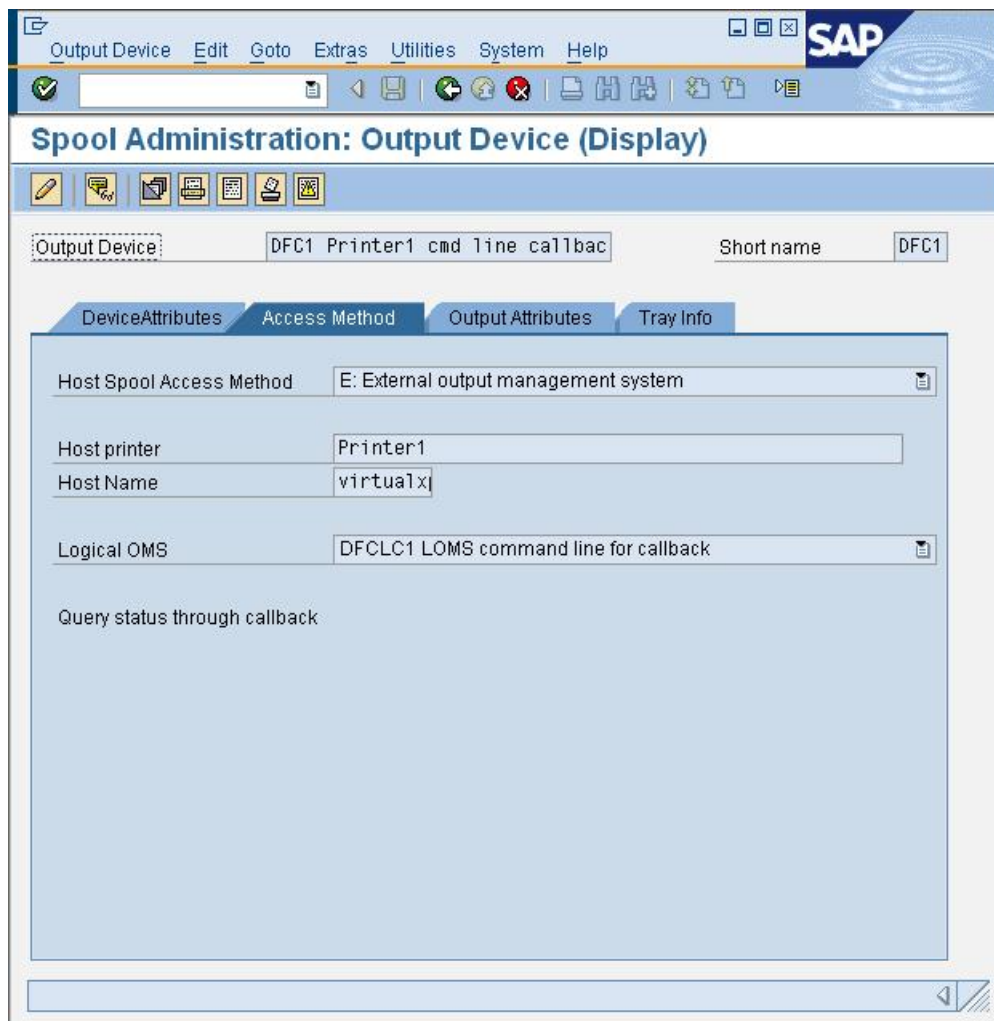
Field	Value
Host Spool Access Method	E: External output management system
Host printer	Printer3
Host Name	virtualxj
Logical OMS	DFCLP1 LOMS command line for polling

There is a checkbox labeled "Query status through polling" with the option "Do Not Query Host Spooler for Output Status" selected.

Printer creation (assigned to callback LOMS)

Transaction: **SPAD** (select Devices / Servers)

The steps are similar as above, but the output device must be assigned to the previously created LOMS for callback and a other 'Host printer' identifier (here **Printer1**) must be used.



User account creation for callback

Transaction: **SU01**

Creation of a user account is out of scope of this short description. In case of an SAP NetWeaver Trial Test System the predefined user **BCUSER** can be used for callback.

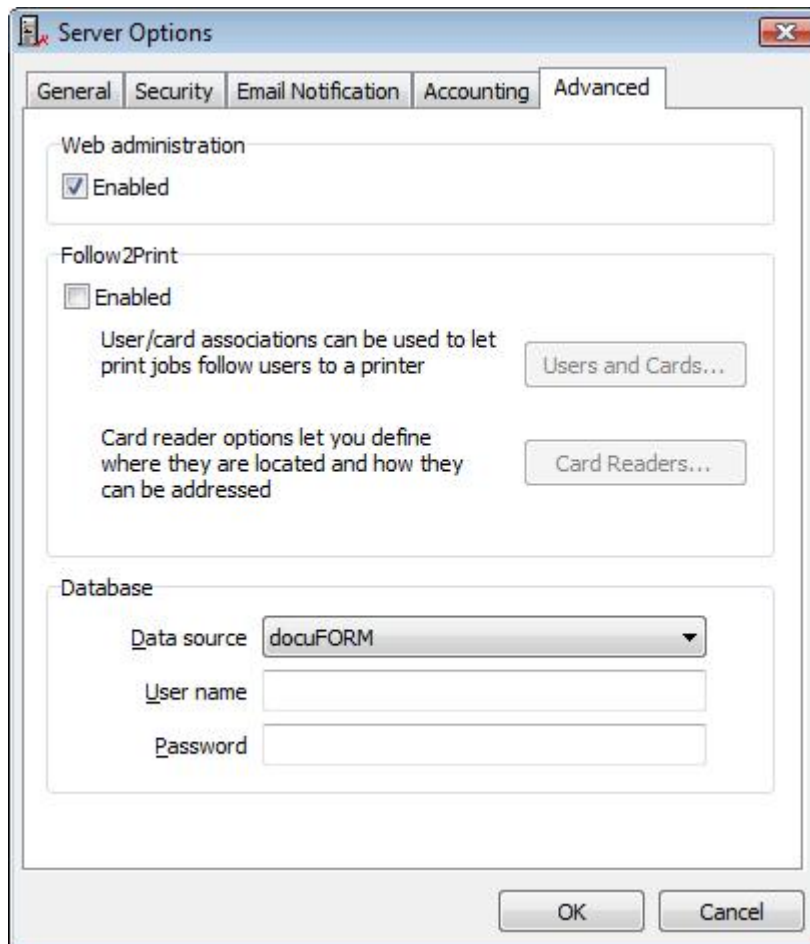
⚠ The account needs at least authorization profile 'S_XMI_XOM_A'. When using BC-XOM Extension for Printer Administration, also 'S_ADMI_FCD'.

At the Mercury OMS system

Enabling the **XML-RPC interface**

Menu: **Setup->Server->Advanced**

Enabling the interface is done by enabling 'Web administration'

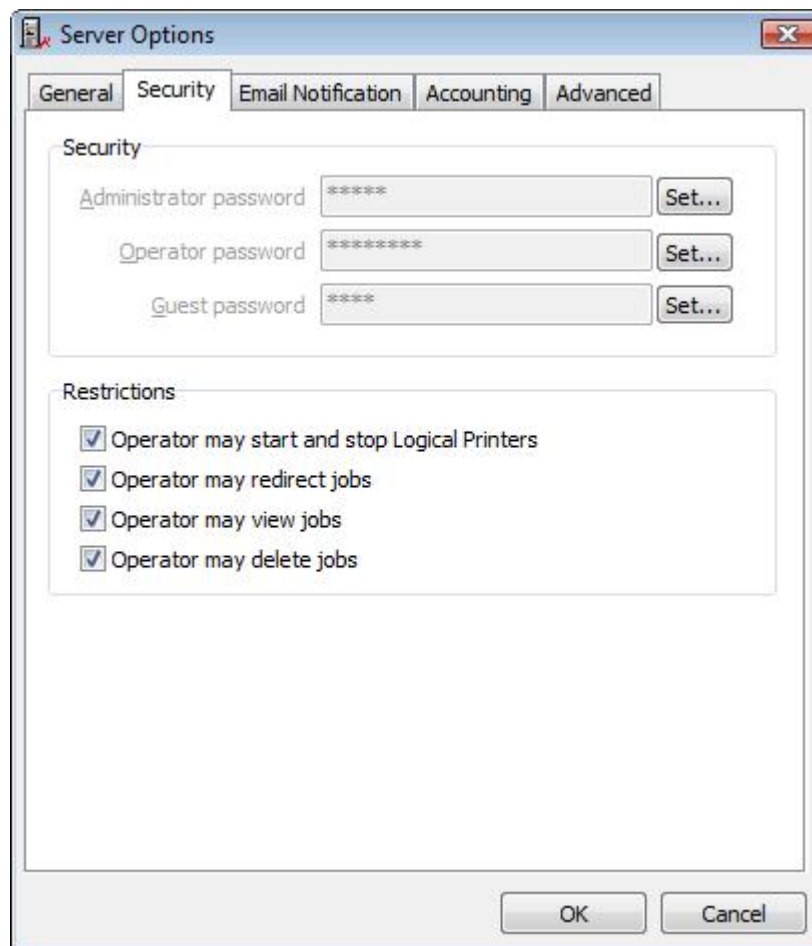


The screenshot shows the 'Server Options' dialog box with the 'Advanced' tab selected. The dialog has a title bar with a close button. Below the title bar are five tabs: 'General', 'Security', 'Email Notification', 'Accounting', and 'Advanced'. The 'Advanced' tab contains three sections:

- Web administration:** A checkbox labeled 'Enabled' is checked.
- Follow2Print:** A checkbox labeled 'Enabled' is unchecked. Below it are two buttons: 'Users and Cards...' and 'Card Readers...'. Text below the buttons explains that user/card associations can be used for printer jobs and that card reader options can be defined.
- Database:** A dropdown menu for 'Data source' is set to 'docuFORM'. Below it are two empty text input fields for 'User name' and 'Password'.

At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Set a password for the account which is intended to be used by the [XOM Command Line Component](#) and set the proper rights.

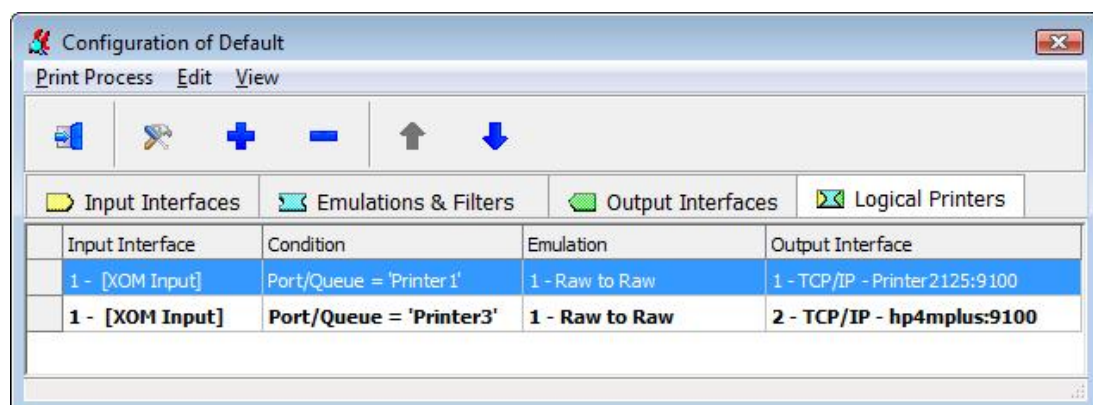


The corresponding account and password must be also configured at the SAP backend. Please refer to [Configure XOM Command Line Component](#).

Print process creation

Add a print process with a XOM input, some proper data processing and output interface (keep in mind that Spooling and Job Monitoring and Keep spooled jobs after printing must be activated at every output interface).

For each output device of the SAP backend (in this example **Printer1** and **Printer3**) add a Logical Printer and set the property Condition of usage->**Destination port/queue** to the corresponding printer names. The print process should be assigned to the ROMS 'OMS Configuration' field, refer to [ROMS creation](#).



The configured port within XOM input must be also configured at the SAP backend. Please refer to [Configure XOM Command Line Component](#).

6.2.20.4.2 XOM input

Configure port

Opens the port configuration with additional options.

⚠ At least Use job information within data stream must be activated.

Connectivity

Defines the parameter needed to connect to the SAP backend.

Callback RFC client

Defines all account parameter for the RFC client to logon. These parameter are similar to the SAP GUI logon parameter.

⚠ The account needs at least authorization profile 'S_XMI_XOM_A'. When using BC-XOM Extension for Printer Administration, also 'S_ADMI_FCD'.

Activate

Activates the RFC client.

⚠ This switch should be only deactivated, if the XOM input is used for polling only, i.e. there are no configured LOMS with RFC callback at the backend.

Connection type

Either '*Group logon*' (load balancing) or a '*Specific application server*' can be used.

⚠ If multiple different callback targets are configured at the SAP backend LOMS definitions, '*Group logon*' must be selected.

Server host

The IP address or DNS name of the backend.

⚠ In case of '*Group logon*' this specifies the MSHOST (Message Server Host).

⚠ In case of '*Specific application server*' this specifies the ASHOST (Application Server Host).

System number

The backend SYSNR (SAP System Number, e.g. '00').

System ID

The backend SYSID or R3NAME (System ID, e.g. 'NSP').

Client

The CLIENT number (e.g. '000').

User

The account USER name for RFC callback.

Password

The account password (PASSWD) for given user name.

SAProuter string

A route string is only necessary, if the backend is only accessible over the internet behind one or more firewalls. Please refer also to [Route String Entry for SAProuter](#).

⚠ There must be an SAProuter installed at the system.

The following fields must be given in case of connection type = 'Group logon' (load balancing).

Logon group

The logon GROUP name. Group names can be defined and assigned at the backend with transaction **SMLG**.

MSSERV

The TCP/IP port or service name of the 'Message Server Service'. This is not needed in most cases.

If the message server does not listen on the standard service name (usually 'sapms<SYSID>', e.g. 'sapmsNSP'), e.g. as specified in the '<systemdrive>:\WINDOWS\system32\drivers\etc\services' file, this parameter must be given.

⚠ However, if the entry in the 'services' file is different or does not exist, you must specify the network port directly (e.g. 3900)

Defintions

Defines the callback client parameter needed to get the ROMS/LOMS/Printer settings from the SAP backend.

Double-click onto an entry to change the name or click onto the browse button to see more details.

The details are only from relevance if the SAP backend has at least a XOM 0.2 interface (SAP NetWeaver Version >= 7.02 (SAP Kernel 7.20)) that supports the 'BC-XOM Extension for Printer Administration'. In that case the RFC client is able to create the given ROMS/LOMS/Printer settings automatically at the SAP system where it logon.

The definitions will be created under following conditions :

1. The SAP backend has a XOM interface version greater or equal to 0.2
2. The logon user has enough access permissions.
3. At the time when the client logon, there exists not already a corresponding ROMS/LOMS/Printer definition within the SAP backend.

⚠ The account needs at least authorization profile 'S_XMI_XOM_A'. When using BC-XOM Extension for Printer Administration, also 'S_ADMI_FCD'.

⚠ Per default all spool server fields are predefined with 'DF_RFC_XOM'. You may either create a logical spool server (transaction **SPAD**) with that name at the SAP backend before the client connects to it, or change the fields with the real spool server names of your system. The name of real spool servers is usually made up of the <host name>_<system ID>_<system number>

ROMS

Enter the ROMS name for which the RFC client is responsible. It must be created first within the backend. Please refer to [ROMS creation](#).

⚠ This is the only parameter needed, when the ROMS/LOMS/Printer settings has been made manually at the SAP backend.

The following parameter are only for 'BC-XOM Extension for Printer Administration', i.e. when

the ROMS should be created automatically at the backend. Only necessary fields are requested, all other fields are set automatically with rational values.

Descriptions

A meaningful description for the definition.

Configuration

Just for information. Will be automatically detected and fills the 'OMS Configuration' field entry of the ROMS.

Initialization instance

Enter the name of the SAP spool server that is to output the command to start the callback client.

Reconfiguration request

Time interval for a reconnect. If a callback target is unavailable, the client attempts to reconnect to this SAP instance cyclically. This value represents the duration of the cycle (in seconds) for reattempting a connection. This value is valid for all logical output management systems in a real output management system.

Full path to command line

Enter the full path name where the [XOM Command Line Component](#) has been installed at the SAP backend (default /usr/xom/dfxom). The client will create a correct 'Initialization Command' field entry for the ROMS.

LOMS

Only for 'BC-XOM Extension for Printer Administration'.
Enter the LOMS name which should be created at the backend.

Descriptions

A meaningful description for the definition.

SAP configuration**Tasking target**

This spool server is used to output anonymous commands, that is, commands that are not matched to a specific server based on the device definition. Anonymous commands are all output management system (OMS) commands except SUBMIT and POLLING.

Target for callback

Enter the name of the SAP spool server that receives the status callback.

Event report level

Report level for events of output management systems. The value describes the level up to which events are reported to the SAP system (01 'Final messages' ... 05 'Also information')

OMS configuration**Send period**

Time interval when job or device status is send to the SAP backend.

Number of events

Maximum number of collected job or device events. If the number of events is reached, they will be sent even the time interval is not expired.

Printers

Only for 'BC-XOM Extension for Printer Administration'.
Enter the Printer short name which should be created at the backend.

Output device

The long name of the device.

Device attributes

Device type

Name of a existing device type. Consult the SAP system for available device types. Per default a HP LaserJet 4 family PCL5 device type is used.

Spool server

Name of the spool server. This name is usually made up of the <host name>_<system ID>_<system number>.

Access method

Host spool access method

Just for information. Must be always E: External output management system.

Host printer

The name under which a Mercury logical printer can be dispatched via Condition of usage->**Destination port/queue**.

Logical OMS

The assigned LOMS. Logical output management systems allow you to operate the devices of a real output management system in different modes. For example, you can operate a group of important devices in callback mode, and at the same time operate other devices in the conventional polling mode.

6.2.20.4.3 XOM Command Line Component

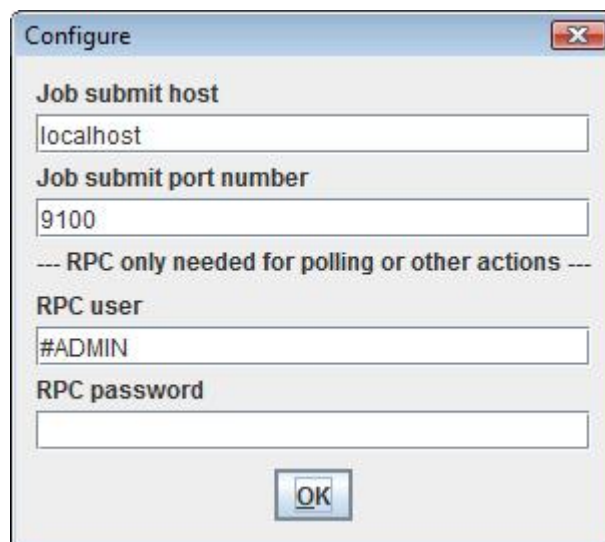
For additional information please refer also to [XOM Command Line Component](#).

⚠ The procedure is similar for Windows or UNIX/Linux like systems, except that Windows systems with no POSIX compatible subsystem usually using the backslash '\' instead slash '/' as path separator.

To configure the component at the SAP backend, open a command shell within the install directory (recommended is /usr/xom) and enter:

```
...:> dfxom -b
```

There should popup a small dialog:



After pressing OK, a configuration file 'CmdLineXom.properties' will be saved within the same directory.

- ⚠ The command shell should have the same access rights as the SAP system which executes `dfxom`. There are **write permissions** needed for saving the configuration. Even if logging is switched off, a log file '`dfxom.log`' will be created (with size of 0 Bytes).

There are additional command line options available, for more information enter **`dfxom --help`**

For instance you might enter **`dfxom -t`** to test the configuration. If this is successfully, there should be shown following at the console output :

```
*** Start of test query. ***
{state=stopped,  description=Default,  processname=Mercury_1.INI}
*** End of test query. ***
```

The output depends on the configured print processes at the [Mercury](#) server.

Job submit host

Enter the IP address or DNS name of the system where [Mercury](#) OMS has been installed.

Job submit port number

Enter the port number of the corresponding [Mercury](#) XOM input (configured with button `Configure port`).

- ⚠ This port must be opened within a possibly existing firewall at the OMS system.

The above 2 settings are used to submit the job to the OMS.

The following 2 RPC (Remote Procedure Calls) user account parameter are used to communicate with the OMS via [XML-RPC interface](#).

- ⚠ Refer to [XML-RPC interface](#) if the RPC account parameter are really needed.

RPC user

Enter the user name you have configured at the OMS system (either `#ADMIN` or `#OPERATOR`).

RPC password

Enter the password you have configured for the user at the OMS system.

- ⚠ When you leave this field empty, the current password (which might be empty or has been not set at the [Mercury](#) system) is not changed.

Logging

Per default logging is disabled for the [XOM Command Line Component](#). For troubleshooting or other reasons, logging can be enabled by performing following steps:

1. Open a command shell within the XOM Command Line Component install directory at the SAP backend.
2. Within the script file `dfxom.bat` (on UNIX/Linux systems `dfxom`) comment out the line containing '`java -jar cmdlinexom.jar ...`' and activate the line containing '`java -Dlog4j.configuration=log.properties -jar cmdlinexom.jar ...`').
3. Within the logging template file '`log.properties`' modify the line '`log4j.rootLogger=<logging level>, ...`', and replace `<logging level>` with the wished level (ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL).

If you have not changed anything else within the logging template file, all logging goes to the file `dfxom.log` within the install directory. The file is wrapped after each 100 KBytes and 1 backup file is kept.

For more information about logging features, please refer to <http://logging.apache.org>.

Here is a example of the logging template file 'log.properties' :

```
# root logger level and appender
# (OFF, ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL)
log4j.rootLogger=ALL, FILE
# print only messages of level WARN or above in the following package
#log4j.logger.de.docuform=WARN

# appender FILE
log4j.appender.FILE=org.apache.log4j.RollingFileAppender
log4j.appender.FILE.File=dfxom.log
log4j.appender.FILE.MaxFileSize=100KB
# Keep one backup file
log4j.appender.FILE.MaxBackupIndex=1
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.ConversionPattern=%d [%t] %-5p %c -> %m%n
```

6.3 Emulations

6.3.1 Introduction

Mercury offers the following emulations for the conversion of print data:

- Barcode Label Software
- Chained Emulations
- Channel Emulation
- Comparex Line Mode
- Forms Overlay Language
- IGP to PCL
- IPDS to PCL5 (Color)
- IPDS to PDF (Color)
- Job Controller
- PCL to PDF
- PCL5 to X-Metacode (Color)
- Prescribe to PCL
- Program Interface
- Raw to Raw
- XES to PCL5
- X-Metacode to PCL5 (Color)

Filters can be placed before or after each emulation (input filters, output filters). This allows incompatibilities between the host and destination printing system or incorrectly formatted commands of character strings before and after the conversion to be compensated by the emulation.

6.3.2 Performance

Data Processing

- **Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this

value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32 KB

- **Block size for output buffering**

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32 KB

Timing

- **Delay before next try on read error**

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read attempts.

Default: 1000 msec

- **Delay before next try on write error**

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.


This prevents a slow output port slowing down the system by frequently repeated write attempts.

Default: 10 msec

- **Timeout for shutdown**

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.

 Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

Bidirectional Communication

- **Produce ACK/NAK responses**

Enable to produce ACK/NAK responses.

When processing print jobs, which are received through a Channel Input Port and when processing IPDS print jobs, usually it is necessary to send feedback responses to the host system. In other situations these responses might not be applicable. For highest flexibility it can be controlled by this option whether feedback responses are sent to the host system or not.

Default: enabled


- **Send stop code to host after writing <n> sec**

Define when stop code is sent to host after writing (in seconds).

This applies if a write access to the printer is not successfully completed within this time. Some hosts require responses within a specific time. If for some reason the printer goes offline or is turned off this might lead to problems with bi-directional communication.

Set to 0 to turn off this option.

Default: 0 (off)

 The effective time depends on the write timeout set at the output port. I.e. if at the output port a write timeout is set to 10 sec and here 15 sec are used, the second timeout after 20 sec will send the stop code. This is because the first timeout after 10 sec is still in valid range of 15 sec.

6.3.3 Resources

Settings regarding the resource files to be loaded into the printer can be made here.

Resource download for the IPDS to PCL5 emulation
Resource download for the PCL5 to PDF emulation
Resource download for the PCL5 to X-Metacode emulation
Resource download for the XES to PCL5 emulation
Resource download for the X-Metacode to PCL5 emulation

6.3.4 Barcode Label Software

6.3.4.1 Overview

For documentation of Mercury Barcode Label Software please see the Barcode Label Software Manual which is a separate document.

6.3.5 Chained Emulations

6.3.5.1 Overview

Under certain circumstances it might be helpful to pass multiple emulations. For example you want to translate IPDS to PDF and there is no emulation available for. In this case you would first translate IPDS to PCL5 and after that PCL5 to PDF which requires two emulations.

To solve this case you have to define and configure the two emulations first. By definition of a 3rd emulation of type Chained Emulations they will be chained.

Only already defined and configured emulations are available in the pull-down menu.

First emulation

Select an already defined emulation from the pull-down menu which should be passed first.

Second emulation

Select an already defined emulation from the pull-down menu which should be passed last.

6.3.6 Channel Emulation

6.3.6.1 Overview

This section describes all the parameters of the Mercury Channel Emulation.

The configuration dialog for the Channel Emulation has the following property pages (tabs):

- [Printer Type](#)
- [Translation Table](#)
- [Control Codes](#)
- [Bold Printing](#)
- [Download](#)
- [FCB](#)
- [Job End Detection](#)
- [Performance](#)
- [Band ID](#)

6.3.6.2 Parameters

6.3.6.2.1 Printer Type

Channel Input Port Assignment**• Use Channel Input**

Select the appropriate Channel Input Device from the pull-down menu.

Selection is possible only if a channel device is physically attached to the Mercury server.

• Printer Emulation

Shows the type of the emulated channel printer. This option is set with the corresponding Channel Input Device.

Selection is possible only if a channel device is physically attached to the Mercury server.

• Ignore UCS

Enable to ignore UCS.

Only available if printer type SNI 3338/39 is selected in the Printer Emulation field. It allows to setup Mercury to ignore the parity of the printer's Universal Character Set Buffer.

Output**• Type**

Select an output type from the pull-down menu.

It allows to select the type of printer which is really connected to Mercury. The type of output

also determines the available tabs in the Channel Emulation Properties dialog.

Available options are:

- Centralized Channel 3203-5
- Centralized Channel 3211
- Decentralized with CR/LF, FF
- Decentralized with PC-Code

Default: Decentralized with CR/LF, FF

Centralized Channel 3203-05 specifies a centralized Printer IBM 3203-05 which is equipped with a channel interface. This output type is only designed for use when Mercury is connected to a Siemens mainframe with operating system BS2000 and the converter emulates either the printers SNI 3338/39 or SNI 3348.

When selected, a check box **Data Check** is available.
Enable Data Check to verify data for each code is printable.

Default: disabled

When selected, the **UCS Control File** (path, file name) is shown below.

Centralized Channel 3211 is similar to Channel 3203. The length of the Universal Character Set Buffer is different to the UCSB of a IBM 3203-5. For this output type no data checking is designed.

Decentralized with CR/LF, FF is used as a synonym for de-centralized printers which are equipped with either parallel Centronics, or serial RS232, or LAN interfaces.

When this output type is selected Mercury converts paper advance information of the received channel commands into the appropriate number of printer control codes (pcc) (carriage return CR, line feed LF and form feed FF). The output data stream has the following format:

```
data, pcc
```

The print data will be converted as specified in section [Translation Table](#).

Decentralized with PC-Code is similar to ASCII. In contrast to the ASCII mode Mercury will not convert the received channel commands into the appropriate number of printer control codes (carriage return, line feed and form feed) but will include the channel command byte in the output data stream to the connected printer. The output data stream has the following format:

```
pcc, data, delimiter
```

The pcc byte represents the channel command byte and a carriage return (x'0D') and a line feed (x'0A') are used as delimiter. The print data will be converted as specified in section [Translation Table](#). The selection of this output type requires a printer which can interpret the pcc byte.

VFU

This option defines the advance control (VFU = Vertical Format Unit) of the emulation. It is only available with [Decentralized with CR/LF, FF](#) as output type.

The following types are available:

- None
- EVFU
- ESC

Default: None

None can be used with all page printers. The print system does not receive any information

about form length. The information included in the Forms Control Buffer (FCB) will only be stored in the emulation. All advance commands received via the channel interface will be converted into the corresponding number of Carriage Return, Line Feed and Form Feed.

EVFU supports a paper advance control format (EVFU = Electronical Vertical Format Unit) commonly known as 'P-series'. This VFU format does not require the signal line Paper Instruction (PI). Form length as well as channel codes included will be loaded into the EVFU memory of the printer. After that the lines containing channel codes can be directly addressed by control codes in range of 10(hex) to 1B(hex) during print operation. An EVFU load cycle starts and ends with special control codes and requires one byte for every print line of the form.

ESC is nearly identical to None. In contrast the printer gets information about the length of the used form by an Escape string. To obtain the highest possible compatibility to various printers, the sequence can be modified, except the values representing the length of the form. To do this, modify the following parameters. As an example, values to define the command 'ESC&lxxF' are given where xx is the required format of the value for form length:

When **ESC** is selected, three options are available:

- String Header
- Format Of Length Information
- String Trailer

- **String Header**

Define the prefix of the command string, i.g. for 'ESC&l' enter '%1b%26%6c' or '%1b&l' here.

- **Format Of Length Information**

Define the format of the value required by the printer.

Available options:

- Decimal 3 digits
- Decimal 2 digits
- Hexadecimal

Default: Decimal 3 digits

- **String Trailer**

Define the suffix of the command string, e.g. 'F'

6.3.6.2.2 Band ID

Band ID

- **EBCDIC**

The print characters of impact printers are on a steel print band. Many print bands for different character sets and national languages are available. The print band moves across the print line. When the desired print character is at the correct position, the hammer for that position strikes to print the character. Each print band has a unique ID, called Band ID. The host system has to be informed about this Band ID to recognize which print band is currently mounted on the printer.

Default: 00000000

6.3.6.2.3 Translation Table

Choose translation

Select a translation.

Available options are:

- No Translation
- EBCDIC to ASCII

Default: No Translation

The code set of data used on a channel interface is normally EBCDIC. Depending on the requirements of the printing system which is really connected to Mercury the data codes can pass Mercury in a transparent way or can be converted to ASCII.

• No Translation

Select No Translation for a code conversion table where the default settings keep all codes untouched.

If necessary, modify the translation table using the button Edit.

• EBCDIC to ASCII

Select EBCDIC to ASCII to convert all received data codes to ASCII.

If necessary, modify the translation table using the button Edit.

6.3.6.2.4 Control Codes

Control Codes (hexadecimal)

This group includes the settings of the control codes form feed (**FF**), carriage return (**CR**) and line feed (**CR/LF**). Here you can specify the hexadecimal codes being used by Mercury in the output data stream to perform the paper advance of the connected printer.

• FF Control Code

Enter the corresponding control code.

Default: 0C

• CR Control Code

Enter the corresponding control code.

Default: 0D

• CR/LF Control Code

Enter the corresponding control code.

Default: 0A

• Ignore Space

Enter the space characters that shall be ignored.

When this switch is activated by defining the hexadecimal code of the space character used by the host Mercury will reduce the resulting output data stream by those space characters, where the content of the whole print line consists only of this single space character.

Default: empty (none)

Lines Per Inch

This feature allows to change the printers line density between 6 and 8 LPI (lines per inches) controlled by the host system, if the connected printer supports such a function. For that two control strings have to be defined. Depending on the line density of the Forms Control Buffer (FCB) loaded from the host the related one will be transferred to the connected printer.

- **Escape Sequence 6 lpi**

Enter the Escape Sequence for 6 lines per inch.

- **Escape Sequence 8 lpi**

Enter the Escape Sequence for 8 lines per inch.

- **Default LPI**

Select the default lpi setting from the pull-down menu.

This option will define the default LPI setting used by Mercury as long as there is no FCB loaded from the host. If there were control strings defined the one related to the default setting will be transferred after every start of the dispatcher.

Default: 6

Page Turn Over

Select the page turn over either by line feed (LF) or by form feed (FF).

Default: by FF

These options allow to control the transition from the bottom of a page to the top of the next page.

Normally, Mercury converts all received channel commands into the appropriate number of carriage returns, line feeds and form feeds. But when the connected printer is a cut sheet printer the alteration of the font size will result in a different line capacity of the page. This problem can be solved if Mercury uses a form-feed command instead of a line-feed command to leave the bottom of the form.

BS 2000

Available only when BS2000 is used.

- **Channel 12 Conversion to FF**

This check box allows to control a special feature designed for the Siemens printer emulations SNI 3338/39, SNI 3348 and SNI 3365 (SNI2050).

The print spooler of OS2000 uses Channel 12 of a Forms Control Buffer (FCB) as the bottom of a form, generally. Page turnings between header/trailer pages and the print jobs will be performed by the channel commands x'E3' (Skip to channel 12 immediate) followed by the command x'09' (Write and space 1 line) where the print line contains 1 single space character.

Enable to make Mercury convert the described command sequence into a single form feed command in the output data stream.

Otherwise depending on the actual line counter Mercury will perform the appropriate number of line feeds to arrive bottom of form before the top of the next page will be arrived with the subsequent write command.

Default: disabled

6.3.6.2.5 Bold Printing

This section includes different converter functions to emphasize text during printing. Impact printers like those emulated by Mercury can emphasize text by over-printing the same line several times only. Depending on the technology this method would not provide the same results on a laser printer.

To get the same effect on laser printers this text must be printed a second time slightly horizontally shifted or using a bold font instead. The third solution of Mercury allows a combination of underlining text and/or making text non-recognizable and/or bold printing text.

As soon as a channel command x'01' (Write without spacing) was received the content of that line will be compared with the following print line. Depending on the selected bold printing mode Mercury will insert both at the beginning and at the end of matching contents the specified control strings.

Bold Printing Mode

Select a bold printing mode from the pull-down menu.

Dependent to the selected mode it might be necessary to define additional fields.

Available options are:

- None
- Bold Font
- Pixelshift
- Attribute

None - This selection will disable the bold mode function.

Bold Font - This selection requires to define 2 control strings which let the printer use a bold font or re-select the normal font.

Pixelshift - This selection requires to define 2 control strings which let the printer start and terminate the pixel shift mode. Often it is not necessary to define a terminating control string because of auto termination at the end of the print line.

Attribute - When this mode is selected, the following parameters are available and must be defined:

- Bold on (e.g. 1B28733342)
- Bold off (e.g. 1B28732D3342)
- Underline on (e.g. 1B266444)
- Underline off (e.g. 1B266440)
- Underline character (e.g. 6D (EBCDIC ' '))
- Overprint character (e.g. 58 (ASCII 'x'))

The used examples are escape control strings known by a HP LaserJet printer.

There are three print attributes: **bold**, **underline** and **overprint** i.e. making text unrecognizable. These attributes can be used in any combination. The received data (normally EBCDIC) will be scanned for the code specified as **Underline Character** to recognize which part of the line should be underlined. **Overprint Character** must contain the code which is used in the output data stream (normally ASCII) to make text unrecognizable.

Example: '.' is used for a space character (blank)

```
1st line: CMD: 01
Normal.Bold.Bold.&.Underline.Over.Over.&.Under
```



```
2nd line: CMD: 01
.....Bold.Bold.&.Underline.YYYY.YYYYYYYYYYYY
3rd line: CMD: 09
.....
```

will cause the following print line:

```
Normal Bold Bold & Underline XXXX XXXXXXXXXXXXX
```

'y' was exchanged by the specified **Overprint Character** (e.g. x) because being unequal to neither the original content nor the **Underline Character** nor the space character.

6.3.6.2.6 Download

This section includes two special features which can be useful in some cases.

Font & Graphic Download

Enable or disable font and graphic download.

Default: disabled

Normally, Mercury converts all received print commands into the appropriate number of carriage control codes like carriage return (0D), line feed (0A), form feed (0C). This mode was designed to allow downloads of data streams which have to be composed by several print lines because they will exceed the maximum length of a print line. But the different parts must not be separated by those carriage control codes, normally generated by Mercury because of the channel commands. When the download mode is enabled the suppression of generated control codes can be switched on and off by control strings included in the data stream.

This selection requires to define 2 control strings which let the printer activate or de-activate the download function. Both control strings have to be at the beginning of a print line for being recognized by Mercury.

- **Download start (Hex)**

Enter control string to let the printer activate the download function.

Default: 50506F6F7BD6D5

- **Download end (Hex)**

Enter control string to let the printer de-activate the download function.

Default: 50506F6F7BD6C6

Bypass

Enable or disable bypass.

Default: disabled

All control strings known by a connected printer can definitely arrive to the printer using the so-called **bypass** function, even if they include non-printable codes. Usage of this function secures that all nonprintable codes will not be converted neither from the spooler nor from Mercury, though that the sequence remains intact. This function is also very useful if the input of the codes cannot be done in hexadecimal notation and the desired code is not related to any key of the keyboard.


For the **bypass** function one printable code of the character set will be appointed to be an indicator for performing the function. The receipt of the specified bypass code instructs

Mercury will combine the following two print characters to one hexadecimal code before the print line will be transferred to the printer. However, this can only be done, if the **bypass** character is followed by either numerals from 0 to 9 or letters from A to F where the letters are not case sensitive.

- **Bypass Character (Hex)**

Enter bypass character in hexadecimal code.

If there are used two **bypass** characters one directly behind the other, Mercury will combine all subsequent print characters to hexadecimal codes up to the line's end if the string is not terminated by another **bypass** character before. This procedure allows to transmit even longer control strings to the printer in a comfortable way.

 If one of both characters which shall be combined is not within 0 to 9 and A to F, in case of a single **bypass** the function will not be executed, in case of a **bypass** string the function will be terminated at the failing position.

6.3.6.2.7 FCB

Forms Control Buffer

All line printers which can be emulated by Mercury on the channel interface to the host are equipped with a forms control buffer (FCB). There the information about form length, line density and vertical tabulators is stored when loaded from the host. This buffer shall copy the 12-channel tape controlled vertical format unit of older printers. The definition of channel codes for any line makes possible direct addressing of those lines by appropriate advance commands. The buffer has a maximum length of 255 bytes, where each byte is related to one line of the form.

- **Ignore FCB**

Enable to ignore forms control buffer.

Default: disabled

When this check box is enabled Mercury will not verify received advance commands that will address to a channel stop, if the desired channel stop is available in the FCB in use. When this option is disabled Mercury will report a data check status to the host when the desired channel stop is not specified in the FCB in use.

- **Suppress FCB**

Enable to suppress forms control buffer.

Default: disabled

When the check box **Suppress FCB** is enabled all FCBs loaded from the host will be ignored. When this function is disabled any FCB loaded from the host will be accepted and used.

Suppress FCB is only available when **Ignore FCB** is enabled.

- **Extended FCB**

Enable for extended forms control buffer function.

Default: disabled

Dependent on the used forms the length of a FCB normally will not have the whole capacity possible (a 12 inches form has 72 lines when 6 lines per inches (LPI) density are used or 96 lines when 8 LPI are used). If the system programmer is able to modify the FCBs of the host to append further information behind the termination byte, this option can be used to download

these extended FCB information to the connected printer. This function allows to perform different pre-settings of the printer depending on the used form.

With activation of [Extended FCB](#) FCBs containing the following parameters are supported:

0x1E10 ('control string of indefinite length in ASCII' 0x1F10)
0x1E11 ('1 byte of columns to suppress, 1 byte of lines to suppress' 0x1F11)
0x1E12 ('name of job max. 16 bytes in ASCII' 0x1F12)
0x1E13 ('name of print queue max. 16 bytes in ASCII' 0x1F13)

All information enclosed by the bytes 1E10 and 1F10 will be transferred in a transparent way to the connected printer.

The information enclosed by the bytes 1E11 and 1F11 will cause Mercury to strip the specified number of columns from the beginning of each print line (1st byte, decimal counter), and/or to strip the specified number of print lines from the beginning of each page (2nd byte, decimal counter).

Both parameters enclosed by the bytes 1E12/1F12 and 1E13/1F13 are only designed for a connected network printer and will cause Mercury to use the desired job name inside the network protocol and send this job to the desired printer queue.

- **Resend FCB**

Enable to resend forms control buffer.

Default: disabled

This option is related to both functions [Extended FCB](#) and [Lines Per Inch](#) (see section [Control Codes](#)). When this check box is enabled Mercury will perform all functions and actions related to the load of the FCB. All the extended information included in the used FCB will be transferred to the printer at the beginning of each job (jobs separated either by [Job End Detection](#) or [Job Timeout](#)) even when there was no FCB loaded from the host since the last FCB load.

- **Default FCB**

Enable to set default forms control buffer.

Default: disabled

Enter a file name and path or search for it with the '...' button.

When this check box is enabled Mercury will perform an initial load of the FCB with the information stored in a file on hard disk drive every time when a print process which includes a channel emulation is started.

If the check box [Suppress FCB](#) is disabled this default FCB will be replaced by any FCB loaded from the host. If [Suppress FCB](#) is enabled the default FCB will never be replaced but all loaded FCBs will be ignored instead.

6.3.6.2.8 Job Management

Job End Detection

Because the data stream which will arrive to the line printers emulated by Mercury will not have an universal indication for an end of job there exists a timeout to terminate a print job. But when there are several print jobs in the spooler of the host the different jobs will not be separated by the timeout. Therefore Mercury offers different methods for job separation.

- **Method**

Select a job end detection method from the pull-down menu.

The following options are available:

- None
- Download End
- Skip 12
- Job End
- Job Start
- Load FCB

Default: None

None disables the job separation function. A job which was started on the channel interface will be terminated by a timeout value, only. This job timeout must be defined in the settings of the Input Port.

Download End requires the activation of the Download Mode option. When being selected, every receipt of the 'Download-end' string will cause an end of job.

Skip 12 will perform a job separation after the receipt of a special channel command sequence used by the Siemens operating system BS2000.

The print spooler of BS2000 uses Channel 12 of a Forms Control Buffer (FCB) as the bottom of a form, generally. Page turnings between header/trailer pages and the print jobs will be performed by the channel commands `x' E3'` (Skip to channel 12 immediate) followed by the command `x' 09'` (Write and space 1 line) where the print line contains 1 single space character. When this method is activated Mercury will terminate a print job immediately after the receipt of that sequence.

Job End allows the definition of a text string as an indicator for an end of job. Additional parameters are available to specify a [Column Offset](#) where to search in a print line or to suppress the text string itself to prevent the string from being printed. If the text string to search for is not the last text on the page there is another parameter that will instruct Mercury to wait until the end of that page.

Selecting Job End makes the following parameters available:

- **Suppress Job End Sequence**

Enable to suppress Job End Sequence.

Default: disabled

- **Wait 'End of Page'**

Default: disabled

- **Job End Sequence (Hex)**

Enter Job End Sequence in hexadecimal code.

- **Sequence Offset**

Enter Sequence Offset.

Job Start is similar to [Job End](#). The selection of this method allows the definition of a text string as an indicator for the beginning of job. Additional parameters are available to specify a [Column Offset](#) where to search in a print line or to suppress the text string itself to prevent the string from being printed.

Load FCB allows a job separation by every FCB load from the host. If the host always will send a [Forms Control Buffer](#) before each job this method can be used for job separation.

Job Name

A Host is able to send a job name or job identification within the print job data. If such an information is present, Mercury can extract that information and pass it on to its own spooling system.

- **Extract job name from data stream**

With this option, the job name might be extracted out of the banner or job separator pages.

- **Trigger**

The trigger is a constant character sequence which precedes the variable job name in the banner page. You would describe this e.g. as a field descriptor. The trigger string has to be on the same line as the job name and before the job name.

- **Cut Offset**

This is the number of character positions (columns) between the last character of the trigger string and the first character of the job name.

- **Cut Length**

This defines the length of the job name, exactly it describes how many characters starting from the cut offset after the trigger string should be used for the job name.

Example: The banner page looks like this:

```
*****START*****START*****START*****
*
* JOBID.....:   JOB08207
* JOB NAME....: MYJOBNAME
* USERID.....: THEUSER
* SYSOUT CLASS: M
*
*****START*****START*****START*****
```

To extract the job name, the trigger would be set to 'JOBID'. Since the Job name here e.g. JOB08207 is eight characters long and starts ten characters after 'JOBID', the cut offset would be set to 10 and the cut length to 8.

A trigger 'JOBID.....:' (with colon) and a cut offset of 9 would also work

6.3.6.2.9 Performance

Data Processing

- **Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32 KB

- **Block size for output buffering**

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off

buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32 KB

Timing

• Delay before next try on read error

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read attempts.

Default: 1000 msec

• Delay before next try on write error

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.


This prevents a slow output port slowing down the system by frequently repeated write attempts.

Default: 10 msec

• Timeout for shutdown

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.

 Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

Bidirectional Communication

• Produce ACK/NAK responses

Enable to produce ACK/NAK responses.

When processing print jobs, which are received through a Channel Input Port and when processing IPDS print jobs, usually it is necessary to send feedback responses to the host system. In other situations these responses might not be applicable. For highest flexibility it can be controlled by this option whether feedback responses are sent to the host system or not.

Default: enabled

• Send stop code to host after writing <n> sec

Define the time in seconds after which a stop code is sent to the host.

This applies if a write access to the printer is not successfully completed within this time. Some hosts require responses within a specific time. If for some reason the printer goes offline or is turned off this might lead to problems with bi-directional communication.

Set to 0 to turn off this option.

Default: 0 (off)



The effective time depends on the write timeout set at the output port. I.e. if at the output port a write timeout is set to 10 sec and here 15 sec are used, the second timeout after 20 sec will send the stop code. This is because the first timeout after 10 sec is still in valid range of 15 sec.

Jobs

- **Split Job after <n> pages**

Enter the page number after which the job shall be split.

Default: 0 (no splitting)

- **reconfigure printer after splitting**

Enable to reconfigure the printer after splitting.

Default: disabled

6.3.7 Comparex Line Mode to IPDS/PCL5 Emulation

6.3.7.1 Overview

This chapter describes all the parameters of the Mercury emulation Comparex to PCL5 and Comparex to IPDS.

6.3.7.2 Parameters

6.3.7.2.1 Common

Printer Emulation

Determines the common behavior of the emulation.

- **Fanfold (continuous form)**
Emulation of a continuous form printer.

- **Cut sheet**
Emulation of a cut sheet printer.

Default form width / length

Determines the FCB width and length, even if there is no (enhanced) Line mode FCB encountered.

- **Use FCB form width / length as media size**

The emulation overwrites the configured paper sizes of the following IPDS emulation with the logical sizes defined within the FCB.

Empty page suppression

Some additional switches to avoid empty pages under certain conditions.

- **Avoid empty pages generated from 'End' or 'Page End' commands within home state**

When active, multiple consecutive 'End' (<2B><5D>) or 'Page End' (<2B><BF>) commands within 'home state' will not produce empty pages, otherwise each command generates a empty page.

- **Consider records like <40><01> as not printable**

This might avoid empty pages when the data stream is generated on a HOST environment using VSE.

When active, a record containing only a space character and a 'Write without spacing' channel command (<40><01>) does not set the page state to dirty, otherwise it will.

6.3.7.2.2 Resources

Resource Path

Determines the directory containing the resources or also where to save resources.

Default Resources

Specifies the used font and FCB of the emulation.

- **Font**

If there exists a proper font within the resource directory, this one may be selected as standard font, otherwise the internal font will be used.

- **FCB**

If there exists a proper FCB (Forms Control Block) file within the resource directory, this one may be selected as standard FCB, otherwise the internal FCB will be used.

The FCB file must be a text file which may consist of multiple text lines. Each byte of the FCB within the text file must be represented by a 2 digit hex value separated by a space character, e.g.:

```
01 00 00 00 00 00 00 00 00 00
00 00 00 00 1C
```

This byte sequence should represent actually the same bytes which would be transferred via the channel command code 0x63 = 'Load FCB'.

Above example shows a simple 3203-5 Type FCB containing 15 lines, line spacing is 6 LPI (Lines Per Inch), assigned channels are channel 1 to line 1 and channel 12 to line 15.

The FCB type may be a simple '3203-5 Type' or a 'Comparex Line Mode Type' with extensions.

1. 3203-5 Type FCB

- Fixed line spacing in a page
- The first byte, bit 3 defines line spacing (6 or 8 lpi only).

Note: Bits 0-2 must be zero, otherwise an error is forced and a default FCB loaded.

Printed Page		FCB	
		0123	4567
Top 1/3 inch		0000 ¹ 0000	0000 0000
ABC	6 LPI CH1	0000	0001
DEF	6 LPI	0000	0000
GHI	6 LPI	0000	0000
JKL	6 LPI	0000	0000
:	:	:	Channel
:	:	:	code
:	:	:	
XYZ	6 LPI	0000	0000
Bottom 1/3 inch		0000 0001 ²	0000 0000
Note: Top and bottom 1/3 inch should not be used for printing. ¹ 0000: 6 lpi, 0001: 8 lpi ² 0001: last line code			

- The channel code is defined by bits 4 - 7

Note: The top and bottom 1/3 inch should not be used for printing.

2. COMPAREX laser printers operating in Line Mode type FCB

- Fixed line spacing in a page
- The first byte, bit 3 is ignored.

Printed Page		FCB	
		0123	4567
Top 1/3 inch		0000 0000	0000 0000
ABC	6 LPI CH1	0000	0001
DEF	6 LPI	0000	0000
GHI	6 LPI	0000	0000
JKL	6 LPI	0000	0000
:	:	:	Channel
:	:	:	code
:	:	:	
XYZ	6 LPI	0000	0000
Bottom 1/3 inch		0000 0001	0000 0000
Note: Top and bottom 1/3 inch should not be used for printing.		2B	
		2B	
		FCB Extension	

For more details about FCB format and extension refer to the 'Comparex Line Mode Functional Specifications'.

Run emulation in static mode

This controls whether the emulation is unloaded between the incoming print jobs (dynamic mode) or not. If not (static mode), the emulation remains loaded for the entire life time of the print processes.

6.3.7.2.3 Job Management

Trigger sets

Administrates trigger sets for job name extraction or job manipulation.

Job name extraction

If a trigger matches, the amount of **length** Bytes with relative **offset** to the matching trigger **end** will be extracted from the line and used as job name.

e.g.

'This is a arbitrary record...'

When a trigger is defined like 'bit',offset=5, length=6, than the text 'record' will be used as job name.

There may be multiple trigger which may be active or inactive at all. When using multiple trigger, the last one encountered is valid.

Job separation

If a trigger matches, the job will be separated at the page end or at page begin.

There may be multiple trigger which may be active or inactive at all. When using multiple trigger, the last one encountered is valid.

Job control table

Allows invoking of assigned FCB and/or UCSB files from the resource directory for a specific job.

When activated, the job must contain a additional EBCDIC encoded record at the beginning.

The record itself must contain at the first position the sequence '\$/info/\$' (0x5B 0x61 0x89 0x95 0x86 0x96 0x61 0x5B) followed by a job specific name.

e.g.

'\$/info/\$BillListing...'



The first control record will be not processed (emulated), i.e. is not part of the printed job.



When the feature is activated and no '\$/info/\$' could be found at the beginning of the first record or no matching control table entry is found, the job will not be emulated, i.e. will be passed through transparently. Additionally the Mercury JOBINFO member '\$userjobid' is set to 'CPX_NO_INFO'. That '\$userjobid' can be used by a following stage to dispatch the job for example into a 'hold queue' (for example with a conditional processing 'Boolean expression' definition).

When a 'Job control table' trigger exists which partially matches to the sequence following immediately '\$/info/\$', the assigned FCB and/or UCSB file within that trigger definition will be invoked.



Here in the example, a existing trigger 'BillListing' or also 'BillList' would match, but a trigger 'Listing' would not match.

For more information regarding the FCB (Forms Control Block) file format refer to [FCB type](#).

The UCSB (Universal Character Set Buffer) file format must be a text file which may consist of multiple text lines. Each byte of the UCSB within the text file must be represented by a 2 digit hex value separated by a space character, e.g.:

```
2B 2B 00 05 04 71 68
FF FF ...
```

This byte sequence should represent actually the same bytes which would be transferred via the channel command code 0xFB = 'Load UCSB'.

Above example has one font equivalence entry assigning the 'local font number' = 0x04 to the 'font ID' = 0x7168.

⚠ The padding bytes 'FF'... can be omitted within the file.

B.6 Load UCSB

Command Code:	X'FB'
Data Length:	304 bytes
Command Format:	LOAD UCSB Data
Data Contents:	Bytes 0 through 303, 3203-5 type UCSB data or the following escape sequence: 2B 2B nn nn ll iii ll iii ... ll iii FF ... FF maximum 100 sets padding
with:	
2B 2B:	Identification for COMPAREX laser printers operating in Line Mode type UCS
nn nn:	Number of font equivalence entries x 3 + 2
ll:	Local font number (X'00' through X'7F')
iii:	Font ID (X'0004' through X'7FFF')

Note:

1. If the data sent with the LOAD UCSB command does not contain an escape sequence, the printer will establish a font equivalence record with local font number 0 and the default font. However, this will only occur, if no font was selected on the operator panel. If a font is selected on the operator panel, the printer will continue using that font even after receiving the LOAD UCSB without escape sequence.
2. If the data sent with the LOAD UCSB contains an escape sequence, the remaining bytes must be padded with bytes that have all bits set (X'FF').
3. If a LOAD UCSB command is received in PAGE STATE, it will be saved and get active when HOME STATE is entered.

For more details about UCSB format refer to the 'Comparex Line Mode Functional Specifications'.

6.3.8 Forms Overlay Language

6.3.8.1 Overview

The FOL™ to PCL5 emulation from Mercury converts print data that were created in the FOL™ printer language, so that they can be output on PCL5 compatible page printers. All the parameters are default values. They are in use until their values are overridden from FOL™ commands in the print data stream.

FOL™ stands for 'Form and Overlay Language'. FOL™ is a page description language, developed by Océ™, to control the printing process in digital printing environments. In combination with Océ™'s 'Job Automation Control' (JAC™), which is also included in the FOL to PCL5 emulation of Mercury, FOL™ is a powerful, easy-to-use tool for the formatting of line printer data and ASCII data streams.

The description of the FOL™ and JAC™ language is included in the 'Océ 9200 Series Programmer's Guide', which is available from Océ™.

6.3.8.2 Parameter

6.3.8.2.1 Printer Capabilities

Printer

- **Type**

Select printer type from the pull-down menu.

The selection of the parameters type and output allows the adaptation of the PCL5 data stream, generated from the emulation, to the requirements of special printer types.

Default: HD Digimaster

- **Output**

Select output from the pull-down menu.

Default: PCL5

Printer Operation

Fine-tuning of the created PCL5 data stream can be done here.

- **Copies by printer**

Enable to generate copies of a print data stream by the printer. Disable to generate copies of a print data stream by the emulation.

Default: disabled

- **Collate**

Enable for collated copies.

Default: enabled

This selection switches between collated and non-collated copy generation. If collate is enabled, copies of the print data stream will be generated on a 'per job' basis, otherwise copies of the print data stream will be generated on a 'per page' basis.

- **Change sign of registration offset on duplex backside**

Enable to change sign of registration offset on duplex backside.

Default: disabled

Some printers move print data on duplex back pages to the wrong direction. This switch will reverse the movements on duplex back pages.

- **Send FOL forms as PCL macros**

Enable to send FOL forms as PCL macros.

Default: enabled

- **Full page mode**

Enable to print data from edge to edge.

Default: enabled

In Full page mode data can be printed from edge to edge on the page, if the printer supports this feature. Otherwise an unprintable margin of some millimeters remains on each printed page (dependent on the type of printer).

- **PCL Graphics Compression**

Enter PCL graphics compression.

Default: 3

The compressing method of the raster graphics in the PCL5 data stream can be configured here.

PJL

- **Use PJL**

Enable to use PJL.

Default: enabled

Controls the integration of PJL (Printer Job Language) instructions in the PCL5 data stream, produced by the FOL to PCL emulation. These instructions are used for example to control the finishing of the print jobs.

- **Set PJL resolution**

Enable to set PJL resolution and select a resolution from the pull-down menu.

Default: disabled

Default resolution: 300 dpi

With this option, the printing resolution of the target printer can be configured. Available resolutions are: 100, 150, 200, 300, 600, 1200, 2400

Job step default action

The finishing options of the print jobs like stitching, stapling etc. can be configured here.

- **enable**

Enable one or more of the following options:

- Jog
- Staple
- Staple Flag Sheet

Jog activates the job based jogging of print jobs.

Default: enabled

Staple activates the job based stapling of print jobs.

Default: enabled

Staple Flag Sheet: Enabling this option will jog and staple the flag sheet of a print job together with the rest of the print job, disabling this option will do an extra jogging and no stapling of the flag sheet page.

Default: disabled

- **at ,Process' label**

Enable one or more of the following options:

- Jog
- Staple

Jog controls the jogging of print jobs while processing a ,process' label of a FOL control file.

Default: disabled

Staple controls the stapling of print jobs while processing a ,process' label of a FOL control file.

Default: disabled

- **at ,Separation' mark**

Enable one or more of the following options:

- Jog
- Staple

Jog controls the jogging of print jobs while processing a ,separation` mark of a FOL control file.

Default: disabled

Staple controls the stapling of print jobs while processing a ,separation` mark of a FOL control file.

Default: disabled

- **at ,Segmentation' mark**

Enable one or more of the following options:

- Jog
- Staple

Jog controls the jogging of print jobs while processing a ,segmentation' mark of a FOL control file.

Default: disabled

Staple controls the stapling of print jobs while processing a ,segmentation' mark of a FOL control file.

Default: disabled

6.3.8.2.2 Paper

Input tray mapping

The allocation of FOL tray assignments to the appropriate PCL tray assignments can be made here.

- **Host tray #**

Enter the number of the respective FOL input tray.

Up to 24 different tray allocations are possible (0 - 23).

Default: 0

- **PCL tray ID**

Enter the ID number of the associated PCL input tray.

- **Registration Offset [mm]**

This option can be used to compensate printer inaccuracies while feeding paper. A tray-dependent offset in millimeters for moving the print data on the page can be configured here.

Horizontal: enter horizontal offset.

Default: 0.00 (mm)

Vertical: enter vertical offset.

Default: 0.00 (mm)

- **Output bin mapping**

The allocation of the FOL output bin assignments to the appropriate PCL output bin assignments can be made here.

- **Host bin #**

Enter the number of the respective FOL output bin.

Up to 24 different tray allocations are possible (0 - 23).

Default: 0

- **PCL bin ID**

Enter the ID number of the associated PCL output bin.

- **Paper Size [mm]**

All adjustments for the configuration of the default paper size can be done here.

- **Type**

Select type of default paper size.

Available paper sizes are:

Standard: Executive, Letter, Legal, Ledger, A5, A4, A3, B5

User defined: User defined, User defined2, User defined3.

The respective paper size appears in the fields 'Width' and 'Height'.

Width: enter paper width in mm.

Height: enter paper height in mm.

- **Printable Area Offset**

Horizontal: enter offset in millimeters, which the printable area of the paper deviates from the left edge of the page.

Vertical: enter offset in millimeters, which the printable area of the paper deviates from the top edge of the page.

- **PCL ID**

Define the PCL identification number of the selected paper size.

- **Paper Assignment...**

Opens a configuration dialog for the allocation of FOL paper name to PCL paper specifications. User definable FOL paper names can be assigned to specified PCL paper sizes. Additionally a media name can be configured.

See [Paper Assignment](#) for details.

Orientation

Page orientation

Select a default page orientation from the pull-down menu.

Available options are:

- Portrait
- Landscape
- Reverse Portrait
- Reverse Landscape

Paper Assignment

Define logical papers in this dialog.

User defined name

Enter a name for the paper size.

Paper Size

Select a paper size from the pull-down menu.

Host Input Tray #

Enter number of host input tray that contains the paper.

Printer Media Name

Enter the printer media name.

Add

Click on 'Add' to add the above defined paper name assignment to the list below.

Remove

Click on 'Remove' to remove the highlighted entry from the list below.

Edit...

Click on 'Edit' to edit the highlighted entry in the list below.

A dialog appears where you can edit

- User defined name
- Paper Size
- Host Input Tray #
- Printer Media Name

Back to [Paper](#)

6.3.8.2.3 Page Properties

In this configuration dialog all page-related default adjustments can be done.

Duplex

- **Default Mode**

Select a default mode from the pull-down menu.

Available options are:

- Simplex
- Long edge duplex
- Short edge duplex

Default: Simplex

- **Force Duplex mode**

Enable to force duplex mode.

Enabling this feature will convert a simplex and duplex mixed print job to a complete duplex print job. Formerly simplex pages are converted now to duplex pages with empty backsides. This option must be set if a printer doesn't support switching between simplex and duplex pages inside a print job.

Default: disabled

- **Page Turnover**

Select a page turnover method.

Available options:

- Duplex Side Selection
- Form Feed

Default: Duplex Side Selection

Default margins [mm]

Enter the adjustments of the default page margins in millimeters.

Available margins:

- Top
- Left
- Right
- Bottom

Default: 0.00 (all)

Bulk text

- **Wrap text lines after position**

Enter number of column after which the emulation will do an automatic carriage return and linefeed.

Default: 512

- **Wrap page after line**

Enter number of lines after which the emulation will do an automatic form feed.

Default: 256

- **Cursor start horizontal [mm]**

Define the initial horizontal cursor position on the page in millimeters.

Default: 0

- **Cursor start vertical**

Indicates the initial vertical cursor position on the page in millimeters.

Default: 0

- **Graphics line width [μm]**

Define the default width of graphic lines in micrometers.

Default: 24

- **Copies (default)**

Define the default number of copies of a print job.

Default: 1

- **Allow empty pages**

Enable to allow empty pages.

Default: disabled

6.3.8.2.4 Fonts

- **FOL Font substitution**

In this dialog the allocations between FOL fonts and appropriate PCL5 fonts are made.

- **FOL Font name**

Enter a FOL font name or select one from the pull-down menu.

Pre-defined options:

- line printer
- univers

Default: line printer

- **PCL Typeface ID**

Enter the PCL typeface ID.

Default: 4102

- **Spacing**

Select the character width from the pull-down menu.

Available options:

- Fixed
- Proportional

Default: Fixed

- **Symbol Set #**

Enter the number of the symbol set, i.e. the ID of the PCL character set of the selected font.

Default: 277

- **Size adjustment**

Enter a balance factor to the character width.

Default: 0.75

- **Add**

Click on 'Add' to apply the new settings.


- **Remove**

Select a FOL Font name and click on 'Remove' to delete the selected font allocation.

 You can undo this deletion by clicking 'Cancel'.

- **Change**

Select a FOL Font name and click on 'Change' to modify an already configured font allocation.

 This will overwrite existing font parameters.

Edit default font...

All assignments of the default font can be changed in this configuration dialog.

See [Edit default font...](#) for details.

Edit Default Font

PCL Typeface ID

Enter the PCL typeface ID.

Default: 4099

Spacing

Select the character width from the pull-down menu.

Available options:

- Fixed
- Proportional

Default: Fixed

Pitch

Define the horizontal distances between characters of fixed spaced fonts. The value must be entered in number of characters per inch.

Default: 10

Height

Define the character height in points (1 / 72 inch) of proportional spaced fonts.

Default: 10

Stroke

Select the stroke of the font from the pull-down menu.

Available options are: Ultra Thin, Extra Thin, Thin, Extra Light, Light, Demi Light, Semi Light, Medium, Semi Bold, Demi Bold, Bold, Extra Bold, Black, Extra Black, Ultra Black

Default: Medium

Style

Select the style of the font from the pull-down menu.

Available styles are: Upright, Condensed Italic, Italic, Condensed, Compressed, Expanded, Outline Shadowed, Outline, Inline, Shadowed

Default: Upright

Symbol Set #

Enter the number of the symbol set, i.e. the ID of the PCL character set of the selected font.

Default: 277

Back to [Fonts](#)

6.3.8.2.5 Performance**Data Processing****• Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this

value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32768 Bytes

- **Block size for output buffering**

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32768 Bytes

Timing

- **Delay before next try on read error**

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read attempts.

Default: 1000 msec

- **Delay before next try on write error**

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.


This prevents a slow output port slowing down the system by frequently repeated write attempts.

Default: 10 msec

- **Timeout for shutdown**

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.

 Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

6.3.8.2.6 Resources

Resource Path

Define the path from where the PCL5 to PDF emulation can access resources like fonts, etc.

Each FOL to PCL emulation has its own directory for storing emulation-specific resources. The FOL files for describing job layouts, forms, flag sheets etc will be stored in this directory.

Separation Information file

Define the file and path where a separation information file (extension .SIF) is stored. This file contains information and criteria for separating print jobs. The FOL to PCL emulation uses this file for dynamic job separation.

Ticket name from job data header

Enable to receive ticket name from job data header.

Default: disabled

Applies only with a channel emulation before the FOL emulation. There you can add the corresponding ticket name from the LCL into the first line (header) of the data stream.

Example:

Get ticket name from job header, if there is a channel emulation with SNI3365 upstream, e.g. \$\$TCK01234\$\$.

The channel emulation (SNI3365 / IPDS Commands) should be set to:

- * "Suppress" / "New LCL Command: Replace"
- * String header: \$\$TCK
- * Format of Length Information: Decimal 5 Digits
- * String Trailer: \$\$

6.3.9 ATLAS FD Emulation

6.3.9.1 Overview

The [Mercury ATLAS FD](#) Emulation uses Design specifications created with the ATLAS Formular Designer for formatting and converting the content of XML data to a printable data stream.

Some Features

- Any local Windows Printer, capable of redirecting his output to a file port, can be used for creating the printdata stream.
- Duplex mode, papersource and number of copies can be selected within the Emulation configuration GUI .
- Via a optional XML-Mapping file, the selection of the used design file can be done dependent of the content of the XML input data.
- The optional XML-mapping file can also used to create one merged printjob using multiple design-files. For each design file printer-options like duplex mode, papersource and job copies can be set separatly.
- Beside using a local printer driver, it is possible to create PDF or PDF/A output.

Things you have to consider before starting the emulation

Besides the variable XML data, you need to place the ATLAS design files and all necessary

external resources used by the design files in the appropriate resource folders of the [Mercury ATLAS FD Emulation](#). Please see [Resources](#) Chapter .

Contents of this manual

In the Chapter **Parameters** you find detailed descriptions of the configuration options available in the graphical user interface of the emulation.

In the Chapter **Appendix** you can find a detailed format description of the optional XML mapping files used for dynamic design file selection.

6.3.9.2 Parameters

6.3.9.2.1 Common

Output driver

Selects the driver used for creating the output data stream after formatting the data using design file.


Choose '**Local printer**' to select a locally installed printer driver to create the output data stream. The specific printer then have to be selected with the '**Printer**' Drop down list below the Output driver selection list.

Choose '**PDF Output**' to create PDF or PDF/A output data. See the [PDF Export settings](#) Chapter for PDF options.

Printer

If '**Local printer**' is selected as Output driver, you have to select one of the locally installed printers for creating the output data stream. The used PDL e.g. Postscript or PCL depends on the selected printer driver. For the conversion process the Output port of the selected printer will be automatically redirected to file output. The generated output file will be captured by the [Mercury](#) ATLAS FD Emulation and send to the configured [Mercury](#) Output interface.

The Duplex mode, papersource, coloring mode and job copy setting of the chosen printer can be done via the **Printer settings** GUI of the [Mercury](#) ATLAS FD Emulation. All other printer specific configuration options will be taken from the default configuration of the selected Windows printer. Please change the options of the chosen printer in the Windows device and printer list so that it matches the needs for your data conversion.

 Because the [Mercury ATLAS FD Emulation](#) runs under the SYSTEM user account, you have to set the default printing options for this user, if you want to create a printer with specific settings. This can be done by first selecting the printer properties of the desired printer in the Windows "Devices and Printers" list. In the "General" section of the printer properties choose the "Preferences..." button to set the default print settings for the SYSTEM user account.

Print error info page, if a conversion error occurs

If you activate this option, a error page will be printed, if a error occurs during print file creation. The error page contains the same information which is written to the [Mercury](#) eventlog. Additionally some more information will be printed like the name of the job, the used designfile and the number of pages which may have been created until the error occurs. Of course this additional information will only be written to the error page, if it was available at the time, when the error occurs.

It is also possible to create a user defined design file for the error page. Please see the "**Design file used for error info page**" option at the [Resources](#) settings for more informations.

Input data splitting

It is possible to split the variable XML input data before the output data stream is created. The splitting will be done based on the XML child node name entered in the edit box below "**Split input XML data stream on node with name:**". For each detected child node with the name

specified here a separate XML file will be created internally. The necessary parent nodes will be automatically added using the content of the original XML input data file. Of course this parent nodes then all have the same content. For each splitted xml data file a single output data file will be generated.

If you wish to keep the content of all output split parts in one single file, you have to activate the **"Keep output data split parts in one single file."** option. But keep in mind that this single file is a concatenation of independent documents. Not every Page description Language does support such a concatenation. E.g. a PDF viewer will generate an error or will only show the first document.

Example:

Variable XML input data:

```
<?xml version="1.0" encoding="UTF-8"?>
<MyXMLData>
  <MetaData>
    <DataType>Library</DataType>
    <DataCount>0008</DataCount>
    <FormType>LibList</FormType>
    <FormSubType>booklist001</FormSubType>
    <FormVersion>1.01</FormVersion>
    <FormPrintType>TestPrint</FormPrintType>
    <FormPrintTime>2015-03-04:18:53:53.182547</FormPrintTime>
  </MetaData>
  <library>
    <book category="COOKING">
      <title lang="en">Everyday Italian</title>
      <author name="Giada De Laurentiis" country="it"/>
      <year>2005</year>
    </book>
    <book category="CHILDREN">
      <title lang="en">Harry Potter</title>
      <author name="J K. Rowling" country="uk" />
      <year>2005</year>
    </book>
  </library>
</MyXMLData>
```

Using the node name **"book"** as splitting node, the data file above will be splitted to the following separate input data files:

File 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<MyXMLData>
  <MetaData>
    <DataType>Library</DataType>
    <DataCount>0008</DataCount>
    <FormType>LibList</FormType>
    <FormSubType>booklist001</FormSubType>
    <FormVersion>1.01</FormVersion>
    <FormPrintType>TestPrint</FormPrintType>
    <FormPrintTime>2015-03-04:18:53:53.182547</FormPrintTime>
```

```

</MetaData>
<library>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author name="Giada De Laurentiis" country="it"/>
    <year>2005</year>
  </book>
</library>
</MyXMLData>

```

File 2:

```

<?xml version="1.0" encoding="UTF-8"?>
<MyXMLData>
  <MetaData>
    <DataType>Library</DataType>
    <DataCount>0008</DataCount>
    <FormType>LibList</FormType>
    <FormSubType>booklist001</FormSubType>
    <FormVersion>1.01</FormVersion>
    <FormPrintType>TestPrint</FormPrintType>
    <FormPrintTime>2015-03-04:18:53:53.182547</FormPrintTime>
  </MetaData>
  <library>
    <book category="CHILDREN">
      <title lang="en">Harry Potter</title>
      <author name="J K. Rowling" country="uk" />
      <year>2005</year>
    </book>
  </library>
</MyXMLData>

```

6.3.9.2.2 Resources

Settings for external resources and their locations.

Global resource path

Formatting data, using the [Mercury ATLAS FD Emulation](#), does not work without using external resource files. At least one design file created with the ATLAS Formular Designer is necessary to create the formatted output datastream. All external resources have to be located directly or in subdirectories of the Global resource path specified here by using the attached **Browse...** button.

If the design file contains locations of external resources using absolute path specifications, during conversion these pathes are automatically converted to pathes relative to the Global resource path, e.g. the path `c:\design\images\myimage.jpg` will be converted to: `<Global resource path>\design\images\myimage.jpg`.

Before the conversion is started, all necessary resources have to be copied to their appropriate folders.

Atlas FD default design file

If the selection of the design file does not depend on the content of the variable XML data stream, or in other words, you need only one dedicated design file for your conversion process, you can specify this file directly using the attached **Browse...** button. The file specified here will also be used, if the dynamic design file selection, using an XML Mapping file, does not succeed.

Override design internal XML data XSL transformation file

In most cases you have to specify a XSL transformation file, when you create a new design using the ATLAS Formular designer. The XSL transformation file will be used to convert the XML input data format so that, amongs other things, all necessary fields can be easily accessed using the Drag and Drop mechanism during the design process. The name and full path of the XSL transformation file is stored inside the design file and will be converted, as noted above in the description of the Global resource path, to a path relative to the global resource path. Alternatively you can specify the location and file name directly here in the Resource settings, if you activate the **Override design internal XML data XSL transformation file** option and select the transformation file with the attached **Browse...** button. The default selection in the Browse file Dialog will be "None", meaning that no XSL transformation file will be used for data conversion.

Use Mapping file for dynamic design file selection dependent on data content

If you activate this option and by selecting a XML mapping file by using the attached **Browse...** button, the selection of the design file used for data conversion will be done dynamically dependent on the content of the variable input XML data stream. If this selection process is not successful the **Atlas FD default design file** will be used. The XML Mapping file have to be created separately with a text or XML editor. Please have a look at the [XML mapping file specification](#) Chapter to find out how to create such a XML mapping file.

Design file used for error info page

If you activate the **Print error info page, if a conversion error occurs** option in the [Common](#) settings, a error page will be created, in the case if a error occurs during print file generation. You can use your own design file created with the ATLAS Formular Designer for the layout of the error page. To do this you have to select the previously created design file using the attached **Browse...** button. The following parameter names can be used during designing the error page to include runtime information:

<code>rt_errorinfo1</code>	: Textual Information about the occurred error.
<code>rt_errorinfo2</code>	: Advanced textual information about the occurred error.
<code>rt_jobname</code>	: Name of the printjob (variable XML data file) which caused the error.
<code>rt designfile</code>	: contains the ATLAS designfile used for conversion (if the file name was available at the time the error occurs).
<code>rt printer used</code>	: Local Windows printer, which is used to create the print output datastream.
<code>rt created pages</code>	: Number of pages, which are created until the error occurs.
<code>rt splitpart</code>	: Number of the variable XML data file split part processed at the

time, the error occurs.

`rt_printjob_submitted`: ="Yes" , if the prinjob was send to the output port, = "No" if no data for this split part was send to the output port.

All Parameters have to be created as are text type parameters.

⚠ If the error page should be used to cover syntax errors in the variable XML data file, the design file should be created without using a XML datasource, otherwise the error page creation itself will fail with a error.

The location of the **Resource path used for temporary files and emulation specific resources** normally should not be changed by the user.

The '**Run emulation in static mode**' option controls whether the emulation is unloaded between incoming print jobs (dynamic mode) or whether it remains loaded for the entire duration of the Print processes (static mode).

Default is disabled.

6.3.9.2.3 Performance

The overall data handling specified in the 'Performance' tab controls specifies the data block size and the timing of the data transfer between the [Mercury](#) components.

Normally these default values don't have to be modified, they are compatible to most systems.

⚠ Please don't change these values yourself, only the support can give you the right values.

6.3.9.2.4 Printer settings

The printer settings are used, if '**Local printer**' is chosen as Output driver in the Common settings.

⚠ Dependent on the number and attributes of the installed printers, it may take a while until the content of the Printer settings page is shown.

⚠ Except for the Job merge mode, all print options also can be dynamically selected during data conversion using a XML mapping file. Please see the [Resources](#) chapter and the Chapter [XML mapping file specification](#) in the Appendix to get more information about XML mapping files.

Job merge mode

The Job merge mode refers to using more than one ATLAS Design file for generating the printer output. You can specify more than one design file, if you use a XML mapping file to accomplish data dependent selection of the design file(s). Please see the [Resources](#) chapter and the Chapter [XML mapping file specification](#) in the Appendix to find out how you can create a XML mapping file for dynamic design file selection.

If you choose '**Append pages to the first job**' as Job merge mode, the first design file specified in the XML mapping file will be used to create the printjob. The pages created from the subsequent design files will be appended to the printjob. This mode will work for all PDL languages, but you cannot change the Duplex- and Job copy setting for a subpart of the job created by a subsequent single design file.

This mode will always be active, if you choose 'PDF Output' as the Output driver in the common settings.

If you choose '**Concatenate job files**' as merge mode, each design file specified in the XML mapping file will create a single isolated printjob. The separate jobs are merged before they are sent to the configured [Mercury](#) Output interface. In this merge mode you can specify different Duplex and Job copy settings for each subpart of the job created by a single design file.

⚠ Please keep in mind, that it depends on the printer driver and the PDL used by the driver, if merging single print jobs in this way will produce valid data stream.

Duplex mode

If the chosen printer is capable of printing on both sides of the paper, you can select the Duplex mode for generating the output datastream by choosing one of the options in the drop down list.

If you choose the option '**from printjob**', the Duplex setting will be taken from the '*PrintDuplex*' parameter setting of the design file. If no *PrintDuplex* parameter is specified, the default setting of the printer driver will be used.

Please see the ATLAS FD Documentation and the [XML mapping file specification](#) Chapter in the Appendix for the possible values of the *PrintDuplex* parameter.

⚠ If you use a XML mapping file for selecting the design file(s) and you specify a '*PrintDuplex*' attribute for the selected design file in the mapping file, the Duplex mode specified in this drop down list and the Duplex mode specified in the design file will be overridden by the *PrintDuplex* setting done in the XML mapping file.

Paper source

Use this drop down list to select the paper tray used as paper source for the pages generated during the conversion. In case of using more than one design file (see Job merge mode) you can specify different paper sources for each sub job in the XML mapping file.

If you choose the option '**from printjob**', the Paper source setting will be taken from the '*Papersource*' parameter setting of the design file. If no *Papersource* parameter is specified, the default setting of the printer driver will be used.

The options shown in the drop down list (and the permissible values for the *Papersource* parameter) depends on the specific printer chosen in the Common settings.

Different paper source for first page

If you want the paper for the first page to be drawn from a different tray, e.g. if the design generates a banner or a separation page, you can set this tray by choosing the appropriate option from the Different paper source for first page drop down list.

If you choose the option '**from printjob**', the Paper source for the first page will be taken from the '*Papersource1*' parameter setting of the design file.

The options shown in the drop down list (and the permissible values for the *Papersource1* parameter) depends on the specific printer chosen in the Common settings.

⚠ If you use a XML mapping file for selecting the design file(s) and you specify a '*Papersource*' or a '*Papersource1*' attribute for the selected design file in the mapping file, the Paper source specified in the Printer settings and the Paper source specified in the design file will be overridden by the *Papersource* (*Papersource1*) parameter value of the XML mapping file.

⚠ If you select a new printer in the common settings the papersource settings will be reset to their default '**from printjob**' value.

Color mode

If the chosen printer does support color printing you can choose the color mode of the conversion by selecting one of the values of this drop down list.

If the default setting "from printjob" is selected the color mode will be taken from the "ColorMode" attribute of the fitting DesignFile node in the mapping file, if present. If not the color mode will be taken from the "ColorMode" parameter of the design file. If no such parameter is specified in the design file, the default setting of the printer will be used for the coloring mode of the print output.

⚠ If the printer driver of a color capable printer does not support the standard application interface for setting the coloring mode of the printout, the default printer setting will be used.

Job Copies

Enter a value greater than 1 to produce additional copies of the print job. A value of 0 indicates, that the number of job copies will be taken from the value of the '*PrintCopies*' parameter of the design file. If this parameter is not specified in the design file, the default setting of the printer driver will be used.

⚠ If you use a XML mapping file for selecting the design file(s) and you specify a '*Printcopies*' attribute for the selected design file in the mapping file, the number of job copies specified in the Printer settings and the value of the *PrintCopies* parameter specified in the design file will be overridden by the *Printcopies* parameter value of the XML mapping file.

6.3.9.2.5 PDF Export settings

The PDF export settings are used, if '**PDF output**' is chosen as Output driver in the Common settings.

Image Quality

Please choose the appropriate Image quality setting from the drop down list. Higher quality settings will produce PDF documents larger output sizes.

Compress fonts and page content

If this option is activated, the resulting PDF file is smaller in size, but it takes more time and resources to export a document to PDF. Conversely, if this option is deactivated, the exporting procedure is quick; however, it produces a larger PDF file.

Generate PDF/A-2b compatible output

Activate this option, if the generated PDF documents should be compliant with the PDF/A-2b specification.

6.3.9.3 Appendix

The appendix give some additional Information.

6.3.9.3.1 XML Mapping file Specification

If your conversion project uses multiple design files, which should be selected dependent on the content of the variable XML input data, you can generate a XML mapping file to accomplish this task.

A XML mapping file can also be used to generate one report by merging the output of a sequence of design files.

In the following chapter you will find all information necessary for creating your own mapping file. You can use any ASCII text or XML file editor (e.g. Notepad++) to create a Atlas FDE XML mapping file. It is recommended to obtain basic knowledge of the XML syntax. To specify Conditions for design file selection some knowledge of XPath expressions may be necessary. Have a look at <http://www.w3schools.com/xml/> .

Syntax of the Atlas FD Emulation XML Mapping file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AtlasFDEMapping version = "1.00">
<Mappings>
  <Mapping>
    <Condition
Reference="Referencetext">XPathExpressionToDataNode</Condition>
    ... insert more more <Condition> statements if necessary ...
    <DesignFile>design1.repx</DesignFile>
    ... optional insert more <DesignFile> statements, which will be
sequentially processed, creating one merged printjob
  </Mapping>
</Mapping>
  ... insert more Mappings for selection of other design files...
</Mapping>
</Mappings>
</AtlasFDEMapping>
```

<Mapping> nodes and <Mappings> list

The Atlas XML Mapping file contains a list of **<Mapping>** nodes which are defined as child nodes of one **<Mappings>** List. Each Mapping must contain at least one **<DesignFile>** node. Normally one ore more **<Condition>** nodes will be added to each Mapping.

The Mapping nodes will be processed from top to down of the mapping file. If all Condition entries of a Mapping are fulfilled, the design file(s) specified inside this Mapping will be used to create the output data stream. All Mapping nodes specified after this selected Mapping will not be processed !

If a Mapping node does not contain a **<Condition>** entry, the design file of this mapping will always be used to create the output data, as soon as this mapping is processed. Therefore the only sense of a Mapping without Condition is to specify a default design file, which will be

used, if the conditions of all other Mappings do not apply. Such a default Mapping entry always have to be the last <Mapping> node in the <Mappings> list.

⚠ If no default design file is specified in the Atlas XML Mapping file, the **Atlas FD default design file** specified in the [Resources](#) settings will be used for data conversion.

Optional attributes for the <Mapping node>:

For each Mapping node it is possible to specify a separate XSL transformation data file which will be used to transform the XML input data to a format which is easier to use during the design process. Please have a look at the [Resources](#) settings and the documentation of the Atlas Formular Designer for more information about the XSL transformation files.

You can specify the XSL transformation file in the Mapping file using the following syntax:

```
<Mapping xslt="xslfilename.xslt">
... <Condition> definitions ...
... <DesignFile> definitions ...
</Mapping>
```

The file `xslfilename.xslt` will be searched in the Global resource path specified in the [Resources](#) settings. You can optionally specify a subdirectory along with the filename, e.g. if you use `<Mapping xslt="xslt\mybooklib.xslt">` to set the XSL transformation file to `mybooklib.xslt`, the file will be searched in the `xslt` subdir of the global resource path. If you specify a empty filename as xslt name (... `xslt=""`...), no transformation file will be used.

⚠ The XSL tranformation file configuration done in the Mapping file will override the XSL file setting in the [Resources](#) settings.

Additionally it is possible to select the printer by using the `PrinterName` or `PrinterNameXPath` attribute of the mapping node:

```
<Mapping PrinterName="windows printer name">
... <Condition> definitions ...
... <DesignFile> definitions ...
</Mapping>
```

or

```
<Mapping PrinterNameXPath="XPath expression to printername">
... <Condition> definitions ...
... <DesignFile> definitions ...
</Mapping>
```

The printer name specified as the `windows printer name` value repectively the content of the variable `xml datafile node` at `XPath expression to printername` should be a valid local printer name in the Devices and Printer list of the [Mercury](#) server machine.

⚠ The printer selection done in the Mapping file will override the printer selection in the [Common](#) settings.

<Condition> nodes

Within a <Condition> node you can specify a condition for selecting the design file(s) specified in the according Mapping entry. The content in the variable XML input data located at the *XPathExpressionToDataNode* XPath Expression will be compared with the text specified in the *Referencetext* attribute of the Condition node. If both values are equal the condition will be considered as fulfilled. If there is more than one <Condition> node specified in the Mapping, all Conditions have to be fulfilled for selecting the associated design file(s).

Example1:

Variable XML input data:

```
<?xml version="1.0" encoding="UTF-8"?>
<MyXMLData>
  <MetaData>
    <DataType>Library</DataType>
    <DataCount>0008</DataCount>
    <FormType>LibList</FormType>
    <FormSubType>booklist001</FormSubType>
    <FormVersion>1.01</FormVersion>
    <FormPrintType>TestPrint</FormPrintType>
    <FormPrintTime>2015-03-04:18:53:53.182547</FormPrintTime>
  </MetaData>
  <library>
    <book category="COOKING">
      <title lang="en">Everyday Italian</title>
      <author name="Giada De Laurentiis" country="it"/>
      <year>2005</year>
    </book>
    ... more <book> nodes ...
  </library>
</MyXMLData>
```

Atlas XML Mapping file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AtlasFDEMMapping version = "1.00">
  <Mappings>
    <Mapping>
      <Condition
Reference="Library">/MyXMLData/MetaData/DataType</Condition>
      <Condition
Reference="LibList">/MyXMLData/MetaData/FormType</Condition>
      <Condition
Reference="booklist001">/MyXMLData/MetaData/FormSubType</Condition>
      <DesignFile>design\atlas_library_01.repx</DesignFile>
    </Mapping>
    ... other conditional mappings ...
    <Mapping>
      <DesignFile>design\atlas_default.repx</DesignFile>
    </Mapping>
  </Mappings>
</AtlasFDEMMapping>
```

In the Example1 above the Atlas FD design file *atlas_library_01.rpx* will only be used for data conversion, if the content of the <DataType> node equals "Library" and the content of the

<FormType> node equals "LibList" and the content of the <FormSubType> node equals "booklist001". If any of the three conditions is not fulfilled and no **other conditional mapping** with matching conditions is defined, the design file *atlas_default.repx* will be used for data conversion. The default designfile Mapping

The Atlas FD design files *atlas_library_01.rpx* and *atlas_default.repx* have to be stored in the *\design\ subdirectory* of the *Global resource path* defined in the [Resources](#) options.

Of course, the XPath language allows much more complex expressions for specifying a location in the XML input data. E.g. the condition

```
<Condition
Reference="COOKING">/MyXMLData/library/book[1]/@category</Condition>
```

will only fulfilled, if the 'category' attribute of the first book entry equals the value 'COOKING'.

Optional attributes for the <Condition> node:

For each <Condition> node, several optional attributes can be specified, which are used to accomplish more complex options for detecting a specific type of XML data.

if you specify the **RefSource="LocalName"** attribute, instead of the content of the *XPathExpressionToDataNode* XML data node(s), the name(s) of the nodes will be used for comparison with the *Reference* value. This may be useful if no Metadata header is available in the variable XML input data. For practical usage the the following example:

Example2:

Variable XML input data:

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<CustomerInfo>
  <Customer>
    <USER_ID>10144</USER_ID>
    <USER_TITLE>Mr</USER_TITLE>
    <USER_FIRST>Pravin</USER_FIRST>
    <USER_LAST>Chirac</USER_LAST>
    ... more customer data ...
  </Customer>
  ... more Customers...
</CustomerInfo>
```

Atlas XML Mapping file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AtlasFDEMapping version = "1.00">
<Mappings>
  <Mapping xslt="">
    <Condition RefSource="LocalName"
Reference="USER_ID">/CustomerInfo/Customer[1]/*[1]</Condition>
    <DesignFile>design\testreport1.repx</DesignFile>
  </Mapping>
  ... other conditional mappings ...
</Mappings>
</AtlasFDEMapping>
```


In the Example2 above the *XPathExpressionToDataNode* specifies the first child node of the first "<Customer>" node of the variable XML input data. Without the `RefSource="LocalName"` attribute the value of this node (10144) would be used for comparison with the Reference value. With `RefSource = "LocalName"` the node name "USER_ID" is used for comparison with the Reference value and the condition will be fulfilled.

You can also specify the operation used to compare the *XPathExpressionToDataNode* node content (or name) with the Reference value. You can do this by specifying the `RefOp="operator"` attribute. The following operators are allowed: `=`, `<`, `>`, `!=`, `>=` and `<=`. If the RefOp attribute is not specified the "=" operator will be used.

Example for using the RefOp attribute:

```
<Condition RefOp=">"
Reference="10000">/CustomerInfo/Customer[1]/*[1]</Condition>
```

Using the variable data file from Example2 above the above condition will be fulfilled, if the content of the first <USER_ID> node will be greater than 10000, which is the case in this example.

 If the *XPathExpressionToDataNode* XPath expression specifies more than one data node all values will be concatenated, using ';' as separator character.

<DesignFile> nodes

Each <Mapping> node must contain at least one <DesignFile> Definition. If the optional conditions specified in the according <Mapping> node are fulfilled, the design file(s) specified in the <DesignFile> node will be used to create the formatted output data.

The basic implementation of a DesignFile node only contains the name of the designfile with his optional subdirectory to the global resource path:

```
<DesignFile>design\design1.repx</DesignFile>
```

In the example above the design file with the name *design1.repx* will be searched in the *design* subdirectory of the global resource path. If the file is found and if it is a valid Atlas FD design file, it will be used for formatting the output data. If the file is not found the conversion will be stopped and a appropriate error message will be logged.

If multiple design files are specified inside one <Mapping> node, all design files will be applied consecutively in the order specified in the <Mapping> node to create one output data file using the Job merge mode specified in the [Printer settings](#). If **PDF Output** is chosen as the **Output driver** in the [Common](#) settings, the job merge mode will always be on page by page basis, meaning that all pages created from subsequent design files are appended to the job created with the first design file.

```
...
<DesignFile>design\design1.repx</DesignFile>
<DesignFile>design\design2.repx</DesignFile>
<DesignFile>design\design3.repx</DesignFile>
...
```

In the example above the variable XML input data first will be used to create formatted output applying *design1.repx*, then applying *design2.repx* and finally applying *design3.repx*. Then the selected Output driver will be used to create one output data stream using the Job merge

mode defined in the [Printer settings](#).

It is also possible to generate the name of the designfile directly from the content of one or more nodes of the variable XML input data. You can use the <DesignFile> node attributes `fromXPath="XPathExpressionToDataNode"` and `fromXPathEnd="XPathExpressionToDataNode"` to specify a XPath expression to a node in the variable XML data file. The content of this node will then be used as the name of the designfile. If you specify more than one <DesignFile> node with the attribute `fromXPath` the content of of the all nodes will be concatenated to generate the designfile name until a <DesignFile> node with the attribute `fromXPathEnd` is detected or no more <DesignFile> nodes are specified. You can specify additional static text in the <DesignFile> node content, which will be appended to the variable XML datafile node content extracted using `XPathExpressionToDataNode`. This option can e.g. used to append the file extension `.repx` to the designfile name.

If the designfile name is extracted from variable data file content, it can be necessary to specify the optional subpath to the designfiles manually. For this task the optional <DesignFile> node attribute `subpath="SubpathfromGlobalResPath"` can be used. The path specified in `SubpathfromGlobalResPath` (ending with "\"), will be concatenated with the Global resource path defined in the [Resources](#) options to specify the location of the design file(s).

Example mapping file using the `fromXPath` and `fromXPathEnd` attributes:

(This example uses the Variable XML input data specified in Example1 at the beginning of this topic)

```
<?xml version="1.0" encoding="UTF-8"?>
<AtlasFDEMapping version = "1.00">
<Mappings>
  <Mapping>
    <DesignFile subpath="design\"
fromXPath="/MyXMLData/MetaData/FormType" >_</DesignFile>
    <DesignFile subpath="design\"
fromXPathEnd="/MyXMLData/MetaData/FormSubType" >.repx</DesignFile>
  </Mapping>
</Mappings>
</AtlasFDEMapping>
```

The example above will generate the designfile name `LibList_booklist001.repx`. The designfile will be searched in the `design` subfolder of the Global resource path defined in the [Resources](#) options.

More Optional attributes for the <DesignFile> node:

For each <DesignFile> node, several optional attributes can be specified, which are used to set some conversion and print settings for a specific design file. If one of those parameters is specied in a <DesignFile> node, it overrides the corresponding settings done in the Emulation GUI and the equal named Parameters, which can be specified in the designfile using the Atlas Form Designer.

Syntax and permissible parameters:

```
<DesignFile PrintCopies= "JobCopies" PaperSource = "PaperTray"
```

```
PaperSource1 = "PaperTrayforPage1" PrintDuplex = "DuplexMode"  
ColorMode = "Colormode_print">designfile.repx</DesignFile>
```

The `PrintCopies= "JobCopies"` attribute sets the number of copies created for the job using the associated design file e.g. `PrintCopies="2"` will create 2 copies of the print job.

⚠ This option works only if **Local Printer** is chosen as **Output driver** in the [Common](#) settings.

⚠ If multiple design files are used and the **Job merge mode** in the [Printer settings](#) is set to **"Append pages to the first job"**, this option can only be set for the first design file. The number of copies then will be applied to the whole print job consisting of all pages created by all design files.

If the `PrintCopies` attribute is not specified and the "Job Copies" parameter in the [Printer settings](#) is set to '0' it will be checked if a parameter with the name "PrintCopies" exists in the design file `designfile.repx`. If not the design file does not contain a "PrintCopies" parameter, the default setting from the printer driver will be used.

The `PaperSource = "PaperTray"` attribute sets the paper tray used as papersource for the print job created using the associated design file `designfile.repx`. If the paper for the first page should be drawn from a different tray additionally the attribute `PaperSource1 = "PaperTrayforPage1"` should be used. The valid values for `PaperTray` and `PaperTrayforPage1` depends on the used printer driver. Except for the "from printjob" entry the permitted values are the same as displayed in the **Paper Source** drop down list shown in the [Printer settings](#).

⚠ This option works only if **Local Printer** is chosen as **Output driver** in the [Common](#) settings.

If the `PaperSource` and/or `PaperSource1` attribute is not specified and the Paper source and/or Different paper source for first page option in the [Printer settings](#) is set to 'from printjob' it will be checked if parameter with the name "Papersource" or "Papersource1" exists in the design file `designfile.repx`. If not the design file does not contain such a parameter, the default setting from the printer driver will be used to select the papersource.

The `PrintDuplex= "DuplexMode"` attribute sets the duplex mode for the job using the associated design file `designfile.repx`.

Valid values for `DuplexMode` are:

Simplex	= only print on the front side of the sheet.
Horizontal	= print on both sides of the sheet, using long edge binding mode.
Vertical	= print on both sides of the sheet, using short edge binding mode.
Default	= using default setting of the printer driver.

⚠ This option works only if **Local Printer** is chosen as **Output driver** in the [Common](#) settings.

⚠ If multiple design files are used and the **Job merge mode** in the [Printer settings](#) is set to **"Append pages to the first job"**, this option can only be set for the first design file. The `DuplexMode` then will be applied to the whole print job consisting of all pages created by all design files.

If the `PrintDuplex` attribute is not specified and the "Duplex mode" option in the [Printer settings](#) is set to 'from printjob' it will be checked if a parameter with the name "PrintDuplex" exists in the design file `designfile.repx`. If not the design file does not contain a "PrintDuplex" parameter, the default setting from the printer driver will be used.

The `ColorMode = "Colormode_print"` attribute sets the coloring mode for output generated with **Local Printer** option set as **Output driver** in the [Common](#) settings.

Valid values for `Colormode_print` are:

Default = use default setting of the printer for generating grayscale or colored printouts.

Color = generate full colored printouts, if the printer have color capability.

GrayScale = generate grayscale printouts.

⚠ Colored documents only can be converted to grayscale, if the chosen printer driver supports the standard application interface for setting the color mode.

⚠ If the `Colormode_print` attribute is not specified and the "Color mode" option in the [Printer settings](#) is set to 'from printjob' it will be checked if a parameter with the name "ColorMode" exists in in the design file *designfile.repx*. If nor the design file does contain a "ColorMode" parameter, the default setting from the printer driver will be used.

Instead of specifying the names of the print options directly in the mapping file, it is also possible to specify XPATH expressions, which are used to retrieve the print options from variable XML data file nodes.

If you want to dynamically change the print options using data file content you have to use these attribute names:

```
<DesignFile PrintCopiesXPath= "XPathToJobCopies" PaperSourceXPath =
"XPathToPaperTray" PaperSource1XPath = "XPathToPaperTrayforPage1"
PrintDuplexXPath = "XPathToDuplexMode" ColorModeXPath =
"XPathToColormode_print">designfile.repx</DesignFile>
```

The rules for content of the data retrieved using the XPATH expressions

XPathToJobCopies, *XPathToPaperTray*,

XPathToPaperTrayforPage1, *XPathToDuplexMode* and

XPathToColormode_print are the same as if you would have specified these print options directly in the mapping file.

Example for additional <DesignFile> options:

```
...
<Mapping>
  ... one or more optional <Condition> nodes ...
  <DesignFile PrintDuplex = "Horizontal" PaperSource = "Tray
1">design\page1.repx</DesignFile>
  <DesignFile PaperSource = "Tray
2">design\main_report.repx</DesignFile>
  <DesignFile PaperSource = "Tray
3">design\summary.repx</DesignFile>
</Mapping>
...
```

In the example above, the print job created using the three design files *page1.repx*, *main_report.repx* and *summary.repx* will be printed on both sides of the paper, using long edge binding mode.

The pages created using the design file *page1.repx* will be drawn from the "Tray 1" papersource option of the printer chosen as **Printer** in the [Common](#) settings.

The pages created using the design file *main_report.repx* will be drawn from the "Tray 2" papersource option of the printer chosen as **Printer** in the [Common](#) settings.

The pages created using the design file *summary.repx* will be drawn from the "Tray 3" papersource option of the printer chosen as **Printer** in the [Common](#) settings.

Specific <DesignFile> attributes, if the Input data splitting option in the [Common](#) settings is used:

If you use input data splitting and want to use a designfile only for a specific split part you can use the `splitpart= "xmlsplitpart"` attribute of the <DesignFile> node. Valid values for `xmlsplitpart` are:

`first` = Use the designfile only at the first split part, e.g. for a banner page.
`1,2,3 ...n` = Use the designfile only for the split part with the specific number.
`last` = Use the designfile only at the last split part, e.g. for a summary page.

6.3.10 AFP to IPDS Emulation

6.3.10.1 Overview

This chapter describes all the parameters of the Mercury emulation AFP to IPDS. The data format AFP® (Advanced Function Presentation) is specified by IBM®.

AFP data streams will be converted into a internal IPDS (Intelligent Printer Data Stream) format.

That means, the generated format is optimized for the IPDS to PCL emulation. Currently there is no possibility to connect directly a IPDS printer at the output, because there is no communication module to negotiate the printer features and adapt the generated IPDS or perform any error recovery.

The AFP data stream may be a fully composed MO:DCA (Mixed Object Document Content Architecture for Presentation) or line data or mixed mode document print file.

This module utilizes following free libraries :

The **ICU** library (**I**nternational **C**omponents for **U**nicode), refer to <http://site.icu-project.org> .

The XML C++ wrapper **Arabica**, refer to <http://www.jezuk.co.uk> .

Some parts of the **Boost** C++ library, refer to <http://www.boost.org> .

6.3.10.2 Parameters

6.3.10.2.1 Common

Continue processing after exception code

Usually processing is aborted when any exception is reported from the AFP reader. Here can be specified for each bit of the exception code if a continuation is wished. The exception code consists of 8 bits which may be set in dependence of the encountered data stream violation. The bits are counted in IBM notation, i.e. bit **0** is the most significant bit and bit **n** (here bit 7) is the least significant bit. In case of exception, the emulation shows a warning message within the log window and reports the exception code and the individual bits as well.

NOTE: It is not guaranteed that the data is converted proper when any exception occur.

Job ticket detection

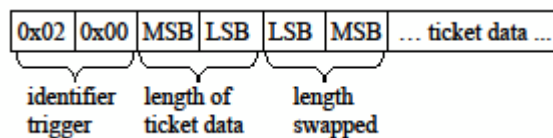
This option can be used for triggering a special job ticket at job begin, which may change or overwrite the current configured settings for the job. The complete job ticket may not be greater than 32767 bytes.

Off

No ticket scanning is done, i.e. the jobs are considered to have never a job ticket.

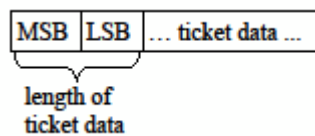
Auto detect

The beginning of the jobs is scanned for a special ticket identifier.
Format of the ticket which will be detected is:



Job has always a ticket

It is assumed that every job starts with a job ticket.
Format of the ticket must be:



NOTE: MSB = Most Significant Byte, LSB = Least Significant Byte.

'**ticket data**' must be EBCDIC encoded and should contain the '-o' pass-through command line options for InfoPrint Manager as partially described in the 'PSF/MVS Download Guide G544-5294-01' (refer to 'JCL Parameters and Corresponding AIX Printing Parameters' at page 9) and/or additionally to the IBM InfoPrint Manager documentation (also ACIF, line2afp, etc.).

Currently only following options are supported:

Option	Parameter	Description
-of=...	the form definition name to use	overwrites 'Resources->FORMDEF' edit field
-oformdef=...		overwrites 'Line Data->PAGEDEF' edit field
-opagedef=...	the page definition name	overwrites 'Line Data->CC' checkbox
-occ=...	'yes' or 'no'	overwrites 'Line Data->TRC' checkbox
-otrc=...	'yes' or 'no'	
-occtype=...	'a' (ANSI EBCDIC) or 'm' (IBM machine)	overwrites 'Line Data->Type' combobox
-ofileformat=...	'stream' or 'record' or 'record,n' (n is the size of the fixed block record)	overwrites 'Line Data->Data mode' combobox, n overwrites 'Line Data->Data mode->Length' edit field

6.3.10.2.2 Line Data

Record format

Here are the basic default settings for AFP line data processing.

NOTE: Some of these settings might be overwritten from a given page definition, either if a PAGEDEF name is given and the file exists within the configured resource path, or the print file contains a page definition.

Data mode

Determines the blocking format of the line data.

- **Stream** Streaming mode. Each record is separated with a line separator (**LS**), the format of **LS** depends on the given **Encoding**.
- **MSB/LSB Record** Variable length record. Each record contains within the first 2 bytes the length of the record.
- **Fixed Record** Fixed length record. Each record has the same size, the record length can be configured.

CC

When activated, process line data with Carriage Control characters of given type (either *ANSI EBCDIC* or *IBM Machine*).

TRC

When activated, process line data with Table Reference Character (font indexing).

NOTE:

When the PAGEDEF designates XMD processing, **Data mode** will be overwritten to **Stream** without line separators (**LS**).

In that case the settings of **CC** and **TRC** are also ignored.

For RCD processing, a possible **CC** within the record is ignored and **TRC**'s are not allowed (setting of **TRC** is ignored).

Only LND processing takes care of both, **CC** and **TRC** settings.

PAGEDEF file name

The given file name will be used as page definition. The page definition file must be in the binary MO:DCA format, i.e. the page definition must exist as Begin/End Page Map (BPM/EPM) objects.

The page definition search and usage priority is :

1. Within the AFP print file.
2. If not found, external under the given file name within the configured resource path.
3. If not found, usage of the *internal hard coded* page definition.

NOTE:

A PAGEDEF is only used for line data processing. For fully composed MO:DCA documents, a PAGEDEF is not necessary.

The *internal hard coded* page definition performs LND processing with following settings :

PRINTABLE AREA: 7.94 X 11.36 Inch

PRINT ROTATION: 0 DEGREES

91 LINES AT 8.2 LPI

POSITION 0 PELS 25 PELS

CHANNEL 1 assigned to LINE 1

Usage of the presentation device default font.

No 'Encoding Scheme ID' Triplet 0x50 given.

Shift Out/Shift In (SOSI) processing

Activates or deactivates SOSI text processing. Refer to IBM's Manual S544-3884-04 'Programming Guide and Line Data Reference' for details about SOSI processing.

Encoding

Determines the encoding of the input line data and also implicit the line separator (**LS**) character codes when processing LND or RCD line data in **Stream** mode.

- | | |
|-----------------|-----------------------------|
| • ASCII | LS = [0x0D] 0x0A |
| • EBCDIC | LS = 0x25 |
| • UTF-8 | LS = [0x0D] 0x0A |
| • UTF-16 | LS = [0x000D] 0x000A |

NOTE: If the current invoked Data Map (BDM) of a PAGEDEF (BPM) contains an 'Encoding Scheme ID' Triplet 0x50, this Triplet 0x50 overwrites the settings of **Encoding**.

6.3.10.2.3 Resources

Resource path

Determines the directory containing the external resources (in the binary MO:DCA format). If a resource can not be found within the AFP print file, it will be searched within this path.

The resource search priority is :

1. Within the AFP print file.
2. If not found, external within the configured resource path.

Use wildcards for file names, i.e. do not care about file extensions

When active, the resource file names of external resources will be searched **case insensitive** and they **may have any file extension with a dot**, otherwise the file name must match exactly the resource name (depending on the underlying file system).

For example, when the page segment resource PAGESEG1 will be searched external, the file PAGESEG1 or PAGESEG1.* will be used (where * means any sequence of characters).

Use resource mapping file

When activated, a resource mapping file 'afpcfg.xml' will be read and used.

The file search priority is :

1. Within the configured resource path.
2. If not found, within the global '<INSTDIR>\Res\AFP' path.

NOTE: There is a default installed file '<INSTDIR>\Res\AFP\afpcfg.xml', where <INSTDIR> means the installation directory of the software package.

The attribute '*name=...*' of the <map> tags is always the resource name to search for.

Currently, there is only the possibility to map **missing coded font resources**.

If a coded font resource is not found and the check box is activated, a proper entry of the mapping file will be used to determine the font character set and the codepage to be used.

The syntax of an <map> entry for coded font resources is:

Syntax: <map name="**coded font name**">**character set name, codepage name**</map>
XML hierarchy: /**afpcfg/maps/codedfonts**.

It is possible to use the wildarg character '?' for coded font and character set names.

e.g. <map name="X?H210AC">C?H200A0,T1V10500</map>

NOTE: There is no warranty for a proper layout and encoding if a mapped coded font entry is used. This is because the font size may vary and the encoding of the text within the AFP file using the coded font, must match the codepage.

Alternatively a coded font name can be remapped:

Syntax: <map name="**coded font name**">**new coded font name**</map>
e.g. <map name="X?S32?">X0S32?</map>

In this example, if a MCF refers to a coded font named 'X1S320' actually the BCF resource named 'X0S320' will be searched. A name 'X3S325' will result into searching a resource named 'X0S325'.

NOTE: Within that alternative format only 1 parameter is given without the ',' separator.

FORMDEF file name

The given file name will be used as form definition. The form definition file must be in the binary MO:DCA format, i.e. the form definition must exist as Begin/End Form Map (BFM/EFM) objects.

The form definition search and usage priority is :

1. Within the AFP print file.
2. If not found, external under the given file name within the configured resource path.
3. If not found, usage of the internal hard coded form definition.

Resource library path

Determines the directory containing the resources using a 'Resource Access Table' (RAT). The RAT is used to map a resource name specified in the MO:DCA data stream to information used to find and process the resource on a given system (e.g. the real file name of the resource). Following resources can be processed via a RAT:

- TrueType/OpenType fonts. The resource name is a full font name.
- 'Color Management Resources' (CMR's). The resource name is a CMR name.
- Data objects (e.g. TIFF, JPEG, GIF, etc.). The resource name is the object name.

The file name for the various RAT's is hard-coded, as follows:

- TrueType/OpenType font RAT: *'IBM_DataObjectFont.rat'*
- Color management resource RAT: *'AFP_ColorManagementResource.rat'*
- Data object RAT: *'AFP_DataObjectResource.rat'*

The resource search priority is :

1. Within the AFP print file.
2. If not found, the RAT will be read to get processing infos and locate the resource within the library.
3. If no RAT is found, the resource will be located under the referenced name within the library.

NOTE: Within the resource library path wildcards are never used, so the referenced name must exactly match (depending on the underlying file system), if there is no RAT given. The order of searching is the 1st library path and then the 2nd, if given.

Run emulation in static mode

This controls whether the emulation is unloaded between the incoming print jobs (dynamic mode) or not. If not (static mode), the emulation remains loaded for the entire life time of the print processes.

Allow job name modification

When activated, the job name of the system will be updated with the extracted document name of an encountered 'Begin Document (BDT)' structured field.

6.3.11 AFP to PCL Emulation

6.3.11.1 Overview

The AFP to PCL emulation of Mercury allows conversion of AFP data streams into PCL.

For an optimum emulation performance the AFP to PCL emulation in Mercury is split in two emulations:

- AFP to IPDS internal
- IPDS to PCL emulation

6.3.12 IGP to PCL Emulation

6.3.12.1 Overview

This chapter describes all the parameters of the Mercury emulation IGP to PCL. The IGP to PCL emulation from Mercury converts data streams created in the IGP[®] / PGL[®] (Intelligent Graphics Printing / Printronix[®] Graphics Language) for Printronix[®] line matrix printers, so that they can be printed on PCL compatible page printers. It should be compatible with the Printronix IGP-100/200/400 board using the Printronix Graphics Language (PGL).

6.3.12.2 Parameters

6.3.12.2.1 IGP

Special Function Control Code

The SFCC may be entered directly as character or as decimal code representation. Each edit field will be updated properly.

e.g. to set the default SFCC to '^', the decimal code 94 may be entered in the numeric field.

Symbol Set

Choose the proper symbol set for your datastream.

Printer Type

Select the proper printer driver which will affect the output of the emulation.

<u>Name</u>	<u>Orientation</u>	<u>Paper width</u>	<u>Default font</u>
HP-LASERJET II (300 DPI)	Portrait	A4	Courier normal
HP-LASERJET IIP LANDSCAPE (300 DPI)	Landscape	A4	Courier normal
HP-LASERJET IIIP (300 DPI)	Portrait	A4	Courier bold
HP-LASERJET IIIP LANDSCAPE (300 DPI)	Landscape	A4	Courier bold
PP2000 IGP LASER	Portrait	A4	Courier bold
HP-LASERJET IIIS (300 DPI)	Portrait	A4	Courier normal
PP4050 (300 DPI)	Portrait	A3	Courier bold
PP4050 LANDSCAPE (300 DPI)	Landscape	A3	Courier bold
PP4050 (300 DPI GOTH)	Portrait	A3	Letter Gothic
PP2000 IGP LATIN9	Portrait	A4	Courier bold

Power up mode

Defines the initial state of the emulation.

LISTEN The emulation interprets commands.

QUIET The emulation interprets no commands until a ~LISTEN is encountered.

Reset function

Defines the functionality of the ~RESET command.

IGNORE The ~RESET command will be ignored.

BOOT Same as *IGNORE*.

INIT All defined forms and logos will be deleted. The values for ~DENSITY, ~LPI and ~EXPAND will be set to default.

EURO character

If *ON*, the character code position 0xD5 (decimal 213) will contain the EURO character '€' (only for expanded ALPHA fields).

Zero style

Defines the shape of the zero character (only for expanded ALPHA fields).

- Slashed* The zero character will be stroked.
Unslashed The zero character will look like an 'O'.

Inverse area

Determines the manner how ALPHA text with the reverse printing option 'R' will be converted.

- Gray* The text will be printed in black on a gray shaded area.
Black The text will be printed in white on a black area.

Default formlength

Defines the formlength used when no length is given on form definition.

Use always

When active, this formlength overwrites always the given length on form definition.

6.3.12.2 Resources**Resource Path**

Determines the directory containing the resources or also where to save resources.

Run emulation in static mode

This controls whether the emulation is unloaded between the incoming print jobs (dynamic mode) or not. If not (static mode), the emulation remains loaded for the entire life time of the print processes.

6.3.13 IPDS to PCL5

see IPDS to PCL Properties inside chapter IPDS to PDF emulation.

6.3.14 IPDS to PDF**6.3.14.1 Overview**

The IPDS to PDF emulation of Mercury allows printers which support PDF to be connected to an IBM host system in the form of IPDS compatible printers.

IPDS (Intelligent Printer Data Stream) is a development of IBM, which stands for the current high demands on print quality. It allows output that is made up of various data types, for example text, matrix images, vector graphics and bar codes.

For an optimum emulation performance the IPDS to PDF emulation in Mercury is split in two emulations:

- IPDS to PCL emulation
- PCL to PDF emulation

This allows to set important PCL parameters, thus allowing more possibilities to optimize the emulation to individual conditions and environments.

The PCL5 to PDF emulation converts print data that were converted from IPDS to PCL so that they can be output on PDF compatible page printers.

6.3.14.2 Parameters

6.3.14.2.1 IPDS to PCL Emulation

Overview

The IPDS to PCL emulation of **Mercury** allows printers which support PCL to be connected to an IBM host system in the form of IPDS compatible printers.

IPDS (Intelligent Printer Data Stream) is a development of IBM, which stands for the current high demands on print quality. It allows output that is made up of various data types, for example text, matrix images, vector graphics and bar codes.

The IPDS emulation allows the Mercury to act in the capacity of an IPDS compatible printer with the following features:

- APA (All-Points-Addressable) prints for text, graphics, matrix images and bar codes at any point within the printable area of a page.
- Printing text in many variations by changing the character set and print direction within a page.
- The use of images and vector graphics to show line drawings, cake and bar graphs, technical drawings, logos, tables and signatures.
- Loading character sets.
- The use of overlays, in other words the storage of forms and letterheads, which are always printed with the same defined appearance.
- The use of page segments, in other words the storage of text, which is always printed in the same style as the remainder of the page.
- Printing of various types of bar codes.
- Printing with almost arbitrary page orientation and print directions.



The IPDS capabilities depend directly on the Host software version and its IPDS functionality.

Parameters

Printer Type

Select the printer type from the pull down menu.

These settings always represent a group of printers. For example, the 3812 stands for all printers that have a resolution of 240 x 240 dpi and use the simplex method. If you have a printer of this type and cannot find it among those in parameter 4, enter 3812. In the same way, 3916 stands for a 300 x 300 dpi duplex printer that is not on the list.

If the printer you use is not included on the list below, and the use of the appropriate type class does not produce the results you require, please contact your service partner.

Printer type	Advertised default resolution (dpi)	Plex type	Remarks
3812 (or types with 240 x 240 simplex)	240	Simplex	Advertises resident fonts
4028 (or types with 300 x 300 simplex)	300	Duplex	Advertises resident fonts
4224	144	Simplex	Use for line printer data (to replace dot matrix printers); Printer advertises resident font
3816 (or types with 240 x 240 duplex)	240	Duplex	Advertises resident fonts
3916 (or types with 300 x 300 duplex)	300	Duplex	Advertises resident fonts
3825-240	240	Duplex	Printer does not advertise resident fonts
3825-300	300	Duplex	Printer does not advertise resident fonts
3116	300	Duplex	
3827-240	240	Duplex	Same as 3825-240, only advertised ID differs
3827-300	300	Duplex	Same as 3825-300, only advertised ID differs
PRN4000	300	Duplex	Advertises extended color capabilities and outline download font capability
PRN4001	300	Duplex	Advertises extended color capabilities, FS4x and outline download font capability
3935	240/300/600 see remark	Duplex	Set default dpi manually; Can be used to replace compatible printer running in 3935 mode
4234	240	Simplex	Use for line printer data with simple graphics (to replace dot matrix printers, more features than 4224); Printer advertises resident fonts
PRN4100	600	Simplex continuous feed	Advertises "object container" support. Currently only used to capture special input trace files
LX-SIMM	300	Duplex	Advertises special resident font set similar to desktop printers using IPDS upgrade SIMMs

Default: PRN4000

Default dpi

Set the printer resolution (in dots per inch, dpi) for the selected IPDS printer type. Possible

resolutions are: 240, 300, 600 dpi.

Default: 'Printer default', that means default resolution of the selected IPDS printer type.

Variable dpi

If set, emulation advertises to the host that it can handle 240, 300 and 600 DPI simultaneously in addition to the printer model's default resolution.

Default: disabled

Plex mode

Printer default, Simplex, Duplex.

Default: 'Printer default'

ID number / Model number

Select an arbitrary ID and model number in variant to the selected Printer Type. If ID number and Model number are both set to 0 (default), the printer ID will not be changed by the emulation.

Default: 0 (ID number / Model number)

Example:

You would like to use features of the 3825/300 printer emulation. On your host system there is already a connection for an IBM 3830 model 1 configured. Enter the value '3380' as ID number and the value '1' as Model number. As soon as the host system sends a 'Sense type and model' request, Mercury will report a 3830 printer model 1 supporting features of the 3825/300 dpi emulation.

IOCA FS

This overrides the IOCA Function Set ID value reported to the host system during registration. Use only after consulting support.

Possible values include Printer default, 10, 11, 42 and 45.

Default: Printer default

Paper Type

Select the paper type. Possible values are

- Cut sheet
- Fanfold

If Fanfold is selected, IPDS reports the paper size with swapped Xm and Ym axes to the host system. This is necessary for laser printers with fanfold stock and simulates an inadequacy in IPDS of industrial standard printers.

Use this setting if you wish to print landscape jobs in portrait format or vice versa. The correct orientation of the axes is taken into account for the internal calculations.

Default: Cut sheet

Suppress Mark Form

Enable to suppress mark form.

The IPDS mark form command prints a fixed IPDS overlay from the host, marking the beginning and the end of a print job. This option suppresses the marking. A typical case for the use of the mark form command are fanfold printers.

Default: disabled

Use SMS

Enable to process the "Set Media Size (SMS)" IPDS command.

Normally, the SMS command is valid for fanfold impact printers only. With this option enabled, the physical media size value is read from the SMS command and sent as a "user defined paper size" PCL output command. This PCL information can then be used by the PCL-to-PostScript or PCL-to-PDF emulations.

Default: disabled

Output Bin Mapping

IPDS print jobs contain host identifications (ID) for the destination of the media, i.e. the output tray. PCL printers do only support output to trays 1 and 2 as a standard. The Output Bin Mapping option helps to assign other output trays to the media destination IDs. If the IPDS data stream does not contain any media destination command, the printer uses the PCL value indicated by the "Default" entry in the mapping table.

The "HostFinisher" entry in the mapping table is used if "use finishing" is indicated in the current IPDS group boundary (via XOH SGO and XOH DGB command extensions). If the "HostFinisher" mapping table entry is set to "disabled", normal IPDS media destination to PCL output bin mapping occurs, independent of the current group boundary's finishing option.

Code Page

Select the code page.

This parameter controls the use of the correct code page for the EBCDIC data stream from the host system if no other code page has been selected in the IPDS instructions.

Please keep in mind that there is no code page information sent by the host so that you have to make an assumption in order to find the best combination of IPDS input and PCL output code pages. The input ("EBCDIC") code page is set through the "IPDS printer / code page" parameter. The output ("ASCII") code page is set with the "[Conversion / Output Code Page](#)" parameter.

Select the appropriate code page from the following possible values for the correct conversion of umlauts and special symbols:

- 9 - Old Icelandic
- 7 - USA/Canada - CECP
- 8 - US/International ASCII
- 56 - Dp: International Set 5
- 59 - Symbols, Set 7
- 60 - Canadian French
- 73 - Germany F.R./Austria - CECP
- 74 - Belgium - CECP
- 75 - Brazil - CECP
- 276 - Canada (French) - 94
- 277 - Denmark/Norway - CECP
- 278 - Finland/Sweden - CECP
- 280 - Italy- CECP
- 281 - Japan (Latin) - CECP
- 282 - Portugal - CECP
- 283 - Spain
- 284 - Spain/Latin America - CECP
- 285 - United Kingdom - CECP
- 286 - Austria/Germany

- 287 - Denmark/Norway
- 288 - Finland/Sweden
- 289 - Spain
- 290 - Japan (Katakana)
- 293 - Apl (USA)
- 297 - France - CECP
- 310 - Graphic Escape Apl/Tn
- 340 - 4224 Resident Code Page
- 361 - International Set 5
- 363 - Symbols, Set 8
- 367 - ASCII
- 382 - Austria, Germany, Switzerland
- 383 - Belgium
- 384 - Brazil
- 385 - Canada (French)
- 386 - Denmark/Norway
- 387 - Sweden/Finland
- 388 - France, Switzerland
- 389 - Italy, Switzerland (Italian)
- 390 - Japan (Latin)
- 391 - Portugal
- 392 - Spain/Philippines
- 393 - Latin America (Spanish)
- 394 - U.K., Austral., Ire., H.K., N.Z.
- 395 - United States, Canada (English)
- 396 - Dw/370 Sonoran Serif
- 420 - Arabic Bilingual
- 423 - Greece - 183
- 424 - Israel (Hebrew)
- 437 - Personal Computer
- 500 - International #5
- 803 - Hebrew Character Set A
- 813 - Greece - ISO/ASCII 8-Bit
- 819 - Latin1 ISO/ANSI 8-Bit
- 829 - Math Symbols
- 831 - Portugal
- 836 - Peoples Republic Of China (Prc)
- 838 - Thailand
- 850 - PC Multilingual
- 851 - Greek PC
- 852 - Latin2 Multilingual PC
- 853 - Latin3 PC
- 855 - Cyrillic PC
- 856 - Hebrew PC
- 857 - Latin5 PC
- 858 - Multilingual Latin I (850 + Euro)
- 860 - Portugal PC
- 861 - Iceland PC
- 862 - Hebrew PC
- 863 - Canadian French - PC
- 864 - Arabic PC
- 865 - Nordic PC
- 866 - Cyrillic #2 PC
- 869 - Greece PC
- 870 - Latin2 Multilingual
- 871 - Iceland - CECP
- 874 - Thailand PC
- 875 - Greece
- 880 - Cyrillic Multilingual
- 890 - Yugoslavia
- 892 - OCR-A, 4234/3812, Palomar
- 893 - OCR-B, 4234/3812, Palomar
- 897 - Japan PC #1
- 899 - Symbols, Set 7 ASCII
- 903 - Peoples Republic Of China - PC
- 904 - Republic Of China (Roc) - PC
- 905 - Latin3 Multilingual

- 912 - Latin2 ISO/ANSI 8-Bit
- 914 - Latin4 ISO/ANSI 8-Bit
- 915 - Cyrillic ISO/ASCII 8-Bit
- 916 - Hebrew ISO/ASCII 8-Bit
- 920 - Latin5 ISO/ANSI 8-Bit
- 1002 - Dcf Rel 2 Compatibility
- 1003 - U.S. Text Subset
- 1004 - Ibm PC Desktop Publishing
- 1008 - Arabic ISO/ASCII 8-Bit
- 1025 - Cyrillic Multilingual
- 1026 - Latin5
- 1027 - Japanese (Latin) Extended
- 1028 - Hebrew Publishing
- 1029 - Arabic Extended ISO/Ascii 8-Bit
- 1038 - Symbols, Adobe ASCII
- 1039 - Gml List Symbols
- 1041 - Japanese Extended - PC
- 1042 - Simplified Chinese Extended - PC
- 1043 - Traditional Chinese Extended PC
- 1046 - Arabic Extended ISO/ASCII 8-Bit
- 1068 - Text With Numeric Spacing
- 1069 - Latin4 EBCDIC
- 1070 - Dp: United States
- 1071 - Dp: Germany/Austria
- 1072 - Dp: Belgium
- 1073 - Dp: Brazil
- 1074 - Dp: Denmark/Norway
- 1075 - Dp: Sweden/Finland
- 1076 - Dp: Italy
- 1077 - Dp: Japan (Latin)
- 1078 - Dp: Portugal
- 1079 - Dp: Latin America (Spanish)
- 1080 - Dp: United Kingdom
- 1081 - Dp: France
- 1087 - Symbols, Adobe
- 1091 - Symbol Set 7, Modified
- 1092 - Symbol Set 7, Modified - PC
- 1140 - U.S./Canada (37 + Euro)
- 1141 - Germany (273 + Euro)
- 1142 - Denmark/Norway (277 + Euro)
- 1143 - Finland/Sweden (278 + Euro)
- 1144 - Italy (280 + Euro)
- 1145 - Spain/Latin America (284 + Euro)
- 1146 - United Kingdom (285 + Euro)
- 1147 - France (297 + Euro)
- 1148 - International (500 + Euro)
- 1149 - Icelandic (871 + Euro)
- 1250 - Latin 2 Windows
- 1251 - Cyrillic Windows
- 1254 - Turkey Windows
- 1300 - Generic Bar Code/OCR-B
- 1301 - Postal Barcodes *Fg*
- 1302 - Postal Barcodes *Fg*
- 1303 - Postal Barcodes *Fg*
- 2001 - Keyboard 1 - U.S. Standard
- 2002 - Keyboard 2 - U.S. Standard
- 2003 - Keyboard 3 - U.S. Standard
- 2004 - Keyboard 4 - U.S. Standard
- 2005 - Keyboard 5 - U.S. Standard
- 2006 - Keyboard 6 - U.S. Legal
- 2007 - Keyboard 7 - U.S. Legal
- 2008 - Keyboard 8 - U.S. Legal
- 2009 - Keyboard 24 - Latin America
- 2010 - Keyboard 25 - Puerto Rico
- 2011 - Keyboard 26 - Germany
- 2012 - Keyboard 27 - Germany
- 2013 - Keyboard 28 - Germany/Austria

- 2014 - Keyboard 29 - Germany/Austria
- 2015 - Keyboard 31 - France/Belgium
- 2016 - Keyboard 32 - France/Belgium
- 2017 - Keyboard 33 - France/Belgium
- 2018 - Keyboard 37 - Canada (English)
- 2019 - Keyboard 38 - Canada (Bilingual)
- 2020 - Keyboard 39 - Canada (Bilingual)
- 2021 - Keyboard 41 - Italy (Standalone)
- 2022 - Keyboard 42 - Nether./So. Afr.
- 2023 - Keyboard 43 - Netherlands
- 2024 - Keyboard 44 - Spain
- 2025 - Keyboard 45 - Spain
- 2026 - Keyboard 46 - Italy
- 2027 - Keyboard 47 - Italy (Comm'Ns)
- 2028 - Keyboard 48 - Switzerland/France
- 2029 - Keyboard 49 - Switzerland/France
- 2030 - Keyboard 50 - Switzerland/Germany
- 2031 - Keyboard 52 - Sweden/Finland
- 2032 - Keyboard 53 - Sweden/Finland
- 2033 - Keyboard 54 - Norway
- 2034 - Keyboard 55 - Norway
- 2035 - Keyboard 60 - Brazil
- 2036 - Keyboard 61 - Brazil
- 2037 - Keyboard 63 - Portugal
- 2038 - Keyboard 66 - United Kingdom
- 2039 - Keyboard 67 - United Kingdom
- 2040 - Keyboard 68 - Japan
- 2041 - Keyboard 69 - Japan
- 2042 - Keyboard 100: U.S. (EBCDIC)
- 2043 - Keyboard 101: U.S. (EBCDIC)
- 2044 - Keyboard 102: U.S. (ASCII)
- 2045 - Keyboard 103: U.S. (ASCII)
- 2046 - Keyboard 254: U.K. (Special)
- 2047 - Keyboard 265: Germany/Austria
- 2048 - Keyboard 269: France/Belgium
- 2049 - Keyboard 277: Canada (Computer)
- 2050 - Keyboard 281: Norway (Computer)
- 2051 - Keyboard 281: Denmark (Computer)
- 2052 - Keyboard 285: Sweden/Finland
- 2053 - Keyboard 289: France/Belgium
- 2054 - Keyboard 313: United Kingdom
- 2055 - Austria/Germany F.R.- DABASE
- 2056 - Belgium/Swiss/Luxembourg-DBBASE
- 2057 - Denmark/Norway/Iceland - DDBASE
- 2058 - Finland/Sweden - DEBASE
- 2059 - France - DFBASE
- 2060 - Italy - DIBASE
- 2061 - Netherlands/Portugal - DNBASE
- 2062 - Spain - Dsbase
- 2063 - Dcf Rel 2 Compatibility
- 2064 - Dp: Canada (French)
- 2065 - International Set 5
- 2066 - Pi Fonts
- 2067 - Austria, Germany, Switzerland
- 2068 - Belgium
- 2069 - Brazil
- 2070 - Canada (French)
- 2071 - Denmark/Norway
- 2072 - Sweden/Finland
- 2073 - France, Luxembourg, Switzerland
- 2074 - Italy, Switzerland (Italian)
- 2075 - Japan (Latin)
- 2076 - Portugal
- 2077 - Spain/Philippines
- 2078 - U.K., Austral., Ire., H.K., N.Z.
- 2079 - United States, Canada (English)
- 2080 - Math Format

- 2081 - Sonoran Display Fonts
- 2082 - Sonoran Petite Font
- 2083 - Lcs 3800-1 Dump Character Set
- 2084 - Lcs 3800-1 Gothic Folded
- 2085 - Dcf Gp12 Compatibility
- 2086 - Lcs 3800-1 Gothic/OCR-B
- 2087 - Lcs 3800-1 Gothic/OCR-A
- 2088 - Lcs 3800-1 Gothic/OCR-A
- 2089 - Lcs 3800-1 Gothic/OCR-A
- 2090 - Lcs 3800-1 Gothic
- 2091 - 6670 Symbol Set (Os/6)
- 2092 - 6670 Symbol Set (12-Pitch)
- 2093 - 6670 Symbol Set (10-Pitch)
- 2094 - Lcs 3800-1 Format Characters
- 2095 - Lcs 3800-1 Gothic/Katakana
- 2096 - Lcs 3800-1 Gothic
- 2097 - Lcs 3800-1 Gothic
- 2098 - Lcs 3800-1 Gothic
- 2099 - Lcs 3800-1 Text 1 & 2
- 2100 - Lcs 3800-1 Gothic
- 2101 - Lcs 3800-1 Gothic
- 2102 - Lcs 3800-1 Gothic/Katakana
- 2103 - Lcs 3800-1 Gothic/Katakana
- 2104 - Lcs 3800-1 Gothic
- 2105 - Lcs 3800-1 Text 1 & 2
- 2116 - United Kingdom - DUBASE
- 2117 - Keyboard 253: U.K. (Special)

These parameters define the position of the output on the physical page. This is done by moving it in the appropriate direction (X or Y direction).

Tray

Select the paper tray number for which you the Tray Alignment settings shall apply.

Default: tray number 0

Unit

Select the unit for the tray alignment settings.

Offset Parameters are designed to move the output on the page. Positive values move the output away from the origin i.e. left top. Negative values move it closer to the origin. Possible settings are:

- 1/1440 inch
- inch
- cm

Default: 1/1440 inch

Whole Page Offsets

Frontside / Duplex back side

These parameters are moving the page as a whole. For front and back sides (for duplex printing) of a sheet different values can be chosen. X refers to horizontal, Y refers to vertical movements, in reference to page in portrait printing mode.

Default: 0 (all)

Print Direction Offsets

Portrait / Landscape / Reverse Portrait / Reverse Landscape

These parameters are moving output of specific print directions on a page relatively; it is a feature which is used for special paper formats in general. X and Y values are specified for specific print directions.

Example: If you would like to move output for Reverse Landscape only, hold the page in front of you that you are able to read the text. If you want to move the lines down, enter a positive Y value.

Default: 0 (all)

Scaling**Zoom**

Changes the size of the IPDS data being imaged on the page. Values greater than 100% enlarge the page, values smaller than 100% reduce the size of the page. This feature is especially useful to make line data printouts fit on a non-edge to edge printer.

Example: set Zoom to 95% to make an A4 edge-to-edge printout fit on a standard PCL5 printer without edge-to-edge capability.

Default value: 100%, meaning no change

Page alignment

If the IPDS logical page is smaller than the paper being used, usually the logical page will start at the top left corner of the paper sheet. However, sometimes it is useful to align the logical page to another corner or to the center of the paper, e. g. when "Zoom" is active.

Possible values:

- "Top Left"
- "Center"
- "Top Right"
- "Bottom Left"
- "Bottom Right"
- "Auto LPD"
- "Auto LPP"

Default: "Top Left"

"Auto LPD" and "Auto LPP" help solving compatibility issues, especially with legacy impact printer jobs. If one of these values is selected, the alignment will be based on the page orientation parameter found in the LPD or LPP. These mappings are used:

LPD / LPP orientation	Alignment selected
Portrait	Top Left
Reverse Landscape	Top Right
Reverse Portrait	Bottom Right
Landscape	Bottom Left

Job separation and finisher commands

Select job separation and finishing commands from the pull down menu.

In IPDS data streams the command 'Alternate Offset Stacker' (AOS) can be used to place

printed pages with an offset in the printer's output bin. The 'JOG OUTPUT' command can be inserted to the FORMDEF file on the host system to add the AOS command to IPDS data. This way it is possible to separate different print jobs. Other settings in Job Management as 'Group Boundary' detection can be used, too.


The IPDS to PCL5 emulation supports adding of necessary PCL or PDL commands for specific printer types. In Job separation and finisher commands can be configured which commands should be sent to the printer.

Possible commands are:

- **PCL Job Separation:** results in a PCL command for job separation. The sequence 'ESC&l1T' will be sent to the printer. This is applicable for printers which do not support PDL.
- **PCL Reset:** results in a PCL reset command, i.e. the sequence 'ESC E'. Fonts and macros sent to the PCL printer before will still be available as they are set to permanent after loading to the printer. This is also applicable for printers which do not support PDL.
- **PDL Standard:** will send a sequence of 'PCL Universal Exit Language', 'PDL Job End', 'Enter Language PCL' and 'PCL Reset' to the printer. This is useful for printers which automatically select their emulation between Postscript, PCL and other printer languages. Sending this sequence will assure a proper start of the PCL interpreter of the printer (default).
- **Xerox Nxx Series Finisher**
- **Xerox DP/DT Series Finisher**
- **Xerox DC Series Finisher**
- **DSC Finisher**
- **Aficio series**
- **Typhoon series**

Using these options allows control of specific accessories of these printer families. In detail the accessories types are:

- Jog (Shifted output)
- Staple (Stapling)
- Punch (Punching)
- Collate (Collating of print jobs in specified order (see printer's manual))
- Slip Sheets (Inserting divisional sheets)

 Some printer models do not support all combinations of these accessories. Also please keep in mind that adjustment of the ID numbers of output bins may be necessary (see IPDS Printer / Output Bin Mapping).

If Xerox DP/DT Series Finisher is selected, a 'Xerox Job Ticket' will be sent to the printer instead of PDL commands.

If DSC Finisher is selected, and if an IPDS FORM name is found, this information can be forwarded to the printer. If set to generate Operator Message, an operator message command is heading to the print job. The job has to be released manually on the printer console so the operator is able to load the printer with the appropriate paper stock.

If set to set Document Media name, an '%%DocumentMedia' command is added at start of the print job requesting a paper of type FORMipdsname from the printer's paper catalog.

Example: If the Host form name is INVOICE, a paper named FORMINVOICE has to be loaded. A 'paper mismatch' will be issued by the printer, if the wrong paper was loaded.

With all configurations the IPDS job name will be sent to the printer using the PDL job command. With Xerox Nxx and DSC finisher 'Full Page' mode will be activated in addition.

- **User Defined:** supports definition of an arbitrary byte sequence. Characters that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code. E.g. to enter the ASCII character <ESC>, type '%1b'. The '%' character itself is entered as '%%'. In addition the job name can be added to this sequence by using the place holder '%s'.

Example: To send the PCL 'Universal Exit Language' Sequence '<ESC>%-12345X' the input should be '%1b%%-12345X'.

The previously shown PCL sequence can be extended by additional PCL commands. i.g. '@PJL SET EDGETOEDGE=ON%0D%0A' to turn on 'Full Page' mode as long as they are supported by the printer.

- **PJL for PS and PDF emulation:** will send a sequence of 'PCL Universal Exit Language', 'PJL Job End', 'Enter Language PCL' and 'PCL Reset' to the printer. In addition to that, an optimized, but non-standard stream of PJL and PCL commands will be generated which can be processed by the PCL-to-PDF, PCL-to-PostScript and PCL-to-GDI emulations. Compressed images and outline fonts will be passed on without modification if applicable. Less output data will be produced and data processing will be faster. See also "[Miscellaneous / PS/PDF font passthrough](#)".

Default: PJL Standard

Media Type Hole Puncher Stack Output with offset

These are additional job finishing and processing options. Select as necessary. Depending on the options installed with the printer, some of these features may not be supported by the printer.

Default: Off

Face down printing

Select **Face up** or **Face down** printing.

Default: Face up

Page order

Select **Page 1 - n** to print the job from the first to the last page.
Select **Page n - 1** to print the job from the last to the first page.

Default: Page n - 1

AFP Job copies

If the IPDS emulation is combined with the AFP emulation, a "job copies" command can be forwarded to the IPDS emulation.

This parameter determines how the "job copies" number is passed on to the printer. The IPDS emulation does not generate job based copies by itself.

Select **"NPS Job ticket"** to pass on job copy information via the internal job ticket. Only works for combined emulations.

Select **"PJL QTY"** to pass on the number of job copies via a PJL job header command.

Default: Off

Misc. PCL Features

- **Send IPDS variable spaces as PCL positioning command**

Disable to translate IPDS spaces (EBCDIC \$40) with attribute 'variable width' to ASCII spaces (\$20) with standard width of the currently selected font. Enable to use PCL cursor positioning.

⚠ Disabling this option will result in more performance but less precision with output.

Default: enabled

- **Early AOS Trigger**

Enable to make the IPDS to PCL5 emulation process the AOS command before the beginning of page command.

To perform job separation as special PCL function successfully, the AOS command in IPDS data must be sent by the host system at the right position i.e. between two pages. Some host applications only enable adding the AOS command to the data stream just behind the beginning of a page as the earliest possibility.

Default: enabled

- **Suppress PCL 'Flush all pages' command**

At the beginning or at the end of an IPDS print job a 'Print Buffered Data' command can be sent by the host system. This command results in a PCL 'Flush All Pages' ('ESC &r1F') command. With some PCL printers using this command will lead to significant performance reduction. Using this option allows to turn off usage of this command.

Default: Off

Color Support

Choose the PCL color support mode.

- | | |
|---------------------------|---|
| Map color to black | non-white text and bitmap colors will be mapped to black and GOCA fill colors will be substituted with gray shading patterns. |
| Map color to gray | the IPDS emulation translates OCA and "extended colors" into corresponding shades of gray. However, IOCA FS42 and FS45 images cannot be converted to gray yet. If you want to convert these image types, use the IPDS-to-PostScript or IPDS-to-PDF emulation, and select "PCL5c color" and "Printer Control / PJI for PS and PDF emulation". Then those images will be converted by the output device's PostScript / PDF RIP. |
| PCL5c color | the IPDS emulation will generate PCL 5 'simple color mode' commands to show colored IPDS text, IOCA bitmap graphics, and GOCA shaded area fills on the printer. IOCA FS42 and FS45 images will be sent with PCL5c full color commands. |

PCL "Page Eject" Command

- **FF - form feed (print all pages)**
- **DPSS - duplex page side selection (suppress empty pages)**

This parameter controls which PCL instruction is sent by the IPDS to PCL5 emulation to eject a page at the printer. Since some older printers do not support the instruction "duplex page side selection" (ESC &a#G), the classical "form feed" (0x0C) can be used as an alternative.

In addition to that, using the "duplex page side selection" command suppresses empty pages on PCL printers and Mercury PCL-to-xxx emulations.

Default: DPSS - duplex page side selection (suppress empty pages)

Area Fill Method

Select the area fill method from the pull-down menu.

Possible values are:

- Bitmap, no conversion 240->300 dpi
- PCL Pattern, old header type
- PCL Pattern, new header type
- PCL shading and rectangles (Default)
- Bitmap, conversion 240->300 dpi
- Solid color

To fill an area with a gray scale pattern, an IPDS host system sends this pattern to the printer. PCL printers, however, do not support this instruction, which means that it has to be simulated by the emulation.

[Bitmap, no conversion 240->300 dpi](#) uses the pattern without change and without scaling from 240 to 300 dpi. This method represents a compromise between speed and accuracy. Moiré effects may occur in the output.

[PCL Pattern, old header type](#) and [PCL Pattern, new header type](#) are faster since a freely definable PCL character is used to generate the pattern. At the edges, however, overlaps and flaw areas may occur depending on the printer being used.

[PCL shading and rectangles](#) uses the PCL command 'Fill Rectangle' to fill the area. This command is only available on fairly recent printers and is the fastest method. When you output a large number of filled areas that are next to each other, however, 'banding' may occur (printing in several stripes) since PCL only supports a maximum of eight gray scales for this command.

[Bitmap, conversion 240->300 dpi](#) is similar to the [Bitmap, no conversion 240->300 dpi](#) method, but the pattern is scaled to 300 dpi. This method is the slowest but also the most accurate.

[Solid Color](#) is similar to "PCL shading and rectangles", however uses a rectangle with a solid color to fill the pattern area. The color is modulated by the original pattern's intensity. The rectangle is opaque, but uses very little output data. This mode is best for creating PDF output if unstructured patterns are used.

Logical Page Processing

determines the way the emulation arranges IPDS logical pages on the output medium.

- **Collect logical pages (default)**

All pages necessary for a sheet of paper are collected within the emulation's memory. The pages are then rearranged, rotated and shifted to match the host system's (PSF) request. This mode has full support for "enhanced N-UP" and page shift and rotation within the "Logical Page Position (LPP)" IPDS command (full AFCCU feature support).

Also, this mode does not require additional memory in the printer to store the temporary pages in PCL macros.

Use this mode for optimal N-UP print results.

- **Stream logical pages**

No pages are buffered, neither in the printer nor in the emulation. This mode has the least memory requirements, but N-UP is not supported. For compatibility to older Mercury installations (Version 3.xx).

- **Enhanced N-UP via PCL macros**

Support for Enhanced N-UP is done by packing the logical page partitions into PCL printer macros and sorting the pages in the printer. For this mode, you might need to upgrade the printer's memory. Not all shift and rotation operations are supported. For compatibility of Mercury installations before Version 4.24.

Bar Code Adjustment


To fine-tune bar code output you may adjust the bars and gaps individually by single pixels. Add or subtract pixels to the single, double, triple and quadruple width elements. Use e.g. -2 to make a gap bar 2 pixels thinner.

Default: all values set to 0

Bit Images PCL Compressed

Enable/Disable PCL compression of bit images.

This parameter controls the way in which the IPDS to PCL5 emulation sends PCL matrix graphics to the printer.

 If disabled, the emulation process is faster but more data are sent to the printer which might slow down output.

Default: enabled

Copies By Emulation

Enable to generate copies through the emulation.

Some printers do not support the PCL command 'Number Of Copies', or they do not print the right number of copies. If this option is active, the IPDS to PCL5 Emulation will generate copies by sending the job data to the printer several times.

Default: disabled

Load IPDS overlay as PCL macro

Enable to load IPDS overlay as PCL macro.

If this option is enabled, IPDS overlays are saved in PCL macros to reduce the transfer of the overlays to every page. This speeds up processing but may cause data loss in printers with not enough memory.

Default: enabled

Use optimized macros

Enable to use several optimizations for loading IPDS overlays as PCL macros.

These optimizations work best with the Mercury *PCL-to-PDF*, *PCL-to-PostScript* and *PCL-to-GDI* emulations.

Not all physical PCL printers support these optimizations.

If this option is enabled, these changes are made to the PCL output data stream:

1. Nested IPDS overlays are saved in nested PCL macros.
2. Images in IPDS page segments are converted into PCL macros
3. PCL macros will be relocated using the PCL "offset registration" command, if possible
4. Images in overlays will be tagged with a "Unique resource ID (URID)" which can be recognized by the Mercury PCL-to-xxx emulations.

This option is usable only if "*Load IPDS overlay as PCL macro*" is selected.

Default: disabled

Use Unit Of Measurement

Enable to use unit of measurement.

With this command you can define whether the conversion of IPDS units into printer units should be completed by the emulation or in the printer. If your printer supports this command, you should activate the parameter.

This will increase the data throughput of the emulation since in this case the coordinates are transformed in the printer and not by the emulation. This saves computer time.

Default: disabled

PCL offset registration fix

Some PCL printers are not fully compatible to HP's specification of the 'offset registration' <ESC>&I#U PCL command. This is a workaround for such printers.

Force Duplex Printing

Enable to force duplex printing.

This is used to define whether print jobs are printed duplex (on both sides) or not. This assumes that the PCL printer you wish to use is capable of this.

Default: disabled

Swap Duplex sides

Enable to swap duplex front and back sides.

This is a workaround for printers which do not support "face down" printing to reverse a print job.

This option only works if both "*Force Duplex Printing*" and "*Logical Page Processing / Collect Logical Pages*" are selected.

Default: disabled

Configure paper handling and installed paper sizes of the IPDS to PCL5 Emulation with these parameters. All the values configured here must match with the values configured on the printer.

IPDS Tray

Select the paper input tray of the real printer for the IPDS to PCL5 emulation.

Up to eight trays are supported.

IPDS Tray #0 is the first tray shown. To change settings for other IPDS tray, change this number to select the tray you want.

Accessible

Enable/Disable if tray is accessible.

By activating this parameter, the selected tray is made accessible for the emulation.

Default: disabled

Identification

Enter a character string to identify the tray. This character string may have a maximum length of 10 characters.

Default: MAIN

MID (Media Source ID)

For the selected tray, this is the ID number that is reported back to the host as the "Media Source ID" in the XOH-OPC "Printable Area Self-Defining Field" reply. Usually, the IDs are ascending numbers starting at zero. Manual feed trays have common values of 99 or 100. However, sometimes it is useful to override these default values.

Please make sure that no duplicate Media Source IDs are entered.

Valid range: 0 to 255

Default: equal to IPDS Tray number or 99 for last (manual feed) tray

Paper size ID

Enter paper size ID.

Using this option allows control of sending an ID defined by PCL for paper size in addition to PCL paper tray selection commands (see PCL Output / Input Tray Selection). This allows usage of special paper formats with the printer.

Examples:

These IDs are predefined for all PCL printers, some manufacturers are using other IDs in addition:

1	Executive (7,25 * 10,5 Inch)
2	Letter (8,5 * 11 Inch)
3	Legal (8,5 * 14 Inch)
6	Ledger (11 * 17 Inch)
26	A4 (210 * 297 mm)
27	A3 (297 * 420 mm)
80	Monarch (3,875 * 7,5 Inch)
81	Com-10 (4,125 * 9,5 Inch)
90	DL (110 * 220 mm)
91	C5 (162 * 229 mm)
100	B5 (176 * 250 mm)

Default: 0, no additional ID will be sent.

Tip: With Kentek K40 printers, 'Full Page' mode can be activated using ID 1026 for A4 sized paper.

Std. Input Tray / Duplex Input Tray

Adjust input tray mapping to printer.

The host system that sends IPDS data can request the printer with a 'SIMS' command to select its paper source from various trays. The numbers of the trays are mapped to the corresponding trays of the PCL printer. Since, however, there are various designs of PCL printers, this gives you an opportunity to tailor this image to your specific printer.

There are two trays you can have the paper pulled from. "Standard input tray" selects the paper tray for simplex jobs and simplex pages in mixed plex jobs. "Duplex input tray" selects the tray for duplex pages. This is useful if your printer pulls in paper in different positions for simplex and duplex (e.g. simplex has to be face-down and duplex face-up).

The "Automatic (ID 0)" tray selection discards the tray number information from the output and lets the printer choose the paper tray depending solely on the "paper size" information.

Default: IPDS tray #0 maps to printer tray #1, IPDS tray #1 maps to printer tray #2 and so on.

Rotate

Define the rotation of all pages sent to the selected paper tray.

This might be useful if the printer allows feeding of large paper formats in one orientation only. Possible settings are:

- 90°
- 180°
- 270°

Default: 0° (no rotation)

Paper Size

Select the installed paper size for the selected tray.

The pre-defined paper sizes A4, Letter, Legal and user-defined values are supported.

Unit

Select the unit for the values in Width and Height for user defined paper sizes. Possible settings are:

- 1/1440 inch
- inch
- cm

Width

Define the width of the user defined paper size. Enter the value for the width in the unit set in Unit.

This option is only accessible if paper size is set to 'User defined'.

Height

Define the height of the user defined paper size. Enter the value for the height in the unit set in Unit.

This option is only accessible if paper size is set to 'User defined'.

Wide carriage/LEF

Default: disabled

Ignore Page Offset


Enable to ignore page offset.

When printing formatted lists ('Fixed Size Records' or 'AFP Line Printer' streams) that are not reformatted using PSF on host side, it might be necessary to print to the largest possible area on the paper. With this option a 'Page Offset' command sent by the IPDS host system can be disabled.

Default: disabled

Media attributes

Depending on the printer type, there can be an additional selection of 'Media Attributes' like 'Paper Type' and 'Media Weight' which can be assigned to the selected IPDS Tray number.

 If you additionally set the 'Std./Duplex input tray' parameters to 'Automatic (ID 0)' for all activated 'IPDS Trays', you can make the printer choose the print medium automatically, based on the media type.

Fonts

• Algorithm

Select the font algorithm.

On printers with recent font headers, which support a resolution of 600 dpi for download fonts, instead of a conversion of the 240 dpi host fonts into a 300 dpi resolution, it is also possible to select a conversion to 600 dpi. This improves the quality of the output, but adversely affects speed as a result of the considerably greater volume of data.

The PCL5 Technical Reference Manual recommends the use of the recent font header. On printers that only support the older font headers, the format of the header can be set using this parameter. The possible settings are:

- 300 dpi Old font header 240->300 scaling alg 1
- 300 dpi New font header 240->300 scaling alg 1
- 600 dpi New font header 240->600 scaling alg 2 (Default)
- 300 dpi New font header 240->300 scaling alg 3
- 300 dpi New font header 240->300 scaling alg 4
- 600 dpi New font header 240->600 scaling alg 5

• Outline Font DPI

IPDS outline (vector) fonts are rendered into bitmap fonts by the emulation. This value specifies the resolution for that bitmap font. Remember: The higher the resolution, the better the print quality and the larger the amount of data generated.

Off means, the Outline fonts are replaced by a Times Roman font of similar size.

• High precision positioning

Some font characters may have character increment values which are non-integer multiples of a pixel width, e. g. half- or quarter-pixel advancement values. Some PCL printers can only do full pixel character advancements and drop the fractional part of the advancement (aka "delta x") value.

If activated, the IPDS emulation will send additional PCL positioning commands between the characters in a word if the difference between accumulated fractional values and truncated pixel widths exceeds one pixel.

This feature can compensate character increment inaccuracies of some PCL printers.

Drawbacks:

- additional positioning commands increase the amount of PCL data sent to the printer
- NPS filters might not be able to find PCL trigger text words because the additional positioning commands might split a word

Default: off

• **Output CP**

For SBCS (Single Byte Character Set) type Bitmap fonts only, this sets the Code Page that is used for the translation of the incoming IPDS bitmap font into the PCL output symbol set.

This feature is used to create "ASCII" output that can be processed by other elements of [Mercury](#) like the "Field Scanner" filter or the "Overlay Manager". In addition to that, it is necessary to create searchable PDF files with the IPDS-to-PDF or PCL-to-PDF emulation.

Please keep in mind that there is no code page information sent by the host so that you have to make an assumption in order to find the best combination of IPDS input and PCL output code pages. The input ("EBCDIC") code page is set through the "IPDS printer / code page" parameter. The output ("ASCII") code page is set with the "Conversion / Output CP" parameter.

Example: for EBCDIC to German ASCII translation use IPDS printer code page "1141" and Output CP "858"

Default: 0 - no translation

see also: [IPDS Printer / Input Code Page](#)

Max. output image dpi

Limits the PCL image resolution for printers that cannot handle resolutions higher than the selected value. The default value 'off' means no limitation is active.

Default: off

Settings

• **Swap N-UP back sides**

Swap logical page partitions on N-UP duplex back sides.

With this option set, compatibility to some older Non-IBM IPDS printers and Mercury up to Version 4.34 can be achieved. The arrangement of logical pages on N-UP back sides with SMO 1 or 3 will be changed between clockwise and counter-clockwise positioning of the partitions.

Default: disabled.

• **Debug Level**

Set the debug level.

Controls how many additional messages the IPDS to PCL5 emulation writes to the Dispatcher log file, when 'Debug' message Level is enabled.

Possible values are 0 to 3.



Use only while doing traces. Contact support service to learn when to set this parameter. This parameter affects performance.

Default: 0 (disabled)

- **PCL Full Page**

The IPDS to PCL5 emulation assumes that the attached PCL printer is able to print to the whole sheet i.e. a so called 'Full Page' mode or 'edge to edge' printing is enabled. Some printers do not support this mode. In this case the unprintable area can be reached using special methods only.

If PCL Full Page set to off, the IPDS to PCL5 emulation will try to reach unprintable area using other positioning methods.

Default: enabled

- **Restricted HP-GL Output**

Enable to avoid HP-GL 'Rotate' and PCL 'picture frame' commands.

With this option set, the IPDS to PCL5 emulation will not generate HP-GL 'Rotate' and PCL 'picture frame' commands. Use this switch, when IPDS output is passed to an PCL to Metacode emulation.

Default: disabled

- **Duplex To Simplex**

This parameter should be used when four conditions meet:

1. The printer connected to the [Mercury](#) system is not capable of 'Duplex' printing
2. The 'Input Paper Size' is set with 'No edge to edge'
3. The print job is 'duplex'
4. The printer does not handle the PCL 'Page offset registration' command correctly when printing a page flagged as 'Duplex backside'

In short, this is a workaround for a bug in some printers' PCL interpreters. Use only when horizontal positioning is scrambled on every second page of the print job. It usually is a better idea to generate the print job on the host system as 'simplex' if the attached printer is capable of printing 'simplex' only.

Default: disabled

Resource Path

See section [IPDS Overlay Supplement](#).

Host Communication

This setting changes the way the IPDS to PCL5 emulation reports initial status (first IPDS ACK-Reply) to the host when the emulation process is started. Keep in mind that the 'Loading' parameter controls how often the IPDS emulation gets loaded per Logical Printer. If it is set to Static Mode the first IPDS ACK will be returned to the host only once at the beginning of the very first print job.

When the host system receives such a 'Normal Printer Restart (NPR)'-ACK-Reply it considers the printer to be in a state directly after power up, i.e. it has no fonts and resources loaded and the host will re-send all necessary resources to the emulated IPDS printer. This enlarges the print job, but can be necessary under certain system environments. Possible settings are:

- **Standard:** no additional ACK is generated, i.e. the 'Normal Printer Restart'-ACK-Reply is generated by the channel device driver, if applicable (default).

- **Normal Printer Restart before job:** the IPDS to PCL5 Emulation will generate one ACK-Reply of this kind at the beginning of the print job.
- **Normal Printer Restart after job:** the IPDS to PCL5 Emulation will generate one ACK-Reply of this kind after the print job, generating no duplicate NPR-ACKs when the transport protocol driver (e.g. channel card) also generates an NPR-ACK.
- **Simulate printer offline:** the first ACK-reply sent to the host says that the simulated printer is offline. This is made so that a host operator can start spooling print jobs into a host printer queue without these jobs being printed immediately. Some hosts require the printer to be present but 'offline' to enable spooled print jobs. To print the jobs later on, you have to set the 'Host Communication'-Parameter back to its previous value.

Default: Standard



Do not use this option with a File Input Port.

Emulation Loading

Set for which job the emulation shall send resources to the printer. Possible settings are:

- **Channel mode (static):** The IPDS to PCL5 emulation sends resources to the printer for the first job only.
- **Standard mode:** The IPDS to PCL5 emulation sends resources to the printer for every job that is received.

Default: Standard mode

Out of paper host message

- **Generate NACK**
- **Generate SENSE**

Depending on the Host system OS, communicating the 'out of paper' printer status needs a 'SENSE' or 'NACK' type packet to be sent back to the host. Original IBM systems require both, some compatible systems need a 'SENSE' only. If you switch off both message types, no 'out of paper' printer status will be sent.

Use Committed page counter

This switch affects the page counters which are reported back to the host system in an ACK/NACK reply. It only applies if 'page confirmation' is enabled in the output port's 'Job Monitoring' tab.

If set, the 'committed' page counter will be incremented in sync with 'received' page counter. If not, the 'committed' counter will be incremented after the page has been confirmed as 'printed' by the printer, which is required by some non-IBM Host OSes.

Max NACKs

If set to a non-zero value, the emulation will abort the print job when the number of NACKs generated exceeds the value set in this parameter. Useful for some hosts which cannot tolerate a large number of NACKs sent in rapid sequence.

Default: 0

Job Name

An IPDS Host is able to send a job name or job identification within the print job data. If such an information is present, Mercury can extract that information and pass it on to its own spooling system.

• Extract job name from Group Boundary command

Enable/Disable extracting job name from Group Boundary command.

Usually, the job name gets sent with the 'Group Boundary' IPDS command. If this parameter is enabled, the IPDS interpreter will use that information.

Default: enabled

• Add form name to job name

Enable to add form name to job name. Extract job name from Group Boundary command must be enabled.

If the job name is extracted from a 'Group Boundary' IPDS command, it is built using job name, job number and user name as given by the host system. This option will also add the form name given by the host system to this job name.

Default: disabled

• Add MVS job class / OS400 spool file name

If connected to an MVS type host, IPDS is able to extract the job class ID out of the IPDS 'Group Boundary' command. When this option is enabled, the job class ID is added to the job's name.

Default: disabled

• Build

Use a text string with variable names to build a customized job name from the "define group boundary" command's contents. Depending on the host system type, not all variables are available. A customized job name can be useful if you want to dispatch the emulation's PCL output depending on the job name. So you could forward the job to different printer types with the host user name or the host form type as the parameter. Or you could use specific names as triggers for the overlay manager filter.

- *Variable names* need to be written in angle brackets, e. g. `<gb_username>`.
- *Blanks* in the variable's content are replaced with underbar characters to avoid whitespace in the job name.
- A variable's content can be *truncated* to a certain length by adding a number in square brackets e. g. `<gb_username[5]>` would cut off the user name after 5 characters.
- EBCDIC variable contents will automatically be translated to ASCII.

Example:

```
<gb_formsname[6]>_<gb_username>_<gb_spoolfile>_<gb_spoolfilenumber>
```

could result in a job name like this:

```
*STD___JOEUSER___SOMEFILE___000002
```

Variable names taken from the [Mercury job info](#), available with all host OS versions:

<creationtime>	Start time of job
<date>	Date
<destaddress>	Destination address of a job
<destport>	Destination port as queue name from LPD protocol
<jobid>	ID of the job
<jobname>	Name of the job
<npsversion>	version of Mercury system
<pages>	Number of pages so far*
<printedpages>	Number of pages confirmed by printer so far*
<srcaddress>	Address of senders as IP address or Novell queue name
<starttime>	Time when job started
<ticks>	Milliseconds since system start
<username>	Name of Mercury user

(*) in most cases, the number of pages will be zero, because the group boundary is evaluated at the start of the print job.

Additional variable names available with **MVS / VSE** host systems:

<gb_jobclass>	Byte 3 The one-character EBCDIC job CLASS parameter associated with the data set being printed.
<gb_jobname[8]>	Bytes 4–11 The eight-character EBCDIC job name parameter associated with the data set being printed.
<gb_jobid[8]>	Bytes 12–19 The eight-character EBCDIC job identification number parameter associated with the data set being printed.
<gb_forms[8]>	Bytes 20–27 The eight-character EBCDIC job FORMS parameter associated with the data set being printed.

Additional variable names available with **VM** host systems:

<gb_spoolclass>	Byte 3 The one-character EBCDIC spool class value for the print file, from the SPOOL CLASS parameter.
<gb_spoolfilename[8]>	Bytes 4–11 The eight-character EBCDIC filename of the spool print file.
<gb_userid[8]>	Bytes 12–19 The eight-character EBCDIC userid of the print file originator.
<gb_spoolformname>	Bytes 20–27 The eight-character EBCDIC spool formname value for the print file, from the spool FORM parameter.
<gb_spoolid[4]>	Bytes 28–31 The four-character EBCDIC spool identification number (spoolid) of the print file.

Additional variable names available with (older) **AS/400** host systems:

<gb_library[10]>	Bytes 3–12 The ten-character EBCDIC library name associated with the specified output queue. This value and the output queue name (bytes 13–22) correspond to DPF's CLASS parameter.
<gb_ouputqueue[10]>	Bytes 13–22 The ten-character EBCDIC output queue name. This value and the library name (bytes 3–12) correspond to DPF's CLASS parameter.
<gb_spoolfile[10]>	Bytes 23–32 The ten-character EBCDIC spooled file name associated with the file being printed.
<gb_spoolfilenumber[4]>	Bytes 33–36 The four-character EBCDIC spooled file number associated with the file being printed.
<gb_jobname[10]>	Bytes 37–46 The ten-character EBCDIC job name associated with the file being printed.
<gb_username[10]>	Bytes 47–56 The ten-character EBCDIC user name associated with the file being printed.
<gb_jobnumber[6]>	Bytes 57–62 The six-character EBCDIC job number associated with the file being printed.
<gb_formsname[10]>	Bytes 63–72 The ten-character EBCDIC forms name parameter associated with the file being printed. This value corresponds to DPF's FORMS parameter.

Additional variable names available with **MVS/VSE** host systems:

<gb_printfiletype>	Byte 3 A one-<gb_field indicating the type of print file being printed where the encoding is as follows: X'80' Job header X'40' Data set header X'20' User data set X'10' Message data set X'04' Job trailer X'00' Type not specified
<gb_jobclass>	Byte 4 The one-character EBCDIC job class associated with the data set being
<gb_jobname[8]>	Bytes 5-12 The eight-character EBCDIC job name associated with the data set being printed.
<gb_jobid[8]>	Bytes 13-20 The eight-character EBCDIC job identification number associated with the data set being printed.
<gb_jobform[8]>	Bytes 21-28 The eight-character EBCDIC job form (specified on the FORMS parameter in the OUTPUT statement) associated with the data set being printed.
<gb_programmername[60]>	Bytes 29-88 The 60-character EBCDIC programmer name (specified on the NAME parameter in the OUTPUT statement) associated with the data set being printed.
<gb_roomnumber[60]>	Bytes 89-148 The 60-character EBCDIC room number (specified on the ROOM parameter in the OUTPUT statement) associated with the data set being printed.
<gb_submissiondate[11]>	Bytes 149-159 The 11-character EBCDIC submission date parameter associated with the data set being printed.
<gb_submissiontime[11]>	Bytes 160-170 The 11-character EBCDIC submission time parameter associated with the data set being printed.

Additional variable names available with **AIX** and **OS/2** host systems:

<gb_printfiletype>	Byte 3 A one-byte field indicating the type of print file being printed where the encoding is as follows: X'80' Job header X'40' Copy separator X'20' User print file X'10' Message file X'08' User exit page X'04' Job trailer X'00' Type not specified
<gb_asciiname[251]>	Bytes 4-254 The 1-251 character ASCII name associated with the job being printed.

Additional variable names available with **AIX** and **Windowsprint** host systems:

<gb_asciiname[251]>	Bytes 3-253 The 1-251 character ASCII name associated with the job being printed.
-----------------------	---

Additional variable names available with the **variable length triplet format**:

<gb_groupid[244]>	Bytes 3-246 A 1 to 244 byte long group ID. The group ID is considered to be binary data, unless there was a preceding CGCSGID triplet in the XOH DGB command.
---------------------	---

Additional variable names available with (newer) **eServer iSeries** and **System i5** or **System i** host systems:

<gb_library[10]>	Bytes 3-12 The ten-character EBCDIC library name associated with the specified output queue. This value and the output queue name (bytes 13-22) correspond to DPF's CLASS parameter.
<gb_outputqueue[10]>	Bytes 13-22 The ten-character EBCDIC output queue name. This value and the library name (bytes 3-12) correspond to DPF's CLASS parameter.
<gb_spoolfile[10]>	Bytes 23-32 The ten-character EBCDIC spooled file name associated with the file being printed.
<gb_spoolfilenumber[6]>	Bytes 33-38 The six-character EBCDIC spooled file number associated with the file being printed.
<gb_jobname[10]>	Bytes 39-48 The ten-character EBCDIC job name associated with the file being printed.
<gb_username[10]>	Bytes 49-58 The ten-character EBCDIC user name associated with the file being printed.
<gb_jobnumber[6]>	Bytes 59-64 The six-character EBCDIC job number associated with the file being printed.
<gb_formsname[10]>	Bytes 65-74 The ten-character EBCDIC forms name parameter associated with the file being printed. This value corresponds to DPF's FORMS parameter.

If a "Build" string is used, ["Add form name to job name"](#) and ["Add MVS job class / OS400 spool file name"](#) are unavailable, because both features can be replaced by using variable names <gb_formsname>, <gb_jobclass> or <gb_spoolfile>.

Default: empty

- **Extract job name from data stream**

Enable to extract job name from data stream.

If the host does not send the 'Group Boundary' IPDS command, the job name might be extracted out of the banner or job separator pages.

Default: disabled

- **Use default job name (IPDS_JOB_#)**

If this option is enabled and no other job name is found (from "group boundary or page data), the IPDS emulation will generate a short job name consisting of "IPDS_JOB_" plus the Mercury job number. This short name fits into printers' LC-Displays.

If this option is disabled and no other job name is found, the standard long Mercury internal job name consisting e.g. of the sender's IP address is used.

Default: disabled

- **Trigger**

Define the trigger for the 'extract job name from data stream' parameter.

The trigger is a constant character sequence which precedes the variable job name in the banner page. You would describe this e.g. as a field descriptor. The trigger string has to be on the same line as the job name and before the job name.

- **Cut Offset**

Set the number of character positions (columns) between the last character of the trigger string and the first character of the job name.

Possible values: 0 to 255.

Default: 7

- **Cut Length**

Set the length of the job name.

It describes how many characters starting from the cut offset after the trigger string should be used for the job name.

Possible values: 1 to 128.

Default: 8

Example: The banner page looks like this:

```
*****START*****START*****START*****
*
*  JOBID.....:   JOB08207
*  JOB  NAME....:  MYJOBNAME
*  USERID.....:   THEUSER
*  SYSOUT CLASS:   M
*
*****START*****START*****START*****
```

To extract the job name, the trigger would be set to 'JOBID'. Since the Job name here e.g. JOB08207 is eight characters long and starts ten characters after 'JOBID', the cut offset would be set to 10 and the cut length to 8.

A trigger 'JOBID.....:' (with colon) and a cut offset of 9 would also work.

- **Allow blanks**

For compatibility reasons to old "Future V" IPDS emulations, blanks will be filtered out of the cut out job name string.

If switched on, the cut out job name will not be modified.

Default: off

- **Prescan 1st page**

When switched on, the first sheet of every job will be scanned before any output is generated. If a match is found, the new job name will be passed on to the dispatcher immediately. This is useful for working with "combined" emulations, so that the first (IPDS) emulation can extract the job name from a banner page. Otherwise, a spooled output port would be needed.

If switched on, the PCL output from "Miscellaneous / preload overlay fonts" might be sent before the PDL job header. Turn off "preload overlay fonts", if necessary.

Default: off

Job Separation

- **Split on command**

"Split on Command" enables job splits on a specific IPDS command contained within the print job. For the XOH and XOA commands, the subcommand order may be entered as a 4-digit hexadecimal value. This is especially useful if the IPDS sender inserts non-standard commands into the data stream, e.g. to drive finisher equipment. For further reference, please use IBM document S544-3417 'IPDS Reference Manual'.

- **Job Split on Trigger**

Define the trigger on which the job shall be split.

Sometimes it is desirable to split a large print job into several smaller ones, e.g. to start printing sooner when the jobs are very large or to trigger an offset stacker at the printer's finisher.

Job split on trigger means that the job will be split after the defined string has been found on the page.

- **Split Job after x pages:** enter the number of page breaks after which the incoming print job is to be split into smaller parts. For example, a print job with 5000 pages can be split into 50 smaller print jobs, each consisting of 100 pages. Also if no specific criterion exists for the end of a print job, and therefore all print jobs from the host system are processed in the Mercury system as a single job, a forced split can be achieved using this parameter. This option is helpful if the printer is collecting print jobs completely before starting printing. If the job is split into smaller pieces, the time until the first page gets printed is reduced. But it must be sure that some queue optimization of the printer is disabled, if the partially printed job pieces have to be in the right order. Maximum number of pages is 9999.

Default: 0 pages

- **Split non-finished sheets:** Enable to split the job after each sheet if the host does not send any finishing options for the current sheet. This is a trick to make sheets without finishing attributes go to the same output bin as sheets which finishing. By doing so, some printers will print faster, especially those that cannot select the paper path via the PCL

output bin command or a PjL header. Use in conjunction with [Printer Control / Job separation and finisher commands](#).

Default: disabled

- **Split job on AOS:** Enable to split the job when an IPDS 'Alternate Offset Stacker (AOS)' command has been encountered in the data stream. If Printer Control/Misc./Early AOS trigger is set, the job will be split before the current page, or after if not set. The position of the AOS command within the data stream can usually be controlled via the 'JOG OUTPUT' command in the FORMDEF on the host system.

Default: disabled

- **Group boundary split:** Enable to split the job when an IPDS 'XOH Define Group Boundary: Terminate' command has been encountered in the data stream and that command is associated with a "keep group together" XOH-SGO attribute
- *off:* group boundary splitting is disabled
- *inner:* split at every "XOH-DGB: terminate" command encountered, with nested or non-nested groups
- *outer:* if groups are nested, split at the outermost "XOH-DGB: terminate" command only. For non-nested groups, the split is executed at every group

Default: inner

- **Split at output bin change:** If enabled, IPDS will split the job whenever the host system changes the 'IPDS media destination ID'.

Default: disabled

- **Split at input tray change**

If enabled, a new print job is generated whenever the IPDS input media source changes, either after a SIMS command or upon an LCC order. Use this feature, if you have a PCL printer that cannot switch input paper trays within a single print job.

Default: disabled

Transparent Data Only

Select an entry from the pull-down menu.

This option will extract text data from the IPDS 'Write Text / Transparent Data' command using different methods. IPDS data is used here as an encapsulation for other type of data which will be processed by an other emulation behind IPDS to PCL5 if necessary.

- **off (default)**

Select to not affect the IPDS to PCL5 emulation.

- **data only**

Select to let the IPDS to PCL5 emulation not generate printer commands, but print text data extracted from IPDS 'Write Text / Transparent Data' command without any conversation instead.

- **data + LF**

Like data only, but each block of data will have a line feed (Hex 0x0A) attached.

- **valid channel order, data, LF**

Like data + LF, but data blocks will get printed only if the first byte is a valid ANSI channel command:

0x40: Space 1 line before printing
0xF0: Space 2 lines before printing
0x60: Space 3 lines before printing
0x4E: Suppress Space before printing
0xF1: Skip to channel 1 and print
0xF2: Skip to channel 2 and print
0xF3: Skip to channel 3 and print
0xF4: Skip to channel 4 and print
0xF5: Skip to channel 5 and print
0xF6: Skip to channel 6 and print
0xF7: Skip to channel 7 and print
0xF8: Skip to channel 8 and print
0xF9: Skip to channel 9 and print
0xC1: Skip to channel 10 and print
0xC2: Skip to channel 11 and print
0xC3: Skip to channel 12 and print

Default: off

Send form and job name via PJJ

Enable to send form and job name via PJJ

If a valid names for job, user or form have been recognized in a IPDS 'Group Boundary' command, they will be send to the printer in special PJJ command at the beginning of the job. These commands can be used as a trigger for selecting the configuration in an Overlay Manager attached to the IPDS emulation.

Example:

A host system is sending job 'MYJOB123' produced by user 'USER001' with form name '*STD'. This will result in a PJJ sequence which will be as:

```
@PJJ SET CSP_HOSTJOBNAME=MYJOB123
@PJJ SET CSP_HOSTFORMNAME=*STD
@PJJ SET CSP_HOSTUSERNAME=USER001
```

In an Overlay Manager attached to the emulation, a configuration trigger like 'CSP_HOSTFORMNAME=*STD' could be defined.

Default: disabled

Substitute incomplete download fonts

Enable to substitute incomplete download fonts.

If this feature is checked, the IPDS emulation will substitute incomplete (broken) download fonts with a courier font.
Use only if necessary since this switch might affect logos/images downloaded as a font.

Default: disabled

Load DBCS bitmap fonts on demand

Check to reduce amount of PCL output.

Segmented IPDS bitmap fonts are usually large and can consist of more than 40000 characters.
Some PCL or PostScript printers do not have enough font memory installed to handle so many

characters and slow down their processing.

If this option is set, only those PCL characters will be generated that are actually used. However, this option needs more processing time within the IPDS emulation.

Default: enabled

Preload overlay fonts

Check to load fonts before an IPDS overlay is processed.

If used in combination with "[Load IPDS overlay as PCL macro](#)" fonts can be downloaded within a PCL overlay.

Some printers and the PCL-to-PS and PCL-to-PDF emulations need much (printer) memory to resolve such fonts.

If this option is set, the IPDS emulation will try and load all fonts needed within overlays before the overlay starts.

Drawbacks:

- more processing time is needed because the overlays need to be pre-scanned recursively.
- before an overlay, there are additional font activations generated which might affect printer performance.

Advantages:

- + PDF and Postscript output will be smaller in size
- + PDF and Postscript output can be processed faster by emulator as well as by printer

This option only works if "[Logical Page Processing](#)" is set to "*Collect logical pages*". Otherwise, this option does have no effect. For optimum effect, please turn off "Load DBCS bitmap fonts on demand".

Default: enabled

Allow oversized bitmap fonts

Check to enable processing of bitmap fonts with a character size greater than 255 by 255 pixels.

Some non-IBM AFP and IPDS generators send bitmap fonts with large characters, exceeding the maximum size defined in the IPDS specification.

With this flag enabled, you can process these fonts without generating NACKs.

Default: off

LX LPD compatibility

Check to enable "quirks mode", emulating a Lexmark bug

Use if you replace printers with older Lexmark SIMMs and bar codes, IOCA images or GOCA areas (maybe containing large font texts) are positioned wrong.

If this option is set, the "Initial I" and "Initial B" cursor position values from the current Logical Page Descriptor IPDS command are added to the area offsets of IOCA, GOCA, and BCOCA objects.

Default: off

AFCCU BCOCA quirk

Check to enable bar code "quirks mode", emulating an AFCCU printer bug.

For printers with an AFCCU (i.e. "IPDS printer / printer type" is 3935, PRN4000, PRN4001,

PRN4100, C130plus and user-defined OPC/STM mode), a bug is emulated: if a BCOCA bar code width is set by the host and the current L-Unit value is 14400 per 10 Inch, the emulated printer calculates the bar code width with 1440 dpi instead of mils (1000th Inch), resulting in a bar code that is too narrow.

If this option is set that bug is emulated and the width is calculated the same way the 4100 printer does (incorrectly), resulting in a bar code that is too narrow.

Default: off

Pos. all GOCA fonts

This is a backwards compatibility switch. Check to make the IPDS emulation behave like before V7.11

If a GOCA text is created using a download bitmap font and the "Set Character Cell (GSCC)" order, all text characters will be positioned individually using the spacing set in the GSCC order.

If this option is the off, GSCC character cell width will be ignored for proportional fonts.

Default: off

Check VPA

IPDS can generate 'Position Check NACK' messages, if the cursor or an imaging area is out of the 'Valid Printable Area' (VPA) bounds. There are options for

- Never** no VPA boundaries are checked. This is the default setting.
- Auto** the VPA boundaries are checked if the host system has sent an 'Exception handling control' (EHC) command with the 'Report Position' flag set.
- Always** the VPA is checked and the EHC flag is ignored.

The Check VPA feature does not apply to secure overlays.

Undefined chars

Check for undefined characters

IPDS can generate 'Undefined Character NACK' messages, if there are invalid characters found. There are options for

- Ignore** the IPDS emulation ignores undefined characters and sends a blank (space) character instead.
- Auto** the IPDS emulation checks for undefined characters if the host system has sent an 'Exception handling control' (EHC) command with the 'Report undefined character check' flag set. This is the default setting.
- Always check** the IPDS emulation always checks for undefined characters and the EHC flag is ignored.

Default: Auto

Input trace to file

For debugging purposes only. If enabled an IPDS input trace is captured directly at the emulation's entry point. The data is written into the emulation's resource folder. Useful if IPDS is a secondary emulation, e.g. if it is coupled with the "Comparex Line Mode" emulation.

Default: Input trace off

PS/PDF font passthrough

With these options checked, outline fonts can be passed on to the next emulation, without rendering them into bitmap fonts. This increases emulation speed and reduces the amount of output data. However, these options only work in combination with a PCL2xxx emulation, e. g. PCL-to-PDF, PCL-to-PostScript or PCL-to-GDI. In addition to checking the font options here, "[Printer Control / Job separation and finisher commands](#)" has to be set to "[PJL for PS and PDF emulation](#)"

- **Type 1 outline fonts**

enables passthrough for Adobe(tm) type 1 outline fonts

- **CID keyed DBCS outline fonts**

enables passthrough for Adobe(tm) CID-keyed double-byte outline fonts

- **TrueType / OpenType fonts**

enables passthrough for TrueType / OpenType fonts outline font objects


Default: all off

IPDS Overlay Supplement

This chapter describes the Overlay Supplement options of the IPDS to PCL5 emulation. With the Overlay Supplement the user will be able to add user defined page elements.

The Overlay Supplement is available in IPDS emulation, version 2.00.114 and higher. The Overlay Supplement enables the user to add recurring supplements such as standard texts, images or forms to an existing print job. These supplements will be furthermore referred to as Overlay Resources or Overlays, respectively. Overlay Resources can be created by any Windows application using the Resources Printer Driver.

Definitions

 Be careful not to confuse those overlays with the overlays that are generated on the host:

IPDS Page Segment

An overlay generated on the host that is added by the IPDS interpreter to the printout on a fix and defined position.

IPDS Overlay

An overlay generated on the host that is added by the IPDS interpreter with a variable or a fix offset to the printout. These overlays can be requested from the host using on the IPDS commands 'Include Overlay' or 'Load Copy Control'. The latter is often referred to as LCC or 'Copy-Group' overlays, respectively.

Overlay Resource

Overlays that are added to the print data stream not directly controlled by the host, but dependent on the rules defined in the file IPDSOVER.INI. These overlays emerge from the output files of the Resources Printer Driver.

Overlay Resources are files - mostly with the extension *.CSP -, that are generated by the Resources Printer Driver when printing a Windows document. The Resources Printer Driver behaves similarly to a standard Windows printer driver in the 'Print to file' mode.

The files generated by the driver contain a copy of the printed page as a PCL bitmap image with a resolution of 300 dpi and a file header, that includes additional information on the resource name and its position and orientation on the printed page.

The file IPDSOVER.INI contains the commands and definitions for the sequential control of the IPDS Overlay Supplements. This is an ASCII text file and can be edited using any Windows text editor such as NOTEPAD.EXE.

On the Mercury system any IPDS to PCL5 emulation in a print process can use its own IPDSOVER.INI file. Default directory of the file IPDSOVER.INI is the resources directory of the referring Mercury Server's emulation.

You can edit this file after clicking the Button Edit by the resource path entry 'Emulation / Configure / Resources'. It is also possible to use a shared file IPDSOVER.INI for several IPDS emulations, thus enabling an easier system maintenance. The resources path of the emulations concerned has to be changed to a shared directory.

Any file name of an overlay resource mentioned in the file IPDSOVER.INI is composed relatively to the IPDSOVER.INI file's path name in the file IPDSOVER.INI by using the command DEFINE FILENAME.

If the overlay resource resides in the same directory as the file IPDSOVER.INI, no path must be added to the file name.

Example:

In a Mercury system there shall be two IPDS emulations in one print process. Both emulations shall share the overlay resource KOPF1.CSP.

It is suggested to save this resource to

```
<Installation Path>\Res\DSPPATH1\KOPF1.CSP.
```

An overlay resource FORM1.CSP for just the first emulation should be saved to

```
<Installation Path>\Res\DSPPATH1\1\FORM1.CSP.
```

The file IPDSOVER.INI in the directory

```
<Installation Path>\Mercury\Res\DSPPATH1\1\
```

contains:

```
DEFINE ID 1 FILENAME ..\KOPF1.CSP  
DEFINE ID 2 FILENAME FORM1.CSP
```

where <Installation Path> is the installation path of Mercury

The file IPDSOVER.INI is a line oriented plain text file in ASCII format. Any line is finished with a CR/LF (Carriage Return/Line Feed). Any line is a blank line, a comment line or a command line, respectively. The file IPDSOVER.INI is opened and evaluated by the IPDS interpreter at the beginning of a print job.

If there are no current IPDS print jobs to be processed by the Mercury system, the IPDSOVER.INI can be edited even when the system is running.

The changes, however, do not occur before the next print job is started. Is the interpreter in 'Static Mode', it is recommended to restart the referring dispatcher.

If there is no IPDSOVER.INI available for the referring interpreter, the IPDS Overlay Supplement for this IPDS interpreter will be disabled. If an Overlay Resource file mentioned in the file IPDSOVER.INI can not be found, the referring rule will be skipped without displaying an error message. This does not affect the other resources in the same IPDSOVER.INI.

For a better overview and documentation of the definitions and rules in this file we suggest that you add comment lines to every definition or rule. Comment lines start with a semicolon (ASCII 59) and should not be longer than 255 characters.

```
; IPDSOVER.INI date ... for application "Example"  
; last modified ... : ...
```

In a command line, any keyword and argument is separated by one or several blanks. Arguments may not contain blanks! Keywords are not case sensitive, but arguments are. They should not contain German Umlauts (ä, ü, ö), German ß or accent characters (´). Punctuation marks and numbers are allowed.

Any command line starts with the keyword DEFINE for a definition or RULE for a rule. Any Overlay Resources file is assigned via definition to an ID number. If you want to use the same resources file at different coordinates on the page (using the keyword OFFSET), you have to assign several ID numbers.

Compliant with the rules, pasting the overlay resource with the ID number is assigned to certain events. Several triggers that were defined over different keywords can be combined as a logical 'AND' operation in one RULE. If the conditions should be combined as a logical 'OR' operation, you have to use more than one RULE entries.

The maximum number of definitions and rules in one IPDSOVER.INI is 40 each.

The IPDS Overlay Supplement uses its own sheet counter in the IPDS interpreter during procession of a print job. At the beginning of a print job or using a RULE with the keyword RESETCOUNTER the sheet counter is reset to 1.

Difference between sheet and page

To simplify arranging the overlays the IPDS Overlay Supplement counts sheets as they are printed and not the IPDS print pages shown on the host. The IPDS counter on the host counts logical pages. A page can contain up to eight logical pages. When printing simplex (one-sided, single-edge) without combination output tray (N-UP) a logical page corresponds to a sheet. The sheet counter makes it easier to find the correct place for an overlay in mixed print jobs. This is done by counting the printed sheets after a test print.

Counting and duplex printing

When printing duplex (double-sided) a sheet contains 2*N logical pages, with N being the number of combined pages on one sheet (normally 1).

The IPDS Overlay Supplement can distinguish between front and back pages. 'Tumble' and 'Calendar' duplex or 'long edge' and 'short-edge binding', respectively, will be treated equally.

Counting and N-UP printing

In the N-UP printer mode of the IPDS ('combination output tray') the host prints several pages on one sheet; i.e. when doing 3-UP printing 3 logical pages per front and back page. Empty parts of pages are also counted. The Overlay Supplement only counts the actually printed sheets, not the IPDS pages.

Resetting the sheet counter

The internal sheet counter starts with 1 and will be reset at the beginning of a new print job or with a RULE and RESETCOUNTER.

Definitions assign an ID number to a resource. In most cases the resource can be used under this ID number. For a better overview it is suggested to put the definitions at the top of the IPDSOVER.INI. Definitions must start with the keyword DEFINE and may contain two or more combinations of keywords and arguments. The keyword's ID and FILENAME must be contained in a definition, too.

ID

ID requires a numeric argument between 1 and 65535. The ID number can be chosen anywhere in this range. It may only be defined once per IPDSOVER.INI.

Example: ID 10

FILENAME

FILENAME requires a alphanumeric argument for the file name of the resources file with file name extension and a relative path to the IPDSOVER.INI if necessary.

Example: FILENAME ..\kopfl1.csp chooses the file KOPF1.CSP from the parent directory.

OFFSET

OFFSET requires two numeric arguments X and Y for positioning the overlay relatively to the origin defined in the overlay file. The unit is 1 PCL dot (a 300th of an inch). This makes an exact positioning of the overlay possible without having to generate a new resource via the printer driver. The point of origin is in the upper left corner of the page at the beginning of the 'PCL Printable Area' (dependent on the printer model). Positive coordinates point to the right hand side or the bottom of the page, respectively.

Example: OFFSET -50 -100 shifts the overlay 50/300 inches to the left and 100/300 inches to the top (that equals 4.2 millimeters to the left and 8.5 millimeters to the top).

MACRO

MACRO does not require any argument. If MACRO is specified, the resource will be stored in the printers as a (permanent) PCL macro. That way it is possible to reduce the size of the data stream sent to the printer, when the overlay in a print job is used repeatedly. Premise is a corresponding memory expansion of the printer.

Example: MACRO

ADDBACKSIDE

ADDBACKSIDE does not require any arguments. ADDBACKSIDE will generate from a simplex printed page a duplex page with an overlay as backside. ADDBACKSIDE must be used together with SIDE SIMPLEXBACK in the referring RULE. When the backside is pasted, other control commands for a form feed will be used.

Example: ADDBACKSIDE

RGBCOLOR

RGBCOLOR requires three numeric arguments ranging from 0 to 255 for the red, green and blue values. 0 0 0 is black, 255 255 255 is white. The colors will be mixed according to the rules for PCL5c color laser printers. The RGBCOLOR command enables the user to color a black-and-white bitmap of the Resource Overlays.

 This works only with PCL5c color laser printers and not with ink jet or other color printers.

Example: `RGBCOLOR 127 0 127` tinges the overlay in a violet color.

Rules set the conditions for an overlay resource to be added to the data stream. A rule is composed of at least the keyword `RULE` plus the keyword `ID` with argument. Such a rule will be valid for every page of the print job.

To specify the scope of application, rule keywords can be combined. Any keyword may be used only once per rule. It is possible to create more than one rule for one and the same overlay resource. Keywords in one rule are associated logically with 'AND'. If more than one rule are processed, the keywords are associated logically with 'OR'.

ID

`ID` requires a numeric argument between 1 and 65535. The ID number can be chosen from this range and must correspond to an ID number in the definition part.

Example: `ID 10`

FROM

`FROM` requires a numeric argument between 1 and 65535. The `FROM` parameter specifies the sheet counter reading from which one on the overlay resource will be pasted.

For duplex and N-Up print jobs that rule is valid for the whole sheet, i.e. the overlay will be pasted to the front and to the back of a sheet. If this should not be done, the rule must be complemented with the `SIDE` keyword, e.g. `SIDE FRONT` (see below).

Note for programmers: The keywords `FROM`, `TO` and `STEP` do function similarly to the `FOR` statement in higher level programming languages.

Example: `FROM 2` pastes the overlay from sheet 2 on to every coming sheet of the print job.

TO

`TO` requires a numeric argument between 1 and 65535. The `TO` parameter specifies the sheet counter reading up to which one the overlay resource will be pasted.

For duplex and N-Up print jobs that rule is valid for the whole sheet, i.e. the overlay will be pasted to the front and to the back of a sheet. If this should not be done, the rule must be complemented with the `SIDE` keyword, e.g. `SIDE FRONT` (see below).

Example: `TO 2` pastes the overlay up to sheet 2 to every sheet of the print job (sheets 1 and 2).

`FROM 1 TO 1` pastes the overlay only to the first sheet.

STEP

`STEP` requires a numeric argument between 1 and 65535. The `STEP` parameter together with the `FROM` keyword specifies the increment with which the reference value should be counted from the start counter on. If the reference value equals the sheet counter, the condition is fulfilled, i.e. if the overlay resource shall be pasted on every second sheet, you have to put in `STEP 2`.

For duplex and N-Up print jobs that rule is valid for the whole sheet, i.e. the overlay will be pasted to the front and to the back of a sheet. If this should not be done, the rule must be complemented with the `SIDE` keyword, e.g. `SIDE FRONT` (see below).

Example: `FROM 2 STEP 2` pastes the overlay on every second sheet, starting with sheet 2.

SIDE

SIDE requires a keyword as a further argument: FRONT, BACK or SIMPLEXBACK. SIDE specifies, if an overlay resource should be pasted to the front and to the back of a sheet.

In duplex jobs the overlay will be pasted to both front and back if SIDE is not chosen. In simplex jobs you can use SIDE SIMPLEXBACK to paste an overlay to the backside of the sheet. To achieve that, an additional page will be added to the data stream.

In print jobs with both simplex and duplex pages there may be two rules required: one with the keyword SIDE BACK and one with SIDE SIMPLEXBACK.

FRONT

Pastes the overlay resource when either the front of a duplex sheet is reached or a simplex page. When printing simplex, SIDE FRONT is redundant.

Example: `SIDE FRONT`

BACK

Pastes the overlay resource when the backside of a duplex sheet is reached. SIDE BACK checks for a logical condition: The sheet was marked from the host as 'duplex' and the back of the sheet will be edited.

If the host marks the page as 'simplex', the condition cannot become true, because there is no back of the sheet in the print job. In this case use SIDE SIMPLEXBACK.

Example: `SIDE BACK`

SIMPLEXBACK

Pastes an additional printed page that contains the chosen overlay resource to the output data stream after a simplex printed page on the same sheet. The corresponding ID number must be marked in the definition as ADDBACKSIDE.

If there is more than one overlay to be included on one page, the RULE with SIMPLEXBACK has to be the last one in the sequence of rules because SIMPLEXBACK will cause a form feed in the print data stream output.

Example: `SIDE SIMPLEXBACK`

FORMNAME

Requires an alphanumeric argument. If the host sends an IPDS 'Group Boundary' command in the data stream that contains a FORMS parameter, this keyword can be used to check for the forms name sent with FORMS. FORMNAME requires a PSF on the host that supports 'Group Boundary'. FORMNAME is case sensitive.



FORMNAME, CSPSTRING and FINDSTRING may not be used together in one and the same rule.

Example: `FORMNAME *STD` pastes the overlay, if the form name is set to default in the (AS/400-) print job properties. The default form name on the host can be linked to the login profile, thus enabling the system operator to associate certain user groups to their own overlay, e.g. its own letterhead for the marketing dept, its own logo for the R & D. If the user does not want to use that form he or she can simply use another form name when printing. We recommend to setup a default reply to the operator message prompting to use another form.

CSPSTRING

Requires an alphanumeric argument. CSPSTRING is case sensitive. CSPSTRING searches the print data stream for the sequence 'CSPOVERLAY!'. As argument from the data stream it uses the string, that stands between the first and another exclamation mark. The whole initial string is in the printout replaced with blanks. It is recommended to put this command at the beginning or the end of the page.

Example: `CSPSTRING Letterhead1` pastes the overlay if the print data stream contains the identification 'CSPOVERLAY!Letterhead1!'. The text 'CSPOVERLAY!Letterhead1!' will be replaced in the printout with blanks.

FINDSTRING

Requires an alphanumeric argument. FINDSTRING is case sensitive. FINDSTRING searches the print data stream for the sequence mentioned in the argument without affecting the print data stream. The search sequence may not contain any blanks.

Example: `FINDSTRING assessment` pastes the overlay, if the data stream contains the word assessment. The word assessment will not be replaced by blanks in the printout.

FINDSTRING2

Requires an alphanumeric argument. FINDSTRING2 is case sensitive. FINDSTRING2 searches the print data stream for the sequence mentioned in the argument without affecting the print data stream.

The search sequence may not contain any blanks. FINDSTRING2 is to be used in conjunction with FINDSTRING in order to achieve an 'AND' condition when searching for two words on the same page.

Example: `FINDSTRING John FINDSTRING2 Doe` pastes the overlay, if the data stream contains the words John and Doe. These two words will not be replaced by blanks in the printout.

JOG

Does not have any arguments. The condition JOG is true if the print data stream contains the command 'Alternate Offset Stacker / AOS'. JOG can also be used to separate print jobs. The operator may create it by using the FORMDEF command 'JOG OUTPUT'.

Example: `JOG` pastes an overlay on the first sheet of a new print job.
If one of the rules contains conditions that scan the sheet counter (FROM, TO or STEP), it is recommended to combine JOG with RESETCOUNTER.

RESETCOUNTER

Does not have any arguments. If a rule uses RESETCOUNTER, the sheet counter will be reset to 1, when the other conditions in this rule become true.

In print jobs with a fix number of pages we recommend to combine RESETCOUNTER with FROM and TO.

In print jobs with a variable number of pages we recommend to combine RESETCOUNTER with JOG, FINDSTRING or CSPSTRING.

Example: `RESETCOUNTER`

MEDIASOURCE

MEDIASOURCE requires a numeric argument between 0 and 255. The condition MEDIASOURCE will become true, if the host sends for the corresponding sheet a control

command for the paper feed with the corresponding number.

- ⚠ Dependent on the host the default tray is in most cases either tray one or tray two. According to the printer model set in the IPDS emulation, only a limited number of paper feed trays can be used. By defining the assignment, the IPDS interpreter enables the use of several logical paper feed trays that chose the same 'real' paper feed tray. In combination with MEDIASOURCE it is possible to create 'virtual form paper feed'.

Example: `MEDIASOURCE 2` pastes the overlay, when paper from slot 2 shall be used.

MEDIADESTINATION

MEDIADESTINATION requires a numeric argument between 0 and 255. The condition MEDIADESTINATION will become true, if the host sends for the corresponding sheet a control command for the paper output tray with the corresponding number.

- ⚠ Dependent on the host the default tray is in most cases either tray one or tray two. According to the printer model set in the IPDS emulation, only a limited number of paper output trays can be used. By defining the assignment, the IPDS interpreter enables the use of several logical paper output trays that chose the same 'real' paper output tray. In combination with MEDIASOURCE it is possible to create 'virtual form output trays'.

Example: `MEDIADESTINATION 2` pastes the overlay, when paper shall be put out to tray 2.

Overlay only on the first sheet

```
DEFINE ID 1 FILENAME ov1.csp
RULE FROM 1 TO 1 ID 1
```

Overlay on all pages from page 2 on

```
DEFINE ID 2 FILENAME ov2.csp
RULE FROM 2 ID 2
```

Overlay on all backsides (duplex printing)

```
DEFINE ID 3 FILENAME ov3.csp
RULE SIDE BACK ID 3
```

Overlay on all backsides (simplex printing)

```
DEFINE ID 4 FILENAME ov4.csp ADDBACKSIDE
RULE SIDE SIMPLEXBACK ID 1
```

Overlay only on backsides, from sheet 2 on (simplex and duplex printing)

```
DEFINE ID 5 FILENAME ov5.csp
DEFINE ID 6 FILENAME ov5.csp ADDBACKSIDE
RULE FROM 2 SIDE BACK ID 5
RULE FROM 2 SIDE SIMPLEXBACK ID 6
```

Overlay on any third front

```
DEFINE ID 7 FILENAME ov7.csp
RULE STEP 3 SIDE FRONT ID 7
```

Chose overlay by form name on the host / IPDS 'Group Boundary' command

```
; FORMS name on the host is LETTER
DEFINE ID 8 FILENAME ov8.csp
RULE FORMNAME LETTER ID 8
```

Chose overlay by CSPOVERLAY!...! command in the print data stream

```
; document must contain CSPOVERLAY!form9!
DEFINE ID 9 FILENAME ov9.csp
RULE CSPSTRING form9 ID 9
```

Chose overlay by text querytext in the print data stream

```
DEFINE ID 10 FILENAME ov10.csp
RULE FINDSTRING querytext ID 10
```

Reset sheet counter after a certain number of sheets

```
; Reset sheet counter after every 500th sheet
RULE FROM 500 TO 500 RESETCOUNTER
```

Reset sheet counter by AOS / JOG output

```
RULE JOG RESETCOUNTER
```

Overlay dependent on 'Input Media Source' / Paper feed

```
DEFINE ID 11 FILENAME ov11.csp
RULE MEDIASOURCE 1 ID 11
```

Overlay dependent on 'Destination Bin' / Output tray

```
DEFINE ID 12 FILENAME ov12.csp
RULE MEDIADESTINATION 2 ID 12
```

Invoice with several pages

The task is to print invoices that consist of a letter with letterhead on the front of sheet 1 and the detailed invoice from sheet 2 on. The print job is simplex. The company's general terms and conditions shall be printed on the back in light gray:

```
; Example print invoice with general terms and conditions
;
; first of all define letterhead
DEFINE ID 1 FILENAME briefk1.csp
; now the invoice form
DEFINE ID 2 FILENAME rech1.csp
; and the text of the general terms and conditions in light gray
DEFINE ID 3 FILENAME agb.csp ADDBACKSIDE RGBCOLOR 127 127 127
;
; rules:
; letterhead only on first sheet
RULE FROM 1 TO 1 ID 1
; invoice form from sheet 2 on
RULE FROM 2 ID 2
; general terms and conditions on all sheets on the back
```

```
RULE SIDE SIMPLEXBACK ID 3
; reset page counter, if several invoices are combined
; in one print job
RULE JOG RESETCOUNTER
```

6.3.14.2.2 PCL5 to PDF Emulation

Overview

The PCL5 to PDF emulation from [Mercury](#) converts print data that were created for PCL printers so that they can be output on PDF compatible page printers.

Parameters

Jobs

Split Jobs

Enable to split jobs due to the given criteria. Select one or any number of criteria.


A selected criterion will be highlighted. Several criteria can be combined. Possible criteria are:

- Job boundary
- Page boundary
- Simplex/duplex change
- Input tray change
- Output tray change
- Paper size/media change
- Job separation command <ESC>&l1T

Default: disabled (no criteria highlighted)

Include all resources in each split

Enable to include all required resources within each split.

 It is not recommended to disable this flag.


Default: enabled

Split port

Select split port from the pull-down menu.

Possible ports are:

- *Input*
- *Output*

 On input splitting the split parts are handled as separate jobs at the [Mercury](#) system, otherwise only the receiving device will see the split parts as separate jobs.

Default: Input

every <x> page(s)

Enter number of pages after which a split should be made. This option is only available if the job split criteria 'Page boundary' is selected.

Allow job name modification

When activated, the job name of the system will be updated with the extracted job name of an

encountered '@PJL JOB NAME=..."' command.

PDF output generation - Compression Level

Set compression level for PDF output.

The higher the compression level the less the size of the PDF output. Levels range from 1 to 9 with 9 as the maximum compression.

Default: OFF

PDF output generation - Image mask method

Set image mask method.

This setting is used to achieve compatibility of certain color images in combination with some printer's PDF implementation. The image type in question uses color palettes in combination with a transparent color value. Possible values are:

- **By Color Index:** Set to transfer the color image with a modified color palette without an extra transparency mask. Use this setting to achieve the lowest amount of output data. Acrobat Reader(TM) V 5.0 cannot display these images, but they can be printed though.
- **Extra Bitmap:** Set to generate an extra transparency bitmap in addition to the color image. This method results in a slight increase of the amount of data. Acrobat Reader V5.0 is able to display the images. (default)
- **Expand To 24-Bit:** Set to expand the image into a 24-bit full color image without a palette or mask. This method has highest compatibility with printers and reader programs at the expense of an increased amount of output data.
- **Paint As Mask:** Set to transfer the image with its attribute set to 'image mask'. For further information about this attribute see Acrobat documentation.

Default: Paint As Mask

PDF output generation - Device mode

Possible settings are:

- *Printing*
It is assumed the configured device is capable to perform the device specific actions sent by the output port.
(e.g. duplex printing, etc.)
- *Viewing*
In this case the pages for a duplex back side in short edge binding mode will be rotated by 180 degree,
in order to be displayed in the expected orientation within a PDF reader.

PDF output generation - Insert info notes

This option is available when '*Device mode*' is set to '*Viewing*'. When active, each page contains a info note with used duplex mode, used input tray, etc.

PDF output generation - Auto rotation

This option is available when '*Device mode*' is set to '*Viewing*'. When active, the PDF page is rotated in sync with the "PCL page orientation" command so that the "Portrait" print direction is always readable.

PDF output generation - Soft font embedding

PDF does not support fonts with incrementally character downloads. Therefore, when a character is added after usage of a font, a new font instance with the new character set must be created. This can result into very large PDF output with multiple copies and instances of a font, especially when TrueType download fonts will be sent using incrementally character downloads. Unfortunately the most Windows PCL drivers utilizing this manner, which is very efficient for PCL printers but not for the PDF world.

- **Immediate:**

The font will be embedded immediate. This can result in large PDF files when characters are incrementally downloaded but the generated PDF is always correct.

- **Defer to page end:**

Creation and embedding of the font is deferred to page end. This method results in smaller PDF output on incrementally character downloads.

 **!!! The output should be proved for correctness !!!**

Activation of this option might result in wrong output under following condition :

```
Download ==> 'A'
Print      ==> 'ABC'
Download ==> 'B'
Download ==> 'C'
```

(NOTE: A real printer would show only 'A', - due to the delay - PDF shows 'ABC')

It works when:


```
Download ==> 'A'
Print      ==> 'A'
Download ==> 'B'
Print      ==> 'AB'
Download ==> 'C'
Print      ==> 'ABC'
```

(NOTE: This is the usual way a Windows PCL driver download a TrueType font.)

PDF output generation - Inline images

When active, simple images with less than 4 KByte are embedded as inline images into the PDF.

In combination with compression this can reduce the size of the generated PDF.

 There might be rare cases where this is **counterproductive**, i.e. when a lot of small images are used within a PCL macro and a large number of pages is using that macro, the size of the resulting PDF might be bigger when activating this option.

Page Output - Face down printing

Select **Face up** or **Face down** printing.

Default: Face up

Page Output - Page order

Select **Page 1 - n** to print the job from the first to the last page.

Select **Page n - 1** to print the job from the last to the first page.

Default: Page n - 1

Stapling

Select the stapling command.

Available options are **Off**, **Front**, **Rear** or **Dual**.

Default: Off

Output Destination

Set output destination (paper tray). These options are device dependent.

AUTOMATIC - Standard output bin (determined by the printer)

UPPER BIN - Usually the 'face-down' output bin

LOWER BIN - Usually the 'face-up' output bin

HC OUTPUTBIN to **HC OUTPUTBINS** - Output bins of a high capacity option

Default: AUTOMATIC

Enable or disable encryption of the document. The access permissions of an encrypted document can be additionally specified.

Encryption

NONE

No encryption will be used.

40 bit (compatibility PDF 1.2, Reader 3.x)

Encrypts the document utilizing RC4 encryption with 40 bit. The document may be viewed with Adobe Reader 3.x and above.


128 bit (compatibility PDF 1.4, Reader 5.x)

Encrypts the document utilizing a stronger RC4 encryption with 128 bit. Additionally there can be more access permissions specified. The document may be viewed with Adobe Reader 5.x and above.

Passwords

Specify a 'user' password (required to open the document) and/or a 'owner' password (required to change permissions and the passwords).

- Opening the document with the correct user password (or opening a document that does not have a user password) allows additional operations to be performed according to the user access permissions specified in the document's encryption dictionary.
- Opening the document with the correct 'owner' password (assuming it is not the same as the user password) allows full (owner) access to the document. This unlimited access includes the ability to change the document's passwords and access permissions.

 The document can be encrypted without specifying any password.

User

When a user password is set, the Adobe Reader or Adobe Acrobat requests this password before allowing access to the document.

Owner


When an owner password is set, the Adobe Acrobat requests this password before allowing changing of the password or access permissions.

Permissions

Specify access permissions for the encrypted document.

Allow printing

The document may be printed.

 When using 128 bit encryption, additionally the quality level for printing may be specified.

Allow modifying document contents

The contents of the document may be modified.

Allow contents copying, extracting and accessibility


Copy or otherwise extract text and graphics from the document, including extracting text and graphics (in support of accessibility to users with disabilities or for other purposes).

Allow modifying and adding text annotations and form fields

Add or modify text annotations, fill in interactive form fields, and, if 'Allow modifying document contents' is also set, create or modify interactive form fields (including signature fields).


Allow form field filling and signing

Fill in existing interactive form fields (including signature fields).

 Only available when using 128 bit encryption.

Allow document assembling (insert, rotate, delete pages, ...)

Assemble the document (insert, rotate, or delete pages and create bookmarks or thumbnail images).

 Only available when using 128 bit encryption.

Enable or disable digital signing of the document. When the document has been digitally signed, any modification of the document will result into invalidating the embedded signature.

There are two alternate formats for storing public key signatures in the signature dictionary of a PDF file. The 'Raw Signature Format' stores certificates and the signed digest directly in the signature dictionary as attributes. The 'PKCS#7 Signature Format' encapsulates the signature and the signed digest into a single PKCS#7 object that is stored in the dictionary. There are advantages and disadvantages to both formats. An advantage of the 'Raw Signature Format' is that the signature is more compact. A disadvantage of using PKCS#7 is the more required space for the PKCS#7 object and the containing not required redundant information.

Digital signature

NONE

No signature will be inserted.

Raw Signature Format (PKCS#1)

The raw signature format stores the signed digest and the signer's certificates directly as attributes in the signature dictionary.

PKCS#7 Signature Format

This syntax use a PKCS#7 object to encapsulate the document signature and signer's X.509 certificates.

 Both formats can be verified by the Adobe Reader.

Certificates

Currently only an automatically generated self signed certificate is used.

Following entries of the generated certificate can be configured:

Common name (CN=...)

e.g. *NPS PDF signer*

Organization name (O=...)

e.g. *My company Ltd.*

email address (E=...)

e.g. *MyEmail@unknown.com*

Appearance

Currently the signature will be always invisible and is always attached to the first page.

These are the user default settings for the PCL environment, i.e. usually the values which can be configured at the printer panel. These values are valid on power on and after a reset as long as the data stream does not overwrite any of these settings.

Paper Size

Select a paper size from the pull-down menu or enter a USER DEFINED paper size.

Determines the default paper size. Also a user defined paper size may be configured. In that case the Width and Height must be given always in portrait view regardless of the given default orientation.

- **Size for portrait in 1/300 Inch**

Define your own paper size.

Width and Height must be given always in portrait view regardless of the given default orientation.


Available only when USER DEFINED is selected as paper size.

Default: 2480/300 inches (Width), 3507/300 inches (Height)

- **edge to edge**

Enable to handle the paper size in full-page mode.

I.e. the PCL logical page size will be enhanced to the physical page size.

 A user defined paper size is always handled in full page mode.

Orientation

Select the default orientation.

Default: PORTRAIT

Page Copies

Enter default number of copies for each printed page.

Default: 1

Line Termination

Select the default control code behavior for the following control codes:

<CR> Carriage Return
<LF> Line Feed
<FF> Form Feed

Available options are:

- <CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>
- <CR> = <CR-LF> ; <LF> = <LF> ; <FF> = <FF>
- <CR> = <CR> ; <LF> = <CR-LF> ; <FF> = <CR-FF>
- <CR> = <CR-LF> ; <LF> = <CR-LF> ; <FF> = <CR-FF>

- **<CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>**

The Carriage Return command causes a carriage return, the Line Feed command causes a line feed, and the Form Feed command causes a form feed.

- **<CR> = <CR-LF> ; <LF> = <LF> ; <FF> = <FF>**

The Carriage Return command causes a carriage return line feed, the Line Feed command causes a line feed, and the Form Feed command causes a form feed.

- **<CR> = <CR> ; <LF> = <CR-LF> ; <FF> = <CR-FF>**

The Carriage Return command causes a carriage return, the Line Feed command causes a carriage return line feed, and the Form Feed command causes a carriage return form feed.

- **<CR> = <CR-LF> ; <LF> = <CR-LF> ; <FF> = <CR-FF>**

The Carriage Return command causes a carriage return line feed, the Line Feed command

causes a carriage return line feed, and the Form Feed command causes a carriage return form feed.

Default: <CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>

Duplex Mode

Select the default simplex or duplex printing from the pull-down menu.

Available options are:

- SIMPLEX
- LONG EDGE BINDING
- SHORT EDGE BINDING

Default: SIMPLEX

Output Destination

Set output destination (paper tray). These options are device dependent.

AUTOMATIC - Standard output bin (determined by the printer)

UPPER BIN - Usually the 'face-down' output bin

LOWER BIN - Usually the 'face-up' output bin

HC OUTPUTBIN to **HC OUTPUTBINS** - Output bins of a high capacity option

Default: AUTOMATIC

Symbol set

The default symbol set as long the data stream does not overwrite it.

Font

The default font as long the data stream does not overwrite it.

Resolution

The job page resolution as long the data stream does not overwrite it via @PJL SET RESOLUTION=...



The page resolution influences mainly the density of HPGL user defined patterns which are drawn in the current resolution.

Resource Path

Define the path from where the PCL5 to PDF emulation can access resources like fonts, etc.

Include JavaScript file

Enable to include JavaScript file.

Enter the name and path of the JavaScript file (.js) or search for it with the Browse function.

Default: disabled

Run emulation in static mode

Enable to run emulation continuously. Disable to finish emulation after each job.

This controls whether the PCL5 to PDF emulation is unloaded between the incoming print jobs (dynamic mode) or whether it remains loaded for the entire duration of the Print processes

(static mode).

Default: disabled

6.3.15 Job Controller

6.3.16 PCL to PDF

see PCL to PDF Properties inside chapter IPDS to PDF emulation.

6.3.17 PCL5 to PostScript® Emulation

6.3.17.1 Overview

The Mercury PCL5 to PostScript® emulation converts print data that were created for PCL printers so that they can be output on PostScript® page printers. At least a PostScript® Level 2 compatible printer is required.

Some Features

- High level PostScript® black and white, gray scale and Color code generation.
- PostScript® Level 3 commands are supported, if available.
- Supports special device features via ppd file.

6.3.17.2 Parameters

6.3.17.2.1 PostScript® Printer

Some notes on PPD files and special device features:

The device specific PostScript® printer features often are invoked by PostScript® code, which is special for a specific printer. This code is stored in a PPD-file (PostScript® Printer Description - file), which should be shipped with the drivers of every PostScript® printer.

Without having the correct PPD-file of your printer, there are only the device features available, which can be coded in standard PostScript® procedures.

If you choose the wrong PPD-file it may occur, that even these base features don't work because the special code inserted in the output data stream is not recognized by your printer.

Choosing the supplied generic PPD file should enable basic device features, because standard PostScript® code sequences are defined inside this ppd file.

But if you want to use features like Input- and Output bin mapping, punching, stapling etc. , you have to select the correct PPD file for your printer.

Even if you have selected the correct PPD-file, it may happen, that only basic device features are available. This can occur, if the printer manufacturer don't uses the intended keywords for finding the code for the enhanced device features in the PPD file.

In this case to things can be done:

1) Call the Mercury support, which will add the needed new code in the Mercury PCL5 to PostScript® emulation to support the enhanced capabilities of this printer.

2) You can create a local customized ppd file of your printer by adding a mapping option in this file, which replaces the manufacturer specific device keywords with the emulation internally used Adobe® conform keywords. You can do this by editing the ppd file with a text editor. You have to add the following statements at the end of the ppd file:

Syntax:

```
*OpenUI *KeyReplacements
*KeyReplacements <Adobekey1>: <Replacekey1>
.
.
*KeyReplacements <Adobekeyn>: <Replacekeyn>
*CloseUI *KeyReplacements
```

The following Example replaces the Keyword `EFOutputBin` used in the ppd file for Outputbin

selection with the emulation internally used Adobe®
conform keyword `OutputBin`:

```
*OpenUI *KeyReplacements
*KeyReplacements OutputBin: EFOutputBin
*CloseUI: *KeyReplacements
```

The following replaceable keywords are used inside the Mercury PCL5 to PostScript® emulation for bin/tray selection and finishing options:

```
Duplex
OutputBin
InputSlot
PunchEnable
PunchOpt
StapleLocation
Collate
Jog
```

At the beginning of each job all device options marked as default in the printers PPD file are written in the PostScript® output data stream. Since PPD files are editable ASCII text file, thus it is possible to set job global printer options which are not accessible through the GUI of the Mercury PCL5 to PostScript® emulation configuration dialog simply by changing the default options in the PPD file with a text editor.

You can find more information about structure and command syntaxes of a ppd file in the "**PostScript Printer Description File Format Specification**" from Adobe® available for free download via internet.

Storing this customized PPD file under a new name, creates a GUI-selectable special configured version of the printer.

Select PostScript® printer

Before you select your printer, you have to copy the printers PPD-file to the subdirectory `../ppd/` of the emulation resource directory.

This directory can be configured in the Resources configuration sheet.

After (re)starting the Mercury PCL5 to PostScript® emulation configuration dialog the Printer model name, which is supported by the PPD file should be added to the list shown in the "Select PostScript® printer" selection box.

After changing the selected printer, all device specific features in all sheets of the configuration dialog are set back to the default options.

Special printer specific settings:

Send only Black layer of separated CMYK images

Enable this option, if you send gray scale images in the black component of separated CMYK color images to black and white printers.

6.3.17.2.2 Input Bin/Tray Mapping

Input bin Mapping

Enter the PCL tray ID in the edit field below the mapping table and press the **Add Mapping** Button to add a assignment for this PCL input paper tray to a PostScript® printer Input slot. Then select one of the choices from the *PostScript printer inputslot* drop down list. The choices shown in this list depends on the chosen printer resp. the *InputSlot* entries in the printer PPD file. If the choices in the input slot drop down list are paper specifications, the assignment from this specification to a specific input tray number have to be done at the PostScript® printers control panel.

If you want to remove a existing mapping, enter the PCL tray ID used for this mapping in the

edit field below the mapping table and press the **Remove Mapping for this tray** Button.

If a PCL tray number is not assigned to a PostScript® printer input slot, automatic tray selection via pagesize is activated. That means the printer decides by means of the actual page size which input slot is used for the current page. The assignment of page sizes and input slot numbers can be done at the printer control panel.

If no PPD file is selected or the *InputsSlot* entry is not found in the PPD file the mapping list will be deactivated and automatic tray selection via paper size is activated.

If the **Use default input bin** Box is enabled all entries in the mapping table are ignored, and the printers default input tray handling is activated. Normally this is automatic tray selection via page size.

6.3.17.2.3 Output Bin/Tray Mapping

Output bin Mapping

Enter the PCL outputbin number in the edit field below the mapping table and press the **Add Mapping** Button to add an assignment for this PCL output bin to a PostScript® printer output bin. Then select one of the choices from the *PostScript printer outputbin* drop down list. The choices shown in this list depends on the chosen printer resp. the OutputBin entries in the selected printer PPD file.

If you want to remove a existing mapping, enter the PCL output bin number used for this mapping in the edit field below the table and press the **Remove Mapping for this bin** Button.

If a PCL output bin number is not assigned to a PostScript® printer output bin, the printers default output bin is used.

If a default output bin is selected in the printers PPD file, the selection of this setting is written to the PostScript® output data stream. Otherwise the default output bin configured at the printer control panel is used. Please consider, that sometimes other device options for example finishing options like punching and stapling require specific output bins.

If the **Use default output bin** option is enabled all entries in the mapping table are ignored, and the printers default output bin is used. If the *OutputBin* key is not found in the selected ppd file, the **Use default output bin** option cannot be disabled.

6.3.17.2.4 Job settings

Use pjl job header / trailer

Enable to include PjL job header commands before and PjL trailer commands after the first resp. last PostScript® commands in the output data stream.

Send PPD default page settings after document setup

Enable this option to write all page level device options marked as default in the printers PPD file in the PostScript® output data stream. The PostScript® code to select these options is written only once after the document setup and before the device setup of the first page.

The written options maybe overwritten by subsequent dynamically changed page level device options.

Binary image data output

If this option is enabled the image data is generated in binary format instead of ASCII base-85 coded. If the printjob contains large color image resources, activating this option can reduce the output data size. If the job contains big black and white only image resources the output size may be much bigger.

Because some printers and postscript postprocessing software experiences problems with binary postscript output, it is not recommended to activate this option as a default setting.

Save PCL image data in reusable streams, if possible (Postscript Level 3)

Activating this option can vastly reduce output data size at the expense of much higher memory usage in the postscript printer. If the print data contains image resources, which are repeatedly used on consecutive pages or sets, this data is stored only once in the printer memory instead of transmitting it on every page it is needed. This option works only with images stored in PCL Macros or if the images are supplied with URIDs, generated from the IPDS to PCL Emulation.

Use this option only, if the printer supports the */ReusableStreamDecode* filter option and if the printer is equipped with enough memory to hold all image resources of the job.

Use overprinting for multiple plane images

This option is active by default, because overprinting is the only way to print CMYK multiple plane images without using reusable streams for storing the image data for every plane. If this option is used, the printer have to support the overprinting feature.

If this option is disabled, the option above (Save PCL image data in reusable streams, if possible (Postscript Level 3)) is automatically enabled, and multiple plane CMYK images are printed with a multiple datasources image procedure.

Job splitting

Entering any page number greater than 0 starts splitting the job at the outport after each sequence of the specified number of pages. In the case of duplex printing, the splitting is delayed one page, if the split page number is reached at the frontside of a duplex sheet. If the option "**Append separator sheet (separator_sheet.ps) to each splitted part**" is activated and a postscript file named "separator_sheet.ps" is available in the global resource path of the emulation, this postscript file will be appended at the end of each splitted part of the output data. Please use the appropriate Windows postscript driver for the printer specified at the postscript printer settings to generate the file separator_sheet.ps.

6.3.17.2.5 Layout

Page content scaling

Scaling factor for page content. 100% means a 1:1 transformation of the PCL content. The origin of the zoomed page content is the lower left corner of the page.

Activating the option **Change papersize according to zoom factor** forces the emulation to select a papersize resulting from multiplying the original papersize with the selected scaling factor.

NOTE:

If resulting papersize is not defined as a PageSize option of the selected ppd file, the automatic tray selection via the pagesize may fail at the printer side !

Page content shift

Use this option to specify a additional horizontal (X:) and vertical (Y:) shift of the page content.

Separate values can be specified for the front- and, in the case of duplex printing, backsides of the print job. The origin is the lower left corner of the page. The unit of the values specified here is 1/72 inch.

6.3.17.2.6 Finishing

The most finishing options are only available, if the PostScript® printer PPD file supports this option and the keyword for activating this option is known by the Mercury PCL5 to PostScript® emulation. Because nearly every printer uses different keywords for the finishing options it maybe necessary to contact the Mercury support for adding a new printer driver to the PCL5 to PostScript® emulation.

Alternatively a local customized version of the PPD file can be created, which contains the necessary device options as default settings.

If the option keyword is not found in the printer PPD file, but there are standard PostScript® commands available to activate this option, the choices for this standard PostScript® commands will be displayed in the option drop down lists.

Please don't forget to check if your printer is equipped with the selected options.

Hole Puncher

The punching choices shown in the **Puncher choices** drop down list depend on the chosen printer model. If no punching option is found in the PPD file the drop down list will be empty and deactivated.

If you enable **Use settings from PCL data stream**, the selected choice in the Puncher choices drop down list will be ignored and PCL data stream supplied settings for punching will be used. Enable this option only, if you are sure, that valid puncher settings are transmitted in the PCL data stream.

Collate

The collate choices shown in the **Collate choices** drop down list depend on the chosen printer model. If no collate option is found in the PPD file the PostScript® default choices for this option are shown.

Default collate choices:

True Turn on collation.
False Turn off collation.

Collating is defined as follows: for three copies of a three page document, collated pages are produced in order 1-2-3-1-2-3-1-2-3, while uncollated pages are produced in the order 1-1-1-2-2-2-3-3-3.

Stapling

The stapling choices shown in the **Stapling choices** drop down list depend on the chosen printer model. If no stapling option is found in the PPD file the drop down list will be empty and deactivated.

If you enable **Use settings from PCL data stream**, the selected choice in the Stapling choices drop down list will be ignored and PCL data stream supplied settings for stapling will be used. Enable this option only, if you are sure, that valid stapling settings are transmitted in the PCL data stream.

Jog

The Jog choices shown in the **Jog choices** drop down list depend on the chosen printer model. If no jog option is found in the PPD file the PostScript® default choices for this option are shown.

Default jog choices:

None	Turn off jogging.
EndOfSet	Jog at the end of the current set.

When jogging is invoked, the next set is offset to the left or right from the previous set in the output bin.

Jogging is also known as "offset stacking".

6.3.17.2.7 Resources

The **Resource Path** window contains the current base resource path for the Mercury PCL5 to PostScript® emulation.

Inside this directory all resources, which are accessed during running the emulation, have to be placed.

The base path directory contains the PostScript® prolog file *base_prolog.ps* as well as the subdirectories *../fonts/* and *../ppd/*.

The *font* subdirectory at least contains the PostScript® encoding vectors (**.enc*), which are necessary to run the emulation.

Additionally the PostScript® font libraries (**.pfb*) for the HP Laserjet 4 standard fonts should be placed here if needed.

The PPD files of all printers, which will be used as output devices for the Mercury PCL5 to PostScript® emulation

have to be copied to the *ppd* subdirectory. Without the correct PPD file there will be no enhanced device options available.

With the **Browse...** button the base resource path can be changed. After changing the path, please don't forget to copy the necessary resources to the new directory resp. the subdirectories *../fonts* and *../ppd*.

6.3.17.2.8 PCL Defaults

These are the user default settings for the PCL environment, i.e. usually the values which can be configured at the printer panel. These values are valid on power on and after a reset as long as the data stream does not overwrite any of these settings.

Paper Size

Select a paper size from the pull-down menu or enter a USER DEFINED paper size.

Determines the default paper size. Also a user defined paper size may be configured. In that case the Width and Height must be given always in portrait view regardless of the given default orientation.

Size for portrait in 1/300 Inch

Define your own paper size.

Width and Height must be given always in portrait view regardless of the given default orientation.

Available only when USER DEFINED is selected as paper size.
Default: 2480/300 inches (Width), 3507/300 inches (Height)

edge to edge

Enable to handle the paper size in full-page mode.
I.e. the PCL logical page size will be enhanced to the physical page size.
A user defined paper size is always handled in full page mode.

Scale to default papersize

If this option is enabled the papersizes transmitted in the PCL data stream are ignored and the content of all pages of the job is scaled to the default papersize specified here.

Orientation

Select the default orientation.

Default: PORTRAIT

Page Copies

Enter default number of copies for each printed page.

Default: 1

Use default instead of job supplied value

If this option is enabled the job copy number specified in the PCL datastream is ignored and the number of copies specified here is used to determine the page copies of the postscript output.

Line Termination

Select the default control code behavior for the following control codes:

<CR> Carriage Return
<LF> Line Feed
<FF> Form Feed

Available options are:

<CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>

<CR> = <CR-LF> ; <LF> = <LF> ; <FF> = <FF>

<CR> = <CR> ; <LF> = <CR-LF> ; <FF> = <CR-FF>

<CR> = <CR-LF> ; <LF> = <CR-LF> ; <FF> = <CR-FF>

<CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>

The Carriage Return command causes a carriage return, the Line Feed command causes a line feed, and the Form Feed command causes a form feed.

<CR> = <CR-LF> ; <LF> = <LF> ; <FF> = <FF>

The Carriage Return command causes a carriage return line feed, the Line Feed command causes a line feed, and the Form Feed command causes a form feed.

<CR> = <CR> ; <LF> = <CR-LF> ; <FF> = <CR-FF>

The Carriage Return command causes a carriage return, the Line Feed command causes a carriage return line feed, and the Form Feed command causes a carriage return form feed.

<CR> = <CR-LF> ; <LF> = <CR-LF> ; <FF> = <CR-FF>

The Carriage Return command causes a carriage return line feed, the Line Feed command causes a carriage return line feed, and the Form Feed command causes a carriage return form feed.

Default: <CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>

Duplex Mode

Select the default simplex or duplex printing from the pull-down menu.

Available options are:
SIMPLEX
LONG EDGE BINDING
SHORT EDGE BINDING

Default: SIMPLEX

Use default instead of job supplied value

If this option is enabled the duplex commands specified in the PCL datastream are ignored and the default value specified here is used as the duplex setting for the postscript output.

Generate empty backside

Enable this option together with forcing the job to duplex (see option above), if you have to print mixedplex jobs and your printer is not able to change the duplex setting during printing the job.

Output Destination

Set output destination (paper tray). These options are device dependent.

AUTOMATIC - Standard output bin (determined by the printer)

UPPER BIN - Usually the 'face-down' output bin

LOWER BIN - Usually the 'face-up' output bin

HC OUTPUTBIN to HC OUTPUTBIN8 - Output bins of a high capacity option

Default: AUTOMATIC

6.3.18 PCL5 to X-Metacode

6.3.18.1 Overview

This document describes all the parameters of the Mercury [PCL5 to X-Metacode](#) emulation. All The parameters are default values. They are in use until their values are overridden from PCL5 commands in the online data stream.

In the configuration dialog for this emulation, you will find the following property pages:

- [Emulation](#)
- [Resources](#)
- [Default Font](#)
- [Xerox Printer](#)
- [Trays](#)

6.3.18.2 Parameter

6.3.18.2.1 Emulation

Orientation

Select **Portrait** or **Landscape** mode for default printing orientation.

Default: Portrait

Page Side

Select **Simplex** or **Duplex** printing for default page side.

Default: Simplex

Symbol Set

Select the default symbol set for printing.

Available symbol sets are:

- Roman 8
- PC-850
- Legal, ISO-ASCII
- ISO-Swedish
- ISO-Italian
- ISO-Spanish
- ISO-German
- ISO-Norwegian
- ISO-French
- ISO-Great-Britain
- ISO-8859-1 Latin 1
- PC8-Codepage 437
- PC8-Danish-Norwegian
- Windows Latin 1 / 3.0
- Windows Latin 1 / 3.1
- Ventura International
- Ventura US

Default: Roman 8

Emulation

Select the actual printer emulation.

Available options: PCL5, ASCII and XEROX META. In ASCII mode no PCL5 commands are interpreted. The XEROX META mode is a passthrough mode for special Xerox data streams.

CR / LF Filter

Select the default control code behavior for the following control codes:

<CR>	Carriage Return
<LF>	Line Feed
<FF>	Form Feed

This option can be used to replace carriage return, line feed and form feed commands with combinations of these commands. Especially ASCII data streams generated from UNIX applications sometimes need those replacements.

Available options are:

- <CR> = <CR> ; <LF> = <LF> ; <FF> = <FF>
- <CR> = <CR-LF> ; <LF> = <LF> ; <FF> = <FF>

- $\langle CR \rangle = \langle CR \rangle$; $\langle LF \rangle = \langle CR-LF \rangle$; $\langle FF \rangle = \langle CR-FF \rangle$
- $\langle CR \rangle = \langle CR-LF \rangle$; $\langle LF \rangle = \langle CR-LF \rangle$; $\langle FF \rangle = \langle CR-FF \rangle$
- **$\langle CR \rangle = \langle CR \rangle$; $\langle LF \rangle = \langle LF \rangle$; $\langle FF \rangle = \langle FF \rangle$**

The Carriage Return command causes a carriage return, the Line Feed command causes a line feed, and the Form Feed command causes a form feed, i.e. no replacements of CR, LF and FF.

- **$\langle CR \rangle = \langle CR-LF \rangle$; $\langle LF \rangle = \langle LF \rangle$; $\langle FF \rangle = \langle FF \rangle$**

The Carriage Return command causes a carriage return line feed, the Line Feed command causes a line feed, and the Form Feed command causes a form feed, i.e. CR is replaced with CR-LF, no replacements of LF and FF.

- **$\langle CR \rangle = \langle CR \rangle$; $\langle LF \rangle = \langle CR-LF \rangle$; $\langle FF \rangle = \langle CR-FF \rangle$**

The Carriage Return command causes a carriage return, the Line Feed command causes a carriage return line feed, and the Form Feed command causes a carriage return form feed, i.e. no replacements of CR, LF is replaced with CR-LF and FF is replaced with CR-FF.

- **$\langle CR \rangle = \langle CR-LF \rangle$; $\langle LF \rangle = \langle CR-LF \rangle$; $\langle FF \rangle = \langle CR-FF \rangle$**

The Carriage Return command causes a carriage return line feed, the Line Feed command causes a carriage return line feed, and the Form Feed command causes a carriage return form feed, i.e. CR is replaced with CR-LF, LF is replaced with CR-LF and FF is replaced with CR-FF.

Default: $\langle CR \rangle = \langle CR \rangle$; $\langle LF \rangle = \langle LF \rangle$; $\langle FF \rangle = \langle FF \rangle$

Default Tray

Select the default tray for feeding paper.

Only trays which have been made accessible in the Trays configuration page are available.

Default: 1 (Default accessible: 1, 2)

Copies

Define the default number of copies for each printed page.


Values from 1 to 999 are supported.

Default: 1

Edge to edge

Enable to use edge to edge printing.

Some PCL printer models can print the whole page, other models have unprintable areas at the edges of the page. This option allows you to select whether the entire page should be used for the output or whether unprintable areas at the edges of the page should be emulated.

 Printing over the entire page should only be enabled if it has been ensured that this option is supported by your application.

Default: disabled

Generate JDE entry

Enable to generate a JDE (Job Descriptor Entry).

This check box is only available in XEROX META mode. If enabled, a JDE entry call to the label

'XASC' is inserted into the data stream at the beginning of each job.

Run emulation in static mode

Enable to run emulation continuously. Disable to finish emulation after each job.


This controls whether the PCL5 to PDF emulation is unloaded between the incoming print jobs (dynamic mode) or whether it remains loaded for the entire duration of the Print processes (static mode).

Default: disabled

Keep downloaded softfonts

Enable to keep downloaded softfonts.

This check box enables the softfont management system of the emulation. The software will remember each softfont downloaded to the printer and prevent the downloading of the same font to the printer twice. The data stream from the emulation to the printer will decrease and the print performance will increase.

 It is recommended to use this option only in combination with the [IPDS to PCL5](#) emulation, because it must be guaranteed that the PCL5 softfont names are unique and that the PCL5 softfonts contain the complete character set necessary for printing the whole print job. Otherwise the softfont management system may fail. It is absolutely necessary to deactivate this option, if the PCL5 to X-Metacode emulation is used in combination with the XMCC / Mercury PCL printer driver.

Softfont prefix

Define or modify the Softfont prefix.

This edit field enables changing the font names the emulation uses to store softfonts at the hard drive of the printer. This allows the simultaneous work of several PCL to X-Metacode emulations under Mercury without overwriting of softfonts at the printer hard drive.

Changing of font names is only possible if the [Keep downloaded softfonts](#) option is enabled. If this option is disabled, all softfonts will be downloaded from the PCL to X-Metacode emulation to the printer using the name-prefix XSFT (i.e. XSFTAA.FNT, XSFTAB.FNT).

Optimize image download

Enable to optimize image download.

If enabled, the emulation remembers the images already sent to the printer. This prevents the multiple download of one and the same image in the same print job. Data volume to be sent to the printer is thus decreased, performance is enhanced. This optimization works only within a job and not for several jobs. The emulation deletes any information about downloaded images at the end of the job.

Default: disabled

6.3.18.2.2 Resources

Resource Path

Define the path from where the PCL5 to PDF emulation can access resources like fonts, etc.

Installation

- **Installed Resources**


This list shows the installed resource files.

- **Install...**

Click button to add new resources.

- **Delete**

Click button to delete the selected resource.

 Be cautious. The selected resource is completely deleted from the disk. Undo is not possible.

6.3.18.2.3 Default Font

These parameters specify the default font for the printouts.

Spacing

Default: Fixed

- **Fixed**

Select Fixed for the default font to have fixed spacing.

- **Characters per Inch**

Define the number of characters per inch, if the default font is a fixed spaced font.

Available values are: 5, 10, 12, 15, 17, 20, 27

Default: 10

- **Points**

Define the height of the characters in points, if the default font is a proportional font.

Available values are: 6, 8, 9, 10, 11, 12, 14, 16, 18, 24

- **Lines per Inch**

Define the number of lines which are printed per one inch on the page.

Changing this parameter increases or decreases the distance between two printed lines.

Available values are: 4, 6, 8, 12

6.3.18.2.4 Xerox Printer

These parameters are used to set the specific values for the Xerox printers. All DJDE and RSTACK settings on this page must be identical with the values configured on the Xerox printer itself.

Xerox Printer

Select the type of Xerox printer to adjust the PCL5 to X-Metacode emulation to the special capabilities of the various Xerox printer models.

The emulation supports Xerox printers 4050, 4090, 4850, 4890, 4135 and 4635.

Default: 4050

Enable Five Word FST Font

Enable for five word FST font.

If it has been enabled, a special Xerox font format is generated by the PCL5 to X-Metacode emulation, which supports font files without any size restrictions.

Option available for Xerox printers 4090, 4850, 4890, 4135, 4635.

Default: disabled

Graphic Compression

Select a graphic compression algorithm the software uses for generating Xerox IMG graphics from the pull-down menu.

The RAW algorithm will generate uncompressed graphics. Using graphic compression will decrease the data stream sent to the printer and increase print performance.

Available algorithms: LIN, HTN, RAW (none)

Default: RAW (none)

DJDE

• DJDE Definition

Define a character string to identify DJDE command lines.

This character string, also known as a 'prefix', may be entered in ASCII or EBCDIC characters and must not exceed a length of 20 characters. Changing the entry in ASCII automatically changes the entry for EBCDIC and vice versa.

Default: DJDE (ASCII), C4D1C4C5 (EBCDIC)

• DJDE Offset

Define a DJDE offset.

Indicates the number of bytes from the beginning of the record user's data portion until the beginning of the DJDE definition string. It is a numerical value which can vary from 0 to 99.

Default: 0

• DJDE Skip

Define a DJDE skip.

Specifies the number of bytes from the beginning of the record's user data portion until the beginning of the DJDE options. This numerical value must be greater than or equal to the output (prefix length + offset value + 1) and less than 128.

Default: (depends on the length of the ASCII string entered above)

RSTACK

• RSTACK Definition

Define a character string to classify a record as an RSTACK record.

The number of entered bytes must be less than 34 characters. The characters may be entered both as ASCII and as EBCDIC characters. Changing the entry in ASCII automatically changes the entry for EBCDIC and vice versa.

Default: RSTACK (ASCII), D9E2E3C1C3D2 (EBCDIC)

• RSTACK Offset

Define the column from which the RSTACK character string starts in the data record. The numerical value may be between 0 and 99.

Default: 1

• Generate RSTACK before each job

Enable to send out a RSTACK command to the printer before each job.

Enable this option, if more than one emulation sends jobs to one common printer.

Default: disabled

6.3.18.2.5 Trays

The paper handling and installed paper sizes of the [PCL5 to X-Metacode Emulation](#) can be configured with these parameters. All the values configured here must match with the values configured on the Xerox printer.

Tray

Select the paper input tray of the Xerox printer to configure it for the PCL5 to X-Metacode emulation. Values from 1 to 6 are supported.

Default: 1

Accessible

Enable to make the selected tray accessible for the emulation.

Default: (trays 1 and 2 accessible)

Identification

Enter a character string to identify the tray.

This string may have a maximum length of 10 characters.

Only available if [Accessible](#) is enabled.

Default: MAIN (tray 1), AUX (tray 2)

Paper Size

Select the installed paper size for the selected tray.

Only available if [Accessible](#) is enabled.

Available sizes are A4 (210 x 297 mm), Letter (216 x 279 mm), Legal (216 x 356 mm) and user-defined.

Default: A4 (210 x 297 mm)

Width

Displays the width of the selected paper size using the selected unit.

Select paper width if user-defined paper size is selected.

Only available if [Accessible](#) is enabled.

Default: 297 (A4)

Height

Displays the height of the selected paper size using the selected unit.

Select paper height if user-defined paper size is selected.

Only available if [Accessible](#) is enabled.

Default: 210 (A4)

Unit

Select the unit for displaying the chosen paper size.

Only available if [Accessible](#) is enabled.

Available units are inch, 1/1440 inch, centimeters.

Default: mm (for paper size default A4)

6.3.18.2.6 Performance

Data Processing

- **Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32 KB

- **Block size for output buffering**

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32 KB

Timing

- **Delay before next try on read error**

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read attempts.

Default: 1000 msec

- **Delay before next try on write error**

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.


This prevents a slow output port slowing down the system by frequently repeated write attempts.

Default: 10 msec

- **Timeout for shutdown**

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.

 Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

Bidirectional Communication

- **Produce ACK/NAK responses**

Enable to produce ACK/NAK responses.

When processing print jobs, which are received through a Channel Input Port and when processing IPDS print jobs, usually it is necessary to send feedback responses to the host system. In other situations these responses might not be applicable. For highest flexibility it can be controlled by this option whether feedback responses are sent to the host system or not.

Default: enabled


- **Send stop code to host after writing <n> sec**

Define the time in seconds after which a stop code is sent to the host.

This applies if a write access to the printer is not successfully completed within this time. Some hosts require responses within a specific time. If for some reason the printer goes offline or is turned off this might lead to problems with bi-directional communication.

Set to 0 to turn off this option.

Default: 0 (off)

 The effective time depends on the write timeout set at the output port. I.e. if at the output port a write timeout is set to 10 sec and here 15 sec are used, the second timeout after 20 sec will send the stop code. This is because the first timeout after 10 sec is still in valid range of 15 sec.

Jobs

- **Split job after <n> pages**

Enter number of page breaks after which a job shall be split into smaller parts.

For example, a print job with 5000 pages can be split into 50 smaller print jobs, each consisting of 100 pages. Also if no specific criterion exists for the end of a print job, and therefore all print jobs from the host system are processed in the Mercury system as a single job, a forced split can be achieved using this parameter.

This option is helpful if the printer is collecting print jobs completely before starting printing. If the job is split into smaller pieces, the time until the first page gets printed is reduced. But it must be sure that some queue optimization of the printer is disabled, if the partially printed job pieces have to be in the right order.

Default: 0 (no splitting)


- **Reconfigure printer after splitting**

Enable to re-initialize the output device at the start of a print job.

If this option is enabled, the emulation will conduct the initialization automatically each time a print job is split.

Default: disabled

6.3.18.2.7 Color

 The color options are only available for a PCL to X-Metacode (Color) emulation!

If the configured [Xerox Printer](#) is a model 4850 or 4890, the PCL5 to X-Metacode emulation can be configured to create color printouts to support the Highlight Color Printing feature of the

4850 and 4890 printers.



Please note that to create color output, it is essential that you use the Mercury printer driver in your application programs. If you use other printer drivers, the correct function of the Highlight color print cannot be ensured.

Enable PCL Color Support

Enable or disable creation of Highlight Color data streams for Xerox printers.

Default: disabled

Primary Color

Select what color should be used by the PCL5 to X-Metacode emulation to create Highlight Color printouts.

Xerox Highlight Color printers support printing with one additional color. This color depends on the installed colored toner of the printer.

Available colors are red, green, blue, magenta, cyan, ruby, violet, brown, gray, cardinal, royal.

Default: Blue

Color Mapping Mode

Select the method used to map the PCL colors on the Xerox Highlight color.

The following methods are available:

- Use always Primary color
- Use first color seen in job
- Use first color seen in page

- **Use always Primary color**

All PCL colors are mapped on the color defined in Primary color.

- **Use first color seen in job**

All PCL colors of one print job will be mapped on the first PCL color seen of the print job.

- **Use first color seen in page**

All PCL colors of one page will be mapped on the first PCL color seen of this page.



With this selection you may have to change the printer's toner during printing your job.

Mapping table for text colors

- **Source color**

Select a source color for the re-mapping of PCL colors for text, rectangles and shadings.

Available colors are black, red, green, yellow, blue, magenta, cyan, white.

Default: Black

- **Destination color**

Select a destination color for the re-mapping of PCL colors for text, rectangles and shadings.

Available colors are black, red, green, yellow, blue, magenta, cyan, white.

Default: Black

Mapping table for graphic colors

The mapping of PCL graphics to Xerox Highlight color graphics can be selected here. Each PCL graphic color can be mapped either to Xerox Highlight Color or to black or to white.

Available colors are white, yellow, magenta, red, cyan, green, blue, black.

Default: White

6.3.19 Prescribe to PCL Emulation

6.3.19.1 Overview

This chapter describes all the parameters of the Mercury emulation Prescribe to PCL5.

PRESCRIBE is a registered trademark of Kyocera Corporation

6.3.19.2 Parameters

6.3.19.2.1 Prescribe Defaults (FRPO)

The names enclosed in braces are the FRPO parameter names (Firmware RePrOgramm) of the origin printer.

e.g.

!R!

FRPO C1, 0; CMNT set default page orientation to portrait;

EXIT;

Page orientation (C1)

Determines the default page orientation.

Default Emulation (P1)

Note: The Prescribe emulation does not support the available emulations.

This option sets only the corresponding default font mode (FTMD) of the chosen emulation. The font mode (FTMD command) determines which attributes will be changed automatically when a bitmap font is selected by the FONT, ALTF or RPF commands.

1 = character spacing

2 = line spacing

4 = fixed or proportional spacing

8 = portrait or landscape orientation

P1 / Emulation mode / Font mode

0 / Line Printer / 15

1 / IBM Proprinter / 0

2 / Diablo 630 / 0

3 / Qume Sprint11 / 13

4 / NEC Spinwriter / 13

5 / Epson FX-80 / 0

6 / HP LaserJet Series II / 13

Font number (C5,C2,C3)

Determines the default font number.

Escape character (P9)

Determines the escape character for entering prescribe mode (usually R for !R!). This character may be entered directly or by decimal code.

Default cassette (R4)

Determines the default input tray (printer specific).

Default stacker (R0)

Determines the default output tray (printer specific).

CR action (P2)

Determines the behavior when a carriage return control code is received (<CR> = 0x0D).

LF action (P3)

Determines the behavior when a line feed control code is received (<LF> = 0x0A).

Symbol set (U6, U7)

Determines the default symbol set.

When no user defined combination of U6 (*country-code*) and U7 (*code-set*) values are used, i.e. **code-set < 50**, the [*index*] value (0...n) of the combo box corresponds to the FRPO U6 value and also to the *country-code* value of the INTL command.

NOTE: Syntax of INTL command : **INTL country-code, code-set;**

⚠ The emulation supports only HP LaserJet symbol sets regardless of the *code-set* value (*code-set is ignored as long code-set < 50*).

That means a INTL *country-code, code-set;* command, with *country-code* = [*index*] value (0...n) and *code-set* < 50, would select the same smbfol set as given within the combo box list.

⚠ If the value of **code-set** **>= 50** (either on INTL command or FRPO U7 parameter), additional available symbol sets of the HP LaserJet can be selected by following formula :

$$\mathbf{kind1 = (('code-set' - 50) * 100) + 'country-code'}$$

Refer to HP's 'PCL5 Comparison Guide' for available symbol sets and corresponding *kind1* values.

EXAMPLE:

To select the HP symbol set 'ISO 8859/10 Latin 6' (*kind1* value = 206 or corresponding PCL symbol set ID = 6N), the following values for *code-set* and *country-code* could be used:

- (1) *code-set* = 50, *country-code* = 206
- (2) *code-set* = 51, *country-code* = 106
- (3) *code-set* = 52, *country-code* = 6

i.e. the Prescribe command 'INTL 206, 50;' or 'INTL 106, 51;' or 'INTL 6, 52;' would select the symbol set 'ISO 8859/10 Latin 6'.

To set this symbol set as default, a small file could be generated and sent once as print data to the emulation :

```
!R!  
CMNT symbol set 'ISO 8859/10 Latin 6' as default;
```

FRPO U6, 6; CMNT *country-code*;
 FRPO U7, 52; CMNT *code-set*;
 EXIT;

NOTE: Other HP LaserJet symbol sets can be also selected with the command **CSET PCL-symbol-set-ID**; , e.g. CSET 6N; for 'ISO 8859/10 Latin 6'.

Comparison table of symbol set difference between Kyocera and HP LaserJet (index 0...9) :

<i>country-code</i> *	<i>Kyocera symbol set (in LaserJet emulation)</i>	<i>corresponding HP PCL symbol set ID (kind1)</i>	<i>actually used PCL symbol set ID</i>
0	ISO 6: US ASCII	0U (21)	8U (27)
1	ISO 25:French	0F (6) **	1F (38)
2	HP German	0G (7) **	1G (39)
3	ISO 4:United Kingdom	1E (37)	1E (37)
4	ISO 60:Norwegian version 1	0D (4)	0D (4)
5	ISO 11:Swedish	0S (19)	0S (19)
6	ISO 15:Italian	0I (9)	0I (9)
7	HP Spanish	1S (51) **	2S (83)
8	ISO 14: JIS ASCII	0K (11) **	0K (11) *
9	US Legal	1U (53)	1U (53)

⚠ * The symbol set index values from 0...9 are actually 7 bit encodings on HP LaserJet 4 or newer printers. But origin Kyocera printers uses for these symbol sets a 8 bit encoding where the upper half is identical to HP Roman 8 [<ESC>(8U)]. Therefore, for index 0, which is 'ISO 6: US ASCII' on Kyocera printers (PCL ID = 0U) , actually a Roman-8 encoding on HP LaserJet printers (PCL ID = 8U) is selected. This HP Roman-8 symbol set (8 bit encoding) is identical to the 'ISO 6: US ASCII' encoding on Kyocera printers (refer to Roman-8 symbol set chart below).

⚠ ** These symbol sets are obsolete and are not present on LaserJet 4 or newer models.

⚠ *** For differences refer to the following ISO Substitution table (source HP's 'PCL5 Comparison Guide')

ISO Substitution Table

This table provides a quick reference for the values of special characters contained in ISO (International Organization of Standardization) symbol sets. ISO symbol sets contain the same characters as the ASCII symbol set, except for the character positions listed in this table. For example, in the ISO 4 (United Kingdom) symbol set, the British pound sign replaces the number sign used in decimal position 35 of the ASCII symbol set.

Table B-1. ISO Substitution Characters

ISO	Name	ID	Decimal Character Equivalents											
			35	36	64	91	92	93	94	96	123	124	125	126
6	ASCII	0U	#	\$	@	[\]	^	·	{		}	-
4	United Kingdom	1E	£	\$	@	[\]	^	·	{		}	-
69	French	1F	£	\$	à	°	ç	§	~	μ	é	ù	è	-
21	German	1G	#	\$	§	Ä	Ö	Ü	·	·	ä	ö	ü	ß
15	Italian	0I	£	\$	§	°	ç	é	~	ù	à	ò	è	ì
11	Swedish for Names	0S	#	□	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
17	Spanish	2S	£	\$	§	í	Ñ	¿	·	·	°	ñ	ç	-
60	Norwegian version 1	0D	#	\$	@	Æ	Ø	Å	·	·	æ	ø	å	-
2	Int'l. Ref. Version*	2U	#	□	@	[\]	·	·	{		}	-
25	French*	0F	£	\$	à	°	ç	§	~	·	é	ù	è	-
	HP German*	0G	£	\$	§	Ä	Ö	Ü	·	·	ä	ö	ü	ß
14	JIS ASCII*	0K	#	\$	@	[¥]	·	·	{		}	-
57	Chinese*	2K	#	¥	@	[\]	·	·	{		}	-
10	Swedish*	3S	#	□	@	Ä	Ö	Å	·	·	ä	ö	å	-
	HP Spanish*	1S	#	\$	@	í	Ñ	¿	·	·	{	ñ	}	-
85	Spanish*	6S	#	\$	·	í	Ñ	Ç	¿	·	·	ñ	ç	-
16	Portuguese*	4S	#	\$	§	Ä	Ç	Ö	·	·	ä	ç	ö	°
84	Portuguese*	5S	#	\$	·	Ä	Ç	Ö	·	·	ä	ç	ö	-
61	Norwegian version 2*	1D	§	\$	@	Æ	Ø	Å	·	·	æ	ø	å	

*These symbol sets are becoming obsolete and are not recommended for future use and are not present on the HP LaserJet 4, 4M, 4Si, 4SiMx, 4L, 4ML, 4P, 4MP, 4PJ, 4LJ Pro, 4LC, 4 Plus, 4M Plus, 4V, 4MV, 5, 5M, 5P, 5L, 5Si, 5SiMx, 5Si Mopier, 6L, 6P, 6MP, LaserJet 1100, 2100, 4000, 5000, 8000, and 8100 series, Color LaserJet family, and DeskJet 1200C, 1600C printers.

Roman-8 symbol set chart (source HP's 'PCL5 Comparison Guide')

8U		Roman-8								(R8)						
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	'	p			NBS	—	â	Å	Á	Þ
	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	SOH	DC1	!	1	A	Q	a	q			À	Ý	ê	î	Ã	þ
	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	STX	DC2	"	2	B	R	b	r			Â	ý	ô	Ø	ã	·
	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	ETX	DC3	#	3	C	S	c	s			È	°	û	Æ	Ð	µ
	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	EOT	DC4	\$	4	D	T	d	t			Ê	Ç	á	â	ð	¶
	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	END	NAK	%	5	E	U	e	u			Ë	ç	é	í	Í	¾
	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	ACK	SYN	&	6	F	V	f	v			Î	Ñ	ó	ø	Ì	—
	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	BEL	ETB	'	7	G	W	g	w			Ï	ñ	ú	æ	Ó	¼
	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	BS	CAN	(8	H	X	h	x			´	ı	à	Ä	Ò	½
	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	HT	EM)	9	I	Y	i	y			`	ı	è	ì	Õ	¾
	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A	LF	SUB	*	:	J	Z	j	z			^	ı	ò	Ö	õ	o
	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	VT	ESC	+	;	K	[k	{			"	£	ù	Ü	Š	«
	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C	FF	FS	,	<	L	\	l				~	¥	ä	É	š	■
	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D	CR	GS	-	=	M]	m	}			Ù	§	ë	ï	Ú	»
	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E	SD	RS	.	>	N	^	n	~			Û	f	ö	ß	ÿ	±
	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F	SI	US	/	?	O	_	o	☒			£	ç	ü	Ô	ÿ	
	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

Top margin (A1, A2)

Determines the default margins.

Left margin (A3, A4)

Determines the default margins.

Page length (A5, A6)

Determines the default margins.

Page width (A7, A8)

Determines the default margins.

Lines / Inch (U0, U1)

Determines the default line spacing.

Chars / Inch (U2, U3)

Determines the default character spacing when fixed spaced default font is used.

6.3.19.2.2 Page Offsets

With these setting the complete page may be shifted.

Visible window short edge offset

Shifts the logical page. This value has the same effect like the PCL top offset registration command `<ESC>&I#Z`

Visible window long edge offset

Shifts the logical page. This value has the same effect like the PCL left offset registration command `<ESC>&I#U`

Vector short edge offset

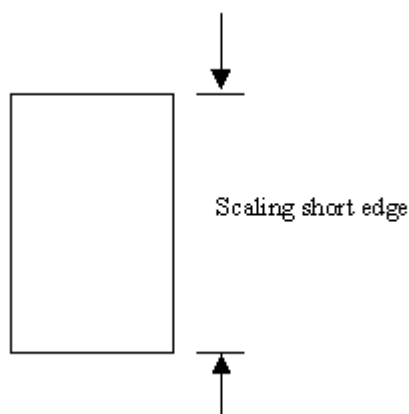
This offset will be added to each positioning command in the context of the short paper edge.

Vector long edge offset

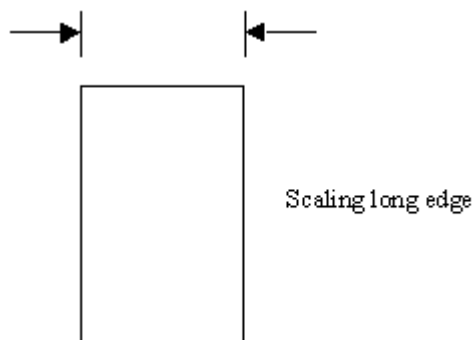
This offset will be added to each positioning command in the context of the long paper edge.

Scaling short edge

If the used printer does not support edge to edge printing, this option can be used to reduce the paper size by the given percentage.

**Scaling long edge**

If the used printer does not support edge to edge printing, this option can be used to reduce the paper size by the given percentage.



6.3.19.2.3 Duplex Offsets

These offsets are only on duplex back sides valid. They will be added to each positioning command in the context of the paper edge.

The values may be given separately for the duplex mode and page orientation.

6.3.19.2.4 Resources

Resource path

Determines the resource directory for optional resources.

Run emulation in static mode

This switch determines whether the emulation will be loaded and rejected between each print job (dynamic mode) or if it will be hold in memory during the complete lifetime of the print process (static mode).

6.3.19.2.5 Miscellaneous

Edge to edge

Choose the proper printer type in order set the printer automatically in edge to edge mode.

Swallow PCL sequences

Determines if PCL sequences within the prescribe data stream are discarded or not.

- **Off**

All PCL commands will be given transparently to the printer.

- **All**

All PCL commands will be discarded.

- **PCL within prescribe**

Only PCL commands within prescribe are discarded. e.g. (<ESC> stands for the control code 0x1B)

```
!R! MAP 1,1; <ESC>(8U EXIT;
```

- **PCL outside prescribe**

Only PCL commands outside prescribe are discarded. e.g.

```
<ESC>(8U !R! MAP 1,1; EXIT;
```

Formfeed behavior

Determines how pages will be separated on encountering a Form Feed (0x0C).

- **Formfeed**

The control code 0x0C will be used. This may result into empty pages on consecutively form feeds.

- **Printer reset**

A complete reset sequence will be sent to the printer. Empty pages are avoided but each page might be handled like a print job.

- **Select front page command**

The PCL duplex page side selection command <ESC>&a1G will be used.

Handling of download characters

Determines the behavior with download soft fonts (LDFC command), i.e. the RAM usage of the emulation at runtime.

- **Discard never**

The font will be send to the printer and stored permanently within the RAM of the emulation. i.e. the font may be changed any time with the SFA command.

- **Discard after SFA command**

The font will be released after the first SFA command. i.e. a second SFA command applied to the same font has no effect.

- **Do not store**

The font will be immediately released after sending to the printer. i.e. a following SFA command has no effect.

6.3.20 Program Interface

6.3.20.1 Overview

The Program Interface Emulation (PIE) enables Mercury to run external programs which emulate, process or change print data in any way. The so modified data can be sent back to Mercury (default) or be put out from the external program itself, respectively.

PIE runs fully automatically and requires no user intervention or monitoring.

6.3.20.2 Parameters

6.3.20.2.1 Miscellaneous

Redirect the data to the program

Define the external application or search for it using the '...' button.

Arguments for this program are

Set program options.

The following fields can be used as variables:

<tempin>	Name of temporary input file
<tempout>	Name of temporary output file
<dspconfig>	Name of configuration file
<emuid>	Emulation-ID
<jobname>	Name of the job
<jobnamenoext>	Name of the job (if filename without file extension)
<srcaddress>	Address of senders as IP address or Novell queue name
<username>	Name of user
<destport>	Destination port as queue name from LPD protocol

<destaddress>	Specific device if defined in Logical Printer
<jobid>	Job-ID
<ticks>	Milliseconds since system start
<date>	Date
<time>	Time
<creationtime>	Start time of job
<starttime>	Emulation time of job
<lpd_c>	LPD control file 'C' command (Class name)
<lpd_j>	LPD control file 'J' command (Job name)
<pages>	number of pages in the job as set in the jobinfo
<printedpages>	number of printed pages (i. e. confirmed by printer) in the job as set in the jobinfo (e. g. after a re-rprint)

Output

Define whether the modified data is further processed by Mercury or by the external program. Set priorities depending on the task.

Available options are:

- Copy temporary file
 - Copy stdout
 - Copy stderr
 - Program handles output
- Default: Copy temporary file

- **Copy temporary file**

Copy the temporarily created file.

- **Copy stdout**

- **Copy stderr**

- **Program handles output**

Select to let the external program handle the output.

Run

Select the mode in which you wish the external program to be displayed on the screen.

Available options are [Normal](#), [Minimized](#), [Hidden](#).

Default: Hidden

Priority

Select how much computing time shall be assigned to the external program. Capacities can thus be used optimally.

Available options are [Idle](#), [Below Normal](#), [Normal](#), [Above Normal](#).

Default: Normal

Stdout to log

Enable to select how the external application's output is documented in the log file.

Default: disabled

Available options are [Info](#), [Extended Info](#), [Warning](#), [Error](#), [Debug](#).

Default option: Info

Stderr to log

Enable to select how the external application's output is documented in the log file.

Default: enabled

Available options are [Info](#), [Extended Info](#), [Warning](#), [Error](#), [Debug](#).

Default option: Error

Job Timeout

Sets job timeout in seconds. After the timeout the plugin is called with an additional '/flush' parameter. Enabled only if plugin supports it.

Default: 0

6.3.20.2.2 Performance

Data Processing

- **Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32 KB

- **Block size for output buffering**

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32 KB

Timing

- **Delay before next try on read error**

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read attempts.

Default: 1000 msec

- **Delay before next try on write error**

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.

This prevents a slow output port slowing down the system by frequently repeated write attempts.

Default: 10 msec

- **Timeout for shutdown**

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.



Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

6.3.21 Raw to Raw

6.3.21.1 Overview

The Raw emulation does not convert the print data, but simply passes them on unmodified.

6.3.21.2 Parameters

6.3.21.2.1 Common Behavior

Count Pages (PCL5, PCL6, PDF, PostScript, Prescribe, RPCS or ASCII only)

Enable to count the pages of each processed print job for showing actual page count on the display.



Input port data type recognition must be enabled! This setting is part of the [Input Interface](#) configuration dialog, tag [Data Type](#).

Default: disabled

- **Split jobs after <n> pages**

Define the number of pages after which the emulation shall split a job.

Default: 0 (no splitting)

Run emulation in static mode

Enable to run emulation continuously. Disable to finish emulation after each job.

This controls whether the emulation is unloaded between the incoming print jobs (dynamic mode) or whether it remains loaded for the entire duration of the Print processes (static mode).

Default: disabled

6.3.21.2.2 Performance

Data Processing

- **Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32 KB

- **Block size for output buffering**

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32 KB

Timing

- **Delay before next try on read error**

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read attempts.

Default: 1000 msec

- **Delay before next try on write error**

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.

This prevents a slow output port slowing down the system by frequently repeated write


attempts.

Default: 10 msec

- **Timeout for shutdown**

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.

 Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

Bidirectional Communication

- **Produce ACK/NAK responses**

Enable to produce ACK/NAK responses.

When processing print jobs, which are received through a Channel Input Port and when processing IPDS print jobs, usually it is necessary to send feedback responses to the host system. In other situations these responses might not be applicable. For highest flexibility it can be controlled by this option whether feedback responses are sent to the host system or not.

Default: enabled


- **Send stop code to host after writing <n> sec**

Define the time in seconds after which a stop code is sent to the host.

This applies if a write access to the printer is not successfully completed within this time. Some hosts require responses within a specific time. If for some reason the printer goes offline or is turned off this might lead to problems with bi-directional communication.

Set to 0 to turn off this option.

Default: 0 (off)

 The effective time depends on the write timeout set at the output port. I.e. if at the output port a write timeout is set to 10 sec and here 15 sec are used, the second timeout after 20 sec will send the stop code. This is because the first timeout after 10 sec is still in valid range of 15 sec.

6.3.22 VIPP Emulation

6.3.22.1 Overview

The [Mercury VIPP®₁](#) (Variable data Intelligent PostScript® Printware) and Postscript® processor converts data streams that have been created for VIPP® capable centralized PostScript® printers into a complete self-describing and all resource containing data stream, that can be output to any decentralized PostScript® printer.

In other words, the VIPP® emulation serves as resource manager and VIPP® command interpreter, it adopts the role of a centralized VIPP® printer.

Additionally this emulation is capable of inserting dynamic overlays into Postscript and ASCII

text datastreams (Additional licence needed).

Some Features

- Supports special device features via ppd file.
- Different output formats, e.g. Postscript®, PDF and (limited) PCL5 support.
- supports all 4 major VIPP® modes (native mode, linemode, database mode and XML mode)
- supports decomposition service functions for generating tiff image VIPP® resources from postscript or PDF documents.
- eps, jpg and tiff resource pass through option for generating smaller postscript output and higher conversion speed.
- highly customizable via additional VIPP® or postscript resource files.
- Overlaymanager for Postscript and ASCII text input data (additional licence needed).

Things you have to consider before starting the emulation

Besides the variable data, usually for the conversion of a VIPP® print job various static resource files are needed.

It is necessary to retrieve all resources belonging to the variable data stream and transferring this resource files

to the appropriate [Mercury](#) resource folders.

Detailed information about this issue can be found in the chapter [VIPP Resources](#).

The [Mercury](#) VIPP® emulation makes usage of the free available postscript level3 interpreter GPL Ghostscript. Please install the recommended version of this software before starting the emulation.

Contents of this manual

In the Chapter **Parameters** you find detailed descriptions of the configuration options available in the graphical user interface of the emulation.

The Chapter **VIPP® Resources** includes all necessary information on how to find and transfer the static resources from the original Xerox® installation to the [Mercury](#) resource folders.

Detailed descriptions of all VIPP® and PostScript® commands are not part of this manual. If you want to learn more about VIPP® and PostScript® please use the reference manuals, which are available for free over the Internet. For VIPP® have a look at the **VIPP® Reference Manual**, for PostScript® please use the **PostScript® Language Reference third edition**.

1) VIPP is a registered trademark of the Xerox® Corp in the USA and throughout the world.

6.3.22.2 Parameters

6.3.22.2.1 Common

Output driver

Selects the format of the converted data stream.

If '**Disable output**' is selected the jobs will be processed but the output is discarded. This is important to find missing resources or to map data stream requests to Device features.

If '**Postscript® output**' is selected the generated Postscript® output consists only of low level Postscript® commands i.e. many bitmap images, so the size of the output data can get rather big, but this insures the compatibility with almost any Postscript® printer. This output driver is only available with Ghostscript versions up to 9.07. If you have installed more than one Ghostscript Version, the [Mercury VIPP®](#) and Postscript processor will automatically select the appropriate (older) Ghostscript version, if this Postscript® driver is selected.

If '**Postscript® output (high level)**' is selected the output is created with high level postscript commands, therefore the code is about ten times smaller but may be less compatible with some postscript printers.

If '**PDF output**' is selected the job will be converted to the Portable Document Format (PDF). This driver delivers the smallest output file size.

Additionally drivers for Black & White as well as Color PCL5 output are available (**B&W PCL (HP LJ4 compatible)** , **Color PCL (HP CLJ 5/5M compatible)**).

Currently these drivers do not support landscape paper formats. Printing rotated (landscape) page content on portrait sized paper is possible. Edge to Edge mode is supported, but only if this feature is activated also at the printer side by sending the correct PJI command, which have to be supplied and selected in the ppd file. These drivers supports Simplex/Duplex printing as well as Input media and Output bin mapping. If you choose one of the PCL5 output drivers please select the special supplied PCL ppd file **genericPCL.ppd**.

⚠ The Color PCL driver is not implemented by default in the standard version of the Ghostscript postscript interpreter.

⚠ Device specific features, e.g. Input slot mapping, duplex or finishing, might not be available due to the nature of the chosen data stream (e.g. PDF).

Drivers for special purposes:

The '**Text extract**' output driver can be used to extract the literal text from postscript or pdf input data. The format of the output data will be plain ASCII text.

If you choose '**Windows Printer**' as Output driver all data will be converted to bitmap page images. Then it is sent to one one of the installed Windows printers.

The specific printer can be chosen using the *Printer* drop down list. There is only limited support of device specific configuration options, when using this output driver. Additionally you have to redirect the output from the Windows printer to a new [Mercury](#) input interface, if you want to do the further data processing within the [Mercury](#) printing system.

The VIPP® **PDFDEVICE variable** is used by some vipp data for generating different output depending on the output format. The original Xerox VIPP® implementation sets this variable always to "true", if PDF output is generated. Using the [Mercury VIPP®](#) implementation, this variable can be set manually.

⚠ For the 'PDF output', 'Text extract' and 'Windows Printer' Output drivers a additional license is needed.

See also [Output driver settings](#)

Printer

The device specific PostScript® printer features often are invoked by PostScript® code, which is special for a specific printer. This code is stored in a PPD-file (**PostScript® Printer Description** - file), which should be shipped with the drivers of every PostScript® printer.

Without having the correct PPD-file of your printer, the device specific requests detected in the input data are converted into standard PostScript® code sequences. This Postscript® code is inserted in the output data at the beginning of the appropriate pages. If the printer does not accept this standard Postscript® code, the desired device feature, like for example input slot selection is not executed at the printer side.

If common device dependent features like input media and output bin selection, finishing, etc. does not work, you have to install and select the correct PPD file for your printer.

Before a printer can be selected, a PPD-file must be copied into the global resource path configured for the emulation. After (re)starting the VIPP® emulation configuration dialog, the printer model name, which is supported by the PPD file, should be shown within the list of the selection box.

⚠ PPD files must be copied into the root directory of the [Global resource path](#).

After selecting the ppd file for your printer, it is necessary to map the device requests found in the input data stream to the suitable device feature options available in the ppd file. For more information regarding mapping device options please see [Input media mapping](#).

Additionally to the page specific device requests, which can be configured in the mapping tables, at the beginning of each job all device options marked as default within the printers PPD file are written into the PostScript® output data stream.

Since PPD files are editable ASCII text file, thus it is possible to set job global printer options which are not accessible through the GUI of the VIPP™ emulation configuration dialog, simply by changing the default options in the PPD file with a text editor.

This process is called "local customization" of the PPD file. Storing this customized PPD file under a new name, creates a GUI-selectable special configured version of the printer.

⚠ Even if a PostScript® printer with the correct PPD-file has been selected, it may happen that nothing is selectable within the mapping dialogs. This can occur, if the printer manufacturer did not use the intended standard keywords (e.g. **InputSlot** for the available input media) for finding the code of the enhanced device features within the PPD file. If this happens, please have a look at the chapter [Handling of ppd files with non standard keywords](#)

- ⚠ For PCL Output a special supplied PCL Printer ppd file must be used (*genericPCL.ppd*)
The Modelname specified in this ppd file is *GenericPCLPrinter*. Do not select a PostScript® printer ppd file, if the PCL output drivers are used !
- ⚠ After changing the selected printer, all device specific mappings in all sheets of the configuration dialog are lost (invalid) and must be reconfigured.
- ⚠ For PDF output format the contents of the selected ppd file is not used.

Generate decomposition service tiff images from PS or PDF input

If you select any option other than "*Decomp Service off*" from this drop down list, the normal *output driver* and *Printer* settings are disabled. If you supply a postscript or pdf document to the [Mercury](#) input interface of your [Mercury](#) logical printer , each page of this document is converted to a single tiff (and postscript) resource. The names and destination folders for the output files are automatically generated according the conventions of the Xerox® decomposition service.

- Use one of the "TIFF..." options, if the decomposition service image resources are called directly from your VIPP® project.
- Use one of the "PS+..." options if your VIPP® project uses decomposition service postscript files to call the image resources.
- Use the "PS+2* tiff G4..." option if your VIPP® project uses decomposition service postscript files which calls 2 image files. The first one will be printed with the *highlightcolor*, the second one with the actual foregroundcolor. In this case the pdf or postscript input document must contain only two pages!

You can change the used *highlightcolor*, if you add a line in a additional resource file (see [Inserting additional resource file](#)), which contains the following postscript definition of the *highlightcolor* key:

```
statusdict /highlightcolor {rvalue gvalue bvalue} put  
rvalue, gvalue and bvalue are numeric values between 0 and 1.
```

Input data compatibility level

Please choose here, which version of the original Xerox VIPP® interpreter was used to generate your input data.

Additional you have the option to set the input data level to **Raw postscript conversion**.

Choose this compatibility level if your input data contains almost or no VIPP® commands at all and therefore do not comply with the minimal standards for a VIPP® data stream.

Use the option **Disable all VIPP Features**, if you want to convert postscript or PDF input data without any additional VIPP commands.

Use the option **Enhanced PDF processing**, if you want to convert PDF input data to postscript and additionally have to set some printer specific attributes like duplex mode or media select commands. E.g. if you pack the command lines

```
DUPLEX_on
```

and

```
(mymedia) WRITEMEDIA
```

in a additional resourcefile (see [Inserting additional resource file](#)), you can generate postscript output which sets the printer to duplex mode and demands "mymedia" as Mediatype at the printer site.

In the **Plain text** mode, the emulation accepts only plain text as input such as linemode and csv data. The data will be converted using the font and page configuration set in the [PS interpreter defaults](#) "*Linemode and Plain text input*" settings. In this input mode the [Postscript textscanner](#) and [Postscript overlays](#) can be used to replace text and insert overlays.

Copies of input data

Select values greater 1 if you want to generate more than one copy of the converted data stream.

⚠ If you don't activate the "*Split job at the end of a set (after ENDOFSET is called)*" option in the [Output driver settings](#), you will get one output datastream consisting of the number of selected copies. Splitting the copies to single streams is not available for the PDF output driver.

6.3.22.2 Resources

Besides the variable data, usually for the conversion of a VIPP® print job various static resource files are needed.

It is necessary to retrieve all resources belonging to the variable data stream and transferring this resource files to the appropriate [Mercury](#) resource folders.

Detailed information about this issue can be found in the chapter [VIPP Resources](#).

The '[Global resource path](#)' specifies the location of all static VIPP resources, which are necessary to convert the print data. You select the common VIPP® resource subdirectory (...res\vipp) of the [Mercury](#) installation folder by browsing with the attached **Browse...** button to it.

The option '[Use VIPP encoding file](#)' is used to insert a specific encoding vector file to adjust language specific characters.

⚠ The selected encoding file must reside within the **encoding** directory of the "global resource path", it is not possible to use a file from a other path.

You can also use one of the internal encoding vectors. The following internal encoding vectors are available:

- The **__SUN8__** encoding vector is identical with the PostScript "ISOLatin1Encoding" encoding except the additional € character a position 128. The array provides information about the encoded character set for key positions 128 through 255 for Latin text fonts produced by Adobe. This is the character set used by most Windows and UNIX systems.
- The **__PC8__** character set provides information about the encoded character set for key positions 128 to 255 for DOS operating systems. Only those characters present in the PostScript Standard Roman Character Set are encoded.
- The default **__PCSUN__** encoding table provides information about the encoded character set for key positions 128 through 255. Key positions 128 through 175 replicate the corresponding key position in the **__PC8__** font encoding table. Key positions 176 through 255 replicate the corresponding key positions in the **__SUN8__** font encoding table.
- The **__EBCDIC__** character table contains translation tables for ebcdic encoded data streams.

To select one of the internal encoding tables press the **Browse...** button and enter the name of the desired encoding table in the file select edit box.

For more information on VIPP fonts and encoding vectors please see [Additional Fonts and Encoding vectors](#).

Activate the option **Use external "fontlist" file**, if fontlist files named "fontlist" and "nullfi" (without extension) have to be used for converting your data. If this option is not activated the internally stored default "fontlist" and "nullfi" files are used. If you use your own fontlist files, please copy this files and all non standard font libraries used by these files to the `..lencoding` subfolder of the Global resource path.

The option '**Use additional VIPP or PostScript® resource file**' is used to insert a new resource file, which will be interpreted on emulation startup. You select the file by browsing with the attached Browse... button to it. You can create a new file or edit a existing file by starting the integrated editor with the **Edit** Button. For more information see [Inserting additional resource file](#).

The option '**Resource path used for temporary files and emulation specific resources**' is used to store temporary files created during conversion and emulation specific resources. Also the mapping files (*.map), e.g. for input media mapping, are stored into (and read from) this resource path. The resource path is individual selectable for each emulation.

With enabling the **Use project path** option, it is possible to use a project path for all project specific resources, even if this option is not activated in the VIPP® variable data files of the project. To use this option first create a resource directory for your VIPP® project resources and then select this path with the **Browse...** button. The necessary SETPROJECT VIPP® command is inserted automatically at the beginning of the variable datastream when the emulation is started.

Please keep in mind that fonts, which are referenced inside the VIPP resources by their original postscript fontname can only be found by Ghostscript if the font library files are stored in the `..lencoding` subdirectory of the *Global resource path* .

The '**Run emulation in static mode**' option controls whether the emulation is unloaded between the incoming print jobs (dynamic mode) or whether it remains loaded for the entire duration of the Print processes (static mode). Default is disabled.

Global resource path

Global resource path

All customer specific VIPP® resources must be copied into the global resource path. This path is automatically detected and should usually not be changed. All VIPP® search paths will be set based on the "global resource path".

Below this path exists following subdirectory structure :

encoding ;font encoding files (part of the path of SETEPATH)
formlib ;forms (*.frm), segments (*.seg), data base manager (*.dbm) files (part of the path of SETFPATH)
imglib ;images like *.tif, *.jpg, *.bmp, etc. (part of the path of SETIPATH)
jdllib ;JDT libraries (part of the path of SETJPATH)
mislib ;miscellaneous files (part of the path of SETMPATH)

⚠ The project path (SETPPATH) will be set also based on the "global resource path", e.g.
 [("global resource path"\\$\$FOLDER.\\$\$PROJECT.\)
 ("global resource path"\\$\$FOLDER.\fshared\
 ("global resource path"\gshared\)] SETPPATH

If for instance the VIPP® command **[(projects) (test)] SETPROJECT** is used, the directories "global resource path"**projects\test** must be created and all resources (forms, images, JDT libraries etc.) must be copied into ...**projects\test**.

⚠ Font resource files also have to be copied to the ..\encoding sub directory, even if a project path is used .

⚠ If the SETEPATH,SETFPATH,SETIPATH,SETJPATH,SETMPATH or SETPPATH is used within the VIPP® project resources, the path specified there overrides the default Global resource path setting.

Hint: You can disable the resource path specification commands by inserting e.g. for the SETIPATH command the following line in a additional resource file:

```
/SETIPATH {pop} def
```

See also VIPP® Resources / [Finding and Preparing the VIPP® resources](#)

6.3.22.2.3 Input media mapping

The VIPP® input media request (left column) will be automatically filled from the emulation whenever a new request is encountered during processing of a job. You can map each new request (left column) to a device feature entry available in the option list for the selected printer (by clicking on the right column).

⚠ If no PostScript® printer is selected or the **InputSlot** entry is not found within the PPD file, the mapping choices list will be empty, when clicking on the right column. If this happens please see PostScript® printer topic in the [Common](#) tab.

In the case of no mapping is done or no PostScript® printer is available, a standard PostScript® command will be placed in the PostScript® output for this VIPP® request.

Example:

A VIPP® data stream included a VIPP® input media request 'YELLOW' for the Mediacolor. (In the VIPP project data this could be done by inserting the command (:YELLOW:) SETMEDIA)

Preconditions:

The PPD file of the PostScript® printer is copied to the global resource path, the PostScript® printer is selected in the '*PostScript® printer*' option in the '*Common*' tab of the emulation and **the data stream is processed at least one time with a PostScript® output driver.**

During the conversion a "*new device specific action encountered*" warning occurs in the log window.

After the conversion the VIPP® input media request '*()(YELLOW)()*' occurs in the left column in the '*Input media mapping*' tab of the emulation.

You have to click in the right column to select the desired device feature entry from the option list available for the selected printer.

Now every time the VIPP® input media request '*YELLOW*' is detected in the input data, it is mapped to the selected entry on the right side of the mapping table.

⚠ To delete all input media request entries (left column + right column) from the mapping table, you have to delete the appropriate mapping file. This file is named *InputSlot.map* and is located in the *Resource path used for temporary files and emulation specific resources*.

6.3.22.2.4 Output bin mapping

The VIPP® output bin request (left column) will be automatically filled from the emulation whenever a new request is encountered during processing of a job. You can map each new request (left column) to a device feature entry available in the option list for the selected printer (by clicking on the right column).

⚠ If no PostScript® printer is selected or the ***OutputBin entry*** is not found within the PPD file, the mapping choices list will be empty, when clicking on the right column. If this happens please see PostScript® printer topic in the [Common](#) tab. In the case of no mapping is done or no PostScript® printer is available, a standard PostScript® command will be placed in the PostScript® output for this VIPP® request.

⚠ To delete all Output bin request entries (left column + right column) from the mapping table, you have to delete the appropriate mapping file. This file is named *OutputBin.map* and is located in the *Resource path used for temporary files and emulation specific resources*.

6.3.22.2.5 Staple mapping

The VIPP® staple request (left column) will be automatically filled from the emulation whenever a new request is encountered during processing of a job. You can map each new request (left column) to a device feature entry available in the option list for the selected printer (by clicking on the right column).

⚠ If no PostScript® printer is selected or the ***StapleLocation*** entry is not found within the PPD file, the mapping choices list will be empty, when clicking on the right column. If this happens please see PostScript® printer topic in the [Common](#) tab. In the case of no mapping is done or no PostScript® printer is available, a standard PostScript® command will

be placed in the PostScript® output for this VIPP® request.

⚠ To delete all Staple request entries (left column + right column) from the mapping table, you have to delete the appropriate mapping file. This file is named *Staple.map* and is located in the *Resource path used for temporary files and emulation specific resources*.

6.3.22.2.6 Jog mapping

The VIPP® jog request (left column) will be automatically filled from the emulation whenever a new request is encountered during processing of a job.

You can map each new request (left column) to a device feature entry available in the option list for the selected printer (by clicking on the right column).

⚠ If no PostScript® printer is selected or the '**Jog**' entry is not found within the PPD file, the mapping choices list will be empty, when clicking on the right column. If this happens please see PostScript® printer topic in the [Common](#) tab. In the case of no mapping is done or no PostScript® printer is available, a standard PostScript® command will be placed in the PostScript® output for this VIPP® request.

⚠ To delete all Jog request entries (left column + right column) from the mapping table, you have to delete the appropriate mapping file. This file is named *Jog.map* and is located in the *Resource path used for temporary files and emulation specific resources*.

6.3.22.2.7 PS interpreter defaults

Paper size

The default paper size is used if no SETPAGESIZE command is encountered within the input data .

Page size names displayed in the choices list are predefined values of the PostScript® interpreter.

The appropriate size values are displayed in the 'size for portrait in 1/72' fields.

Fixed paper size

When active, the configured paper size is always used, i.e. the values of the VIPP® command SETPAGESIZE are ignored.

Size for portrait in 1/72 Inch

Is active if as **Paper size** the option "**User Defined**" is selected.

Special paper size values can be defined here.

Output resolution in DPI

Use default

When active, the default resolution of the PostScript® interpreter is used, otherwise the given resolution.

The value of the default resolution is 720 DPI.

If in the VIPP® input data the command SETRES is used, the value defined here is ignored.

Duplex mode

In the default setting "*Job supplied*" the interpreter will use the duplex settings supplied from the input data stream. In any other setting, the duplex commands from the input datastream will be ignored, and the setting chosen here will be used for the print job.

Ignore numcopies

This option is only relevant, if you convert raw postscript datastreams. VIPP® datastreams normally do not use the numcopies pagedevice command for creating page or job copies. Activate this option, if you want to control the number of copies of the postscript input data by yourself. You can do this by reading the input data multiple times (see [Common settings](#)) or by setting the numcopies value manual in the [Output driver settings](#).

Linemode and Plain text input settings

If you read plain text files or you read the VIPP® linemode datastreams without setting the basic page and font parameters in the Job descriptor ticket (JDT), the default settings chosen here will be used.

⚠ In the plain text mode no carriage return or linefeed commands will be added after reaching the number of **Characters per line** set here. You have to set the number of characters per line so that it matches the formatting of the input data. If you choose auto font sizing (Font height = 0) the font width will be calculated automatically so that the number of characters will fit to the width of the selected **Paper size**.

Everytime a formfeed (default value: 0xc) is detected in the input data stream a page break will be generated. If the datastream does not contain any formfeeds or the number of lines per page does exceed the specified **Lines Per page** value a page break is generated.

⚠ If your linemode data uses other characters or strings to indicate the beginning of a new page, you can place the VIPP function SETPBRK inside a additional resource file (See [Inserting additional resource files](#) for more information about using additional resource files).

⚠ If you choose auto font sizing (Font height=0) the font height and line spacing will be calculated automatically so that the number of **Lines per page** will be fit the height of the selected **Paper size**.

Set the **Skip lines** value to a line number greater than 0 if you want to remove lines from the beginning of a line mode file.

⚠ This option can be useful if you want to remove the first line of a CSV file, which contain column names.

Activate the **Remove formfeed line** option, if the line containing the formfeed char does generate a erroneous additional empty line at the beginning of a new page. Only the line and its content will be removed. The formfeed will still be used to generate a page break.

Use the **Font** drop down list to choose one of the 36 postscript standard fonts, which will be used to print the line mode data, which is not covered and removed by one of the [Postscript textscanner](#) definitions.

Choose one of the **Page orientation** options to select the page orientation for the print job generated from the line mode data.

Select a **N-Up** mode other than One Up, if you want to place more than one logical page generated from the linemode data to one physical page. If you do not choose a option with "1:1" in the name, the page content will be scaled down to fit the selected paper size.

⚠ If you want to create more sophisticated N-Up configurations, you can call the SETMULTIUP VIPP® function in a additional resource file. See [Inserting additional resource files](#) for more information about using additional resource files.

Process XML header

Activate this option, if you want to remove preceding XML headers from input data streams. Not removing a XML header will result in a syntax error during data conversion. If you want to process the information stored in the XML header you can overwrite the internal used *process_xmlheader.xjt* XML Job Ticket. To do this you can redefine it using the XGFRESDEF VIPP® command in a additional VIPP® resource file (see [Inserting additional resource files](#)).

Default XML Jobticket definition:

```
(process_xmlheader.xjt)
{
  BEGINXPD
  BTA /*DEF
  { }
  { }
  { }
  ETA

  ENDXPD
}XGFRESDEF
```

Ignore stdout messages generated from EPS resources

Switched on normally this options prevents embedded EPS or PS resources from sending messages to stdout of the postscript interpreter.

If you switch this option off, messages generated from those resources are transformed into warnings shown in the [Mercury](#) client log window.

Use old font renderer

Since Ghostscript Ver 9.00 Ghostscript uses Freetype as the default font renderer. If you experience problems with font rendering you can reactivate the old font renderer using this option.

Set Ghostscript serialnumber

If you activate this option the running number which is part of the [Mercury](#) licence string is implemented as the serial number returned by the *serialnumber* postscript operator. This can be necessary e.g. for postscript resources like CID fonts from vendors, which need a unique serial number to generate licences for their products.

6.3.22.2.8 Postscript overlays

With Postscript Overlays you can add a variety of Overlays to pages of a existing Postscript or ASCII formatted print job.

⚠ Postscript overlays are not designed to add overlays to VIPP® input data streams.

⚠ Please don't forget to set the **Input data compatibility level** in the [Common](#) settings to the correct type of your input data (**Raw postscript conversion** or **Plain text**)

You can add multiple overlays to each page of the print data. The overlays can consist of static PS, EPS jpg or tif files as well as dynamic Text or Barcode whose content can be extracted from the print job. The overlays can be assigned to static page numbers or repeated groups of pages. Also it is possible to trigger overlays dynamically on specific pages based on the content of the print job or Media selection commands. Job global is it possible to trigger groups of overlay based on the postscript Job Title or [Mercury](#) Job name.

Common settings of a overlay group are stored in a **Insert specification** (see [Insert specifications](#)). Each insert specification can contain a group of **Overlay definitions** (see [Overlay definitions](#)).

Input data driver mode

Because Postscript is a programming language the insertion of elements in a existing postscript datastream is highly dependent on the driver, which generates the input data. Therefore you have to select a basic **Input data driver mode**.

Select "**Windows (Xerox)**" if the driver generate code which is compliant with the Adobe Postscript Language DSC conventions . That means, that at least the basic markers for the page begin (`%BeginPageSetup`) and page end (`%PageTrailer`) are available. For example the most Windows drivers, which uses the default Windows postscript libraries to produce their postscript output, are compliant with the Adobe Postscript Language DSC conventions.

If you are not sure about the source of you postscript data or you data does not contain any DSC comments choose **Generic (Xerox)** as Input data driver mode. If your postscript data does not contain the "`%BeginPageSetup`" DSC comment it is not possible to insert Background overlays, because the beginning of the page cannot be located.

If no Input driver mode works for your postscript data, please contact the [Mercury](#) support, who will check if the implementation of a custom Input driver mode is possible.

Mediaselction and Finishing commands can be highly device specific additions to the postscript language. The [Mercury](#) postscript overlay processor supports the pass through of the basic Adobe conform device options used by the most Xerox printers.

The following device specific options are passed through from the input data to the postscript output datastream:

Duplex options,
Mediatype, MediaWeight and MediaColor,
MediaPosition,
Staple,

Jog,
Collate,
and the numcopies pagedevice parameter.

Please contact the [Mercury](#) support, if you have to use a postscript driver, which does not use Adobe conform device specific commands.

Additionally this input commands can be mapped to the according commands of other printers by using the ppd file of this printer and the available mapping tables.

If you use plain text files as input data, you have to choose "**Plain text**" as Input data driver mode. For the basic formatting options of plain text files also see the according options in the [PS interpreter defaults](#) settings.

- [Insert specifications](#)
- [Overlay definitions](#)

Insert specifications

Insert specification table

A **Insert specification** (upper table) contains the common settings for a group of **Overlay definitions** (lower table). These are for example the trigger type and value as well as location of the overlays on the page. Each insert specification, which is shown as one line in the upper table have its own group of overlay definitions shown in the lower table.


Press the right mouse button in the empty table area to add a new insert specification or to delete all existing insert specifications. Press the right mousebutton on a existing specification line to delete this insert specification or to insert a new insert specification.

Trigger type and Definition name

Left click in the Trigger type column of the desired insert specification row to set the trigger type for the overlay group of this specification. Select one of the trigger types from the drop down list.

JobName

Select JobName as trigger type, if the overlays should be inserted if the character sequence entered in the **Definition name** field is equal to (a part of) the [Mercury](#) jobname of the input data. Enter the part of the job name, which should be used for comparison, in the **Definition name** field of the insert specification.

 For this trigger type multiple insert specifications with the same definition name are allowed!

The content of the [Mercury](#) jobname variable is stored in the postscript variable **VIPP_JobTitle**. If you need to manipulate the content of this variable you can do this inside a additional resource file (see [Inserting additional resource files](#)) using the postscript programming language.

Title

Select Title as trigger type, if the overlays should be inserted if the character sequence entered in the **Definition name** field is equal to (a part of) the postscript job title supplied in the %%Title: DSC line, which normally can be found somewhere near the beginning of the input data. Enter the part of the Job title, which should be used for comparison, in the **Definition name** field of the insert specification.

⚠ For this trigger type multiple insert specifications with the same definition name are allowed!

MediaType

Select MediaType as trigger type, if the overlays should be inserted if the character sequence entered in the **Definition name** field is equal to the *media_name* used in the **/MediaType (media_name) setpagedevice** command which could be found (if used) at the beginning of each page of the postscript input data stream. Once a MediaType value is set in a postscript datastream it stays valid until another Mediatype value is supplied. If the Mediatype is set once at the beginning of the print job all pages uses the same media tray at the printer side. There are several postscript syntax options to set the /MediaType pagedevice parameter. But they normally begins with the *"/MediaType"* keyword, followed by the *media_name* enclosed in parenthesis.

⚠ For this trigger type multiple insert specifications with the same definition name are allowed!

Textscanner

Select Textscanner as trigger type, if the overlays should be inserted only on the pages, where a specific text can be found in the input data. Use the [Postscript textscanner](#) to define the according trigger definitions. Enter a unique name in the **Definition name** field. In the according textscanner definition, select this definition name in the **Select triggered insert specification** drop down list. Please keep in mind that you could not insert a *background* overlay (at the beginning of the page) if the trigger text occurred later on the same page. In this case use a *foreground* overlay on this page or use the trigger to insert the *background* overlay at the following page.

Static

In the static mode the insert specification is valid for all pages of all print jobs processed by the VIPP and Postscript processor. The pages on which overlays are inserted are solely controlled by the **Repeat** value and the page numbers supplied in the **Page Nr** field of the overlay definitions.

Location

Left click on the desired insert specification line to select the active specification. Then left click in the **Location** column and select the desired logical location inside the page definition, where the overlays should be inserted.

Background overlays are inserted at the beginning of the existing page content code. Therefore the content of these overlays can be blanket out by the following existing page content. More than this some postscript drivers erase the whole page before drawing anything on the page. If this is the case, you will not see your overlays at all ! Additionally some postscript generators produces non DSC compliant output, which makes it impossible to find the beginning of a page definition. So if it is possible, always use foreground overlays.

Foreground overlays are inserted at the end of the existing page content code. Therefore these overlays cannot be blanketed by the existing page content.

The overlays will overwrite any existing content located at the same position on the page. Technically it is easier to insert overlays at the end of an existing page definition. So if it is possible, always use foreground overlays.

Use the **New page** location to place the overlays on a new page, which will be inserted after the current and before the following page.

Exclude first page

Set this option to **Yes** if the first page of your print job is a single banner page which should be excluded from the subsequent calculations of repeated page groups.


Repeat

The numerical value (page count) entered here controls the repetition of a page group which uses the Page Nr values defined in the according overlay definition table.

- The starting page of a page group depends on the validness of the trigger definition used in the insert specification.
- The validness of the trigger definition used in the insert specification does override the Page Nr values specified in the overlay definition table.

Examples:

1. If you use the *Static* trigger type, specify 1 as the Repeat/alternative value and define one overlay definition with a Page Nr value of 1, this overlay will be inserted on every page of the print job.
2. If you use the configuration above, but change the Page Nr value of the overlay definition to 2, the overlay will not be inserted on any page, because the logical group page number 2 will never be reached.
3. If you use the *Static* trigger type, specify 5 as the Repeat/alternative value and define one overlay definition with a Page Nr value of 2, this overlay will be inserted on the pages 2,7,12,17, ...
4. If you use the configuration above, add a second (alternative) overlay definition with a Page Nr. value of 0, the first overlay will be inserted on the pages 2,7,12,17,... . The second overlay will be inserted on all other pages.
5. If you use the configuration of example Nr. 3 and change the trigger to a dynamic trigger e.g. a *Textscanner* definition, which is valid at page Nr. 2 and then stays valid for 10 pages, the overlay will be inserted on page Nr. 3 and page Nr. 8.
6. If you use the configuration above and add a second (alternative) overlay definition with a Page Nr. value of 0, the first overlay will be inserted on page Nr. 3 and page Nr. 8. The second (alternative) overlay will be inserted on the pages 2,4,5,6,7,9,10 and 11.

 Insert specifications and overlay definitions are internally identified by the definition name. Therefore you have to choose a unique definition name for each new insert specification and overlay definition.

Overlay definitions

Before start reading here, please first have a look at the [Insert specifications](#) chapter.

Overlay definition table

The overlay definition table shows all configured overlays for the selected Insert specification.

Press the right mouse button in the empty table area to add a new overlay definition or to delete all existing overlay definitions. Press the right mouse button on a existing definition line to delete this overlay definition or to insert a new overlay definition.

- You can create more than one overlay definition for the same page .
- If you plan to lay several overlays on top of each other on the same page, the overlay which lies in the background have to be the first one in the definition table.

Page Nr

Left click at the desired overlay definition line to select the active definition. Then left click in the Page Nr column and enter the logical Page Nr. on which the overlay should be inserted. The absolute page number(s) are calculated, depending on the value of the **Repeat** field and the Trigger validness of the according insert specification. Please have a look at the description of the **Repeat** value and the examples section in the [Insert specifications](#) chapter.

- You can create more than one overlay definition for the same page number.
- Overlays created for the **Page Nr. 0** are treated as alternative overlays, which are inserted on every page for which no other overlay definition exists.

Name

For Barcode-, Text- and Frame-type overlays enter a unique name for the overlay definition. For File type overlays the name is automatically set to the selected overlay filename.

- After setting the **Type** to **File**, you could browse to the desired file or select a text scanner variable containing a file name by right clicking inside the **Name** column and select **Browse for file...** from the context menu. It is also possible to insert multiple page Postscript or PDF files. In this case these files cannot be used as overlays.

Type

Left click at the desired overlay definition line to select the active definition. Then left click in the Type column and select one of the overlay types from the drop down list.

File type overlays:

Choose **File** as the overlay type if you want to insert a static PS, EPS, JPG, TIF file or a VIPP code segment as a overlay.

EPS,PS and VIPP code segment files should be located in the **mislib** subfolder of the **Global resource path** specified in the [Resources](#) settings.

JPG and TIF images should be located in the **imglib** subfolder of the **Global resource path**

specified in the [Resources](#) settings.

- After setting the **Type** to **File**, you could browse to the desired file by right clicking inside the **Name** column and select **Browse for file...** from the context menu.

⚠ If you manually clear the **Name** entry of the overlay definition, no visible overlay will be created for **Foreground** and **Background** located insert definitions, but the code specified in the **Mediatype** or **{VIPP code}** field will be executed. For insert definitions with **New page** as Location setting, a empty overlay definition name will result in adding a empty page.

Hints for creating full page postscript overlay files:

- You can use almost every Windows postscript Printer driver to create a suitable postscript overlay file using the print option of your Windows application, but we recommend using the Xerox global printer driver.
- Configure the printer driver to print to the file port.
- Configure the printer driver to download all used fonts to the printer.
- Disable all printer specific non postscript enhancements as for example embedded XML job tickets.
- If you use the Adobe Reader to generate a postscript file from a PDF resource, disable the "set papersource from PDF- page size" option in the Print settings.

Barcode overlays:

Select **Barcode** as type, if you want to insert a dynamic or static barcode overlay. Then right click at the definition row and select **Properties...** from the context menu to edit the barcode properties. See [Barcode overlays](#) for more information.

Text overlays:

Select **Text** as type, if you want to insert a dynamic or static text overlay. Then right click at the definition row and select **Properties...** from the context menu to edit the text properties. See [Text overlays](#) for more information.

Frame overlays:

Select **Frame** as type, if you want to insert a static frame e.g. as a background for your text overlay. Then right click at the definition row and select **Properties...** from the context menu to edit the frame properties. See [Frame overlays](#) for more information.

Scale

If you want to enlarge or reduce the size of your overlay enter a scaling factor greater or less than 1.0.

Rotation

Enter a degree value other than 0, if you want to rotate the content of the overlay. Values greater 0 result in a counterclockwise rotation, values less than 0 will do a clockwise rotation. The origin point for the rotation is the lower left corner of the overlay.

ShiftX [mm] and ShiftY [mm]

Enter mm values greater or less than 0 for shifting the overlay content on the page. The origin for the shift is the lower left corner of the page.

Positive **ShiftY** values will result in a up shift, negative **ShiftY** values will produce a down shift of the overlay content.

Positive **ShiftX** values will result in a right shift, negative **ShiftX** values will produce a left shift of the overlay content.

Side

For duplex printing you can select, if the overlay should appear on the **frontside**, **backside**, or **both** sides of the sheet. Select the appropriate entry of the drop down list by left clicking on the **Side** column of th desired overlay definition row.

valid for Copy Nr:

If you have entered a **Copies of input data** value greater than 1 in the [Common](#) settings, you can specify copy numbers for which the overlay definition should be valid. Leave this field empty, if the overlay should be valid for all copies of the input data stream. Enter copy numbers (separated with Space), if the overlay should be valid only for specific copies of the input data stream.

MediaType, {VIPP code} or [overlay condition]

Printer tray selection via **MediaType**:

If the page media for the overlay page should be drawn from a specific printer tray, you can specify a MediaType name, which will result in a */MediaType setpagedevice* call inserted at the beginning of the page with the overlay content. If your printer does not recognize */MediaType* commands, you can map

the Mediatype values to the specific commands of your printer using the [Input media mapping](#) table and the correct ppd file of your printer. After processing the input data at least one time the Mediatype name entered here will automatically inserted in the left column of the input media mapping table. For more information about ppd files and input media mapping the *Printer* topic in the [Common](#) settings and have a look at the [Input media mapping](#) help section.

Inserting specific VIPP or Postscript code before or instead of a overlay:

You can enter custom VIPP or Postscript code in the **MediaType or {VIPP code}** edit box, if you enclose this code with curly brackets.

If you select **File** as Overlay **Type**, and enter a empty file name in the **Name** column, no overlay will be generated and only the custom VIPP or postscript code will be executed.

Examples:

The VIPP code


```
{ENDOFSET}
```

will generate a end of set directive after the page on which the overlay is called. Together with activating the *Split job at the end of a set* option in the Postscript **Output driver settings** the job will be splitted after the page with the overlay.

The VIPP code

```
{true DUPLEXON}
```

will activate Duplex printing with short edge binding (tumble duplex printing). Use the code sequence `{false DUPLEXON}` to activate Duplex printing with long edge binding and `{DUPLEXOFF}` to deactivate duplex printing.

⚠ Please don't forget that switching the duplex mode only can be done on a front side of a sheet. Additionally the overlay containing the duplex switching code should be called only once on the first page of a duplex section. If your overlay with the "Duplex on" code is called on every page of the document, every page will be forced to a new front side by the printer. To solve this problem, you can create a second triggered insert specification with 0 as **Repeat** option containing only one overlay definition with a "empty" file overlay and the duplex switching code.

If you have to **insert printer specific postscript code** in the output data stream, you can also do this by entering some additional VIPP commands in the additional resource file (see [Inserting additional resource files](#)) and in the `{VIPP code}` section of the overlay definition.

Example:

If you have to set the printer to Black and White printing mode at a specific page, job copy, or split part

- 1) Define a overlay, which is called at the desired page.
- 2) Insert a ADDINSERTPSVAR instruction to the additional resource file, which generates a variable with the desired postscript code:

Syntax ADDINSERTPSVAR:

```
(var name) (postscript code which should be inserted in the output)
ADDINSERTPSVAR
```

Example:

```
(Black and White) (<</ProcessColorModel /DeviceGray>> setpagedevice)
ADDINSERTPSVAR
```

- 3) Call INSERTPS with the variable name as the argument in the `{VIPP code}` section of the overlay definition:

Example1:

```
{ (Black and White) INSERTPS }
```

Example2:

```
{ COPYNR 1 gt CPPCOUNT 1 eq and {(Black and White) INSERTPS}if }
```

The example above will insert the postscript code only on the first page beginning with the second job copy .

Specifying a condition for executing a overlay:

VIPP or postscript code specified between "[" and "]" will be used to generate a condition for executing the overlay definition. The result of the expression have to be the boolean value "true" or "false". You can use any built in or textscanner generated variable inside the expression.

Examples:

1)
[var_textscan_01 (order) eq]

The overlay will only be executed, if the variable var_textscan_01 contains the text "order". The variable var_textscan_01 have to be defined using the [Postscript textscanner](#).

2)
[PPCOUNT 3 gt]

The overlay will only be executed, if the physical page counter is greater than 3

3)
[jobname_in (order) anchorsearch {pop pop true}{pop false}ifelse]

The overlay will only be executed, if the jobname begins with "order"

4)
[jobname_in BCOUNT 1 gt PPCOUNT 1 le and]

The overlay will only executed on the first page and if the jobname contains more than one space character.

1) Using single page files as static overlays:

Choose **File** as the overlay type if you want to insert static single page PS, EPS, JPG, TIF files or VIPP code segments as overlays.

EPS, PS and VIPP code segment files should be located in the *lmislib* subfolder of the **Global resource path** specified in the [Resources](#) settings.

JPG and TIF images should be located in the *limglib* subfolder of the **Global resource path** specified in the [Resources](#) settings.

- After setting the **Type to File**, you could browse to the desired file by right clicking inside the **Name** column and select **Browse for file...** from the context menu. In the Browse for file dialog select the desired file or us the **Select variable** drop down list to choose a Text scanner variable containing a file name (without path). The Name of the overlay then will be automatically set to the selected file name (without path) or variable name. The variable name will be automatically enhanced with a preceding '/' to mark this name as a variable. As soon as you have chosen a variable from the drop down list, the content of the file selection box will be ignored.
- To use the selected file or variable content as a single page overlay set the drop down list selection below the Select variable drop down list to **"Use file content as single page overlay"**. The file deletion option which can also be set in this dialog has no meaning in this case.

2) Insert multiple page PDF or Postscript files:

Choose **File** as the overlay type if you want to insert multiple page Postscript or PDF files. The content of those files cannot be used as overlays. The file can be located outside the **mislib** subfolder of the **Global resource path** specified in the [Resources](#) settings.

- After setting the **Type** to **File**, you could browse to the desired file by right clicking inside the **Name** column and select **Browse for file...** from the context menu. In the Browse for file dialog select the desired file or use the **Select variable** drop down list to choose a Text scanner variable containing a file name or file template. The Name of the overlay then will be automatically set to the selected file name (with path) or variable name. The variable name will be automatically enhanced with a preceding '/' to mark this name as a variable. As soon as you have chosen a variable from the drop down list, the content of the file selection box will be ignored.
- For file insertion it is also possible to use templates in the file name specification, e.g. the file name "c:/input/*.pdf" will insert (alphabetically sorted) all pdf files located in the c:/input/ folder.
- To insert the selected file or variable content set the drop down list selection below the Select variable drop down list to "**Insert file (multiple page PDF files allowed)**". If you want to delete the inserted files after finishing the job, set the file deletion option to "**Delete file after the last copy**".

⚠ PS or EPS files must not contain a XML header !

⚠ VIPP segment files have to end with the ".seg" extension !

⚠ If you generate Postscript output you should activate the resource pass through mode, if your printer is capable of storing all overlay files in internal memory. This will lead to faster processing and smaller output data size. Please have a look at the **Resource pass through for postscript output** topic in the [Output driver settings](#) section.

Select **Barcode** as **Type** in the overlay definition table, if you want to insert a dynamic or static barcode overlay. Then right click at the definition row and select **Properties...** from the context menu to edit the barcode properties.

Barcode Overlay Properties

Activate the option **Vertical positioning relative to preceding overlay** to place this overlay right below the last output position of the preceding overlay in the overlay definition list. The **ShiftY** parameter of the overlay definition is then used as a offset value to the last output position of the preceding overlay.

Select barcode type:

Select the desired barcode type from the drop down list.

One dimensional code properties

The one dimensional code properties are only available for one dimensional type barcodes like

Code128B.

Height:

Enter the desired height for the barcode in Points (1/72").

⚠ To change the height and width of two dimensional barcode types, adjust the **Scale** value in the Overlay definition table.

Hor. Stretch:

Optional horizontal stretch factor for the barcode. For example 200% means that the barcode width will be double as wide as the barcode normally would be for the given height value.

Show barcode as Human Readable Text

Activate this option, if the barcode should also shown as Human Readable Text (HRT)

HRT font

Select one of the 35 Postscript standard fonts from the drop down list or select "*custom font*" (the last in the list) and enter a valid postscript fontname in the HRT font edit box .

⚠ If you use a custom font, a valid postscript font library file containing a font with the entered name have to be placed inside the *encoding* subfolder of the **Global resource path** specified in the [Resources](#) settings.

HRT size

Enter the desired size of the human readable text in points (1/72").

HRT vertical offset

Enter the vertical offset in Points (1/72") of the human readable text relative to the lower edge of the barcode.

A negative value will result in downshift of the HRT (below the barcode), e.g. if you specify a HRT size of 8 Points and a HRT vertical offset of -8 Points, the human readable text will be placed exactly below the barcode.

Select variable whose contents should be printed as a barcode:

If you want to print the content a variable defined prior in the [Postscript textscanner](#) settings as a barcode, select this variable from the drop down list.

If you want to add static text to a variable content or if you want to concatenate the content of several variables use the edit field below the drop down list to enter the barcode content string.

Variable names in the content string have to be enclosed with "[=" and "=]".

- ⚠ As soon as you have entered some characters in the content string edit box the variable name selected in the drop down list will be ignored !
- ⚠ Beside the variables defined in the Postscript textscanner, you can also use all VIPP variables, public emulation internal variables (see [Build in variables](#)) and custom variables defined e.g. in a additional resource file (see [Inserting additional resource files](#)).

Example for barcode content string:

If the current date is 12.10.2012 and you have defined a textscanner variable named `custnr` which have the content 12345, the barcode content string:

```
Customer Number: [=custnr=] Date: [=D DD=].[=D MO=].[=D YYYY=]
```

will print the following text as a barcode:

```
Customer Number: 12345 Date: 12.10.2012
```

Select **Text** as **Type** in the overlay definition table, if you want to insert a dynamic or static text overlay. Then right click at the definition row and select **Properties...** from the context menu to edit the text properties.

Text Overlay Properties

Activate the option **Vertical positioning relative to preceding overlay** to place this overlay right below the last output position of the preceding overlay in the overlay definition list. The **ShiftY** parameter of the overlay definition is then used as a offset value to the last output position of the preceding overlay.

Basic text formatting

Choose the desired basic text formatting for your text overlay from the drop down list.

- **Single line left aligned** : The text will be placed left aligned on the position specified in the overlay definition table. No line wrapping will occur, if the specified Column width is exceeded. The specified **Text align** value is ignored in this basic formatting mode.
- **Single line right aligned** : The text will be placed right aligned relativ to the position specified in the overlay definition table. No line wrapping will occur, if the specified **Column width** is exceeded. The specified **Text align** value is ignored in this basic formatting mode.
- **Line wrapping** : Text positioning will begin at the position specified in the overlay definition table. Automatic word based line wrapping will occur, if the specified **Column width** is exceeded. If the specified Columnwidth is 0, no line wrapping will occur. If the given line spacing value is 0, the line spacing is calculated from the actual page size and the **Lines per Page** specified in the [PS interpreter defaults](#). For aligning the text, the value selected

from the **Text align** drop down list is used.

- **Single line truncated** : Text positioning will begin at the position specified in the overlay definition table. For aligning the text, the value selected from the **Text align** drop down list is used. If a column width value greater than 0 is specified and this value is exceeded, the text will be truncated (word based).

Height:

Enter the desired point size of the selected font.

Hor. Stretch:

Optional horizontal stretch factor for the text output. For example 50% means that the text width will be half as wide as the text would be for the given font height.

Font:

Select one of the 35 Postscript standard fonts from the drop down list or select "*custom font*" (the last in the list) and enter a valid postscript fontname in the Font edit box .

⚠ If you use a custom font, a valid postscript font library file containing a font with the entered name have to be placed inside the `encoding` subfolder of the **Global resource path** specified in the [Resources](#) settings.

Column width:

If you enter a Column width value greater than 0, this value will be used for the **Line Wrapping** and **Single line truncated** Basic text formatting modes to calculate the end of line position.

Line spacing:

Enter the line spacing value used for the **Line Wrapping** basic formatting mode. If the given line spacing value is 0, the line spacing is calculated from the actual page size and the **Lines per Page** specified in the [PS interpreter defaults](#).

Text align:

Select the text align mode used for the **Line Wrapping** and **Single line truncated** Basic text formatting modes.

Select variable whose contents should be printed:

If you want to print the content a variable defined prior in the [Postscript textscanner](#) settings, select this variable from the drop down list.

If you want to add static text to a variable content or if you want to concatenate the content of several variables use the edit field below the drop down list to enter the text content string.

Variable names in the content string have to be enclosed with "[=" and "=]".

- ⚠ As soon as you have entered some characters in the content string edit box the variable name selected in the drop down list will be ignored !
- ⚠ Beside the variables defined in the Postscript textscanner, you can also use all VIPP variables , public emulation internal variables (see [Build in variables](#)) and custom variables defined e.g. in a additional resource file (see [Inserting additional resource files](#)).
- ⚠ If you have triggered multiple data values with one single trigger sequence on the same page, and you have chosen to store this values by selecting a fieldname with a appended underscore char, this values will be automatically printed to consecutive lines by selecting the according fieldname of the variable (see also [Postscript textscanner](#) for more information).

Example for text content string:

If the current date is 12.10.2012 and you have defined a textscanner variable named `custnr` which have the content 12345, the text content string

```
Customer Number: [=custnr=] Date: [=D DD=].[=D MO=].[=D YYYY=]
```

will result in the following printed text :

```
Customer Number: 12345 Date: 12.10.2012
```

- ⚠ Inside the content string VIPP® font and color switch sequences can be used.

Text background

You can specify a background for the printed text. First select a basic background color from the **Background base color drop** down list. Then select the background type and color saturation from the **Text background** drop down list.

The name attachments in the **Text background** drop down list specify the saturation of the base color.

Meaning of the name attachments (light to dark order):

- W : white
- XLT : extra light
- LT : light
- LMED : light medium
- MED : medium
- DMED : dark medium
- DRK : dark
- XDRK : extra dark
- BaseColor : Use full saturated base color

You can also specify a custom background using the VIPP® BAT (Background Attribut) format. To do this enter **Custom BAT key** from the Text background drop down list and enter the BAT specification in the edit box right beside the list. For more information about BAT keys have a look at the Xerox **VIPP® Reference Manual**.

Select **Frame** as **Type** in the overlay definition table, if you want to insert a static frame e.g. as a background for your text overlay. Then right click at the definition row and select **Properties...** from the context menu to edit the frame properties.

Frame Overlay Properties

Activate the option **Vertical positioning relative to preceding overlay** to place this overlay right below the last output position of the preceding overlay in the overlay definition list. The **ShiftY** parameter of the overlay definition is then used as a offset value to the last output position of the preceding overlay.

Width:

Enter the desired width of the frame in mm

Height:

Enter the desired height of the frame in mm

Corner radius:

Enter a corner radius greater than 0 to generate a frame with rounded corners.

Multiple frames

If you want to generate more than one frame at the same time, enter a value > 0 in the **Repeat frame** edit field. Each new frame will be displaced relative to the preceding one using the specified **Horizontal displacement** and **Vertical displacement** values. Positive vertical displacement values will result in a upshift of the next frame, negative values will produce a downshifted follow up frame.

Frame appearance

The frame appearance is controlled by selecting on of the VIPP® GEP (Graphic Element Properties) keys from the **Frame appearance** drop down list.

GEP key naming conventions:

- GEP keys beginning with "F" followed by a color name will generate a colored background without a margin line.
- GEP keys beginning with "S" will produce a solid line at the frame margins.
- GEP keys beginning with "D" will produce a dotted line at the frame margins.
- The numbers at the end of the GEP key represent the thickness of the margin lines

- Preceding character sequences followed by a "_" represent the saturation and base color (last char before the underscore) of the margin lines. The OW_.. GEP keys using white as a opaque background color).
- GEP keys beginning with GR_ are gradient color GEP keys. The last character represents the direction of the gradient ("H"= horizontal, "V"=vertical).
- GEP keys which only consist of one of the following saturation abbreviations followed by a one char base color identifier will produce a colored background without margin lines.

Color saturation abbreviations (from light to dark):

- XLT : extra light
- LT : light
- LMED : light medium
- MED : medium
- DMED : dark medium
- DRK : dark
- XDRK : extra dark

Also see [Gray GEP Key Examples](#) for some visual examples of gray GEP keys.

You can also specify a custom frame appearance using the VIPP® GEP key format. To do this enter **Custom GEP** from the Frame appearance drop down list and enter the GEP specification in the edit box right beside the list. For more information about GEP keys have a look at the Xerox **VIPP® Reference Manual**.

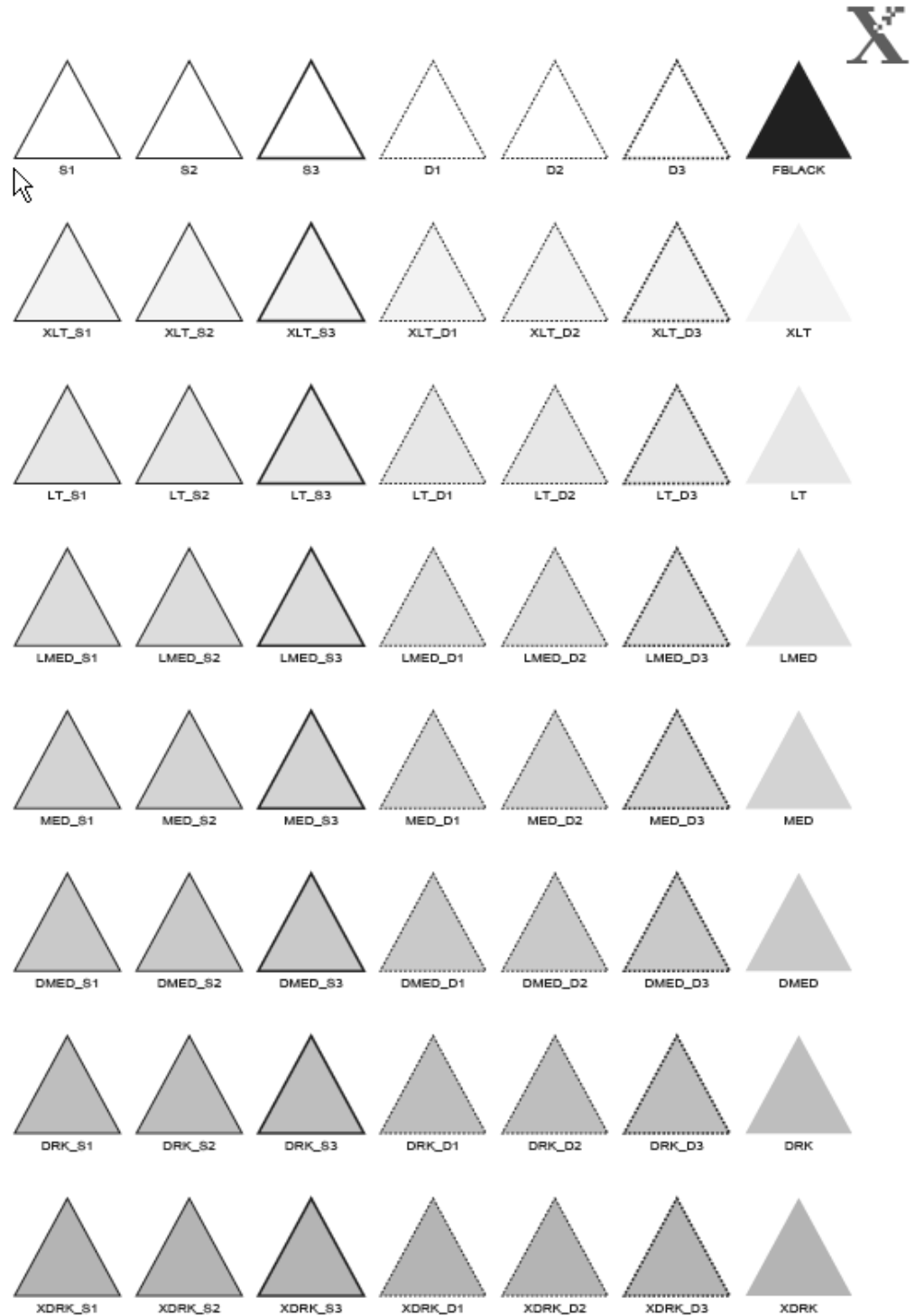
Built in variables, which can be used in Barcode or Text overlays:

- **PSJobTitle** : contains the name supplied in the %%Title DSC line somewhere near the beginning of the postscript printjob.
- **VIPP_JobTitle** : contains the [Mercury](#) jobname of the current printjob.

VIPP variables:

- **D_DD** : Day of the Month (1-31).
 - **D_DOY** : Day of the year (1-366).
 - **D_DWL** : unabbreviated week day (Monday, Tuesday, etc.)
 - **D_DWS** : abbreviated week day (Mon, Tue, etc.).
 - **D_MO** : numerical month name (01-12) .
 - **D_MOL** : unabbreviated month name (January, February, etc.).
 - **D_MOS** : abbreviated month name (Jan, Feb, etc.).
 - **D_YY** : 2-digit year.
 - **D_YYYY** : 4-digit year.
 - **T_AMPM** : set 12-hour clocks to an AM or PM designation .
 - **T-HH** : Time (hours) for your application. This variable is used to set 24-hour clocks.
 - **T-HH2** : Time (hours) for your application. This variable is used to set 12-hour clocks.
 - **T-MM** : Time (minutes) (00-59).
 - **T-SS** : Time (seconds) (00-59).
 - **T_TZN** : Zone (PST,PDT)
-
- **PPCOUNT** : Physical pagecounter, containing all copies generated with the *Copies of input data* option on the [Common](#) configuration sheet.
 - **CPPCOUNT** : Physical pagecounter of the actual job copy.

- **COPYNR** : Number of the actual job copy



VIPP gray GEPkeys samples

6.3.22.2.9 Postscript textscanner

The Postscript textscanner can be used to scan raw postscript or plain text input data for the occurrence of specific character sequences (Trigger sequence) painted on the page by the typical postscript text output operators.

The occurrence of this character sequence can be used as a trigger for a insert specification (see [Postscript overlays](#)), which will be triggered on the pages on which the configured textscan field is valid.

Additionally the text after (or before) the trigger sequence can be stored in a variable (Field Name). The content of this variable can be used in subsequent text- or barcode overlays (see [Text overlays](#) and [Barcode overlays](#)).

⚠ Please don't forget to set the **Input data compatibility level** in the [Common](#) settings to the correct type of your input data (**Raw postscript conversion** or **Plain text**)

Textscan definition table

At the top of the Postscript textscanner configuration sheet you will find the textscan definition table. Press the right mousebutton inside the empty table area and choose the "Add row" menu option with the left mouse button to add a new text scan field definition. A new definition line with default values will be added in the definition table. The basic configuration parameters will be shown in the columns of the table. The advanced parameters of the selected definition (table row), will be shown below the table. Apart from the options in the "Global parameters" group all settings below the table belongs to the current selected row in the table and therefore can be set to different values for each textscan field.

To delete or insert a definition move the mouse cursor over the desired table row, left click to select the row, then click the right mouse button and choose the desired option from the context menu.

Field Name

Left click at the desired row to select the definition. Left click in the Field Name column and enter a unique name for your textscan definition.

⚠ The textscan and overlay definitions are internally identified by the Field Name. Therefore you have to choose a unique Field Name for each new textscan or overlay definition.

Special fieldnames:

- The field names *jobname*, *username*, *destaddress* and *destport* can be used to initialize the equal named variables of the [Mercury](#) Job information data.
- If you append a underscore to the field name and the triggersequence appears more than once at a page, the triggered data will be stored automatically to variables with names consisting of the field name and a appended autoincremented number.
E.g. if you select *field_* as the field name and the trigger appears four times on the page the four data values will be stored in the variables *field_*, *field_1*, *field_2* and *field_3*.
If you choose such a fieldname (with underscore, but without number) as a variable in a Text overlay (see [Text overlays](#)) the data values will be automatically printed to consecutive rows.
- Valid for "Trigger+Data" Type: If you insert the character sequence *#data#* into the fieldname, the character sequence after *#data#* will be compared to the found data string. Only if this part of the field name is found at the beginning of the data string, the trigger will be marked as found. The data will be stored, even if it is not equal to the field name part after *#data#*.
E.g. if you choose a field name *Myfield01#data#12345* and a Trigger sequence of

CustomerNr: , the trigger will only be found, if the trigger string CustomerNr: is found and the associated extracted data begins with the character sequence 12345.

Type

Left click at the desired row to select the definition. Left click in the Type column and select the desired field type from the drop down menu.

Trigger:

Choose this option, if you want to use the text entered in the trigger sequence column only as a trigger for a insert specification (see [Postscript overlays](#)). In this mode no data will be stored for later usage in [Text overlays](#) or [Barcode overlays](#). The *Data Options* below the table will be grayed out for this textscan definition.

Trigger+Data:

Choose this option, if you want to use the text entered in the trigger sequence column as a trigger for a insert specification (see [Postscript overlays](#)). In this mode the data before or after the trigger sequence will be stored under the variable name entered in the Field Name column. The content of this variable can be used in subsequent [Text overlays](#) or [Barcode overlays](#).

Data:

Choose this option, if you only want to extract data from the input data stream. The trigger sequence entered in the Trigger sequence column will only be used to find the data, but cannot be used as a trigger for a insert specification. The data before or after the trigger sequence will be stored under the variable name entered in the Field Name column. The content of this variable can be used in subsequent [Text overlays](#) or [Barcode overlays](#).

Startpage

Left click at the desired row to select the definition. Left click in the Startpage column and enter the absolut physical page number at which the scanner should begin with the search of the character sequence entered in the Trigger sequence column.

Endpage

Left click at the desired row to select the definition. Left click in the Endpage column and enter the absolut physical page number at which the scanner should stop with the search of the character sequence entered in the Trigger sequence column. If the entered value is 0 or less than the value entered in the Startpage column, the job will be searched for the occurrence of the trigger sequence up to the last page.

Trigger sequence

Left click at the desired row to select the definition. Left click in the Trigger sequence column and enter the character sequence which should be found in the input datastream.

Special trigger sequences for selecting specific lines of the input data:

You can extract a specific line or a range of lines of a plain text file by entering the char sequence

```
#_#line_nr
```

followed by a "=", "<" or ">" operator and the desired line number.

If you want to add a second line number compare condition enter a "&" or a "|" char for a "and" or "or" combination followed by the second compare condition.

The compare expression have to be terminated with a concluding "#" char (see example below).

If the line number is negative the line is searched backwards from the end of the page. After the concluding "#" char, optional a normal trigger sequence can be added.

Example trigger sequence:

```
#_#line_nr>5&<10#customer number
```

In the example above the trigger sequence "customer number" is searched beginning with the sixth line of the input data. The search is stopped after the 9. line.

⚠ If the input data is a postscript stream pseudo line numbers are generated and incremented every time the vertical print position is changing. In this case negative line numbers for searching backwards from the end of a page are not supported.

⚠ Some postscript drivers use hexadecimal character sequences to feed the postscript text output operators. You can find this out by viewing the postscript input data with a text viewer.

Example:

```
429 1634 M <17061819051A0C1B1B0C0310> [55 56 55 54 29 66 54 50 49 57  
21 0]xS
```

The line above shows a typical example for printing a line of text using a hexadecimal character representation.

The green marked part is a positioning command, the blue marked part is a text output operator name and the read marked part is the character sequence.

You can use hexadecimal sequences as a trigger sequence. the hexadecimal characters have to be enclosed between "<" and ">" If you want to use e.g. the first six character of the example above as the trigger sequence, you have to enter <17061819051A> in the Trigger sequence field. If you want to determine the ASCII representation of the trigger string, you have to look at the postscript data with a postscript viewer like GSView.

Advanced configuration options

Field validity

Here you can enter how many pages the trigger sequence is valid after the trigger sequence is found.

Available options:

Until end of printjob

The trigger is valid from the page at it is found first to the last page of the printjob. The data is initialized only at the first occurrence of the trigger sequence.

Until next occurrence

The trigger is valid from the page at it is found first to the last page of the printjob. The page at which the triggersequence is found as well as the associated data is updated at each new occurrence of the triggersequence.

Number of pages

The trigger and the associated data is valid a number of pages beginning with the page it is found first. Enter the desired page count in the attached edit box.

Data options

The data options are valid only for the **Data** or **Trigger+Data** textscan definition types.

Occurrence of the trigger sequence

Choose **Use first occurrence of the trigger sequence** to store the data, which is found at the first occurrence of the trigger sequence on the page.

Choose **Use last occurrence of the trigger sequence** to store the data, which is found at the last occurrence of the trigger sequence on the page.

⚠ If you use a field name with a appended underscore, and you want to store multiple data values for each occurrence of the trigger sequence on the page, you have to set this option to **Use last occurrence of the trigger sequence**.

Limitation mode, Offset/FieldNr and Length

Use the **Offset/length** option as limitation mode, if you want to specify a character count offset relative to the trigger sequence to find the beginning of the data character sequence and if you want to specify a specific character count to determine the length of the data sequence.

Enter the **Offset** and **Length** values in the edit boxes below the limitation mode selector box.

Choose a negative offset value, if you want to retrieve data left from the trigger sequence.

If the length value is 0, all character of this line (and the *Collect lines* following lines, if **Collect lines** value is > 0) are stored in the field name variable.

If the length value negative, it is used as a offset value from the end of the line.

Use the **FieldNr/Separator** option, if the text sequences, you want to extract, are limited by a special field separator character. Enter this character in the **Field separator** edit box. The value in the Offset/FieldNr edit box then is interpreted as field number after (if positive) or before (if negative) the trigger sequence.

⚠ The Length value is not used in this limitation mode.

Positive field numbers starting with 0. This will retrieve the text data after the trigger sequence up to the first occurrence of the field separator char.

Negative field numbers starting with -1. This will retrieve the text data before the trigger sequence up to the first occurrence of the field separator char.

⚠ If the **Space terminates value** option is selected, only the characters up to the next space char are stored as data value.

⚠ Leading and trailing spaces are stripped from the extracted character sequence.

⚠ If the data sequence is not at the same line as your trigger sequence, you have to set the **Collect lines** parameter to a value > 0

⚠ Retrieving data from lines before the trigger sequence is somewhat limited, because the storing of text lines of a page is started with the first occurrence of a trigger sequence defined in the textscan definition table.

Store trigger sequence

Activate this option, if you want to store the trigger character sequence along with the data portion.

⚠ Special trigger sequences as e.g. `#_#line_nr#` are not stored.

Space terminates value

Activate this option, if a space char should terminate the data sequence before the length value or field separator char is reached.

Global parameters

The configuration values in this section are valid for all textscan definition defined in the definition table.

Collect lines (of a page)

Enter a value > 0 , if you want to extract text data, which is not in the same line as the trigger sequence.

The storing of lines of a page is started with the line on which the first trigger sequence defined in the textscan definition table occurs. The line data character sequences are separated by the line separator character entered in the edit box labeled with "**Line separator char:**".

Separator at moveto

Some postscript drivers produce spaces between words not by using the space character but by inserting a moveto command for positioning the next word on a specific page position. To extract those words separately it is necessary to insert a separation character each time when a moveto operator is called.

Activate this option, if you experience problems in separating consecutive words of a text line.

Delete triggered text

Raw postscript input data:


If you activate this option the postscript processor will try to delete the triggered text. Which text is really deleted depends on the implementation of the text output function used by the postscript driver which generated the input data. If each character is painted on the page by a separate function call, only one character will be deleted. If the text of a whole line is painted by one function call the whole line will be deleted. Use this function with care. You have to check, if it works for your type of data.

Plain text input data:

The whole line containing the trigger sequence will be deleted.

Delete all text

If you activate this option all text output of the input datastream will be deleted. Text painted as a bitmap and all other non text graphic elements will remain on the page.

 Deleted text can still be extracted and used in triggered text- or barcode overlays.

Strip leading and trailing spaces

If activated, strips leading and trailing spaces from linemode data rows and postscript text lines. Should not be activated, if the spaces are used and needed for proper horizontal formatting of consecutive rows.

Select triggered insert specification

For textscan definitions of the type **Trigger** and **Trigger+Data** you can select a insert specification (see [Insert specifications](#)), which will be triggered on the pages, on which the according textscan definition is valid. The list of available insert specification will only contain previously defined insert specifications with the "Textscanner" Trigger type. Please read the [Postscript overlays](#) section for more information on how to create overlays for postscript and plain text datastreams.

Manipulation of the scanned text data

You can manipulate the content of the scanned variables with custom postscript code placed inside a additional Postscript resource file (see [Inserting additional resource file](#) for more information about additional resource files).

You can redefine the postscript function MODIFYTEXTVARS, which by default does nothing. The function is called for every scanned variable with the following values on the operand stack:

```
<name>varname      <string>varcontent  <bool>pageticket  MODIFYTEXTVARS
```

The boolean value `pageticket` is true, if the variables are used outside the postscript interpreter. This is especially the case for the variables transferred to the Mercury Job information (field names: *jobname*, *username*, *destaddress* and *destport*).

In the following example the function MODIFYTEXTVARS is used to substitute the content of the *PSJobTitle* *) variable for the scanned value of the *jobname* variable.

```
/MODIFYTEXTVARS
{
  {
    % boolean value pageticket = true

    1 index /jobname eq % check if the variable
                        % name is "jobname" ...
    { pop % ... if yes, discard the
          % original scanned value.
          PSJobTitle % use the content of the
                    % internal PSJobTitle variable
                    % instead
    }if
  }
  {}ifelse % pageticket=false: do not modify jobname
           % for usage inside the postscript interpreter
           % e.g. for Text overlays
}def
```

*) The internal *PSJobTitle* variable contains the job title supplied with the `%%Title:` postscript DSC comment line, which often can be found near the beginning of a postscript job. Only the part after the `%%Title:` character sequence is written to this variable.

6.3.22.2.10 Performance

The overall data handling specified in the 'Performance' tab controls specifies the data block size and the timing of the data transfer between the [Mercury](#) components.

Normally these default values don't have to be modified, they are compatible to most systems.

⚠ Please don't change these values yourself, only the support can give you the right values.

6.3.22.2.11 Output driver settings

PCL and PostScript® output settings

Job copies (only Postscript® output)

The destination printer device generates the given number of identical copies for each job. The number of copies specified here are generated from the output data, after the copies specified in the VIPP® input data are generated.

⚠ A standard PostScript® sequence (<< /Collate true /NumCopies numberofcopies >> setpagedevice) will be only send, when the number of copies is greater than 1.

⚠ The 'Job copies' sequence is sent after sending the PPD defaults, i.e. overwrites the copy number.

Job Splitting

If a page number greater than 0 is specified, the output data will be splitted after each sequence of the specified number of pages.

In the case of duplex printing the splitting is delayed one page, if the split page number is reached at the frontside of a duplex sheet.

If the option **Split job at the end of a set** is activated, the page number entered above is ignored and the output is splitted after each page on which the VIPP function ENDOFSET is called. If ENDOFSET is not used in the VIPP project, no splitting will occur.

If you generating copies by reading the input data more than one time (set in the [Common](#) settings), the ENDOFSET function will be automatically called after each copy.

Page content shift and zoom

Shift settings:

All other values than 0 shift the whole page content of each page to the right if positive values for the horizontal shift are specified and to the left, if negative values are entered. The direction of the vertical shift depends on the output driver. In case of Postscript®, positive values causes a upshift. If a PCL output driver is used, positive shifting values causes a downshift of the page content.

Zoom setting:

Change this setting, if you want to change the size of the content of every page. The default value of 100% results in a 1:1 scaling of the PCL input to the postscript output.

Resource pass through for postscript output (recommended)

Normally all tiff and jpg as well as all native ps and eps resource data used in the VIPP® project is interpreted from the Ghostscript postscript interpreter and reassembled by the output driver of Ghostscript. The data is inserted in every location at every page it is needed. Depending on the size, format and the number of resources and pages this can lead to low conversion rates and big output size.

It is possible to establish a bypass mode for all this resources. The resources then are directly inserted in the postscript

output and are no longer interpreted and reassembled by Ghostscript. If your printer is equipped with a Level 3 Postscript® Interpreter and enough memory, **using this option can lead to dramatically increased conversion speed and smaller output sizes of the print jobs** at the cost of higher memory usage in the printer controller.

⚠ If you want to use the resource pass through mode with the **Postscript output (high level)** output driver, you have to install a Ghostscript Version ≥ 9.01 .

Parameters for control the resource pass through option :

The Checkboxes **Enable EPS and image pass through** enables the bypass mode for every eps, ps, jpg and tiff resource used by the VIPP® functions SCALL, SETFORM, SETBFORM, ICALL, RUNTIF and RUNDD . If you enable only the first (**Enable EPS**) checkbox, only ps and eps resources used by the VIPP® functions SCALL, SETFORM and SETBFORM are passed through.

Additionally suitable image and eps resources embedded in the input datastream using the VIPP® command XGFRESDEF are passed through to the output datastream.

By Enabling the **Disable pass through for: ICALL, SEFORM or embedded resources** checkboxes, you can explicitly disable the bypass mode for all resource files called from this VIPP® functions.

In the **lower size limit**: edit field you can specify a lower file size limit in Bytes. All eps,ps tiff or jpg resources which have a filesize smaller than the value specified here are excluded from the bypass mode.

Additional to the options specified in the configuration GUI, you can **exclude a resource explicitly by its filename**. If you experience problems with a specific resource named *resourcefilename_1* , you can add the following line in a additional VIPP® resourcefile (see [Inserting additional resource file](#)):

```
[/FormPassExclude [(resourcefilename 1)] SETPARAMS
```

If you have to exclude more than one resourcefile you can simply add the desired filenames in the parameter list:

```
[/FormPassExclude  
[(resourcefilename 1)(resourcefilename 2)...(resourcefilename n)]]  
SETPARAMS
```

Because some postscript interpreters or viewers may have problems with binary resources or with the DSC comments used in eps files, it is possible to encode all bypassed resources in the ASCII base85 format. This option can be activated by enabling the **use ASCII base 85 encoding** checkbox.

Turning on this option may lead to somewhat bigger output size and slower conversion speed.

It is strongly recommended to turn on the option **Hold all resources in Printer VM**, if the printer is equipped with enough memory to do this.

If this is the case the resource data is inserted only on the first page it is used and remains in the printer memory for all following pages of the print job.

This memory image can be used on every page where this resource is needed. If you have printjobs with many thousands of pages this will drastically reduce the size of the output data stream and also leads to faster conversion speed.

For passing through resources it is necessary to recognize the format of the resource. If the `%%BoundingBox:` statement is missing in the header of a postscript resource file, it may happen that this resource is excluded from the bypass mode. If you experience such a problem try to enable the **Disable EPS BoundingBox Check** option.

Activating the **Copy Prolog** option makes sense only, if you plan to convert mixed Postscript / VIPP input data and the input data contains a prolog with postscript procedures necessary for executing the passed through eps or ps resources. This is the case for example with input datastreams generated by XMPie uProduce.

After activating this option the postscript input data between the DSC comments

```
%%BeginProlog
```

```
and
```

```
%%EndProlog
```

is copied to the prolog section of the postscript output datastream. Additionally the checking for special (non postscript binary) content inside the passed through eps and ps resources is disabled. All special requirements of the passed through eps and ps resources must be handled by the procedures stored in the copied prolog data.

PCL driver specific settings:

Enable edge to edge mode

Activate this setting, if you want to use the edge to edge mode of your PCL printer. Additional to this setting you have to activate the edge to edge mode at the printer side ! You can do this at the printer panel or by sending the appropriate PCL commands. The supplied **GenericPCLPrinter** ppd file contains PCL commands for the most common printer manufacturers. Please select or insert the correct PCL command for your printer.

See the next topic on how to do this.

If your printer does not support edge to edge printing or you use the wrong PCL command, activating the edge to edge option will result in wrong positioning of the page content.

If the option **Send PPD default PCL setup** is activated, the PCL commands specified as the default choice (**DefaultPCLSetup*)

for the **PCLSetup* key are inserted at the beginning of the job. Currently this option only works with the PCL output drivers.

You can insert your own PCL commands by modifying the content of the used ppd file, which is selected indirectly

via the **Printer** model name in the [Common](#) tab.

If you want to learn more about ppd files please see the **PostScript® Printer Description File Format Specification** from Adobe®.

PDF output settings

Quality

Default
Screen Optimized
ebook
Print Optimized
Prepress Optimized

⚠ **Screen Optimized** provides the lowest quality and the smallest file size of the PDF output and **Prepress Optimized** delivers best quality at the cost of bigger file sizes .

More information about this predefined conversion settings can be found in the Adobe® Acrobat Distiller manual.

Auto rotate pages

If you activate this option, the interpreter tries to find the correct page orientation for each page depending on the major text direction and then rotating the page according to this text orientation. This option should be used if the intend of the output is for viewing not for printing.

Generate PDF/A-1b compatible output

Activate this option, if you have to generate PDF/A-1b compatible output. You have to specify a ICC profile for generating PDF/A-1b compatible PDF files. More information about ICC profiles as well as some downloadable profiles can be found e.g. here:
<http://www.eci.org/doku.php?id=de:downloads>

6.3.22.3 VIPP Resources

VIPP® Resources contains chapters with additional information about resources:

[Finding and Preparing the VIPP resources](#)

[Standard Fonts](#)

[Additional Fonts](#)

[Encoding vectors](#)

[Inserting additional resource files](#)

[Default properties and predefined system parameters](#)

[Replacement of Ghostscript standard fonts](#)

6.3.22.3.1 Finding and Preparing the VIPP resources

To get the Mercury VIPP® emulation running, it is necessary to retrieve all VIPP® resources belonging to the VIPP® variable data stream.

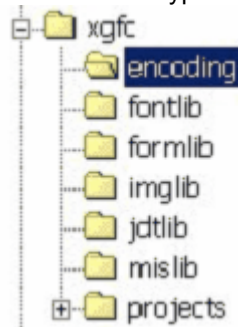
Normally these resource files can be found on the internal hard disk of the Xerox printer used for printing the job, on the associated printer controller or somewhere in the data folders of the VIPP® document design software used for creating this resources.

Sometimes the VIPP® commands, which have to be executed before the first variable data content, are stored separately from the variable data and are appended (or sometimes even dynamically created) by some kind of shell script before the data stream is send to the printer. All the information which is necessary to create the variable data stream with this first VIPP® commands has to be supplied.

Often it happens that the Xerox VIPP® interpreter default configuration files are changed for converting customer specific data. Then you have to find the changes made to this files and add the changed VIPP® code lines to a additional resource file.

Please see the [Default properties and predefined system parameters](#) and [Inserting additional resource files](#) topics to find out how this can be done.

These are the typical folder names for storing project specific customer VIPP® resources:



All the files in these folders, belonging to the variable data stream, which should be converted, have to be copied to the equal named sub folders of the Mercury VIPP® Global resource path.

This folder (named „..\vipp“) normally can be found within the ..\ProgramData\docuForm\Mercury\Res\ subdir (Win7) resp. ..\Dokumente und Einstellungen\All Users\Anwendungsdaten\DocuFORM\Mercury\Res\ subdir (WinXP) in the sub folder „..\Res“. The location of this folder can be changed within the GUI of the emulation configuration.

If you are not sure, which files belong to the variable document data stream, copy all the files in this folders to the appropriate Mercury VIPP® resource sub folders.

The ..\encoding folder contains font libraries (*.pfb, *.pfa) encoding tables and font lists.

The ..\formlib folder contains static form-, overlay resource- and database master files (*.frm, *.xgf, *.dbm)

The ..\imglib folder contains image resources (*.jpg, *.tif).

The ..\jdtlib folder contains the Job descriptor ticket files (*.jdt).

The ..\mislib folder can contain miscellaneous resource files.

If the first VIPP® commands before the variable data contains a line like this:

```
[ (projects) (myproject) ] SETPROJECT
```

all VIPP® resource files have to be placed in the sub folder `..\myproject` of the folder `..\projects`.

Additionally in this case all font related resources should be copied to the global `..\encoding` folder. This is necessary because Ghostscript registers all fonts at startup time, but the project resource folder is specified at runtime. Therefore Ghostscript could not register fonts, which are stored in the project resource folder only.

Please keep in mind, that the file extensions and folder names for VIPP® resources are not fixed and can be redefined inside the VIPP® code of the project. So files with any or no extension residing in some other subdirectory of the printer file system can be used as a VIPP® resource.

It is possible to use pure Postscript® resources inside the VIPP® project data. Due to the nature of Postscript® as a programming language it can happen, that, in a badly designed VIPP project, it is not enough to copy the original resources to the [Mercury](#) resource folders. If project resources contains local customizations e.g. absolute path definitions for file resource locations, it may be necessary to modify those original resource files or inserting patch commands in an additional resource file.

Absolute path definitions can be ignored during conversion for specific types of resources. The SETXPATH VIPP® commands can be disabled by placing for example the following statement in a additional resource file:

```
/SETIPATH {pop} def
```

The line above disables the SETIPATH command, which redefines the folder(s) searched for image resources.

The additional VIPP® parameter `/IgnorePath` can be used to disable path extensions specified in almost all VIPP commands which accepts filenames as a parameter. The following line coded in a additional resource file leads to ignoring path extensions used when calling image resources e.g. with ICALL and referencing fontfiles e.g. with SETENCODING:

```
[ /IgnorePath << /fnt true /img true >> ] SETPARAMS
```

Syntax for the `/IgnorePath` parameter:

```
/IgnorePath << dirkey1 true/false dirkey2 true/false ...dirkeyn true/false>>
```

Valid values for *dirkey* are: `/fnt`, `/enc`, `/frm`, `/jdt`, `/img` and `/mis`. *true* as *dirkey value* enables the ignore path option for the specified resource folder.

As a part of the [Mercury](#) VIPP® emulation configuration options it is possible to specify an additional PostScript® or VIPP® resource file which can be created and edited by the user. This file should be placed in the [Mercury](#) VIPP® global resource path (`..\Res\vipp`).

The content of this file is inserted in the data stream directly before the VIPP® variable data.

The usage of this additional VIPP® resource can be necessary for example, if the VIPP® project uses printer resident fonts, which are no longer available at the new (non Xerox®) target printer.

Fonts which are not found during the data conversion are substituted by default fonts, which not always deliver visual satisfying results.

Example:

VIPP® code for adding fonts in the additional VIPP® resource file:

```
[/NMOR (monospace821.pfb)] (myencoding) SETENCODING  
[/NMOB (monospace821 Bold.pfb)] null SETENCODING
```

The two VIPP® code lines above make the fonts monospace821 and monospace821_Bold with the VIPP® font names NMOR and NMOB available within the project.

The first font is encoded using the VIPP® encoding vector „myencoding" (stored in the file „..\encoding\myencoding.").

The second font uses the default encoding vector.

The font library files „monospace821.pfb" and „monospace821_Bold.pfb" have to be placed in the ..\encoding VIPP® resource directory.

If you need more information about the [Mercury](#) font handling, take a look at the chapters [Standard Fonts](#), [Additional Fonts](#) and [Encoding vectors](#).

6.3.22.3.2 Standard Fonts

Mercury directly supports the 35 PostScript® standard fonts, which are resident in every PostScript® Printer.
(See also [Font mapping Table](#))

For each of these fonts an VIPP® font name exists.
Additional 16 VIPP® fonts, that are resident in some Xerox® printers, are mapped to one of the 35 standard fonts included with Ghostscript.
For licence reasons Ghostscript uses his own optical equivalent fonts with different PostScript® font names.
The mapping from the VIPP® font names over PostScript® font names to the Ghostscript font names takes place automatically.
For details take a look at the font mapping table.

6.3.22.3.3 Additional Fonts

If unknown fonts are detected during the conversion of VIPP® data, they would be substituted automatically by „Courier" and a font substitution warning message occurs in the log window, like for example:

„Font NORCB not found, substituted by: Courier" .

If a the missing font is referred directly by his font file, no substitution is made, the conversion stops and an error message like for example:

„/VIPP®_unable to locate..."

referring to the missing resource, occurs.

If more fonts are required for converting the data or a Ghostscript substitute font should be replaced by an original font, you have to copy the required font files to the sub folder *..encoding* of the „Global resource path".

If a specific project resource path (For example: *...\\vippl\myproject*) is created for a VIPP® data project via the SETPROJECT command, you also have to copy the resource file in this sub folder.

Mercury supports all fonts in the „pfa" and „pfb" format.
The required font files are normally located in the sub folder *...encoding* of the original Xerox® VIPP® installation.
Unfortunately these file often don't have self-explanatory names and the usual extensions "pfa" or „pfb".
With the help of a text editor you can detect if a suspicious file is a font resource.

A pfa font file usually begins with a character string like „%!PS-AdobeFont-1.0".
In a pfb font file six binary bytes are prefixed to this string.

You can find the PostScript® font name by searching in a font file for the character string „/FontName“.

After this character string you'll directly find the font name, which could be used by VIPP®- or Postscript®- code.

Mapping between VIPP®- and PostScript® font name:

Usually in VIPP® resources the font is not referred by his PostScript® font name, but by a VIPP® font name, which is assigned to the appropriate PostScript® font name.

To find the mapping between this two names, search for the VIPP® font name in the VIPP® resources of a project, which you got from the font substitution alert message and find this VIPP® font name and the PostScript® font name in front of one of the VIPP® commands INDEXFONT or SETENCODING.

Now the font file with the PostScript® font name used in this commands have to be found and copied to the `..encoding` sub folder.

Very often project specific fonts will be defined by manipulating the default properties of the VIPP® interpreter, instead of using the project specific resources.

In this case the mapping command SETENCODING may be missed for these particular fonts, also resulting in a font substitution warning message.

In the Xerox® VIPP® Installation the default font list is located in a file named *fontlist*. These fonts are registered by executing the content of the *xgf.def* file, in which also more VIPP® parameters will be initialized with default properties. Within the *xgf.def* file a SETENCODING command is executed with the used font list file name as the argument. Please find this file in the `..encoding` sub folder and inspect its content for the missing font name from the font substitution warning message. Every line in this font list file maps a VIPP® font name to a Postscript® font.

Because in Mercury all default properties including the registration of the standard fonts are stored internal, project specific properties, which in Xerox® VIPP® are executed by manipulating the default properties, have to be defined in Mercury in an additional resource file.

This additional resource file could be created with every text editor and his name is given by the customer.

In this file for every missing font mapping a additional SETENCODING command have to be added.

Example:

Create a text file named „newres.txt“

The PostScript® font is /OCRB defined in font file OCRB____.pfb and should be registered under the VIPP® font name /NOCRB.

To register you have to insert one of the 3 following VIPP® statements in the additional resource file („newres.txt“).

Option 1, directly over the font file:

```
[/NOCRB (OCRB .pfb) null SETENCODING
```

Option 2, indirect over the PostScript® font name, defined in the font file (OCRB____pfb):

```
[/NOCRB /OCRB] null SETENCODING
```

If the VIPP® interpreter already knows a font named /OCRB before executing this statement the known font will be used and registered as /NOCRB, not the font defined in the OCRB____.pfb file.

⚠ If, during the execution of this SETENCODING command, Ghostscript does not find a font file, which have defined /OCRB as the PostScript® font name at startup time, Ghostscript

substitutes this font without any substitution warning !

Option 3, creating a separate font list file:

```
(myfontlist) null SETENCODING
```

In this option the mapping between the VIPP® font name and the PostScript® font name is saved in the file "myfontlist"

located in the sub folder `..\encoding` of the global resource path. This option can be used to register more than one project

specified font or if a project font list file already exists.

In this example the content of myfontlist is composed of only one statement:

```
/NOCRB (OCRB .pfb)
```

See also [Encoding vectors](#)

Encoding vectors

The function SETENCODING expect as second parameter the name of a Postscript® encoding vector.

The encoding vector defines the mapping of single characters (ASCII Value), which are defined by a text string of one of the VIPP® text output functions, to a character graphic, which is defined in the font file.

If the encoding vector name is *null* the mapping table from the font file will be used.

You may have to adjust the encoding vector, if region specific character occurs like German

Ä,Ö,Ü,ä,ö,ü,...

For all standard fonts except the symbol fonts /Symbol and /ZapfDingbats the VIPP® interpreter implements automatically the internal stored encoding vector `__PCSUN__`, which included a for the German speaking region modified version of the standard PostScript® encoding vector /ISOLatin1Encoding.

If the internal standard encoding vector should be used for the project specific font list *myfontlist*, instead the font internal encoding vector, the VIPP® command have to look like this:

```
(myfontlist) ( pcsun ) SETENCODING
```

For VIPP® data whose origin is the US American speaking area, it could be necessary to change the default setting of the encoding vector.

Changing the encoding vector for standard fonts could be done in the register "**Resources**" of the VIPP® emulation configuration GUI by activating the option "**Use VIPP® encoding file**" and select the file vector by browsing to the file with the attached browse button.

In project specific resources of the original Xerox® VIPP® installation often the standard encoding vector file "*pcsun*" from will be customized without changing the name. Normally you find the encoding vector used for the standard fonts of the project in the subfolder `..\encoding` of the global resource path.

The corresponding SETENCODING command is normally located in the file *xgf.def* which can be found in the sub folder `..\xgf\src\` of the Xerox® VIPP® installation.

Using CID fonts

CID fonts are PostScript resources containing a large number of glyphs (e.g. glyphs for Far East languages, Chinese, Japanese and Korean). Please refer to the PostScript Language Reference, third edition, for details.

CID font resources are a different kind of PostScript resource from fonts. In particular, the SETENCODING command cannot be used. CID font resources must first be combined with a CMap resource, which defines specific codes for glyphs, before it can be used as a font. This allows the reuse of a collection of glyphs with different encodings.

The simplest method to request a font composed of a CID font resource and a CMap resource in a VIPP® document is using a fontname build like /CIDFont-CMap where CIDFont is a name of any CID font resource, and CMap is a name of a CMap resource designed for the same character collection. The interpreter will compose the font automatically from the specified CID font and CMap resources. Font names build like explained above then can be used with the INDEXFONT or SETFONT command.

Another method is possible using the composefont postscript operator.

CID fonts must be placed in the /mislib/CIDFont/ subdirectory of the [Global resource path](#). CMaps must be placed in the /mislib/Cmap/ subdirectory of the [Global resource path](#).

6.3.22.3.4 Inserting additional resource files

As a part of the Mercury VIPP® emulation configuration options it is possible to specify an additional PostScript® or VIPP® resource file which can be created and edited by the user.

This file should be placed in the Mercury VIPP® global resource path (..\Res\vipp).

The content of this file is inserted in the data stream directly before the VIPP® variable data.

All project specific settings, which are not already done inside the available project resource files, have to be inserted in this file. This can be, for example, changes made to the Xerox® default setting files, which are not used by the Mercury VIPP® emulation (see also [Default properties and predefined system parameters](#)).

After inserting all necessary project specific parameters to the additional resource file, this file have to be stored in the global resource path.

If you have to execute more than one additional resource file you can use the VIPP® command RUN inside the first additional resource file to execute another file. The files executed by the RUN command have to be placed in the *mislib* resource folder.

For example the command

```
(xgf.def) RUN
```

will execute the content of the file xgf.def located in the *..\mislib* subfolder of the VIPP® global resource path.

For executing the content of this resource file at the start of the VIPP® interpreter, you have to activate the option "Use additional VIPP® or PostScript® resource file" in the configuration dialog of the VIPP® emulation.

You select the file by browsing with the attached Browse... button to it.

6.3.22.3.5 Default properties and predefined system parameters

Within Xerox® VIPP® the default settings and the most predefined parameters are stored in files, which are coded in VIPP® native mode. These files are loaded and their content is executed every time the VIPP® interpreter is started.

Usually these files can be found in the subfolder `..xgf\src` of the Xerox® VIPP® installation.

The following files belong to these static resources:

Filename:	Contents:
xgfunix.run or XGFDOS.RUN	search path definitions for all types of VIPP® resources.
xgf.pcc	definitions of PCC tables
xgf.gep	color and fill patterns, graphics element properties
xgf.mup	definitions needed for physical pages which contains more than one logical page
xgf.bat	background attribute definitions
xgf.def	initialization of various default parameters, loading of the default font list from the file <i>font list</i>

Because of system integrity reasons inside **Mercury** the contents of these files are integrated hard coded in the system. Therefore these files are used for the data conversion.

Your first look should go to the file `xgfunix.run` or `XGFDOS.RUN`. Which one is used for the actual installation depends on the operating system.

This file contains all search path definitions, using the VIPP® commands `SETFPATH`, `SETIPATH`, `SETJPATH`, `SETEPATH`, `SETMPATH` and `SETPPATH`. During the interpretation of the VIPP® jobs all associated external resources are searched in the folders defined here.

Unfortunately the contents of these Xerox® files are often manipulated by VIPP® programmers to adjust the parameters for their project specific needs, instead of using project specific resource files for those modifications.

If new parameters are inserted "*undefined*" errors may occur during conversion, if existing parameters are changed some differences in the presentation of the data can occur.

In both cases it is necessary to examine the Xerox® resource files for project specific changes to the original content, extract the particular VIPP® native mode code lines and insert these lines in an additional project specific resource file (see [Inserting additional resource files](#)).

Example:

In the VIPP resources of a project the color definition named `MYDARKBLUE` is used, but nowhere defined in the project resource files.

During conversion you will get the error message:

ERROR: undefined OFFENDING COMMAND: MYDARKBLUE ...

If this happens you have to look in the Xerox® `xgf.gep` file for a line which looks like this:

```
/MYDARKBLUE [0 0 .5] SETCOL
```

Copy this line in an additional resource file and include this file in your project configuration.

 Though this color definition should be found in the `xgf.gep` file, it is possible that those

additional definitions are inserted in every other resource file which is loaded at the start of the Xerox® VIPP® interpreter, e.g. the *xgf.def* file.

6.3.22.3.6 Replacement of Ghostscript standard fonts

If you have own font files for PostScript® standard fonts like /Helvetica and these font files should be used instead of the optical similar Ghostscript fonts, it is not sufficient to copy your font libraries to the sub folder

...\encoding

of the global resource path.

Additionally you have to disable the mapping between the PostScript® font name and the Ghostscript font name.

You have to modify the Ghostscript font mapping file "Fontmap.GS", located in the sub folder ...lib of the Ghostscript installation folder.

You find the list following the "%Aliases" comment line in the font mapping file. Comment out the lines with the font names, which should be replaced with your own fonts.

Example:

In the case of replacing Helvetica the modification of Fontmap.GS look like this:

```
...
%Aliases
...
% /Helvetica           /NimbusSanL-Regu;
/Helvetica-Oblique     /NimbusSanL-ReguItal;
/Helvetica-Bold        /nimbusSanL-Bold;
/Helvetica-BoldOblique /NimbusSanL-BoldItal:...
```

6.3.22.4 Appendix

The appendix give some additional Information.

[Font mapping table](#)

[Handling of ppd files with non standard keywords](#)

6.3.22.4.1 Fontmapping Table

VIPP® font name	Postscript font name	VIPP® internal substitution	Ghostscript font name	Ghostscript font file
/NCR	/Courier		/NimbusMonL-Regu	n0220031.pfb
/NCRB	/Courier-Bold		/NimbusMonL-Bold	n0220041.pfb
/NCRO	/Courier-Oblique		/NimbusMonL-ReguObli	n0220231.pfb
/NCRBO	/Courier-BoldOblique		/NimbusMonL-BoldObli	n0220241.pfb
/NHE	/Helvetica		/NimbusSanL-Regu	n0190031.pfb
/NHEB	/Helvetica-Bold		/NimbusSanL-Bold	n0190041.pfb
/NHEO	/Helvetica-Oblique		/NimbusSanL-ReguItal	n0190231.pfb
/NHEBO	/Helvetica-BoldOblique		/NimbusSanL-BoldItal	n0190241.pfb
/NHEN	/Helvetica-Narrow		/NimbusSanL-ReguCond	n0190431.pfb
/NHENB	/Helvetica-Narrow-Bold		/NimbusSanL-BoldCond	n0190441.pfb
/NHENO	/Helvetica-Narrow-Oblique		/NimbusSanL-ReguCondItal	n0190631.pfb
/NHENBO	/Helvetica-Narrow-BoldOblique		/NimbusSanL-BoldCondItal	n0190641.pfb
/NBKL	/Bookman-Light		/URWBookmanL-Ligh	b0180121.pfb
/NBKLI	/Bookman-LightItalic		/URWBookmanL-LightItal	b0180321.pfb
/NBKD	/Bookman-Demi		/URWBookmanL-DemiBold	b0180151.pfb
/NBKDI	/Bookman-DemiItalic		/URWBookmanL-DemiBoldItal	b0180351.pfb
/NTMR	/Times-Roman		/NimbusRomNo9L-Regu	n0210031.pfb
/NTMB	/Times-Bold		/NimbusRomNo9L-Medi	n0210041.pfb
/NTMI	/Times-Italic		/NimbusRomNo9L-ReguItal	n0210231.pfb
/NTMBI	/Times-BoldItalic		/NimbusRomNo9L-MediItal	n0210241.pfb
/NPAR	/Palatino-Roman		/URWPaladioL-Roma	p0520031.pfb
/NPAB	/Palatino-Bold		/URWPaladioL-Bold	p0520041.pfb
/NPAI	/Palatino-Italic		/URWPaladioL-Ital	p0520231.pfb
/NPABI	/Palatino-BoldItalic		/URWPaladioL-BoldItal	p0520241.pfb
/NINCSR	/NewCenturySchlbk-Roman		/CenturySchL-Roma	c0590131.pfb
/NINCSB	/NewCenturySchlbk-Bold		/CenturySchL-Bold	c0590161.pfb
/NINCSI	/NewCenturySchlbk-Italic		/CenturySchL-Ital	c0590331.pfb
/NINCSBI	/NewCenturySchlbk-BoldItalic		/CenturySchL-BoldItal	c0590361.pfb
/NAGB	/AvantGarde-Book		/URWGothicL-Book	a0100131.pfb
/NAGBO	/AvantGarde-BookOblique		/URWGothicL-BookObli	a0100331.pfb
/NAGD	/AvantGarde-Demi		/URWGothicL-Demi	a0100151.pfb
/NAGDO	/AvantGarde-DemiOblique		/URWGothicL-DemiObli	a0100351.pfb
/NSY	/Symbol		/StandardSymL	s0500001.pfb
/NZDB	/ZapfDingbats		/Dingbats	d0500001.pfb
/NZCMI	/ZapfChancery-MediumItalic		/URWChanceryL-MediItal	z0030341.pfb
/NHEL	/Helvetica-Light	/Helvetica	/NimbusSanL-Regu	n0190031.pfb
/NHEBL	/Helvetica-Black	/Helvetica-Bold	/NimbusSanL-Bold	n0190041.pfb
/NHELO	/Helvetica-LightOblique	/Helvetica-Oblique	/NimbusSanL-ReguItal	n0190231.pfb
/NHEBLO	/Helvetica-BlackOblique	/Helvetica-BoldOblique	/NimbusSanL-BoldItal	n0190241.pfb
/NHEC	/Helvetica-Condensed	/Helvetica-Narrow	/NimbusSanL-ReguCond	n0190431.pfb
/NHECB	/Helvetica-Condensed-Bold	/Helvetica-Narrow-Bold	/NimbusSanL-BoldCond	n0190441.pfb
/NHECO	/Helvetica-Condensed-Oblique	/Helvetica-Narrow-Oblique	/NimbusSanL-ReguCondItal	n0190631.pfb
/NHECBO	/Helvetica-Condensed-BoldObli	/Helvetica-Narrow-BoldOblique	/NimbusSanL-BoldCondItal	n0190641.pfb
/NKR	/Korinna-Regular	/Palatino-Roman	/URWPaladioL-Roma	p0520031.pfb
/NKRB	/Korinna-Bold	/Palatino-Bold	/URWPaladioL-Bold	p0520041.pfb
/NKRI	/Korinna-KursivRegular	/Palatino-Italic	/URWPaladioL-Ital	p0520231.pfb
/NKREI	/Korinna-KursivBold	/Palatino-BoldItalic	/URWPaladioL-BoldItal	p0520241.pfb
/NGML	/Garamond-Light	/Times-Roman	/NimbusRomNo9L-Regu	n0210031.pfb
/NGMB	/Garamond-Bold	/Times-Bold	/NimbusRomNo9L-Medi	n0210041.pfb
/NGMLI	/Garamond-LightItalic	/Times-Italic	/NimbusRomNo9L-ReguItal	n0210231.pfb
/NGMBI	/Garamond-BoldItalic	/Times-BoldItalic	/NimbusRomNo9L-MediItal	n0210241.pfb

6.3.22.4.2 Handling of ppd files with non standard keywords

❗ Even if a PostScript® printer with the correct PPD-file has been selected, it may happen that nothing is selectable within the mapping dialogs. This can occur, if the printer manufacturer did not use the intended standard keywords (e.g. *InputSlot* for the available input media) for finding the code of the enhanced device features within the PPD file. In that case you have to options to add the none standard keywords of this printer into the scope of the VIPP™ and Postscript processor (prerequisites some basic knowledge of the format of PPD-files):

Option 1: Create two additional text configuration files for mapping the non standard keywords to the standard keywords:

- a) Locate the used keyword within the PPD file.
- b) Create a file 'nostdppd.txt' within the **global resource path**.
- c) Insert a line with the name of the PostScript® printer model (the '*modelName:' entry within the PPD file, also displayed within the list of 'PostScript® printer') enclosed in double quotation marks (") into 'nostdppd.txt'
- d) Create a file with the name of the PostScript® printer model (displayed within the list of 'PostScript® printer') within the resource path.
- e) Insert a line with the standard keyword and the used substituted keyword of the PPD file enclosed in double quotation marks (") into the PostScript® printer model file.

Example1:

The printer manufacturer 'UnknownOEM' uses for his printer model 'OEMPrinter' within the provided PPD file the keyword *InputBin* instead the standard *Inputslot* for the input media definitions, and the keyword *OEMStaple* instead the standard *StapleLocation* for the staple options.

In that case the file 'nostdppd.txt' contains the line :

```
"OEMPrinter"<NL>
```

If you use more than one printer with non standard keywords, you have to add a new line containing the '*modelName:' entry for each printer.

The file 'OEMPrinter' must exist and contains the 2 redefinition lines :

```
"Inputslot""InputBin"<NL>
"StapleLocation""OEMStaple"<NL>
```

NOTE: <NL> stands for new line = [0x0D]0x0A

Example2:

This part of the contents from ppd file for the Fiery S450 65C-KM PBS Color Server shows the relevant sectors for finding the model name and selecting the output bin.

...

```
*NickName: "Fiery S450 65C-KM PS Color Server v1.0 eu"
*modelName: "Fiery S450 65C-KM PS Color Server v1.0 eu"
*Manufacturer: "EFI"
```

...

```

*%EFIFlags *EFOutputBin Column|Command|Impose|Spooler
*%EFIGroup *EFOutputBin FPFinishing/Finishing
*OpenUI *EFOutputBin/Papierausgabemagazin:PickOne
*OrderDependency: 50.0 AnySetup *EFOutputBin
*DefaultEFOutputBin: AutoSelect
*EFOutputBin AutoSelect/Automatisch: "
/XJXsetOutputBinV2 where { pop 99 XJXsetOutputBinV2 } if "
*End
*EFOutputBin Outbin1/Hefter-Hauptablage: "
/XJXsetOutputBinV2 where { pop 4 XJXsetOutputBinV2 } if "
*End
...

```

In this example the keyword for the output bin is "EFOutputBIN".

The Content of the nostdppd.txt file has to be:
"Fiery S450 65C-KM PS Color Server v1.0 eu"

The file name for the second additional resource file is "Fiery S450 65C-Kb PS Color Server v1.0 eu".

This file has to contain the following statement:
"OutputBin""EFOutputBin"

Because this printer uses the standard keyword "InputSlot" for the input slot selection, no additional line for this feature has to be added.

Option 2: Add a KeyReplacements section to the ppd file of the printer:

2) You can create a local customized ppd file of your printer by adding a mapping option in this file, which replaces the manufacturer specific device keywords with the emulation internally used Adobe® conform keywords. You can do this by editing the ppd file with a text editor.

You have to add the following statements at the end of the ppd file:

Syntax:

```

*OpenUI *KeyReplacements
*KeyReplacements <Adobekey1>: <Replacekey1>
...
*KeyReplacements <Adobekeyn>: <Replacekeyn>
*CloseUI *KeyReplacements

```

The following Example replaces the Keyword EFOutputBin used in the ppd file for Outputbin selection with the emulation internally used Adobe® conform keyword OutputBin :

```

*OpenUI *KeyReplacements
*KeyReplacements OutputBin: EFOutputBin
*CloseUI: *KeyReplacements

```

It is recommended but not necessary to change the modelname entry in the modified ppd file and rename the filename of the ppd file.

This helps distinguishing the modified file from the original one.

```
*ModelName: "Fiery S450 65C-KM modified"
```

If you want to learn more about ppd files please see the **PostScript® Printer Description File Format Specification** from Adobe®.

6.3.22.4.3 Concatenating raw data files with VIPP startup commands

Raw linemode or database mode data files needs to be prefixed with at least one VIPP® startup command like STARTLM or STARTDBM, which calls a job descriptor ticket or a database master. Normally the startup line(s) are already inserted at the beginning of the data files, before the files are fed to the Mercury VIPP® emulation.

If this preprocessing step is missing, you can use the special VIPP command INSERTLMHEADER coded in a additional resource file to prefix your data files with some VIPP commands depending on a part of the data file name.

Syntax INSERTLMHEADER:

```
(startuptxtfile.txt) /F INSERTLMHEADER
```

or

```
(startuptext) /S INSERTLMHEADER
```

In the first case you have to code the VIPP® startup lines in a text file stored in the `../mislib` resource folder.

Using the second syntax option, you can specify the VIPP startup commands directly as a string argument of the INSERTLMHEADER command.

If you use INSERTLMHEADER together with the VIPP® variable `VIPP_JobTitle` and some other VIPP commands like CASE and ENDCASE, you can switch between different startup lines depending on (a part of) the name of the data file.

Example:

Assuming you have linemode data files named like

```
type_a_0001.dat
type_a_0002.dat
...
type_b_0001.dat
type_b_0002.dat
...
```

The files beginning with `type_a` needs to be processed with a job descriptor ticket (JDT) stored in the file `typea.jdt` and the files beginning with `type_b` needs to be processed with a JDT stored in `typeb.jdt`.

All resources used for processing the data files should be stored in the project subfolder

`../projects/myproject/` of the [Global resource path](#).

Then you have to add the following lines of VIPP® code in a additional resource file (see [Inserting additional resource file](#) for more information how to use a additional resource file):

Read first 6 characters of the data file name and store them in a variable named

```
job_datatype.
```

(Of cause you can specify any other part of the filename by modifying the `getinterval` parameters. You can find more information about postscript commands like `getinterval` in the PostScript® Language Reference third edition) .

```
VIPP JobTitle 0 6 getinterval /job datatype exch def
```

Load appropriate VIPP header depending on the first six chars of the data file name:

```

CASE job datatype { % default action enclosed in {},
                  % executed if job datatype is not equal to any
of the following choices
                  % could be e.g. a errormessage:
                  VIPP JobTitle /e ni v error
                  }
(type a) % if job datatype is equal to "type a" (the first six
chars of the data file name)
        % execute the following VIPP native mode code (enclosed
in{}):
        {
          ((projects) (myproject)] SETPROJECT
          (typea.jdt) STARTLM)
          /S INSERTLMHEADER
        }
(type b) % if job datatype is equal to "type b" (the first six
chars of the data file name)
        % execute the following VIPP native mode code (enclosed
in{}):
        {
          ((projects) (myproject)] SETPROJECT
          (typeb.jdt) STARTLM)
          /S INSERTLMHEADER
        }
(...) { % ..... here you can add additional VIPP header code
for other job data types
      }
ENDCASE

```

6.3.23 XES to PCL5

6.3.23.1 Overview

The XES emulation from Mercury converts print data that were created in the XES programming language of Xerox, so that they can be output on PCL5 compatible page printers.

6.3.23.2 XES Commands

6.3.23.2.1 Supported commands

Syntax	Command
<ESC>+A,comment <CR>/<LF>	Font Add
<ESC>aX,Y <CR>/<LF>	Text Placement Absolute
<ESC>b	Bold Start
<ESC>+B,comment <CR>/<LF> font list <CR>/<LF>	Font Delete
<ESC>+B,comment <CR>/<LF> form.FRM list <CR>/<LF>	Form Delete
<ESC>cn	Select Paper Tray
<ESC>+Cn,comment <CR>/<LF>	Multiple Copies (Collated) (see Note 1)
<ESC>d	Tabs Clear Horizontal
<ESC>e	Tabs Clear Vertical
<ESC>+En,comment <CR>/<LF>	Multiple Copies (Non Collated)
<ESC>+F,comment <CR>/<LF>	Font Load
<ESC>gaT;X,Y,r,S1,E1,Sn,En <CR>/<LF>	Graphic Arc
<ESC>gbT;X1,Y1, X2,Y2 <CR>/<LF>	Graphic Box
<ESC>gIT;X1,Y1, X2,Y2,Xn,Yn <CR>/<LF>	Graphic Line
<ESC>gpT;X,Y,r,%1/F1/O1,%n/Fn/On <CR>/<LF>	Graphic Pie Chart
<ESC>grM(or Mx/My),I,R;X,Y, Xn,Yn <CR>/<LF>	Graphic Window Repeat
<ESC>gwm(or Mx/My),I,R;X,Y, Sx,Sy <CR>/<LF>	Graphic Window
<ESC>h	Superscript Start
<ESC>in	Line Spacing
<ESC>ipn <CR>/<LF>	Line Spacing Absolute
<ESC>j	Justification Start
<ESC>k	Justification Stop

<ESC> I	Subscript Start
<ESC> +M ,comment <CR>/<LF>	Merge Page Load
<ESC> m H,T,B,L,R <CR>/<LF>	Margins
<ESC> n	Font Change
<ESC> +n font name <CR>/<LF>	Font ID Assignment
<ESC> +n form name. FRM <CR>/<LF>	Form ID Assignment
<ESC> +N form name <CR>/<LF>	Form Load
<ESC> p	Bold Stop
<ESC> +P ,comment <CR>/<LF>	Print Job (single font orientation)
<ESC> q	Center
<ESC> +Q ,comment <CR>/<LF>	Print Job (mixed orientation)
<ESC> r Dnc	Text Placement Relative
<ESC> s	Super/Subscript Stop
<ESC> +T ,comment <CR>/<LF> data record <CR>/<LF>	Character Reassignment (at job boundaries)
<ESC> tn 1,n2,..., n8 <CR>/<LF>	Tabs Horizontal
<ESC> u	Underline Start
<ESC> +U ,comment <CR>/<LF>	Font Unload
<ESC> +V ,comment <CR>/<LF>	Merge Page Unload
<ESC> vn 1,n2,..., n8 <CR>/<LF>	Tabs Vertical
<ESC> w	Underline Stop
<ESC> +X ,comment <CR>/<LF>	Reset
<ESC> x X,Y,L,T,S <CR>/<LF>	Line Draw X
<ESC> y X,Y,L,T,S <CR>/<LF>	Line Draw Y
<ESC> za X,Y <CR>/<LF>	Baseline Placement Absolute
<ESC> zbn	Form Merge Start
<ESC> zd	Merge Stop
<ESC> ze	Merge Start
<ESC> zf	Units 1/300 Inch
<ESC> zg	Units 1/60 Inch
<ESC> zhc	Form Merge Stop
<ESC> zj	Justification Unlimited Start
<ESC> zkl <CR>/<LF>	Margin Left
<ESC> zlc	Language (see Note 2)
<ESC> zmR <CR>/<LF>	Margin Right
<ESC> znT <CR>/<LF>	Margin Top
<ESC> zox	Overstrike Start
<ESC> zop	Overstrike Stop
<ESC> zrn	Output Tray
<ESC> zsnc	Character Spacing Absolute
<ESC> zt <CR>/<LF> data record <CR>/<LF>	Character Reassignment (within a job)
<ESC> zx form1,form2,form3, form4 <CR>/<LF>	Cycleforms
<ESC> zya comment text <CR>/<LF>	Comment
<ESC> zydn <CR>/<LF>	Duplex Start
<ESC> zye <CR>/<LF>	Duplex Stop
<ESC> zyfn <CR>/<LF>	Duplex Invert Start
<ESC> zyin	Duplex Side Select
= UDK =c	User Defined Key (UDK)

Note 1: Only supported if the PCL Printer supports collated copies (@PJM SET QTY = n, see parameter csp60). Otherwise treated as <ESC>**+En**,comment <CR>/<LF>

Note 2: See parameter Language for supported languages.

6.3.23.2.2 Not supported commands

Syntax	Command
<ESC> +B ,comment <CR>/<LF> graphic.GRA list <CR>/<LF>	Graphic Delete
<ESC> c	Channel Select Bisync
<ESC> +D	Data Monitor
<ESC> f N,M,P1,P2,P3 <CR>/<LF>	Assign Ink Color

<ESC> g fP;X,Y,Xn,Yn <CR>/<LF>	Graphic Fill (see Note 1)
<ESC> +G	Code Print
<ESC> +H ,message <CR>/<LF>	Operator Text Message
<ESC> HT c1,c2,...c255 <CR>/<LF>	Tabs Horizontal Set
<ESC> +J n <CR>/<LF>	Bisync
<ESC> +K n,comment <CR>/<LF>	Job Prioritize
<ESC> +I n	Page Format Select
<ESC> m H,T1,B1,L1,R1,m T2,B2,L2,R2 <CR>/<LF>	Job Summary Report
<ESC> o	Margins Double Page (see Note 2)
<ESC> +R n,comment <CR>/<LF>	Offset
<ESC> z c	Special Paper Enable
<ESC> z isc	Complex Print Mode
<ESC> z uX1,Y1,Xn,Yn,T,E,S <CR>/<LF>	Ink Change (see Note 3)
<ESC> z vcn1,n2,...,n8 <CR>/<LF>	Vector Draw
<ESC> z w	VFU Stops
<ESC> z ybc	VFU Stops Clear
<ESC> z yc	Bypass Paper Size or Manual
<ESC> z yh mS <CR>/<LF>	Feed Paper Size Select
<ESC> z ysv1,v2,v3,v4 <CR>/<LF>	Time Stamp
<ESC> z zn	Text Highlight
= MCK = command/language/ END <CR>/<LF>	Shift
	Plane Prioritize
	Mode Switching (see Note 4)

Note 1: Partially implemented. Boxes will be filled only.

Note 2: Partially implemented. (pure text pages only)

Note 3: Partially implemented. Works only with horizontal and vertical lines in 8 distinct shadings (instead 16).

Note 4: Only used for switching explicit into Transparent mode when:

=MCK= EMULATE/PCL/END

=MCK= EMULATE/POSTSCRIPT/END

or XES mode when:

=MCK= EMULATE/XDCS/END

=MCK= EMULATE/XES/END

Otherwise the command is ignored.

6.3.23.3 Parameters

6.3.23.3.1 Common Page Settings

Offsets

Normally a PCL5 printer for output in portrait format will leave a margin of 70 dots (6 mm) on both sides and a header and footer margin of 50 dots (4.25 mm) each, where it cannot print. A Xerox printer, on the other hand, can print over almost the entire area of a page.



Non-printable margin of a Xerox printer (very narrow)



Non-printable margin of a PCL5-compatible printer, if all the parameters are set to '0'. Neither the visible window, nor one of the vectors has been moved.

To define the print area, you can influence the output relative to the "short edge" (in portrait format the horizontal edge of the page using the parameters Visible window short edge offset and Vector short edge offset.

You can influence the output relative to the "long edge" (in portrait format the vertical edge of the page using the parameters Visible window long edge offset and Vector long edge offset.



This illustration shows the effect if the parameter Visible window long edge offset is set to "a" and the other parameters are set to "0". The visible window is moved away from the "long edge" by the value "a"/100 inches.



This illustration shows the effect if the parameter Vector long edge offset is set to "b" and the other parameters are set to "0". The visible window remains unchanged but the vectors have been changed by the value "b"/100" inches. This results in the text being moved horizontally.

If you only wish to move your output (not scale it), you can combine these parameters as you wish. The available step value is 1/100 inch.

- ⚠ The Mercury XES emulation offers these options for both Long edge feed, and Short edge feed. Which group of parameters is used depends on the setting of the paper tray used in Tray Mapping. If the paper is taken from a tray for which the option Short edge feed paper has been set, the parameters of the Short edge feed group will apply, otherwise those from the Long edge feed will apply.

Offsets used with Long Edge Feed

The following parameters are available to achieve a similarly large printable area using PCL5 printers:

- **Visible window short edge offset**

Define a value to move the image area downwards in portrait format in steps of 1/100 inch.

Default: 0.32 inch

- **Visible window long edge offset**

Define a value to move the image area to the right in portrait format in steps of 1/100 inch.

Default: 0.00 inch

- **Vector short edge offset**


Define a value to move the print area downwards in portrait format in steps of 1/100 inch.

Default: 0.00 inch

- **Vector long edge offset**

Define a value to move the print area to the right in portrait format in steps of 1/100 inch.

Default: 0.00 inch

 Please ensure that the parameters have been allocated to portrait format and the relevant edges in portrait format do not change. The long edge in portrait format remains the long edge even in landscape format.

Offsets used with Short Edge Feed

The following parameters are available to achieve a similarly large printable area using PCL5 printers as with Xerox printers:

- **Visible window short edge offset**

Define a value to move the image area downwards in portrait format in steps of 1/100 inch.

Default: 0.00 inch

- **Visible window long edge offset**

Define a value to move the image area to the right in portrait format in steps of 1/100 inch.

Default: 0.00 inch

- **Vector short edge offset**


Define a value to move the print area downwards in portrait format in steps of 1/100 inch.

Default: 0.00 inch

- **Vector long edge offset**

Define a value to move the print area to the right in portrait format in steps of 1/100 inch.

Default: 0.00 inch

 Please ensure that the parameters have been allocated to portrait format and the relevant edges in portrait format do not change. The long edge in portrait format remains the long edge even in landscape format.

Scaling

In some cases it is not sufficient to simply move the print area. The best solution in these cases is to adjust the XES form to the printable area of the PCL5 printer by means of scaling.

Scaling

- **Scaling short edge**

Define the percentage reduction in the horizontal direction in portrait format and the vertical direction in landscape format.

- **Scaling long edge**

Define the percentage reduction in the vertical direction in portrait format and the horizontal direction in landscape format

If you wish to output a document in portrait format, the parameter [Scaling short edge](#) allows the output to be scaled in terms of its height. The parameter [Scaling long edge](#), on the other hand, allows the output to be scale in terms of its width.

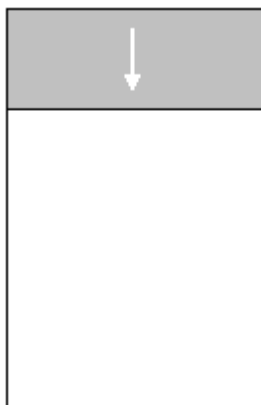
If moving the output does not produce the result you want, you can use the parameters [Scaling short edge](#) and [Scaling long edge](#) to change the size of the output, in other words to scale it. This will adjust existing XES forms to the printable area of the PCL5 printer.



Please note that all lines and positions can be scaled by this process.

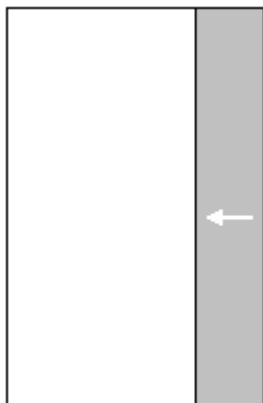
Examples:

Scaling short edge = 15,0



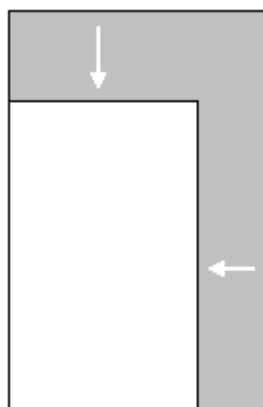
Reduction along the long edge of the page (in other words in the vertical direction) by 15%.

Scaling long edge = 15,0



Reduction along the short edge of the page (in other words in the horizontal direction) by 15%.

Scaling short edge = 15,0 and Scaling long edge = 15,0



Reduction by 15% in both directions.

Number Of Copies

Define the number of default copies.

Default: 1

6.3.23.3.2 Duplex Settings

Mode

Define the standard print mode.

Available options are: Simplex, Duplex short edge binding, Duplex long edge binding

Back Side Offsets (Portrait)

Set offsets that only apply for the back of the page in portrait format.

You can define the following offsets:

- Long edge binding top
- Long edge binding left
- Short edge binding top
- Short edge binding left

The settings will be done for portrait format in units of 1/100 (0.01) inch.

Default: 0.00 (all)

Back Side Offsets (Landscape)

Set offsets that only apply for the back of the page in landscape format.

You can define the following offsets:

- Long edge binding top
- Long edge binding left
- Short edge binding top
- Short edge binding left

The settings will be done for landscape format in units of 1 inch. Minimum steps are 0.01 inches.

Default: 0.00 (all)

6.3.23.3.3 Page Layout

Margins

Set the margins for printing text.

Set margins apply after every reset command (<ESC>+X <CR>/<LF>), as long as no new margins are set by the data stream.

Available options are:

- Standard
- User Defined
- Portrait/Landscape

Default: Standard

- **Standard**

Set to apply the default margins of the Xerox XES 4213 printer.

- **User Defined**

Set the margins to a freely chosen value.

The values apply from the edge of the page and apply to both portrait and landscape formats.

User defined (inch)

- Left margin
- Right margin
- Top margin
- Bottom margin

Default: 0.000 (all)

- **Portrait/Landscape**

Define the margins for portrait and landscape.

Portrait (inch)

- Left margin
- Right margin
- Top margin

- Bottom margin

Default: 0.000 (all)

Landscape (inch)

- Left margin
- Right margin
- Top margin
- Bottom margin

Default: 0.000 (all)

6.3.23.3.4 PCL5 Printer Type

Use this dialog to set various properties and special abilities for the PCL printer you are using.

Pattern Mode

Select a pattern mode.

The Pattern mode setting allows the output of fill patterns on printer which, although they are HP/GL-compatible, are not PCL5-compatible.

Available options:

- PCL5 pattern (PCL5 fill patterns)
- HPGL pattern (HPGL fill patterns)

If you have selected PCL5 pattern and you wish to output the pattern in black, you must select HPGL pattern.

Default: HPGL pattern (HPGL fill patterns)

Fonthead Mode

Select the fonthead mode.

This mode allows printers, which although they can process a PCL5 font header are not PCL5 compatible, to use download fonts.

Available options:

- Resolution specified (PCL5e font header)
- Old style (PCL5 font header)

Default: Old style

Full Page Mode

Select full page (edge to edge printing) mode of one of the listed printer types.

Default: Off



If [Full page mode](#) is active, the PJL commands should also be enabled since the edge to edge print is generally set using a PJL command.



If [Full page mode for the Xerox 4517](#) is active, the landscape X-offset should be set to 32/100 inch.

PJL Commands

Select if PJL (HP print job language) commands will be used.

- **Not used**

Select to use PJL commands.

Default: selected

- **Will be used**

Select to use PJL commands.

Default: not selected

Use 'Collated Copy' is used to generate job based copies. It is only available when you select **Will be used**.

Default: disabled


If the PCL printer you are using supports the PJL collated copy command (@PJL SET QTY = n) (for example, HP LaserJet 5 Si Mopier or Xerox N24/N32), this option should be activated. Then the XES multiple copies collated command (<ESC>+Cn <CR>/<LF>) can be converted directly. Otherwise this XES command generates Page-based copies (PCL command: <ESC>&l#X).

Stitching

Enable or disable stitching.

If the PCL printer you are using supports stitching, this option can be configured here.

Default: disabled

 **Stitching will be activated by a specific PJL-Command. For that PJL Commands should be activated**

If enabled, the following options are available:

- At XES Job boundaries
- At XES Offset command

Default: At XES Job boundaries

- **At XES Job boundaries**

The XES job commands <ESC>+P or <ESC>+Q enable stitching, the reset command <ESC>+X disables it, that is all pages between these commands will be stitched.

- **At XES Offset command**

The XES offset command <ESC>O disables stitching, that is all pages between two <ESC>O commands will be stitched.

- **Mode**

Select a stitching mode from the pull-down menu.

Available options are:

- Dual
- Front

- Rear

Default: Dual

This option sets the position of the staples. With **Rotate** additional positions can be defined. The settings relate to a page in portrait format.

Dual: Select to insert two staples on the left hand side of the page.
Enable **Rotate** to make on the right hand side of the page.

Front: Select to insert one staple on the left hand side of the top of the page.
Enable **Rotate** to insert one staple on the right hand side of the foot of the page.

Rear: Select to insert one staple on the left hand side of the foot of the page.
Enable **Rotate** to insert one staple on the right hand side of the top of the page.

Alphanumeric ID Command supported

Enable for support of alphanumeric ID command.

If the PCL printer you are using supports the Alphanumeric ID Command `<ESC>&n#W`, this option should be enabled. This command will be used to access fonts which are saved as PCL macro file to a Hard/Flash disk of the printer.



This parameter usually will be used with parameter Page Buffer Flags - Save using HP PJI file system for swapping out resources to the printer.



If the printer produces lines of text ending with '_cspFM', this typically indicates that the printer does not support the Alphanumeric ID Command.

Default: disabled

Landscape X offset: <x> inches

Define a value for the Landscape X offset.

This command moves the entire output of a landscape page by the entered amount. This offset does not apply for a portrait page.

Default: 0

6.3.23.3.5 Tray Mapping

The two languages XES and PCL5 use different names for tray mapping. This allows individual adjustments to be made. Each XES tray number is converted into a corresponding PCL tray number.

Output Bins

- **Bin <n> map to <x>**

Assign a PCL5 bin name to each XES bin number by selecting an entry from the 'map to' list field.

Available bins to map to: Ignore (no mapping), Upper bin, Lower / rear bin, bin 1, ..., bin 9

Default: 0 (bin) map to Upper bin

Input Trays

- **Tray <n> map to <x>**

Assign a PCL5 tray name to each XES tray number by selecting an entry from the 'map to' list field.

Available trays to map to: Ignore (no mapping), Paper from main tray, Paper from manual input, Envelop from manual input, Paper from lower tray, Paper from optional paper source, Envelope from optional envelope feeder, Tray 7, ... Tray 25

Default: 0 (tray) map to <Ignore> (no mapping)

The two languages XES and PCL5 use different names for tray mapping. This parameter is available to compensate for these different names.

- **Input tray switching**

Enable for input tray switching.

Default: enabled

Paper Size To Input Tray Assignment

- **Tray <n> contains <x>**

Assign a paper size to a paper input tray.

Available options: Undefined format, DIN A4, DIN A3, Lexmark DIN A5, DIN A5, User defined format

When a paper size has been assigned, a PCL page size command (<ESC>&1#A) is also sent to select the tray.

If no paper size has been assigned ([Undefined format](#)), no PCL page size command is sent.

If [User defined format](#) is assigned, an arbitrary paper size can be set using the appropriate parameters.

Default: Tray 0 contains Undefined format


- **Short edge feed paper**


Enable to allow compatibility to various XES printers (for example, Xerox 3700 or Xerox 4235).

Default: disabled

If a paper input tray has been defined as a [short edge feed paper](#), a portrait font must be selected, for example, to print a landscape page, or a landscape font to print a portrait page.

 The four extra offset parameters for [Short edge feed](#) under [Common Page Settings](#) ([Visible window short edge offset](#), [Visible window long edge offset](#), [Vector short edge offset](#), [Vector long edge offset](#)) apply for a page printed in this way. The four offset parameters for [Long edge feed](#) apply for all others. This means that the outputs from different paper trays may be offset independently of each other.

 If [DIN A5](#) has been assigned, the page is printed in landscape (PCL page orientation command <ESC>&1#0 is used). In addition the vector offset [DIN A5](#) parameter apply, using which the page can be offset.

 [Short edge feed paper](#) should not be used in conjunction with [DIN A5](#) because the orientation has already been changed by the page orientation command.

Defaults**• Default Stacker**

Define the standard output bin.

The XES tray number is given; in other words this number is converted to the appropriate PCL number depending on the mapping.

Default: 0

• Default Cassette

Define the standard paper input tray.

The XES tray number is given; in other words this number is converted to the appropriate PCL number depending on the mapping.

Default: 1

6.3.23.3.6 Performance**Data Processing****• Max block size for reading**

Define the maximum size in kilobytes which the emulation shall attempt to read at the Input port by means of one read access.

If there are fewer data there, only so many bytes are read as actually exist. If there are already more data there, the maximum number of data that will be read corresponds to this value, and the remaining data are not read until the next read process.

 Frequent read access attempts with few data will reduce the system performance.

Default: 32 KB

• Block size for output buffering

Define the minimum number of kilobytes which must exist for output before a write access is attempted.

The data processed by the emulation are passed to the output port. To reduce the number of write access attempts at the output port, a value can be entered here to define the minimum number of kilobytes which must exist for output before a write access is attempted. To turn off buffering set the value to 0.

 Frequent write access attempts with few data will reduce the system performance.

Default: 32 KB

Timing**• Delay before next try on read error**

Define the delay before next try on read error.

If a problem occurs at the input port during reading the data, the emulation waits for the number of milliseconds set here before its next read attempt.

This prevents a slow input port slowing down the system by frequently repeated read

attempts.

Default: 1000 msec

- **Delay before next try on write error**

Define the delay before next try on write error.

If a problem occurs at the output port during writing the data, the emulation waits for the number of milliseconds set here before its next write attempt.


This prevents a slow output port slowing down the system by frequently repeated write attempts.

Default: 10 msec

- **Timeout for shutdown**

Define the time in milliseconds which the Print process shall wait for completion of the emulation thread.

If the set time has elapsed before the emulation thread has been terminated correctly by itself, it will be terminated by the Print process.

 Ensure that you set an adequate time to allow the emulation thread to terminate by itself. The set value should not generally be changed.

Default: 5000 msec

Bidirectional Communication

- **Produce ACK/NAK responses**

Enable to produce ACK/NAK responses.

When processing print jobs, which are received through a Channel Input Port and when processing IPDS print jobs, usually it is necessary to send feedback responses to the host system. In other situations these responses might not be applicable. For highest flexibility it can be controlled by this option whether feedback responses are sent to the host system or not.

Default: enabled


- **Send stop code to host after writing <n> sec**

Define the time in seconds after which a stop code is sent to the host.

This applies if a write access to the printer is not successfully completed within this time. Some hosts require responses within a specific time. If for some reason the printer goes offline or is turned off this might lead to problems with bi-directional communication.

Set to 0 to turn off this option.

Default: 0 (off)

 The effective time depends on the write timeout set at the output port. I.e. if at the output port a write timeout is set to 10 sec and here 15 sec are used, the second timeout after 20 sec will send the stop code. This is because the first timeout after 10 sec is still in valid range of 15 sec.

Jobs

- **Split job after <n> pages**

Enter number of page breaks after which a job shall be split into smaller parts.

For example, a print job with 5000 pages can be split into 50 smaller print jobs, each consisting of 100 pages. Also if no specific criterion exists for the end of a print job, and therefore all print jobs from the host system are processed in the Mercury system as a single job, a forced split can be achieved using this parameter.

This option is helpful if the printer is collecting print jobs completely before starting printing. If the job is split into smaller pieces, the time until the first page gets printed is reduced. But it must be sure that some queue optimization of the printer is disabled, if the partially printed job pieces have to be in the right order.

Default: 0 (no splitting)

- **Reconfigure printer after splitting**

Enable to re-initialize the output device at the start of a print job.

If this option is enabled, the emulation will conduct the initialization automatically each time a print job is split.

Default: disabled

6.3.23.3.7 Emulation

Xerox Printer

Select 'No specific' for no printer emulation.

Select 'Use Following' and the XES printer(-s) you wish to emulate.

The emulation may use the peculiarities of the following Xerox XES printer models:

2700
3700
4010
4030
4045
4046
4213
4235

- **Xerox 4030**

Enable to make the Merge Start Command '<ESC>ze' not terminate a constant page definition.

Default: disabled

- **Xerox 4045**

Enable to emulate Xerox 4045 XES peculiarities.

Text lines with several different font sizes are treated slightly differently. A font is considered even if it is only selected, but not used, within the line. Activation can therefore influence the line spacing between text lines containing different fonts.

Default: disabled

- **Xerox 4046**

Enable to keep an activated user defined table (<ESC>z1H) active even after a reset (<ESC>+X).

Default: disabled

Xerox 4213

Since the emulation also entries fonts with the same names in a list, the selection process can be controlled for fonts entered more than once with the same name. As default the last loaded font is used with the required name. If this parameter has been activated, the first loaded font with the required name in the list is used.

Xerox 4235

The Assign Font ID command '<ESC>+0FontName<LE>' assigns and selects the font 0 immediately. This is valid for ID 0 only!

Because the Xerox 4235 allows a larger page length than physically possible, the page length of a Set All Margin command '<ESC>m...<LE>' will not be restricted to the physically possible size.

Constant pages '<ESC>+M<LE>' are only printed if the page contains printable data.

Forms '<ESC>+N<LE>' are always printed when Merging '<ESC>zbn' is allowed, regardless whether the page contains printable data or not.

Line ending

Defines the properties at the line end. In many applications, for example, only CR's are used as the line ending code. In this case Auto LF (automatic line feed) must be selected. A similar thing applies for Auto CR (automatic carriage return).

Data encoding

The following settings are possible:

ISO 6937 Encoding
ASCII (Default)
IBM PC code page 437
IBM PC code page 850
ISO 8859/1 Latin 1 (ANSI)

Default Job

Defines the default properties of the emulation if there is no job command in the data stream.


<ESC>+P Only one orientation (portrait or landscape) is allowed on the page.
<ESC>+Q Both orientations are allowed on the page.

Default Font

Defines the default font and therefore the page orientation, if there is no font selection command in the data stream. The font number assignment is shown in the font status page (XES Command: <ESC>cspstat<CR><LF>). Alternatively the font name can be specified.

Font Rotation

If you have marked this option, all download fonts are rotated (portrait to landscape and vice versa). If the font name contains the character chain "-P" or "-L" (a non-binding indicator for portrait/landscape in the name), "P" is converted to "L" and vice versa.

 This parameter does not affect internal fonts.

Examples:

The XES portrait font Univers10iso-P is sent to the printer with activated font rotation by the emulation. In fact the font is converted to a landscape font and renamed to Univers10iso-L.

The XES portrait logo font MYLOGOP is sent to the printer with activated font rotation by the emulation. In fact the font is converted to a landscape font but keeps its name MYLOGOP.

Language

Forces the emulation to load a predefined, country-specific table as long as no XES select language command (<ESC>zlc) appears in the data stream. The following tables are available:

Character Set	Note
U.S. English	default
U.K. English	
French	
Dutch	
Spanish	
Italian	
Danish	
Norwegian	
Finnish/Swedish	
German	
Swedish	not supported
Belgian	not supported
French Canadian	
Portuguese	not supported
Latin American	not supported
User defined table	

 The Language parameter is only valid if Data Encoding is set to ASCII.

6.3.23.3.8 Resources

The settings on this page influence the treatment of resources (fonts, forms, etc.). The XES to PCL5 emulation is able to store frequently used resources locally or externally on the printer.

The following files may be optionally present in the resources directory (Path).

RESOURCE.XES may contain fonts and forms in XES syntax
 PREJOB.XES may contain arbitrary commands in XES syntax
 POSTJOB.XES may contain arbitrary commands in XES syntax
 1.XES ... 9.XES 1 ... 9 profiles, may contain arbitrary commands in XES syntax

Depending from the parameter Loading the file RESOURCE.XES will be read and the XES commands interpreted. Before or after processing a print job the files PREJOB.XES and POSTJOB.XES will be used. When a XES reset command encountered in the print data stream one of the profile will be used.

Resource Path

The directory is to be entered, which contains the resources that are to be sent to the printer.

 This parameter can not be changed while configuring a remote system.

Installation

The list contains resources files which have already been installed. By selecting Install..., new files may be added to the list.

 This parameter can not be changed while configuring a remote system.

Loading

Load only once

The resources file RESOURCE.XES are only converted and sent to the printer for the first job.

Reload every Job

The resources file RESOURCE.XES are converted and sent to the printer for every job that is received.

Profile

A profile may be selected from nine possible (the corresponding file 1.XES ... 9.XES is to be selected). If no profile has been selected, the values set in the configuration dialog are used as the default settings. If a profile has been selected, the XES commands in this profile file are also interpreted. A profile is executed immediately after each XES reset command (<ESC>+X).

Using a profile arbitrary, additional default settings, such as line spacing, form overlaying, etc. can be adjusted to suit individual needs.

 This parameter can not be changed while configuring a remote system.

Page Buffer Flags

Defines the properties of the emulation in terms of XES matrix graphics windows (for example XES command graphic window repeat <ESC>gr...) and the storage of resources.


Matrix graphic windows are stored in the RAM memory of the printer in the form of PCL macros, in other words they can be printed on the page several times but not modified (for example no X or Y zoom). With Page Buffer Flags you can modify the behavior of graphic windows and combine different types which are making sense.

Graphic window throughput

Matrix graphic windows are loaded into the RAM memory of the printer as PCL matrix graphics, in other words they can only be printed once on the page and not modified. (For PCL printers with a small memory)

Graphic window hold in memory

Matrix graphic windows are held in the RAM memory of the emulation, in other words they can be printed several times on the page and also modified. (<ESC>gr... is fully supported)

 Matrix graphic windows within a constant page (<ESC>+M) or a form (<ESC>+N) are always stored in the RAM memory of the printer as PCL macros (see Partial), regardless of the setting of Page buffer Flags.

Save fonts to local disk

Fonts will be saved automatically in Sixel format to the hard disk of the Mercury server using their name and the extension *.FNT. They will be resent to the printer if required. (Any font load command as <ESC>+F or <ESC>+A will be removed)

Save forms to local disk

Constant forms will be saved automatically to the hard disk of the Mercury server using their name and the extension *.FRM. They will be resent to the printer if required. (A form load command <ESC>+N and form end Command <ESC>+X, FF, etc. will be removed).

Save fonts to printer flash

Fonts will be saved in PCL bitmap font format to the optional external FLASH memory of the printer. The Emulation always saves its actual state to the local disk of the Mercury server in a file named FONTTAB.XES. This file contains font metrics which are used by the emulation. This reduces the amount of data transfer, because the fonts have to be sent only once to the printer.

Save fonts to printer disk

The meaning of this parameter equals to Save fonts to printer flash, but the font will be saved to the hard disk however.

Save to Lexmark Optra S

Lexmark Optra S. Defines the target printer for the external saving.



The parameter must be enabled to use PJJL commands.

HP PJJL file system commands (@PJJL FSDOWNLOAD ...) will be used for saving fonts to a hard/flash disk.

The target printer must support the @PJJL file system command (@PJJL FSDOWNLOAD ...) and the Alphanumeric ID command <ESC>&n#W. Additionally you have to enable the parameter and the parameter .

When saving resources externally PJJL commands will be used. These commands can occasionally separate a page into several jobs. To avoid this behavior you should respect the following procedure when saving resources externally:

1. Configure the emulation to save resources externally.
2. Send all resources to the Emulation.
3. Configure the emulation for normal operation.

With this procedure the resources will be saved externally and known by the printer. Now you can send your print jobs without resources.

Examples:

To save the fonts to an Optra S flash externally and to save the forms to the local hard disk of the Mercury Server automatically the following Page Buffer Flags have to be set:

Save forms to local disk
Save fonts to printer flash
Save to Lexmark Optra S

To save the fonts to a flash disk of a HP LaserJet 4000, the parameter Page Buffer Flags has to be set as follows:

Save using HP PJJL file system
Save fonts to printer flash

Additionally the checkbox located at dialog page PCL5 Printer Type must be checked.

Drive Number

Determines the partition number of the printer hard disk when parameter Page buffer Flags is set to 'Save fonts to printer disk' and 'Save using HP PjL file system'. In this case the fonts will be saved to the directory

[DRIVENUMBER]:\PCL\MACROS

where the currently set number will be used instead of [DRIVENUMBER].

6.3.23.3.9 Common Behavior

The settings on this page define the general behavior of the emulation.

Suppress Form Feed

Defines how the pages are to be separated at a form feed (0x0C).

Form Feed

The control code 0x0C is output. This may generate several empty pages if there are consecutive form feeds.

Printer Reset

A complete reset sequence is sent to the printer. This prevents empty pages.

Select Front Page Command

The PCL duplex command <ESC>&a1G is sent to the printer.

Automatic

The converter decides about page ejection. Empty pages and unnecessary reset sequences will be suppressed.


Automatic (no reset)

Like Automatic. Further XES job commands (e.g. <ESC>+X) generate no PCL job commands (e.g. <ESC>E or <ESC>%-12345X).

This method can increase printing speed of some PCL printers with optional output trays (finisher).

Shift Logical Page

If this option has been activated, the emulation automatically moves the visible window, if a position has been occupied outside the window limits. A manual adjustment of the parameters and is not necessary in this case.

 Some printer types from various manufacturers may create an undesirable print image if this parameter has been activated.

Fonts**Compress PCL Bitmap Fonts**

The PCL bitmap-fonts are sent to the printer in a compressed format. This may be useful if the font resources are to be stored externally on the printer and the memory there is limited (for example with a 2MByte flash DIMM).

Print Font List with Example

A specimen text is printed in the appropriate font on the font status page (XES command:

<ESC>cspstat<CR><LF>).

Exclude Font ID's

This allows individual definition of intervals of PCL font ID's that should not be used by the emulation. With the default setting the emulation starts for PCL fonts by the assignment of the ID 0 and continues in ascending order to 32767. If for example 30 – 100 and 450 – 500 is defined here, the emulation no longer uses fonts with ID's 30 –100 and 450 – 500.

Macros

Exclude Macro ID's

This allows individual definition of PCL macro ID intervals that should not be used by the emulation. With the default setting the emulation starts for PCL macros by the assignment of the ID 0 and continues in ascending order to 32767.

Split job after <n> times XES reset command

This setting causes a split of an incoming XES print job after a XES reset command has been detected <n> times. For example, a large print job can be split into smaller print jobs. Also if no specific criterion exists for the end of a print job, and therefore all print jobs from the host system are processed in the \$ProductName\$ system as a single job, a forced split can be achieved using this parameter.

Reconfigure printer after splitting

If you use the option Split job after <n> times XES reset command, it may be necessary to re-initialize the output device at the start of a print job. If this option is active, the emulation will conduct the initialization automatically each time a print job is split.

6.3.24 X-Metacode to PCL

6.3.24.1 Overview

This chapter describes all the parameters of the Mercury emulation X-Metacode to PCL5. All The parameters are default values.

In the configuration dialog for this emulation, you will find the following property pages:

- [Start Command](#)
- [System Generation](#)
- [Miscellaneous](#)
- [Resources](#)
- [Messages](#)
- [Finisher Options](#)
- [Color](#)

6.3.24.2 Parameter

6.3.24.2.1 Start Command

START Command

This option allows setting of default parameters for a report. This is similar to the original START command of a Xerox 4050/4090 printer.

```
( START [[jde][, ,jdl] [, , [copies][, FORM=form]] ] )
```

JDE/JOB

Field for the job descriptor entry.

JDL

Field for the job descriptor library.

Copies

Defines the number of copies. If a value is given other than zero, this value overwrites any given value specified in a job descriptor entry or within a DJDE command.

Form

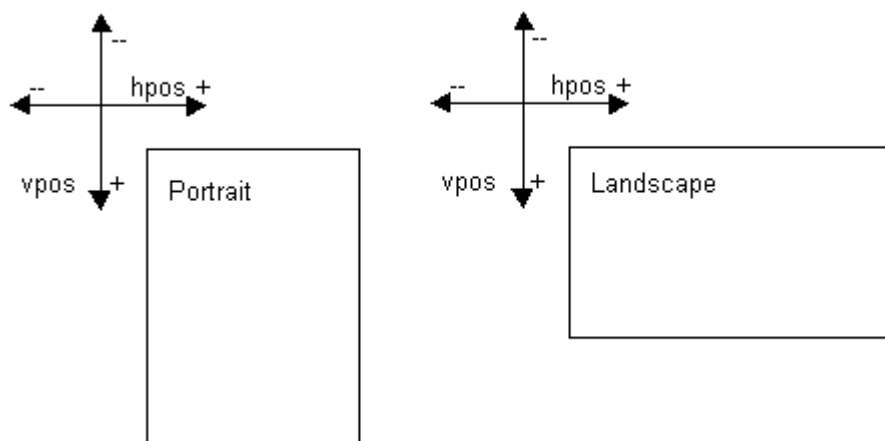
Specifies a FORM as overlay for printed pages.

Process Mode

The processing mode of the emulation. Currently only Online is supported.

Y, X Offset (in 1/300 inch)

This is a common vector offset ($Y = vpos$, $X = hpos$) to shift the complete content of a page. The offset may be given for portrait and landscape pages independently with signed values according to the following scheme:

**Destination Printer Feature**

Enables or disables special features which the destination PCL printer may utilize.

Use Collating Multiple Copies (@PJL SET QTY=n)

Activating this option produces copies on a job basis i.e. the whole print job can be copied.

⚠ If the printer does not support this feature, copies are made on page base, i.e. each page will be copied.

Emulation generates copies

Copies will be produced by the emulation.

⚠ This results into much bigger jobs (bad performance). Should be activated only if the order of the copies does not match compared to the origin system.

Use PCL jogging command

If activated, a PCL command (depends on given <Printer type>) will be used to jog (OFFSET) the page. If <No Specific> is set, the PCL Job Separation Command <ESC>&l1T will be used.

Printer type

Optimizes the data output for the given printer type. If <No Specific> is set only standard PCL and PCL commands will be used.

Edge to Edge

Activates the edge to edge printing feature of the selected printer model.

6.3.24.2.2 System Generation

JSL File Name

The name of the used Job Source Library file. The file must contain all job description entries and libraries (JDE, PDE, CME, ..., etc.), to which a report may reference to via DJDE.

⚠ In contrast to the origin Xerox 4050/4090, the JSL file needs no incremental line numbering. The lines are interpreted from top to down and left to right within the given column width.

JSL Column Width

Specifies the maximum column position for interpreting a line. If this value is set to 0, the line will be interpreted up to the terminating line end (<CR><LF>).

Treat constant strings without qualifier (...) as ASCII (A...)

Usually a string constant without qualifier is handled as EBCDIC, therefore

```
IDEN PREFIX='DJDE' is the same like IDEN PREFIX=E'DJDE'
```

If this option is activated, the system handles a string constant without qualifier as ASCII, therefore

```
IDEN PREFIX='DJDE' is the same like IDEN PREFIX=A'DJDE'
```

This is sometimes useful in LAN environments, if data has been already translated from EBCDIC to ASCII e.g. by an Codepage Filter.

Paper Size

Specifies the default paper size.

Use PCL paper size command (<ESC>&l#A)

If this option is activated a paper size command will be send to the printer, otherwise not.

Feed (input tray select assignments)

Assigns selection strings to any cluster or stock set name. To edit control bytes, the exclamation mark '!' is used for indicating a 2 digit hex value (e.g. !1B = 27 dec.). Two consecutive exclamation marks must be used to get the '!' itself (!! = !).


If a sequence for <Duplex alias> is given, this alternative tray will be used on duplex printing, otherwise only the <Selection string> will be used.

Example: The definition `MAIN = !1B&11H` results into the PCL paper source selection string `<ESC>&11H` when a `DJDE FEED=MAIN` is encountered.

Jobs


Pause print process on OTEXT=(...,WAIT)

When activated, the print process at the Mercury server will be suspended every time a `DJDE OTEXT=(...,WAIT)` command is encountered.

 The operator must observe the printprocess and manually set the processing state to continue.

Notify channel host on error condition

When activated, the channel host will be informed with a 'intervention required' (on channel input only) whenever a error message (usually red) occur during processing the job.
NOTE: Any data after a error condition will be discarded.

 The operator must abort the failing job at the host and restart the printprocess (STOP/START) on the Mercury server before any new job is released at the host.

Split jobs

Activates job splitting. There are multiple split criterias which may be combined.

Report boundary

Splits jobs on given report boundaries (e.g. BANNER or RSTACK criteria).

Page boundary

Splits jobs on every given page count.

ROFFSET detection

Splits jobs at page boundary whenever a ROFFSET criteria will be true.

Collated copy change

Splits jobs at page boundary whenever the copy count changes within collated mode.

Finishing change

Splits jobs at page boundary whenever the finishing device changes.

OTEXT=(...,WAIT)

Splits jobs at page boundary whenever a `DJDE OTEXT=(...,WAIT)` command is encountered.

Trigger string

Splits jobs at page boundary whenever the given trigger string matches within the job data.

Include resources in each split

When activated, all necessary resources are included in each part of the splitted jobs. It is not recommended to deactivate this option.

Split port

Output

The jobs will be splitted at the output port, i.e. the printer receives multiple smaller jobs, but on the Mercury server only one job will be shown.

Input

The jobs will be splitted at the input port, i.e. the printer receives multiple smaller jobs, also on the Mercury server multiple jobs will be created and shown.

6.3.24.2.3 Miscellaneous

Job name

Modify

The system generated job name may be modified.

<NONE>

Nothing will be appended.

Add page count

The amount of printed pages or sheets will be appended to the jobname (*_<pagecount>).

Add sheet count

The amount of printed pages or sheets will be appended to the jobname (*_<sheetcount>).

Replace system job name with HRPTNA field of RSTACK or BANNER

When a HRPTNA statement within the current JDL/JDE is given, this extracted field will be used as job name.

Replace system job name with extracted trigger data (offset, length)

The job name will be replaced with a sequence triggered from the job data, e.g. from a banner page. The extracted sequence will be **length** bytes from the **offset** relative to the matching trigger end.

Accounting

If activated, a ASCII file containing job information (e.g. number of printed pages, etc.) will be generated within the current resource directory. For each job a line with comma separated fields will be generated.

For advanced users only

Here may be done some special settings which are usually not necessary or might be harmful for normal operation.

Suppress macro control commands

During converting of the data stream no PCL macro commands will be used (<ESC>&f#X). It is not recommended to activate this.

Remove image macro after usage

Raster images are usually sent as permanent macro to the printer and called on usage. If this option is activated the macro will be deleted immediately after usage.

Print FORMS before page data

Usually the variable data is printed first and then the form will be invoked (overlay). If this option is activated the form will be invoked before the variable data (underlay).

Embed form name via Alphanumeric ID Command

When active, the DJDE form name will be embedded into the generated PCL macro via the PCL Alphanumeric ID Command '<ESC>&n#W' using the operation code 'z', e.g. for the form 'TEST.FRM', a '<ESC>&n5WzTEST' would be embedded. This should be usually ignored from a PCL interpreter.

For oversized paper only

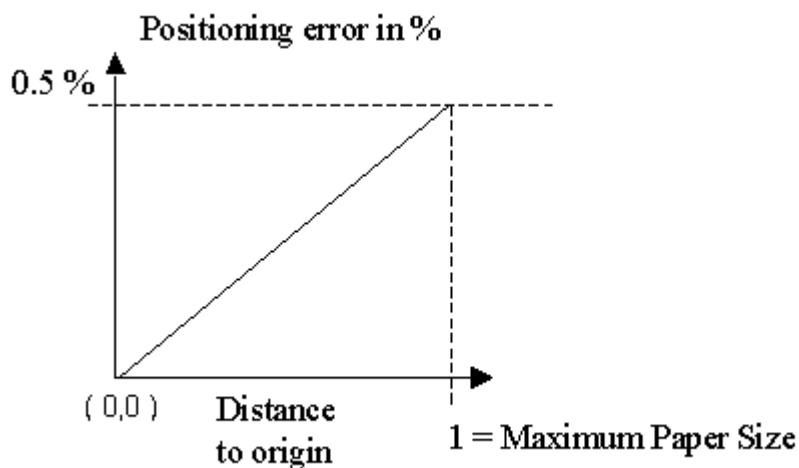
These settings are only valid for jobs using a paper size where the dimension $X > Y$
e.g. PAPER SIZE=(X,Y) = (17,14)

IMAGE Y,X Offset

Each raster image will be shifted with the given offset relative to the real position.

TEXT BEGIN vpos error correction

On large paper sizes, e.g. 17 x 14 Inch, a error correction may be necessary. The inserted value must be the expected error at the maximum paper size in 1/10 percent, e.g. 5 for 0.5 %. It is a linear error behavior in dependence to the distance assumed.

**6.3.24.2.4 Resources**

These settings affect the treatment of resources like fonts, images, forms, etc..

Resource Path

Specifies the directory, which contains the resources. Also dynamically loaded resources are saved in this directory.

Resource Script Filename

Specifies the name of the resource script files. These files contains information about resources and must reside in the given resource path.

Global

Contains the standard resource entries.

Target

Contains the dynamically via data stream loaded resource entries.

Run emulation in static mode

This controls whether the emulation is unloaded between the incoming print jobs (dynamic mode) or not. If not (static mode), the emulation remains loaded for the entire life time of the print processes.

- ⚠ This affects the behavior of downloading resources, since resource script files are read and interpreted at loading time (initialization) of the emulation.

6.3.24.2.5 Messages

The emulation uses 4 types of messages (Errors, Warnings, Messages, Trace Messages) which correspond to the log messages (Errors, Warnings, Info, Debug Info) of the [Mercury Client](#). Please see also chapter Log Level.

- ⚠ Due to the fact that the emulation uses the log mechanism of the [Mercury Client](#), the corresponding log message must be enabled in the Log Level Tab of the print process, in order to enable a message type of the emulation.

In addition, every message type has a verbose level and a category to allow more flexibility on displaying messages. The level uses a range from 0 to 99, where 0 has the highest priority and 99 the lowest. This allows to turn on/off several types of messages. Important messages usually have a verbose level of 0.

A specific message type can be switched off completely by setting the verbose level to -1.

- ⚠ It is not recommended to switch off the error and warning messages.

Verbose Level

Sets the current verbose level of the message type.

Warning

Show only once

When activated, no multiple warnings with the same text are given out.

Category

Sets the current allowed category of the message type.

- ⚠ A selected (highlighted) category is activated, a deselected is deactivated.

A message will be displayed if the verbose level of the message is less or equal to the currently set verbose level and its category corresponds to the currently set category, that means:

```
(Verbose Level of message) <= (current set verbose level)
AND
(category of message) == (current set category)
```

6.3.24.2.6 Finisher Options

These options are printer dependent.

- ⚠ Choose always the proper 'Printer type' (if known) within the tab 'Start Command'. If 'Printer Type' is set to 'No specific', a common HP PjL command will be used to activate the given option (e.g. for stapling @PjL SET FINISH=STAPLE, in this case the staple options TOP, BOTTOM or BOTH are ignored).

Staple position

Determines the staple options. Stapling will be activated by the option OUTPUT STAPLE=YES

Bookletmaker

Determines the bookletmaker options. The bookletmaker will be activated via the option OUTPUT SF1FUNCTION=YES or DJDE SF1FUNCTION=YES

Punch Position

Determines the puncher options. The puncher will be activated via the option OUTPUT SF2FUNCTION=YES or DJDE SF2FUNCTION=YES

6.3.24.2.7 Color**Output****Black/White**

The emulation ignores all Highlight color information. The output contains no PCL color command.

Color

The Highlight color information will be converted into corresponding PCL color commands using the PCL Imaging Mode (<ESC>*v#W).

Simple color

Same as *Color* but using the PCL Simple Color Mode (<ESC>*r#U), i.e no user defined palettes are allowed to adjust the *Text color* or *Image color*.

ISL Filename (Ink Source Language)

The optional name of the used Ink Source Language file. The file must contain all used ICATALOG entries.

⚠ The standard ICATALOG definitions 'DFAULT', 'XEROX' and 'XEROX1' are known within the emulation. As long as there are no other or different definitions, no ISL file (<NONE>) will be necessary.

ISL column width

Specifies the maximum column position of interpreting the line. If value = 0 is given, the line will be interpreted up to the terminating line end (<CR><LF>).

IRESULT (default)

Determines the real IRESULT value (BLACK or COLOR), if a DJDE IRESULT=DEFAULT is used.

⚠ The IRESULT option will be honored only within 2 color images, because in this case a distinction on pixel boundary can be done. Otherwise the color of overlapping objects depends on the order they are printed.

Text color

Allows to adjust the RGB values of a primary color. This adjustment is valid for objects like text, horizontal and vertical lines and logos.

Image color

Allows to adjust the RGB values of a primary color. This adjustment is only valid for image objects.

6.4 Filters

6.4.1 Overview

Overview

You can define various filters for input and output to compensate for incompatibilities between the host system and the destination print system in the character sets and for non-supported commands or character string in the print data before and after conversion by an emulation.

The following filters are available:

- Barcode Filter
- Codepage Filter
- Conditional Stringchange Filter
- CPCA Remover Filter
- Field Scanner Filter
- Job Copier Filter
- Job Separator Filter
- Pre-/Post-Job Filter
- Recordchange Filter
- Regular Expressions Filter
- Stringchange Filter
- Textextractor Filter

6.4.2 Barcode Filter

6.4.2.1 Application

The [Mercury Barcode Filter](#) enables PCL printers to print a wide variety of the most popular industry standard bar codes, including 2D-barcodes like PDF 417 and Datamatrix . The [Mercury Barcode Filter](#) primarily generates bar codes algorithmically in PCL 5 emulation. [Mercury Barcode Filter](#) includes support for:

- OCR-B as a scalable font
- independently scalable bar heights and widths
- the capability to embed text with selectable positioning
- automatic bar code data compression, when required
- automatic bar code checksum calculation and character placement, when required
- automatic placement of special characters (prefix, start/stop, guard bars, and so on), when required
- in PCL 5 emulation is the functional equivalent of Hewlett Packard's Bar Codes & More™ font set and is a fully compatible alternative to an HP LaserJet with JetCAPS BarSIMM/DIMM (v1.8e), except for the FREESCAPE alternate escape code feature.
- converting nearly every text in your print job to a barcode

With the [Mercury Barcode Filter](#), you have two possibilities for generating a barcode definition, which will be printed as a valid barcode:

[Barcode SIMM Emulation](#)
[Barcode and Field Scanner](#)

6.4.2.2 Barcode SIMM Emulation

Generating a barcode by inserting a font-like PCL-sequence in your document

Font selection commands in the incoming PCL5 data stream are analyzed for typeface characteristic values given in the 'Esc(s#T' command table. If the typeface value is one of those given, special bar code generation processes the font selection command. If the typeface value is not one of those given, PCL emulation proceeds as normal. The following commands enable you to further define characteristics of the bar code.

Command structure considerations

The incoming data is analyzed for valid parameters such as string length and string characters. These command structure considerations should be followed when implementing bar code applications using the [Mercury Barcode Filter](#):

The typeface parameter must always be specified, since it directs processing to the bar code generation.

Characteristic values, other than the default value, must be explicitly specified. If no value is specified, the default value is used. Previously specified non-default values are not retained. To print additional bar code data with the same characteristics as the previous data, set a new cursor position and send just the bar code data. It is not necessary to re-send the escape sequence portion of the command.

When other characteristics are specified, it must be in combination (chained) with the typeface characteristic. The order in which the characteristics are combined does not matter.

Characteristics not combined (chained) with a bar code typeface value are processed as standard PCL emulation font {text} selection sequences.

Bar codes may be selected using either the primary font selection command, 'Esc(s#T', or the secondary font selection command, 'Esc)s#T'.

Available typefaces

Typeface:	Barcode Symbology:
24600	UPC_A
24610	UPC_E
24620	EAN_8
24630	EAN_13
24640	INTERLEAVED_2_5
24641	INTERLEAVED_2_5_CD
24650	INDUSTRIELL_2_5
24651	INDUSTRIELL_2_5_CD,
24661	MATRIX_2_5_CD
24670	CODE_39
24671	CODE_39_CD
24690	CODE_93
24691	CODE_93E
24700	CODE_128_AUTO
24702	CODE_128_B
24704	CODE_128_C
24720	EAN_128
24750	CODABAR
24751	CODABAR_MOD16_CD
24760	MSI
24761	MSI_MOD10_CD
24762	MSI_MOD10_MOD10_CD
24763	MSI_MOD11_MOD10_CD
24770	POSTNET_5
24771	POSTNET_9
24772	POSTNET_11
24850	PDF417_2D
24820	DATAMATRIX_2D

Cursor positioning

Cursor positioning is set as follows when printing with the [Mercury Barcode Filter](#):

Command	Characteristic
Esc(s#P	Determines location of human-readable text printed with the bar code.
Esc(s#H	Selects the font used for human-readable text.
Esc(s#V	Selects bar heights.
Esc(s#B	Selects bar widths.
Esc(s#S	Selects space widths.

- From the current cursor position, bar code printing begins at the bottom left corner of the left bar.
- The cursor is located at the bottom right corner of the right bar when printing is complete.
- This cursor positioning applies both with and without accompanying human-readable text.

Human-readable text

Human-readable text can be specified with the bar codes.

- Six different typefaces are available for the text. See the 'Esc(s#H' command in 'PCL Emulation Escape Sequences'.
- The specified text is automatically centered on the bar code.
- Text is automatically scaled based on the width and height of the bar code and on the method of embedding used.

Transparent Print Data command

The Transparent Print Data command 'Esc&p#X[data]' provides printing access to characters with code values < 32 decimal. These characters are normally considered unprintable. Use the Transparent Print Data command to print bar codes which contain characters < 32 decimal.

Examples:

Esc(s4p2h36v24701T	no data specified with bar code selection command.
Esc&p8X20•08•98	desired data, "20•08•98" (8 bytes), specified with Transparent Print Data command immediately following the bar code selection command, where '•' is the character associated with decimal code 7 in the PC-8 symbol set.

Compressed data formats

Some bar code symbologies, such as UPC-E and Code 128, use compressed data formats. The [Mercury Barcode Filter](#) accepts data for these symbologies in either compressed or uncompressed format. The incoming data is analyzed to determine if it is compressed or not, and if required, compression is automatically performed prior to printing the bar code.

Checksum calculation

Some bar code symbologies define checksum characters as a requirement of the data format. When this is the case, the [Mercury Barcode Filter](#) performs the checksum calculation and automatically places the character(s) per the symbology specifications. Fixed length data formats can be sent with or without checksum characters.

Returning to normal text printing

To ensure a proper return from bar code printing to normal text printing, PCL escape

sequences that generate bar codes must be immediately followed by an escape sequence that fully specifies a normal text font.

Example:

Esc(s1p10v0s0b61444T returns the text following a bar code to 10-point Helvetica with proportional spacing, normal style and medium stroke weight.

PCL Emulation Escape Sequences

Command Function/Parameters

Command Esc(s#P or Esc)s#P

Determines the location of human-readable text printed with the bar code.

0	Use default value
1	Do not print text
2	Print text embedded
3	Print text half embedded
4	Print text under bar code
5	Print text above bar code

Add the numbers listed below to produce the corresponding effect:

10	prints UPC-A checksum at middle left of the bar code instead of the bottom left
10	prints start/stop * characters with Code 3 of 9 text
20	formats German and French postal bar codes
100	prints the checksum character with text

For the UPC-E symbologies, checksum characters are always printed if human-readable text is specified. The checksum characters are printed half-height if 100 is added to the p parameter, and are printed the same height as the text if 100 is not added to the p parameter.

2-D BARCODES:
PDF 417:

0 to 8	defines the ECC (error correction) level
1000	automatic sizes based calculation of the ECC level

Command Esc(s#1,#2,#3,#4B

Selects bar widths in dots (1/720 inch).

Some bar codes, such as Interleaved 2 of 5 and Code 3 of 9, require only two values for this command.

Others, such as UPC, EAN and Code 128, require all four values.

#1,	first (narrowest) width
#2,	second width
#3,	third width
#4,	fourth width

2-D BARCODES:
PDF-417: Symbol format parameters

#1,	Maximum number of rows for the PDF symbol
#2,	Maximum number of columns for the PDF symbol
#3,	=1 #1 and #2 are the mandatory size for the PDF symbol =0 or not specified: #1 and #2 are maximum size (default)
#4,	=1 The PDF-417 symbol is truncated on its right side =0 The PDF-417 symbol is not truncated (default)

Example: 8 columns, 10 lines mandatory size, non truncated: 8,10,0b

Data Matrix: Symbol format parameters

#1, Small module height in dots (1/720 inch)

Command Esc(s#1,#2,#3,#4S

Selects space widths in dots (1/720 inch).

Some bar codes, such as Interleaved 2 of 5 and Code 3 of 9, require only two values for this command.

Others, such as UPC, EAN and Code 128, require all four values.

#1, first (narrowest) width
 #2, second width
 #3, third width
 #4, fourth width

2-D BARCODES:

PDF-417: Symbol format parameters

#1, Symbol black/white module height compared to the width (1 to 10).
 Unit=minimum module width (default=3)
 #2, X parameter for the symbol X/Y size ratio (default: 2 for 2/3)
 #3, Y parameter for the symbol X/Y size ratio (default: 3 for 2/3)
 #4, Symbol module width (1 to 100). Unit=1/100 inch (default value=10).

Example: square PDF, minimum module width, half module size : 1,1,1,5s

Example:

Esc(s13p24670T selects Code 3 of 9 with human-readable text half embedded and start/stop * characters. Because the v, b, s and h parameters are not specified, their values default to:

- **v** .4 inch
- **b1, s1** .01 inch
- **b2, s2** .03 inch
- **h** automatically sized Courier bold

Command Esc(s#H or Esc)s#H

Selects the font used for human-readable text. The numeric value of # is of the form **CBA**, where:

A = typeface

0 Courier (default for all bar code symbologies)
 1 Letter Gothic
 2 Univers
 3 Univers Condensed
 4 CG-Times
 5 OCR-B

B = size

0 Automatic font size (default)

C = style

0 Default (bold)
 1 Regular
 2 Italic
 3 Bold
 4 Bold Italic

Example:

Esc(s305h24620T selects EAN/JAN-8 with automatically sized, OCR-B Bold text. Because the v, p, b and s parameters are not specified, their values default to:

- **v** .7 inch
- **p** print text half embedded
- **b1, s1** .013 inch
- **b2, s2** .027 inch
- **b3, s3** .04 inch
- **b4, s4** .053 inch

Additional configuration options:

Normally for this type of barcode generation no additional configuration options have to be set. All necessary parameters are coming with the print job data. If there are some configuration options, which cant be encoded in the font-like PCL- sequence, please open the properties dialog of the [Mercury Barcode Filter](#). In the *Barcode Definition Editor* dialog you have to insert a new definition with default values. Then you have to select "defined in print job" as *Barcode Type*. In the properties dialog of this definition you can choose the desired additional options.

You can only create one single definition of this kind in the *Barcode Definition Editor*.

6.4.2.3 Barcode and Field Scanner**Generating a barcode by retrieving data from print job**

Even easier as described in [barcode SIMM emulation](#), it is possible to print nearly every text and number in your document as a barcode without the necessity to implement special PCL sequences to define the code properties.

All things you need is the [Mercury Field Scanner Filter](#) to extract the desired text and the [Mercury Barcode Filter](#) to define the barcode properties. Both filters are configurable via a comfortable graphical user interface.

With the help of the **Barcode example** showed below it should be easy to set up the [Mercury Barcode Filter](#) filter for your own print jobs.

Setting up the [Mercury Field Scanner Filter](#):

- First you have to add the Field Scanner and Barcode Filter to your emulation. The barcodefilter has to be chained after the fieldscanner filter. Alternatively you can insert the Fieldscanner Filter as an Input Filter and the Barcode Filter as an Output Filter.
- Then you have to setup the Field Scanner Filter. In our example we use a *triggered PCL job* as input stream type. The recognized text data, which should be printed as a barcode is *reported to later components* (Barcode Filter).
- In the *PCL Scanner Configuration* you have to define a name for the barcode definition (*Field Name*). This name has to be redefined later in the *Barcode Definition Editor*, so that the Field Scanner data can be properly assigned to the correct barcode definition. If you define a additional field with the same name attached with the suffix "_stop", this field can be used to stop the barcode generation on follow up pages, if one of the continuation options in the barcode Filter is set. All Barcode definitions, whose name beginning with the same character sequence, as defined before the "_stop" suffix will be stopped as soon as the stop definition delivers a string value with more than 0 chars.
- Additionally in the *PCL Scanner Configuration* you have to define a *Trigger Sequence*, in our example the text "Kd-Nr.". This text is searched in the input data stream. When this trigger sequence is detected the next text data (after *offset* characters) with the length of *Length* characters is reported to the Barcode Filter. This text, in our example the number "106047" will be printed as a barcode.
- The record entity, defined in the *PCL Scanner Configuration* has to be 1 Page. That means, the barcode is printed on every page where the according trigger sequence is detected.

Additional you can use one of the continuation options at the [Mercury Barcode Filter](#), if the barcode should also printed on subsequent pages.

- For a detailed description how to set up the [Mercury Field Scanner Filter](#) please see the according section int the [Mercury](#) help file.

[Mercury Barcode Filter](#) configuration:

- After configuring the [Mercury Field Scanner Filter](#) you have to redefine the Field name in the Barcode *Definition Editor* as *Definition Name*. In this main dialog of the Barcode Filter configuration you have to select the *Barcode Type* and the *Code Source* via Drop down lists. In our example we choose to print the text as a 2D Datamatrix barcode. The code (in our case the number 106047) is captured directly from the source document, so we choose *Print Job* as *Code Source*. Alternatively you can define a static code inside the properties dialog of the barcode definition (*Code Source = Editor*).
- After defining *Definition Name*, *Barcode Type* and *Code Source* you should enter the definition properties Dialog via context menu or with button named *Definition properties*.
- Here you define the properties of barcode type specific parameters like **HRT (Human Readable Text)**, **bar and space width** and **bar height**. The options are the same as described above in Chapter 1) *Generating a barcode by inserting a font-like PCL-sequence in your document* . If you are not familiar with barcode specific things you can use the supplied default parameters, which are automatically generated for every barcode type. If the barcode requires a checksum it is calculated automatically.
- For the 2D Datamatrix code showed in the example below, the only barcode type specific parameter is the *small module height* which defines the width and height of the smallest square entity of this code. With 2D-barcodes no HRT is printed.
- In addition to the barcode type specific parameters there are some settings which have to be defined for all barcode types. In the *Position, rotation* dialog at least the **x-Position** and **y-Position** values have to be specified. With this parameters the printing location of the barcode on the page is fixed. The values are measured from the left top side of the sheet and define the location of lower left edge of the barcode (without HRT). For your convenience you can choose the desired measurement unit (inch, mm or decipoints). Please consider the page margins and the nonprintable area of the sheet. A barcode or HRT which sticks outside this borders is not printed correctly. Due to the barcode is printed as an overlay over the existing page content, you have to choose a position with enough "free" space which satisfies not only the size of the barcode, but also the necessary quiet zone around the code.
- In addition to the position, you can specify the orientation (**rotation**) and **color** of the barcode. The orientation can be chosen in 90° steps. The rotation is performed counterclockwise relative to the lower left corner of the barcode. If you want to print the barcode in another color than black, your printer has to support the PCL Imaging mode. Choose the desired barcode color in the *special settings* dialog.
- If you wish the barcode to be printed not only on the page, where the trigger sequence defined in the field scanner is detected, you can set up one of the **continuation options** defined in the **Job settings** at the *Special settings* configuration page. Here you can choose if the barcode should be printed on every page until the end of the job, a number of pages ore as long as a specific page number is reached. Additional you can specify a follow up barcode definition, which is executed after the continuation condition of the current barcode definition is no longer valid. With the option "Exclude first page", you can exclude the page on which the fieldscanner detected the barcode definition name field from the barcode execution. This can be useful if the code content is e.g. specified on a banner page. Also you can specify here, if the barcode should only be printed on the front or backside of duplex jobs.

If you need to generate more different barcodes in one ore more print jobs, you simply have to add unique trigger sequences and field definitions in the [Mercury Field Scanner Filter](#) and according barcode definitions in the [Mercury Barcode Filter](#).

Assigning a barcode definition to a specific kind of print job:

Normally all barcode definitions are executed for all jobs fed through the attached emulation. If you specify a field name containing "BCF_Jobtype" in the field Scanner filter, the contents of this field can be used to attach a barcode definition to a specific kind of printjob. The content of this field will be compared the content of the "BCF_Jobtype" field specified in the barcode definition list. Only if this content matches the one detected in the printjob, the barcode

definition will be executed. The default content of this field in the barcode definition list will be "all", meaning that the barcode definition will be executed for all printjobs, regardless of the content of the BCF_Jobtype field, if detected from the field Scanner filter. The barcode definitions shown in the barcode definition list will always grouped on the content of the BCF_Jobtype column.

You can define multiple fields in the field Scanner filter whose field names containing "BCF_Jobtype" e.g. "BCF_Jobtype1" or "BCF_Jobtype2". The contents of all those fields will be compared to the one "BCF_Jobtype" field shown for every barcode definition in the barcode definition list. Of course the BCF_Jobtype field should be detected at least on the same page or on a page before the first barcode field is detected. If this is not the case, only the barcode definitions with BCF_Jobtype = "all" will be printed.

You can assign one [Mercury Field Scanner Filter](#) field definition to multiple barcode filter definitions. To do this you can define multiple barcode filter definitions with the same definition name and different BCF_Jobtype content. Generating more than one barcode definition using the same definition name AND BCF_Jobtype content is not allowed.

The barcode filter definition will only be executed, if the actual BCF_Jobtype field scanner value matches the BCF_jobtype content of the barcode definition.

Displaying pure text without any barcode:

If you select **Text Only** as *Barcode Type* the text extracted from the field scanner is printed without generating any barcode.

If you additionally select the **Barcode Definition Editor** as the *Code Source* you can enter special predefined field names to insert job information values generated dynamically during printing. The following field names are available:

<DD.MM.YYYY> (date, german notation)

<MM-DD-YYYY> (date, US notation)

<HH:MM:SS> (time with seconds)

<HH:MM> (time without seconds)

<PageNr> (current page number)

<Variable> (content of the Variable defined in the field scanner, if "Variable" is specified as field type)

Example for text only output:

If you enter the text "Time: <HH:MM> Date: <DD.MM.YYYY>" in the Barcode Definition editor you will get for example the following printed output line:

"Time: 09:34 Date: 27.08.2010"

Overwriting some part of the printed output:

If you select **Text Only** as *Barcode Type* and disable the text output in the *Human readable text* barcode properties, the quiet zone *hor.* and *ver.* values specified in the *Position, rotation* barcode properties are interpreted as width and height of a rectangular area, which is filled with the *barcode colour* specified in the *Special settings*.

Barcode example:

Printing a customer number as a 2D barcode of the type Datamatrix :

Mechanics GmbH

Mechanics GmbH Nirgerdwostr. 66 D- 999 Woachimmer

Herrn/Frau/Firma
Tünnes Röstmän
Teststr. 6

58093 Testhausen

Rechnung / Kopie

Kd-Nr. 106047 Datum 02.05.05

Teile-Nummer	Bezeichnung	Menge	E-Preis	
A124 880 14 36	LEITBLENDE	1	100,49	100,49
				618,51
				16,84
				150,58
				21,89
				21,89
				3,42
				3,42
				8,94
				6,60
				1,06

Field Scanner Filter Configuration

Scan Method:

Data stream type is: **ja triggered PCL job** [Configure]

Data Storage:

Recognized data is: **reported to later components** [Configure]

PCL Scanner Configuration

#	Field Name	Trigger Sequence	Offset	Length	Trim	Find	Default
1	Kundennr	Kd-Nr.	3	6	<input type="checkbox"/>	First	

Barcode Definition Editor

Definition Name	Barcode Type	Code Source
Kundennr	Data Matrix	Print Job

Barcode definition: Kundennr Barcode type: 'Data Matrix'

2D Barcode

Position, rotation

Barcode position (lower left corner of the code):
(measured from the top left side of the sheet)

x-Position: 130 mm

y-Position: 110 mm

Quiet zone:
(minimum values)

hor. edge: 0.35278 mm

ver. edge: 0.35278 mm

Rotation (counterclockwise):
no Rotation relative to the lower left corner of the code

Measurement unit:
 inch
 mm
 1/720 inch

OK Cancel

Herrn
Tünne
Test:
58093 Testhausen

PCL Output:
Kd-Nr. 106047 Datum 02.05.05

Rechnung / Kopie

Teile-Nummer	Bezeichnung	Menge	E-Preis	
A124 880 14 36	LEITBLENDE	1	100,49	100,49
A124 880 40 70	STOSSFAENGER H	1	618,51	618,51
A124 880 00 97	REP.SATZ	1	16,84	16,84
A124 885 14 21	SCHIENE	1	150,58	150,58
A124 885 31 37	DAEMPFER	1	21,89	21,89
A124 885 34 37	DAEMPFER	1	21,89	21,89
A124 885 03 85	DICHTUNG	1	3,42	3,42

6.4.3 Codepage Filter


6.4.3.1 Application

Each operating system uses a certain standard character set to save text. The most common character sets are ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange), which is a 7-bit system, and EBCDIC (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchange **C**ode), which is an 8-bit system. Many operating systems use the 8th bit with ASCII code for language-specific characters (for example umlauts and special characters).

The Windows operating system under which *Mercury* works, uses the ASCII character set, extended by language-specific characters in the form of an 8-bit code table.

Normally the emulation of print data in *Mercury* is transparent in the system, in other words *Mercury* does not mind which code is used. In cases where the data received by *Mercury* are supplied by an EBCDIC host, however, and the output device to which *Mercury* supplies the data only understands ASCII code, or a *Mercury* emulation requires ASCII code, the data must be converted. This also applies the opposite way round and for different language-specific special characters.

Use the *Codepage filter* from *Mercury* to convert the character tables for the received data before the emulation and / or the outgoing data after the emulation.

 To use the Codepage filter, you must have installed the Windows [NT Language Pack](#), which is supplied on the installation CD for Windows NT.

6.4.3.2 NT Language Pack

On the installation CD for Windows NT, you will find the directory

```
\LANGPACK
```

which contains all the files required to install the Windows NT language pack. Log on with an account which has administrator rights and start the Windows NT Explorer to install these files. Click on the INF file that you wish to install with the right mouse button. Then click on 'Install' in the context menu.

To install all the variants, use the 'EXCHSRVR.INF' file and proceed as described above.

Another section is in the standard installation directory for Windows NT, for example \I386. Insert the die Windows NT CD as soon as you are asked to do so.

After the installation, the following character sets and codepages will be available:

EBCDIC character set:

Codepage and character set

- 037 IBM EBCDIC U.S./Canada
- 500 IBM EBCDIC International
- 870 IBM EBCDIC Latin-2 Multilingual/ROECE
- 875 IBM EBCDIC Greece
- 1026 IBM EBCDIC Latin-5 Turkey
- 20273 IBM EBCDIC Germany
- 20277 IBM EBCDIC Denmark/Norway
- 20278 IBM EBCDIC Finland/Sweden
- 20280 IBM EBCDIC Italy
- 20284 IBM EBCDIC Latin America/Spain
- 20285 IBM EBCDIC United Kingdom
- 20290 IBM EBCDIC Japanese Katakana Extended
- 20297 IBM EBCDIC France

- 20420 IBM EBCDIC Arabic
- 20423 IBM EBCDIC Greek
- 20833 IBM EBCDIC Korean Extended
- 20838 IBM EBCDIC Thai
- 20871 IBM EBCDIC Icelandic
- 20880 IBM EBCDIC Cyrillic
- 20905 IBM EBCDIC Turkish
- 21025 IBM EBCDIC Cyrillic (Serbian, Bulgarian)

Language-specific character sets

- Arabic Codepage and NLS API support
- Language support for Cyrillic
- For european languages with latin fonts
- Language support for Greek
- Hebrew Codepage and NLS API support
- Japanese Codepage, NLS API and font support
- Korean Codepage, NLS API and font support
- Chinese (Taiwan) Codepage, NLS API and font support
- Codepage and NLS API support for Thai
- Language support for Turkish
- Language support for English (US)
- Vietnamese Codepage and NLS API support

To select the Codepage conversion or to define different character conversions, configure the Codepage filter using the [configuration dialog](#).

6.4.3.3 Configuration

The conversion of the codepage requires the definition of the codepage used for the incoming data and the definition of the required codepage for the outgoing data.

Input Codepage

Use this to set the codepage for the incoming data (source data). The selection available depends on the codepages installed on the system. See the notes on the [NT Language Pack](#).

Output Codepage

Use this to set the required codepage for the destination data. Here, too, the available selection depends on the codepages installed on the system. See the notes on the [NT Language Pack](#).

Regardless of the codepages setting, the conversion of individual characters that differ from the standard can be defined manually using the so-called [Patch Table](#).

6.4.3.4 Patch Table

In the case that you require the conversion of an individual character which differs from the installed codepages, define the new code of the character here.

The table shows the current codes for all 255 characters. To change a special character, select the current code and enter a new value in the field **Hex Value**. After confirmation with the button **Apply**, the new code will be set.

After Replacing all necessary characters close this dialog by pressing **OK**.

Example:

You want to change ASCII code line feed (LF, decimal 10, hexadecimal 0x0A) to Carriage Return (CR, decimal 13, hexadecimal 0x0D).

Select in line 0x (from 0x0A) column xA (from 0x0A). In the field [Hex Value](#), the current code 0A will be displayed. Change this code to 0D and confirm pressing [Apply](#).

6.4.4 Conditional Stringchange Filter

6.4.4.1 Application

The Conditional Stringchange filter is, like the normal Stringchange filter, used to replace or remove individual characters and character strings in the data stream before and / or after the emulation of the print data. When a certain defined condition is matched, the Stringchange filter is triggered.

see Stringchange filter

Example:

- The input data stream contains special characters, such as umlauts (e.g. 'ä'). If a certain character string is found in the data stream, for instance 'English', a Stringchange is triggered to replace the umlauts with other character strings (e.g. 'ae').

6.4.4.2 Configuration

The Conditional Stringchange filter requires four pieces of information to replace or remove a character or a character string:

- Character or character string (Trigger) that causes one or more Stringchanges.
- Character or a character string to be replaced ('Search for')
- Character or a character string, which is to replace the source ('Replace with') or empty, if the source is to be removed.
- Consideration to low or high case letters.

If Data contains

Shows a list of all the defined strings that will trigger a Stringchange.

Add

Add and define one or more new triggers.

see [Definition of a trigger](#)

Edit

Edit the highlighted trigger(s).

see [Definition of a trigger](#)

Remove

Remove the highlighted trigger(s).

see [Definition of a trigger](#)

Ignore Case

Enable to ignore case of string.

This allows to define whether the replacement is to be completed taking case into account, in

other words whether 'abc' and 'Abc' or just 'abc' is to be replaced. The setting applies for all replacements.

Default: disabled

Search for

Shows a list of all the defined source character strings for replacements, in other words which characters or character strings must be replaced.

Replace with

Shows a list of all the defined destination character strings for replacements, in other words the characters or character strings that are to replace the source character strings. If the display is empty, no replacement will be made and the source character string will simply be removed.

Add

Add and define a new replacement.

see [Definition of a replacement](#).

Edit

Edit an already [defined replacement](#).

Remove

Delete a previously [defined replacement](#).



The number of triggers / replacements and the length of each trigger / replacement are limited as follows:

- A trigger / replacement character string is limited to 512 characters.
- The possible number of triggers / replacements is calculated using the following formula:

$$\text{Total of lengths of all sources} + \text{number of sources} + 1 \leq 32767$$

6.4.4.3 Definition of a trigger

Define a trigger.

Enter the following information:

If Data contains

Define character string(s) that will trigger a Stringchange.



Characters that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code. The '%' character itself is displayed as '%%'.

Examples:


- 'Hello' replaces the word 'Hello'

- 'Ä' replaces the German umlaut 'Ä'
- '%0D%0A' replaces the ASCII control characters Carriage Return (CR) and Line Feed (LF).

Add / Edit / Remove

Search for:

Define / Edit / Remove one or more trigger(s).

-  Unprintable characters or data bytes may be entered as hexadecimal value with preceding '%', i.e. %0D is interpreted as ASCII - CR

6.4.4.4 Definition of a replacement


Define a replacement.

Enter the following information:

Search for

Define character string which is to be replaced when trigger is found (see [configuration](#)).

If case is ignored depends on the setting of '[Ignore Case](#)'.

-  Characters that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code. The '%' character itself is displayed as '%%'.

Examples:

- 'Hello' replaces the word 'Hello'
- 'Ä' replaces the German umlaut 'Ä'
- '%0D%0A' replaces the ASCII control characters Carriage Return (CR) and Line Feed (LF).

Replace with

Define character or character string to replace the string defined in 'Search for'.

Leave empty to delete the searched for character string from the data stream. Replacements that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code.

Examples:

- Search for 'Hello World' Replace with 'Hey You' replaces 'Hello World' with 'Hey You'
- Search for 'Ä' Replace with 'Ae' replaces the umlaut 'Ä' with 'Ae'
- Search for '%0D%0A' Replace with '%0A' replaces CR/LF with LF
- Search for '%1BD' Replace with '' removes ESC-D

6.4.5 CPCA Remover Filter

6.4.5.1 Application

The **CPCA remover filter** is used to remove the CPCA protocol headers from a print job. Only the pure PDL data (e. g. PCL5) is kept.

CPCA means "*Common Peripheral Controlling Architecture*". CPCA jobs are created by Canon™ printer drivers.

CPCA Jobs start with the three hexadecimal bytes CD CA 10.

If the CPCA remover filter detects an error in the CPCA data stream, all following data will be skipped.

An error message "**CPCA-Filter:: Invalid CPCA header! Skipping all data.**" is written to the log file window.


6.4.5.2 Configuration

The CPCA remover filter does not need any configuration. Just add the filter to an emulation.

6.4.6 Field Scanner Filter

6.4.6.1 Application

The Field Scanner filter is used to scan print job data for data fields. This filter does not modify the original print job data. The filter scans the print job data using one of several scanning methods to detect data fields. Detected data values are passed to one of several storage methods or may be used to modify job parameters.

 Processing and usage of data field values depends on appropriate components, which do support usage of the Field Scanner filter.

6.4.6.2 Configuration

Before you can configure any other parameters, be sure to setup data scanning.

Data Scanning

The Field Scanner filter is configured by defining an appropriate scanning method and an appropriate storage method. The scanning method defines how data fields can be detected within the print job data and is therefore dependent of the print job data format (e.g. ASCII, PCL and others).

- [Triggered Binary Job](#)
- [Triggered PCL Job](#)
- [PCL Job Scanned For Text Areas](#)
- [CSV Formatted ASCII Data](#)
- [ASCII Formatted With Fix Column Width](#)

Data Processing

The storage method of detected data fields depends on what is to be done with that data. When passing to other components, this also requires components, which do support the selected storage method.

- [Reporting to later components](#)
- [Saving To ODBC Database](#)
- [Writing EASYWARE export files](#)

Job Data Manipulation

Each job passing Mercury will have some dynamically created administrative information. One of the most common is the jobs' name itself. Some of these parameters may be overwritten with data retrieved by the data scanner. These are:

<jobname> The name of the job displayed during processing and passed over to further processing and output devices.

<username>	The name of the user who created the print job.
<destaddress>	The destination address where the job is send to, if defined with the logical printer. This may be e.g. the e-mail address of the receiver when using an SMTP output port.
<destport>	The destination port or printer as delivered from input port or protocol. This is e.g. the name of the queue where the job has to be processed, when using an LPD input port.

Availability of these job information fields depends on the input port and protocol used. Not all protocols and ports do support all of this information.

Depending on the output port and protocol used, these parameters may be passed over to the next step in processing or to the output device. This mechanism e.g. enables display of a job name on the output device, which was previously detected by the scanner within the job data stream.

⚠ When using job data manipulation, [spooling](#) has to be turned on at output port. Also the option '[Start printing even if job is not completely spooled](#)' must not be enabled. Be careful which data is to be modified, especially don't modify the job's destination address if you don't get the result straight.

⚠ If you want to use job data manipulation only, without storing the retrieved data, select [Reporting to later components](#) as storage method at data processing setup.

The replacement for the field in the left column is entered as a string in the right column. This string can be simply a field name retrieved by the scanner with a preceding '\$' character in which case this will be a one by one substitution. E.g. to substitute the jobs' name with the field 'CustomerId' type '\$CustomerId'.

To use more than one field for generating substitution strings follow the rules for the interpreter. To leave a parameter as is, set the right column to empty.

Interpreter Rules

The built-in interpreter allows calculations with nested braces and expressions according to the common order of operators. The maximum nesting depth for braces is about 2000.

Variables and data fields must be preceded by a \$ character. Entering e.g. \$1 results in the output of e.g. Smith. Data fields and strings can be concatenated by addition. As a result of \$1 + ", " + \$2 the output will be e.g. Smith, Will.

Strings are delimited by " characters. Strings can be concatenated by addition. As a result of "Name" + ", " + "Prename" the output will be e.g. Name, Prename.

⚠ To use double quotes as a character within a string constant, type '\"'.

The following types of numbers are supported:

12.234	Floats and also integers
1.2e4 = 1.2*10^4	Exponential numbers
0x1B0A = 6922	Hexadecimal numbers (0x...)
0177 = 127	Octal numbers (0....)

Predefined [constants](#):

E	Exp(1)
G	Gravity constant
PI	3.1415...

The following operators are supported:

+ - * /	Simple calculations
%	Modulo, remainder
^	Power of
	OR (bitwise for numbers, logical for conditions)
&	AND (bitwise for numbers, logical for conditions)
~	Bitwise exclusive or

Examples:

4 7 = 7	7=0111b has Bit 2 (2^(2)=4) already set.
4 7.6 = 4 7	Equal by rounding
123 & 1 = 1	Odd number (Bit 0 is 1)
124 & 1 = 0	Even number (Bit 0 is 0)
124 ~ 1 = 125	Bit 0 (0) will be 1
125 ~ 1 = 124	Bit 0 (1) will be 0

The priority order of operators is defined as common. Here is the order from highest to lowest priority:

-	Signed numbers
^	Power of, right to left
* / %	Left to right
+ -	Left to right
&	Binary / logical AND, left to right
	Binary / logical OR, left to right
~	Binary exclusive OR, left to right



It is important that with logical operators each partial condition must be put in brackets. As an example a correct expression would be:

```
IF ( ($Gender="F") & ($Job="Waiter"); "Waitress"; $Profession )
```

Missing brackets as in

```
IF ( $Gender="F" & $Job="Waiter"; "Waitress"; $Profession )
```

will give a wrong result!

Calculation within expressions are done due to commonly known rules:

a+b+c	= (a+b)+c	Associative property for + - * /
a/b/c	= (a/b)/c	As above
a+b*c	= a+(b*c)	Multiplication before addition
a*b^c	= a*(b^c)	Power of has higher priority as '*'
a^b^c	= a^(b^c)	Associative property for power of
sin x*pi	= (sin(x)) * pi	
a^sqrt e	= a ^ (sqrt(e))	

Structural functions:

IF(condition;then;else)	Returns result depending on 'condition'. 'Condition' is an expression using '<', '>', '=' and '<>' for not equal, which may be true or false. The expression given by 'then' will be returned if 'condition' is true, else the expression given by 'else' will be returned.
LSCRIPT(scriptfile)	Returns the numeric result of a Lua script. Lua is an extension programming language designed to support general procedural programming. For a description of the lexis, the syntax, and the semantics of Lua you can visit www.lua.org . Inside a Lua script use the function <i>prsGetVariable(variable)</i>

to get a value of a variable which would also be available for the built-in interpreter. To set the value of such a variable you can use the function *prsSetVariable(variable, value)*. Both functions are an extension of the Lua programming language.

String functions:

VAL(string)	Converts string to numerical value
STRPOS(string; substring)	Returns the position of the first occurrence of substring within string, beginning at position 1. If substring is not a member of string, 0 is returned.
TEXT(value)	Converts numerical value to string
HEX2ASCII(string)	Converts a string of hex values to the appropriate ASCII characters
LENGTH(string)	Returns number of characters in string
LEFT(string [, #chars])	Returns #chars left characters of string. If #chars is missing, 1 is assumed
MID(string; pos; #chars)	Returns #chars of string beginning at position pos
RIGHT(string [, #chars])	Returns #chars right characters of string. If #chars is missing, 1 is assumed
LOWER(string [, #chars])	Changes first #chars of string to lower case. If #chars is missing the whole string is changed
UPPER(string [, #chars])	Changes first #chars of string to upper case. If #chars is missing the whole string is changed
CHR(code)	Provides the character given by the code number. The code number is valid from 0 to 255 and references a character of the ASCII code. Values higher than 255 are truncated to a byte's valid range.

Numeric functions:

The following mathematical standard functions are implemented and supported:

transcendental Fct.:	sin, cos, tan, cot, asin, acos, atan, acot
Hyperbolic Fct.:	sinh, cosh, tanh, coth, asinh, acosh, atanh, acoth
Logarithmic Fct.:	ln, lg, exp
Squareroot Fct.:	sqrt
Square Fct.:	sqr
Rounding Fct.:	round, floor, ceil, frac
Factorial:	fac
Modulus:	abs

Further implemented functions are:

LN a	Natural logarithm of a (base is e)
LG a	Logarithm of a (basis is 10)

There is no problem to calculate with + or - infinite. The expression '1/0' will result in 'INF' (infinite), also '-1/0 = -INF'. This allows e.g. calculation of arcus tangens of infinite:

tan (pi/2) = inf; not exactly, due to inaccuracies...
atan (inf) = pi/2

6.4.6.2.1 Triggered Raw Byte Stream

This scan method will search within any byte stream. To recognize data fields, trigger sequences, which will always precede the data values, need to be defined.

All field definitions are entered into the table. To add or remove lines, right click into the table and use the popup menu.

Field Name

Defines the name of a data field. This name needs to be unique and should not contain blanks.

Type

Defines the type of data used for this field. **Data** is used for job data recognized using a trigger sequence. **Variable** is used to set a field's value using a job information field. **Constant** is used to set a field's value always to the same constant value.

Trigger Sequence

This defines a sequence of bytes, which is always preceding the field value. This string is case sensitive. Characters, which may not be typed in using the keyboard, can be entered with a percent sign '%', followed by a two digit hex code, e.g. '%0d' for carriage return. To use the percent sign, type '%%'.

Offset

Defines the number of bytes following the trigger sequence, before the field value starts.

Length

Defines the number of bytes after the trigger sequence and offset to be used as field value. If **length** is set to 0, the trigger sequence itself will become field value, if present. If not, the field value will be empty.

Example: A field named 'Doctype' with a trigger sequence as 'DOC', an offset of 8 and a length of 4. The print job data contains a text as 'any text in DOC0123456789abc followed with text'. This will be recognized as data field 'Doctype' with value '89ab'.

Regular Expression

Defines the syntax for a regular expression to extract field values from retrieved data using a search pattern.

Default Value

Defines the default value to be used if the trigger sequence can not be found.

Find

This option defines, which appearance of the data is to be used as field value. If a unique trigger sequence can be defined, a data field is found only once. If this is not possible, here you may decide if the **first**, the **last** or **all** appearances will become data field value. In case of **all**, all detected values are concatenated to a long character string, each value separated from each other by a space character.

Trim Internal Blanks

This switch controls internal trimming of the field value. Regardless of the optional trimming of leading and/or trailing whitespace characters, this option will remove space and tab characters within the field value. As an example, a field value of ' V A L U E ' with this option enabled will result in a value of ' VALUE '. To remove the leading or trailing blanks, the options trim leading and/or trim trailing whitespace characters need to be used.

Variable

This list contains job information data as [job id](#), [job name](#), [user name](#) etc. All this information - whenever it becomes available - is dynamically set for each job. By selecting field type [Variable](#) and one of these job information fields, you may use job information as a field value.

Constant

This field is used to define a constant value. It might be necessary to set a field value always to a constant value. To do this, select [Constant](#) as field type and enter the value to be used here.

Premature whitespace terminates value

When e.g. the value is defined with a length of 8 bytes, there may be spaces within this length. With this option set, the space character will terminate the value even before the length of 8 is reached. The result will be '1234' instead of '1234 678' when a space is found between the digits 4 and 6. This is also done for other whitespace characters as Tab, Carriage Return, Line Feed and Form Feed.

Trim leading whitespace chars

If this option is set, whitespace characters are removed from the left end of the data field value string.

Trim trailing whitespace chars

If this option is set, whitespace characters are removed from the right end of the data field value string.

Record entity is defined by...

[Whole print job](#), this treats the complete print job as one data record.

[Field repetition](#), this treats data as a new data record, whenever one of the fields is recognized again.

[Trigger sequence](#), this requires a string entered in the field below. This string will be used as a criterion for the end of a data record. You may e.g. use the trigger sequence 'Page 1', to start a new record whenever a new document starts within the print job.



Handling of data records also depends of the used storage method. If the storage method does not support variable number of records, only one record per job will be available. Newly recognized field values automatically overwrite past values.

6.4.6.2.2 Triggered PCL Job

This scan method will search within the ASCII readable part of each page of the print job. To recognize data fields, trigger sequences, which will always precede the data values, need to be defined.

All field definitions are entered into the table. To add or remove lines, right click into the table and use the popup menu.

Field Name

Defines the name of a data field. This name needs to be unique and should not contain blanks.

Type

Defines the type of data used for this field. **Data** is used for job data recognized using a trigger sequence. **Variable** is used to set a field's value using a job information field. **Constant** is used to set a field's value always to the same constant value.

Trigger Sequence

This defines a sequence of bytes, which is always preceding the field value. This string is case sensitive. Characters, which may not be typed in using the keyboard, can be entered with a percent sign '%', followed by a two digit hex code, e.g. '%0d' for carriage return. To use the percent sign, type '%%'.

Offset

Defines the number of bytes following the trigger sequence, before the field value starts.

Length

Defines the number of bytes after the trigger sequence and offset to be used as field value. If **length** is set to 0, the trigger sequence itself will become field value, if present. If not, the field value will be empty.

Example: A field named 'Doctype' with a trigger sequence as 'DOC', an offset of 8 and a length of 4. The print job data contains a text as 'any text in DOC0123456789abc followed with text'. This will be recognized as data field 'Doctype' with value '89ab'.

Regular Expression

Defines the syntax for a regular expression to extract field values from retrieved data using a search pattern.

Default Value

Defines the default value to be used if the trigger sequence can not be found.

Find

This option defines, which appearance of the data is to be used as field value. If a unique trigger sequence can be defined, a data field is found only once. If this is not possible, here you may decide if the **first**, the **last** or **all** appearances will become data field value. In case of **all**, all detected values are concatenated to a long character string, each value separated from each other by a space character.

Trim Internal Blanks

This switch controls internal trimming of the field value. Regardless of the optional trimming of leading and/or trailing whitespace characters, this option will remove space and tab characters within the field value. As an example, a field value of ' V A L U E ' with this option enabled will result in a value of ' VALUE '. To remove the leading or trailing blanks, the options trim leading and/or trim trailing whitespace characters need to be used.

Variable

This list contains job information data as **job id**, **job name**, **user name** etc. All this information - whenever it becomes available - is dynamically set for each job. By selecting field type **Variable** and one of these job information fields, you may use job information as a field value.

Constant

This field is used to define a constant value. It might be necessary to set a field value always to a constant value. To do this, select [Constant](#) as field type and enter the value to be used here.

Premature whitespace terminates value

When e.g. the value is defined with a length of 8 bytes, there may be spaces within this length. With this option set, the space character will terminate the value even before the length of 8 is reached. The result will be '1234' instead of '1234 678' when a space is found between the digits 4 and 6. This is also done for other whitespace characters as Tab, Carriage Return, Line Feed and Form Feed.

Trim leading whitespace chars

If this option is set, whitespace characters are removed from the left end of the data field value string.

Trim trailing whitespace chars

If this option is set, whitespace characters are removed from the right end of the data field value string.

Record entity is defined by...

[One page](#), this treats each page as a new data record.

[One sheet](#), this treats each sheet as a new data record.

[Whole print job](#), this treats the complete print job as one data record.

[Field repetition](#), this treats data as a new data record, whenever one of the fields is recognized again.

[Trigger sequence](#), this requires a string entered in the field below. This string will be used as a criterion for the end of a data record. You may e.g. use the trigger sequence 'Page 1', to start a new record whenever a new document starts within the print job.



Handling of data records also depends of the used storage method. If the storage method does not support variable number of records, only one record per job will be available. Newly recognized field values automatically overwrite past values.

6.4.6.2.3 PCL Job Scanned For Text Areas

This scan method will search within each page of a PCL print job. To recognize data fields, rectangular areas containing the field values, need to be defined.

All field definitions are entered into the table. To add or remove lines, right click into the table and use the popup menu.

Job Template

Select a print job file to use it on the display for defining fields by rectangular areas. You may either enter the complete path of the job file into the edit field or click on the right button to browse for a file. After selection of the file, clicking on the [Open](#) button shows a field designer window with controls for drawing rectangles.

Field Name

Defines the name of a data field. This name needs to be unique and should not contain blanks.

Type

Defines the type of data used for this field. **Data** is used for job data recognized using a trigger sequence. **Variable** is used to set a field's value using a job information field. **Constant** is used to set a field's value always to the same constant value.

Text Position

Defines a rectangular area by position of the **left top** corner and its **width** and **height**. All values are entered as 1/100 millimeters. When using the field designer for defining field areas, these values are entered automatically.

Scope Of Scan

By default a field area is valid for every page. You may also define field, which appears only on a specific page as e.g. page 2. To select a specific page, enter the page number here.

Default Value

Defines the default value to be used if the defined area contains no text.

Regular Expression

Defines the syntax for a regular expression to extract field values from defined areas using a search pattern.

Trim Internal Blanks

This switch controls internal trimming of the field value. Regardless of the optional trimming of leading and/or trailing whitespace characters, this option will remove space and tab characters within the field value. As an example, a field value of ' V A L U E ' with this option enabled will result in a value of ' VALUE '. To remove the leading or trailing blanks, the options trim leading and/or trim trailing whitespace characters need to be used.

Variable

This list contains job information data as **job id**, **job name**, **user name** etc. All this information - whenever it becomes available - is dynamically set for each job. By selecting field type **Variable** and one of these job information fields, you may use job information as a field value.

Constant

This field is used to define a constant value. It might be necessary to set a field value always to a constant value. To do this, select **Constant** as field type and enter the value to be used here.

Premature whitespace terminates value

When e.g. the value is defined with a length of 8 bytes, there may be spaces within this length. With this option set, the space character will terminate the value even before the length of 8 is reached. The result will be '1234' instead of '1234 678' when a space is found between the digits 4 and 6. This is also done for other whitespace characters as Tab, Carriage Return, Line Feed and Form Feed.

Trim leading whitespace chars

If this option is set, whitespace characters are removed from the left end of the data field value string.

Trim trailing whitespace chars

If this option is set, whitespace characters are removed from the right end of the data field value string.

Record entity is defined by...

[Whole print job](#), this treats the complete print job as one data record.

[Number of pages](#), this treats a specified number of pages as one data record.



Handling of data records also depends of the used storage method. If the storage method does not support variable number of records, only one record per job will be available. Newly recognized field values automatically overwrite past values.

6.4.6.2.4 CSV Formatted ASCII Data

This scan method will read ASCII data formatted in a CSV compatible format. CSV format is database oriented and requires one line for each record. A line break means a new record begins. All data fields of one record are separated with a separator character and may be enclosed with a delimiter character.

To recognize data fields, the fields names either need to be defined or must be listed in the first data line.

The table is used to define the list of fields. This list needs to be set up in the same order as the fields appear within the job data. To add or remove lines, right click into the table and use the popup menu.

Field Name

Defines the name of a data field. This name needs to be unique and should not contain blanks. If a column is not needed, select '[--- Skip data ---](#)'.

Read field names from first line

If this option is set, it is assumed that all data field names are listed as first record in the first line of the data to be processed. In this case, you need not to enter any field name in the table. Only a place holder for each field needs to be defined, whether the field is skipped using '[--- Skip data ---](#)' or not as with '[From 1st line -](#)'.

Ignore first line

In case that data field names are listed in the first line but you want to use other field names defined by yourself, use this option. You may define each field name and the first line during processing is simply ignored.

Field separator

This defines the character used as separator for data fields within a record. Most common is the [semicolon](#). Also often used are [colon](#), [space](#) and [tab](#). You may also define your own character using [other](#) and typing the character in the field on right side.

Delimiter character

If the character used as [field separator](#) also may appear within the data fields, these need to be enclosed with a delimiter character. Commonly used are " and '.

Trim leading whitespace chars

If this option is set, whitespace characters are removed from the left end of the data field value string.

Trim trailing whitespace chars

If this option is set, whitespace characters are removed from the right end of the data field value string.

6.4.6.2.5 ASCII Formatted With Fix Column Width

This scan method will read ASCII data with a fix width for each data column. There are no separator or delimiter characters present or required. Each data field is defined by its column width, which must be the same over all records

To recognize data fields, the fields either need to be defined or must be listed in the first data line.

The table is used to define the list of fields. This list needs to be set up in the same order as the fields appear within the job data. To add or remove lines, right click into the table and use the popup menu.

Field Name

Defines the name of a data field. This name needs to be unique and should not contain blanks. If a column is not needed, select '[--- Skip data ---](#)'.

Chars

Defines the number of characters used for this field. This is the exact width in characters to be read as field value or to be skipped.

Read field names from first line

If this option is set, it is assumed that all data field names are listed as first record in the first line of the data to be processed. In this case, you need not to enter any field name in the table. Only a place holder for each field needs to be defined, whether the field is skipped using '[--- Skip data ---](#)' or not as with '[- From 1st line -](#)'.

Ignore first line

In case that data field names are listed in the first line but you want to use other field names defined by yourself, use this option. You may define each field name and the first line during processing is simply ignored.

Trim leading whitespace chars

If this option is set, whitespace characters are removed from the left end of the data field value string.

Trim trailing whitespace chars

If this option is set, whitespace characters are removed from the right end of the data field value string.

6.4.6.2.6 Reporting To Later Components

This storage method is notifying all components placed and working after the Field Scanner filter whenever a data field or a record change has been recognized. Notification is done at the moment, when the scanner method reports one of these events. All components, which do support Field Scanner notification, will receive the recognized data immediately.

Recognized data fields are not stored within this method. If a data field is recognized more than once, the newer value always overwrites the past.

Record entities are reported as configured within the scanner method. Although this supports multiple records, there will be always one record per job, if the components, which receive and handle notification, do not take care of this.

This storage method does not need any configuration.



Processing and usage of data field values depends on appropriate components, which do support usage of the Field Scanner filter.

6.4.6.2.7 Saving To ODBC Database

This storage method is writing recognized data into an ODBC data source.

Recognized data fields are buffered until a record entity is recognized as complete. If a data field is recognized more than once, before completion of a record entity, the newer value may overwrite the past. As soon as a record entity is complete, all data of this record is saved into the ODBC data source table as configured.

Configuration is done by defining

- [Data Source](#)
- [Data Table](#)
- [Data Fields](#)

Data Source

Name

Select the ODBC data source from the list of known and configured data sources.

User

Enter the user name to be used for connecting the ODBC data source, or leave empty if not needed.

Password

Enter the password to be used for connecting the ODBC data source, or leave empty if not needed.

Connect

Click on this button to establish a connection to the ODBC data source. A connection must be established to complete the configuration of the data table, which is used to receive recognized data.

ODBC Manager

This button opens an applet for setting up ODBC data sources. The applet is the same as

known from the system control. An ODBC data source needs to be defined and configured within this dialog, before it can be used for setting up this storage.

Data Table

Table creation

This controls whether a new table is created or an existing table is used to store the recognized data.

[Only use existing table](#), will not create a new tables. An existing table must be selected in the [table name](#) field below. If data fields are scanned that are not part of the selected table, this data fields will **not** be stored. To be stored the name defined within the scanner must match a field defined within the data table. If the table does not exist, the storage fails.

[Use existing or create new](#), will try to use an existing table with the name given in the [table name](#) field below. If this table has disappeared or does not exist, a new table will be created. Saving data fields into an existing table requires field names within the table matching the names at scanner definition as described above. For creation of a new table, the data type of the fields scanned for need to be defined in the [data fields](#) page.

[Drop existing and always create new](#), will drop any existing table with the name given in the [table name](#) field below. Then a new table is created according to the definitions done at scanner setup. The data type of the fields scanned for need to be defined in the [data fields](#) page.

Table name

If using an existing table, just select the required table name from this list. For creation of new tables, the table name may be either 'JobId_<jobid>' or 'JobName_<jobname>', where the variables '<jobid>' and '<jobname>' will be replaced by their actual value at processing time. Since these values may be completely numeric, which is with some databases not allowed as a table name, the prefixes are added.

Records of new jobs are added...

[To existing records](#), will just insert new records into the table and thus collects records of multiple print jobs cumulatively.

[After deleting existing records](#), will delete any existing record from the table, before adding new records.

Data Fields

Field Name

Defines the name of a data field. This name needs to be unique and should not contain blanks. Usually the names are set at scanner definition or detected automatically.

Field Type


This defines the type of the data stored in the database field. [Varchar](#) is the simplest one, where everything may be saved to. A [Varchar](#) field may have any length up to 255 characters. Other types as [Integer](#), [Double](#) or [Date](#) require the field values properly formatted. [Longtext](#) is for alphanumeric text with more than 255 characters. The maximum length of [Longtext](#) is database specific. When using any other type than [Varchar](#), make sure the data values have the right format, e.g. a field value of 'catch22' cannot be saved to a [Date](#) field.

Length

Defines the length of the database field.

6.4.6.2.8 Writing EASYWARE import files

This storage method is writing recognized data into [EASYWARE](#) import files.

 Usage of this storage method requires output of print jobs to file. Additionally the [default filename](#) needs to be configured to '<Drive>:\<Directory Path>\<jobid>.<Extention>' where <Drive>, <Directory Path> and <Extention> needs to be replaced with appropriate values as for example 'D:\Output\<jobid>.prn'. The filename **must** be set to <jobid>.

Recognized data fields are buffered until a record entity is recognized as complete. If a data field is recognized more than once, before completion of a record entity, the newer value always overwrites the past. As soon as a record entity is complete, all data of this record is written to an [EASYWARE](#) import file is will be created in the same directory as the output file. The filename used is the job id and therefore the same as the filename of the print job.

Configuration is done by defining

- [General Options](#)
- [Data Fields](#)

General

Folder

Enter the name of map title to be used for [EASYWARE](#) import.

Charset

Select the charset used in the print job data. You may select either [ASCII](#) or [ANSI](#).

Date Format

is used to re-format data fields marked as date. Any date recognized is changed in its format according to the settings made here.

Order

This defines the order of the dates parts. 'Y' stands for year, 'M' for month and 'D' for day. German date format e.g. are 'DMY' as english formats are using the 'MDY' order.

Separator

This defines the separator character used to divide the date parts. You may also leave this field empty if no separator is required. E.g. the order of YMD without separator as e.g. '20050301' is commonly used because this supports numeric sorting of dates.

Two-digit day and month

If checked, the day and the month is always formatted with 2 digits as e.g. '03' instead of '3'.

Four-digit year

If checked, the year is formatted with 4 digits as e.g. '2005' instead of '05'.

Data Fields

All field definitions are set in the table and will reference field definitions defined at the scanner configuration. To add or remove lines, right click into the table and use the popup menu.

Value

Select a data field name configured with the scanner method or [<Document file>](#) to refer the print job output.

Field Type

EASYWARE uses [text fields](#) and [document fields](#). To control usage of a recognized data field, select the appropriate field type here. Text fields are written to import file using the field identifier "FI", document fields are written to import file using the field identifier "BI".

Field ID

Each data field used for import also needs to be referenced by a unique ID. Select the appropriate value here. Field IDs for text fields are allowed in the range of 1001 up to 1999. Document fields may have IDs in the range of 2001 up to 2999.

Date

If checked, the field is marked as a date which has to be re-formatted, using the settings made in the date format group of the [general page](#).

As an example, a scanner is defined to recognize the fields 'Customer ID' and 'Sales Account'. The EASYWARE map title is set to 'CAPTURE'. Both recognized data fields are passed as text fields using IDs 1001 and 1002. The print job file itself is passed as a document field with ID 2001. Then, the created import file will look as follows:

```
@CHARSET,ANSI
@FOLDER,FT:^CAPTURE^,FI:1001,FI:1002,BI:2001
^50003930^,^821^,^D:\Output\42.pcl^
@FOLDER,FT:^CAPTURE^,FI:1001,FI:1002,BI:2001
^50003955^,^321^,^D:\Output\42.pcl^
```

6.4.7 Job Copier Filter

6.4.7.1 Application

The Job Copier filter is used to create multiple copies of processed print jobs.

Copies can be created parallelly or serially. Parallel means the job is copied parallelly to multiple destination directories. You may define a list of directories, which will receive a copy of the print job for further processing. This filter does not modify the print job's data.

Serial copies are created by appending the print job several times to itself. This will produced one job file containing several copies of the job in serial.

6.4.7.2 Configuration

Define a destination directory for each copy to be created.

Destination Directory

To add another copy of the print jobs, click on the [Add](#) button and select a directory. The path will be added to the list of job directories. To change the destination directory, click on the [Edit](#) button. To remove a copy, select the required job directory and click on the [Remove](#) button.

Appendices

If this count is greater than 0, it defines the number of copies of the print job which, will be added to the output file as serial copies. E.g. the value of 2 will produce a print job file, which contains the original print job and 2 copies appended. In summary the original print job will be 3 times there.

Also pass job information data

If sending print jobs to a Mercury input port, additional information about the print job as source, name etc. can be transferred. This information will be transferred before the print data. To use this feature, the option of the corresponding input port must be checked, too. This option must not be used, if the receiver is not a Mercury input port.

Add copy no. to file name

If checked, the number of the copy will be appended to the file name. If using the same job directory twice this option is forced, because the copies must have different file names.

6.4.8 Job Separator Filter

6.4.8.1 Application

The Job Separator filter is used to divide PCL print jobs to several partial jobs depending on criteria present within the print job data.

Examples:

- The input data stream contains several constant character sequences, which are used to classify each page. Using the Job Separator filter the original job may be divided into several jobs where each partial job contains only pages of one type.
- The input data stream contains one constant character sequence followed by a specified number of variable characters or digits as e.g. customer IDs. Using the Job Separator filter the original job may be divided into several jobs where each partial job contains only pages for one customer.

6.4.8.2 Job Separation

The Job Separator filter is using two different methods for dividing jobs. Each method requires a definition of one or more trigger sequences. Select the required method, then define the trigger sequences.

Method

[Scan for set of constant triggers](#)

This method is used to divide PCL print jobs by constant character sequences that are used to classify pages. Each trigger sequence is used as a criterion to be searched on each page. If found, the page is assigned to a partial job containing only pages of this type. If more than one trigger sequence is present, the first to be found is significant. If no trigger sequence is present, the page is classified as unassigned.

[Scan for one trigger followed by variable data](#)

This method is used to divide PCL print jobs by variable character or digit sequences present on each page. There is only one trigger sequence to be defined, which precedes the variable part. This trigger is part of the final criterion as also the specified number of characters or digits. E.g. a trigger sequences of 'DOC' with a significant data length of 3 automatically generates criteria like 'DOC123', 'DOC744' and others as found within the print job data.

Default: Scan for set of constant triggers

Significant size of data

This is used with method '[Scan for one trigger followed by variable data](#)' and defines the number of characters or digits to be used for the variable part of the criterion.

Trigger Sequence

Each trigger sequence is used as a criterion to be searched on each page. If found, the page is assigned to a partial job depending on the selected method. Characters, which may not be typed in using the keyboard, can be entered with a percent sign '%', following a two digit hex code, e.g. '%0d' for carriage return. To use the doubled percent sign, type '%%'.

To add lines for further trigger sequences, click on the header line of the grid and press the Insert key. To remove a line click into a column of the line to be removed, the click on the header line and press the Del key. With method '[Scan for one trigger followed by variable data](#)', only one trigger sequence is allowed.

Valid

Until next trigger

All pages passing by after the page with the trigger sequence found are classified the same way as the page containing the trigger until another trigger is found. This is also done for subsequent pages containing no trigger.

One page only

The next page after the page with the trigger sequence found is classified as unassigned, if there is no trigger sequence present. This is also done for subsequent pages containing no trigger.

Destination Directory

Defines a path, where the partial job, which will be created depending on the trigger sequence is to be saved.

Name Prefix

This is an optional prefix added to the partial job's file name. The complete file name is built as


<Prefix>.<Job ID>.<Trigger sequence>

<Prefix> is the optional string of up to 8 characters defined in this field. <Job ID> is the optionally added job ID of the incoming print job. <Trigger sequence> is either a constant trigger as defined or the combination of trigger sequence and variable part.

Premature whitespace character terminates variable part

This is only used with method '[Scan for one trigger followed by variable data](#)'. When e.g. the variable part is defined as a sequence of 8 characters, there may be spaces within this length. With this option set, the space character will terminate the variable part even before the length

of 8 is reached. With a trigger sequence as 'DOC', the criterion will be 'DOC1234' instead of 'DOC1234 678' when a space is found between the digits 4 and 6. This is also done for other whitespace characters as tab, carriage return, line feed and form feed.

 The number of triggers and their length are limited as follows:

- The length of a trigger sequence is limited to 512 characters.
- The possible number of trigger sequence is calculated using the following formula:

$$\text{Total of lengths of all trigger} + \text{number of trigger} + 1 \leq 32767$$

6.4.8.3 Banner Pages

Each partial job created by the Job Separator filter may be optionally enclosed with banner and/or trailer pages.

Add banner page

If this option is set, a banner page is added to each partial print job created by the Job Separator filter.

Add trailer page

If this option is set, a trailer page is added to each partial print job created by the Job Separator filter.

Paper tray

Here, the paper tray is defined, where the paper is drawn for the banner and trailer pages. The paper tray for banner pages may be set to a different tray as for trailer pages.

Orientation

Here, the orientation of the banner and trailer pages is defined. The orientation for banner pages may be set to different as for trailer pages.


User defined banner text

This is an optional field for adding user-specified text to the banner and trailer pages. Any text entered here will be printed to each banner and each trailer page.

Additionally, variables can be used within this field. When typing in the arbitrary text, variables are enclosed with '<' and '>' as '<myvariable>'. Predefined variables are:

<jobname>	Name of the job
<jobnamenoext>	Name of the job (if filename without file extension)
<srcaddress>	Address of senders as IP address or Novell queue name
<username>	Name of user
<destport>	Destination port as queue name from LPD protocol
<destaddress>	Specific device if defined in Logical Printer
<jobid>	Job-ID
<creationtime>	Start time of job
<starttime>	Emulation time of job

This field also supports user-defined variables acquired using the Field Scanner filter.

 The Field Scanner filter must be part of the license to be used.

6.4.8.4 Miscellaneous

This page contains some global options concerning the Job Separator filter.

Scan PJI data for trigger

If this option is set, the PJI data part of the print job is also scanned for trigger sequences.

Add job id to names of job files

When this option is set, the ID of the incoming print job is added to the names of the partial print jobs to be created. The complete file name is built as

<Prefix>.<Job ID>.<Trigger sequence>

<Prefix> is an optional string of up to 8 characters. <Job ID> is the job ID of the incoming print job. <Trigger sequence> is either a constant trigger as defined or the combination of trigger sequence and variable part.



If a second job is processed after another, this avoids overwriting created partial job files. As a result the name is different because of the different job id. If this option is not set, the names will be the same and newer created jobs will overwrite previously created jobs.

Add job information data to job files

If sending print jobs to a Mercury input port, additional information about the print job as source, name etc. can be transferred. This information will be transferred before the print data. To use this feature, the option of the corresponding input port must be checked, too. This option must not be used, if the receiver is not a Mercury input port.

Standard output receives...

This option controls what is sent to the standard output port of the print process:

[nothing](#)

There is no output at all. All pages, classified by trigger sequences, are divided into the partial print jobs; unassigned pages are dropped.

[unassigned pages only](#)

All pages, classified by trigger sequences, are divided into the partial print jobs; unassigned pages are sent to the standard output port.

[everything](#)

All data is transparently sent also to the standard output port.

Remove trigger sequence

This option controls what happens to the trigger sequences which were detected within the print job data:

[not at all](#)

The trigger sequences are untouched.

[exactly](#)

The trigger sequences are removed as found with their exact length of characters.

[and all until next escape](#)

The trigger sequences and all subsequent data until the next PCL escape sequence are removed.

[and all until end of line](#)

The trigger sequences and all subsequent data until the next line feed are removed.

[and all until end of page](#)

The trigger sequences and all subsequent data until the end of the page are removed.

6.4.9 Pre-/Post-Job Filter

6.4.9.1 Application

The Pre-/Post-Job filter is used to add the content of a file to the beginning or the end of a job.

6.4.9.2 Configuration

File to insert before job data:

The content of this file will be inserted before the job data.

File to add after job data:

The content of this file will be added after the job data.

Variables, that can be used:

<creationtime>	Start time of job
<srcaddress>	Address of senders as IP address or Novell queue name
<destaddress>	Specific device if defined in Logical Printer
<destport>	Destination port as queue name from LPD protocol
<username>	Name of user
<jobid>	Job-ID
<jobname>	Name of the job
<jobnamenoext>	Name of the job (if filename without file extension)
<starttime>	Emulation time of job

or any variable of the field scanner filter.

6.4.10 Recordchange Filter

6.4.10.1 Application

The [Recordchange filter](#) of the [Mercury Server](#) is used in order to change block- or record-oriented data streams from one format to another at the input and/or output of the [Mercury emulations](#).

The following table shows the current possible input and corresponding output formats of the RECORDCHANGE filter, which may be configured.

From Input Format :	To Output Format :
NPS internal Format	NPS internal Format S/370 Byte or Word Length Record Format (Barr)
S/370 Byte or Word Length Record Format (Barr)	NPS internal Format S/370 Byte or Word Length Record Format (Barr)
Line printer Format	NPS internal Format
Channel Input Format	NPS internal Format
IBM Tapes Input Format	NPS internal Format
BS2000 Channel Input Format	Line printer Format
Microfocus-Cobol Input Format	NPS internal Format
Bull Input Format	NPS internal Format
RS6000 Input Format	NPS internal Format
LCDS Input Format	NPS internal Format
ASCII (Barr) Format	NPS internal Format
CUSTOM#01:<length><user><PCC><data...>	NPS internal Format

6.4.10.2 Configuration

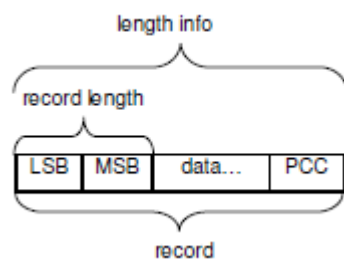
In the configuration dialog of the [Recordchange filter](#) the input as well as the output data format is selected. Depending upon the selected format further configuration options can be chosen.

Filter formats


NPS Internal Format

This format is used by [X-Metacode](#) to [PCL5](#) emulation as input data format. Likewise [PCL5](#) to [X-Metacode](#) emulation produces this format as output data format.

There are no configuration options required for this format.



LSB = Least significant byte, MSB = Most significant byte, PCC = Printer carriage control (1 byte)

 The record length (2 bytes) contains the complete length of the record in bytes, including the length of record length itself, e.g. <03><00><0B>. In this case the record contains no user data, only the length info and a PCC byte (<0B>).

The PCC byte is one of the following IBM 1403 (IBM 3211) machine control codes.

Code (hex)	Function	Code (hex)	Function
01	Write (no automatic space)*	0B	Space 1 line immediately
09	Write and space 1 line after printing	13	Space 2 lines immediately
11	Write and space 2 lines after printing	1B	Space 3 lines immediately
19	Write and space 3 lines after printing	8B	Skip to channel 1 immediately
89	Write and skip to channel 1 after printing	93	Skip to channel 2 immediately
91	Write and skip to channel 2 after printing	9B	Skip to channel 3 immediately
99	Write and skip to channel 3 after printing	A3	Skip to channel 4 immediately
A1	Write and skip to channel 4 after printing	AB	Skip to channel 5 immediately
A9	Write and skip to channel 5 after printing	B3	Skip to channel 6 immediately
B1	Write and skip to channel 6 after printing	BB	Skip to channel 7 immediately
B9	Write and skip to channel 7 after printing	C3	Skip to channel 8 immediately
C1	Write and skip to channel 8 after printing	CB	Skip to channel 9 immediately
C9	Write and skip to channel 9 after printing	D3	Skip to channel 10 immediately
D1	Write and skip to channel 10 after printing	DB	Skip to channel 11 immediately
D9	Write and skip to channel 11 after printing	E3	Skip to channel 12 immediately
E1	Write and skip to channel 12 after printing	03	No operation

*A record with a write-type control character immediately preceded by a record with the 'Write (no automatic space)' control character is processed according to the option selected for overprint handling

IBM 1403 (IBM 3211) machine control codes

Additionally, we enhanced the IBM 1403 (IBM 3211) machine control codes, in order to support channel commands in the range of 0 ... 15 for online jobs. The X-Metacode to PCL5 emulation supports also following additional codes.

Code (hex)	Function	Code (hex)	Function
81	Write and skip to channel 0 after printing	83	Skip to channel 0 immediately
E9	Write and skip to channel 13 after printing	EB	Skip to channel 13 immediately
F1	Write and skip to channel 14 after printing	F3	Skip to channel 14 immediately
F9	Write and skip to channel 15 after printing	FB	Skip to channel 15 immediately

enhanced IBM 1403 (IBM 3211) machine control codes

BARR S/370 byte or word record lengths format

This is a typical format of a BARR PC spooling system. There are two S/370 record formats available, S/370 byte length and S/370 word length record format. Files in S/370 record format have a 4 byte format header at the beginning of the file.

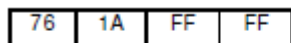
4 byte format header	record 1	record 2	...	record n
----------------------	----------	----------	-----	----------

Format of files in S/370 record format

The header indicates which format (byte or word length) is being used, therefore the filter detects automatically which format is used based on the header.

76	1A	FF	00
----	----	----	----

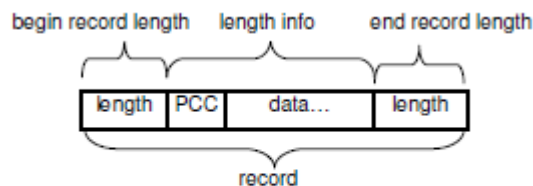
Format header S/370 byte length record format



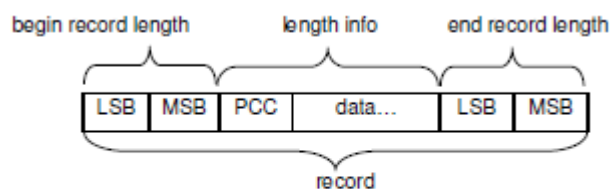
Format header S/370 word length record format

If the format is selected as input format of the [Recordchange filter](#), you can configure here whether the filter has to remove the BARR spool header record and/or the BARR FCB record from the data stream. Details on this data format can be found in the documentation of the BARR S/370 spool software.

If this format is selected as output format of the [Recordchange filter](#), the S/370 word length record format will be used. You can configure here the settings for the BARR spool header record, the BARR FCB record and the BARR spool trailer record. For the possible values see the BARR S/370 spool software documentation. The parameter 'JOB NAME ' of the BARR spool header record is configured by the filter software automatically with the name of the current print job.



S/370 byte length record format



S/370 word length record format

⚠ The begin record length and end record length bytes must have the same value. The record length contains the number of data bytes + PCC byte only.

The PCC is a IBM 1403 (IBM 3211) machine control code. Additionally, the Barr software reserves 3 special codes for the PCC.

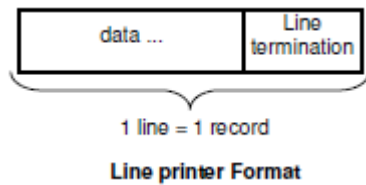
Code (hex)	Function
C6	3211 FCB
FD	Translate next command
FF	Start of spool header record

Commands reserved by Barr software

Line printer Format

This format assumes pure ASCII or EBCDIC printer lines. Each line must be terminated with [`<CR>`]`<LF>` or [`<CR>`]`<FF>`

- `<CR>` = Carriage Return (optional)
- `<LF>` = Line Feed
- `<FF>` = Form Feed

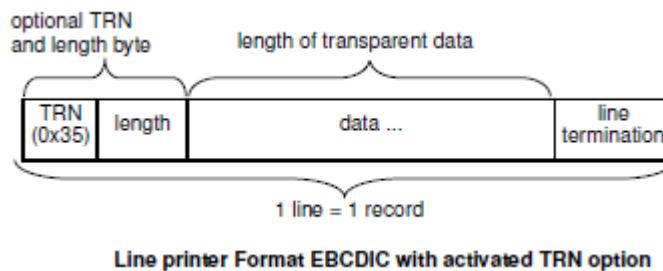


⚠ The line termination <FF> will always be translated into a 'Skip to channel 1' command. The line termination code may not be part of the data.

The line termination code depends on the configured mode ASCII or EBCDIC.

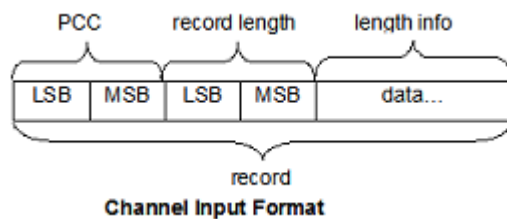
Line termination	Code ASCII (hex)	Code EBCDIC (hex)
<CR>	0D	0D
<LF>	0A	15 or 25
<FF>	0C	0C

There exists a check box option 'Scan for TRN code (0x35)' for the EBCDIC mode. If activated, the optional transparent code TRN (0x35) followed by one length byte -- if TRN exists-- must be at the beginning of the line. Within the length of the transparent data the 'line termination' code is disabled.



Channel Input Format

This data format is produced by the firmware of the channel input card of the [Mercury Server](#).



⚠ The record length (2 bytes) contains only the length of the data in bytes. PCC is a IBM 1403 (IBM 3211) machine control code.

IBM Tapes Input Format

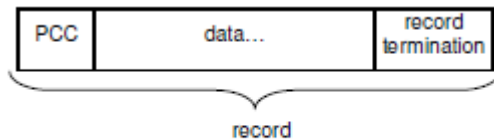
This format may vary from fixed blocked (FB) to variable blocked (VB) as well with different PCC types like IBM 1403 (IBM 3211) machine control codes or ANSI control codes in US ASCII or EBCDIC. Common to all variations and typical for this format is that the PCC is always at

the left position of the user data portion.



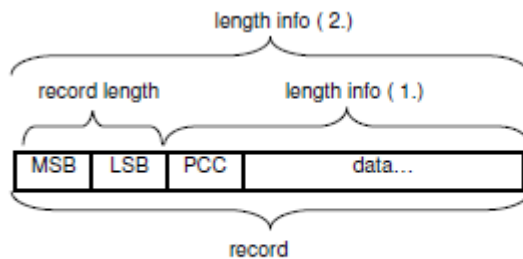
IBM Tapes Input Format (fixed blocked (FB))

⚠ The record length is on fixed blocked (FB) format always constant and may be configured (in bytes). The data must be padded with blanks up to the record length.



IBM Tapes Input Format (variable blocked (VB))

⚠ The record termination on variable blocked (VB) format must be always [`<CR>`]`<LF>` ([hex 0D optional] hex 0A), i.e. these hex codes may not be part of the data record



IBM Tapes Input Format (variable blocked (VB) 2 byte length info)

⚠ Instead of a record termination, this format has 2 leading bytes containing the length of the record. There exists two ways of interpreting the record length info:

1. The record length does not include the 2 length byte itself.
2. The record length includes the 2 length byte.

The ANSI control codes (PCC) for IBM tapes are shown in following table.

US ASCII graphical character representation	US ASCII tapes code (hex)	EBCDIC tapes code (hex)	Function
Space (blank)	20	40	Space 1 line before printing
0	30	F0	Space 2 lines before printing
-	2D	60	Space 3 lines before printing
+	2B	4E	Suppress space before printing*
1	31	F1	Skip to channel 1 and print
2	32	F2	Skip to channel 2 and print
3	33	F3	Skip to channel 3 and print
4	34	F4	Skip to channel 4 and print
5	35	F5	Skip to channel 5 and print
6	36	F6	Skip to channel 6 and print
7	37	F7	Skip to channel 7 and print
8	38	F8	Skip to channel 8 and print
9	39	F9	Skip to channel 9 and print
A	41	C1	Skip to channel 10 and print
B	42	C2	Skip to channel 11 and print
C	43	C3	Skip to channel 12 and print

*A record with a write-type control character immediately followed by a record with the 'Write (no space)' control character will be processed according to the job descriptor entry option selected for overprint handling

ANSI control codes for IBM tapes

BS2000 Channel Input Format

Each record must be terminated with [<CR>]<LF>



BS2000 Channel Input Format

⚠ The record termination code may not be part of the data.

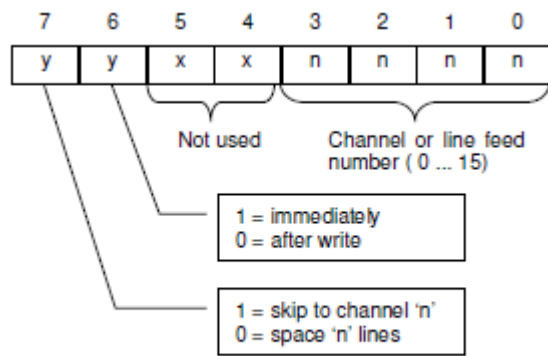
The record termination code depends on the configured mode ASCII or EBCDIC.

Record termination	Code ASCII (hex)	Code EBCDIC (hex)
<CR>	0D	0D
<LF>	0A	15 or 25

Record termination codes of 'BS2000 Channel Input Format'

⚠ When 'Line printer Format' is used as output format, depending on the 'PCC input type' configuration, ASCII or EBCDIC line printer data is generated. Also all 'Skip to channel x' commands are translated into a <FF> only.

The PCC is one byte long, where each **bit** has following function.



PCC function of BS2000 Channel Input Format

- ⚠ The PCC can be translated via a user defined translation table.

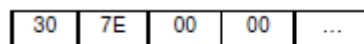
Microfocus-Cobol Input Format

This format was a special customer adaptation. There was no reliable information or documentation about this format. Microfocus-Cobol files may have an optional 128 byte control block header at the beginning of the file.



Format of files in Microfocus-Cobol record format

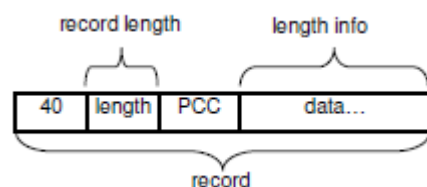
The meaning of the complete header was not documented, but a header can be recognized within the first 4 bytes.



Microfocus-Cobol control block header recognition

- ⚠ The 128 byte header will be recognized automatically and removed.

A record starts always with the hex code 0x40.



- ⚠ The record length (1 byte) contains only the length of the data in bytes.

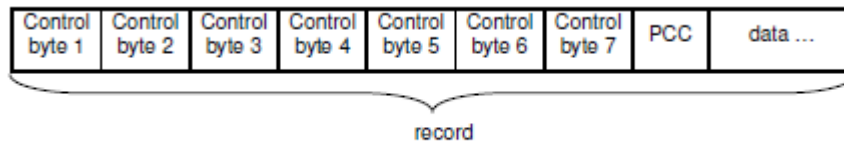
The meaning of the PCC is following :

Code (hex)	Function
50	Write and skip to channel 1 after printing
30 ... 3F	Write and space 'n' lines after printing, where 'n' = 0 ... 15 (the low nibbles of the codes 30 ... 3F) If 'n' = 0 → Write (no automatic space)

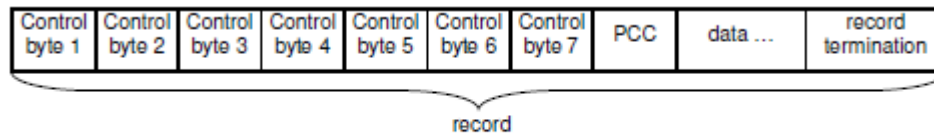
Microfocus-Cobol PCC control codes

Bull Input Format

This format was a special customer adaptation. There was no reliable information or documentation about this format. It may be fixed blocked (FB) or variable blocked (VB). Each record starts always with 8 control bytes, where byte 8 is the PCC.



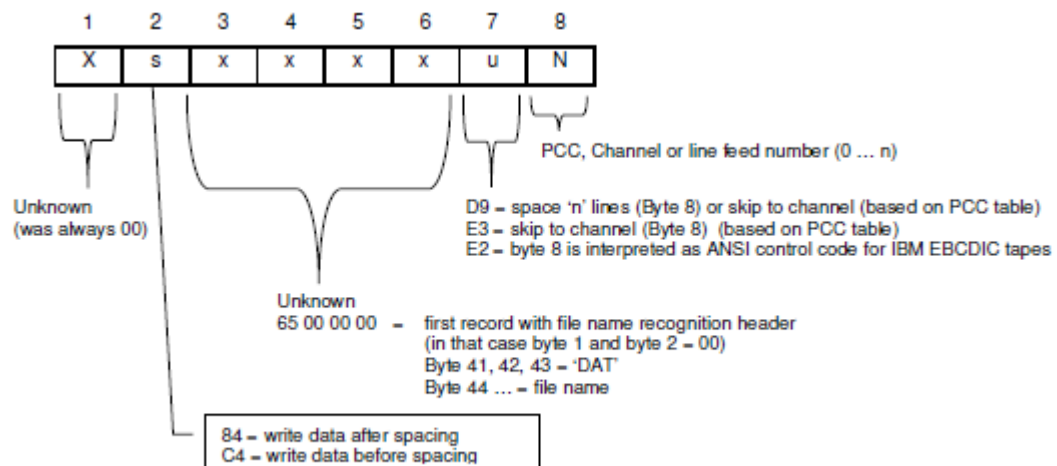
Bull Input Format (fixed blocked (FB))



Bull Input Format (variable blocked (VB))

⚠ The record termination on variable blocked (VB) format must be always [<CR><LF> ([hex 0D optional] hex 0A), i.e. these hex codes may not be part of the data record.

The meaning of the 8 **control bytes** is following. The most information has been extracted by experience.



Meaning of the 8 control bytes of 'Bull Input Format'

⚠ Value of byte 8 must be translated in accordance to the 4 PCC tables.

There may exist 4 configurable PCC tables (PCC1, PCC2, PCC3 PCC4), which translates the PCC (control byte 8) into following functions. These values has been extracted from some PCC table definitions of the customers JSL's and some has been determined by experience.

Code (hex)	Function	Code (hex)	Function
00	Skip to channel 1	0A	Space 10 lines
01	Space 1 line	0B	Space 11 lines
02	Space 2 lines	0C	Skip to channel 1
03	Space 3 lines	0D	Space 13 lines
04	Space 4 lines	0E	Space 14 lines
05	Space 5 lines	0F	Space 15 lines
06	Space 6 lines	40	Space 1 line
07	Space 7 lines		
08	Space 8 lines		
09	Space 9 lines		

Note: Every other code results per default in a 'Write (no automatic space)' function

PCC1 translation table of 'Bull Input Format'

Code (hex)	Function	Code (hex)	Function
00	Skip to channel 1	0A	Space 10 lines
01	Space 1 line	0B	Space 11 lines
02	Space 2 lines	0C	Space 12 lines
03	Space 3 lines	0D	Space 13 lines
04	Space 4 lines	36	Skip to channel 2
05	Space 5 lines	37	Write (no automatic space)
06	Space 6 lines	3A	Skip to channel 3
07	Space 7 lines	3D	Skip to channel 4
08	Space 8 lines	40	Space 1 line
09	Space 9 lines		

Note: Every other code results per default in a 'Write (no automatic space)' function

PCC2 translation table of 'Bull Input Format'

Code (hex)	Function	Code (hex)	Function
00	Skip to channel 1	0A	Space 10 lines
01	Space 1 line	0B	Space 11 lines
02	Space 2 lines	0C	Space 12 lines
03	Space 3 lines	0D	Space 13 lines
04	Space 4 lines	37	Write (no automatic space)
05	Space 5 lines	40	Space 1 line
06	Space 6 lines		
07	Space 7 lines		
08	Space 8 lines		
09	Space 9 lines		

Note: Every other code results per default in a 'Write (no automatic space)' function

PCC3 translation table of 'Bull Input Format'

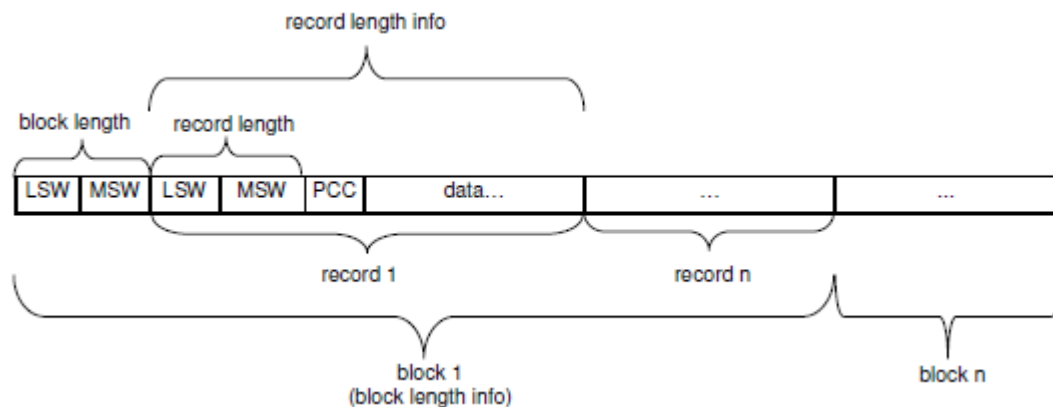
Code (hex)	Function	Code (hex)	Function
00	Skip to channel 1	0E	Skip to channel 4
01	Space 1 line	0F	Skip to channel 5
02	Space 2 lines	1A	Skip to channel 11
03	Space 3 lines	1C	Skip to channel 6
04	Space 4 lines	1D	Skip to channel 7
05	Space 5 lines	23	Skip to channel 8
06	Space 6 lines	26	Skip to channel 2
07	Space 7 lines	37	Write (no automatic space)
08	Space 8 lines	38	Skip to channel 9
09	Write (no automatic space)	3C	Skip to channel 15
0A	Space 10 lines	3E	Skip to channel 12
0B	Skip to channel 13	3F	Skip to channel 3
0C	Skip to channel 14	40	Space 1 line
0D	Skip to channel 10		

Note: Every other code results per default in a 'Write (no automatic space)' function

PCC4 translation table of 'Bull Input Format'

RS6000 Input Format

This format was a special customer adaptation. There was no reliable information or documentation about this format.

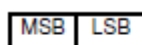


LSW = Least significant word, MSW = Most significant word, PCC = Printer carriage control (1 byte).

⚠ The block length and record length is 4 bytes long. The block length contains the complete length of the block in bytes (including the 4 bytes block length info). This block may contain 1 ... n records. The record length contains the complete length of the record in bytes (including the 4 bytes record length info). The data stream itself may contain 1 ... n blocks.

The PCC is a IBM 1403 (IBM 3211) machine control code.

The byte order of the word (LSW or MSW) is following :



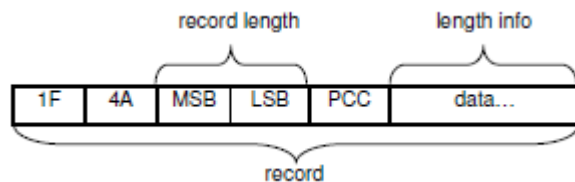
word organization of RS6000 record length

LSB = Least significant byte, MSB = Most significant byte

LCDS Input Format

This format was a special customer adaptation. There was no reliable information or documentation about this format.

A record starts always with a constant two byte record identifier 0x1F 0x4A



⚠ The record length (2 byte) contains only the length of the data in bytes. The record identifier will be stripped.

The PCC is a IBM 1403 (IBM 3211) machine control code.

ASCII (BARR) Format

This format was a special customer adaptation. There was no reliable information or documentation about this format.

This format is similar to the 'Line printer Format' with two exceptions.

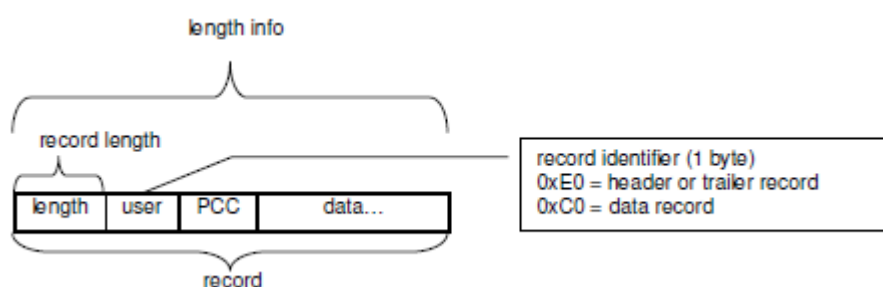
Exception 1: Empty lines will be padded with a space (blank) character.

Exception 2: Only ASCII input (no EBCDIC) can be processed.

⚠ Additionally this filter can be configured to strip a user defined block at the beginning of the data stream.

CUSTOM#01:<length><user><PCC><data...>

This format was a special customer adaptation. There was no reliable information or documentation about this format. Also no information how to interpret a header or trailer record was available.



⚠ The record length (1 byte) contains the complete length of the record in bytes, including the length of record length itself. The user byte is a constant record identifier.

⚠ A header or trailer record (user byte = 0xE0) will be stripped.

The PCC is a IBM 1403 (IBM 3211) machine control code.

6.4.11 Regular Expressions Filter

6.4.11.1 Application

The Regular Expressions filter is used to replace or remove individual characters and character strings in the data stream before and / or after the emulation of the print data. The search strings are defined by using the PCRE (Perl Compatible Regular Expressions) syntax.

Examples:

- The input data stream contains control commands which are not supported by the emulation or the output device. Use the Stringchange filter in this case to remove these commands.
- The input data stream contains special characters, such as umlauts, which must be replaced by other characters or character strings, for example 'Ä' becomes 'ae', etc. Use the Stringchange filter in this case to replace characters or character strings.

6.4.11.2 Configuration

The Regular Expressions filter requires the following information to replace or remove a character or a character string:

Ignore case

Enable to ignore case of string.

This allows to define whether the replacement is to be completed taking case into account, in other words whether 'abc' and 'Abc' or just 'abc' is to be replaced. The setting applies for all replacements.

Default: disabled

Greedy mode

See PCRE (Perl Compatible Regular Expressions) syntax

Replacements

Shows a list of all the defined destination character strings for replacements, in other words the characters or character strings that are to replace the source character strings. If the display is empty, no replacement will be made and the source character string will simply be removed.

Add

Add and define a new replacement.

see [Definition of a replacement](#).

Edit

Edit an already [defined replacement](#).

Remove

Delete a previously [defined replacement](#).

6.4.11.3 Definition of a replacement


Define a replacement.

Enter the following information:

Search for

Define character string which is to be replaced. The syntax is described in PCRE (Perl Compatible Regular Expressions) syntax

If case is ignored depends on the setting of '[Ignore Case](#)'.

 Characters that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code. The '%' character itself is displayed as '%%'.

Examples:

- 'Hello' replaces the word 'Hello'
- 'Ä' replaces the German umlaut 'Ä'
- '%0D%0A' replaces the ASCII control characters Carriage Return (CR) and Line Feed (LF).

Replace with

Define character or character string to replace the string defined in 'Search for'. The expression is evaluated by a parser. The parser syntax is described in Interpreter Rules. The found string can be used using field \$1. Variables, detected by the Field Scanner Filter, may also be used (e.g. \$Name).

Leave empty to delete the character string from the data stream. Replacements that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code. Variables, detected by the Field Scanner Filter, may also be used (e.g. \$Name).

Examples:

- Search for 'Hello World' Replace with 'Hey You' replaces 'Hello World' with 'Hey You'
- Search for 'Ä' Replace with 'Ae' replaces the umlaut 'Ä' with 'Ae'
- Search for '%0D%0A' Replace with '%0A' replaces CR/LF with LF
- Search for '%1BD' Replace with "" removes ESC-D

6.4.12 Stringchange Filter

6.4.12.1 Application

The Stringchange filter is used to replace or remove individual characters and character strings in the data stream before and / or after the emulation of the print data.

Examples:

- The input data stream contains control commands which are not supported by the emulation or the output device. Use the Stringchange filter in this case to remove these commands.
- The input data stream contains special characters, such as umlauts, which must be replaced by other characters or character strings, for example 'Ä' becomes 'ae', etc. Use the Stringchange filter in this case to replace characters or character strings.

6.4.12.2 Configuration

The Stringchange filter requires three pieces of information to replace or remove a character or a character string:

- Character or a character string to be replaced ('Search for')
- Character or a character string, which is to replace the source ('Replace with') or empty, if the source is to be removed.
- Consideration to low or high case letters.

Ignore Case

Enable to ignore case of string.

This allows to define whether the replacement is to be completed taking case into account, in other words whether 'abc' and 'Abc' or just 'abc' is to be replaced. The setting applies for all replacements.

Default: disabled

Search for

Shows a list of all the defined source character strings for replacements, in other words which characters or character strings must be replaced.

Replace with

Shows a list of all the defined destination character strings for replacements, in other words the characters or character strings that are to replace the source character strings. If the display is empty, no replacement will be made and the source character string will simply be removed.

Add

Add and define a new replacement.

see [Definition of a replacement](#).

Edit

Edit an already [defined replacement](#).

Remove

Delete a previously [defined replacement](#).



The number of replacements and the length of each replacement are limited as follows:

- A replacement character string is limited to 512 characters.
- The possible number of replacements is calculated using the following formula:

Total of lengths of all sources + number of sources + 1 <= 32767

6.4.12.3 Definition of a replacement


Define a replacement.

Enter the following information:

Search for

Define character string which is to be replaced.

If case is ignored depends on the setting of '[Ignore Case](#)'.

 Characters that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code. The '%' character itself is displayed as '%%'.

Examples:

- 'Hello' replaces the word 'Hello'
- 'Ä' replaces the German umlaut 'Ä'
- '%0D%0A' replaces the ASCII control characters Carriage Return (CR) and Line Feed (LF).

Replace with

Define character or character string to replace the string defined in 'Search for'.

Leave empty to delete the character string from the data stream. Replacements that cannot be entered using the keyboard are specified with a preceding '%' character and a two-character hexadecimal code.

Examples:

- Search for 'Hello World' Replace with 'Hey You' replaces 'Hello World' with 'Hey You'
- Search for 'Ä' Replace with 'Ae' replaces the umlaut 'Ä' with 'Ae'
- Search for '%0D%0A' Replace with '%0A' replaces CR/LF with LF
- Search for '%1BD' Replace with "" removes ESC-D

6.4.13 Textextractor Filter

6.4.13.1 Application

The [Textextractor filter](#) of the [Mercury Server](#) is used to extract pure printable data from PCL data streams at the input and/or output of the [Mercury](#) emulations.

The resulting data stream may be used for further processing and consists of pure printable data which may contain ASCII control codes like <CR>, <LF> or <FF> (depends on configuration of the filter).

6.4.13.2 Configuration

The default settings of the [Textextractor filter](#) are set to generate a ASCII formatted output that will represent approximately the printable page layout of the origin PCL data stream.

6.4.13.2.1 Optimize

Extract also text within macros

When active, the printable data within a PCL macro will be inserted to the output at each macro execute or call.

Detect simulated bold printing and remove multiple characters/words

Some applications might print the same text elements twice or more at slightly the same X position in order to simulate bold printing on a laser printer.

e.g. ...<ESC>&f0SBolded text<ESC>&f1S<ESC>&a+3HBolded text...

Such a sequence would result into printable output like ...Bolded textBolded text...

To avoid these multiple text strings activate the detect bold option.

Sort lines in ascending Y order

The printable data will be send to the output in dependence of the Y position of the text element from top of page to bottom of page. Otherwise the text elements are given out at the scope they are encountered.

Combine lines within threshold

When active, all text elements on a Y position \pm threshold (in 1/300 Inch) are considered as text elements on the same line position.

Insert multiple space delimiters for horizontal moves

When active, for each horizontal X movement between text elements, one or multiple space delimiter (if given, see [Delimiter](#)) are inserted.

Space delimiter width

Define here the width of the space delimiter. This determines the number of inserted space delimiter between text elements.

i.e. 'number of inserted space delimiter' = 'X movement' / 'width of the space delimiter'

Fixed user width in CPI

Sets the width of the space delimiter to the given fix value in CPI (Characters Per Inch).

Use current HMI

Sets the width of the space delimiter to the current HMI (Horizontal Motion Index) of the PCL environment.

6.4.13.2.2 Delimiter

Each delimiter may be a arbitrary character sequence. Non printable codes may be entered as 2 digit hex values preceded by a '%' character, e.g. %0C for a form feed <FF>.

There exists some special keywords for delimiters which must be enclosed into <...> and may be used as dynamic variables. These variables will be replaced with the current values.

Possible delimiter keywords :

Keyword :	Function :
<pjljobname>	The @PJM JOB NAME=...
<pagecount>	Current PCL page counter
<jobname>	Name of the job
<srcaddress>	Address of senders as IP address or Novell queue name
<username>	Name of user
<destport>	Destination port as queue name from LPD protocol
<destaddress>	Specific device if defined in Logical Printer
<jobid>	Job-ID
<ticks>	Milliseconds since system start
<date>	Date
<time>	Time
<creationtime>	Start time of job
<starttime>	Emulation time of job

Job delimiter

Job header

Inserts the given sequence at the begin of the job.

Job trailer

Inserts the given sequence at the end of the job.

Page prefix

Inserts the given sequence at the begin of each page.

Page postfix

Inserts the given sequence at the end of each page.

Text line delimiter

Prefix

Inserts the given sequence at the begin of each line.

Postfix

Inserts the given sequence at the end of each line.

Space delimiter

Defines a sequence for a space delimiter which may be inserted at horizontal X movements.

(if activated, see [Optimize](#))

6.5 Overlay Manager

6.5.1 Overview

Overlay Manager - Overview

Using the Overlay Manager it is possible to define overlays as PCL macro data which is added to an ASCII formatted print job or PCL jobs. Each page of the print job can have a different overlay which will be added to the corresponding page.

This application is similar to usage of preprinted paper forms. Using plain paper sheets, the form will be added as overlay to the print job.

The Overlay Manager enables you to define different configurations for different print jobs. Using this feature, it is possible to use specific overlays for print jobs when a so called trigger string has been detected.

A specific configuration for overlays is called Configuration Profile. Which profile you are working on at configuration time, is displayed in the combo box. Each profile is using the same parameters:

- [ASCII / PCL Format](#)
- [General](#)
- [Overlays](#)
- [Pages without overlay](#)
- [Fonts](#)

6.5.2 Configuration


6.5.2.1 ASCII / PCL Format

Plain Text Format (ASCII)

Select this option if print job data is delivered containing plain ASCII formatted data without any control codes. In this case print jobs have to be formatted using spaces, carriage returns and line feeds for a proper print out on the paper.

Printer Command Language (PCL)

Select this option if print job data is delivered as a complete PCL print job. In this case print jobs must contain data with standard PCL commands.

 Some printer drivers especially in Windows are using printer specific extensions of PCL standard. These extensions may not be processed by the Overlay Manager or produce problems with the printed output. Please use a printer driver which produces PCL closest to the standard, like HP LaserJet Series II or HP LaserJet 5Si.

6.5.2.2 Profile

New

Creates a new Configuration Profile. The new profile is automatically selected and preset with some standard values.

Add...

Adds a new profile from a disk file to the list Configuration Profile. The added profile becomes also the current profile.

Replace...

Replaces the current profile with a profile which will be read from a disk file.

Save as...

Saves the current profile to a file on the disk. This file later on can be read using the buttons Add and Replace.

Delete


Deletes the current profile, if you answer the question if you sure with yes.

Profile Name

Arbitrary name of the current Configuration Profile

Trigger String

Here you have to define a character sequence (Trigger String), which is unique and should be near the beginning of the print job. This character sequence - if present - will activate the current profile.

 This input field requires character strings which are printed as readable text. Umlauts and characters which can not be entered using the keyboard, may be entered by their hexadecimal value with a preceding '%' sign. The '%' sign itself is entered as '%%'.

Example: '€%' is entered as '%80%%'.

Condition

This field lets you define a conditional expression. If such an expression becomes true the actual configuration profile will be used. The expression will be evaluated by a built-in interpreter, whose parsing syntax is described here: [Interpreter Rules](#). Typically an expression will contain one or more data fields, which are the result of a scan of the [Field Scanner](#) filter module. This filter scans the print job data using one of several scanning methods to detect data fields.

Default Configuration

If this field is switched on, the current Configuration Profile will be used as standard profile, i.e. this configuration will be used as long as no Trigger String will activate another Configuration Profile.

Keep current configuration after job

If this field is switched on, the current Configuration Profile will be used until another Configuration Profile is loaded by detection of its Trigger String also for new jobs.

If turned off, the Configuration Profile will be changed due to a Trigger String, but after the job the standard profile will be used again.


Transparent Mode

If this field is switched on, the current Configuration Profile as a standard profile can also process jobs which are not in ASCII format. In this case this Configuration Profile cannot add any overlay


This is commonly used to define a transparent standard profile, which is able to process all kind of print jobs and later is switched to other Configuration Profiles for processing overlays due to Trigger Strings in special jobs.

6.5.2.3 General

Use Init Sequence

 This option is only available with [ASCII formatted print jobs](#).

Select this field to initialize the printer at beginning of the print job. The initialization consists of parameters for Page Definition, parameters for Default Font and additional User Commands.


 Usually you must initialize the printer at least using a Default Font. Because of this, this field will be activated in most cases. But if the print job has passed an emulation which will control the fonts it might be necessary to suppress the initialization.

PJL User Commands

 This option is only available with [ASCII formatted print jobs](#).


Here you can add your own specific PJP commands which will be send for initialization at the beginning of the print job.

Page Definition

 This option is only available with [ASCII formatted print jobs](#).

Using this button you will enter a dialog for paper oriented parameters concerning paper size, orientation, single side or double side printing. Here you also define how your printer is switched to edge-to-edge printing.

Default Font

 This option is only available with [ASCII formatted print jobs](#).

Using this button you will enter a dialog for setting up the default font for the printing of the print job data. Possible settings are:

[Usual Setting](#), for using a standard PCL Font,

[Font File](#), for using a soft font from a file on disk,

[User Defined](#), for using a special PCL command, i.e. if you want select a font from an additional font cartridge of the printer.

Using parameters Characters Per Inch and Lines Per Inch it is possible to control the size and the so called horizontal (HMI) or vertical (VMI) motion index the printer is using with the selected font.

PCL User Commands

 This option is only available with [ASCII formatted print jobs](#).

Here you can add your own specific PCL commands which will be send for initialization at the beginning of the print job.

Page Separation Command

⚠ This option is only available with [ASCII formatted print jobs](#).

The character sequence which is defined here is interpreted as the page separator. Using ASCII formatted data it is always the character for form feed (FF = '%0C'). Other character strings can be used instead.

⚠ This input field requires character strings which may contain characters which can not be entered using the keyboard. These are usually control characters as <FF>. You can enter these characters by their hexadecimal value with a preceding '%' sign. The '%' sign itself is entered as '%%'.

Lines Per Page

⚠ This option is only available with [ASCII formatted print jobs](#).

If no page separation command can be found within the print job data, this limitation of lines per page controls, when a page break is added. A line is detected by the ASCII line feed character (LF = '%0A').

⚠ Separating pages by a number of lines has higher priority than a page separator command. I.e. if the configured number of lines is reached, a page break will always be added, although the 'real' page break command might be placed later within the data stream.

Duplicate Page Separation

⚠ This option is only available with [ASCII formatted print jobs](#).

If a print job should be printed as a duplex job but was created as a simplex job on the host system, this option can be used to duplicate the page separation. If enabled, after each frontside an empty backside will be inserted, which may also be used for insertion of overlays.

⚠ This switch is enabled only, if a duplex configuration is set under Page Definition.

Skip Empty Pages

⚠ This option is only available with [ASCII formatted print jobs](#).

Activate this field if blank pages should be suppressed.

Macro ID numbers start at

⚠ This option is only available with [PCL formatted print jobs](#).

To avoid overwriting preexisting macros on your printer, you may define a start value that is higher than any of the existing. Macros created by the Overlay Manager will have IDs starting at the defined value.

6.5.2.4 Overlays

Page


To each page of a print job a PCL data file or a PCL command sequence can be added as an overlay. To spare a page just leave the input fields for the overlays blank. If more pages are printed than overlays are defined, the overlay order is repeated from page one.

Example 1: The first page should be printed using overlay A and the second page using overlay B.

Therefore you enter at Page 1 the file for overlay A and the file for overlay B at Page 2. All odd pages 1,3,5,... are printed now using overlay A and all even pages 2,4,6,... using overlay B.

Example 2: Additional to example 1 each 3rd page should be printed without an overlay.

To manage that, enter the overlay files for page 1 and 2 as described above. Additionally enter an empty field for overlay data at Page 3.

 The counting of pages will re-start at page 1 after . Pages, which are defined as , due to a unique byte sequence in their data, are not counted at all.

Copy

Furthermore it is possible to generate copies of each page using different overlays for each copy. Therefore the data of the print job will send more than once to the printer.

Example: Page 1 is to be printed with 2 copies. The original page is to be printed using overlay A, the first copy using overlay B and the second copy using overlay C.

Therefore enter at Page 1, Copy 1 the file for overlay A, at Page 1, Copy 2 the file for overlay B and at Page 1, Copy 3 the file for overlay C.

Background Overlay

Enter name and path of the PCL file here, to be added to the page **before** printing job data.

Rotation

Select the rotation of the background overlay in steps of 90 degrees.

Foreground Overlay

Enter name and path of the PCL file here, to be added to the page **after** printing job data.

Rotation

Select the rotation of the foreground overlay in steps of 90 degrees.

Paper Source

This field controls the paper source for the page using the overlay. Select the desired paper source of the printer from the list of paper trays.

Output Bin

This field controls the output tray for the page using the overlay. Select the desired output bin of the printer from the list of output bins.

Horz. Text Offset


Using this parameter you can move the output of the print job's text horizontally on the overlay. The movement will be done relatively to the original position.

Vert. Text Offset

Using this parameter you can move the output of the print job's text vertically on the overlay. The movement will be done relatively to the original position.


Duplexing

Single side printed sheets can be changed to double side printed sheets here. Select the mode for duplex printing or select 'none' to keep single sided printing.

 This option is available with [PCL formatted print jobs](#) only. Inserting backsides will be done for single sided sheets only. For already double side printed sheets the setting will be ignored.

Backside Macro

Enter name and path of the PCL file here, to be inserted an additional backside according to the settings at Duplexing.

 This option is available with [PCL formatted print jobs](#) only. Inserting this file as backside will be done for single sided sheets only. For already double side printed sheets the setting will be ignored.

Rotation

Select the rotation of the backside macro in steps of 90 degrees.

Reset Trigger

A Reset Trigger can be defined for every page. If this trigger is detected in the data stream, execution will continue at the page defined in column Page.


6.5.2.5 Pages without overlay**Trigger String**

Here you define character sequences (Trigger Strings) which define pages without overlays. If one of these sequences is found on a page, this page will be printed without an overlay. Also this page will not be counted in the calculation of the overlay order defined under [Overlays](#).

Paper Source

This field controls the paper source for the page using the overlay. Select the desired paper source of the printer from the list of paper trays.

Horz. Text Offset

 This option is only available with [ASCII formatted print jobs](#).

Using this parameter you can move the output of the print job's text horizontally on the overlay.

Vert. Text Offset

 This option is only available with [ASCII formatted print jobs](#).

Using this parameter you can move the output of the print job's text vertically on the overlay.

6.5.2.6 Fonts



This options are only available with [ASCII formatted print jobs](#).

Trigger String

Here you enter a list of character sequences (Trigger String) which correspond to fonts or PCL command sequences for font selection. If one of these character sequences is found in the print data it will be replaced by the corresponding PCL command sequence or the corresponding PCL soft font which was read from the entered font file will be activated.

Font File / PCL Command

Using the button you will enter a dialog for setting up the default font for the printing of the print job data. Possible settings are:

[Usual Setting](#), for using a standard PCL Font,

[Font File](#), for using a soft font from a file on disk,

[User Defined](#), for using a special PCL command, i.e. if you want select a font from an additional font cartridge of the printer.

Using parameters [Characters Per Inch](#) and [Lines Per Inch](#) it is possible to control the size and the so called horizontal (HMI) or vertical (VMI) motion index the printer is using with the selected font.

6.6 Output Ports

6.6.1 Overview

Output ports are used as ports for the output of print data. Currently two different output types are supported:

- [Channel](#)
- [System Device](#)
- [File](#)
- [Printer](#)
- [SMTP](#)
- [TCP/IP / Local TCP](#)
- [Fiery](#)

To define a new output

To define a new output port in the system, proceed as follows:

1. In the main program window, select a Logical Printer.
2. Edit the highlighted Logical Printer, either by clicking on the 'Edit Logical Printer...' button or by menu: setup/edit Logical Printer...
3. Select the current output (default: Fiery) from 'Output Interface'.
4. Click on 'Remove...' to remove the current output.
5. Click on 'Add...' to add a new output.

6. A dialog box appears. Select the required output from the list.
7. Click 'OK'. The new output is now active.

6.6.2 Properties

6.6.2.1 Fiery

Connection

IP Address

Enter IP Address or DNS name of the Fiery Controller.

Username

Enter username for connecting to the Fiery Controller.

Default: admin

Password

Enter password for connecting to the Fiery Controller.

Properties

Logical Printer

Select logical printer of the Fiery Controller where the print jobs will be sent to.

Possible values are:

- direct
- print
- hold

Default: print

Timeout

Set timeout for the communication with the Fiery Controller.

Possible values range from 3 to 120 seconds.

Default: 30 seconds

Disconnect after job transmission

Enable this option to disconnect [Mercury](#) from the Fiery Controller after each print job.

If a new job occurs [Mercury](#) will re-connect. Disable this option if connecting to the Fiery Controller takes too long.

Default: disabled

6.6.2.2 Settings

Description

An optional description of the interface for display

Execute command after printing

It is possible to execute an external program after a job is completed. Therefore you have to enter the path and filename of the program file here. The specified program is executed, after the job is printed.

Optional placeholders can be used as parameters:

<jobid>	ID of the job
<jobname>	Name of the job
<srcaddress>	Address of senders as IP address or Novell queue name
<dstaddress>	Destination address of a job
<username>	Name of user
<destport>	Destination port as queue name from LPD protocol
<starttime>	Time when job started
<endtime>	Time when job was printed
<jobsize>	Size of the job
<pages>	Number of pages
<filename>	Name of the output file (file ports only)
<filesize>	Size of the output file (file ports only)

Buffer

Read block size

Block size for read operations in KB. When data are received, the data are read in blocks of the specified size as long as the port supports this. However, this only concerns output ports that supply a response, such as a bi-directional printer port. Normally a block size of 32 KB is used.

Default: 32 KB

Timeouts



These settings are only accessible by the support

Read Timeout

Timeout for data reception. This value allows the time properties of Mercury for read operations to be influenced. This value should only be changed by experienced users.

Default: 50 msec

Write Timeout

Timeout for data output. This value allows the time properties of Mercury for write operations to be influenced. This value should only be changed by experienced users.

Default: 1000 msec

Delay, if data received

Print jobs that arrive at the same time are processed in parallel. Each print job is allocated a certain amount of processor time for this purpose, which it must share with other print jobs. This value allows you to enter the time in milliseconds that a print job may be delayed in

receiving data so that other print jobs can be completed fluently.

Default: 0 msec

Delay, if no data received

If no data are currently being received for a print job, the print job may be delayed and the available processor time can be assigned to other print jobs. This value defines the time in milliseconds that a print job can be delayed if no data are currently being received for this print job.

Default: 50 msec

6.6.2.3 Device Assignment

It is possible to connect alternative devices to an *output port*, to which data will be sent. If the main device is out of order or not available the alternative devices will be used. This can be used to realize a simple kind of load balancing or a so called printer pool. At maximum 25 alternative devices can be defined. If the same device should be used, but the address is not known at configuration time (e.g. when address is delivered by protocol), use '*' as device address.



Only devices connected to the same type of *output port* can be used as alternative devices. This means it is not possible to use for example a device which is connected to an *output port* of type *printer* as an alternative device of an *output port* of type *TCP/IP*.

Use if error occurred

The priority of usage depends on the position in the list where high priority is placed on top and low priority on bottom. If a print job could not be sent to the first device, the second one will be used. If the second device is also unavailable the third one will be tried and so on. If all devices were not reachable the first one will be tried again.

Round-robin

For every print job the next device in the list is used.

Load Balancing

Every print job will be sent to a device, which is currently not used. The priority of usage depends on the position in the list.

Conditional Processing

For every device a lower limit of the page count can be specified. Jobs will be sent only to the appropriate device. The jobs have to be spooled, so that the total page count is known. Jobs with unknown page count will be sent to the default device. A different page count for color and b/w pages can be specified.



Please note that the page counting of a raw emulation must be activated in the configuration and the data type recognition must be activated in the input port configuration.



If the Mercury license does not include *Clustering*, only the 'Use if error occurred' method is available!

Add...

Click here to add an alternative device to the list.

Edit...

Click here to edit an alternative device.

Delete...

Click here to delete the selected alternative device from the list.

Move up

Click here to move the selected alternative device to a position above. This will increase its priority of usage. If using the 'Page count' method, the position is not relevant.

Move down

Click here to move the selected alternative device to a position below. This will decrease its priority of usage. If using the 'Page count' method, the position is not relevant.

Redirect after x sec, if busy

If a connect to a device fails, it will be retried within the given amount of time. If a connect was not successful after this time, the job will be reprinted using the next alternative device.

Default: 0 (off)

Redirect complete job / missing pages after x min, if no print activity

If no progress of the print job is detected within the given amount of time the job will be canceled and be reprinted using the next alternative device. It is recommended that [Job Monitoring](#) is enabled. A job may only be reprinted if [Spooling](#) is enabled.

If a job was canceled by the [Progress Timeout](#) while it was printing, the whole job will be reprinted using an alternative device if this option is activated.
If deactivated, only the missing pages are redirected to an alternative device.

 If the Mercury license does not include [Reprinting](#), this option is not available!


Default: disabled

6.6.2.4 Job Monitoring

Job Monitoring**Active**

Enable this option to get feedback concerning the output state of a print job from the printer.

Job Monitoring also must be enabled to get feedback from the printer about printed pages. The job state will be set to 'Printed' not before the printer has reported the job as completed. If disabled, only the converted pages are shown but not the printed and a job becomes the state 'Printed' as soon as the job data have been transferred to the printer.

 Job Monitoring is not supported by all output interfaces. A bi-directional connection must be established during the printing process.

Page Notification


The host system will get feedback when a page has been printed.

Job Notification

The host system will get feedback when a job has been printed.

Display job information


Display the job id of the last printed job on the printer panel. The option is only accessible, if printer monitoring is done by PjL.

 Job Notification is not available if the [Spooling](#) option is enabled.

Default: Job Monitoring active and Job Notification enabled.

SNMP Device Monitoring**Active**

Enable this option to get feedback concerning the state of a printer by SNMP.

 SNMP Device Monitoring is not supported by all output interfaces. Timeout and interval may be adjusted in the server options.

Default: off

Always monitor device

A device will be monitored by SNMP continuously.

Default: off

Stop monitor after x min

The monitoring of the device will be stopped <x> minutes after a job has been printed.

Default: on (0 minutes)

Community

SNMP Community

Default: public

Port

SNMP Port

Default: 161

6.6.2.5 Data (Output Port)

Post Processing

A Post Processing module for data processing after the emulation can be specified and configured here.

Data encryption

No encryption


The data will be sent uncompressed and decrypted.

Compress data

The print job data will be compressed. The compression ratio depends on the data. The receiving Mercury system will unpack and decrypt print job data if the option Decompress and decrypt data automatically is activated at the corresponding input port. This option is useful if network load is to be reduced.

Compress and encrypt data

If this option is activated, the print job data will be compressed and encrypted. The encryption algorithm can be chosen from AES (Rijndael), Blowfish or DES. DES is no longer considered secure, but is available for compatibility with older Mercury versions. The receiving Mercury system will unpack and decrypt print job data if the option Decompress and decrypt data automatically is activated at the corresponding input port. This option is useful if network load is to be reduced and / or due to security reasons.

 Use 'Compress data' or 'Compress and encrypt data' only when a second Mercury System is receiving the data on the other side.

Zip Output

The output data will be zipped.

Password

Password used to encrypt the data. The receiving Mercury must have the same password specified in its receiving input port. If there is no password specified, Mercury will use its internal default password.

Send job information within data stream

If sending print jobs to a Mercury input port, additional information about the print job as source, name etc. can be transferred. This information will be transferred before the print data. To use this feature, the option of the corresponding input port must be checked, too. This option must not be used, if the receiver is not a Mercury input port.

Strip PJI Feedback

If activated, PJI feedback of a printer will be stripped off. Deactivate this option, if the sending host should receive PJI feedback from the printer.

 Only available if supported by the specific output port.

 If a Post Processing module is specified, data encryption is not available

6.6.2.6 Data Capturing (Output Port)

Converted Data Capturing

Data capturing allows to write data to a file after the data is converted by the emulation. This feature should only be used for testing purposes because it slows down the system performance.

Active

Set to 'Active' to log sent data.

Default directory for the log file is [Mercury application path]\DATA\DSPPATHx, default name of the log file is Oxxxx.DAT. Each job is placed in a separate file.

Default: not active

Without protocol data / With protocol data

Set if you want the protocol data to be saved to the log file, too.

Default: Without protocol data

Maximum files to capture

Define the maximum number of files to be captured.

Default: 99

 If the specified maximum number of files is reached, older files will be overwritten.

• Maximum size (MB)

Define the maximum number of megabytes per job to be captured.

Default: 0 (unlimited)

Maximum hours to keep

Files will be deleted after the specified number of hours.

Default: 720

 Files may be deleted earlier, if the specified maximum number of files is reached.

Add job name to captured filename

Enable if you want the job name to be added to the captured filename.

The job name will be added in parenthesis to the captured filename.

Default: enabled

6.6.2.7 Spooling

Spooling is recommended if print jobs shall be re-printed. Input jobs are stored on the hard disk. Mercury Server then automatically takes care of the output of the print jobs spooled on the hard disk. Spooled jobs may manually be re-printed later.

If this setting is not active, no spooling takes place. A job is printed directly without being

buffered. The sender will be blocked until the data have been output in full.

 If Spooling is enabled [Page Notification](#) is not possible and will be disabled.


Active

Set to 'Active' to spool input jobs (i.e. to buffer these jobs on the hard disk).

Default: not active

Spool directory

Define directory where the integrated spooler shall save the print jobs and the corresponding job information.

 This directory must be placed on a drive with sufficient capacity for storage.

Retry after <x> sec, if busy

Set interval when to retry to output again on a currently busy printer.

The output of a spooled print job is delayed by this time. After this time a fresh attempt is made to output the data on the device.

Default: 15 seconds

Process at most <x> jobs concurrently

Define the maximum number of spooled print jobs that can be output at the same time at a specific device. Other print jobs are not processed until the number falls below the maximum again. Possible values range from 0 to 99. If no value is entered, the number of concurrent print jobs is not limited.

Default: 1

Allow alteration of job sequence if applicable

If disabled, the job sequence is strictly maintained. It is guaranteed that older jobs are printed first. Please note, that a job in an error state may prevent succeeding jobs from being printed.

If enabled, the system may change the job sequence. A job in an error state does not prevent succeeding jobs from being printed.

Default: enabled

Compress spool files

Enable to compress spool files.

This is recommended if disk space is sparse.

Default: disabled

Start printing even if job is not completely spooled

Enable to send print jobs to the output port as soon as the integrated spooler is collecting the data of the jobs.


If this option is disabled, print jobs are only sent to the output port as soon as they are

completely processed by the emulation.

Default: disabled

Cancel job on printer error

If an error occurs while the job is being printed, the job will be canceled. A user intervention is required. If not activated, the job will be printed again.

 This behavior may be overwritten by the settings in [Device Assignment](#)

Default: enabled

Delete active jobs after unexpected restart

If an unexpected restart of a print process occurs (for example after software crash or power failure), jobs which have been printing can be deleted at the next restart. Thus repeating crashes may be avoided, if a corrupted print job was the cause.

Default: disabled

Scheduled printing

Set time controlled job processing.

Time of printing, delay time and time restriction for processing of jobs can be defined.


see [Scheduled Printing](#)

Default: enabled

Keep spooled jobs after printing

Enable to keep spooled jobs after printing.


Here you define whether print jobs are held for security purpose after printing and how long they are held. Held print jobs can be sent to the output port again if there were problems sending them before. The amount of time the print jobs are held is defined in hours and/or minutes. If both values are set to zero, the jobs are held until they are deleted manually.

 Print jobs require storage capacity in the spool directory.

Default: disabled

Keep at the most <x> jobs

Enter the number of print jobs to be kept at the most.

 For any further job above this number, the oldest job is deleted automatically.

Leave empty for unlimited number of jobs.

Default: empty (unlimited)

6.6.2.8 Scheduled printing

Affects the time when jobs are being printed. Printing of jobs can be delayed or jobs can be printed only at a specific time.

Delayed printing

- **Do not delay printing:** Set if print jobs shall be processed normally. The output, however, can be restricted in time by setting a print period.
- **Delay printing until:** Set to delay output of print jobs to a specific date / time. All processed jobs are held until the specified date / time.
- **Delay printing for <x> sec:** Set to delay output of print jobs using a specific number of seconds. All incoming jobs are held for this time before they get processed. During this time it is possible to manipulate the job attributes and the job processing time.
- **Hold jobs (print manually/Follow2Print):** Set to hold all print jobs and not print them automatically. The print jobs are hold until they are manually selected for printing. Unprinted jobs may optionally be deleted after the specified time.

Default: Hold jobs (print manually)

Print period

Set to process print jobs all the time but only to be sent to the output port at a specified time. Then all arriving jobs will be emulated and held by the integrated spooler until the specified time is reached. The specified time period is repeating daily.


Example: You want to print on the output device only between 11:00 PM and 05:00 AM. In this case all arriving jobs are held and only sent to the output port during the specified period of day time.

Default: disabled

 [Follow2Print](#) has to be licensed.

6.6.3 Channel

6.6.3.1 Channel Output

 A channel output device must be installed on your Mercury server in order to work with this interface.

6.6.4 System Device

6.6.4.1 System Device (Output Port)

Device

This is where the required system device or the port is selected. The list contains all known system devices on the system. If the required system device is not on the list, it can be entered manually.

Queue sizes

Device-internal buffer size for receiving and sending data. If read errors occur when data are being received, it may be due to the fact that the data cannot be processed quickly enough which causes the device-internal buffer to overflow. In this case the input buffer size should be increased. The buffer size of the output is normally 0.

Serial Settings

The data for a serial connection, such as the baud rate, number of data bits, parity, number of stop bits and the handshake method can be defined here.

If the system device is not a serial device, this setting must not be activated.

RTS/CTS handshake should be used wherever possible as the handshake method.

If the XON/XOFF method is used, the XON and XOFF character should also be defined (the default values are ASCII characters 17 (XON) and 19 (XOFF)). The XON limit defines the maximum number of bytes in the input buffer before the XON character is sent. The XOFF limit defines the maximum number of bytes in the input buffer before the XOFF character is sent.

6.6.5 File

6.6.5.1 Filename (Output File Port)

Converted data can be written to a file, not only be sent to a printer.

Default Filename


The file output port writes the data into the file defined here (including path) if 'Default device' has been entered in the Logical Printer editor.


File patterns may also be entered. In this case a new, unique file name is then created for each job. Entering C:\Mercury\OUT*.BIN creates a file called OUTxxxx.BIN for each job in the directory C:\Mercury.

Job specific data, too, can be used for file name creation (see Variables)

An asterisk '*' can be used optionally after the variables. This will create unique file names. Examples:

- 'C:\OUTPUTDIR\<jobname>.PRN' will create files as XYZ.PRN, where XYZ is the job name.
- 'C:\OUTPUTDIR\<jobname>*.PRN' will create unique files as XYZ1234.PRN.

 The Mercury service runs under the local system account. Note that the local system account has no access rights on network shares.

 Which variables of the job information really contain values is depending on which Input Port and protocol is used. A LPD Input Port for example cannot give a Novell queue name, a Novell Input Port does not support a destination port, etc.

Don't overwrite existing file


If the output file already exists, it is not overwritten. The output of the data is delayed until the existing file has been deleted.

Delete existing file

If the output file already exists, it is deleted first.

Append if file exists

If the output file already exists, the data are appended to this file.

 If a file has been opened by another program or process, Mercury may not be able to open this file. To ensure that data can be written into the file and the job can continue to be processed, the external access to the file must be terminated. If a file is write-protected, Mercury will also be unable to output to this file.

6.6.6 Printers

6.6.6.1 Printers

Default Printer

The list contains all the printers known to the operating system (Windows NT/9x). The required printer can be selected or an arbitrary printer, which exists in the network, can be specified manually, to which the data are to be output.

If a printer is added in Windows it will appear in the list after the next start of the Mercury Server. If you want to use a printer which is currently not displayed in the list, you can enter the name of the printer manually.

6.6.7 SMTP

6.6.7.1 SMTP

Mercury is able to send print jobs as emails using a mail server. To do this a mail server using SMTP protocol is required and some parameters, which are defined in this dialog page:

SMTP server

The hostname or the IP address of the mail server which is to be used for sending the print jobs. If a hostname is entered here instead of an IP address, resolving via DNS must be guaranteed.

Port

The TCP port which is to be used for communication with the mail server. Usually port 25 is used, but others are possible. Please note that the required TCP port is defined by the mail server.

Security

Encryption method for the SMTP connection.

User

User name of the account, if the server needs authentication.

Password

Password of the account, if the server needs authentication.

Sender

Email address of sender using format name@xyz.com or "Friendly Name". This name will be used for display in the receiving email application.

From address

Email address of sender using format name@xyz.com. Some SMTP servers deny sending emails if not using an email address which is known by the server and valid for the login.

Receiver

The email address of the receiver to which the print job has to be sent to. Variables may be

used.

The address of the receiver can be overwritten by job information data. E.g. if a print job has been received by a POP3 Input Port and the mail body contained a definition for Destination in form of an email address, this address can be used as receiver. To get this working, please use the option 'Device as delivered by protocol' when defining the rules.

Confirmation

Optional email address where a reading confirmation should be sent to.

Default extension

Attachments will be named using the name of the print job. If the print job name does not already have a file extension, the specified extension will be added.

Force extension

The specified Default Extension will be added, even if the print job already has another extension.

Mail text...

Optional subject and/or email text can be defined here. Instead of an email text a filename can be defined. In this case the content of the file will be used as email text.

Job specific data, too, can be used in the mail text or subject. The following fields can be used as variables:

<jobname>	Name of the job
<jobnamenoext>	Name of the job (if filename without file extension)
<srcaddress>	Address of senders as IP address or Novell queue name
<username>	Name of user
<destport>	Destination port as queue name from LPD protocol
<destaddress>	Specific device if defined in Logical Printer
<jobid>	Job-ID
<ticks>	Milliseconds since system start
<date>	Current local date
<time>	Current local time
<creationtime>	Start time of job
<starttime>	Emulation time of job
<pages>	Number of pages

6.6.8 TCP/IP

6.6.8.1 Address (TCP/IP Output)

Local Address

In case the Mercury system is using multiple IP addresses, enter here one IP address of the computer that is to be used for communication. If the computer has only one local address, the field may remain empty.

Port

TCP/IP port, to which the data are to be sent.

Default Remote Address

IP address or DNS name of the printer or host computer to which the data are to be sent.

⚠ If a Local TCP Port is used, the Default Remote Address may not be changed and is always 127.0.0.1 (localhost). The data will be encrypted and may only be sent to a Mercury Local TCP Input Port.

Connect Timeout

Defines the maximum of time in seconds, for which Mercury tries to establish a connection to a remote host / printer.

Close Timeout

Defines the maximum of time in seconds waiting to close a connection to a remote host / printer gracefully. If empty, the connection will be closed without waiting for confirmation by the remote host.

Linger Timeout

Defines the maximum of time in seconds, for which a TCP session may try to send remaining data before closing connection.

If no value is entered here, the connection is not closed before all data was sent to the receiver. The session will then be closed normally using a FIN packet.

If a value is entered here, the connection will be closed as soon as the defined time has passed. If not all data could be sent, the connection gets closed using a RST packet and the receiver won't get the remaining data. If all data could be sent during the defined time, the connection will be closed using a FIN packet as normally.

Do not connect to a device which is in an error state

Usually there will be no connection to a device that is reporting a SNMP error state. By deactivating this option an attempt to connect to a device will be made regardless of its error state.

⚠ To check the SNMP error state of a device requires an enabled SNMP device monitoring in the [Job Monitoring](#) settings.

Do not connect to a device after connect error for <x> seconds

If a connection to a device fails, no further connection attempts for that device will be done for the specified amount of time. If defined, an [alternative device](#) will be used.

6.6.8.2 Protocol (TCP/IP)

[LPD/LPR](#) protocol

6.6.8.3 LPR Output

Using this protocol option allows sending print jobs to printer or host which supports LPD (Line Printer Daemon) functionality.

Queue

Name of the LPR queue offered by the remote print server. This name must be supported by the remote print server. If jobs are sent to a printer which supports one queue only or if the queue doesn't matter, the name 'AUTO' can be used. Leave it blank to use the queue which was delivered by the input protocol. Variables may also be used.

Timeout

Timeout in milliseconds the output port is waiting for replies of the remote print server.

Print banner page

If enabled, a banner page is printed before each job. Not all remote print servers support this feature.

Set job name as title

If enabled, the title / name of the print job is set to job name which was used by Mercury. The remote print server then will use this job name, too.

Default: On

Send control file before data

If enabled, the LPR control file which contains information about the job, such as job name, job size, etc., is sent to the remote printer server before job data.

Default: On

Filesize 0 supported

The printer needs not to be informed about the size of the job. The printing may start, even if the total size of the job is not known yet.

Default: On

Override filesize

The printer has to be informed about the size of the job before the data is sent. If the total size is not known at the beginning, because the host is still sending data, some printers accept a filesize of 0. If your printer does not accept a filesize of 0, but you don't want to spool the complete job before printing it, you can try to enable this option. If enabled, a filesize of about 2 GB is assumed. Some printers will accept this, even if the final job size will be smaller.

Default: Off

Filesize needed

The printer has to be informed about the size of the job before the data is sent. The printing will not start before the total size of the job is known.

Default: Off

6.6.9 IPP Client

6.6.9.1 IPP Output

Use the IPP Output port to establish a bi-directional connection to an IPP server or a printer which is capable of the IP-Protocol.

This port does support TLS encrypted connections.

If Job Monitoring is activated, the job status is repeatedly requested, until the remote printer/IPP server returns the job completed or job canceled/aborted status.

If supplied by the remote system, the number of printed pages will be reported to the Mercury Jobinfo during the job monitoring process.

The port use the following URI format for establishing the connection:
Scheme://remoteaddress:TCPport/resource
Please consider the IPP server/printer manual for the correct URI settings.

URI Scheme

Select the correct URI scheme for the connection. If the ipp or https scheme is selected, a TLS encrypted connection will be triggered.

Remote address

Enter the correct FQDNS name of the remote system. If no encrypted connection is required also the IP address can be used.

Verify remote certificate

During the TLS handshake the remote system send the remote (Server) certificate to the client. The client should verify this certificate.
To do this, this certificate or the corresponding CA certificate have to be installed to the appropriate Windows certificate stores prior starting the connection to the remote system. If you don't have the certificate or CA certificate of the remote system, you can disable the remote certificate verification by deactivating this option.
The connection then still will be encrypted.
But please keep in mind, that, if you connect to a remote system without the Verify remote certificate option enabled, the trustability of the remote system will not be checked !

TCP port

TCP/IP port, to which the data are to be sent.
For unencrypted connections the default port for IPP connections is 631. For TLS/SSL encrypted connections the most printers use port 443.
Please consider the printer manual/configuration which TCP/IP port to use.

Connect timeout

Defines the maximum of time in seconds, for which Mercury tries to establish a connection to a remote host / printer.

URI Resource

For almost all printers the default URI resource setting "ipp/print" can be used. Only some printers require some special format of this URI part.
Please consider the printer manual, if some special resource setting is needed.
Optionally some printers support an additional queue name, which can be appended to the default resource string.
The Mercury IPP input port e.g. can handle resource strings with the format ipp/print/printername. The printername part of the URI will be copied to the destination port parameter of the Mercury JobInfo.

Client certificate

If you leave this input field empty, Windows will use the default client authentication credentials to establish the encrypted connection with the remote system.
Alternatively you can specify a certificate name, which will be used to find a certificate for identifying the client system to the remote system.

IPV6 support

Disable this option, if the connection should not support the IPV6 address format.

Use default document format


If you send a print job to the remote system and the type of the print job data is known (e.g. Postscript or PCL), this information will be forwarded to the remote system. But due to the fact, that some printers then claims to not support this format, even if they obviously do so, it maybe necessary to use the ipp default document format "application/octet-stream".

Using this format will trigger the format autodetecting feature of the printer.

If a printer refuses to print a job with returning a "mime media type not supported" error, activate this option.

Do not connect to a device which is in an error state

Usually there will be no connection to a device that is reporting a SNMP error state. By deactivating this option an attempt to connect to a device will be made regardless of its error state.

 To check the SNMP error state of a device requires an enabled SNMP device monitoring in the [Job Monitoring](#) settings.

6.6.10 Fiery

6.6.10.1 Fiery Output

see [Fiery Properties](#)

Index

A

- Access levels 17
- Accounting 26
- Accounting Server 25
- Activation 11
- Active Jobs 33
- Adapter 61, 62
- Add Logical Printer 29
- Address 19
- Administrator 21
- Advanced Reprint 34
- AFP to IPDS 138
 - Common 138
 - Line Data 139
 - Overview 138
 - Resources 141
- AFP to PCL 142
 - Overview 142
- AirPrint Server 46, 48
- Alternative Devices 368
- AOS Trigger 154
- Atlas 130
 - Appendix 130
 - Common 122
 - Performance 126
 - Resources 124
- Attachment 53

B

- Backup configuration 37
- Backup log file 22
- Banner page 379
- Barcode Filter 308
 - Barcode Filter and Field Scanner 313
 - Barcode SIMM Emulation 309
- Barcode Label Software 92
- Basic information 28, 35
- BC-XOM 67, 69, 73, 74
- Bind to IP address 65
- By page count 368

C

- Cancel job 34, 36
- Capture Files 22
- Capturing 56, 372
- Card reader 24, 375
- Certificate 21
- Chained Emulations 92
- Channel Emulation 93
 - Band ID 95
 - Bold Printing 98
 - Control Codes 96
 - Download 99
 - FCB 100
 - Job End Detection 101
 - Job Management 101
 - Performance 103
 - Printer Type 93
 - Translation Table 96
- Channel Input 41, 43
- Channel Output 375
- Client 18
- Code Page 146
- Codepage Filter 317
 - Application 317
 - Configuration 318
 - Configuration ,Patch Table 318
 - NT Language Pack 317
- Color support 157
- Common print process settings 27
- Comparex 105
- Comparex Line Mode to IPDS/PCL5 105
 - Common 105
 - Job Management 108
 - Overview 105
 - Resources 106
- Components 13
 - The Simply Application 13
- Concurrent jobs 27
- Conditional Stringchange Filter 319, 320, 321
 - Configuration 319, 320, 321
 - Replacement 321
 - Trigger 320
- Configuration 37, 40
 - Backup 37
 - Introduction 40
 - Restore 37

Configuration Dialog 40
 Connecting to a server 17
 Connectivity 19
 Conversion 32
 Run 32
 Start 32
 Stop 32
 CPCA Remover Filter 308, 321
 Configuration 322
 Customer 25
 Cycle 368

D

Data Capturing 56, 372
 Data conversion 29
 Data Encryption 371
 Data Type 57
 Database 19
 Dealer 25
 Debug information 28
 Default remote address 378
 Delay job output 27
 Delayed printing 375
 Delete job 34, 36
 Device Assignment 368
 Direct Socket 65, 378
 DNS 19
 DNS name 17, 18

E

EASYWARE import files 335
 Edit Logical Printer 29, 30
 E-Mail 53
 Email notification 23
 EML format 53
 Emulation 14, 90, 92
 AFP to IPDS 138
 AFP to PCL 142
 Atlas™ 120
 Chained Emulations 92
 Channel Emulation 93
 Comparex Line Mode to IPDS/PCL5 105
 Filter 14
 FOL 109
 IGP to PCL 143

IPDS to PCL5 146
 IPDS to PDF 145
 PCL to PostScript® 194
 PCL5 to PDF 186
 PCL5 to X-Metacode 201
 Performance 90
 Prescribe to PCL5 212
 Program Interface 219
 Raw to Raw 222
 Resources 92
 Theory 14
 VIPP™ 224
 XES to PCL5 280
 X-Metacode to PCL5 300

Encryption 23, 377
 Error 35
 Error messages 28, 35
 ESCON 44
 Event 35
 Date/time 35
 Debug 35
 Error 35
 From 35
 Information 35
 List 35
 Log levels 35
 No. 35
 Warning 35
 Event log files 22
 Event logging 28, 35
 Events 28
 Basic information 28
 Debug information for troubleshooting 28
 Extended information 28
 Logging 28
 Warning notes 28
 Extended information 28, 35

F

Field Scanner Filter 322
 ASCII Formatted With Fix Column Width 332
 Configuration 322, 325, 327, 329, 331, 332, 333, 334, 335, 336
 CSV Formatted ASCII Data 331
 PCL Job Scanned For Text Areas 329
 Reporting To Later Components 333
 Saving To ODBC Database 333, 334

- Field Scanner Filter 322
 - Triggered PCL Job 327
 - Triggered Raw Byte Stream 325
 - Writing EASYWARE Import Files 335, 336
 - Fiery Port 366
 - Disconnect after job transmission 366
 - IP Address 366
 - Logical Printer 366
 - Password 366
 - Timeout 366
 - Username 366
 - File Menu 35
 - Login 35
 - Quit 35
 - Start 35
 - Stop 35
 - File Name 45, 376
 - File Port 376
 - File Name 376
 - Filter 14, 308, 359, 361, 362, 364, 365
 - Barcode Filter 308
 - Codepage Filter 317
 - Conditional Stringchange Filter 319
 - CPCA Remover Filter 321
 - Field Scanner Filter 322
 - Job Copier Filter 336
 - Job Separator Filter 337
 - Overlay Manager 359, 361, 362, 364, 365
 - Overview 308
 - Post Emulation filter , 14
 - Pre Emulation filter 14
 - Pre-/Post-Job Filter 341
 - Recordchange Filter 341
 - Regular Expressions Filter 353
 - Stringchange Filter 354
 - Textextractor Filter 356
 - Finisher 154
 - Fleet Management Statistics 25
 - FOL 109
 - Edit Default Font 117
 - Fonts 116
 - Page Properties 115
 - Paper 112
 - Paper Assignment 114
 - Performance 118
 - Printer Capabilities 110
 - Resources 120
 - Follow2Print 24, 375
 - Font Algorithm 163
 - FSM Server 25
- ## G
- General server settings 19
 - GOCA raster images 173
 - Group boundary 167
 - Guest 21
- ## H
- Help Menu 38
 - Index 38
 - Info 38
 - Hold jobs 375
- ## I
- Icon Bar 38
 - IGP to PCL 143
 - IGP 143
 - Overview 143
 - Resources 145
 - Index Server 24
 - Input 40
 - Input filter 29
 - Input interface 29
 - Input Port 14, 40, 56, 58, 59
 - Add job name to monitor filename 56
 - AirPrint 46
 - Channel [ISA] 41
 - Channel [PCI] 42, 43
 - Data 56
 - Data Capturing 56
 - Data Type 57
 - ESCON 44
 - File 44, 45
 - IPP 45
 - IPP/IPPS Server 46, 48
 - Job Timeout 58
 - Look for session every <x> msec 58
 - LPD 49
 - Maximum files to monitor 56
 - Maximum number of input sessions 58
 - Maximum number of pending jobs 58
 - Novell 50
 - Novell Port 50
 - ODBC 51
-

- Input Port 14, 40, 56, 58, 59
 - Parallel Input 52
 - POP3 52, 53
 - PPD 55
 - Protocol 14
 - Session Timeout 58
 - Sessions 58
 - Sessions ,Job Start Sound 58
 - Settings 59
 - SNA LU 6.2 60
 - SNA LU1/LU3 62
 - SNA Port 62
 - SNA Port LU 6.2 61
 - Spooling 14
 - System Device 63
 - TCP/IP 65
 - Theory 14
 - Transport 14
 - Twain 66
- Input Trays 160
- Installation 11
- Introduction 10
- IOCA FS 146
- IP address 17, 18
- IPDS 146
- IPDS Overlay Supplement 177, 178, 179, 180, 181, 184, 185
 - Commands 179
 - Definitions 180
 - Examples 184, 185
 - IPDSOVER.INI 178, 179, 180, 181
 - Overlay Resources 178
 - Overview 177
 - Path 178
 - Rules 181
 - Sheet and page counter 179
 - Simplex and duplex printing 179
- IPDS to PCL5 146
 - 240 DPI IM1 Images 173
 - Accessible 160
 - Add form name to job name 167
 - Add MVS job class to job name 167
 - Analyze Mode 165
 - Bar Code Adjustment 157
 - Bit Images PCL Compressed 157
 - Check VPA 173
 - Conversion 163
 - Cut Length 167
 - Cut Offset 167
 - Cut Sheet Emulation 146
 - Debug Level 163
 - Early AOS Trigger 154
 - Edit Resource Replacement 165
 - Emulation Loading 165
 - Enlarge GOCA raster images 173
 - Extract job name from data stream 167
 - Extract job name from Group Boundary command 167
 - Face down printing 154
 - Finisher options 154
 - Font Algorithm 163
 - Hole Puncher 154
 - Host Communication 165
 - ID number 146
 - Input Tray 160
 - Input Trays 160
 - IOCA FS 146
 - IPDS Printer 146
 - IPDS Printer ,Code Page 146
 - IPDS Tray 160
 - Job Management 167
 - Job Management ,Split at output bin change 167
 - Job Management ,Wait for page confirmation 167
 - Job separation and finisher commands 154
 - Media Attributes 160
 - Media Type 154
 - Miscellaneous 173, 218
 - Model number 146
 - Normal Printer Restart after job 165
 - Normal Printer Restart before job 165
 - Outline Font DPI 163
 - Output Bin Mapping 146
 - Overview 146
 - Page order 154
 - Paper Size 160
 - Paper Size ,Height 160
 - Paper Size ,Ignore Page Offset 160
 - Paper Size ,Rotate 160
 - Paper Size ,Wide carriage/LEF 160
 - Paper Type 146
 - PCL Color Support 157
 - PCL Full Page 163
 - PCL Output 157
 - Plex mode 146
 - Print Direction Offsets 153
 - Printer Control 154
 - Printer Type 146

- IPDS to PCL5 146
 - Resource Path 165
 - Resources 165
 - Scaling 153
 - Send IPDS variable spaces as PCL positioning command 154
 - Simulate printer offline 165
 - Smoothing 163
 - Stack Output with offset 154
 - SVI Threshold 163
 - Tray 153
 - Tray Alignment 153
 - Trigger 167
 - Unit 153, 160
 - Whole Page Offsets 153
 - Width 160
- IPDS to PDF 145
- IPP Input 45
- IPP/IPPS 46
- IPP/IPPS Client 380
- IPP/IPPS Server 46, 48

J

- Job 32, 34
 - Advanced Reprint 34
 - Cancel 34
 - Converted 32
 - Created 32
 - Delete 34
 - Keep 34
 - List 32
 - Name 32
 - Pause 34
 - Printed 32
 - Properties 34
 - Reprint 34
 - Size 32
 - Status 32
 - Undo keep 34
 - Undo pause 34
- Job authentication 24
- Job Copier Filter 336
 - Configuration 336
- Job End Detection 101
- Job list 32
 - Active Jobs 33
 - Printed Jobs 33
- Job Management 101, 167

- Job Menu 36
 - Cancel 36
 - Delete 36
 - Keep 36
 - Pause 36
 - Print 36
 - Printed 36
- Job monitoring 32, 369
- Job Name 167
- Job Notification 369
- Job separation 154, 167
- Job Separator Filter 337
 - Banner Pages 339
 - Configuration 337, 339, 340
 - Job Separation 337
 - Miscellaneous 340
- Job timeout 58

K

- Keep job 34, 36

L

- Licensing 11
- Load Balancing 368
- Local address 378
- Log event 28, 35
- Log levels 28
- Logfile size 22
- Logging 28, 35
- Logging in 16, 17, 18, 35
- Logical Page Processing 157
- Logical Printers 16, 29, 30, 32
 - Add 29, 30
 - Data conversion 29
 - Description... 30
 - Edit 29, 30
 - Input filter 29
 - Input interface 29
 - Output filter 29
 - Output interface 29
 - Properties 30
 - Remove 29, 30
 - Setup and configuration 29
 - Testing 32
 - Theory 15
- LPD Input 49

LPR Port 379
Print banner page 379
Queue 379
Send control file before data 379
Set job name as title 379
Timeout 379

M

MAC Address 61, 62
Mail on error 23
Menus 35, 36, 37, 38
File Menu 35
Help Menu 38
Job Menu 36
Setup Menu 37
Mercury Server ID 25
MIME format 53

N

Notification on error 23
Novell Port 50
NT Language Pack 317

O

ODBC Database 333
ODBC Input Port 51
Operator 21
Operator Permissions 21
Outline Fonts 163
Output 365
Output Bin Mapping 146
Output filter 29
Output interface 29
Output Port 15, 365, 366, 367, 368, 369, 371, 372
Capturing 372
Channel 375
Data 371
Data ,Data Encryption 371
Data ,Job information within data stream 371
Data ,Pack and encrypt data 371
Data ,Password 371
Data Capturing 372
Device Assignment 368

Device Assignment ,Add 368
Device Assignment ,Delete 368
Device Assignment ,Edit 368
Device Assignment ,Move down 368
Device Assignment ,Move up 368
Fiery 366
Fiery Output 382
File Port 376
IPP/IPPS Client 380
Job Monitoring 369
Printer 377
Protocol 15
Settings 367
Settings ,Buffer 367
Settings ,Delay if data received 367
Settings ,Delay if no data received 367
Settings ,Execute command after printing 367
Settings ,Maximum size of receive buffer 367
Settings ,Read Block Size 367
Settings ,Read Timeout 367
Settings ,Timeout 367
Settings ,Write Timeout 367
SMTP 377
Spooling 15, 372
System Device 375
TCP/IP 378
Theory 15
Transport 15

Overlay 236, 237, 240, 244, 245, 247, 250
Overlay Manager 359
ASCII / PCL 359
ASCII / PCL ,Plain Text Format (ASCII) 359
ASCII / PCL ,Printer Command Language (PCL) 359
Configuration 359, 361, 362, 364, 365
Fonts 365
General 361
Overlays 362
Pages without overlay 364
Profile 359

P

Page confirmation 167
Page Notification 369
Page Offsets 153, 160
Paper Size 160

- Paper Type 146
- Parallel Input 52
- Password 17
 - Administrator 17
 - Guest 17
 - Operator 17
- Passwords 21
- Pause job 34, 36
- PCL Output 157
- PCL to PostScript® 198
 - Finishing 198
 - Input Tray Mapping 195
 - Job Settings 196
 - Layout 197
 - Output Bin Mapping 196
 - PCL Defaults 199
 - PostScript® Printer 194
 - Resources 199
- PCL5 to PDF 186
 - Compression Level 186
 - Duplex Mode 190
 - Encryption 189
 - Face down printing 188
 - Image Mask Method 186
 - Include all resources in each split 186
 - Line Termination 190
 - Orientation 190
 - Output Destination 188, 190
 - Page Copies 190
 - Page order 188
 - Paper Size 190
 - PCL Defaults 190
 - PDF Common 186
 - PDF Miscellaneous 188
 - PDF Security 189
 - PDF Signatures 190
 - Resources 192
 - Signatures 190
 - Split Jobs 186
 - Stapling 188
- PCL5 to X-Metacode 201
 - Color 210
 - Default Font 205
 - Emulation 202
 - Performance 208
 - Resources 204
 - Trays 207
 - Xerox Printer 206
- Permissions 21
- PJL 39
- Plex mode 146
- POP3 53
- Port 19, 378
- PPD Input 55
- Pre-/Post-Job Filter 341
 - Configuration 341
- Prescribe 212
- Prescribe to PCL5 212
 - Overview 212
 - Page Offsets 217
 - PCL Features 218
 - Prescribe Defaults (FRPO) 212
 - Resources 218
- Principles 13
 - Components 13
 - Specimen Application 15
 - Theory 13
- Print direction 153
- Print job 36
- Print period 375
- Print process 27, 28
 - Basic information 28
 - Common settings 27
 - Concurrent jobs 27
 - Debug information for troubleshooting 28
 - Delay job output 27
 - Error messages 28
 - Event logging 28
 - Extended information 28
 - Log event 28
 - Log levels 28
 - Name 27
 - Setup and configuration 27
 - Warning notes 28
- Print Process Bar 38
- Printed Jobs 33
- Printer 377
- Printer Control 154
- Printer Pool 368
- Printer Type 146
- Processes 15
 - Print Process 15
 - Threads 15
- Program Components 16
 - Logical Printers 16
 - System Service 16
- Program Interface 219

Program Interface 219
Miscellaneous 219
Performance 221
Protocol 379
Input Port 14
Output Port 15
TCP/IP Port 379

Q

Quit 35

R

Raw to Raw 222
Common Behavior 222
Performance 223
Recordchange Filter 341
Configuration 342
Regular Expressions Filter 353
Configuration 353, 354
Definition of Replacements 354
Remove Logical Printer 29
Reply mail on error 53
Reprint 34
Requirements 10
Resources 92, 165, 192
Restore configuration 37
Restrictions 21
Retrieve printers from FSM Server 25
Round-robin printing 368
Run conversion 32

S

SAP 67, 69, 73, 74
Scaling 153
Scheduled printing 372, 375
Security 377
Serial Settings 63, 375
Server 16, 17, 19, 21, 23, 24, 25, 26
Accounting 26
Address 19
Advanced Settings 24
Automatic start of Logical Printers 19
Capture files 22
Configuration 19
connecting to 17

Connectivity 19
Database 19
DNS available 19
Email Notification 23
Follow2Print 24
FSM Server 25
General 19
list 17
log files 22
Logical Printers 19
Mail on error 23
Notification 23
Passwords 21
Port 19
Properties 19
Security 21
TCP/IP port 19
Timeout 19
Sessions 58
Settings 367
Setup Menu 37
Add logical printer 37
Backup 37
Edit logical printer 37
Print process 37
Remove logical printer 37
Server 37
Shorten log file 22
Size of event log files 22
SMTP 23, 377
SMTP Port 377
Confirmation 377
Default extension 377
Force extension 377
From address 377
Receiver 377
Security 377
SMTP Port 377
SMTP Server 377
Username 377
SNA Port LU 6.2 61
Adapter 61
Connect Timeout 61
IDBLK 61
IDNUM 61
MAC Adresse 61
Remote Control Point 61
Remote Network Identifier 61
Retry connect after 61

SNA Port LU 6.2 61
 SNA Connection 61
 Strip 3270 orders 61
SNA Port LU1/LU3 62
 Adapter 62
 Connect Timeout 62
 IDBLK 62
 IDNUM 62
 MAC Adresse 62
 Retry connect after 62
 SNA Connection 62
 Strip 3270 orders 62
SNMP 39
Specimen Application 15
Spooling 372, 375
 Input Port 14
 Output Port 15
 Scheduled Printing 375
SSL 21
SSL/TLS 23, 377
Start 32, 35
Starting 16
Statistics 25
Stop 32, 35
Stringchange Filter 354
 Configuration 355, 356
 Definition of Replacements 356
System Device 52, 63, 375
System password 25
System Service 16

T

TCP/IP Port 65, 378
 Bind to IP address 65
 Buffer Size 65
 Default Remote Address 378
 Input Buffer Size 378
 Linger Timeout 378
 Local Address 378
 Port 65, 378
 Protocol 379
 Protocol ,LPR 379
Test Logical Printer 32
Textextractor Filter 356
 Application 356
 Configuration 356
 Delimiter 357

 Optimize 356
Textscanner 253
Theory 13
Threads 15
 Performance 15
Timeout 19, 58
TLS 23, 377
TLS/SSL Settings 21
Transfer Settings 25
Tray Alignment 153
Trays 153, 160
Twain Port 66

U

Unprinted jobs limit 27

V

variables 253
View job 36
VIPP™ 267
 Additional Fonts 267
 Appendix 274
 Common 226
 Default properties and predefined system parameters 271
 Encoding vectors 269
 Finding and Preparing Resources 264
 Fontmapping table 275
 Global resource path 230
 Handling of ppd files with non standard keywords 276
 Input media mapping 231
 Insert additional resources 270
 Jog mapping 233
 Output bin mapping 232
 Output driver settings 259
 Performance 259
 PostScript® interpreter defaults 233
 Replacement of Ghostscript standard fonts 273
 Resources 229
 Standard Fonts 267
 Staple mapping 232
 VIPP Resources 263

W

Warning notes 28, 35
Watch Print Devices 39
Web administration 24
Wide carriage/LEF 160

X

XES to PCL5 280
 Commands 280, 281
 Common Behavior 299
 Common Page Settings 282
 Duplex Settings 286
 Emulation 294
 Not supported commands 281
 Number Of Copies 286
 Offsets 282
 Page Layout 287
 Partial supported commands 281
 PCL5 Printer Type 288
 Performance 292
 Resources 296
 Scaling 285
 Supported commands 280
 Tray Mapping 290
X-Metacode to PCL5 300
 Category 306
 Collating Multiple Copies 300
 Color 307
 Copies 300
 Destination Printer Feature 300
 Edge to Edge 300
 Feed (input tray select assignments) 302
 Finisher Options 306
 Form 300
 Global 305
 JDE/JOB 300
 JDL 300
 JSL Column Width 302
 JSL File Name 302
 Messages 306
 Miscellaneous 304
 Paper Size 302
 Process Mode 300
 Resource Path 305
 Resources 305
 Resource Script File Name 305

Run emulation in static mode 305
Split jobs on report boundary 302
Start Command 300
Suppress empty logical skip to channel 1
 moves 304
System Generation 302
Target 305
Treat constant strings without qualifier 302
Verbose Level 306
Y/X Offset (in 1/300 inch) 300
XOM 67, 69, 73, 74, 85, 86, 88, 89

Z

Zip log files 22