.print Client Linux

for TCP/IP (.print version 6.2)

Manual

ThinPrint AG
Alt-Moabit 91 a/b
10559 Berlin
Germany/Alemania

ThinPrint Pty. Ltd.
L 10, 275 Alfred Street
North Sydney/NSW/2060
Australia

ThinPrint, Inc.
20006 Detroit Road, Suite 303
Cleveland, OH 44116
USA/EEUU

ThinPrint, Inc.
7600 Grandview Avenue, Suite 200
Denver, Colorado 80002
USA/EEUU

E-mail: info@thinprint.com
Web: www.thinprint.com
Issued: August 12, 2009 (v41)

© Copyright

® Registered trade marks

Safety warning

All ThinPrint products are pure software solutions. Please note the safety warnings in the technical documentation from your hardware vendor and from the manufacturer of each device and component. Before beginning installation, we recommend closing all windows and applications and deactivating any virus scanner.

How to use this manual

The following conventions are used throughout this manual to represent recurring actions and text types.

Conventions

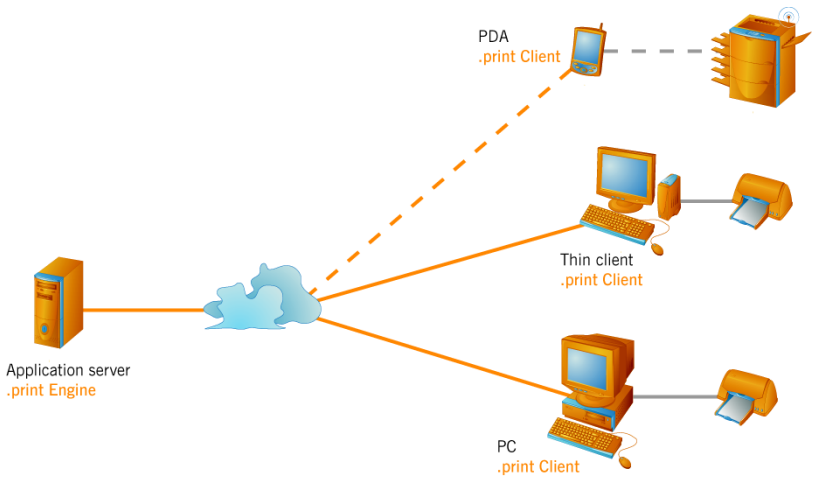
Note!	Important tip, explanation, exception
SMALL CAPS	Menu, command, dialog panel, option
Italic “Name”	Proper name, emphasis, variable
`Courier`	Keyboard input
→	Consecutive menu or command
■	Enumeration, head note
Example	Example
– 1. 2.	Procedural steps
Page 5 www.thinprint.com	Links

Introduction

What is ThinPrint .print?

Overview

ThinPrint .print is a software solution and consists of a server and a client component (Illus. 1).



Illus. 1 Using of ThinPrint .print with server and client components

Server component

The **.print Engine** is the actual core of the ThinPrint .print framework. It provides complete printer driver management. The .print Engine performs the following main functions:

- Bandwidth control for print data
- Encryption for print data
- Compression and streaming[1] for print data

Information on installation and configuration of the server components including sample configuration can be found in the user manual for the specific .print Engine (Page 26).

Client component

A software component on the client side, **print Client** is generally responsible for receiving print data, decompressing and decrypting it, and sending it to the printer(s). These print devices can either be installed on the local port (e.g., *lp1, lp2*) or addressed remotely over LPR/LPD or CUPS.

Many .print Clients are available for different end devices and areas of deployment: for Linux, all Windows versions incl. Windows CE and Windows Mobile, for Dos, Win OS/2, and Java, for BlackBerry and Symbian as well as for internal and external print servers for network printers.

1 print data sent in a stream of 32 KB packets

.print Client Linux supports the following (Windows) server components:

- .print AutoConnect
- .print Connection Service

Installation

Requirements

Before installing .print Client Linux, make sure your Linux machine is **network capable**.

CUPS and/or LPR/LPD

If you want to use CUPS, it needs to be installed on your Linux machine as well. If you would like to forward print data from the .print Client to printers via CUPS, the printers must be configured in CUPS before installation of .print Client Linux.

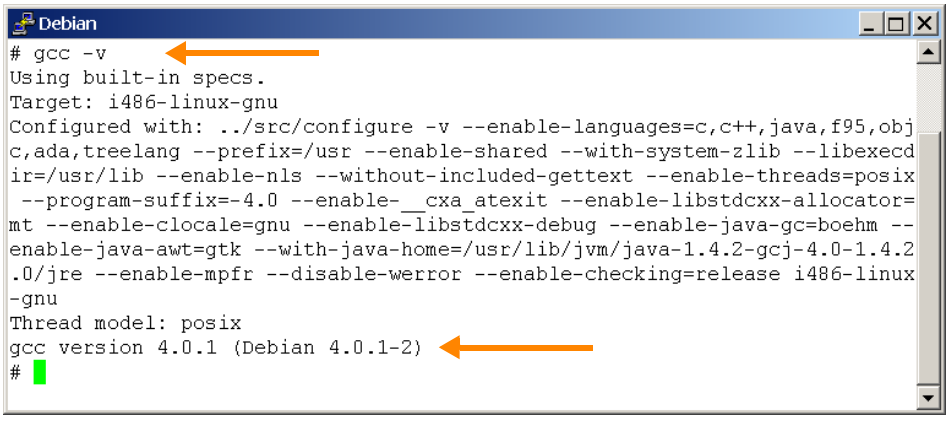
It is not necessary to configure LPR/LPD because .print Client Linux itself supports the LPR protocol per RFC 1179. For more information, see also “LPR/LPD” and “CUPS”, beginning on Page 22.

gcc versions

.print Client Linux can be used under the following gcc versions:

- 2.95.4
- 3.3.5 and higher

You can see the gcc version of your Linux system by entering the `gcc -v` command (Illus. 2).



Illus. 2 Finding the gcc version of the Linux system

Operating conditions

The following permissions should be granted on the Linux machine:

Location	Permission
Config file /etc/**thinprint.rc** (Page 18)	• Write permission for *root* • Read permission for user
.print Client program directory	• Read and execute permission for user • Full control for admins
.print Client program file **thnucInt**	• Read and execute permission for user

Administrator permissions

Administrator permissions are required for all installation and configuration procedures. Therefore, log on as **root**.

Safety warning

All ThinPrint products are pure software solutions. For safety warnings for your hardware, please consult the technical documentation provided by the respective manufacturer of each hardware device and component.

Server preparations

.print Application Server Engine

Windows terminal servers

A plug-and-play installation runs the first time .print Engine is installed on a Windows terminal server, so that **.print Clients Windows** can print without any further configuration. For **.print Clients Linux,** though, a few settings must be made on the server after .print Engine installation.

Templates for .print AutoConnect

Templates are printer objects in the Windows server's PRINTERS AND FAXES folder from which terminal session printers inherit their properties (e.g., bandwidth, network protocol, and printer driver).

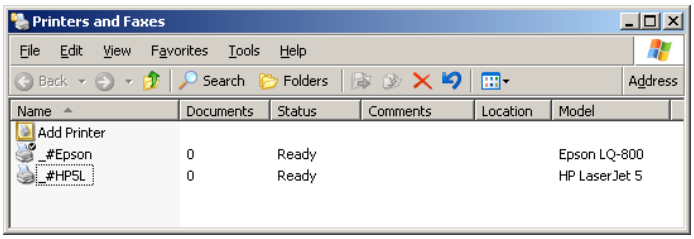
_#ThinPrint Output Gateway printer template appears in the server's PRINTERS AND FAXES folder after plug-and-play installation. This template cannot be used for .print Clients Linux because a Windows specific printer driver is in use (ThinPrint Output Gateway). Instead, other printer drivers[2] must be installed on the server for the print devices found on the client side.

2 Native printer drivers instead of ThinPrint Output Gateway

– Open the PRINTERS AND FAXES folder on the Windows server and install the necessary templates, either
  • A separate template for each type of client printer and therefore a separate printer driver, or
  • A single template for one group of client printers with a universal driver

All of these printer templates must be connected to a ThinPrint port for TCP/IP. An example of one template for all HP compatible print devices and one for all Epson compatibles is shown in Illus. 3.

– Turn the printers into templates during installation by placing **_#** in front of each printer name, deleting the permissions for all users except ADMINISTRATOR, and adding SYSTEM with FULL ACCESS.



Illus. 3 New templates on a Windows terminal server (example)

.print AutoConnect

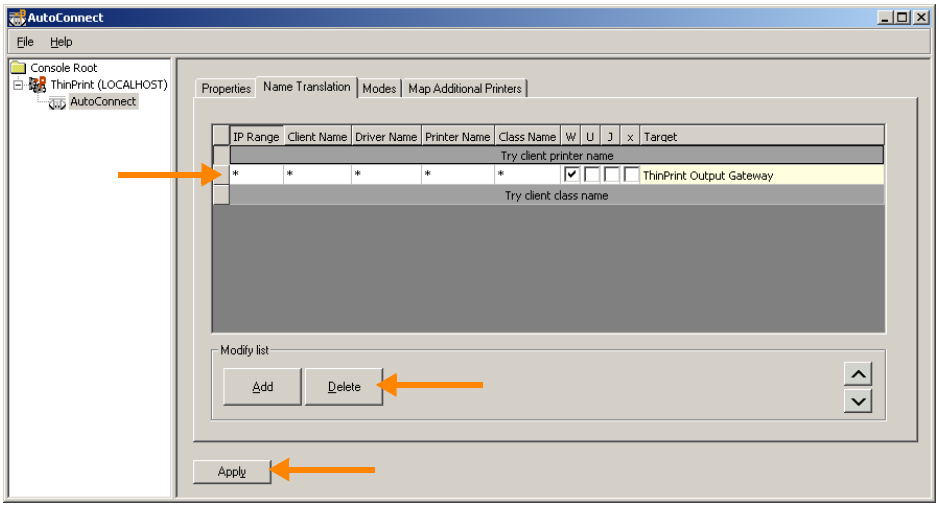
The **.print AutoConnect** component is also installed during .print Engine installation on a windows terminal server. .print AutoConnect automatically creates all client printers in a terminal session.

During installation the following entries are also automatically made in the AutoConnect name translation table (Illus. 4):

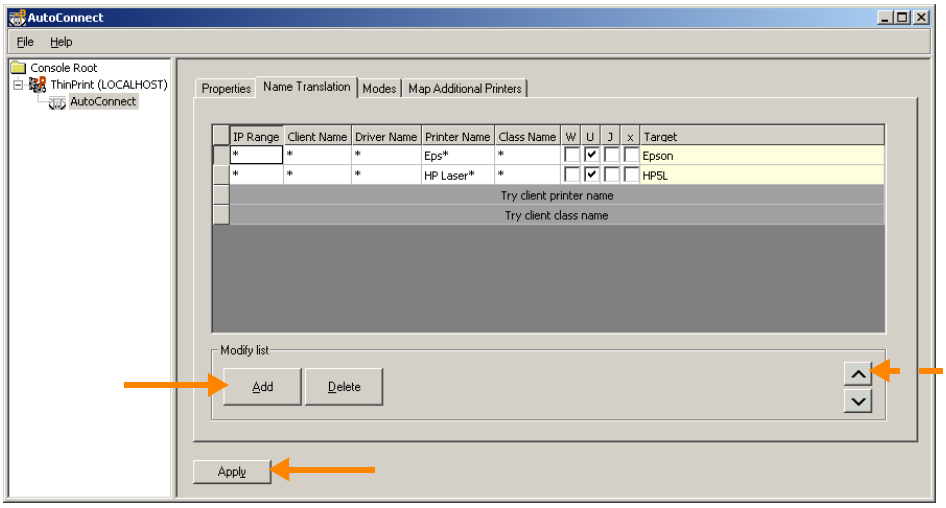
*	ThinPrint Output Gateway

– This entry has to be deleted, because ThinPrint Output Gateway cannot be used under Linux.
– Instead, you can define a separate name translation for each printer type (Illus. 5); e.g.:

Eps*	Epson	(for all Epson printers
HP Laser*	HP5L	(for all black&white laser printers from HP)



Illus. 4 Delete “ThinPrint Output Gateway” from the name translation table



Illus. 5 Two new entries in the name translation table

Without AutoConnect

The client printer can also be created on the server manually. No templates are required. The printers are also connected to ThinPrint TCP/IP ports. Their descriptions must have the following syntax:

Either: *printer#client_name:printer_ID*
Example: HP LaserJet 5#linuxclient:3
Or: *printer#IP_address:printer_ID*
Example: HP LaserJet 5#191.168.1.17:3
Or: *client_name:printer_ID#printer*
Example: linuxclient:3#HP LaserJet 5
Or: *IP_address:printer_ID#printer*
Example: 191.168.1.17:3#HP LaserJet 5
The *Printer ID* is assigned by the .print Client Linux (see examples 1 to 3, Page 14).

For more information, please see also Page 22 and the “.print Application Server Engine” user manual (Page 26).

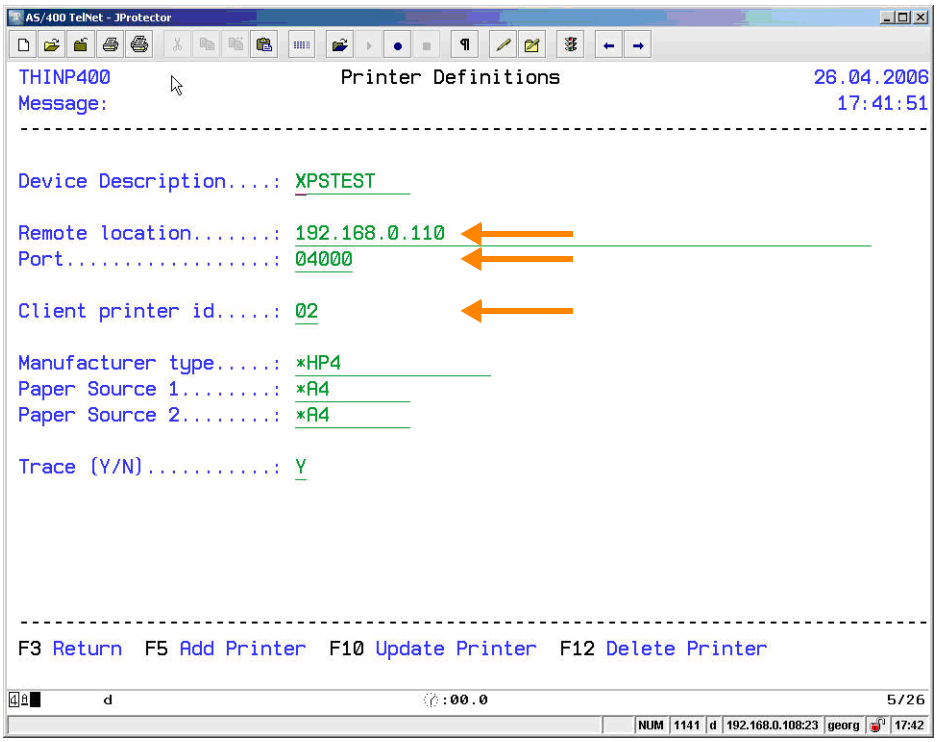
Other Windows servers

The same applies to remote desktop connections to Windows XP computers as for terminal servers under Windows (see above). Further information is also found in the ".print Desktop Engine" user manual (Page 26).

You can also configure all other Windows computers without Terminal Services – for example, central dedicated print servers – exactly like Windows terminal servers without .print AutoConnect (see above). For more information, please see also Page 22 and the ".print Server Engine" and ".print Desktop Engine" user manuals (Page 26).

AS/400 hosts

On an AS/400 you should create a printer description for each .print printer. Use this description to set up the IP address and TCP port number where the .print Client is installed (Illus. 6). This is also where you define print characteristics like printer model or page size.



Illus. 6 Printer definitions in .print Engine AS/400

Parameter	Description
DEVICE DESCRIPTION	Specifies the name of the unit description
REMOTE LOCATION	IP address of the .print Client
PORT	TCP port number for communication with the .print Client
CLIENT PRINTER ID	Printer ID in .print Client for targeting a specific printer

Parameter	Description
MANUFACTURER TYPE	Displays manufacturer, type, and model of a printer; this value is used by the OS/400 HostPrint Transform program to determine which control characters are to be used when the spool file is converted from SCS or AFPDS to ASCII (default: *HP4).
PAPER SOURCE 1	Paper format in paper tray 1; see table below for valid paper formats
PAPER SOURCE 2	Paper format in paper tray 2; see table below for valid paper formats
TRACE (Y/N)	Enter Trace=Y to list the program process sequence of an SSL connection for analysis. The trace output file is created with the name **T_devd**; devd is the name of the device description.

For more information, please see also the ".print Engine AS/400" user manual (Page 26).

Installing .print Client Linux

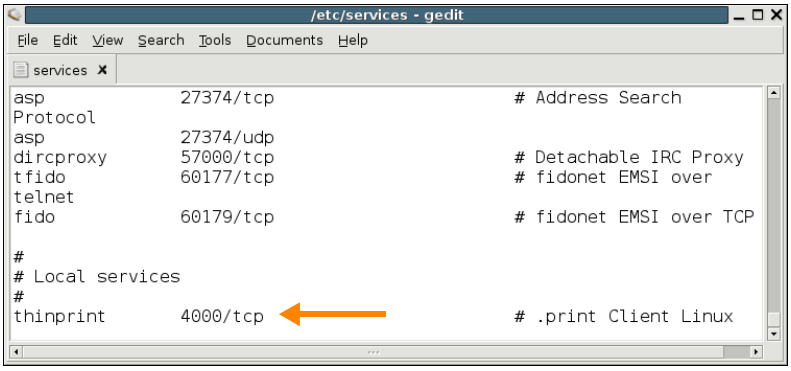
Install .print Client as follows:

1. Start Linux and log on as `root`.
2. Paste the line

   ```
   thinprint→ 4000/tcp
   ```

   into the **/etc/services** file. At least one tab space (→) separates `thinprint` and `4000/tcp`. The 4000 indicates the TCP port number that the .print Client uses[3].

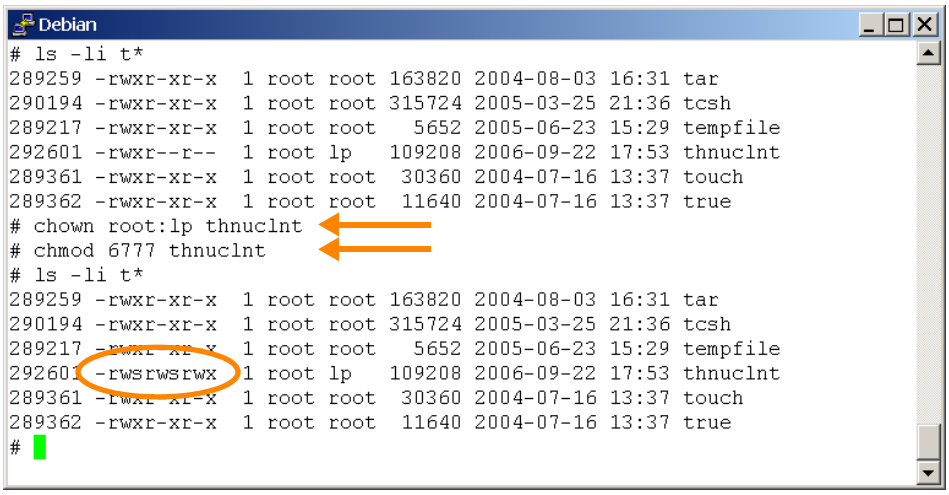
3 Be sure that the TCP port number is the same on both .print Client and .print Engine (see also Illus. 6). Otherwise, communication between the two .print components is impossible. The default port number is 4000.



Illus. 7 TCP port 4000 in the /etc/services file

3. Copy the .print Client program file **thnuclnt** into a directory to which all users have read access, e.g.:
   /usr/local/bin
4. Change this file's permissions as follows (Illus. 8):

```
chown root:lp thnuclnt
chmod 6777 thnuclnt
```



Illus. 8 Editing permissions for the *thnuclnt* file

Tip! To prevent .print Client Linux from always running with root permissions, you should configure permissions appropriate for your environment.

5. Start .print Client (see "Configuration 1: Starting .print Client from the command line", Page 14, and
   "Configuration 2: Starting .print Client with a configuration file", Page 18).

Closing .print Client Linux

ThinPrint Client closes automatically at user logoff. There are two ways to close .print Client before logging off:

Command line

Either:

```
kill -15 process_ID
```

Or:

```
killall thnuclnt
```

You can find out the process ID with the following command:

```
ps ax | grep thnuclnt
```

Configuring .print Client Linux

Overview

Parameters

.print Client Linux does not have a separate configuration menu. Settings can be entered as parameters when starting .print Client from the command line (see Chapter “Configuration 1: Starting .print Client from the command line”).

In addition to the command line, the following files are always read in automatically for configuration of the .print Client:

/etc/**thinprint.rc**	for all users (Page 18) and
/home/*user_name*/**.thinprint**	for specific users (Page 20)

Users require at least read access to /etc/**thinprint.rc** and /home/*user_name*/**.thinprint**.

.print Client can be run with or without printer class support.

Priorities

When .print Client starts, the first printer in the command line is always the default printer. Otherwise, the following priority applies:

Priorities	Configuration method	See ...
1	Command line	... next Chapter and Page 28
2	User-specific configuration file	... Pages 20 and 28
3	Global configuration file	... Pages 18 and 28
4	Default values	... Page 28

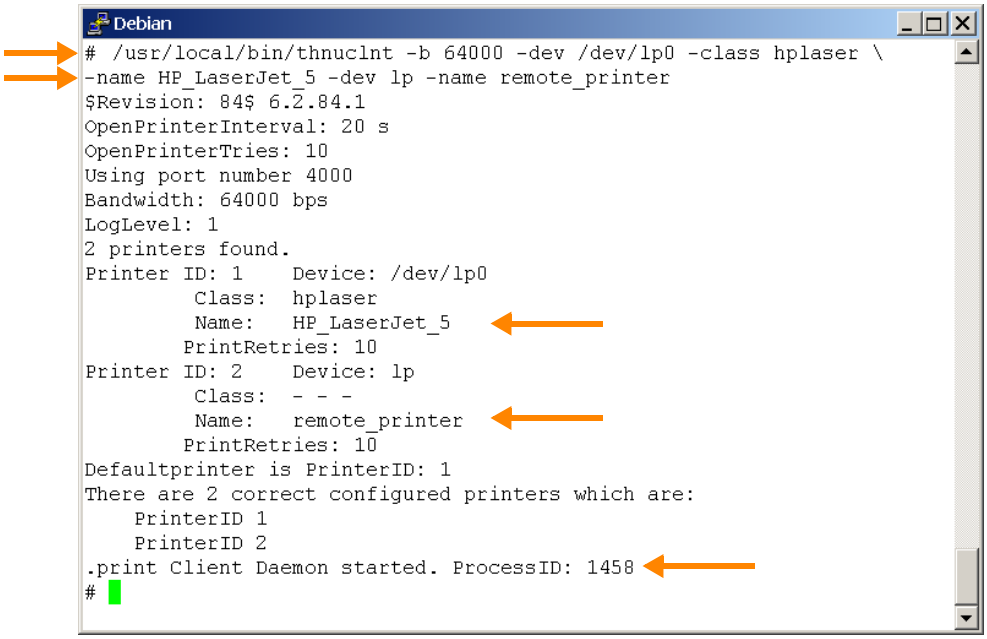
Configuration 1: Starting .print Client from the command line

Program execution

Run:

```
# /usr/local/bin/thnuclnt [-h] [-v] [-q] \[4]
[-s OpenPrinterInterval,OpenPrinterTries] [-b bandwidth] \
[-dev device [-class class_name] -name printer_name]
```

Example 1: An example with a local printer (lp0) and a network printer (lp) is shown in Illus. 9.

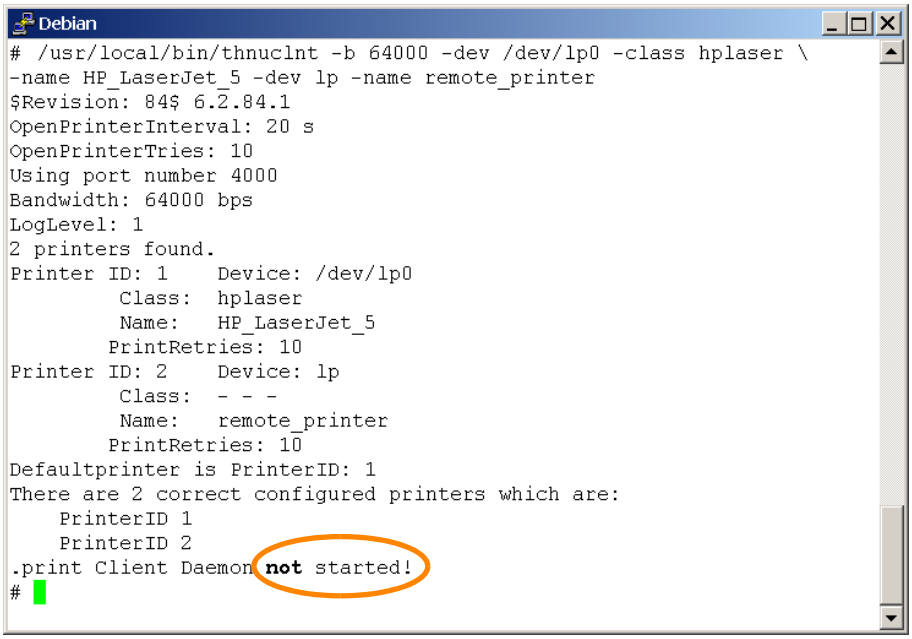


Illus. 9 .print Client started with a local and a network printer

Tip! See Page 28 for an explanation of command line parameters.

Successful startup of .print Clients is acknowledged with the message, ".print Client Daemon started" including its process ID (Illus. 9). If the attempt to start .print Client fails, the message ".print Client Daemon not started" appears in the command line (Illus. 10). In this case, the TCP port (Illus. 7) is usually already occupied by another application – possible a different running instance of .print Client.

4 The backslash \ can be used optionally to show that the command will be continued on the next line.



Illus. 10 Failed startup of .print Client

Additional examples

Without AutoConnect

The following example shows the command to open .print Client without .print AutoConnect; in other words, only printers previously manually created on the server can be used. The parameters **`-dev`** and **`-name`** must be used. Using the parameter **`-name`** without entering a printer name has the effect that AutoConnect will not recognize the printer and therefore cannot create a printer.

Example 2: **W**ithout AutoConnect

– Enter in the command line:

```
# /usr/local/bin/thnuclnt -dev /dev/lp2 -name \
-dev /dev/lp1 -name
```

The following is displayed on the monitor:

```
$Revision: 84$ 6.2.84.1
OpenPrinterInterval: 20s
OpenPrinterTries: 10
Using port number 4000
Bandwidth: using server specification
LogLevel: 1
2 printers found
Printer ID: 1   Device: /dev/lp2
                Class:
                Name:
        PrintRetries: 10
Printer ID: 2   Device: /dev/lp1
                Class:
                Name:
        PrintRetries: 10
Default printer is PrinterID: 1
There are 2 correct configured printers which are:
        PrinterID 1
        PrinterID 2
.print Client Deamon started. ProcessID: 1458
```

Note: In Example 2, bandwidth is 265000 bit/s (server default).

With AutoConnect

It's also possible to select different printers with the parameter **`-name`**. AutoConnect will automatically create all selected printers on the server and connect them with a ThinPrint port – provided templates (Page 7) exist and .print AutoConnect is running on the server. For more information, please refer to the ".print Application Server Engine" or ".print Desktop Engine" manual (Page 26).

Printer names: Unix conventions generally apply when entering printer names (consult the relevant Linux manual). If a printer name contains a blank space, for instance, it is captioned in quotation marks **`"`**. For example, the name for *HP LaserJet 5L*:

```
-name "HP LaserJet 5L"
```

If the printer name contains quotation marks, a backslash is entered at the beginning. For example, the name for *HP "super" LaserJet 5L*:

```
-name "HP \"super\" LaserJet 5L"
```

Printer classes: The parameter **`-class`** can be used to set a class for each printer. Entering a printer name is always mandatory, whereas assigning a class is optional.

Unlike printer names, class names may not be longer than 7 characters and may not included blank spaces or special characters.

Default printer: The first printer entered in the parameter list is both the "default printer" in .print Client and the default printer in a terminal session if .print Auto-

Connect is used. Print jobs are forwarded to the “default printer” if the printer ID can’t be determined (e.g., no ID is given in the printer name on the server).

Example 3: With AutoConnect

– Enter in the command line:

```
# /usr/local/bin/thnuclnt -b 65535 \
-dev /dev/lp2 -name HP5L \
-dev /dev/lp1 -class Epson \
-name EpsonStylusColor720
```

The following is displayed on the monitor:

```
OpenPrinterInterval: 20s
OpenPrinterTries: 10
Using port number 4000
Bandwidth: 65535 bps
LogLevel: 1
2 printers found
Printer ID: 1   Device: /dev/lp2
                Class:
                Name: HP5L
        PrintRetries: 10
Printer ID: 2   Device: /dev/lp1
                Class: Epson
                Name: EpsonStylusColor720
        PrintRetries: 10
Default printer is PrinterID: 1
There are 2 correctly configured printers which are:
        PrinterID 1
        PrinterID 2
.print Client Deamon started. ProcessID: xxxx
```

Note: In Example 3, .print AutoConnect attempts to connect the printers *HP5L* and *EpsonStylusColor720*. Only the *EpsonStylusColor720* can also be connected with the *Epson* class

The session printers resulting from Example 3 are shown in Illus. 11. The printer[5]

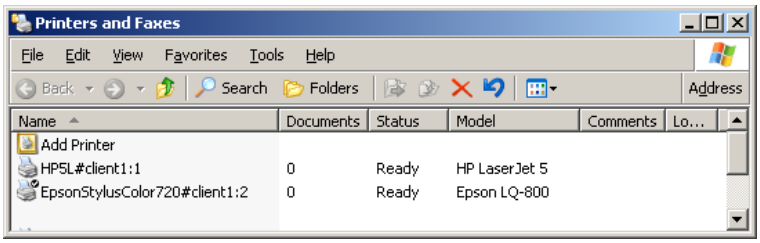
```
HP5L#client1:1
```

is created according to its printer name (HP5L) with the template _#HP5L (Illus. 3) and therefore inherits the template’s properties: bandwidth, TCP/IP protocol, HP LaserJet 5L printer driver. In contrast, the printer

```
EpsonStylusColor720#client1:2
```

5 Here, *client1* has been chosen as client name. Instead, the IP address could be used.

is created according to its class name (Epson) with the template _#Epson (Illus. 3) and therefore inherits that template's properties: bandwidth, TCP/IP protocol, Epson LQ-800 printer driver (to create the printer EpsonStylusColor720#client1:2 according to its printer name, the template _#EpsonStylusColor720 must exist on the server or a relevant entry in the name translation table).



Illus. 11 Printers created by AutoConnect in a Windows terminal session as in Example 3

Configuration 2: Starting .print Client with a configuration file

Global configuration file

In addition to the command line (Page 14), the file

/etc/**thinprint.rc**

is also automatically read in for configuration. Users require at least read access to this file.

Structure

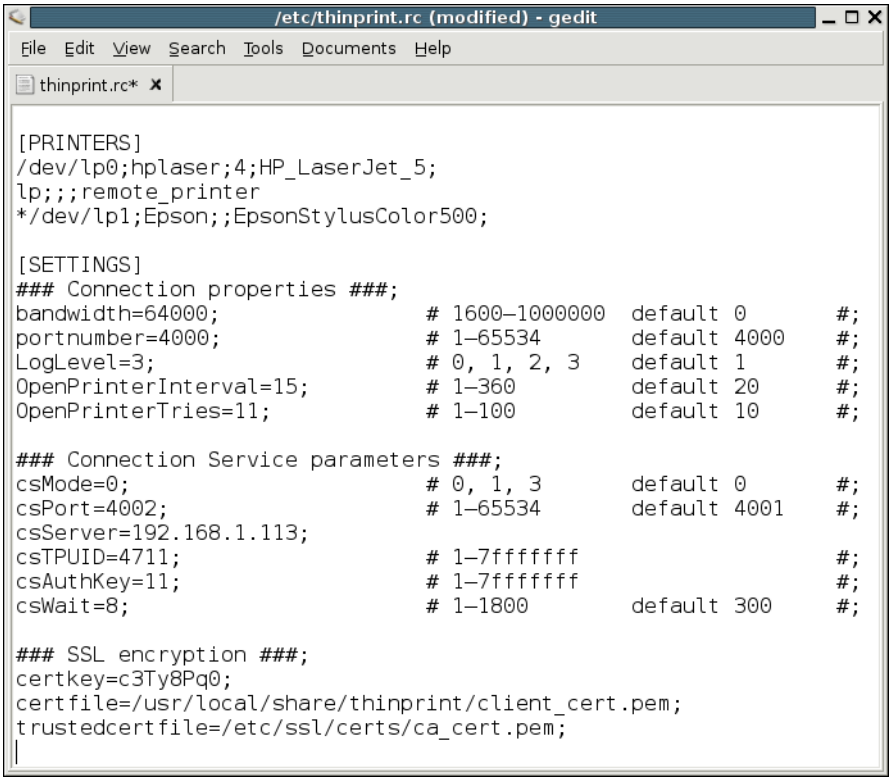
It's not absolutely necessary that a configuration file exists, because configuration can also be set from the command line and default values; see also "Configuration 1: Starting .print Client from the command line" (Page 14) and "Parameters and default values" (Page 28).

The configuration file should be built with the following structure:

```
[PRINTERS]
    [*]device;[class];[printretries];name;

[SETTINGS]
    parameter=value;
```

Example 4: Illus. 12 shows an example of a global configuration file.



Illus. 12 Contents of the global configuration file, /etc/thinprint.rc

Details

[PRINTERS] Printers must be entered with the following syntax:

`[*]`*device*`;[`*class*`];[`*printretries*`];`*name*`;`

It's also important to ensure that these names are entered completely and correctly. Only class name may be omitted.

The printer that is marked with an asterisk (`*`) at the beginning of the line is both the Default Printer[6] in .print Client and the default printer in a Windows terminal session (in case of several asterisks, the last one is applied).

[PRINTERS] + *[SETTINGS]* **At the end of each line is semicolon (`;`) followed by an ENTER.** The individual statements are:

device	Print device path; enter the print device to which .print Client Linux sends the reconstructed print job
class	Printer class name for .print AutoConnect (optional; maximum length: 7 characters)
name	Printer name for .print AutoConnect

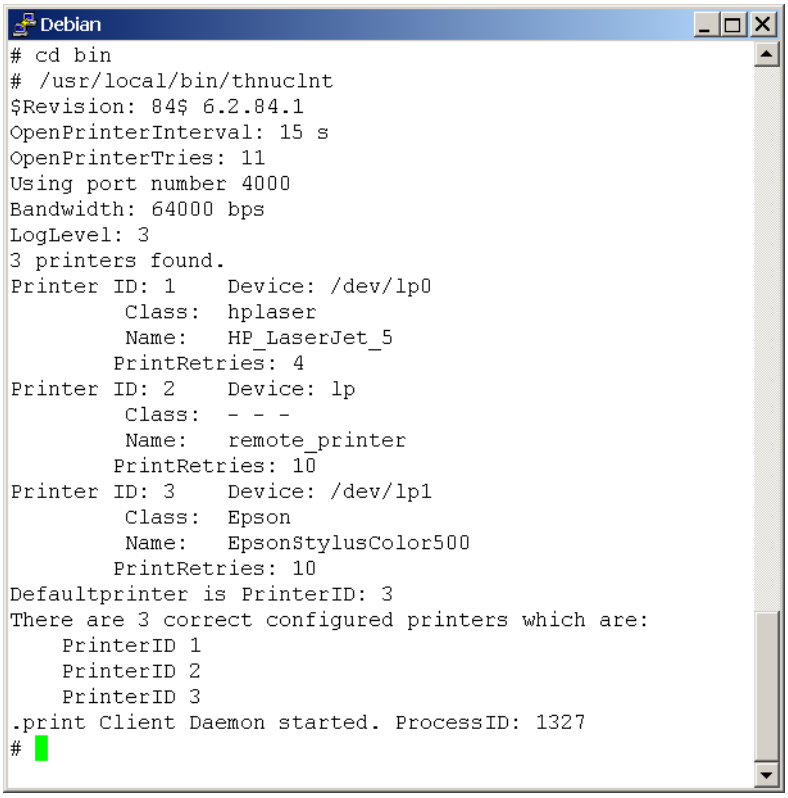
6 Print jobs are produced with the default printer if the printer ID can't be determined (e.g., no ID is given in the printer name on the server-).

printretries	Number of retries for print failure (formerly: maxkeepalives)
* (asterisk at the beginning of the line)	Default printer in .print Client Linux and default printer in a Windows terminal session
; (semicolon)	Divider
parameter	Name of the parameter
value	Value of the parameter

Starting .print Client

– - Change to the directory where you saved .print Client (Illus. 13).
– - Start .print Client from the command line with:

```
# /usr/local/bin/thnuclnt
```



Illus. 13 Starting .print Client with a global configuration file (for Example 4)

AutoConnect

The description of how .print AutoConnect functions is also true for the configuration file (see Page 16). Here, too, class names may not be longer than 7 characters and may not include blank spaces or special characters.

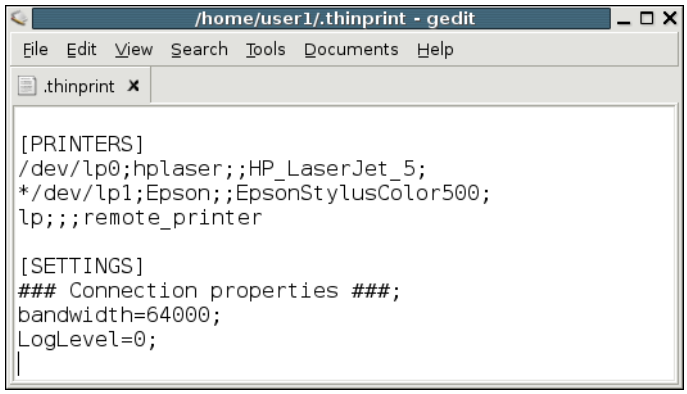
User-specific configuration file

In addition to the command line (Page 14) and the global configuration file (Page 18), the file

/home/*user_name*/**.thinprint**

is also automatically read in for user-specific configuration. Users require read permissions to this file.

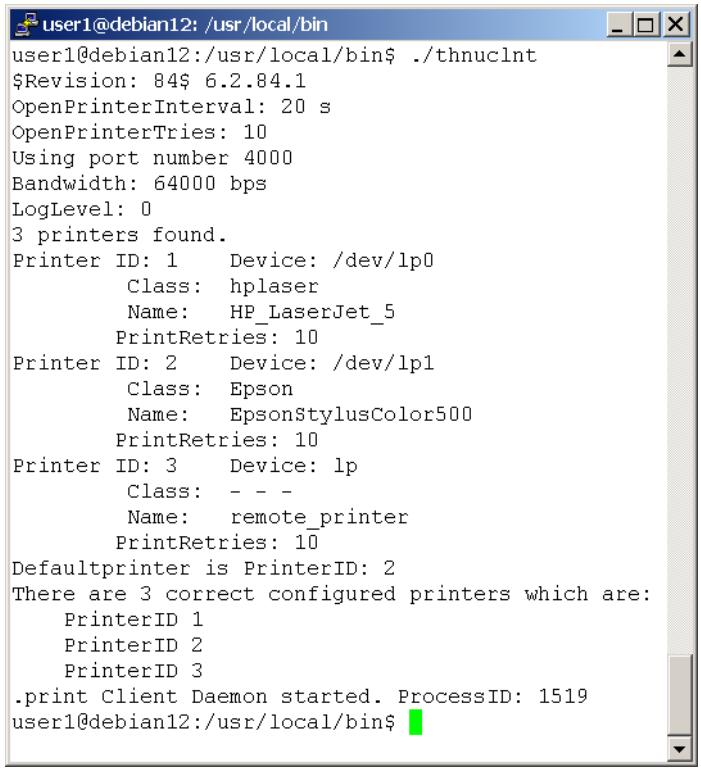
Example 5: Illus. 14 shows an example of a user specific configuration file.



Illus. 14 Contents of the user specific configuration file, /home/user1/.thinprint

– Start .print Client from the command line again (Illus. 15) with:

```
# /usr/local/bin/thnuclnt
```



Illus. 15 Starting .print Client with a user specific configuration file (for Example 5)

LPR/LPD

Besides local printers (targeted by their device names /dev/lp...), network printers can also be targeted over *LPD*. The **queue name** or one of the printer queue's **alias names** is simply added **from** the file **/etc/printcap** to the configuration file or the command line instead of the device name. It is not necessary to start the local LPD daemon here because .print Client Linux itself supports the LPR protocol per RFC 1179.

For example, a line from the configuration file **/etc/thinprint.rc**:

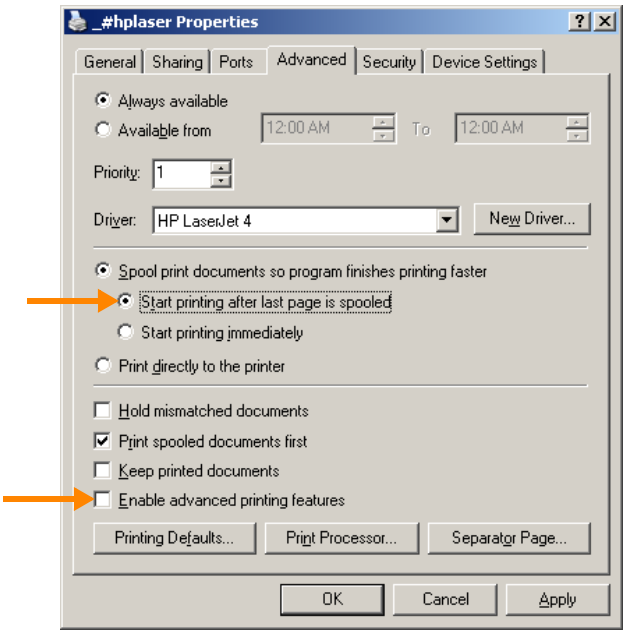
```
HPLaserJet6L;HP5L;7;my_HPLaserJet6L;
```

For LPD printing the **thnuclnt** program must use the *root* account.

Preparations on Windows servers

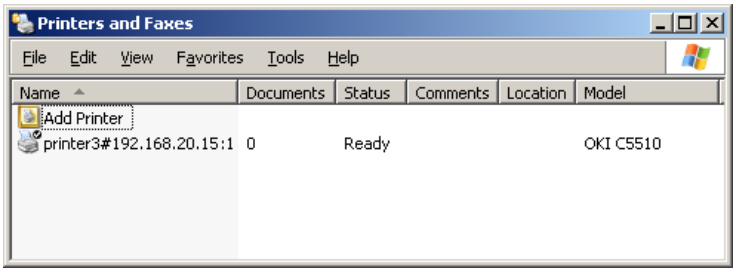
Make sure that the printers completely spool the print data on the hard drive before sending. To do so, select GENERAL under PRINTER PROPERTIES (Illus. 16).

– Select START PRINTING AFTER LAST PAGE IS SPOOLED.
– Disable the ADVANCED PRINTING FEATURES.



Illus. 16 Preferences for client-side LPD printing on a Windows server

– Create a .print printer. This must contain the Linux client's address in its name. To do so click ADD PRINTER in the server's PRINTERS AND FAXES folder (Illus. 17) and assign a ThinPrint Port that uses TCP/IP as print protocol (e.g. ThinPort:).



Illus. 17 Printer from which print jobs are forwarded to an LPD device by .print Client Linux

CUPS

Printers can also be targeted in CUPS. The **queue name** is simply added to the configuration file or the command line instead of the device name.

Troubleshooting If you have problems printing with CUPS, please check the settings below.

1. The file **/etc/cups/mime.convs** should contain the following line, which you may need to add at the end of the file:
   `application/octet-stream application/vnd.cups-raw 0 -`
2. The file **/etc/cups/mime.types** should contain the following line, which you may need to add at the end of the file:
   `application/octet-stream`

SSL/TLS encryption

Creating certificates

Three types of SSL/TLS certificates are required for encrypted transmission of .print print data:

- Client certificate
- Server certificate
- Root certificate

When planning your security strategy, you should consider that you will need to create an individual certificate for each client computer and for each server. As the name shows, client certificates are installed there where the .print Clients are running, and server certificates there where .print Engines are running. Root certificates are associated with the certification authority – the computer that issues the certificates – and are therefore always the same. Root certificates are installed on each server on which a .print Engine is installed as well as on each computer on which a .print Client Linux is running; they are used for checking the authenticity of received client certificates.

You can create your own SSL/TLS certificates with, for example, OpenSSL or a Microsoft certificate server. To use the latter, see the “Creating certificates for printing with .print” white paper (Page 26). In general, you must ensure that the server and client certificates each receive a private key; this key is always password protected.

Certificates created for ThinPrint .print are stored in a certificate file. Whereas Windows or Java environments allow different file types (e.g., .pfx, .p12, or .cer) for installing certificates, certificates for Unix or Linux must be stored as a **.pem file**. This is easiest if you create your certificates with OpenSLL. Or, you can first create the certificates with a different system (e.g., with the Microsoft certificate server); they must then be converted to .pem with OpenSSL.

Installing certificates

On client

1. Once you have created your certificates, you can install them onto the client machines. Copy the certificate issued for a specific computer (e.g. **client_cert.pem**) as well as the root certificate (e.g., **ca_cert.pem**) into a directory, specifically created for this purpose, on that client; e.g.:
   /usr/local/bin

Example:

Client certificate	/usr/local/bin/client_cert.pem
Root certificate	/usr/local/bin/ca_cert.pem

2. Next, open the **/etc/thinprint.rc** or **/home/*user_name*/.thinprint** configuration file and enter the following (Illus. 12):

```
certkey=password;
certfile=/usr/local/bin/client_cert.pem;
trustedcertfile=/usr/local/bin/ca_cert.pem;
```

Troubleshooting

.print Client Linux requires the following libraries for SSL/TLS:

- libssl.so
- libcrypt.so

I a required file is not found, start .print Client and then check the log file (Page 29) to see which files were sought by your Linux installation, then create a link to the file version installed in your Linux system; e.g.:

- libssl.so.0.9.7 (not: libssl.so.0.9.8)
- libcrypto.so.0.9.7 (not: libcrypto.so.0.9.8)

To create the links, enter the following in the command line (Illus. 18):

```
ln -s /usr/lib/libssl.so.x.x.x /usr/lib/libssl.so
ln -s /usr/lib/libcrypto.so.x.x.x /usr/lib/libcrypt.so
```

Illus. 18 Creating links for SSL/TLS libraries (example)

Connection Service

If you want to use the .print Connection Service, you must enter a variety of parameters in one of the two configuration files, **/etc/thinprint.rc** or **/home/*user_name*/.thinprint** (Illus. 12). Please also note the information about installing and configuring the .print Connection Service in your .print Engine manual and in the ".print Connected Gateway" manual (Page 26) and in the list of parameters on Page 28.

Parameter	Meaning	Explanation
csAuthKey	Authentication key	Enter the .print Client's authentication key for the Connection Service.
csMode	Use Connection Service	3 = Use of the .print Connection Service enabled
csPort	Connection Service TCP port	Enter the number of the TCP port via which the .print Client connects to the Connection Service. Default port number is 4001.
csServer	Connection server address	Enter the IP address of the Windows server on which the .print Connection Service has been installed.
csTPUID	Client ID	Here is where the client ID is entered. The Connection Service uses the client ID to forward print jobs to a client.
csWait	Connection retry interval	Enter the wait time (in seconds) after which a new attempt should be made to establish a connection if the .print Connection Service is not reachable. Default: 0.

Additional tips

- The /etc/printcap file, the CUPS configuration, and the /etc/thinprint.rc and /home/*user_name*/.thinprint configuration files are only read in at startup of the .print Client; i.e., they are not refreshed while the .print Client is running.
- When the response "... `There are n correctly configured printers` ..." appears after starting the .print Client it shows that the printers listed in the following lines are marked with a name in the list of parameters or in the configuration file. If a printer has no name and this printer should be the default printer, the next available printer becomes the default printer. If no other printer is possible or if no other printer has been marked as default printer, the printer **-1** is set as default printer.
- Waittime in the event of printer failure (PrintRetries) can only be set from the configuration file.
- The variable OpenPrinterInterval is the print job's waittime in the printer queue while attempting to start the printer. This is different than the variable PrintRetries, which specifies the print job's waittime when the printer is unresponsive (paper jam, offline, etc.).
- If printers that are not connected (the device names do not exist) are to be added, they are given the suffix **/dev/null**. For example, if the device name **/test** does not exist:

```
Printer ID: 4, class: PCL, name: test, path: /dev/null
because wrong device name: /test
```

Appendix

Customer service and technical support

Customer Service www.thinprint.com/→ SUPPORT
support@thinprint.com

Additional sources

Further information about ThinPrint .print can be downloaded from our website.

Manuals The following manuals (amongst others) are available at www.thinprint.com/→ PRODUCTS→ OVERVIEW→ <product name>.

- .print Connected Gateway
- .print Engine for VMware View
- .print Engine AS/400
- .print Application Server Engine
- .print Server Engine
- .print Desktop (Blade) Engine
- .print Client user manuals

White papers

The following white papers (amongst others) are available at www.thinprint.com/ → PRODUCTS→ OVERVIEW→ <product name> or ...→ SUPPORT→ WHITE PAPER DOWNLOAD.

- Creating certificates for printing with .print
- Licensing
- Unattended installation and licensing of .print server components

Parameters and default values

Parameter for the command line	Parameter for the configuration file	Meaning	Explanation
`-h`	—	Help	Displays available parameters and their function
`-v`	—	Version	Displays current version of .print Client
`-q`	—	Quiet	Messages are not sent. Only the message ".print Client Daemon started" is shown.
`-s`	OpenPrinter-Interval	formerly: sleeptime	Maximum waiting period until next attempt to print in case of unresponsive printer, in seconds; default: **20**
	OpenPrinterTries	formerly: maxtries/ waitattempts	Maximum number of attempts to establish a connection to the printer; default: **10**
	OpenPrinterInterval and *OpenPrinterTries* specify the amount of time a print job waits in the queue while a (connection to the) printer is being opened. (Sample calculation: 10 x 20 s = 200 s)		
—	PrintRetries	formerly: maxkeepalives	Number of repeated attempts when printing fails; maximum: 100; default: **10**
`-b`	bandwidth	Bandwidth	You can enter a bandwidth value which is the same or smaller than that set in .print Engine. If the value is larger or if client control is disabled at the server, server settings are applied. No input or 0 (zero) means that the .print Engine value will be applied; default: **0** (corresponds to server setting; server default: 256 000 bit/s).
`-dev`	—	Device	Path to printer definition; for network printers, the queue name. Enter the print device to which .print Client sends the decompressed and decrypted print job. All devices listed in the **/dev** directory can be used. Examples: `-dev /dev/lp0` (= default setting for a Linux device on LPT1) `-dev hplaser` (= definition of an LPD or CUPS device)
`-class`	—	Class	Printer class name for .print AutoConnect (optional; maximum length: 7 characters, without blank spaces or special characters)
`-name`	—	Printer name	Printer name for .print AutoConnect

Parameter for the command line	Parameter for the configuration file	Meaning	Explanation
`-log`	loglevel	LogLevel	Level of logbook entries: 0 = No entries will be written to the log file 1 = Only error messages will be logged (= default) 2 = Only error messages and warnings will be logged 3 = Log everything The logbook is an LPR logfile in /var/log.
—	certkey	Password	Password for client certificate
—	certfile	Client certificate	Path to client certificate, e.g., /usr/local/share/thinprint/client_cert.pem
—	trustedcertfile	Root certificate	Path to root certificate, e.g., /usr/local/bin/ca_cert.pem
—	csAuthKey	Authentication key	Value can be defined by Admin on the connection server; on the other hand, it is irrelevant for the first logon, but may not be changed thereafter; default: **0**
—	csMode	Use Connection Service	0 = Receiving print jobs without .print Connection Service (Listen Mode; = default) 3 = Using .print Connection Service Static Mode 1 = both modes at the same time (0 + 3)
—	csPort	Connection Service TCP port	TCP port for communication with the .print Connection Service; be sure: same TCP port number as .print Connection Service's "ClientPort"; default: **4001**
—	csServer	Connection server address	IP address of the computer on which .print Connection Service is running
—	csTPUID	Client ID	Client ID for the static mode – an unambiguous ID must be assigned for all clients (TPUID)
—	csWait	Connection retry interval	Wait time for connection retries if .print Connection Service is not reachable (in seconds); maximum: 1800); default: **300**

Tip! `-dev`, `-class`, and `-name` must always be used together and in this order; furthermore, they must be included at the end of every line. Only printer class may be omitted.

Abbreviations

CA	Certification Authority
CUPS	Common Unix Printing System
ID	Identification (number)
gcc	GNU project C and C++ compiler
GNU	GNU's Not Unix
GPL	GNU General Public License
HP	Hewlett-Packard
LAN	Local Area Network
LPD	Line Printer Daemon
LPR	Line Printer Remote
NAT	Network Address Translation
PCL	Printer Command Language
RAW	Standard print data type
RFC	Request for Comments; Series of documents, begun in 1969, which describe the Internet Protocol Suite and relevant experiments
SBC	Serverbased Computing
SSL	Secure Socket Layer
TCP/IP	Transport Control Protocol/Internet Protocol
TLS	Transport Layer Security
TP	ThinPrint
WAN	Wide Area Network

.print Clients and supported features

.print Client + possible – not possible	Network protocol			.print AutoConnect	Compression and streaming	Bandwidth control	SSL/TLS encryption	Output Gateway	.print Viewture	Virtual Channel Gateway	Connection Service
	ICA	TCP/IP	RDP								
Windows (TCP/IP)[a]	–	+	–	+	+	+	+	+	+	–	+
Windows (ICA)	+	–	–	+	+	+	+	+	+	+	–
Windows (RDP)[b]	–	–	+	+	+	+	+	+	+	+	–
Service Windows (TCP/IP)[a]	–	+	–	+	+	+	+	+	–	–	+
ActiveX Windows (TCP/IP)[a]	–	+	–	+	+	+	+	+	+	–	+
Linux (TCP/IP)[a]	–	+	–	+	+	+	+	–	–	–	+
Linux (RDP)	–	–	+	+	+	+	+	–	–	+	–
Java (TCP/IP)[a]	–	+	–	+	+	+	+	–	–	–	+
Java as an applet (ICA)	+	–	–	+	+	+	+	–	–	+	–
Win16 (TCP/IP)[a]	–	+	–	+	+	+	–	–	–	–	–
Win OS/2 (TCP/IP)	–	+	–	+	+	+	–	–	–	–	–
MS-DOS (TCP/IP)[a]	–	+	–	+	+	+	–	–	–	–	–
Windows CE (TCP/IP)[a]	–	+	–	+	+	+	–	–	–	–	–
Windows CE (ICA)	+	–	–	+	+	+	–	–	–	+	–
Windows CE (RDP)	–	–	+	+	+	+	–	–	–	+	–
Windows Mobile (TCP/IP)[a]	–	+	–	+	+	+	–	–	–	–	–
Windows Mobile (ICA)	+	–	–	+	+	+	–	–	–	–	–
Windows Mobile (RDP)	–	–	+	+	+	+	–	–	–	–	–
ActiveX Windows Mobile (TCP/IP)[a]	–	+	–	+	+	+	–	–	–	–	+
LPD device	–	+	–	–	+[c]	+	–	–	–	–	–

a Also usable for ICA/RDP sessions, provided a TCP/IP connection exists between server and client
b RDP 5.0 required
c If a decompression filter can be installed