

## Redirection Port Monitor EE 1.8.0 - Documentation - beta -



### Overview

**This is the BETA version of RedMon EE (Extended Edition) documentation !**

**The Software is at Release Level 1.8.0**

The **RedMon** port monitor redirects a special printer port to a program. **RedMon** is commonly used with Ghostscript and a non-PostScript printer to emulate a PostScript printer, or with Ghostscript as a PDF writer.

**RedMon** can be used with any program that accepts data on standard input.

Using **RedMon** you create redirected printer ports. If you connect a Windows printer driver to the redirected printer port, all data sent to the redirected port will be forwarded by **RedMon** to the standard input of a program. This program is then responsible for processing the data and producing new output.

A PostScript Windows printer redirected to a RedMon port can be shared on a network. When this printer is configured to use Ghostscript and a non-PostScript printer, it appears as a PostScript printer to other network clients.

For **SoHo Users** RedMon EE supports printers sharing and local printing within the same configuration. There is no need to setup a local printer and a **shared printer**. RedMon EE may be configured to support **network print servers**.

PostScript written to a RedMon port can be converted to a PDF file using Ghostscript.

RedMon supports Windows 95/98, Windows 2000, NT 4.0 and 3.51.

Configuring **RedMon** is not a job for novices.

[RedMon Extended Edition](#) supports Windows XP, Windows 2000 and Windows NT 4.0.

**This documentation covers RedMon EE**

[RedMon Extended Edition](#)

[RedMon principals of operation.](#)

[Installation](#)

[Uninstallation](#)

[Add Port](#)

[Port Settings](#)

[Log File Properties](#)

[What is Ghostscript?](#)

[Ghostscript Example](#)

[Output configuration](#)

[Security Settings](#)

[Server Configuration](#)

[RedMon RPC Server](#)

[Common Problems](#)

[Copyright](#)



### RedMon Extended Edition

RedMon extended edition allows You to share printers with redmon running by using and setting up the redmon Server feature.

The Server feature uses RPC to communicate and interact with the user sharing the printer. To use RPC You have to:

- have redmonS.exe running on each client using the redmon printer on the network.
- configure the redmon Server feature on Your print server.
- configure redmonS.exe

The [RPC feature](#) uses [named pipes](#) or [TCP/IP](#) as the communication layer. RPC using named pipes should be much easier to handle than [TCP/IP](#) ! But for security reasons some installation will not work with RPC using named-pipes. The printer runs in the system (kernel) context, which seems not to have the security rights needed to communicate over named-pipes. You will get an error if named-pipes don't work.

If You are using [TCP/IP](#) as communication layer You must have a WINS server running in Your network or You have to update Your HOSTS file.

Redmon running on the print server must get the ip-address (e.g. 10.10.70.42) of the workstation running the redmonS.exe (RPC server). The only address information which the printer server gets out of the print-job parameters are:

- workstation name e.g. \\client01
- username e.g. user615

these informations will be sufficient if You use RPC with named-pipes. But TCP/IP needs:

- the IP-Address (e.g. 10.10.70.42)
- the port the server is running !

If there is only **one user** running on the client - users don't share systems - You can use a default port for TCP/IP communications. This is the recommended usage.

If You have more than one user running the on the same client, redmon will construct a port-number from the username by

generating a 16-bit CRC from the username. This seems to be the only working method to assign different ports to different users on the same client.

To retrieve the IP-Address from the client- or workstation-name You should have WINS running in Your network. If You don't have setup a WINS-Server, You will have to add each workstation in the HOSTS file on Your print server !

Configuring **RedMon Extended Edition** is not a job for novices.

**RedMon Extended Edition** comes with a set of tools to test the RPC-feature:

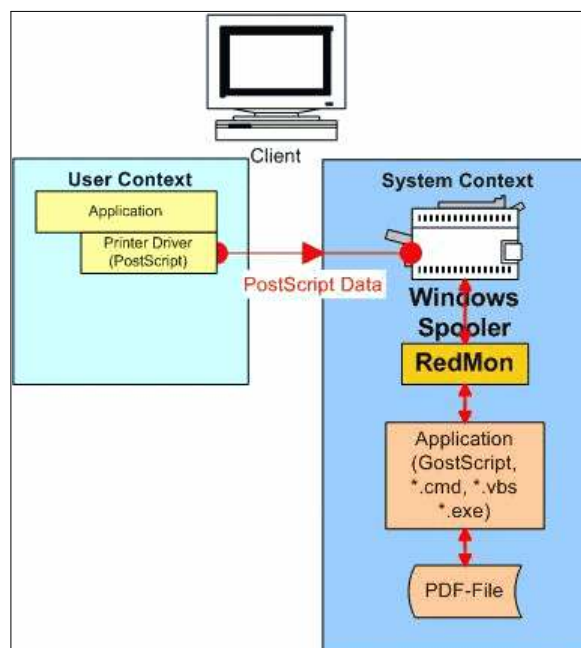
1. redmoncl.exe is a command-line utility to test the RPC communication against a RedMon RPC Server (redmonS.exe)
2. redmoncs.exe is a command-line primitive RPC Server to test the RPC Communication

Please note, that these test drivers always run in a user-context! If the RPC-feature is executed in the system context of the windows spooler system, RPC may generate error and may be unexecutable. This seems to hold for using named pipes as the communication layer.

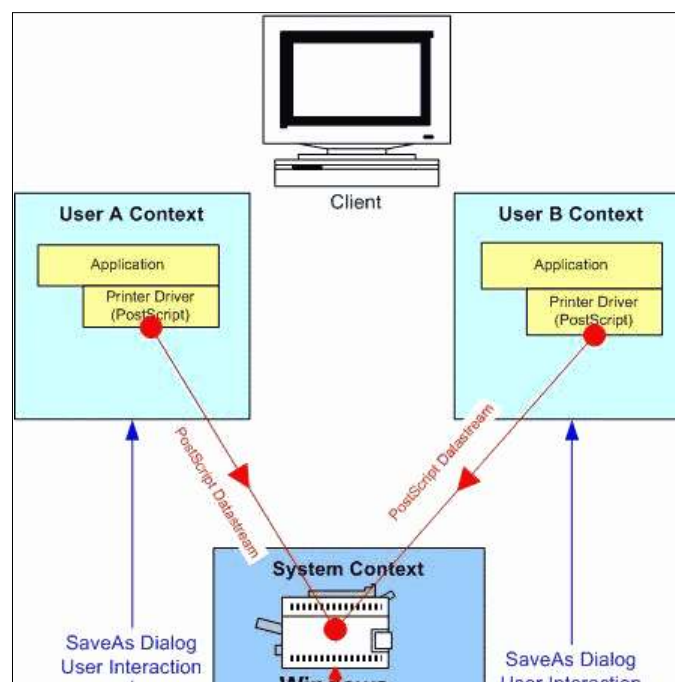
If RedMon EE and the user submitting the print job are both running on the same machine, the RPC-feature will not be used by RedMon !

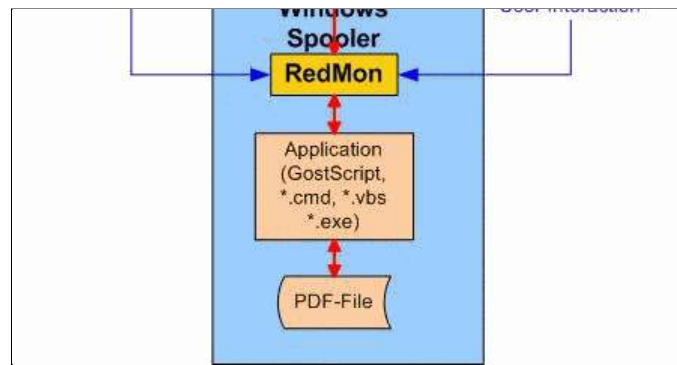
### RedMon principals of operation.

RedMon is always running in the system context of the windows printing and spooling sub-system ! On a single user system with a local RedMon enabled printer Applications run in a User-Context and execution of RedMon is under control of the spooler subsystem. On Windows NT systems this is not really a problem. It could be a problem if You are running on W2K and it has been a really problem if running Windows XP.

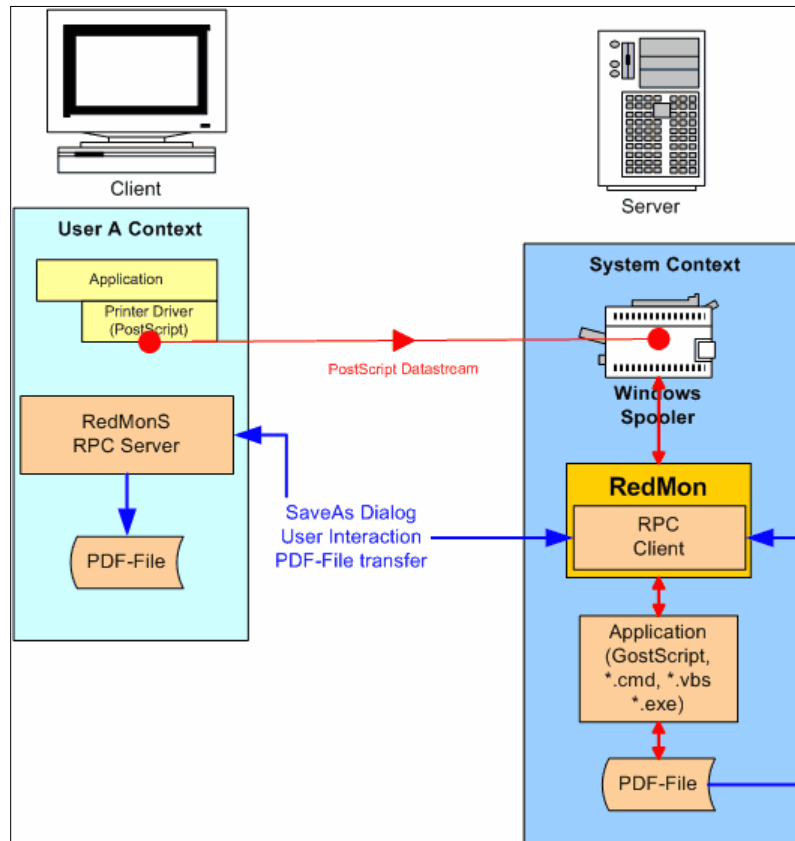


On a client used by multiple users RedMon runs in the system-context and each user has is own user-context running applications. RedMon EE is able to address the right user context e.g. for user interaction by a SaveAs Dialog, or if a user process should be started after the print job is finished.





If a RedMon Process is running on a print-server the only way for user-interaction is to use IPC features or client/server programming techniques. To make the IPC communication as simple and robust as possible, RedMon EE comes with an RPC feature (Remote Procedure Call).



Clients will have to run the RedMon EE RPC-Server if user-interaction is needed for the printing process or even if the generated print-file should be send back to the client for further processing. This RPC feature and the ability to pass print-output back to the client avoids the whole stuff with drive-mapping and access rights. Only the spool-process it self must have write access to some temporay storage, which should not be a real problem.

If You are running more than one print server with RedMon You should not run in any problem

Note RedMon's RPC feature reverses Client and Server.  
The **print server** always acts as a **client**!  
The **client** starting the print job always runs the **RPC-Server**!

We have a single RPC-Client talking with multiple servers

Really hard to understand. But things work easier this way.



## Installation

Run **setup.exe** to install the **RedMon** redirection port monitor. The files **setup.exe**, **redmon95.dll**, **redmonnt.dll**, **redmon35.dll**, **redmon.hlp** and **unredmon.exe** must all be in the same directory.

Installing **RedMon** under Windows NT, W2K, XP will require Administrator privileges.

Follow the [Ghostscript Example](#) for the rest of the configuration.



## Uninstallation

To uninstall, you must first disconnect all printer drivers from the **Redirected** ports and then delete the **Redirected** ports. Use **Add/Remove Programs** from the **Control Panel** to remove **RedMon - Redirection Port Monitor**.



## Add Port

In the **Add Redirected Port** dialog box, enter the name of a port to be serviced by **RedMon**. By default, the first port you define will be named **RPT1:**.

**Windows 95/98** After adding a port, you need to configure it with [Port Settings](#) and **Spool Settings**. Any printer driver connected to this port **must have** the **Spool Settings** set to **Disable bi-directional support for this printer**.

**Windows NT** After adding a port, you need to configure it with [Configure Port](#). **Enable bidirectional support** must be disabled.

You cannot use the same port name as an existing port. In particular, you cannot create a redirected port named LPT1:



## Port Settings

In the **Port Settings** dialog box, specify a redirection program which is to receive on standard input all data sent to this port. The program name is entered in the first edit field and the arguments in the second edit field.

The **Output** from the program can be set to one of

1. [Program handles output](#)
2. [Prompt for filename](#)
3. [Template Filename](#)
4. [Copy stdout to printer](#)
5. [Copy temporary file to printer](#)
6. [Copy pipe to printer](#)

If one of the latter three, you need to select a **Printer**.

If you select [Print Errors](#), an error report page will be printed.

The program can be run **Normal**, **Minimized** or **Hidden**.

[Run as User](#) is available only under Windows NT (and up) and is usually disabled.

A [Log File](#) can be configured to receive standard output and standard error. With a log file, you can view error messages that were written to standard output or standard error.

You may define a [Programm to Start after works](#) which may perform some action on the generated output file.

If this module has a GUI and is interacting with the user You **must** set **Run As User**.

See the [Ghostscript Example](#) and [Common Problems](#).

[Program handles output](#)  
[Prompt for filename](#)  
[Template Filename](#)  
[Copy stdout to printer](#)  
[Copy temporary file to printer](#)  
[Copy pipe to printer](#)  
[Programm to Start after works](#)  
[Print Errors](#)  
[Environment Variables](#)  
[Run as User](#)  
[Shut down delay](#)



## Program handles output

When output is set to **Program handles output**, the redirection program is entirely responsible for all output. If the redirection program wants to write to another printer, it must do this itself. If the redirection program wishes to generate no output at all, this is also acceptable.



## Prompt for filename

When output is set to **Prompt for filename**, the redirection program should write its output to a file. The name of the file is obtained from a **Save as** dialog, can be obtained by inserting **%1** in the program arguments. If you wish to place **%1** in the program arguments and do not want it substituted with the filename, you must instead use **%%1**. For example, the program arguments might include:

```
-sOutputFile="%1"
```

This method is recommended for use with Ghostscript, and is commonly used with a PostScript printer driver and the Ghostscript pdfwrite device to create a [PDF writer](#).

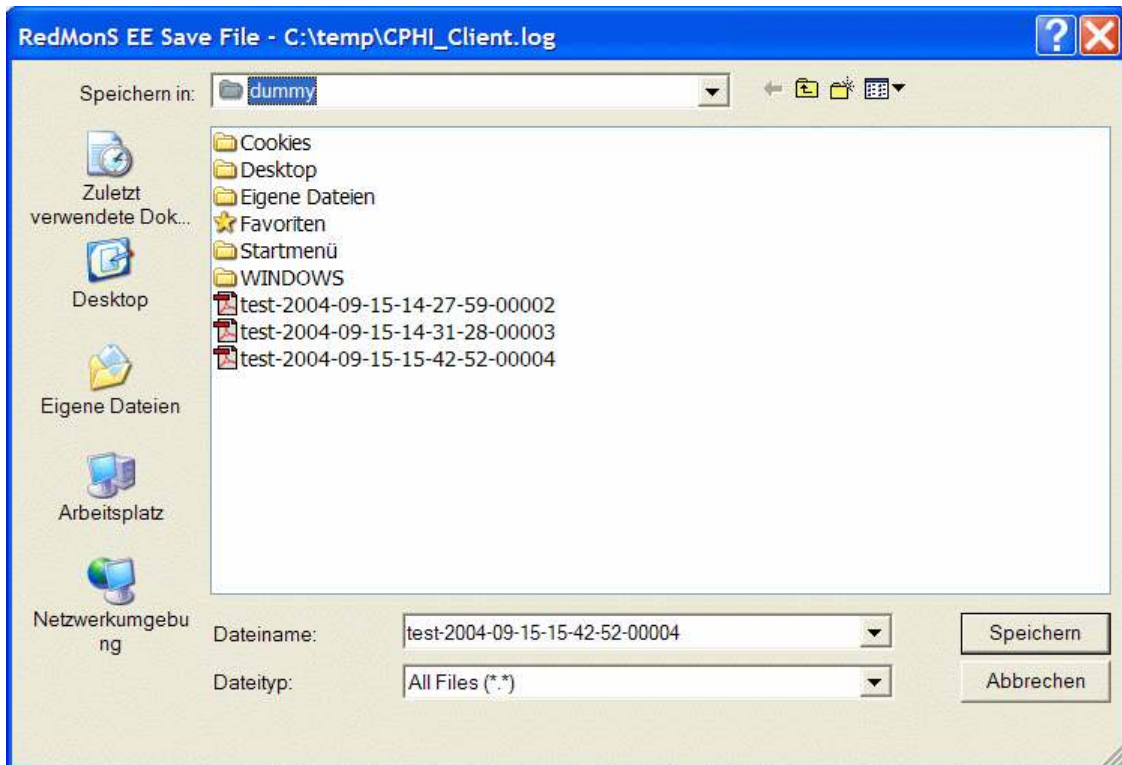
Do not share a printer which uses RedMon with **Prompt for filename**. RedMon will not allow this because the **Save as** dialog box would appear on the server computer, not the client which submitted the print job.

If You share a printer which uses RedMon You may setup the [RedMon Extended Edition](#). The extended edition of redmon comes with a build in RPC feature which allows You to share a printer which uses RedMon.

To use this feature You must setup the RPC Configuration in the [Server Configuration](#) part of the configuration dialog !

To set more options set [Filename prompt Options](#) in the [Output configuration](#) dialog.

The SaveAs Dialog show will look like this:



At a close look You will see the file which has been generate by a filename template setup as **%temp%\test-%d-%t-%n.pdf** And in the title You will have the **Documentname**.



### Template Filename

When output is set to **Template Filename**, You have to enter the **Template for Output File** in the **Output configuration** tab of the configuration dialog.



### Copy stdout to printer

When output is set to **Copy stdout to printer**, the redirection program should write its output to standard output. RedMon will copy this to the specified printer. Standard output will not be written to the log file.

This method should not be used with Ghostscript 5.50 or earlier.



### Copy temporary file to printer

When output is set to **Copy temporary file to printer**, the redirection program should write its output to a temporary file. RedMon will copy this temporary file to the specified printer. The name of the temporary file is specified by RedMon, and can be obtained by inserting **%1** in the program arguments. If you wish to place **%1** in the program arguments and do not want it substituted with the temporary filename, you must instead use **%%1**.

For example, the program arguments might include:

```
-sOutputFile="%1"
```

This method is recommended for use with any version of Ghostscript.



## Copy pipe to printer

When output is set to **Copy pipe to printer**, the redirection program should write its output to a pipe created by RedMon. The handle as a hexadecimal value can be obtained by inserting **%h** in the program arguments. RedMon will copy this pipe to the specified printer.

For example, the program arguments might include:

```
-sOutputFile="%%handle%%h"
```

This method is recommended for use with Ghostscript and requires Ghostscript 6.0 or later.



## Program to Start after works

Use this entry to define a program which is started after the print process is finished. E.g. to run Acrobat reader to show the generated .pdf file.

[Run as User](#) **must** be set only if this program has a GUI and needs interaction with the user. If RedMon is running on a **Server** the [Run as User](#) feature should never be used !



## Print Errors

If **Print Errors** is selected and the redirection program terminates prematurely, RedMon will print an error report page. This will contain the contents of the [Log File](#). Selecting **Print Errors** and **Debug** will result in much wasted paper.

**Print Errors** is most useful when a redirection program writes an error message to standard output, which is captured by the log file.

**Print Errors** is ignored if [Program handles output](#) is selected.



## Environment Variables

The redirection program **and** the AfterWorksProcess can obtain some print job parameters from the environment variables. Those set by RedMon are:

<b>REDMON_PORT</b>	is the port name, typically "RPT1:".
<b>REDMON_JOB</b>	is the job number, an integer.
<b>REDMON_PRINTER</b>	is the name of the printer (e.g. "HP DeskJet 500").
<b>REDMON_MACHINE</b>	is the name of the machine that created the job (e.g. "\\Client01").
<b>REDMON_USER</b>	is the name of the user who owns the job (e.g. "JCITIZEN").
<b>REDMON_DOCNAME</b>	is the name of the print job (e.g. "Printer Test Page"). <b>%d</b> in the program arguments will be replaced by the name of the document.
<b>REDMON_FILENAME</b>	is the file name of the output file if <a href="#">Prompt for filename</a> is used, or the temporary file if <a href="#">Copy temporary file to printer</a> is used.
<b>REDMON_SESSIONID</b>	is the session id of the Windows Terminal Server client session. This will be zero if Windows Terminal Server is not used. The sessionid will also be present on XP systems if You use fast user switching !

## String substitution

RedMon parameters and program variables may be setup with some substitution variables. For Your convenience RedMon offers a set of standard substitutions as follows:

<b>%1</b>	substitutes the output filename this may be generated by a template or a temporary file name generated by RedMon
<b>%S</b>	is substituted by the <a href="#">PDF Security options</a>
<b>%n</b>	is substituted by the print sequence number
<b>%h</b>	is replaced by a hex number of the pipe used
<b>%p</b>	is substituted by the user hompath
<b>%d</b>	is substituted by the documentname Be carefull if using this name. Some applications may generated "unusual" filenames which may be not well behaved
<b>%c</b>	is substituted by the actual date yyyy-mm-dd memonic "creation date"

<b>%t</b>	is substituted by the time hh-mm-ss
<b>%u</b>	will be replaced by the username submitting the print job
<b>%%anyvalue%%</b>	is substituted by %anyvalue%
<b>%env-var%</b>	is substituted by the proper environment variable

**A Note on environment variable substitution:**

Always keep in mind that RedMon is running in the system context of the Windows Spooler !

This context **does not** have the user environment set up !

If running on a remote print server the environment will be completely different from what You may expect !

**Before You use environment variable substitution take a close look at the setup to be quite sure what You will get !**

**RedMon** environment variables will be replaced by the environment variable value !

Rules of string substitution:

- environment variable substitution is executed first.
- after expansion of environment variables RedMon variables will be replaced
- always use separators: %username%%d may be misspelled. For a clear substitution write %USERNAME%-%d

**Hint:**

If You want a substitution of environment variables like %username% this may conflict with the RedMon %u.

It's an good idea to write environment variables in **UPPERCASE** e.g. %USERNAME%, this will avoid any conflict with the RedMon substitution functions.

**RedMon substitution is CaSeSeNsItIvE !**

**TEMP** and **TMP** are directories for temporary files.

Note that under Windows NT, the redirection program runs in the environment of the SYSTEM account. Apart from the above environment variables, very few are set. The registry HKEY\_CURRENT\_USER and user profile are that of "Default User".

**Run as User**

"Run as User" causes the program to be run as the local user who submitted the print job.

"Run as User" is only available for NT 4.0 or later.

Do not enable this if the printer is shared on a network.

Remote users can't run programs on your computer. If a remote user tries to submit a job, the EXE will fail with a message like:

```
Initialization of dynamic link library
C:\WINNT4\system32\kernel32.dll failed.
The process is terminating abnormally.
```

The spooler service will then be locked up until you stop and restart it.

**Run as User** was intended for launching a GUI program such as GSview **locally** via RedRun.

- The program will have the **security attributes** of the user who submitted the print job !
- The program will have the environment of the SYSTEM account.
- User profile information are not guaranteed to be correct.

Some notes on this feature:

- **Never use "RunAsUser" if Your programm needs any handles**  
As an example GhostScript uses stdin to get the input stream from RedMon ! Win32 is unable to pass any handle from a system context to a user context !  
*Keep always in mind:* RedMon is an integral part of the Win32 spool system !  
Therefor it's really useless to start a process that receives input from readmon via stdin with "RunAsUser" RedMon always tries to pass the standard handles stdin, stdout and stderr to the called process. This must fail and RedMon automatically start this process again in the system context !

**RedMon EE always starts the primary process in the system context !**

The primary process is known as "Redirect this port to the program:" in the configuration dialog.

- You **must** switch on "RunAsUser"  
If Your "After Works" process needs user interaction.  
If You want to start e.g. Acrobat Reader to view the pdf-file produced by GhostScript, this process **must** be startet in the user-context !  
If only want to send the output-file to Your E-Mail system without any user-interaction this process may run under control of the system context.
- You **never** may use "RunAsUser"  
if RedMon is running on a print server, these servers don't have a user context for a remote user starting a print-job !  
Neither the redirected program nor the "after works" process could be run in a user context.  
By default RedMon does not start any process with "RunAsUser" if a print-server context is recognized by RedMon
- Short conclusion  
**only the "RunAsUser" feature for a process that is started after the printer data stream processing is finished**





## Shut down delay

After writing all data to the redirection program, **RedMon** will normally wait up to 300 seconds for the redirection program to close before writing any output to the printer and closing the log file. If the program takes longer than 300 seconds to terminate, the output of the program may be lost or truncated. Some standard output may not get written to the log file. If this occurs, you may need to increase the **Shut down delay**.



## Log File Properties

If you tell **RedMon** to use a log file, all standard output from the redirection program will be written to this file. This is useful for capturing error messages from the program to which you have redirected standard input.

To use a log file, you must check **Use Log File** and then specify a valid file name for the log file.

If you check **Debug**, RedMon will write extra debugging messages to the log file. These messages are probably only useful to the author of RedMon. If you want the log file to contain only the standard output of the redirected program, uncheck **Debug**.

The log and debug options are set in the [Output configuration](#) dialog.

See the [Ghostscript Example](#).



## What is Ghostscript?

Ghostscript is the name of a set of software that provides:

- An interpreter for the PostScript (TM) language and the Adobe Portable Document Format (PDF -- sometimes confused with Acrobat, Adobe's PDF browser/editor product), and
- A set of C procedures (the Ghostscript library) that implement the graphics and filtering (data compression / decompression / conversion) capabilities that appear as primitive operations in the PostScript language and in PDF.

Ghostscript is available for Microsoft Windows, IBM OS/2, DEC VMS and many versions of Unix from

<http://www.cs.wisc.edu/~ghost/>

Ghostscript is a copyright work owned by Aladdin Enterprises.

GSview is a previewer from Ghostgum Software Pty Ltd, based on Ghostscript. GSview is available from

<http://www.cs.wisc.edu/~ghost/gsview/>

If you wish to use Ghostscript with **RedMon**, you should install Ghostscript using the **setup** program provided with it. Use GSview to make sure that Ghostscript is running correctly before trying to use RedMon.



## Ghostscript Example

Windows 95/98 and Windows NT have slightly different configurations.

In the following examples, the Ghostscript command

```
c:\gs\gs7.03\bin\gswin32c.exe
```

and the Ghostscript include path

```
-Ic:\gs\gs7.03\lib;c:\gs\fonts
```

will both need to be changed to match your version of Ghostscript.

[Ghostscript Example for Windows NT 4.0](#)

[Ghostscript Example with HP LaserJet printer PDF writer](#)



## Ghostscript Example for Windows NT 4.0

You can use Ghostscript with **RedMon** to create a PostScript printer.

To print to a HP DeskJet 500 printer whose queue name is "HP DeskJet 500", first create a file named c:\gs\djet500.rsp which contains the following:

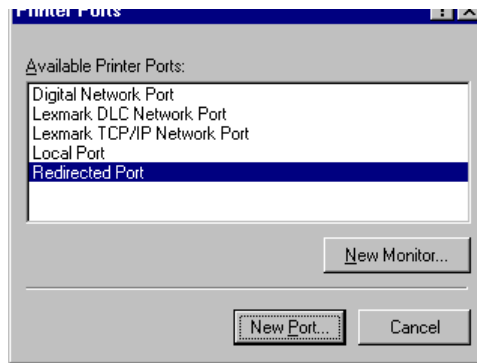
```
-Ic:\gs\gs7.03\lib;c:\gs\fonts
-sDEVICE=djet500
-dNOPAUSE
-dSAFER
-sPAPERSIZE=a4
```

See the Ghostscript files **Use.htm** and **Install.htm** for an explanation of these options.

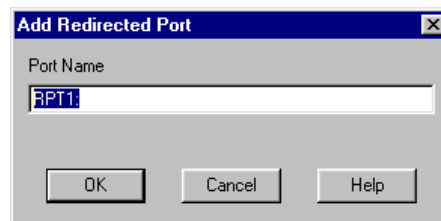
Start the **Add Printer Wizard**. On the second page select [Add Port](#).







Select **Redirected Port** then **New Port**. The RedMon [Add Port](#) dialog will appear. Accept the defaults.



Press **Close** then [Configure Port](#). The **RPT1 Properties** dialog will appear.

Enter the program name

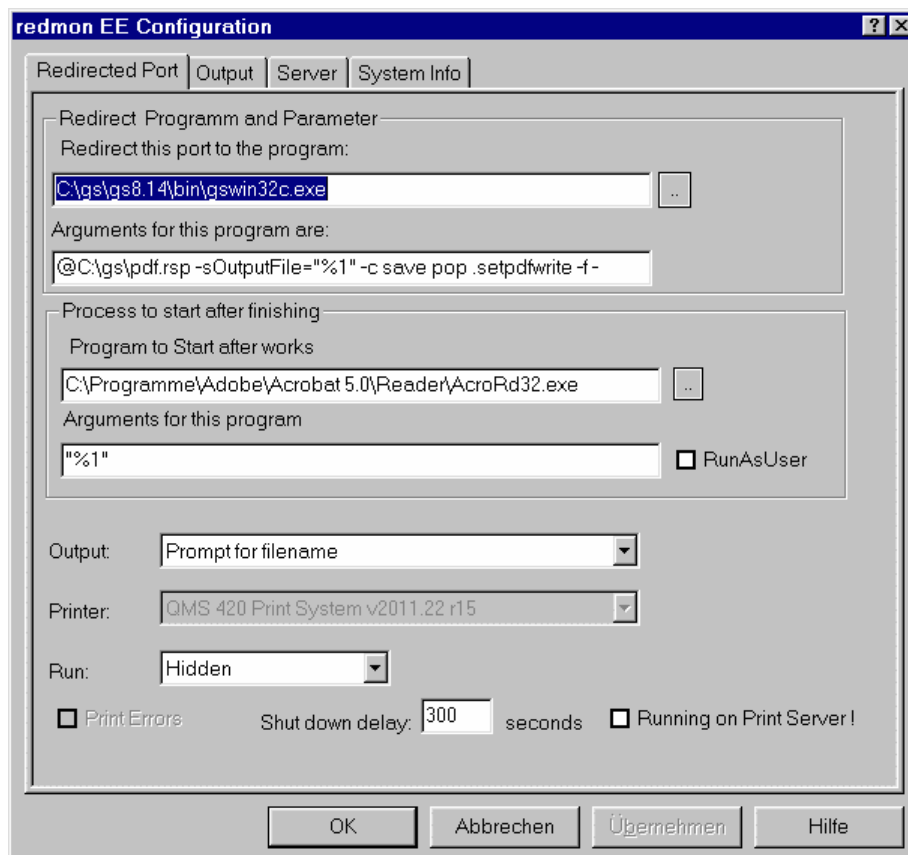
```
c:\gs\gs7.03\bin\gswin32c.exe
```

and the arguments

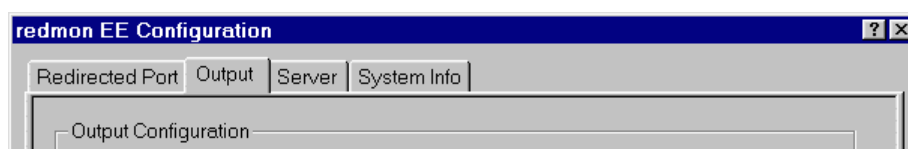
```
@c:\gs\djet500.rsp -sOutputFile="%1" -
```

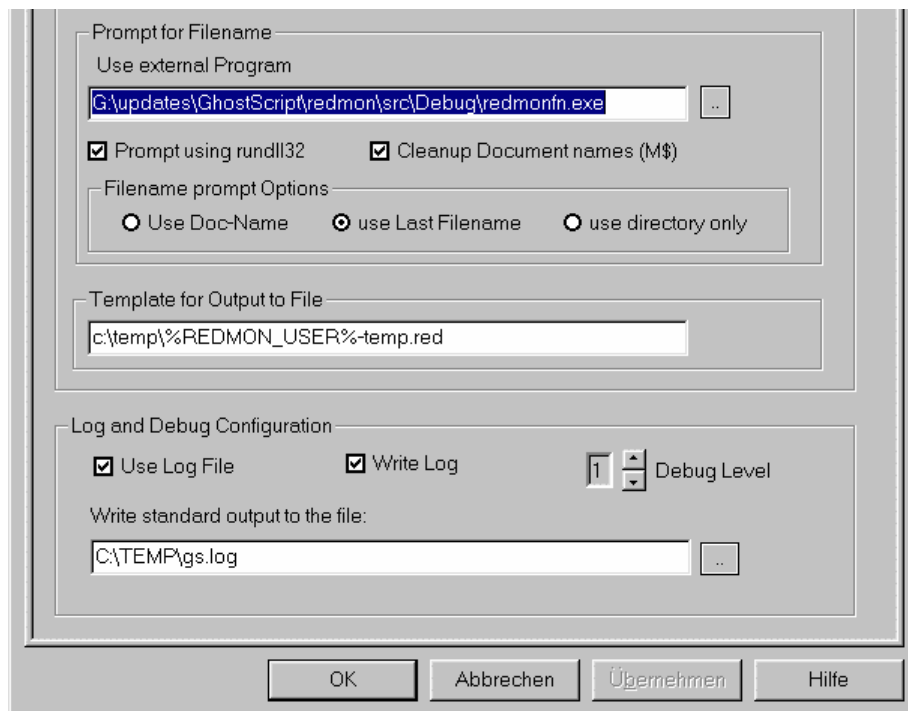
Make sure you include the space and **dash** at the end of the line. Failure to do this could result in Ghostscript stalling the print queue.

Set output to [Copy temporary file to printer](#) and set printer to **HP DeskJet 500**.



If you want to see what is written to stdout (i.e. you want to look at error messages), see [Log File Properties](#).





Press the OK buttons on the [Log File Properties](#) and **RPT1 Properties** dialog boxes.

Select the **Apple LaserWriter II NT v47.0** from the manufacturer and model page of the Add Printer Wizard.

Printing to this **Apple LaserWriter II NT** printer will now print using Ghostscript. If you share this printer on a network, it will appear as a PostScript printer to other network clients.

For a list of available Ghostscript printer devices, change to the Ghostscript directory c:\gs\gs7.03\bin then run

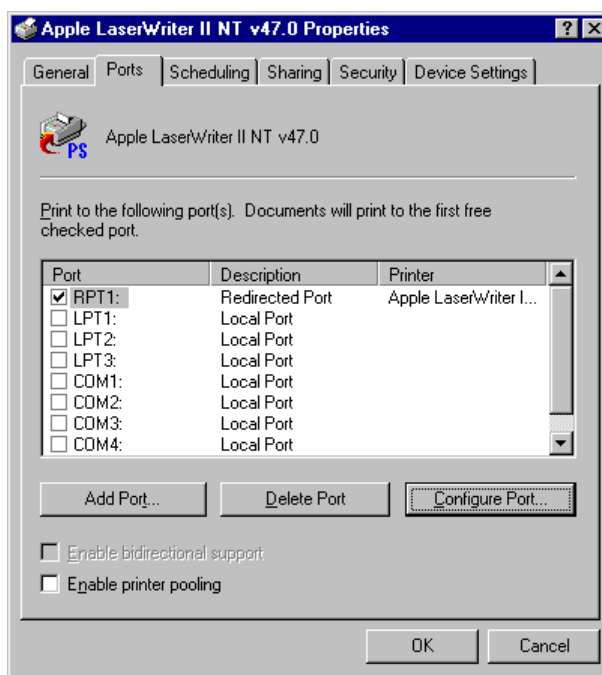
```
gswin32c -h
```

If your printer is not listed, have a look at

<http://www.cs.wisc.edu/~ghost/doc/printer.htm>

to see if one of the existing Ghostscript printer devices will work with your printer.

If you want to configure the redirected port later, open the printer properties, select the **Ports** tab, then **Configure Ports**. RedMon requires that bi-directional support be disabled, but it appears that it can't be enabled.



### Ghostscript Example with HP LaserJet printer

For the HP LaserJet 4L printer, you need to use the Ghostscript device ljet4 with resolution set to 300dpi. c:\gs\ljet4l.rsp would contain:

```
-Ic:\gs\gs7.03\lib;c:\gs\fonts
-sDEVICE=ljet4
-r300
-dNOPAUSE
-dSAFER
-sPAPERSIZE=a4
```

The Ghostscript command would be

```
c:\gs\gs7.03\bin\gswin32c.exe
```

and the arguments

```
@c:\gs\ljet4l.rsp -sOutputFile="%1" -
```

Make sure you include the space and **dash** at the end of the line. Failure to do this could result in Ghostscript stalling the print queue. Output would be

```
Copy temporary file to printer
```

and printer

```
HP LaserJet 4L
```

All other settings should be the same as the previous example.



## PDF writer

PostScript written to a RedMon port can be converted to a PDF file using Ghostscript.

Install a printer driver for a colour PostScript printer, e.g. **Apple Color LaserWriter 12/600**. If you select a black and white printer such as **Apple LaserWriter II NT** you will end up with your colour images becoming greyscale.

To use RedMon and Ghostscript to create a PDF writer, you need to use the Ghostscript device pdfwrite. c:\gs\pdfwrite.rsp would contain:

```
-Ic:\gs\gs7.03\lib;c:\gs\fonts
-sDEVICE=pdfwrite
-r300
-dNOPAUSE
-dSAFER
-sPAPERSIZE=a4
```

The Ghostscript command would be

```
c:\gs\gs7.03\bin\gswin32c.exe
```

and the arguments

```
@c:\gs\pdfwrite.rsp -sOutputFile="%1" -c .setpdfwrite -f -
```

Make sure you include the space and **dash** at the end of the line. Failure to do this could result in Ghostscript stalling the print queue. Output should be set to

```
Prompt for filename
```

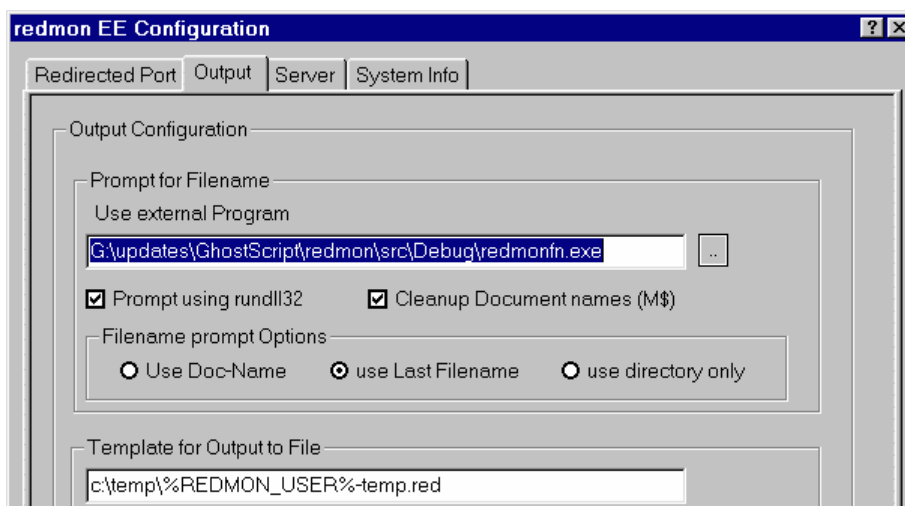
All other settings should be the same as the previous examples.

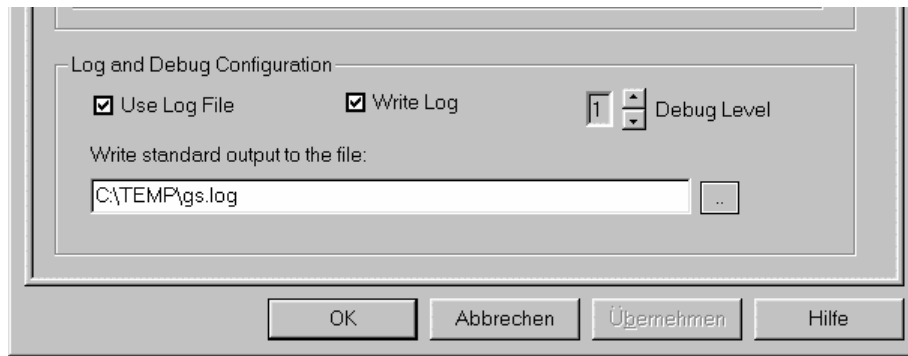
You may wish to rename the printer from **Apple Color LaserWriter 12/600** to **Ghostscript PDF writer**.



## Output configuration

Configuration of output Options





Within this dialog You have to configure where Your output goes. If You have set the [Prompt for filename](#) option You can use:

1. an [external Programm](#) to bring up the SaveAs Dialog
2. use redmons internal Dialog by setting [Prompt using rundll32](#)

Set the [Cleanup Document names](#) switch if You want to remove **Microsoft** - string from document names.

The SaveAs dialog may be started with different [Filename prompt Options](#)

[Prompt using rundll32](#)  
[Use External Programm](#)  
[Cleanup Document names](#)  
[Filename prompt Options](#)



### Prompt using rundll32

By setting **Prompt using rundll32** in the [Output configuration](#) redmons internal SaveAs Dialog is shown by calling an entry-point in **redmonnt.dll**. The dialog-box will always be shown in the user context.

If You like to have a separate program or if You want to use a user-written module to prompt for the filename enter a program in the **Use External Programm** entry.



### Use External Programm

You may specify an external program name in the **Use External Programm** to prompt the user for a filename. This external program is called with name of a pipe as first parameter and a filename as the second parameter. e.g.:

```
myprogram.exe <pipename> <filename>
```

The external program has to:

- open the pipe
- get the filename by user interaction
- write the filename into the pipe
- close the pipe
- exit

You may use **redmonfn.exe** if You don't like to have a call to **redmonnt.dll**.



### Cleanup Document names

The **Cleanup Document names** switch in the [Output configuration](#) dialog will remove:

- Microsoft Word -
- Microsoft PowerPoint -
- Microsoft Excel -
- Visio-Doc-

from Your document name. Sorry this feature of building clean names may not be configured by users. It's hard coded in redmon.



### Filename prompt Options

The [Save as](#) Dialog may be started with different options which could be set in the [Output configuration](#) dialog.

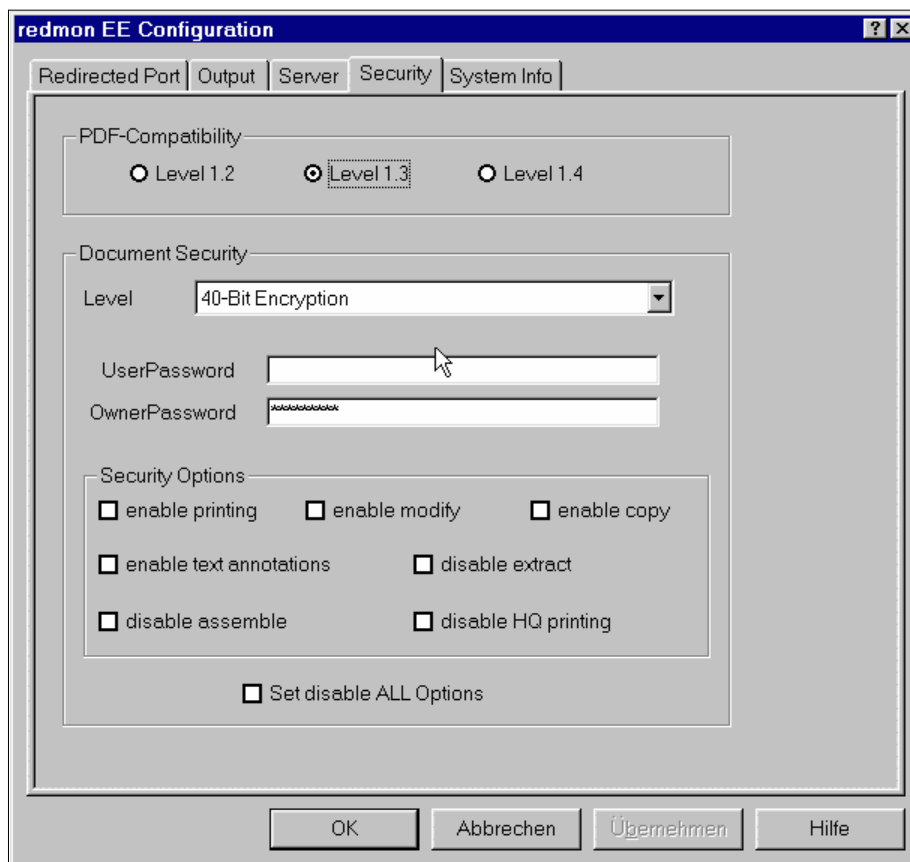
To start the SaveAs Dialog with:

1. the document name of the printer job set **Use Doc-Name**
2. the last used filename set **use Last Filename**
3. only with a directory set **use directory only**

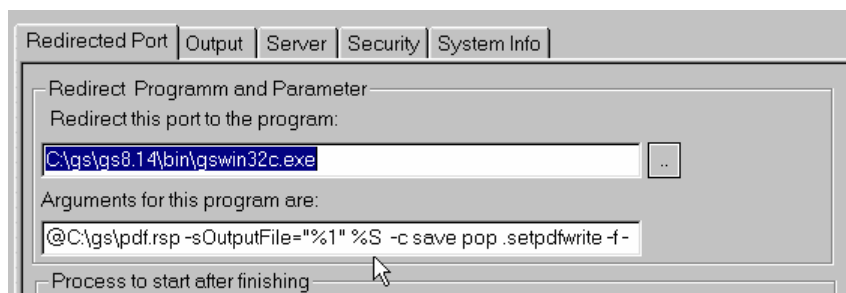


## Security Settings

[RedMon Extended Edition](#) supports PDF Security Setting. To use it You have to configurate these Setting and You have to activate it in the [Port Settings](#).



You have to add **%S** in the GhostScript Arguments to activate PDF Security. **Watch the captation !**



[Level 1.2](#)

[Level 1.3](#)

[Level 1.4](#)

[OwnerPassword](#)

[UserPassword](#)

[Security Options](#)

### Level 1.2

PDF Level 1.2 does not support Security setting !

### Level 1.3

PDF Level 1.3 Supports only 40-Bit encryption.

### Level 1.4

PDF Level 1.4 Supports only 40-Bit encryption and 128-Bit encryption of documents. At least You **must** specify the

[OwnerPassword](#) !

### OwnerPassword

Specifies the Owner's password for encryption of documents. This password **must** be specified to activate PDF security.

### UserPassword

Specifies the User's password for accessing encrypted of documents. If this password is used, user's **must** enter the password if opening the Document.

### Security Options

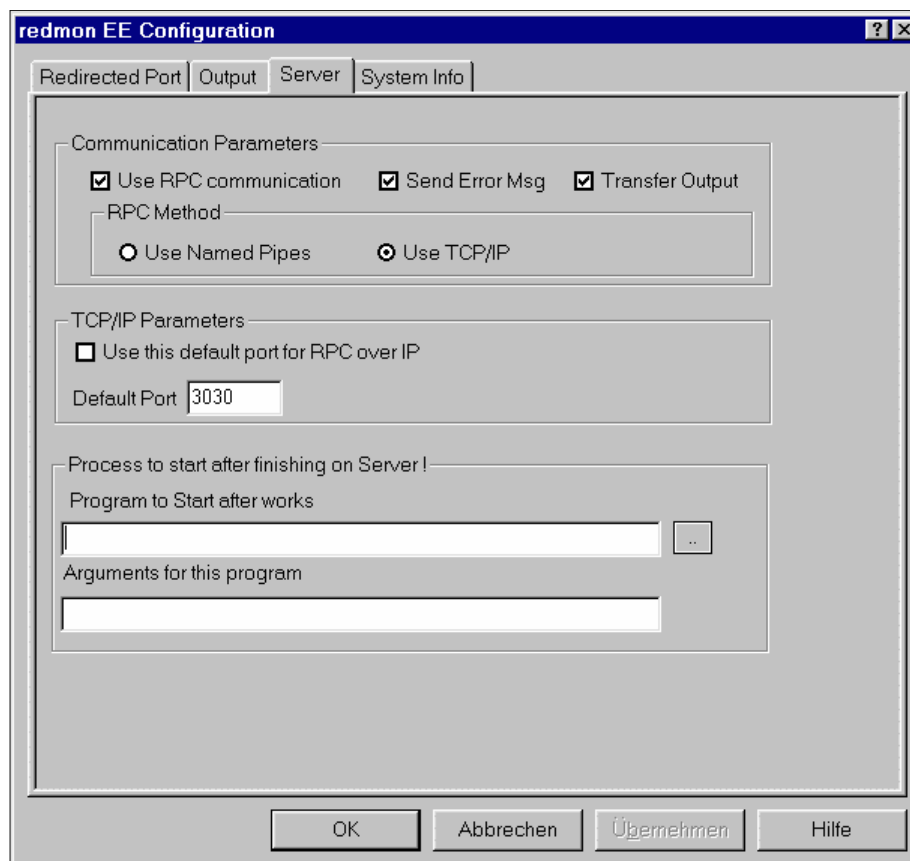
You may set the following options:

<b>enable printing</b>	Print document (possibly not at the highest quality level)
<b>enable modify</b>	Modify contents of document, except as controlled by bits 32, 256 and 1024.
<b>enable copy</b>	Copy text and graphics from document other than that controlled by bit 512
<b>enable text annotations</b>	Add or modify text annotations, fill in interactive form fields, and if bit 256 is set, create or modify interactive form fields
<b>disable extract</b>	Extract text and graphics (in support of accessibility to disabled users or for other purposes).
<b>disable assemble</b>	Assemble the document (insert, rotate, or delete pages and create bookmarks or thumbnail images), even when 16 is clear
<b>disable HQ Printing</b>	disable high quality printing
<b>disable ALL options</b>	will set all possible options to OFF



### Server Configuration

[RedMon Extended Edition](#) comes with an RPC feature which allows You to share printers running RedMon.



Set [Use RPC Communication](#) if You want to use the RPC communication feature.

The **Send Error Msg** option will send error-reports back to the client. This option is the same feature if You switch on [Print Errors](#) for local printing.

The [Transfer Output](#) option will send the generated file back to the client running the RPC-Server **redmonS.exe**

Use [RPC Method](#) to choose the RPC Communication layer. RPC could use [Named pipes](#) or [TCP/IP](#) for communication. [Named pipes](#) should be the preferred method, but for security reasons may not work because the printer runs in the system context.

To configure TCP/IP You may set a [Default port](#) which is to be used by TCP/IP.

If RedMon runs as a shared printer on a server, You may setup a external program which will be executed after the printing

has finished. Enter a modul in [Programm to Start after works](#) and the appropriate parameters for this modul.

[Use RPC Communication](#)  
[RPC Method](#)  
[Transfer Output](#)  
[Default port](#)  
[Named pipes](#)  
[TCP/IP](#)



### Use RPC Communication

You may setup RedMon to use RPC for communication with the client. The RPC communication layer may use [Named pipes](#) or [TCP/IP](#) for communication.

The **RPC feature** allows You to transfer the generated file back to the client if You set the [Transfer Output](#). To use this feature clients have to run the [RedMon RPC Server](#)



### RPC Method

Configuration of the RPC Communication layer to be used.

You may use TCP/IP or named pipes. Named pipes should be the preferred communication layer, but



### Transfer Output

This option will transfer the generated output file to back to the client. If configured the user on the client is prompted for a filename to save the output file.



### Default port

Set the default port used by the [RPC feature](#). Only needed if You are using [TCP/IP](#) for communication.



### Named pipes

Setup RPC to use named-pipe as communication layer. As an alterante You may use [TCP/IP](#) for the RPC communication layer.

"Named Pipes" should be the preferred method for RPC communications. But there may be some security reasons which will prevent You to set up RPC communication via this transportation layer.

You should test if RedMon EE works with "named pipes" first. If You get an error "Access denied" You have to switch to [TCP/IP](#) as the RPC communication layer.



### TCP/IP

Setup RPC to TCP/IP as communication layer. As an alterante You may use [Named pipes](#) for the RPC communication layer.

To setup and establish a communication, RedMon needs - as every TCP/IP communication -:

- a true IP-address e.g. 10.10.70.156
- a port e.g. 3030

But RedMon running on a remote printer will only have the following information:

- Clientname **\\client615**
- Username **anyUsery**

With the help of a DNS-service or a WINS-service running in Your network TCP/IP will be able to retrieve the ip-address 10.10.70.156 from the clientname "\\client615".

If You are using a single-user - single computer network setup You should setup a fixed port for communication. If You have a multi-user - single computer network setup RedMon will generated a port-number by calculating a CRC16 checksum from the username. This CRC16 checksum will be used as the port-number for communication.



### Common Problems

If you have problems, it is recommended that you configure the RedMon port to use a Log File with debugging turned on.

Before configuring RedMon, make sure that your redirection program runs correctly from the command line.

**Problem:** When attempting to print, a message box titled **Printers Folder** appears saying **There was an error writing to RPT1: for the printer (Apple LaserWriter II NT): The system cannot write to the specified device.**



RedMon couldn't start the program you specified. Look in the Log File for more details.

Make sure you have disabled bi-directional printer support.

**Problem:** Ghostscript locks up.

You might have forgotten to append a space and dash to the end of the Ghostscript arguments. This is needed to make Ghostscript read from standard input.

**Problem:** Ghostscript will process some jobs, but not others. If you print to FILE:, Ghostscript is able to process these jobs.

Make sure the **Spool Settings** are set to

**Disable bi-directional support for this printer.** If this is the problem, the log file with debugging turned on will include the message.

```
REDMON ReadPort: You must disable bi-directional printer support
for this printer.
```

**Problem:** Attempting to configure a port gives the error message: "This option is not available or not allowed".

You may need to configure the RedMon port from the computer running RedMon, rather than a remote computer.

**Problem:** Windows NT lpd treats data as text instead of raw.

This isn't a problem with RedMon, but you might just need this information. When printing from Unix to NT, two common methods are to use Samba to write to a RedMon port, or to use lpr to write to the lpd service on NT. The lpd service is not installed by default.

Windows NT lpd sometimes decides to treat the incoming stream as text rather than raw, causing the PostScript code to be printed instead of being interpreted. This can be fixed with an appropriate command option for lpr at the remote Unix box, or at the Windows NT end with regedit, see:

<http://www.jsiinc.com/TIP0400/rh0473.htm>

Edit the registry and search for:

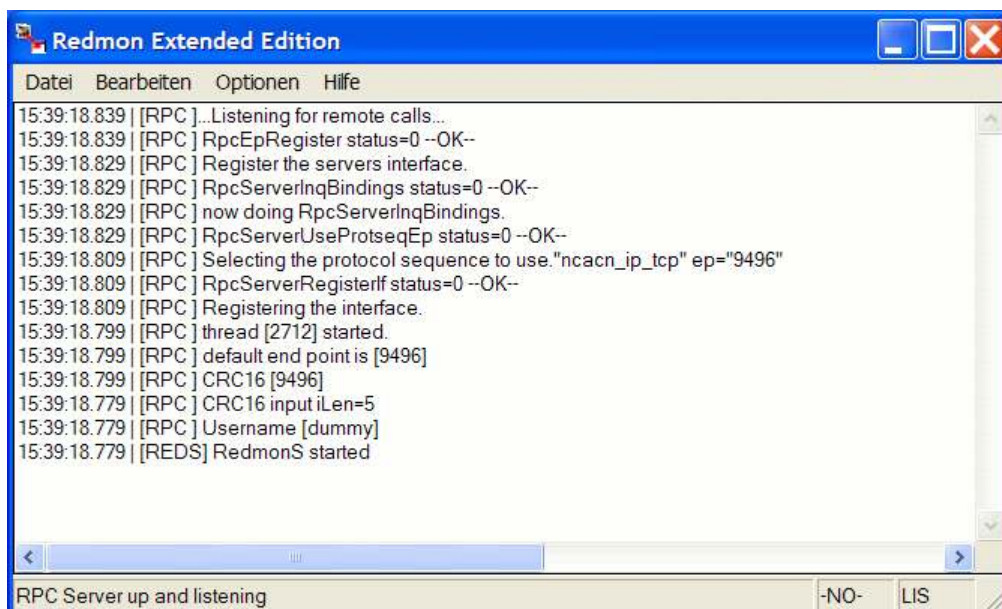
```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LPDSVC\Parameters
```

Add Value name SimulatePassThrough of type REG\_DWORD and set it to 1.



## RedMon RPC Server

To use the RPC communication feature user have to run the **RedMon RPC Server** program **redmonS.exe** on clients, which print their output on a remote print server.



### Hint:

If You use Windows XP with Service Pack 2 installed and have trouble with RPC please read:

[Microsoft Technet](#) !

May be You have to add the following key to Your registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies \Microsoft\Windows NT\RPC\RestrictRemoteClients - DWORD - VALUE 0
```

RedMonEE has changed to overcome this *feature* of SP2 with release 1.8.0.5

Output generated by a printer running RedMon could be send back to the **RedMon RPC Server**. You even can use the **RedMon RPC Server** with RedMon running on the Client.

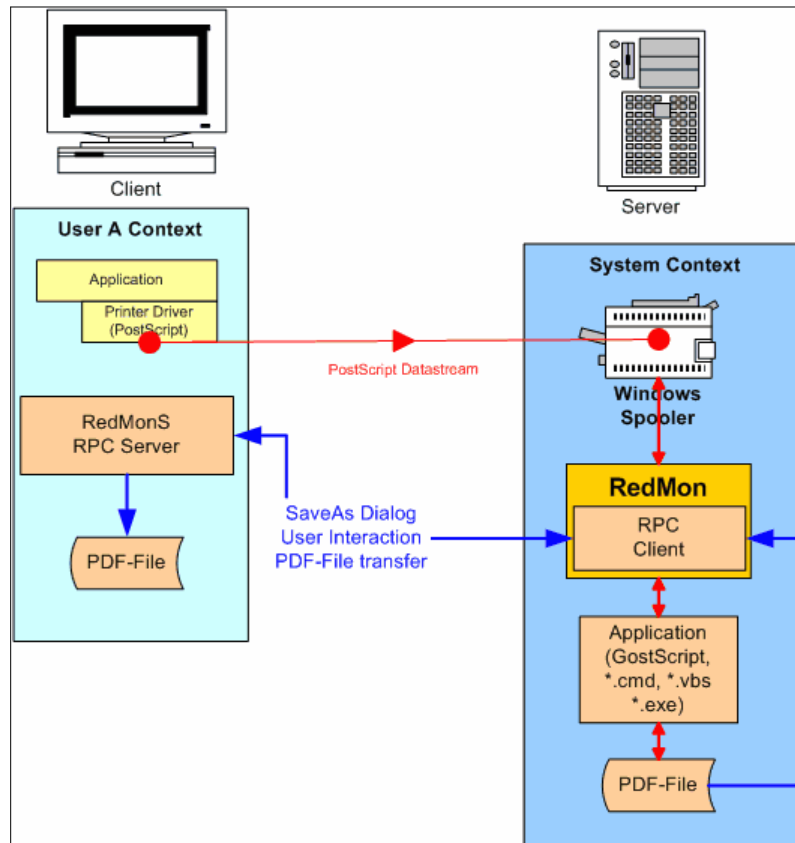
If RedMon is unable to communicate with a **RedMon RPC Server** on the client, the RPC Communications feature will not be used by RedMon, the RedMon RPC feature will be deactivated for this specific job.

If You don't need user interaction, or if the processing of the output is done on on the server The RedMon EE RPC-Server feature will not be needed.

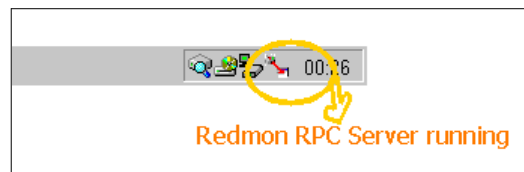
If Your users like to have their ouput back on the client You should install the redmon RPC Server on all clients using the print server.

Clients connected to a print server don't need to run RedMon or GhostScript. They only need to have RedMonS installed and setup.

The figure shows the basic operation of RedMon EE with the RPC feature installed.



The Redmon EE RPC Server could be configured to start when Your system is initialized. Users will see only a small entry in the task-bar, if a task-bar is present. Please note that some WTS-systems don't use taskbar's on the users desktop.



Before using the RPC Server, users will have to setup some parameters in the configuration menu.

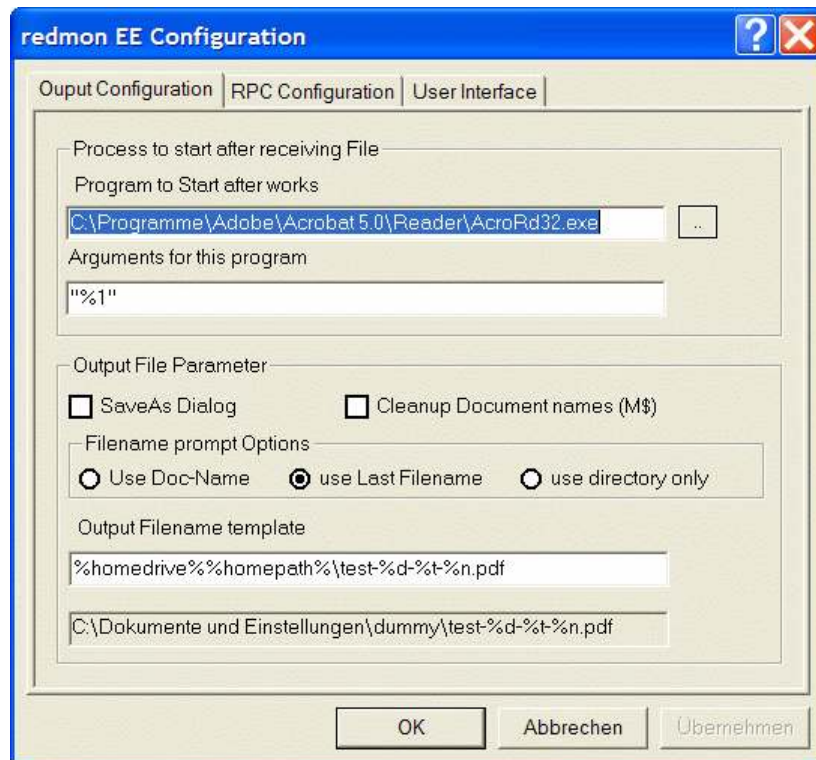
On the [RPC Output Configuration](#) page You can define a standard program that is started after the ouput from the server has been received on the client. Acrobat Reader is a good example for a process to start after the PDF-File has been **printed**.

On the [RPC configuration](#) page the basics of the RPC communication have to be defined. These settings **must** confirm to the [Server Configuration](#) of RedMon running on the print server ! If Redmon on the Server is configured to use TCP/IP and a default port, the [RPC configuration](#) on the Client have to be set to the same values. Because of the different security levels of a user and RedMon running on the print server, it's not possible to read these settings directly form the server.

On the [RPC User Interface](#) page You may setup the user-interface of RedMonS and the trace and log facility of the RCP Server.

[RPC Output Configuration](#)  
[RPC configuration](#)  
[RPC User Interface](#)  
[output file template](#)  
[Named Pipes](#)



**RPC Output Configuration**

The **RPC Output Configuration** Dialog is similar to the redmon [Output configuration](#).

Set the [Programm to Start after works](#) to any program You like to start after the print job is finished. for some users it might be a good idea to start Acrobat Reader to view the results coming out of the print job.

By setting the program arguments entry to **%1** the generated output file will be passed to the program.

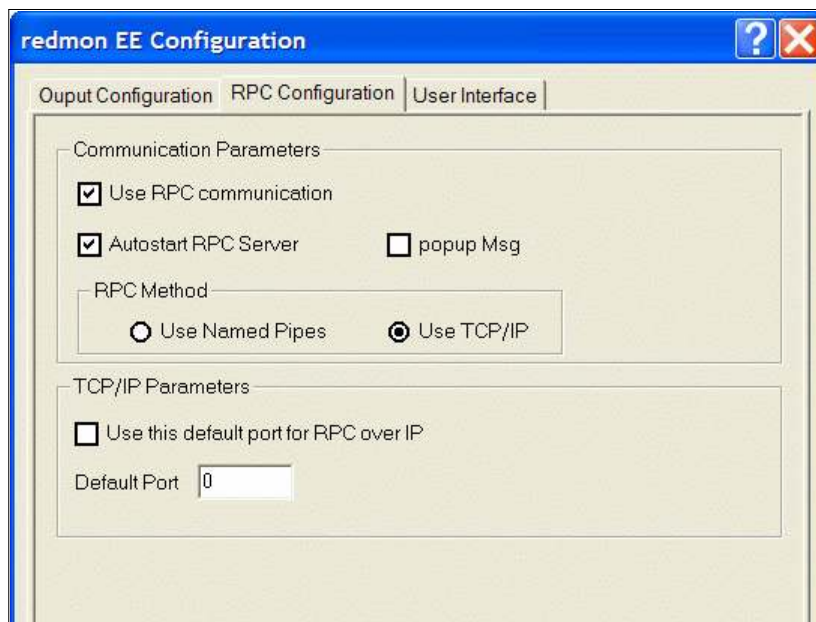
If You set **SaveAs Dialog** to on a dialog box will be presented to set the filename and path of the document generated by the print job.

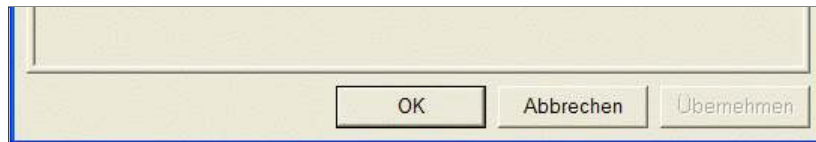
If You switch **SaveAs Dialog** to off You may define a [output file template](#) to generate fixed filenames for each and every output file received from the print server.

If You are prompted by the SaveAs Dialog, the default filename may be

- the Document name **Use Doc-Name**
- the last Filename **use Last Filename**
- the last given directory **use directory only**

To generate clean document names You may switch on [Cleanup Document names](#) to remove some strange prefixes like **Microsoft Word -**

**RPC configuration**



The RPC-Configuration is not for novice Users!

You have to set [Use RPC Communication](#) if You want to use this feature. It's useless to have redmonS up and running if You don't want to use the RPC feature. But we may have some trouble shooting situations where it could be usefull to switch the RPC feature to off.

If You set **Autostart RPC Server** the RPC server feature will be started automatically if RedMonS is started. Users may start and stop the RPC Server manually from the menu-bar.

If You would like to have Popup messages for errors from the print server set the **popup Msg** switch.

The most important thing to configure is the RPC communication layer to be used. Your setting on the client **must** match the setting on the print server !

The default value should be [Named Pipes](#) because of the advantage of a direct addressability by building the names of the pipe from computername and username, these are the only information the print server will have to identify the client submitting the print job.

Because of the nature of the print spooler, there could be some security reasons that will not allow the system context to access named pipes. If You run the RPC server and RPC client with [Named Pipes](#) from the user context everything will work. The same is not true if the RPC client runs within redmon in the system context. Therefore You may use [TCP/IP](#) as the RPC communication layer. This will work even in the system context where RedMon is executed.

**Please note:** The only address information RedMon will have on a print server are:

1. computername
2. username

this tuple of information may be sufficient to get the true IP-address (10.123.56.67) of the RPC server. To access the RedMon RPC server running on the client, the IP-address is not sufficient. We have to define a port on which the server is listening.

If Your network is designed on a one Computer on User basis You don't have any issues. The best solution is to have a **well-known** port for all clients.

If Your network is designed on a one Computer many Users basis, we have an issue because we need different ports for different users. The RPC-Client running un the print server must generate a port for each active user on the same client. To generate a port number RedMon's RPC feature uses a 16-bit-CRC code generated over the username.

You may have an issue if:

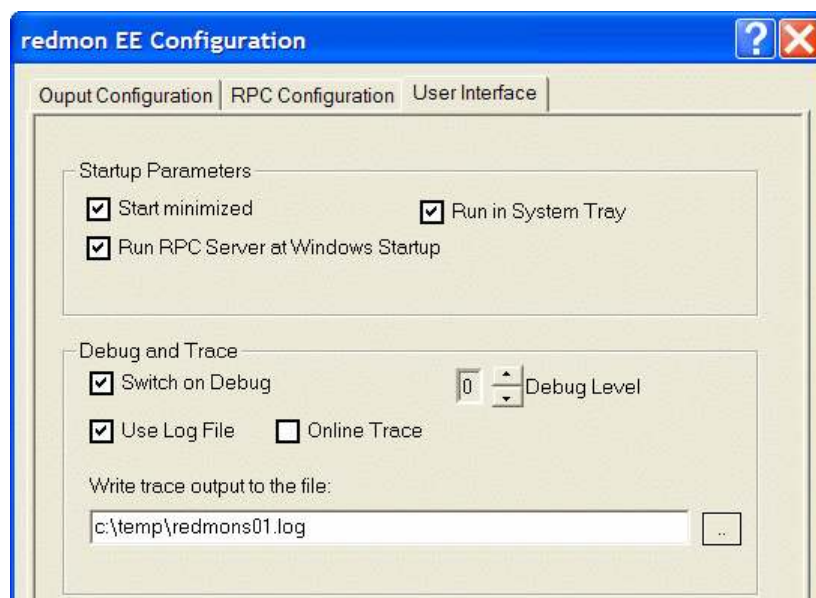
1. You don't have a WINS service running in Your network
2. and if You don't have a DNS server
3. and if You use DHCP

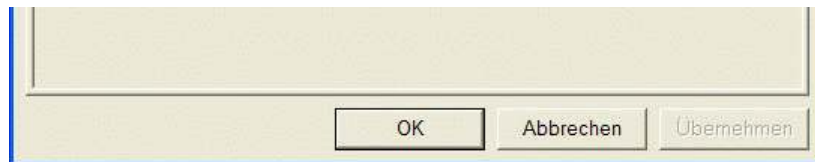
because RPC and TCP/IP may be unable to get the IP-address only from the computername.

We will have RPC-Test feature in RedMon in the next release, which will help You to setup the RPC parameters.



## RPC User Interface

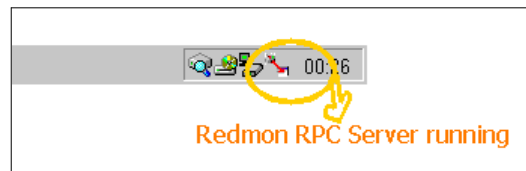




The User Interface page in the configuration dialog sets the options for the UI and debug and trace options.

The **Start minimized** option could be used to run RedMonS in the background.

The **Run in System Tray** option could be used to have a RedMonS Icon in the system tray of the taskbar. The Users will see only a small entry in the task-bar, if a task-bar is present. Please note that some WTS-systems don't use taskbar's on the users desktop.



To have the RPC Server RedMonS up and running each time windows is starting, use the **Run RPC Server at Windows Startup** switch.

In case of any trouble with the RCP-Server RedMonS set **Switch on Debug**. RedMonS will generate detailed information on what's going on under the hood. Information Messages are always written to RedMonS window. If You like to have debug and trace messages **online** the **Online Trace** switch. Debug-messages may also be written to a log-file set the log-file template in **Write trace output to file**.



#### output file template

The output file template is used to generate default filenames. You may use any environment variable as %my var" substitutions to generate this filenames.



#### Named Pipes

named pipes are the preferred method for PRC communication because we don't have any issues with addressing.



#### Copyright

Copyright (C) 1997-2000, Ghostgum Software Pty Ltd. All rights reserved.  
Copyright (C) 2004, internet-support foehr.com. All rights reserved.

This file is part of RedMon Extended Edition.

This program is distributed with NO WARRANTY OF ANY KIND. No author or distributor accepts any responsibility for the consequences of using it, or for whether it serves any particular purpose or works at all, unless he or she says so in writing. Refer to the RedMon Free Public Licence (the "Licence") for full details.

Every copy of RedMon must include a copy of the Licence, normally in a plain ASCII text file named LICENCE. The Licence grants you the right to copy, modify and redistribute RedMon, but only under certain conditions described in the Licence. Among other things, the Licence requires that the copyright notice and this notice be preserved on all copies.

RedMon:  
Author: Russell Lang, Ghostgum Software Pty Ltd.  
Internet: <http://www.ghostgum.com.au/>

RedMon Extended Edition:  
Author: Lutz Ahrens, internet-support foehr.com  
Internet: <http://www.is-foehr.com/>



000347

[impressum](#)

© Ghostgum Software Pty Ltd., is-foehr.de released: 7. 2. 2005