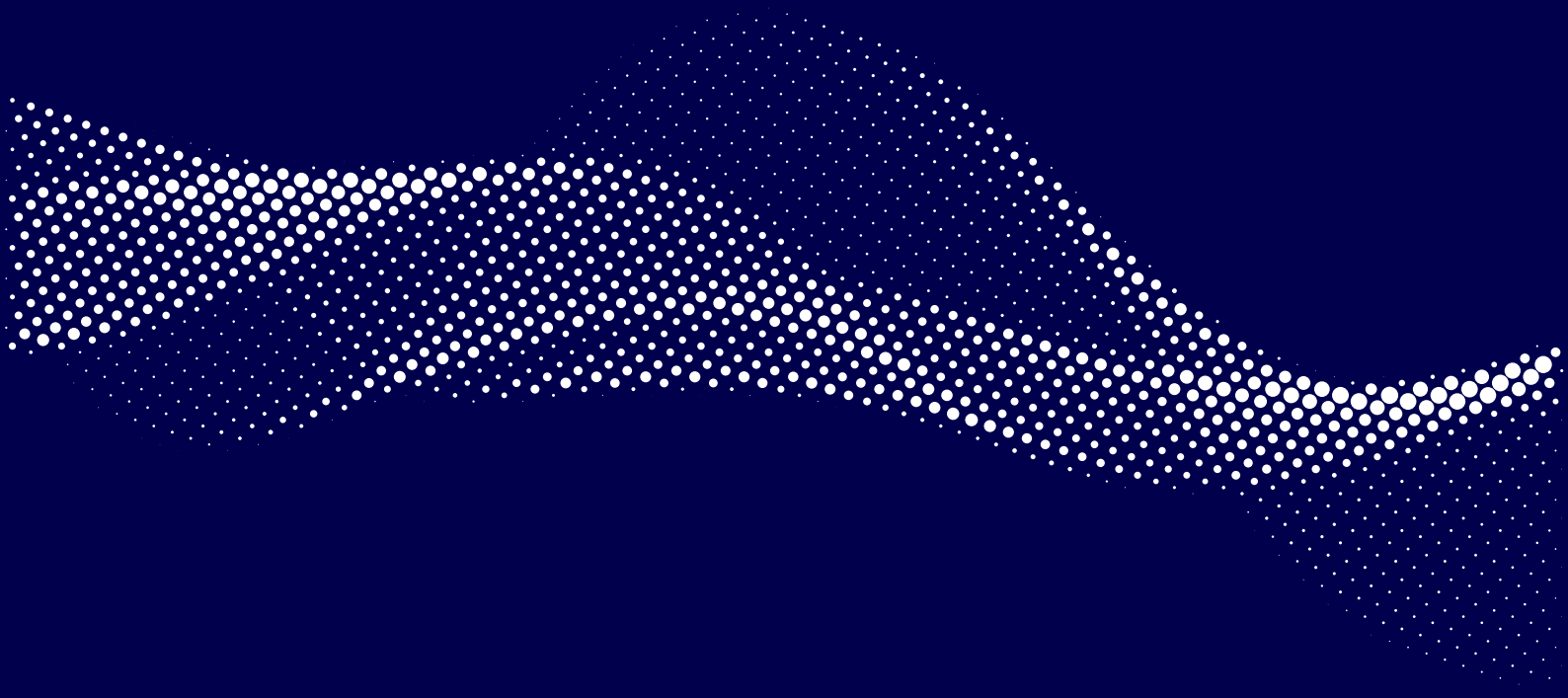

Quad Ink Descriptor File Format

# Specification

For *QuadToneRIP 2.8*

**Abstract**

This document provides a detailed description of the *Quad Ink Descriptor File* format for working with *QuadToneRIP 2.8*.

**Keywords:** black and white printing, alternative photographic printing, digital negative, EPSON printer, ink.
**Download:** www.chainick.github.io/qidf-spec/
**Discussion:**

July 2, 2022

## Change history

| Version | Date | Comment |
|---|---|---|
| 0.0.1 (current) | 2022–07–02 | Initial version. |

# Table of Contents

# Preface

Inkjet printing continues to increase in popularity as its quality now surpasses many traditional types of printing. It has finally come of age.

In some situations, you may look for greater control over printing settings than the standard printer driver provides, or you may want to use a special base (fine art papers, transparencies, etc.) that the driver does not support. In this case, a *Raster Image Processor* (RIP) may help.

*QuadToneRIP* (Quad Tone Raster Image Processor) is software used as a printer driver for certain Epson printer models[1]. *QuadToneRIP* focuses on monochrome printing. It was initially designed to make positive black and white inkjet prints. More recently, however, *QuadToneRIP* is also used to print high-quality digital negatives[2] for many alternative photographic printing processes.

*QuadToneRIP* is available for Mac OS and Windows. On the Mac, works as a printer driver. On Windows, it is a set of batch programs outputting through the standard Windows printer driver.

*QuadToneRIP* is a shareware program, for `$50`, and is available for download on quadtonerip.com. The best source of information on and about *QuadToneRIP* is the QuadToneRIP@groups.io group.

[1] The full list of all supported printers is presented in Appendix A on page 47.

[2] The *Digital Negative* is the collective name for methods used by photographers to create either negatives or positives on transparency film for the contact printing of alternative photographic techniques.

# Introduction

As mentioned, *QuadToneRIP* gives complete control over the printer's inks. It tells the printer how much of each ink to use and what color for every tone in the image. All this information about the use of inks is stored in service files called *Quad Curve Files*. A simplified printing process by *QuadToneRIP* is below:

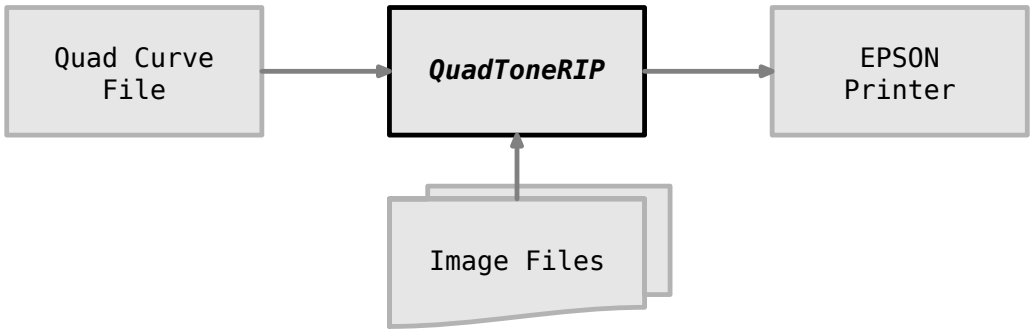


**Figure 1** Simplified illustration of the print process through *QuadToneRIP*.

Since *Quad Curve Files* are service files containing numbers only, they are not intended for human use. In order, to make the process of describing the use of ink more understandable to a user, *Quad Ink Descriptor File* is used. The receipt process of files is below:

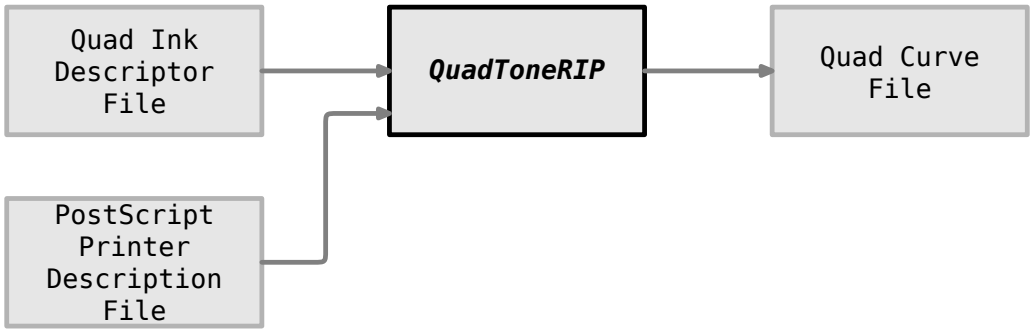


**Figure 2** The principle of generating *Quad Curve Files*.

On Mac OS, the process of obtaining a *Quad Curve File* is called an *installation* and performed by a user every time a *Quad Ink Descriptor File* is modified. On Windows, the process is called *compilation* and can

be performed automatically. Whatever the process is called, a *Quad Curve File* is always placed in the `QuadTone` directory.

A summary of the short information mentioned files is below.

***Quad Curve File*** (QUAD file, called also *QTR Curve*) is a text file, simply unsigned `16`-bit integers that signify the ink level for each of the `256` steps in the grayscale that make up an `8`-bit tonal scale. These `16`-bit numbers are the percentage of the total ink the printer is capable of spraying onto the paper at any given point. `0` is `0%` ink, and `65565` is `100%`. These combined curves set is generally called a *QTR Curve*.

QUAD files have the extension `.quad`, are stored in the `QuadTone` directory, and are the final result of installing/compiling the `.txt`/`.qidf` file profiles. These files should not be required edited manually.

***PostScript Printer Description File*** (PPD file) is a text file that contains commands used to invoke features for the print job. PPD files function as drivers for all PostScript printers. See the *PPD File Format Specification*[1] for more details.

PPD files have the extension `.ppd`, are stored in the `PPD` directory, are supplied with *QuadToneRIP*, and should not be required edited manually.

***Quad Ink Descriptor File*** (QIDF file, called also *QTR Profile* or simple a *profile*) is a plain text file that is a description of the inkset characteristics for the part of Epson printers.

The name of a QIDF file can contain only letters, digits, underscore, or dash. There is also the limitation length file name `40` characters max.

QIDF files are stored in the `Profiles` directory and can be edited either using any text editor (not a word processor) or directly through the *QuadToneRIP*ś interface.

On Windows, the files have the extension `.qidf`, and on Mac OS the extension `.txt` is used. The files' extension should be renamed if you move a file from one OS to the other.

[1] http://partners.adobe.com/public/developer/en/ps/5003.PPD_Spec_v4.3.pdf

# Working with *QuadToneRIP* on Windows

When *QuadToneRIP* is installed, two directories are created under the *QuadToneRIP* directory: `Profiles` and `QuadTone`.

The `Profiles` directory contains further subdirectories for each of the printer/ink combinations supported and each of these subdirectories contains files with `.qidf` extensions. As a rule, one QIDF file is for each paper (and neutral, cool, warm, etc. characteristics). These QIDF files are editable either as text files in a text editor or through the *QuadToneRIP* interface.

The second directory, `QuadTone`, contains the same structure and each QUAD file there corresponds to a QIDF file in the `Profiles` directory. As mentioned, QUAD files are not recommended to edit manually since these are the ones used by the application to print.

## Editing of QIDF files

There are two ways to edit a profile. The first one is using the *Curve Creation* tool of *QuadToneRIP*ś GUI under *Tools* → *Curve Creation*. (See Figure 3 on page 7.)

If you edit a QIDF file through the *Curve Creator* and save it, there is no issue since the process of saving the file in the program compiles the required QUAD file automatically.

The second way to edit a profile is using a text editor (*Notepad++*, *Sublime Text*, or whatever you prefer). However, you need to keep in mind, that if a QDIF-file is edited in a text editor, after saving the changes (no spaces in the name are allowed), it should be compiled manually into a QUAD file. About a compilation process is below.

If you don’t feel confident yourself, it’s best to edit profiles using the *Curve Creator* rather than a text editor. The *Curve Creator* will ensure the file remains internally consistent and when you’re ready to print using the profile you can compile it with the *Save*/*Save as* command.

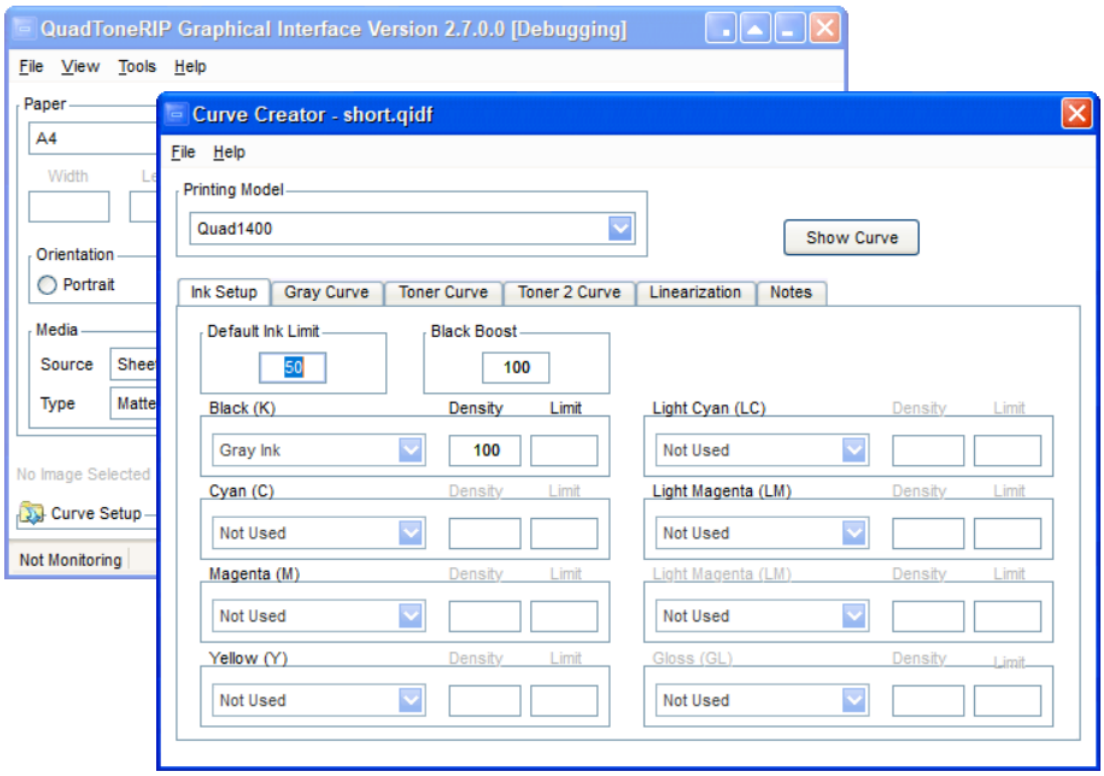


**Figure 3** *Curve Creator* is the main tool for working with QIDF files.

## Compilation of QIDF files

As mentioned, the easiest way to compile a QIDF file into a QUAD file is to use the *Save* / *Save as* command under the *Curve Creator* window. In this case, a QUAD file will be created automatically.

However, there is a way to compile a QIDF file into a QUAD file through a command line. Use for that `quadprofile.exe` supplied with *QuadToneRIP*. The syntax of using it is following:

```
d:\QuadToneRIP\bin>quadprofile.exe "..\Profiles\R200-gq\R200-cool.qidf"
```

> Where `"..\Profiles\R200-gq\R200-cool.qidf"` is a path of QIDF file. For compiling there should **not use** a PPD file as shown in Picture 2 on page 4 since *QuadToneRIP* will find and apply it automatically.

Depending on the contents of the QIDF file, the program's response may be as follows:

```
QuadProfile 2.8.0 for Windows

Creating curve R200-cool
LINEARIZE       = "94.17 90.51 86.93 84.10 81.50 73.02 68.96 64.63
60.37 55.82 52.56 48.46 41.95 35.01 29.37 23.27 16.32 12.39 5.69 4.03"

LINEARIZE_CURVE = 0,0 4.06,5 8.03,10 11.17,15 14.06,20 18.46,25 23.46,30
27.97,35 32.77,40 37.5,45 42.54,50 46.16,55 50.71,60 57.93,65 65.63,70
71.89,75 78.66,80 86.37,85 90.73,90 98.16,95 100,100
  0     0
 10  3167 **
 21  6745 *****
 31 10988 ********
 42 15099 ************
 53 17857 **************
 63 20557 ****************
 74 23667 ******************
 85 26610 ********************
 95 29334 ************************
106 32033 *************************
116 35439 ***************************
127 38774 *******************************
138 41037 ********************************
148 42641 **********************************
159 44395 ***********************************
170 46384 *************************************
180 48444 ***************************************
191 50751 ****************************************
201 52487 *****************************************
212 53943 ******************************************
223 56569 ********************************************
233 59221 **********************************************
244 60068 ***********************************************
255 65535 ************************************************
```

**Listing 1** The output of the command of QIDF file compilation.

Now, the created curve is available to use in printing out.

# Syntax of Quad Ink Descriptor Files

QIDF file is a configuration ini-style file that consists of text-based content comprising key–value pairs for properties description. It is human readable and simple to understand.

The basic element contained in a QIDF file is the key. Every key has a name and a value, delimited by an equal sign. The name appears to the left of the equal sign (`KEYNAME=value`). A key-value pair is always on a separate line. A line that starts with `#` char is commented out. Empty lines are also ignored.

## General provisions

Key names are case-insensitive. However, they are written in uppercase for better readability.

Keys without a value will be skipped or there will use a default value for some mandatory keys. This specification provides the following five mandatory keys: `GRAY_GAMMA`, `GRAY_SHADOW`, `GRAY_HIGHLIGHT`, `PRINTER`, `DEFAULT_INK_LIMIT`. However, the shortest profile must include at least the `PRINTER` key, for example:

```
PRINTER=Quad1400
```

The response of the *QuadToneRIP* after compiling on Windows the above profile is as follows (`short.qidf` file was used):

```
d:\QuadToneRIP\bin>quadprofile.exe "..\Profiles\1400-cl\short.qidf"

QuadProfile 2.8.0 for Windows

No default ink limit found, using 100
Could not find gray highlight, using 4
Could not find gray shadow, using 4
GRAY_GAMMA missing, using 1
Creating curve short
```

## Elementary types

For improved readability, this specification provides a few elementary types presented below in Table 1.

| Type | Description |
|---|---|
| `<string>` | A text comprised of any available characters except equals sign (=). |
| `<integer>` | A non-fractional number that has no sign consisting of one or more digits `0` to `9`. |
| `<real>` | A fractional number that has no sign consisting of one or more digits followed by a dot (.) or comma (,) followed by one or more digits. There are not include scientific or exponential notation. |
| `<filename>` | A text representing a file name. It can be the name of the file itself, or it might be a path to the file. |
| `<bool>` | A text representing a Boolean value that must be either `YES` or `NO`. |
| `<seq>` | A text representing a sequence (list) of values arranged in a single line and separated by a space. **A sequence must be enclosed in double quotation marks**. |
| `<ink>` | A letter code of the ink color. |

**Table 1** Elementary value types under the specification.

The codes of ink colors are defined in PPD files and expressed as presented below in Table 2.

| Code | Ink color |
|---|---|
| `B` | Blue |
| `C` | Cyan |
| `GL` | Gloss Optimizer |
| `GR` | Green |
| `K` | Black |
| `LC` | Light Cyan |
| `LK` | Light Black |
| `LLK` | Light Light Black |
| `LM` | Light Magenta |
| `M` | Magenta |
| `MK` | Matte Black |
| `OR` | Orange |
| `PK` | Photo Black |
| `R` | Red |
| `Y` | Yellow |

**Table 2** Allowed codes of ink colors under the specification.

# Allowed keys

## General keys

**Represented keys:** `PRINTER;` `LINEARIZE;` `CURVE_NAME;` `GRAPH_CURVE;` `CALIBRATION;`

### PRINTER

This key sets which printer the QIDF file is used for.

```
PRINTER=<string>
```

**Where**:

- `<string>` is a codename of a printer corresponding to the name of the PPD file typed without a file extension.

`PRINTER` is the mandatory key. If there is either no this key or not set a value, a creating QUAD file operation will fail.

See Appendix A on page 47 for the list of all allowed codenames of printers.

## LINEARIZE

This key is used to linearize a full profile, in other words, the key is affected on *Gray*, *Toner*, and *Toner 2* partitions at the same time. Linearize basically means a measure the native output from the printer and generating a correction curve to produce standardized evenly distributed output densities. There are either by using the `GRAY_CURVE` key (with a scanner of densitometer) or with the `LINEARIZE` key (requires spectrophotometer).

```
LINEARIZE="<seq>"
```

**Where**:

- `<seq>` is a sequence of density values separated by a space. Each value is `<real>`. The sequence must be enclosed in double quotation marks.

The densities can be absolute or relative to Dmin. Just list them in order from step-0 (`Dmin`) to step-100 (`Dmax`):

```
LINEARIZE="Dmin d1 d2 d3 ... Dmax"
```

The value is a constantly increasing sequence, for example:

```
LINEARIZE="0.034 0.102 0.170 0.241 0.315 0.391 0.498 0.566 0.653 0.739
0.822 0.890 0.966 1.043 1.123 1.203 1.289 1.373 1.444 1.506 1.559"
```

However, a decreasing sequence from `Dmax` to `Dmin` can be used. There is no limitation for the sequence size.

### CURVE_NAME

This key sets the curve (i.e. profile) name, whose syntax is:

```
CURVE_NAME=<string>
```

**Where**:

- `<string>` is a curve name.

`CURVE_NAME` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

### GRAPH_CURVE

This key tells the *QuadToneRIP* to show visual pseudo-graphs in the terminal every time the QIDF file is processed. (See Listing 1 on page 8 as an example on Windows.)

```
GRAPH_CURVE=<bool>
```

**Where**:

- `<bool>` is the Boolean value `YES` or `NO` telling the *QuadToneRIP* to show pseudo-graphs or not accordingly.

On Mac OS graphs will be shown for both ink curves and linearized curves but on Windows, graphs will be shown for linearized curves only.

`GRAPH_CURVE` is useful during the writing and diagnosing of profiles.

## CALIBRATION

This key indicates the profile is intended for *QuadToneRIP* calibration (i.e., printing the Ink Pattern Page).

```
CALIBRATION=<bool>
```

**Where**:

- `<bool>` is the Boolean value `YES` or `NO` telling the *QuadToneRIP* the profile is or is not intended for calibration accordingly.

The ink pattern test page `inkseparation.tif` is a very special file. Printing it out and using it is different than all the normal printing you will do with *QuadToneRIP*. The idea with this test page is to be able to print all the inks separately in raw uncorrected steps. I.e. you will see what each ink is doing on a separate 21 step graduated wedge. The first thing you will notice upon opening the file is that it is a `RGB` file instead of a grayscale. It's very important that you NOT color manage this file when you open it or when you print it. You also must not resample or resize it – it's set at 360ppi and fits on an 8.5×11 sheet of paper. The colors of the file will look strange because the `R` channel is selecting which ink and the `G` channel is the data.

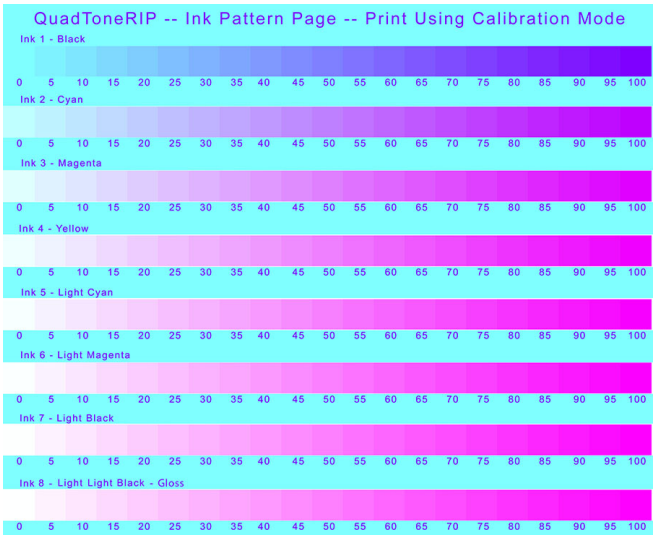


**Figure 4** The ink pattern test page.

# Ink setting up keys

**Represented keys:** DEFAULT_INK_LIMIT; BOOST_K; LIMIT_<ink>; CURVE_<ink>; N_OF_INKS; COPY_CURVE_<ink>; N_OF_UNUSED; UNUSED_INK_<integer>;

## DEFAULT_INK_LIMIT

This key sets the default value using for all inks for which the ink limit is not individually specified in LIMIT_<ink>.

```
DEFAULT_INK_LIMIT=<real>
```

**Where**:

- <real> is a default ink value (percent, in the range [0..100]).

DEFAULT_INK_LIMIT is the mandatory key. If there is either no this key or not set a value, will use its max value 100.

## BOOST_K

This key sets a new limit of ink has set as a Gray Ink 1 reaching its limit exponentially.

BOOST_K=<real>

**Where**:

- <real> is a new limit of ink (percent, in the range [0..100]).

If BOOST_K is greater than the LIMIT_<ink>, the <ink> ink will increase exponentially to the new higher value.

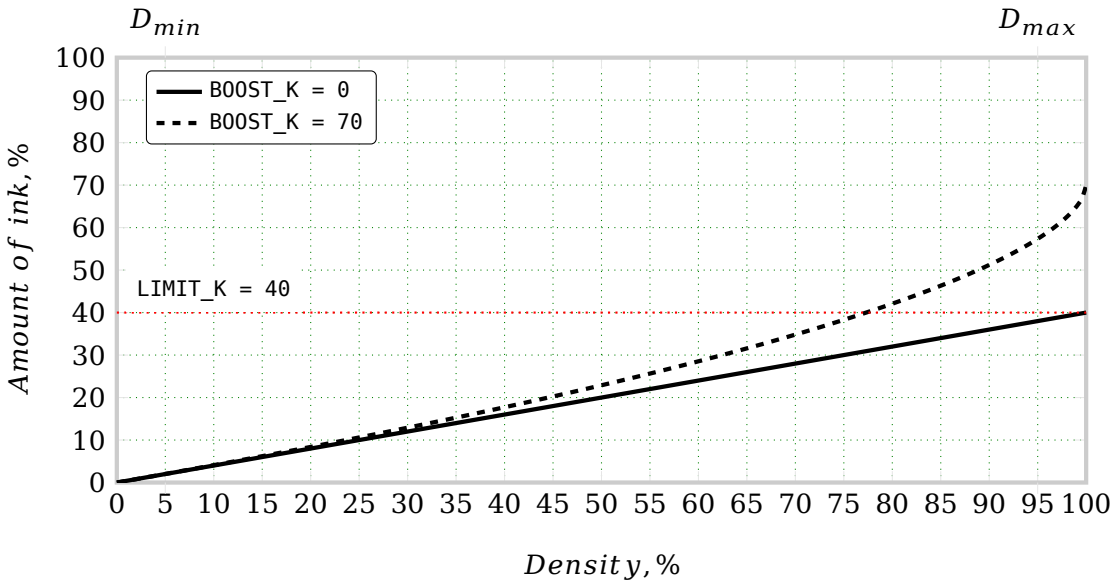


**Figure 5** Influence of the BOOST_K key on the curve in a one *Gray* part profile.

Generally, the dark black is set as the Gray Ink 1, GRAY_INK_1=K. However, any other supported ink can be used. For example:

```
# Cyan ink will increase exponentially to 80%
# despite 60% as a limit value

BOOST_K=80
LIMIT_K=50              GRAY_INK_1=C            GRAY_INK_2=K
LIMIT_C=60              GRAY_VAL_1=100          GRAY_VAL_2=50
                                                ...
```

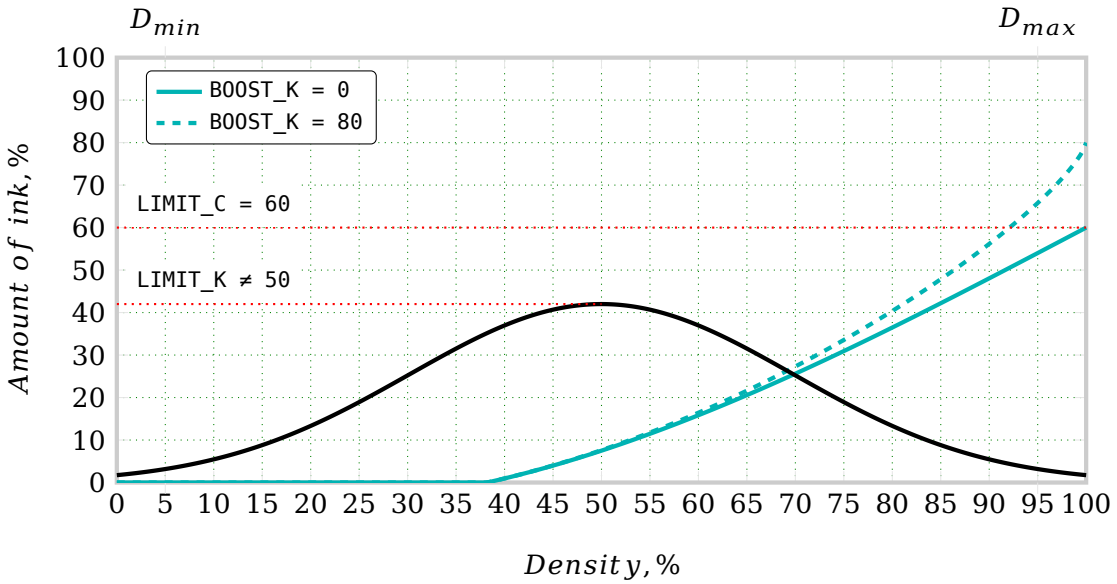


**Figure 6** Influence of the BOOST_K key on the first curve only in a two *Gray* parts profile.

You can also see above, the K curve does not reach the set limit value LIMIT_K=50. This is due to the peculiarity of the program algorithms. Read more about this in the chapter dedicated to LIMIT_<ink> key on page 18.

In the case of the black being set as the Gray Ink 1, BOOST_K increases the contrast and makes it easier to achieve pure white on the printed negative.

Even though some inks can be followed the distribution of the K ink, COPY_CURVE_<ink>=K, they will not follow the BOOST_K distribution.

Ink remapping like the one shown above (i.e. when GRAY_INK_1 is not K) is only possible using QIDF files. The *QuadToneRIP*ś GUI does not have this opportunity.

## LIMIT_<ink>

This sets the maximum limit for <ink> ink that can be used by the printer for any single dot.

LIMIT_<ink>=<real>

**Where**:

- <ink> is a code of the ink color.
- <real> is a ink limit (percent, in the range [0..100]).

The limit is a percentage of the total possible ink the printer is able to print at 100% black in an image. This is usually too much ink for the paper to absorb, therefore a lower ink limit is needed to prevent excessive density, ink bleed, or pooling on the printing media. If the limit is set to 50 then the *QuadToneRIP* will use up to 50% of the ink in that channel for creating the overlapping ink curves.

Leaving this value blank means that the limit of <ink> ink will be determined by the DEFAULT_INK_LIMIT. Setting the value to 0 would turn <ink> ink off completely. Setting this value to some number will override the DEFAULT_INK_LIMIT.

If *QuadToneRIP* includes some overlap with the adjacent inks, it won't use all 100% of the ink limit for each channel in creating the curves. It will be some percentage of whatever the ink limit is set to. The example which demonstrates that behavior for the code presented below is shown in Figure 7 on page 19:

```
...
BOOST_K=0

GRAY_INK_1=K
GRAY_INK_2=C
GRAY_INK_3=M

LIMIT_K=70
LIMIT_C=70
LIMIT_M=70
...
```

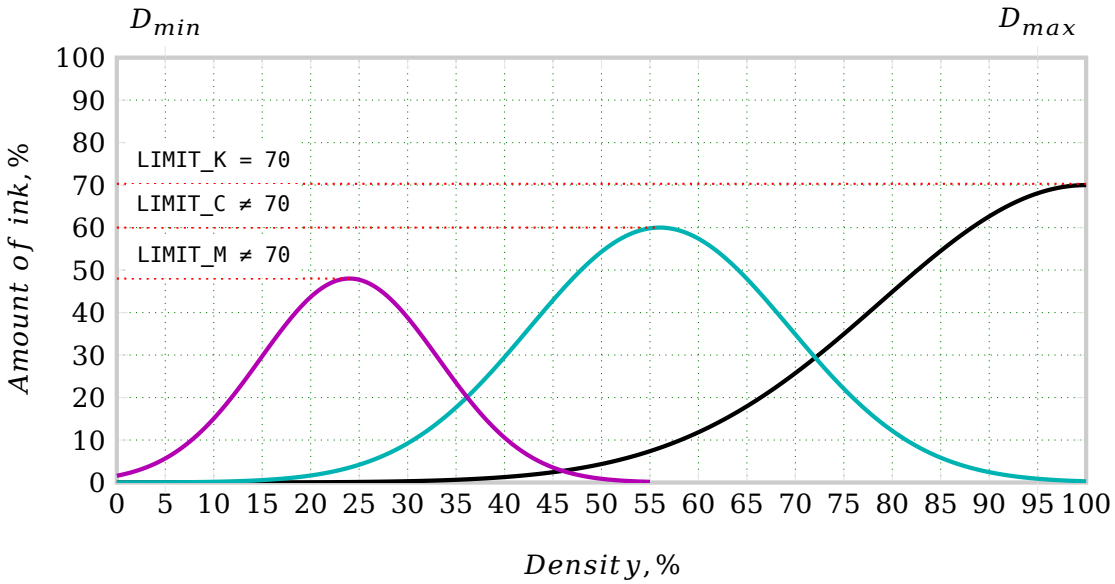


**Figure 7** In multi *Gray* parts profile, all adjacent inks except `GRAY_INK_1` do not reach `100%` of their limits.

### CURVE_<ink>

This key allows manual control of <ink> ink using a Photoshop adjustment curve or curve coordinates.

```
CURVE_<ink>=<filename>|"<seq>"
```

**Where**:

- <ink> is a code of the ink color.
- <filename> is a path to the adjustment curve file (.acv or .raw). In this case, quotation marks must not be used.
- <seq> is a sequence of curve input-output coordinates pairs. The sequence must be enclosed in double quotation marks.

As a value, there are either <filename> or <seq> types that can be used. For example:

```
CURVE_K=C:\QuadToneRIP\Profiles\2100-ut7\curve_k.acv
```

or

```
CURVE_C="0;0 18.75;3.59 50;13 60;18.1 75;30 96.5;55.5 98;81 100;100"
```

Pairs are separated by a space and coordinates in the pair are separated by a semicolon (;), and have <real> type. The first pair must be 0;0, and the last pair must be 100;100.

The advantage of using number sequences is that all curve points are in one file. These points allow more of them and more accuracy using decimal values than the Photoshop points. On the other hand, a disadvantage is that there is not a graphical representation of the curve. However, if you like the Photoshop curve editing do it with .acv's or .raw's as you are working on the curve, and then just transcribe the points later on.

The curve affects only the specified <ink> ink, and that ink is not affected by others in the profile. Though curving individual inks is not

often used, it can be useful in a one *Gray* part profile, for example, for the gelatin silver process where all inks follow K.

If black is specified as the Gray Ink 1, `GRAY_INK_1=K`, and `GRAY_CURVE` is being used, you cannot also use a `CURVE_K` key at the same time. It can only be specified once.

## N_OF_INKS

This key tells the *QuadToneRIP* a number of inks in the printer's inkset. it is created automatically when working with the *QuadToneRIP*ś GUI on Windows.

```
N_OF_INKS=<integer>
```

**Where**:

- `<integer>` is a number of inks in the printer.

`N_OF_INKS` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

The number and names of inks are taken from the PPD file. *QuadToneRIP* supports printers with the following number of inks in the printer's inkset: `4`, `6`, `7`, `8`, `10`.

## COPY_CURVE_<ink>

In a multi *Gray* part system, this key tells which ink to follow which part.

COPY_CURVE_<ink_1>=<ink_2>

**Where**:

- <ink_1> is a code of the ink color that is setting up.
- <ink_2> is a code of the ink color that will <ink_1> be followed on.

This key tells the printer to distribute <ink_1> ink exactly as it distributes <ink_2> ink.

If LIMIT_<ink_1> is different than that for LIMIT_<ink_2>, both ink curves will have the same shape but their amplitude will be different.

COPY_CURVE_<ink>=K ignores the effect that BOOST_K has on K ink distribution.

For example, there are two parts in a two-part profile: K as Gray Part 1, GRAY_INK_1=K, and LK as Gray Part 2, GRAY_INK_2=LK. Black (K) is most often the first part and Light Black (LK) is most often the second part. Dark inks (C, M, Y) often follow black (K) and light inks (LC, LM, LLK) often follow light black (LK):

```
COPY_CURVE_C=K
COPY_CURVE_M=K
COPY_CURVE_Y=K

COPY_CURVE_LC=LK
COPY_CURVE_LM=LK
COPY_CURVE_LLK=LK
```

### N_OF_UNUSED

This key tells the *QuadToneRIP* a number of unused inks.

```
N_OF_UNUSED=<integer>
```

**Where**:

- `<integer>` is a number of unused inks.

`N_OF_UNUSED` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

### UNUSED_INK_<integer>

This key sets an unused ink in the profile.

```
UNUSED_INK_<integer>=<ink>
```

**Where**:

- `<integer>` is an order number of an unused ink.
- `<ink>` is a code color of an unused ink.

`UNUSED_INK_<integer>` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

`<ink>` ink is considered used then one of the following conditions is met:

1. `GRAY_INK_1` and `GRAY_VAL_1` keys are used.
2. `COPY_CURVE_<ink>` key is used.
3. `LIMIT_<ink>` key not equal `0`.

Failure of all these conditions at the same time makes the `<ink>` ink unused.

## Gray partition keys

**Represented keys:** `N_OF_GRAY_PARTS;` `GRAY_INK_<integer>;` `GRAY_VAL_<integer>;` `GRAY_CURVE;` `GRAY_HIGHLIGHT;` `GRAY_SHADOW;` `GRAY_GAMMA;` `GRAY_OVERLAP;`

*Gray* in QTR profiles can, in principle, be presented with anywhere from one to ten parts depending on the number of inks in the printer.

Multi *Gray* parts approach gives the feeling of smooth tones since, throughout most of the tonal range, the two or more inks are either covering much of the substrate or overlapping each other. Each *Gray*'s ink must be defined by two keys: `GRAY_INK_<integer>` and `GRAY_VAL_<integer>`.

## N_OF_GRAY_PARTS

This key tells the *QuadToneRIP* a number of *Gray* parts defined in the profile.

```
N_OF_GRAY_PARTS=<integer>
```

**Where**:

- `<integer>` is a number of *Gray* parts.

`N_OF_GRAY_PARTS` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

The number of *Gray* parts is taken from the number of `GRAY_INK_<integer>` keys. *QuadToneRIP 2.8* supports from `1` to `10` *Gray* parts defined in the profile however, it cannot be greater than a number of inks that use printer supports.

```
PRINTER=Quad1400
DEFAULT_INK_LIMIT=70

GRAY_INK_1=K
GRAY_VAL_1=100
GRAY_INK_2=C
GRAY_VAL_2=50

GRAY_HIGHLIGHT=0
GRAY_SHADOW=0
GRAY_OVERLAP=0
GRAY_GAMMA=1
```

**Listing 2** The example of the simplest a two *Gray* parts profile.

### GRAY_INK_<integer>

This key defines specific one ink as a *Gray* part.

GRAY_INK_<integer>=<ink>

**Where**:

- <integer> is an order number of *Gray* part in the range [1..10].
- <ink> is a code of the ink color used as a *Gray*.

There are possible multi *Gray* parts in the profile. The darkest ink (as a rule it is the black K) is generally the first *Gray* part. Whatever ink is defined as Gray Ink 1 will also be the ink that is affected by the BOOST_K key.

Each *Gray*'s ink must be defined by two keys: GRAY_INK_<integer> and GRAY_VAL_<integer>. (See Listing 2 on page 25.)

### GRAY_VAL_<integer>

This key sets the relative density for ink defined as a *Gray* part.

```
GRAY_VAL_<integer>=<real>
```

**Where**:

- `<integer>` is an order number of *Gray* part in the range `[1..10]`.
- `<real>` is a density (percent, in the range `[0..100]`).

The value of this key is the cross-over point and designates how much of the grayscale is made up by each channel. Since these settings determine how the gray inks intersect and overlap they are some of the more important settings in profile creating.

Each *Gray*'s ink must be defined by two keys: `GRAY_INK_<integer>` and `GRAY_VAL_<integer>`. (See Listing 2 on page 25.)

## GRAY_CURVE

This key is used to linearize the *Gray* partition of a profile. To linearize a full profile the `LINEARIZE` key is used.

```
GRAY_CURVE=<filename>|"<seq>"
```

**Where**:

- `<filename>` is a path to the adjustment curve file (`.acv` or `.raw`). In this case, quotation marks must not be used.
- `<seq>` is a sequence of curve input-output coordinates pairs. The sequence must be enclosed in double quotation marks.

As a value, there are either `<filename>` or `<seq>` types that can be used.

Pairs are separated by a space and coordinates in the pair are separated by a semicolon (`;`), and have `<real>` type. The first pair must be `0;0`, and the last pair must be `100;100`.

When the `GRAY_CURVE` key is used the measurements can be carried out by using a scanner or densitometer. In the case of using the `LINEARIZE` key, a spectrophotometer is recommended.

## GRAY_HIGHLIGHT

This key controls ink deposition in the highlights of a print. Increasing the value, makes the light tones lighter and makes the curves of *Gray* inks steeper in this area.

GRAY_HIGHLIGHT=<real>

**Where**:

- <real> is a *Gray* highlight value in the range [0..10000]. If there is either no this key or not set a value, will use default value 4.

GRAY_HIGHLIGHT is the mandatory key. Fractional values should be typed with a zero before the decimal point or they will not be recognized.

This setting does not modify the total amount of ink in used in the lighter dilutions, but it does move them down the tonal scale.

GRAY_HIGHLIGHT, GRAY_SHADOW, and GRAY_GAMMA keys are used to push the ink distribution curves into rough linearity.

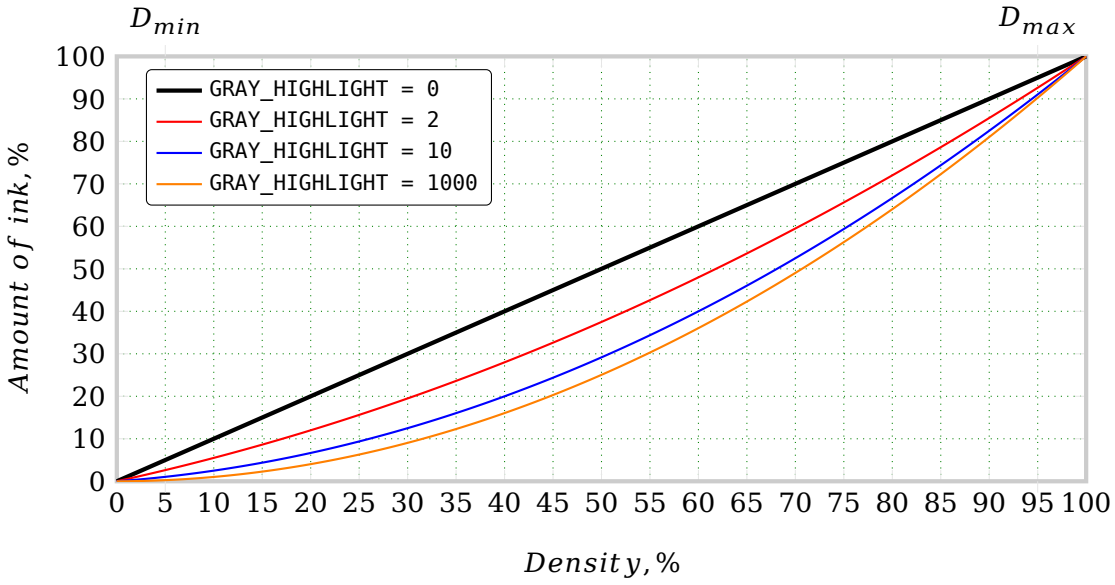


**Figure 8** Influence of the GRAY_HIGHLIGHT key on the K curve in a one *Gray* part profile.

## GRAY_SHADOW

This key controls ink deposition in the shadows of a print. Increasing the value, makes the shadow tones lighter and makes the curves of *Gray* inks steeper in this area.

```
GRAY_SHADOW=<real>
```

**Where**:

- `<real>` is a *Gray* shadow value in the range `[0..10000]`. If there is either no this key or not set a value, will use default value 4.

`GRAY_SHADOW` is the mandatory key. Fractional values should be typed with a zero before the decimal point or they will not be recognized.

`GRAY_HIGHLIGHT`, `GRAY_SHADOW`, and `GRAY_GAMMA` keys are used to push the ink distribution curves into rough linearity.

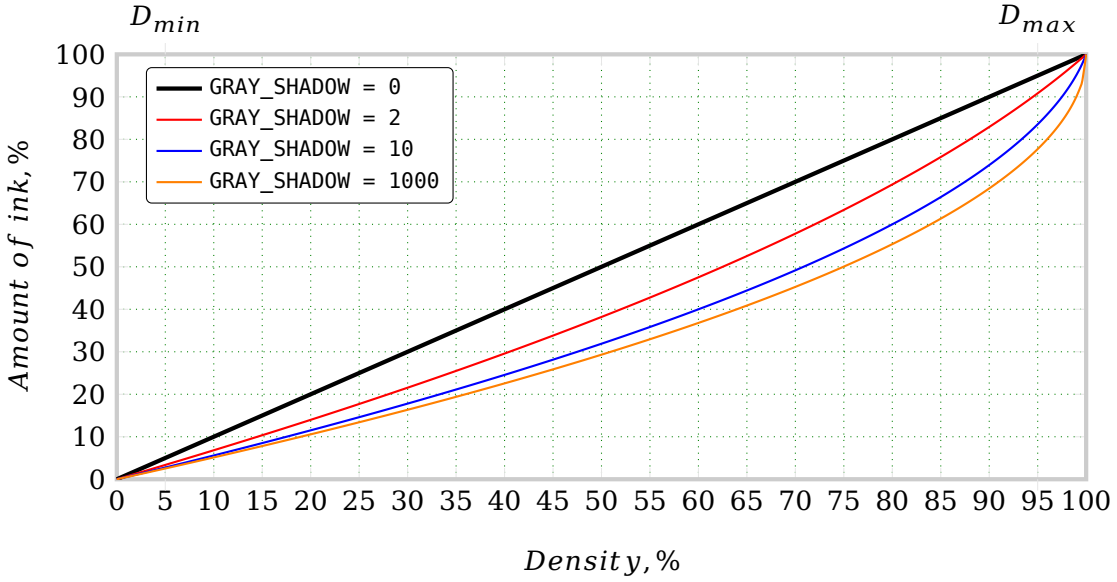


**Figure 9** Influence of the `GRAY_SHADOW` key on the `K` curve in a one *Gray* part profile.

## GRAY_GAMMA

This key controls ink deposition in the middle areas of a print. Increasing the value, makes the middle tones lighter and makes curves of *Gray* inks steeper in this area.

GRAY_GAMMA=<real>

**Where**:

- <real> is a *Gray* gamma value in the range [0.1..10], and changed in 0.1 unit increments. If there is either no this key or not set a value, will use default value 1.

GRAY_GAMMA is the mandatory key. Fractional values should be typed with a zero before the decimal point or they will not be recognized.

GRAY_HIGHLIGHT, GRAY_SHADOW, and GRAY_GAMMA keys are used to push the ink distribution curves into rough linearity.

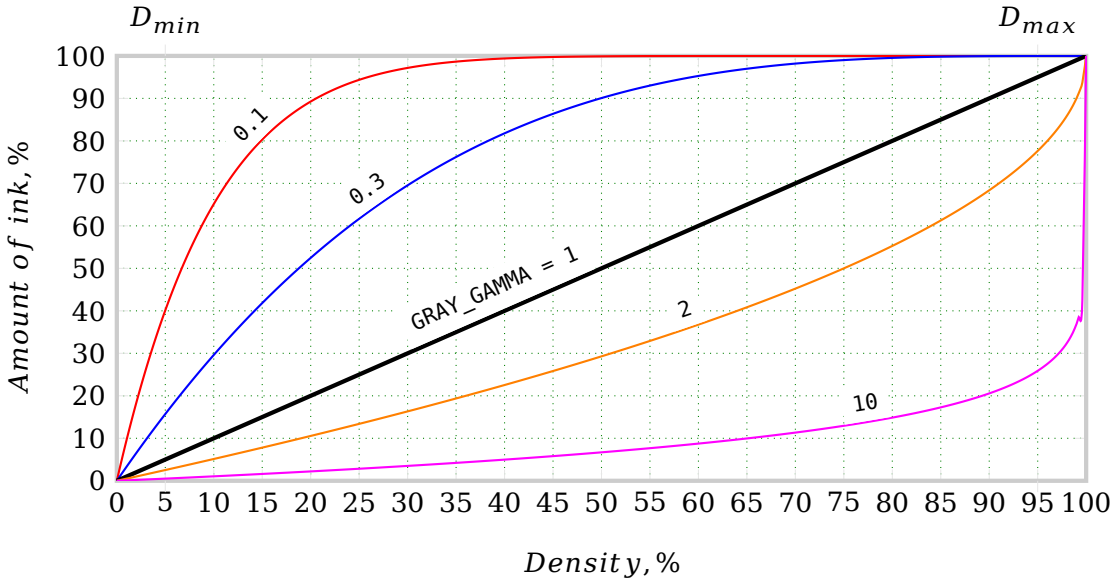


**Figure 10** Influence of the GRAY_GAMMA key on the K curve in a one *Gray* part profile.

### GRAY_OVERLAP

In some cases, it is useful to have more inks overlapping another than what is created by default in the program. The `GRAY_OVERLAP` key controls how much additional overlap of each shade of gray inks is added above the built in *QuadToneRIP*ś formula. In other words, this key controls the point in density values where *Gray* inks will cross over.

```
GRAY_OVERLAP=<real>
```

**Where**:

- `<real>` is a *Gray* overlap value in the range `[0..100]`. If there is not set a value, will use default value `100`.

To minimize the appearance of bands or sharp edges in the ink curves each ink to overlap to smooth the slope and elongate overlap with the next darker ink. This creates a much more gradual ink distribution.

For profiles with three or more *Gray* parts the `GRAY_OVERLAP` key is often used to smooth out transitions from one part to another. For example, the view of curves for a three *Gray* parts profile from Listing 2 on page 25 is below:

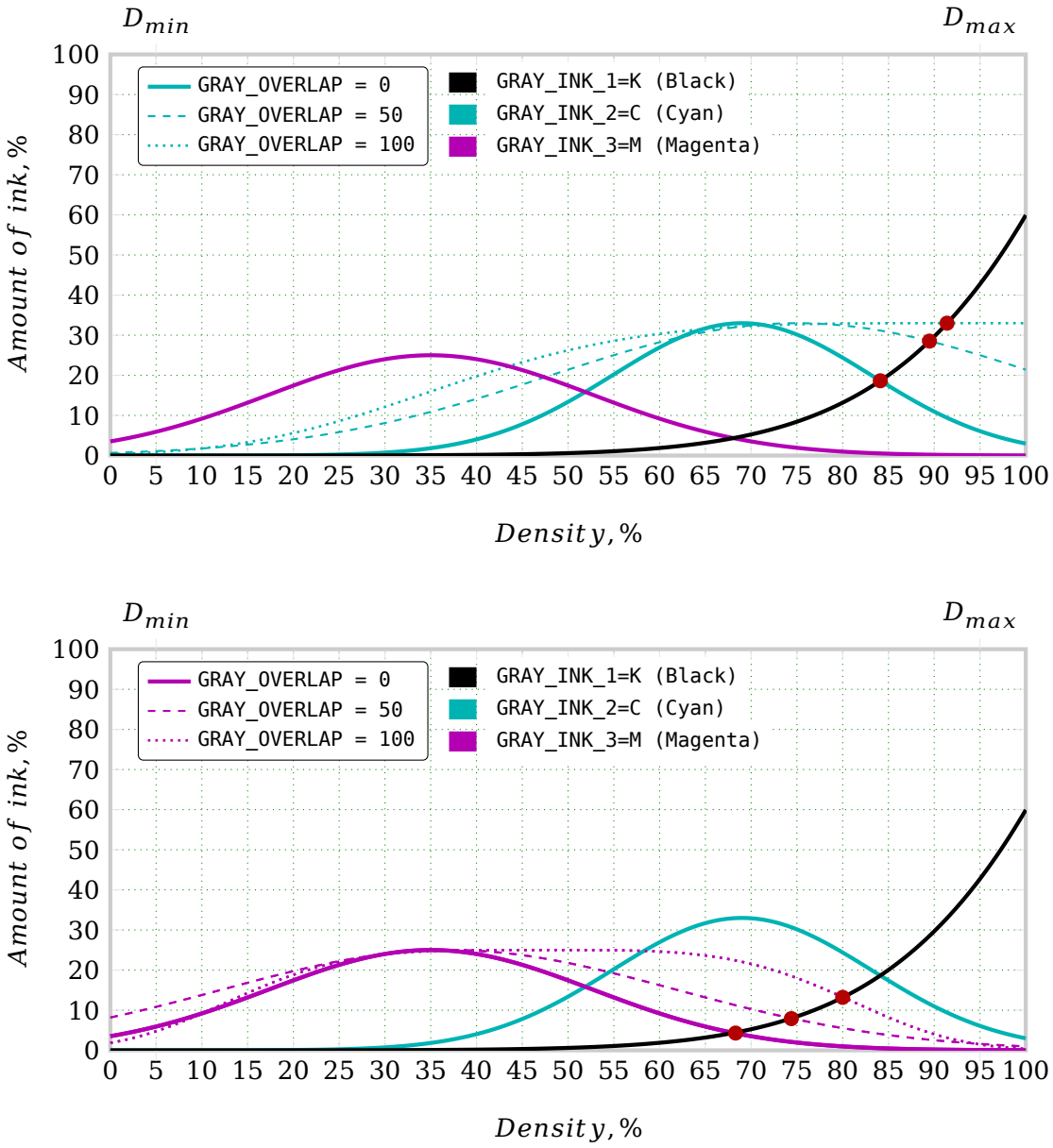


**Figure 11** Influence of the GRAY_OVERLAP key on the curves of the three *Gray* parts profile.

If a negative creating, as GRAY_OVERLAP is increased, the ink curves spread out and overlap with preceding and succeeding inks, useful for smoother transitions, to reduce grain, or to provide greater density in the negative. Too much overlap can lead to too much ink and puddling on the transparency film.

## Toner partition keys

**Represented keys:** N_OF_TONER_PARTS; TONER_INK_<integer>; TONER_VAL_<integer>; TONER_HIGHLIGHT; TONER_SHADOW; TONER_GAMMA; TONER_CURVE; N_OF_TONER_2_PARTS; TONER_2_INK_<integer>; TONER_2_VAL_<integer>; TONER_2_HIGHLIGHT; TONER_2_SHADOW; TONER_2_GAMMA; TONER_2_CURVE;

The concept of a toner is similar to the *Gray*. The *Gray*, *Toner*, and *Toner 2* can be considered as three separate gray partitions or *channels* that can be applied separately throughout the whole grayscale. These are then all can be linearized to create a single consistent density increase at the final stage of the curve creation process.

The bulk of the grayscale is usually made up of the partitioned gray inks with their higher ink limits. The toning inks and toning partitions usually have lower ink limits and run underneath the main *Gray* partition. Although, there is no reason why you can't have two toner partitions with higher ink limits that make up the whole of the grayscale (it illustrates the multiple possibilities when using toning profiles).

The two different toner curves can be used together (or separately) to put different color inks into a mix or partition two or more *Gray*'s inks of a different hue for toning the main gray partition. These toning curves can be partitioned from any number of inks, just like the gray curve, and they use the same method of partitioning using cross-over points. The toning curves also use the same x_HIGHLIGHT, x_SHADOW, x_GAMMA, and x_CURVE keys to define the partitioned gray curve.

```
...
GRAY_INK_1=K            TONER_INK_1=C           TONER_2_INK_1=Y
GRAY_VAL_1=100          TONER_VAL_1=60          TONER_2_VAL_1=5
LIMIT_K=40              LIMIT_C=15              LIMIT_Y=10

                        TONER_INK_2=LC
                        TONER_VAL_2=100
                        LIMIT_LC=15             ...
```

**Listing 3** The example of creating three partitions as the *Gray*, *Toner*, and *Toner 2* channels.

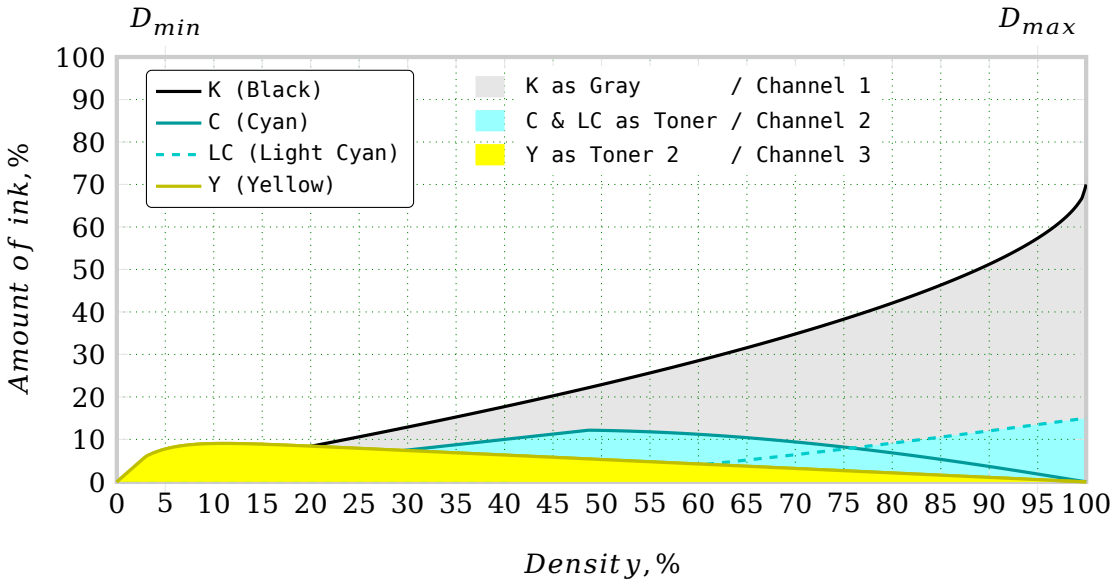


**Figure 12** *Gray*, *Toner*, and *Toner 2* as a channels concept.

Alternatively, each toner can be set to use a single ink, un-partitioned, that runs the whole length of the scale underneath the main partitioned *Gray*'s inks.

Options for *Gray*, *Toner*, and *Toner 2* are important and are how the *QuadToneRIP* knows what to do with each ink and where to put it in the final set of curves.

This approach is used for a black and white photography, and is not used so much for negative prints.

## N_OF_TONER_PARTS

This key tells the *QuadToneRIP* a number of Toner parts defined in the profile.

```
N_OF_TONER_PARTS=<integer>
```

**Where**:

- `<integer>` is a number of Toner parts.

`N_OF_TONER_PARTS` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

The number of Toner parts is taken from the number of `TONER_INK_<integer>` keys. *QuadToneRIP 2.8* supports from `1` to `10` Toner parts defined in the profile.

Each Toner ink must be defined by two keys: `TONER_INK_<integer>` and `TONER_VAL_<integer>`.

## TONER_INK_<integer>

This key defines specific one ink as a Toner.

```
TONER_INK_<integer>=<ink>
```

**Where**:

- `<integer>` is an order number of the Toner in the range `[1..10]`.
- `<ink>` is a code of the ink color used as the Toner.

Each Toner ink must be defined by two keys: `TONER_INK_<integer>` and `TONER_VAL_<integer>`.

## TONER_VAL_<integer>

This key sets the relative density for ink defined as a Toner.

```
TONER_VAL_<integer>=<real>
```

**Where**:

- `<integer>` is an order number of the Toner in the range `[1..10]`.
- `<real>` is a density (percent, in the range `[0..100]`).

Each Toner ink must be defined by two keys: `TONER_INK_<integer>` and `TONER_VAL_<integer>`.

## TONER_HIGHLIGHT

This key controls ink deposition in the highlights of a print. Increasing the value, makes the light tones lighter and makes the curves of *Toner* inks steeper in this area.

```
TONER_HIGHLIGHT=<real>
```

**Where**:

- `<real>` is a of *Toner* highlight value in the range `[0..10000]`. If there is either no this key or not set a value, will use default value `4`.

The `TONER_HIGHLIGHT` key is similar to the `GRAY_HIGHLIGHT` except it is limited to the *Toner* partition scope only.

## TONER_SHADOW

This key controls ink deposition in the shadows of a print. Increasing the value, makes the shadow tones lighter and makes the curves of *Toner* inks steeper in this area.

```
TONER_SHADOW=<real>
```

**Where**:

- `<real>` is a *Toner* shadow value in the range `[0..10000]`. If there is either no this key or not set a value, will use default value `4`.

The `TONER_SHADOW` key is similar to the `GRAY_SHADOW` except it is limited to the *Toner* partition scope only.

## TONER_GAMMA

This key controls ink deposition in the middle areas of a print. Increasing the value, makes the middle tones lighter and makes the curves of *Toner* inks steeper in this area.

```
TONER_GAMMA=<real>
```

**Where**:

- `<real>` is a *Toner* gamma value in the range `[0.1..10]`, and changed in `0.1` unit increments. If there is either no this key or not set a value, will use default value `1`.

The `TONER_GAMMA` key is similar to the `GRAY_GAMMA` except it is limited to the *Toner* partition scope only.

## TONER_CURVE

This key is used to linearize the Toner partition of a profile. To linearize a full profile the `LINEARIZE` key is used.

```
TONER_CURVE=<filename>|"<seq>"
```

**Where**:

- `<filename>` is a path to the adjustment curve file (`.acv` or `.raw`). In this case, quotation marks must not be used.
- `<seq>` is a sequence of curve input-output coordinates pairs. The sequence must be enclosed in double quotation marks.

As a value, there are either `<filename>` or `<seq>` types that can be used.

Pairs are separated by a space and coordinates in the pair are separated by a semicolon (`;`), and have `<real>` type. The first pair must be `0;0`, and the last pair must be `100;100`.

When the `TONER_CURVE` key is used the measurements can be carried out by using a scanner or densitometer. In the case of using the `LINEARIZE` key, a spectrophotometer is recommended.

## N_OF_TONER_2_PARTS

This key tells the *QuadToneRIP* a number of Toner 2 parts defined in the profile.

```
N_OF_TONER_2_PARTS=<integer>
```

**Where**:

- `<integer>` is a number of Toner 2 parts.

`N_OF_TONER_2_PARTS` is the legacy key unaffecting to the work of *QuadToneRIP*. However, it could be used to improve profile readability. Any mention of this key will be ignored.

The number of Toner 2 parts is taken from the number of `TONER_2_INK_<integer>` keys. *QuadToneRIP 2.8* supports from `1` to `10` Toner 2 parts defined in the profile.

Each Toner 2 ink must be defined by two keys: `TONER_2_INK_<integer>` and `TONER_2_VAL_<integer>`.

See `N_OF_TONER_PARTS` section for more details, `N_OF_TONER_2_PARTS` works the same.

## TONER_2_INK_<integer>

This key defines specific one ink as a Toner 2.

```
TONER_2_INK_<integer>=<ink>
```

**Where**:

- `<integer>` is an order number of the Toner 2 in the range `[1..10]`.
- `<ink>` is a code of the ink color used as the Toner 2.

Each Toner 2 ink must be defined by two keys: `TONER_2_INK_<integer>` and `TONER_2_VAL_<integer>`.

See `TONER_INK_<integer>` section for more details, `TONER_2_INK_<integer>` works the same.

## TONER_2_VAL_<integer>

This key sets the relative density for ink defined as a Toner 2.

```
TONER_2_VAL_<integer>=<real>
```

**Where**:

- `<integer>` is an order number of the Toner 2 in the range `[1..10]`.
- `<real>` is a density (percent, in the range `[0..100]`).

Each Toner 2 ink must be defined by two keys: `TONER_2_INK_<integer>` and `TONER_2_VAL_<integer>`.

See `TONER_VAL_<integer>` section for more details, `TONER_2_VAL_<integer>` works the same.

### TONER_2_HIGHLIGHT

This key controls ink deposition in the highlights of a print. Increasing the value, makes the light tones lighter and makes the curves of *Toner 2* inks steeper in this area.

```
TONER_2_HIGHLIGHT=<real>
```

**Where**:

- `<real>` is a of *Toner 2* highlight value in the range `[0..10000]`. If there is either no this key or not set a value, will use default value 4.

The `TONER_2_HIGHLIGHT` key is similar to the `GRAY_HIGHLIGHT` except it is limited to the *Toner 2* partition scope only.

### TONER_2_SHADOW

This key controls ink deposition in the shadows of a print. Increasing the value, makes the shadow tones lighter and makes the curves of *Toner 2* inks steeper in this area.

```
TONER_2_SHADOW=<real>
```

**Where**:

- `<real>` is a *Toner 2* shadow value in the range `[0..10000]`. If there is either no this key or not set a value, will use default value 4.

The `TONER_2_SHADOW` key is similar to the `GRAY_SHADOW` except it is limited to the *Toner 2* partition scope only.

## TONER_2_GAMMA

This key controls ink deposition in the middle areas of a print. Increasing the value, makes the middle tones lighter and makes the curves of *Toner 2* inks steeper in this area.

```
TONER_2_GAMMA=<real>
```

**Where**:

- `<real>` is a *Toner 2* gamma value in the range `[0.1..10]`, and changed in `0.1` unit increments. If there is either no this key or not set a value, will use default value `1`.

The `TONER_2_GAMMA` key is similar to the `GRAY_GAMMA` except it is limited to the *Toner 2* partition scope only.

## TONER_2_CURVE

This key is used to linearize the Toner 2 partition of a profile. To linearize a full profile the `LINEARIZE` key is used.

```
TONER_2_CURVE=<filename>|"<seq>"
```

**Where**:

- `<filename>` is a path to the adjustment curve file (`.acv` or `.raw`). In this case, quotation marks must not be used.
- `<seq>` is a sequence of curve input-output coordinates pairs. The sequence must be enclosed in double quotation marks.

As a value, there are either `<filename>` or `<seq>` types that can be used.

Pairs are separated by a space and coordinates in the pair are separated by a semicolon (`;`), and have `<real>` type. The first pair must be `0;0`, and the last pair must be `100;100`.

When the `TONER_2_CURVE` key is used the measurements can be carried out by using a scanner or densitometer. In the case of using the `LINEARIZE` key, a spectrophotometer is recommended.

# Other keys

**Represented keys:** UC_NEUTRALIZER; UC_NEUTRALIZER2;

## UC_NEUTRALIZER

The Epson Ultrachrome Inks for the 2200, 7600, and 9600 printer models have a light-black ink which is quite warm. This key allows a precise amount of light-cyan, light-magenta and yellow in exact proportion to the light-black to be introduced and not have any yellow ink used.

```
UC_NEUTRALIZER=<bool>
```

**Where**:

- <bool> is the Boolean value YES or NO telling the *QuadToneRIP* to directly neutralize the color of the light-black.

This is done internally by taking the LK curve and copying it to the LC, LM, and Y curves. Then using the LIMIT_LK, LIMIT_LC, LIMIT_LM, and LIMIT_Y keys you can balance the color for an exactly neutral tone.

This does take some trial and error because the relatively large amount of LC and LM introduced increases the density quite a bit.

To compensate LIMIT_LK must be reduced – the idea is to get an LK, LC, LM, and Y combination to produce the same density as a full LK amount.

UC_NEUTRALIZER should only be used on 7 or 8 ink printers.

## UC_NEUTRALIZER2

This key is identical to the UC_NEUTRALIZER key except for the color of the black is neutralized for getting an exactly neutral tone.

```
UC_NEUTRALIZER2=<bool>
```

**Where**:

- <bool> is the Boolean value YES or NO telling the *QuadToneRIP* to directly neutralize the color of the black.

This is done internally by taking the K curve and copying it to the C and LM curves.

UC_NEUTRALIZER should also only be used on 7 or 8 ink printers.

# Appendices

## A Appendix: Supported printers

There is the full list of all printers supported by the *QuadToneRIP* below. See quadtonerip.com for more details.

| Codename | Printer model | Num. | Ink codes |
|---|---|---|---|
| Quad860 | EPSON Stylus Color 860 | 4 | K,C,M,Y |
| Quad870 | EPSON Stylus Photo 870 | 6 | K,C,M,Y,LC,LM |
| Quad890 | EPSON Stylus Photo 890 | 6 | K,C,M,Y,LC,LM |
| Quad980 | EPSON Stylus Color 980 | 4 | K,C,M,Y |
| Quad1200 | EPSON Stylus Photo 1200 | 6 | K,C,M,Y,LC,LM |
| Quad1270 | EPSON Stylus Photo 1270 | 6 | K,C,M,Y,LC,LM |
| Quad1280 | EPSON Stylus Photo 1280 | 6 | K,C,M,Y,LC,LM |
| Quad1290 | EPSON Stylus Photo 1290 | 6 | K,C,M,Y,LC,LM |
| Quad1400 | EPSON Stylus Photo 1400 | 6 | K,C,M,Y,LC,LM |
| Quad1430 | EPSON Stylus Photo 1430 | 6 | K,C,M,Y,LC,LM |
| Quad2000 | EPSON Stylus Photo 2000 | 6 | K,C,M,Y,LC,LM |
| Quad2100 | EPSON Stylus Photo 2100 | 7 | K,C,M,Y,LC,LM,LK |
| Quad2200 | EPSON Stylus Photo 2200 | 7 | K,C,M,Y,LC,LM,LK |
| Quad3000 | EPSON Stylus Color 3000 | 4 | K,C,M,Y |
| Quad3800 | EPSON Stylus Pro 3800 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad3880 | EPSON Stylus Pro 3880 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad4000 | EPSON Stylus Pro 4000 | 7 | K,C,M,Y,LC,LM,LK |
| Quad4800 | EPSON Stylus Pro 4800 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad4880 | EPSON Stylus Pro 4880 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad4900 | EPSON Stylus Pro 4900 | 10 | K,C,M,Y,LC,LM,LK,LLK,OR,GR |
| Quad7000 | EPSON Stylus Pro 7000 | 6 | K,C,M,Y,LC,LM |
| Quad7500 | EPSON Stylus Pro 7500 | 6 | K,C,M,Y,LC,LM |
| Quad7600 | EPSON Stylus Pro 7600 | 7 | K,C,M,Y,LC,LM,LK |

**Table 3a** The list of all printers supported the *QuadToneRIP*

| Codename | Printer model | Num. | Ink codes |
|---|---|---|---|
| Quad7800 | EPSON Stylus Pro 7800 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad7880 | EPSON Stylus Pro 7880 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad7890 | EPSON Stylus Pro 7890 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad7900 | EPSON Stylus Pro 7900 | 10 | K,C,M,Y,LC,LM,LK,LLK,OR,GR |
| Quad9000 | EPSON Stylus Pro 9000 | 6 | K,C,M,Y,LC,LM |
| Quad9500 | EPSON Stylus Pro 9500 | 6 | K,C,M,Y,LC,LM |
| Quad9600 | EPSON Stylus Pro 9600 | 7 | K,C,M,Y,LC,LM,LK |
| Quad9800 | EPSON Stylus Pro 9800 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad9880 | EPSON Stylus Pro 9880 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad9890 | EPSON Stylus Pro 9890 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad9900 | EPSON Stylus Pro 9900 | 10 | K,C,M,Y,LC,LM,LK,LLK,OR,GR |
| Quad10000 | EPSON Stylus Pro 10000 | 6 | K,C,M,Y,LC,LM |
| Quad11880 | EPSON Stylus Pro 11880 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| Quad15000 | EPSON Stylus Photo 15000 | 6 | K,C,M,Y,LK,R |
| QuadEX | EPSON Stylus Photo EX | 6 | K,C,M,Y,LC,LM |
| QuadP400 | EPSON Stylus Photo P400 | 8 | MK,C,M,Y,R,OR,PK,GL |
| QuadP600 | EPSON Stylus SureColor P600 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadP700 | EPSON Stylus SureColor P700 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadP800 | EPSON Stylus SureColor P800 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadP900 | EPSON Stylus SureColor P900 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadP5000 | EPSON Stylus SureColor P5000 | 10 | K,C,M,Y,LC,LM,LK,LLK,OR,GR |
| QuadP6000 | EPSON Stylus SureColor P6000 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadP7000 | EPSON Stylus SureColor P7000 | 10 | K,C,M,Y,LC,LM,LK,LLK,OR,GR |
| QuadP8000 | EPSON Stylus SureColor P8000 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadP9000 | EPSON Stylus SureColor P9000 | 10 | K,C,M,Y,LC,LM,LK,LLK,OR,GR |
| QuadR200 | EPSON Stylus Photo R200 | 6 | K,C,M,Y,LC,LM |
| QuadR220 | EPSON Stylus Photo R220 | 6 | K,C,M,Y,LC,LM |
| QuadR260 | EPSON Stylus Photo R260 | 6 | K,C,M,Y,LC,LM |
| QuadR280 | EPSON Stylus Photo R280 | 6 | K,C,M,Y,LC,LM |
| QuadR300 | EPSON Stylus Photo R300 | 6 | K,C,M,Y,LC,LM |
| QuadR320 | EPSON Stylus Photo R320 | 6 | K,C,M,Y,LC,LM |
| QuadR340 | EPSON Stylus Photo R340 | 6 | K,C,M,Y,LC,LM |

**Table 3b** The list of all printers supported the *QuadToneRIP*

| Codename | Printer model | Num. | Ink codes |
|---|---|---|---|
| QuadR380 | EPSON Stylus Photo R380 | 6 | K,C,M,Y,LC,LM |
| QuadR800 | EPSON Stylus Photo R800 | 8 | MK,C,M,Y,R,B,PK,GL |
| QuadR1800 | EPSON Stylus Photo R1800 | 8 | MK,C,M,Y,R,B,PK,GL |
| QuadR1900 | EPSON Stylus Photo R1900 | 8 | MK,C,M,Y,R,OR,PK,GL |
| QuadR2000 | EPSON Stylus Photo R2000 | 8 | MK,C,M,Y,R,OR,PK,GL |
| QuadR2400 | EPSON Stylus Photo R2400 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadR2880 | EPSON Stylus Photo R2880 | 8 | K,C,M,Y,LC,LM,LK,LLK |
| QuadR3000 | EPSON Stylus Photo R3000 | 8 | K,C,M,Y,LC,LM,LK,LLK |

**Table 3c** The list of all printers supported the *QuadToneRIP*

# Index of keys