

# ArgyllCMS Tutorial

Fogra Colour Management Symposium 2010

Graeme Gill [graeme@argyllcms.com](mailto:graeme@argyllcms.com)

What is ArgyllCMS ?

Installation Microsoft Windows, Apple OS X, Linux

Introduction to using the shell

Introduction to using Argyll tools

Printer Profiling - Test chart creation

Printer Profiling - Test chart printing

Printer Profiling - Test Chart Reading

Printer Profiling - Creating Profiles

Printer Profiling - Designing a black Curve

colprof options

Device Link Creation

Linking intents

CIECAM02 Viewing conditions

Transforming raster files - cctiff

Fluorescent Whitener Additive Compensation

Printer Calibration

Gamut visualization and comparison

[Quick overview of other tools and topics](#)

Diagnostic Tools

Display calibration & Profiling

Camera & Scanner profiling

Soft-proofing link

Tailoring test charts

Refining proofing profiles

# What is ArgyllCMS ?

ArgyllCMS is an ICC compatible colour management system, available as Open Source under the GNU Copyleft license. It can be used freely, but any re-distribution must comply with the GNU license conditions.

It is cross platform, running on Microsoft Windows, Apple OS X and Linux.

It is a collection of command line tools that can be used for a wide variety of colour management purposes.

A range of colour measuring instruments are directly supported including DTP20, DTP41, ColorMunki, SpectroScan, Eye One, DTP94, Spyder.

Archives of executables, documentation and sample files available for download from <<http://www.argyllcms.com/>>

On line documentation is at  
<<http://www.argyllcms.com/doc/ArgyllDoc.html>>

# Installation - 1

Options are Source or Binary installation:

Source:

- Download source archive

- Setup compile environment and build tool (Jam)

- Compile

Details <<http://www.argyllcms.com/doc/Compiling.html>>

Binary:

- Download binary archive

- De-archive it in suitable location

- Setup command line environment to make use of it.

- Deal with any instrument access issues.

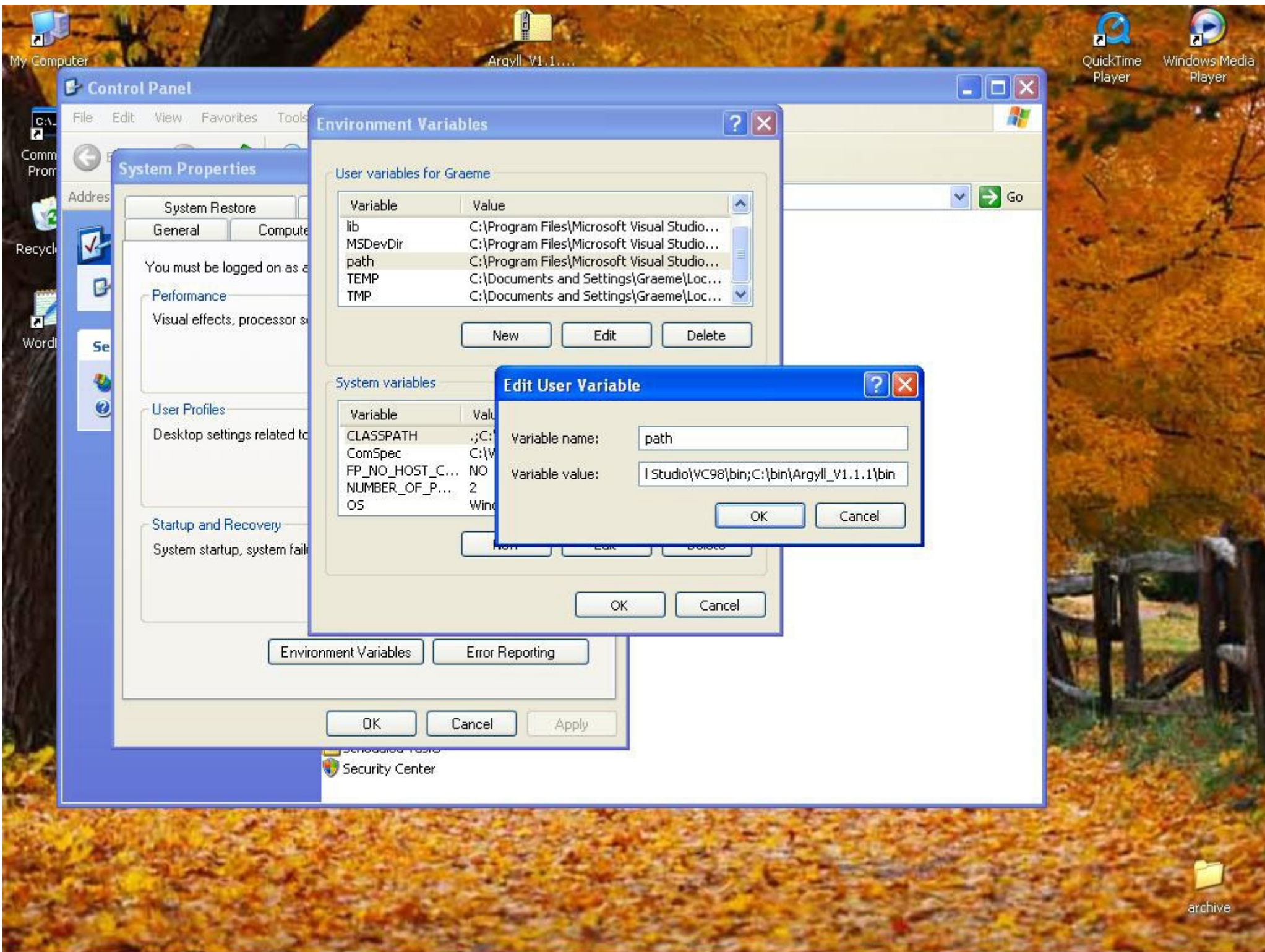
# Installation – 2, Microsoft Windows

[We assume that the main drive is C:]

1. Download binary archive [Argyll\\_V1.1.1\\_win32\\_exe.zip](http://www.argyllcms.com/Argyll_V1.1.1_win32_exe.zip) and save it somewhere (e.g. to the desktop) from <http://www.argyllcms.com/> after navigating to the “[Download V1.1.1 Main Windows 32 executables](#)” page.
2. De-archive it to a suitable directory, e.g. by right click “extract all”, follow the wizard and set the directory to “[C:\](#)”, so that it ends up in [C:\Argyll\\_V1.1.1](#)
3. Add the Argyll executables to your %PATH% - i.e. for XP:

Start->Settings->Control Panel->System  
->Advanced->Environment Variables  
->user variables->path

then edit the path to add the Argyll executables directory:  
..existing paths.. [;C:\Argyll\\_V1.1.1\bin](#)

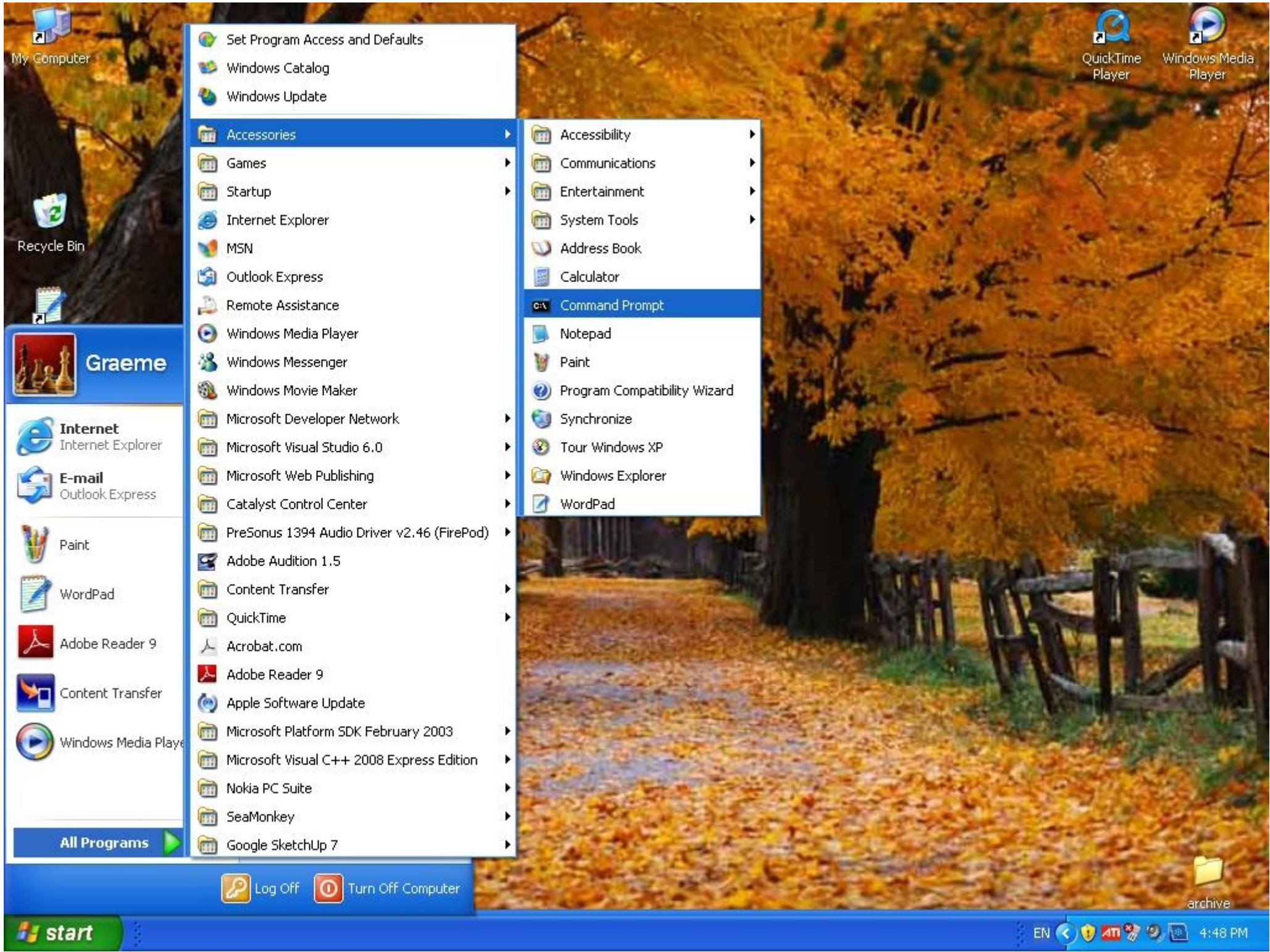


# Installation – 3, Microsoft Windows

4. Open a command prompt window, i.e.  
start->All Programs->Accessories->Command Prompt

(Dragging Command Prompt to the desktop or start menu is a good idea to make it more quickly accessible).





**Graeme**

**Internet**  
Internet Explorer

**E-mail**  
Outlook Express

Paint

WordPad

Adobe Reader 9

Content Transfer

Windows Media Player

**All Programs**

- Set Program Access and Defaults
- Windows Catalog
- Windows Update
- Accessories**
- Games
- Startup
- Internet Explorer
- MSN
- Outlook Express
- Remote Assistance
- Windows Media Player
- Windows Messenger
- Windows Movie Maker
- Microsoft Developer Network
- Microsoft Visual Studio 6.0
- Microsoft Web Publishing
- Catalyst Control Center
- PreSonus 1394 Audio Driver v2.46 (FirePod)
- Adobe Audition 1.5
- Content Transfer
- QuickTime
- Acrobat.com
- Adobe Reader 9
- Apple Software Update
- Microsoft Platform SDK February 2003
- Microsoft Visual C++ 2008 Express Edition
- Nokia PC Suite
- SeaMonkey
- Google SketchUp 7

- Accessibility
- Communications
- Entertainment
- System Tools
- Address Book
- Calculator
- Command Prompt**
- Notepad
- Paint
- Program Compatibility Wizard
- Synchronize
- Tour Windows XP
- Windows Explorer
- WordPad



# Installation – 4, Microsoft Windows

5. Check that you can access Argyll tools:

```
C:\>echo %PATH%
```

```
C:\WINDOWS\system32;C:\WINDOWS;c:\Argyll_V1.1.1\bin
```

```
C:\>iccdump -?
```

```
Dump an ICC file in human readable form, V2.12
```

```
Author: Graeme W. Gill
```

```
usage: iccdump [-v level] [-t tagname] [-s] infile
```

```
-v level          Verbose level 1-3 (default 2)
```

```
-t tag           Dump this tag only (can be used m..
```

```
-s              Search for embedded profile
```

```
-i              Check V4 ID value
```

```
C:\>
```



# Installation – 5, Microsoft Windows

## 6. Install instrument USB drivers:

- a) If you currently don't have any applications that talk to your instrument using USB, then it is relatively simple to use the drivers provided with Argyll. On plugging the instrument in, MSWindows should pop up a "New Hardware" dialog, and ask you for drivers for the particular USB instrument. Using the dialog navigate to the [C:\Argyll\\_V1.1.1\libusbw](#) directory, and select the appropriate **.inf** file.
- b) If you currently have applications other than Argyll accessing your USB connected instrument, then you won't automatically be prompted to install the drivers needed to access it. Instead you have to manually install the drivers, and then switch back and forth between the Argyll and other drivers if you want to switch between applications, using **device manager**.

Details <[http://www.argyllcms.com/doc/Installing\\_MSWindows.html](http://www.argyllcms.com/doc/Installing_MSWindows.html)>

# Installation – 6, Apple OS X

1. Download binary archive [Argyll\\_V1.1.0\\_osx10.4\\_i86\\_bin.tgz](http://www.argyllcms.com/) and save it somewhere (e.g. to the desktop) from <http://www.argyllcms.com/> after navigating to the “Download V1.1.1 Main OS X executables” page.
2. De-archive it, e.g. by control-click “Open With” [BOMArchiveHelper](#) or [Archive Utility](#). Drag the resulting folder to where you want it, e.g. into your home folder ([/Users/username](#) where username is your username).
3. Open a Terminal shell. This will be in [Applications->Utilities->Terminal](#) (Dragging it to the dock is a good idea to make it more accessible).
4. Add the Argyll executables to your \$PATH, by editing your [.profile](#):

```
~$open .profile
```

And add the following line:

```
PATH=$PATH:$HOME/Argyll_V1.1.1/bin
```



# Installation – 7, Apple OS X

5. Close the terminal, and open another one using **File->New Shell**

6. Check that you can access Argyll tools:

```
~$echo $PATH
```

```
/bin:/sbin:/usr/bin:/usr/sbin:/Users/username/Argyll_V1.1.1/bin
```

```
~$iccdump -?
```

```
Dump an ICC file in human readable form, V2.12
```

```
Author: Graeme W. Gill
```

```
usage: iccdump [-v level] [-t tagname] [-s] infile
```

```
-v level          Verbose level 1-3 (default 2)
```

```
-t tag            Dump this tag only (can be used m..
```

```
-s               Search for embedded profile
```

```
-i               Check V4 ID value
```

```
~$
```



# Installation – 8, Linux

Similar to OS X, but differing in details that depend on the Linux distribution.

1. Download binary archive from <<http://www.argyllcms.com/>> after navigating to the “Download V1.1.1 Main Linux executables” page.
2. De-archive it using “`tar -xvzf archive`” to an appropriate directory (e.g. `~/Argyll_V1.1.1`).
3. Add the Argyll executables to your `$PATH`, by editing your shell profile.
4. See if usb permissions need to be changed by installing the `/etc/udev/rules.d/55-Argyll.rules` file and possibly creating and adding yourself to the plugdev group.
5. Attend to any other Linux specific issues.

Details <[http://www.argyllcms.com/doc/Installing\\_Linux.html](http://www.argyllcms.com/doc/Installing_Linux.html)>

# Introduction to using the shell - 1

	<u>MSWindows</u>	<u>OS X/Linux</u>
Print current directory	cd	pwd
List files in a directory	dir /W	ls
List files details in directory	dir	ls -l
Check a particular file exists	dir filename	ls -l filename
Change directories	cd dirname	cd dirname
Change dir. and save current	pushd dirname	pushd dirname
Return to previous dir.	popd	popd
Parent directory	..	..
Current directory	.	.
Home directory	%HOME%	~
Root directory, path separator	\	/
Create a new directory	mkdir dirname	mkdir dirname
Delete an empty directory	rmdir dirname	rmdir dirname
Delete director and contents	rmdir /S dirname	rm -r dirname
Delete a file	del filename	rm filename
Copy a file	copy src dest	cp src dest
Rename a file, directory	ren old new	mv old new
Type a file to terminal	type filename	cat filename
Paths to files. e.g.	..\Argyll_V1.1.0\ref\file	../Argyll_V1.1.1/ref/file

# Introduction to using the shell - 2

Note that on MSWindows filenames and directories are case insensitive, while on OS X and Linux they are case sensitive.

There are lots of tutorials to help you – Google is your friend!

e.g. For MSWindows Google: “windows command prompt tutorial”,  
For OS X : “OS X shell tutorial”,  
For Linux: “Linux shell tutorial”,  
For OS X and Linux: “Unix shell tutorial”

# Introduction to Argyll tools

Invoke tools by their name, followed by flags, options and finally arguments such as filenames.

To get a brief listing of the possible arguments and usage of any of the tools, run it with just an "-?" argument, i.e. `targen -?`

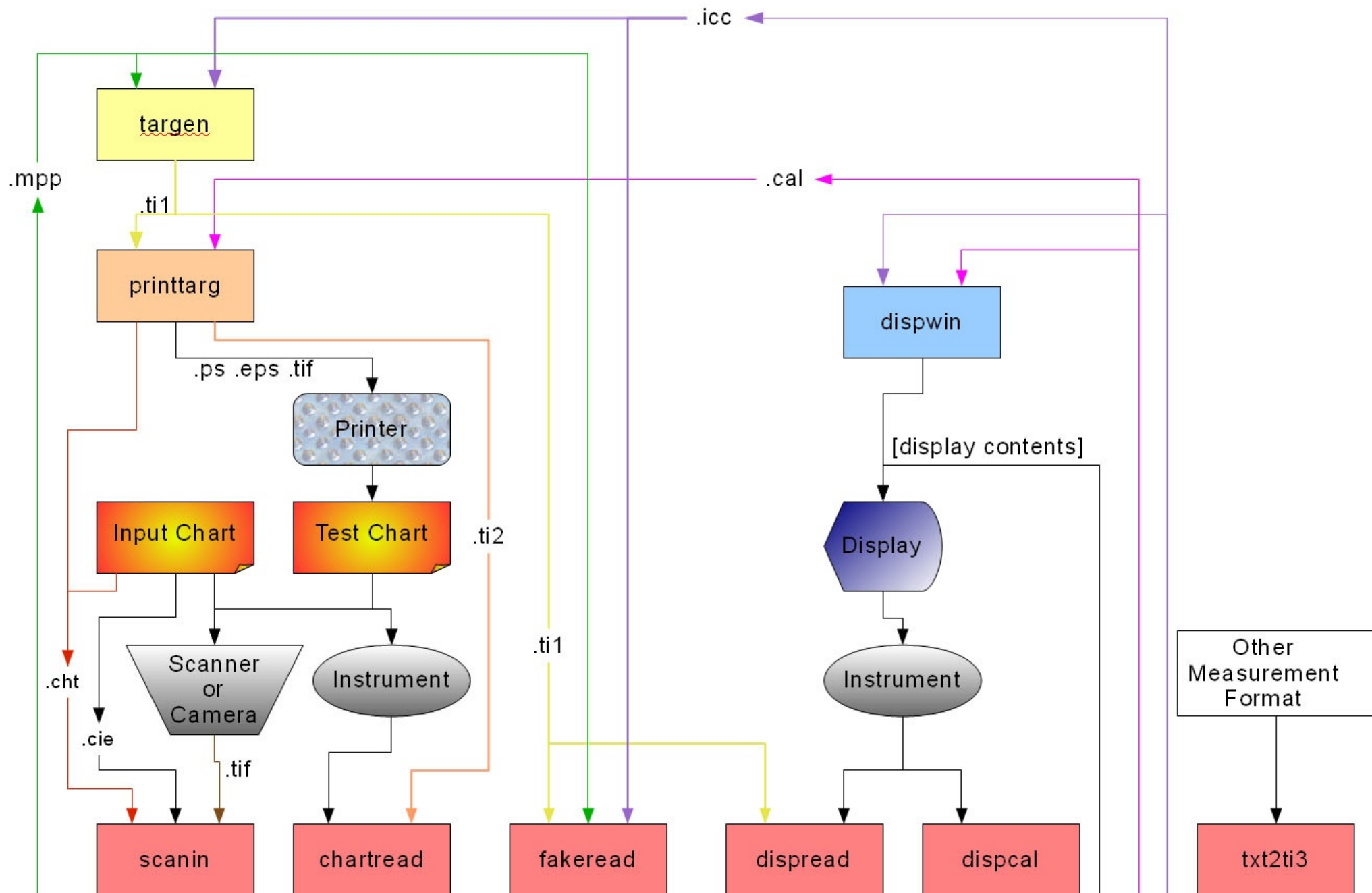
`tool -?`

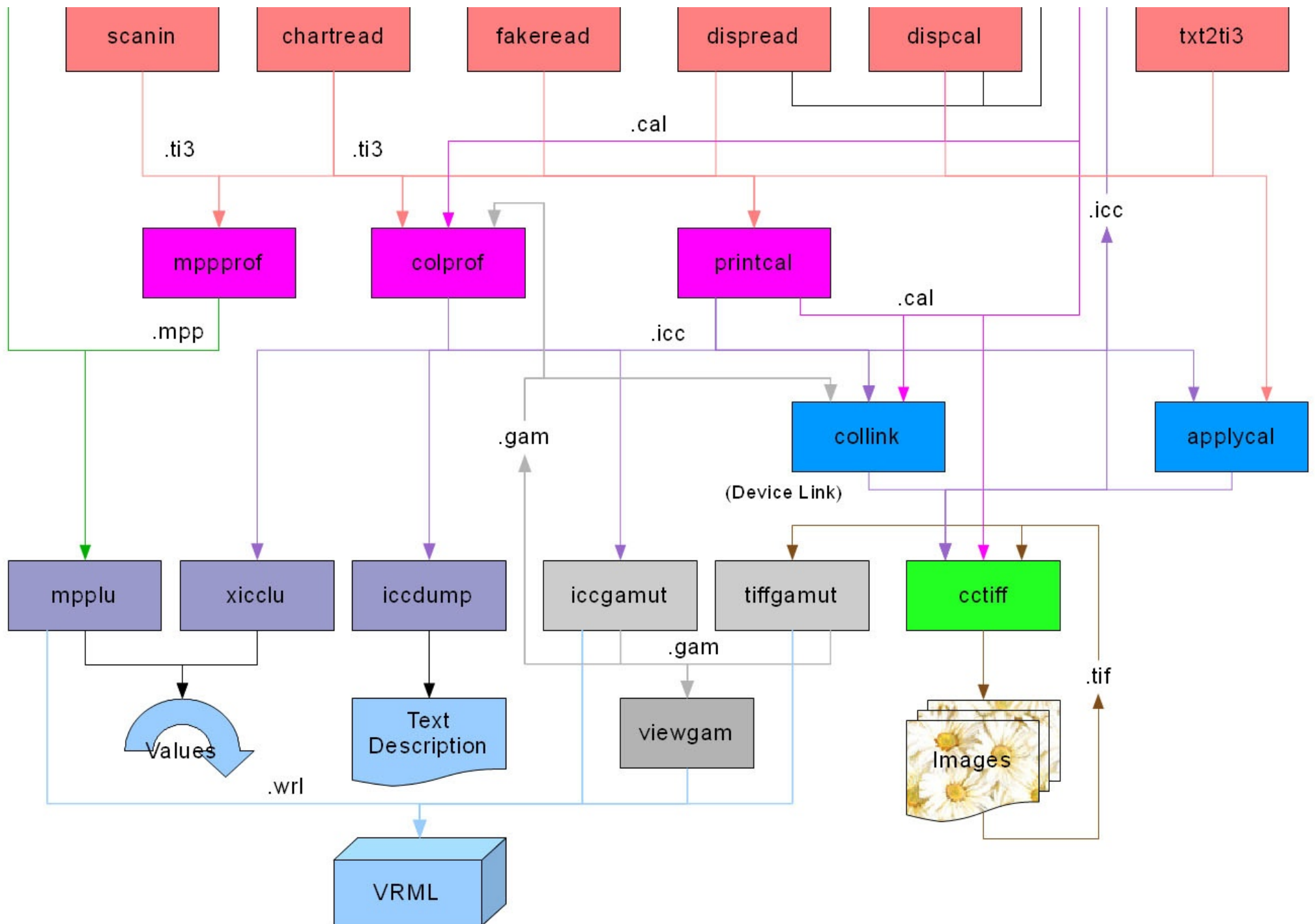
usage: `tool [options] infile outfile`

<code>-v</code>	Verbose mode
<code>-d n</code>	Choose a depth 0-4
<code>-r</code>	Use a random depth
<code>-f [nn]</code>	Use full range. nn optional range 0 - 100.
<code>-M</code>	Manual
<code>infile</code>	Input file
<code>outfile</code>	Output file

All arguments need to be separated by whitespace, so a string with white space will need double quoting. Creating directories or filenames with spaces in them will make things difficult, so don't do it !







# Printer Profiling – 1. Test chart creation

Need to choose:

- Colourspace (Total ink limit for CMYK)

- Number of test patches or paper size & number of sheets

- Type of test value distribution and any special values

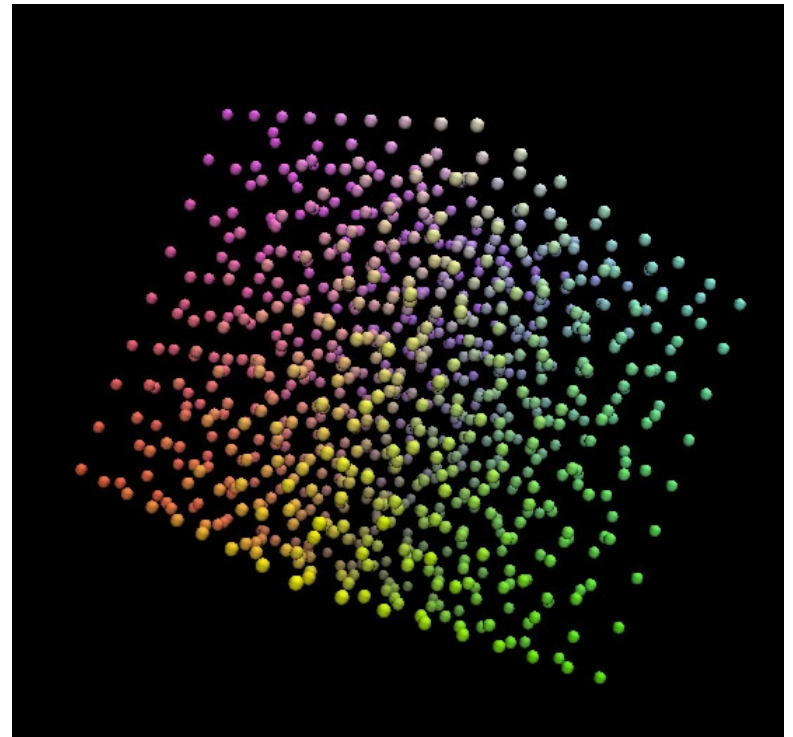
- Whether to use a previous or similar profile for “pre-conditioning”

then use the **targen** tool to create the **.ti1** file containing the values:

e.g. For an “RGB” printer using an i1 Pro,  
2 x A4 sheets holds 882 patches,  
default “farthest point” distribution  
+ default 4 white patches:

```
targen -v -d2 -f882 PrinterA
```

Because with didn't supply a profile, it  
defaults to assuming an sRGB like  
colourspace, but adaptation is low  
(0.1).



iRGB test chart

Full spread patches = 882

Adaptation weights: Device = 0.935, Perceptual = 0.065, Curvature = 0.010

Perceptual cache resolution = 11

Acceleration grid res = 10

There are 1 unique fixed points to add (4 non-unique)

There are 878 far spread points to add

Added 879/879

After seeding points: MinPoint = 9.124, Min = 4.970, Avg. = 8.714, Max = 11.039, 1.1 secs

Re-seeding

It 1: Maxmv = 0.050462, MinPoint = 5.731, Min = 4.673, Avg. = 8.751, Max = 10.897, 1.1 secs.

Re-seeding

It 2: Maxmv = 0.046132, MinPoint = 6.649, Min = 5.530, Avg. = 8.749, Max = 10.857, 1.1 secs.

Re-seeding

It 3: Maxmv = 0.037650, MinPoint = 6.552, Min = 5.542, Avg. = 8.692, Max = 10.934, 1.2 secs.

Fixing up veronoi

It 4: Maxmv = 0.021614, MinPoint = 8.112, Min = 5.580, Avg. = 8.647, Max = 10.323, 0.6 secs.

Fixing up veronoi

It 5: Maxmv = 0.010272, MinPoint = 8.374, Min = 5.580, Avg. = 8.627, Max = 10.468, 0.5 secs.

Fixing up veronoi

It 6: Maxmv = 0.000793, MinPoint = 8.412, Min = 5.578, Avg. = 8.626, Max = 10.450, 0.5 secs.

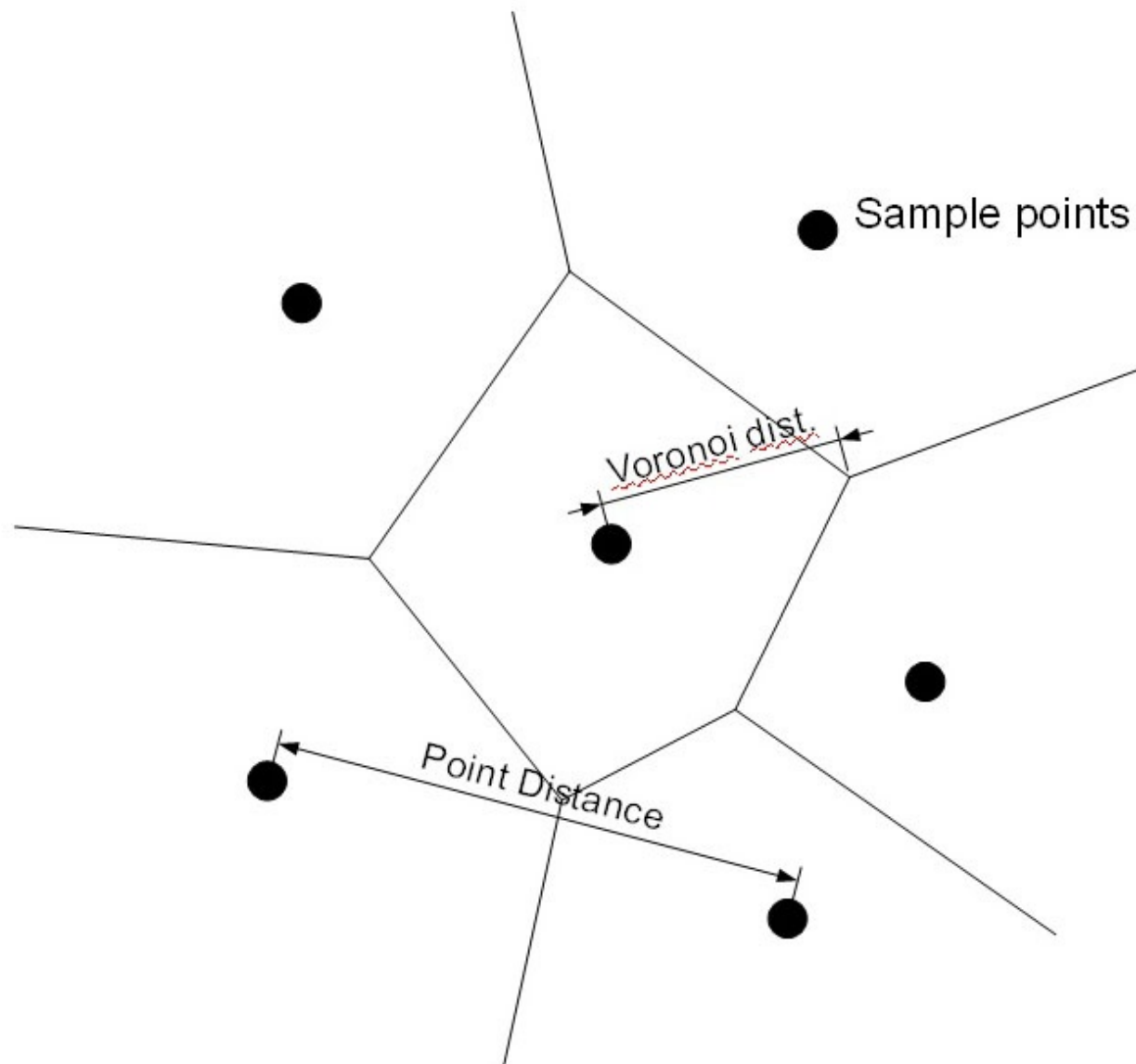
After optimization: MinPoint = 8.412, Min = 5.578, Avg. = 8.626, Max = 10.450

Total number of patches = 882

Execution time = 6.140000 seconds



Maxmv = 0.010272, MinPoint = 8.374, Min = 5.580, Avg. = 8.627, Max = 10.468



CMYK would be similar:

4 x A4 sheets holds 1764 patches, default “farthest point” distribution  
+ default 4 white patches:

```
targen -v -d4 -f1764 -l280 -c PrevPrinterB.icc PrinterB
```

Because we supplied a previous profile, adaptation will default to high (1.0)

CMYK test chart

Full spread patches = 1764

Ink limit = 280.0% (underlying 280.0%)

Adaptation weights: Device = 0.350, Perceptual = 0.650, Curvature = 1.000

Perceptual cache resolution = 7

Acceleration grid res = 7

There are 1 unique fixed points to add (4 non-unique)

There are 1760 far spread points to add

Added 1761/1761

After seeding points: MinPoint = 11.151, Min = 3.815, Avg. = 11.816, Max = 14.622, 19.0 secs

Re-seeding

It 1: Maxmv = 0.094141, MinPoint = 6.502, Min = 5.815, Avg. = 11.921, Max = 15.184, 21.8 secs.

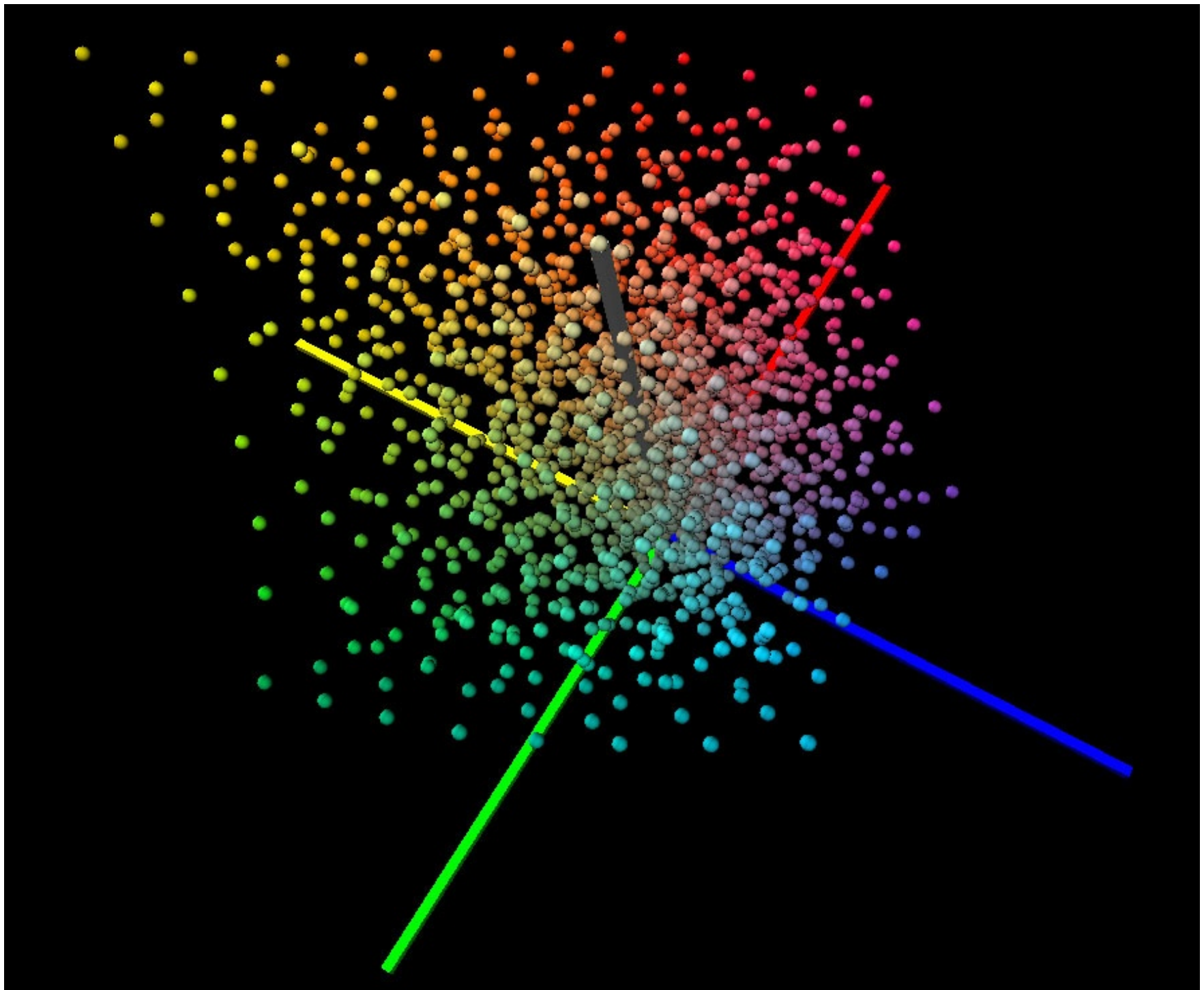
.....

It 6: Maxmv = 0.003157, MinPoint = 8.964, Min = 5.815, Avg. = 11.775, Max = 14.090, 22.6 secs.

After optimization: MinPoint = 8.964, Min = 5.815, Avg. = 11.775, Max = 14.090

Total number of patches = 1764

Execution time = 153.578000 seconds



usage: targa [options] outfile

-v [level]      Verbose mode [optional level 1..N]  
-d col\_comb     choose colorant combination from the following:  
                2: Print RGB  
                3: Video RGB  
                4: CMYK  
                5: CMY  
  
-G              Generate good optimized points rather than Fast  
-e patches     White test patches (default 4)  
-s steps       Single channel steps (default grey 50, color 0)  
-g steps       Grey axis RGB or CMY steps (default 0)  
-m steps       Multidimensional device space cube steps (default 0)  
-f patches     Add iterative & adaptive full spread patches to total  
                Default is Optimised Farthest Point Sampling (OFPS)  
-t              Use incremental far point for full spread  
-r              Use device space random for full spread  
-R              Use perceptual space random for full spread  
-q              Use device space-filling quasi-random for full spread  
-Q              Use perceptual space-filling quasi-random for full spread  
-i              Use device space body centered cubic grid for full spread  
-I              Use perceptual space body centered cubic grid for full spread  
-A adaptation   Degree of adaptation of OFPS 0.0 - 1.0 (dflt 0.1, 1.0 if -c profile)  
-l ilimit       Total ink limit in % (default = none)  
-c profile     Optional device ICC or MPP pre-conditioning profile filename  
                (Use "none" to turn off any conditioning)  
-w              Dump diagnostic outfile1.wrl file (Lab locations)  
-W              Dump diagnostic outfile1.wrl file (Device locations)  
outfile        Base name for output(.ti1)



# Printer Profiling – 2. Test chart printing

Next step is creating a file to print using the **printtarg** tool.

Need to choose the type of output, type of instrument & paper size.  
We'll use the default PostScript, A4 paper and the i1 Pro instrument:

```
printtarg -v -ii1 -pA4 PrinterA
```

```
printtarg -v -ii1 -pA4 PrinterB
```

As well as the .ps, .eps or .tif file, it will create a **.ti2** file to convey the device, expected CIE values and patch locations.

```
printtarg -v -ii1 -pA4 PrinterA
```

Paper chosen is A4 [210.0 x 297.0 mm]

Patches = 882

Test patches per row = 21

Rows per page = 21,

patches per page = 441

Rows in last strip = 21,

patches in last row = 21

Total pages needed = 2

Worst case delta E = 9.303770

Worst case direction distinction

delta E = 53.631759

Optimising layout for strip reader:  
100%

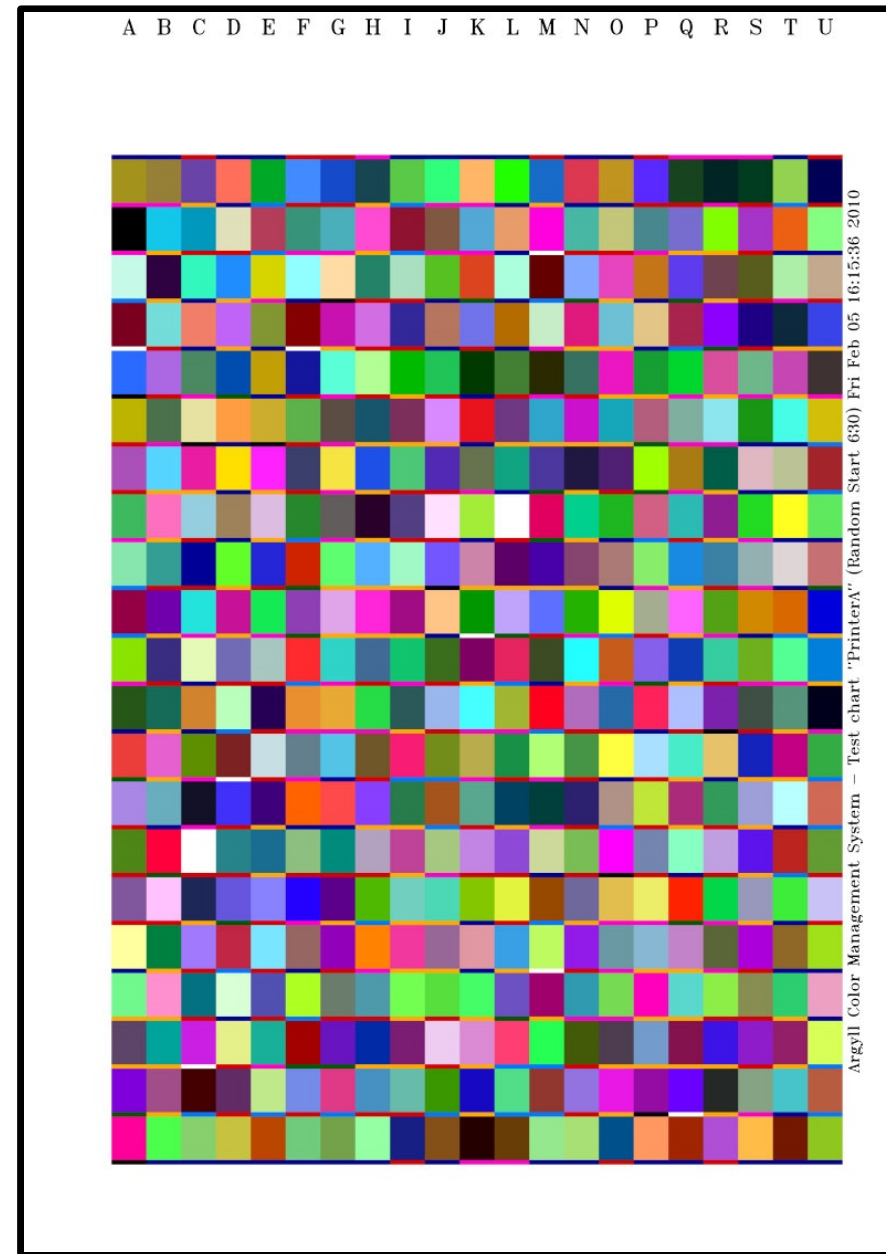
After optimisation, worst

delta E = 90.995117

Worst case direction distinction

delta E = 86.312667

Creating file 'PrinterA.ps'



```

usage: printtarg [-v] [-i instr] [-r] [-s] [-p size] basename
  -v                Verbose mode
  -i 20 | 22 | 41 | 51 | SS | i1 | CM Select target instrument (default DTP41)
                    20 = DTP20, 22 = DTP22, 41 = DTP41, 51 = DTP51,
                    SS = SpectroScan, i1 = i1Pro, CM = ColorMunki
  -h                Use hexagon patches for SS, double density for CM
  -a scale           Scale patch size and spacers by factor (e.g. 0.857 or 1.5 etc.)
  -A scale           Scale spacers by additional factor (e.g. 0.857 or 1.5 etc.)
  -r                Don't randomize patch location
  -s                Create a scan image recognition (.cht) file
  -e                Output EPS compatible file
  -t [res]           Output 8 bit TIFF raster file, optional res DPI (default 100)
  -T [res]           Output 16 bit TIFF raster file, optional res DPI (default 100)
  -Q nbits           Quantize test values to fit in nbits
  -R rsnum           Use given random start number
  -K file.cal        Apply printer calibration to patch values and include in .ti2
  -I file.cal        Include calibration in .ti2 (but don't apply it)
  -x pattern         Use given strip indexing pattern (Default = "A-Z, A-Z")
  -y pattern         Use given patch indexing pattern (Default = "0-9,@-9,@-9;1-999")
  -m margin          Set a page margin in mm (default 6.0 mm)
  -M margin          Set a page margin in mm and include it in TIFF
  -p size            Select page size from:
                    A4          [210.0 x 297.0 mm]
                    A4R        [297.0 x 210.0 mm]
                    A3          [297.0 x 420.0 mm] (default)
                    A2          [420.0 x 594.0 mm]
                    Letter      [215.9 x 279.4 mm]
                    LetterR     [279.4 x 215.9 mm]
                    .....
  -p WWWxHHH        Custom size, WWW mm wide by HHH mm high
  basename           Base name for input(.ti1), output(.ti2) and output(.ps/.eps/.tif)

```

# Printer Profiling – 3. Test Chart Reading

Next step is reading the chart using the **chartread** tool.

```
usage: chartread [-options] outfile
-v              Verbose mode
-c listno       Set communication port from the following list (default 1)
    1 = 'usb:/bus0/dev1 (GretagMacbeth i1 Pro)'
    2 = 'COM1'
    3 = 'COM2'
-t              Use transmission measurement mode
-d              Use display measurement mode (white Y relative results)
-y c|l          Display type (if emissive), c = CRT, l = LCD
-e              Emissive for transparency on a light box
-p              Measure patch by patch rather than strip
-x [lx]         Take external values, either L*a*b* (-xl) or XYZ (-xx).
-n              Don't save spectral information (default saves spectral)
-l              Save CIE as D50 L*a*b* rather than XYZ
-r              Resume reading partly read chart
-I file.cal      Override calibration info from .ti2 in resulting .ti3
-N              Disable auto calibration of instrument
-B              Disable auto bi-directional strip recognition
-H              Use high resolution spectrum mode (if available)
-T ratio        Modify strip patch consistency tolerance by ratio
-W n|h|x        Override serial port flow control: n = none, h = HW, x = Xon/Xoff
-D [level]      Print debug diagnostics to stderr
outfile         Base name for input[ti2]/output[ti3] file
```

## chartread PrinterA

Place the instrument on its reflective white reference Serial no. 125607,  
and then hit any key to continue,  
or hit Esc or Q to abort:  
Calibration complete

Ready to read strip pass A

Press 'f' to move forward, 'b' to move back, 'n' for next unread,  
'd' when done, Esc or 'q' to quit without saving.

Trigger instrument switch or any other key to start:

Strip read OK

Ready to read strip pass B

Press 'f' to move forward, 'b' to move back, 'n' for next unread,  
'd' when done, Esc or 'q' to quit without saving.

Trigger instrument switch or any other key to start:

Strip read OK (Strip read in reverse direction)

Ready to read strip pass DA

Press 'f' to move forward, 'b' to move back, 'n' for next unread,  
'd' when done, Esc or 'q' to quit without saving.

Trigger instrument switch or any other key to start:

Strip read OK

Ready to read strip pass DA (All rows read)

Press 'f' to move forward, 'b' to move back, 'n' for next unread,  
'd' when done, Esc or 'q' to quit without saving.

Trigger instrument switch or any other key to start:

d

# Printer Profiling – 4. Creating Profiles

Last step is creating an ICC profile using the **colprof** tool. Many options are possible, depending on the intended use, e.g.:

“RGB” printer intended for use with conventional CMM. Perceptual and saturation intent sources are sRGB gamut or sRGB “like”. Domestic type viewing conditions (typical monitor, practical print).

```
colprof -v -D"Printer A" -qm -S sRGB.icm -cmt -dpp PrinterA
```

```
No total ink limit being used
```

```
No black ink limit being used
```

```
No of test patches = 882
```

```
Estimating white point
```

```
Approximate White point XYZ = 0.834211 0.866517 0.763857, Lab = 94.590276 -0.244278 -4.266827
```

```
Creating optimised per channel curves
```

```
Initial White Point XYZ 0.834211 0.866517 0.763857, Lab 94.590276 -0.244278 -4.266827
```

```
About to optimise temporary matrix
```

```
100%
```

```
About to optimise input curves and matrix
```

```
100%
```

```
About to optimise output curves and matrix
```

```
100%
```

```
About to optimise input curves and matrix again
```

```
100%
```

About to optimise input, matrix and output together

100%

About to adjust a and b output curves for white point

About to create grid position input curves

Create final clut from scattered data

\*\*\*\*\*

Doing White point fine tune:

Before fine tune, rel WP = XYZ 0.96284 0.99876 0.82524, Lab 99.95236 -0.02857 -0.10957

After fine tune, rel WP = XYZ 0.96420 1.00000 0.82490, Lab 100.00000 0.00000 0.00000

Creating fast inverse input lookups

White point XYZ = 0.833058 0.865455 0.764174, Lab = 94.545112 -0.269302 -4.371644

Find black point

Black point XYZ = 0.004645 0.004969 0.004681, Lab = 4.488657 -0.592339 -1.098184

Done A to B table creation

.

(Gamut mapping information for Perceptual and Saturation)

.

profile check complete, peak err = 6.879761, avg err = 0.974908, RMS = 1.158450

For a CMYK printer where we don't wish to override the ink limits set in the test chart, or set a particular black generation curve, the approach is similar:

```
colprof -v -D"Printer B" -qm -S sRGB.icm -cmt -dpp PrinterB
```

.



# Printer Profiling – 5. Designing a black Curve

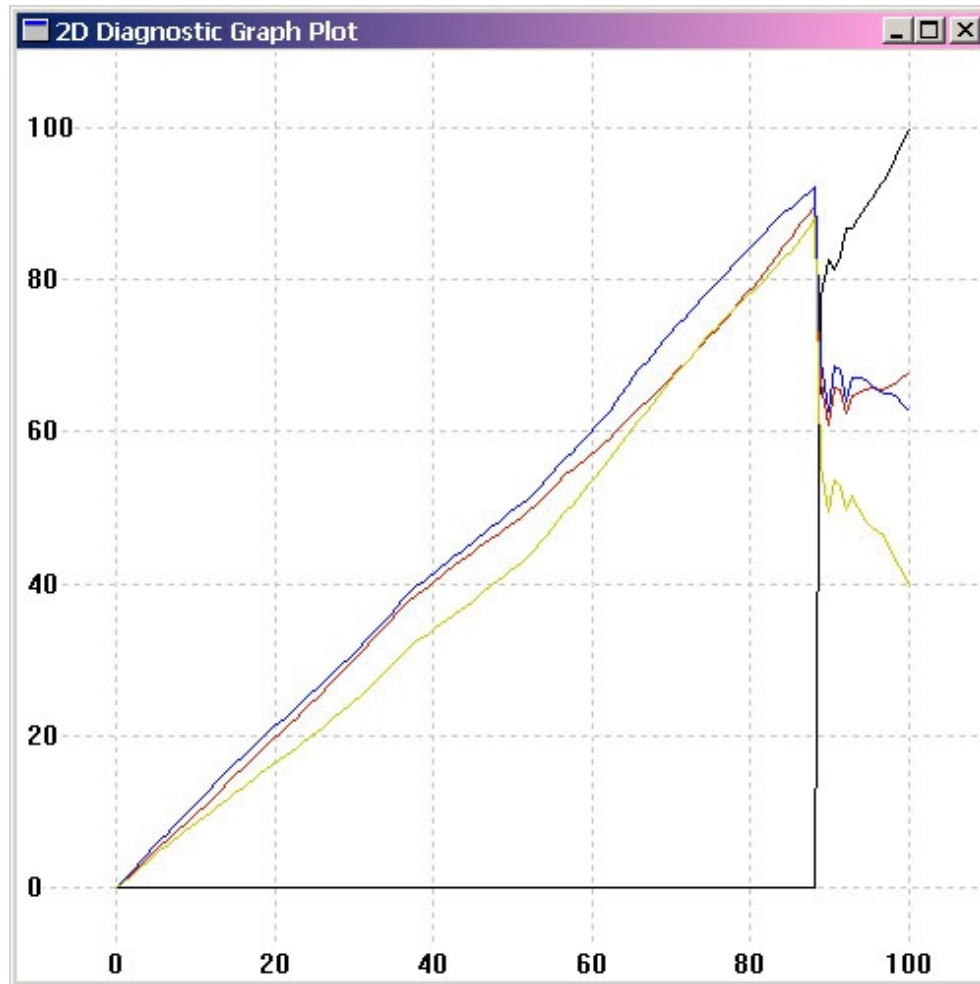
Sometimes we may want to choose a specific black curve for a CMYK profile, and the **xicclu** tool can help us with this. First step is to make a preliminary profile:

```
cp PrinterB.ti3 PrinterBt.ti3  
colprof -v -qm -b -cmt -dpp PrinterBt
```

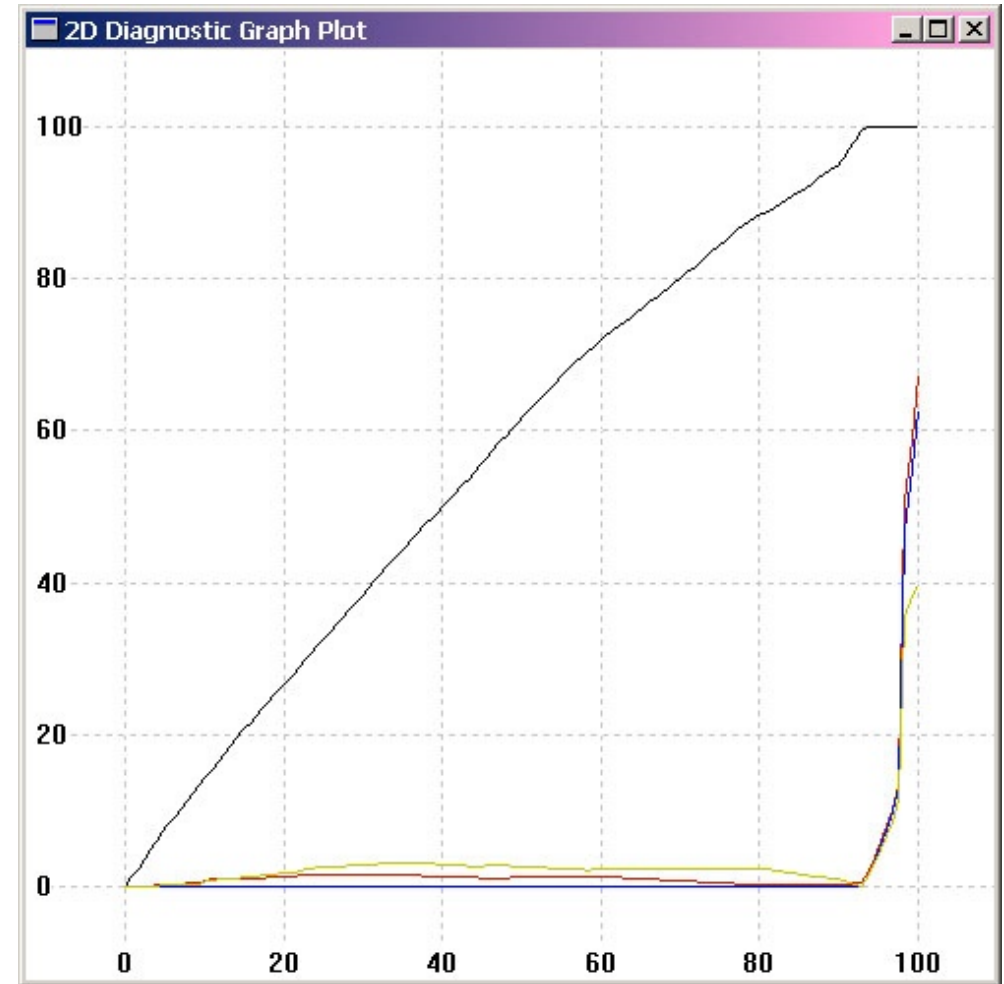
To speed this up we don't create Perceptual and Saturation tables, and in fact make the B2A table small by using **-b**. We can then use **xicclu -g** to see how a particular black generation choice affects the resulting CMYK down the neutral axis.

Because xicclu isn't using the .ti3 file, and ink limits aren't stored in the ICC profile, we have to supply them.

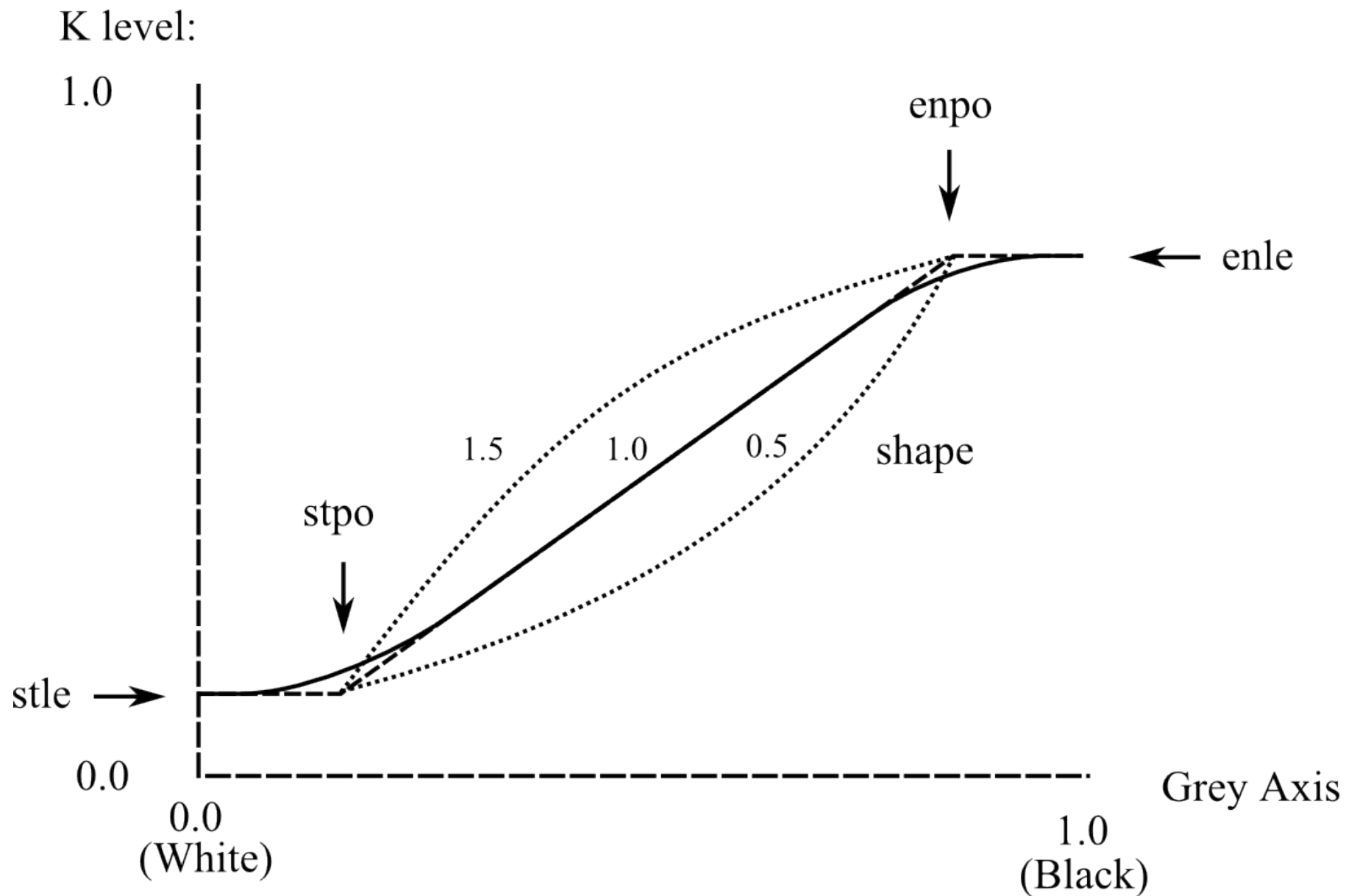
We used a total limit of 280% for the test chart, so a limit of 270% is the default that **colprof** would normally use.



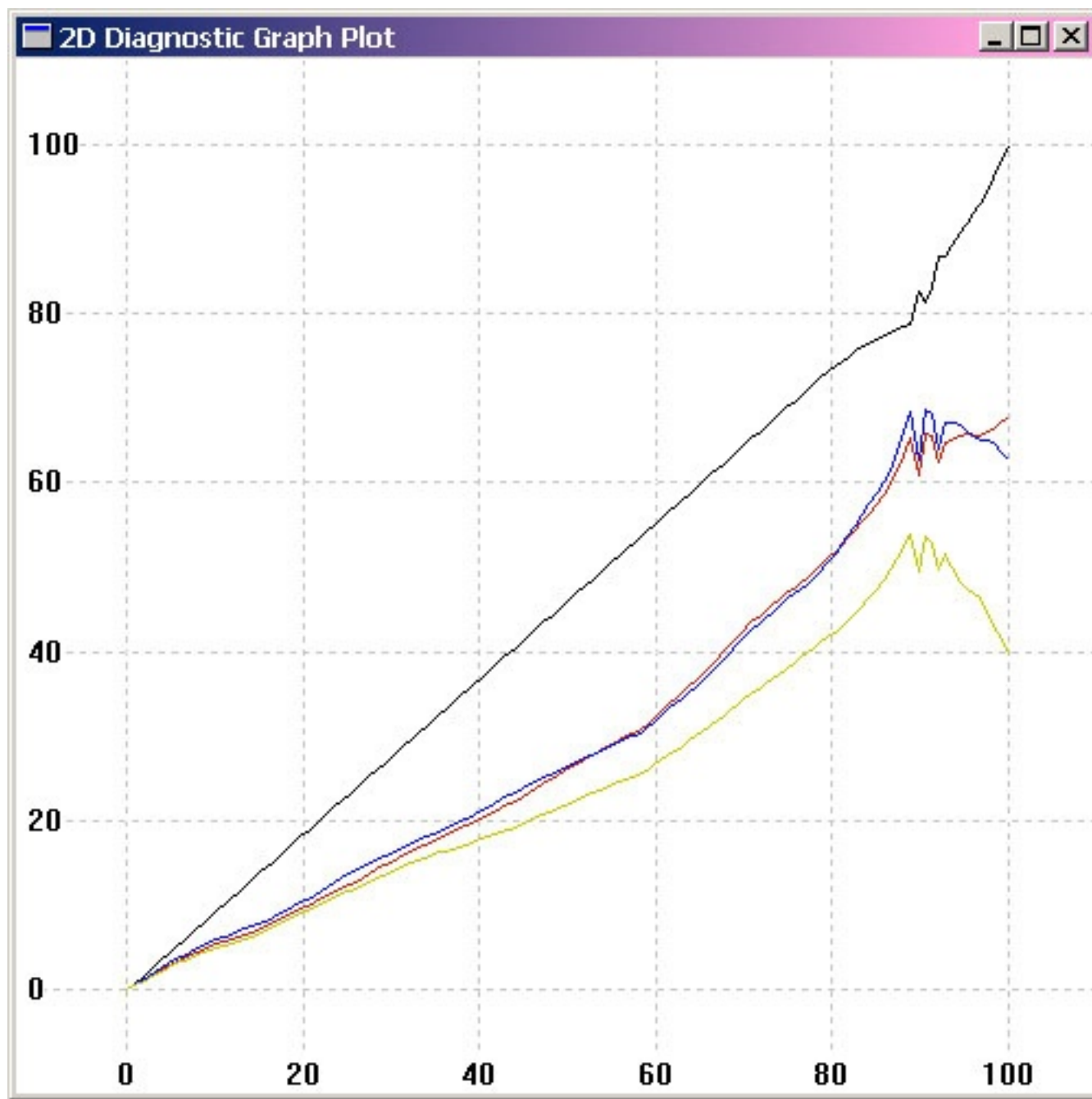
`xicclu -g -kz -l270 -fif -ir PrinterBt.icm`



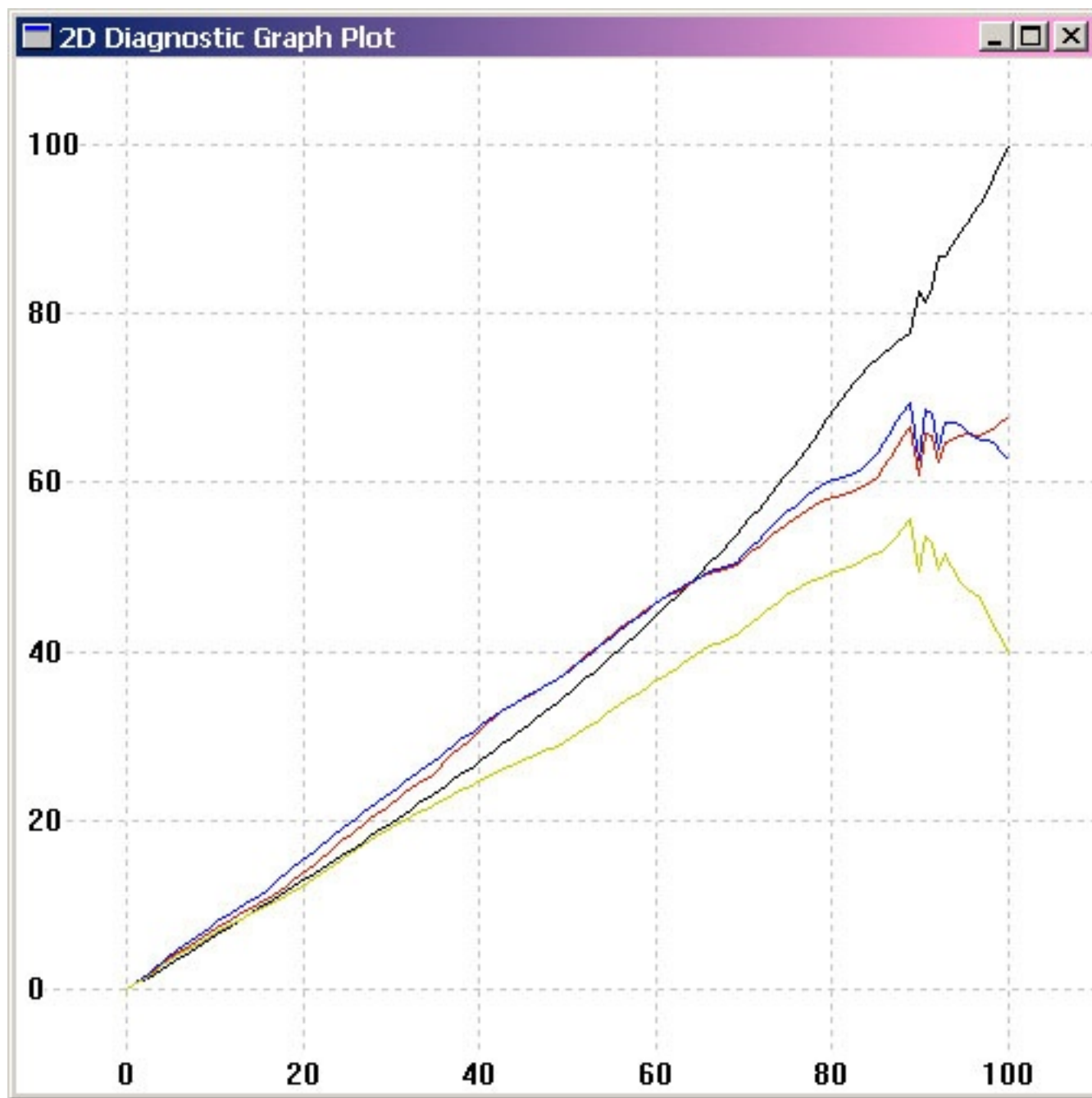
`xicclu -g -kx -l270 -fif -ir PrinterBt.icm`



-k parameters in order: stle, stpo, enpo, enle, shape



```
xicclu -g -kp 0 0 .87 .80 1.0 -1270 -fif -ir PrinterBt.icm
```



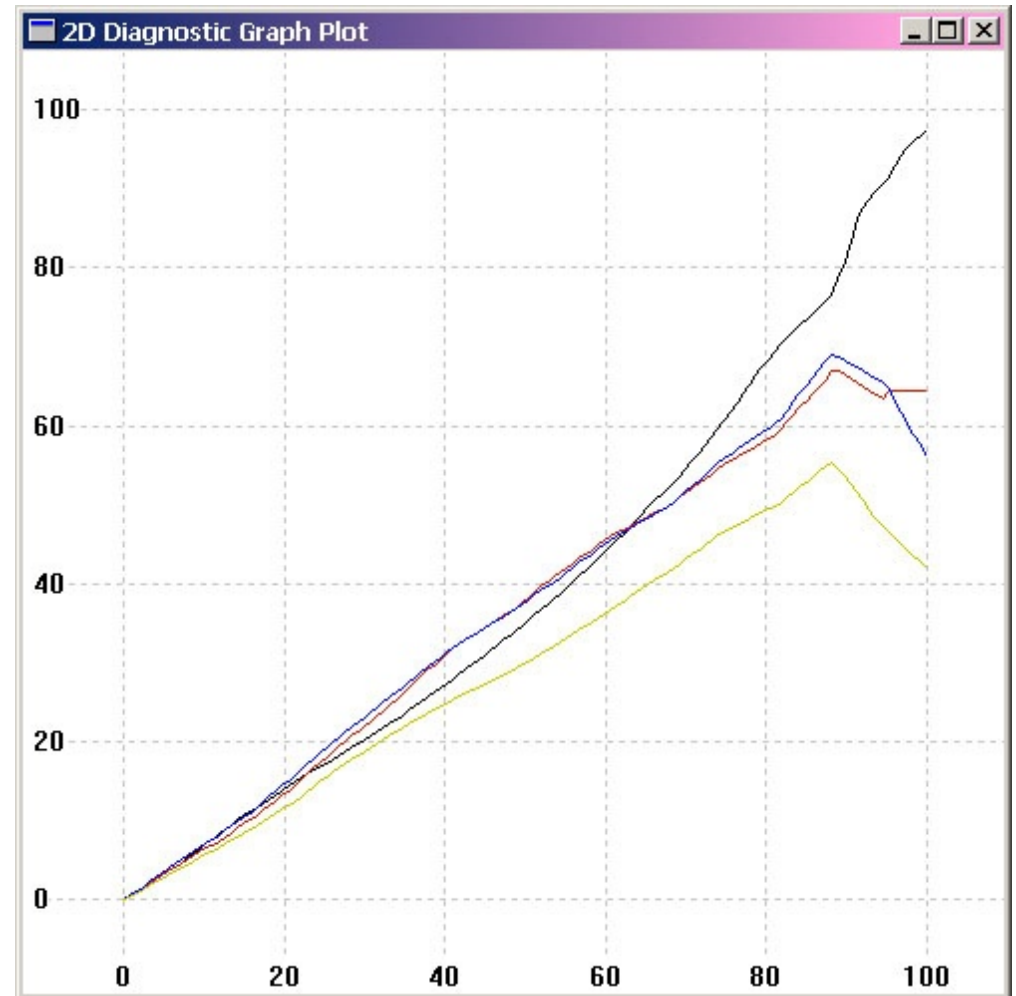
```
xicclu -g -kp 0 0 .87 .80 .65 -1270 -fif -ir PrinterBt.icm
```

We can then apply the chosen curve to making the final profile:

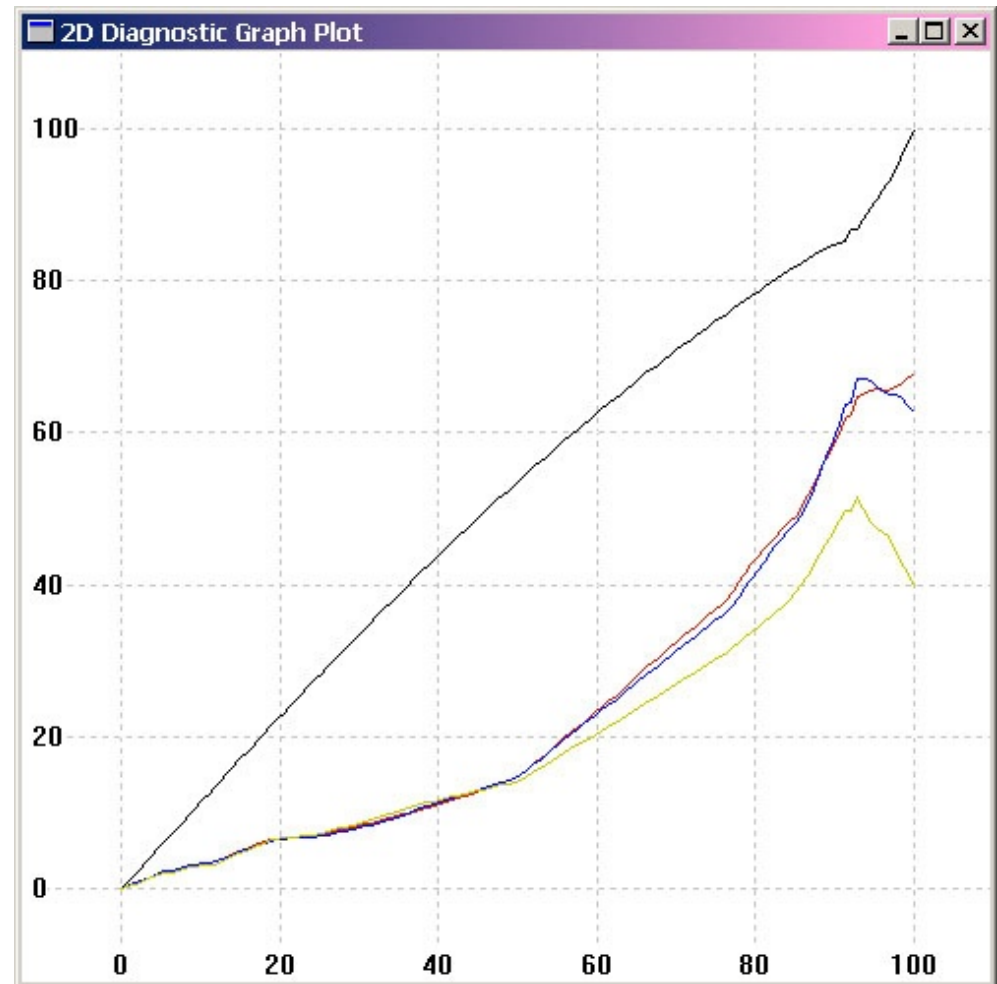
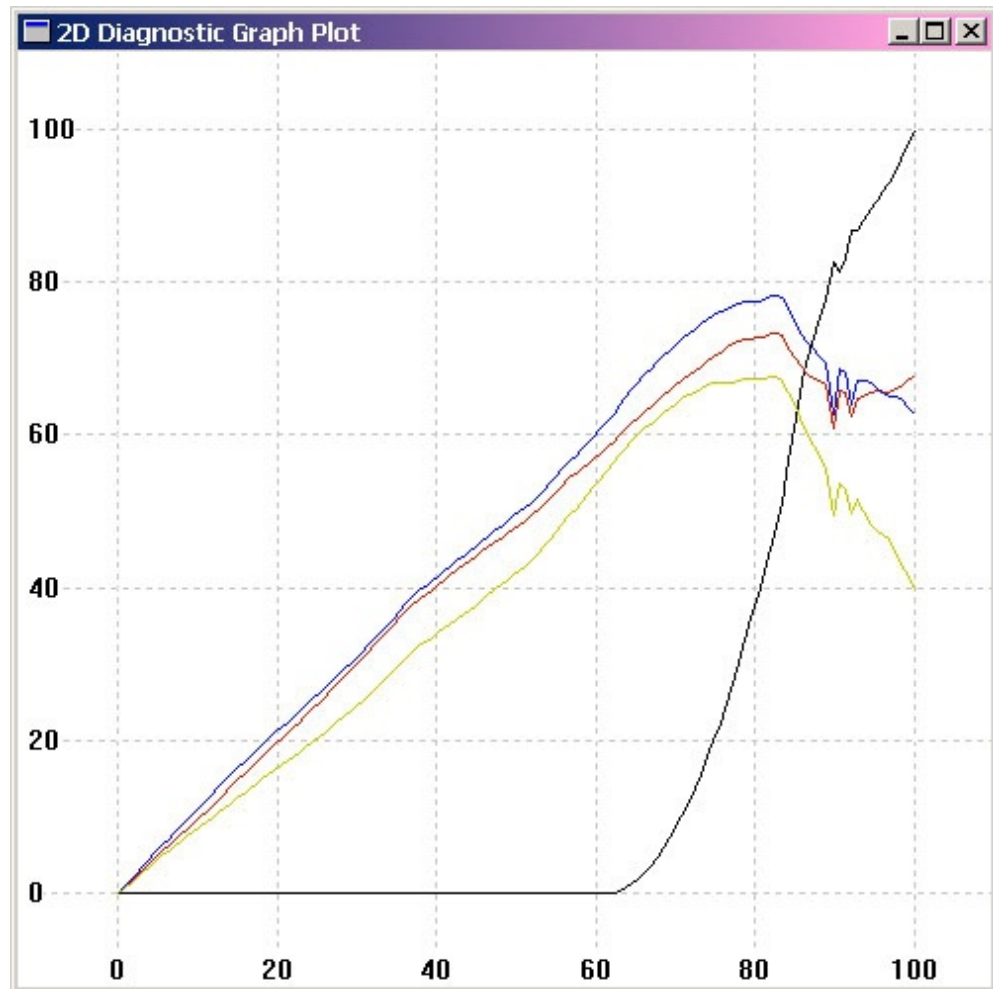
```
colprof -v -kp 0 0 .87 .80 .65 -D"Printer B" -qm  
-S sRGB.icm -cmt -dpp PrinterB
```

Check that we got the desired curve:

```
xicclu -g -fb -ir PrinterB.icm
```



```
xicclu -g -kp 0 .7 .93 .87 1.0 -1270 -fif -ir PrinterBt.icm
```



```
xicclu -g -kp 0 0 .93 .87 1.2 -1270 -fif -ir PrinterBt.icm
```



# colprof options - 1

usage: colprof [-options] inoutfile

- v                    Verbose mode
  
- A manufacturer    Manufacturer description string
- M model            Model description string
- D description      Profile Description string (Default "inoutfile")
- C copyright       Copyright string
- q lmhu             Quality - Low, Medium (def), High, Ultra
- b [lmhun]          Low quality B2A table - or specific B2A quality or none for input device
  
- ni                  Don't create input (Device) shaper curves
- np                  Don't create input (Device) grid position curves
- no                  Don't create output (PCS) shaper curves
- nc                  Don't put the input .ti3 data in the profile
  
- k zhxr             Black value target: z = zero K,  
                    h = 0.5 K, x = max K, r = ramp K (def.)
- k p stle stpo enpo enle shape  
                    stle: K level at White 0.0 - 1.0  
                    stpo: start point of transition Wh 0.0 - Bk 1.0  
                    enpo: End point of transition Wh 0.0 - Bk 1.0  
                    enle: K level at Black 0.0 - 1.0  
                    shape: 1.0 = straight, 0.0-1.0 concave, 1.0-2.0 convex
  
- l tlimit           override total ink limit, 0 - 400% (default from .ti3)
- L klimit           override black ink limit, 0 - 100% (default from .ti3)

# colprof options - 2

-a lxXgsmGS	Algorithm type override l = Lab cLUT (def.), x = XYZ cLUT, X = display XYZ cLUT + matrix g = gamma+matrix, s = shaper+matrix, m = matrix only, G = single gamma+matrix, S = single shaper+matrix
-u	If Lut input profile, make it absolute (non-standard)
-U scale	If input profile, scale media white point by scale
-i illum	Choose illuminant for print/transparency spectral data: A, C, D50 (def.), D65, F5, F8, F10 or file.sp
-o observ	Choose CIE Observer for spectral data: 1931_2 (def), 1964_10, S&B 1955_2, shaw, J&V 1978_2
-f	Use Fluorescent Whitening Agent compensation
-r avgdev	Average deviation of device+instrument readings as a percentage (default 0.5)
-s src.icm	Apply gamut mapping to output profile perceptual B2A table for given source
-S src.icm	Apply gamut mapping to output profile perceptual and saturation B2A table
-nP	Use colormetric source gamut to make output profile perceptual table
-nS	Use colormetric source gamut to make output profile saturation table
-g src.gam	Use source image gamut as well for output profile gamut mapping
-p absprof	Incorporate abstract profile into output tables

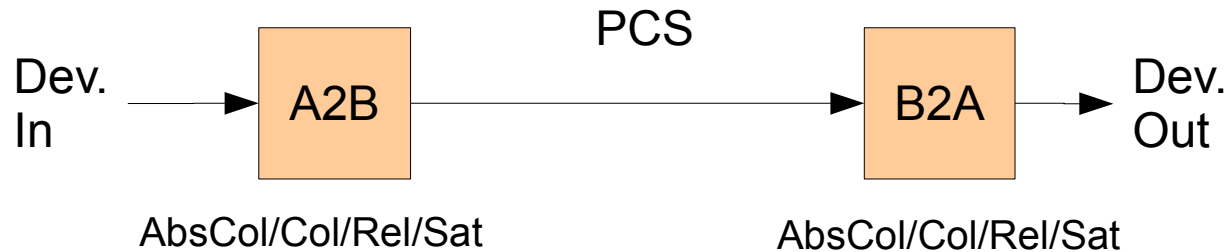
# colprof options - 3

-t intent       Override gamut mapping intent for output profile perceptual table:  
-T intent       Override gamut mapping intent for output profile saturation table:  
.  
r - White Point Matched Appearance [ICC Relative Colorimetric]  
.  
s - Enhanced Saturation [ICC Saturation]  
.  
-c viewcond     set input viewing conditions for output profile CIECAM02 gamut mapping,  
                  either an enumerated choice, or a parameter  
-d viewcond     set output viewing conditions for output profile CIECAM02 gamut mapping  
                  either an enumerated choice, or a parameter  
                  Also sets out of gamut clipping CAM space.  
                  either an enumerated choice, or a series of parameters:value changes  
pp - Practical Reflection Print (ISO-3664 P2)  
pe - Print evaluation environment (CIE 116-1995)  
pc - Critical print evaluation environment (ISO-3664 P1)  
mt - Monitor in typical work environment  
mb - Bright monitor in bright work environment  
md - Monitor in darkened work environment  
jm - Projector in dim environment  
jd - Projector in dark environment  
pcd - Photo CD - original scene outdoors  
ob - Original scene - Bright Outdoors  
cx - Cut Sheet Transparencies on a viewing box  
-P               Create gamut gammap\_p.wrl and gammap\_s.wrl diagnostics  
-O outputfile   Override the default output filename.  
infile           Base name for input.ti3/output.icm file

# Device Link Creation - 1

**collink** links two device profiles together. It has three modes:

## Simple Mode



(This corresponds with typical CMM linking modes)

```
collink -v -qm -s -ip -op sRGB.icc PrinterA.icc sRGB2PrinterA.icc
```

Got options

Configured options

Loading input A2B table

Using Y to L\* and L\* to Y curves for input

Loading output B2A table

Gamut mapping mode is 'Simple'

Creating link profile

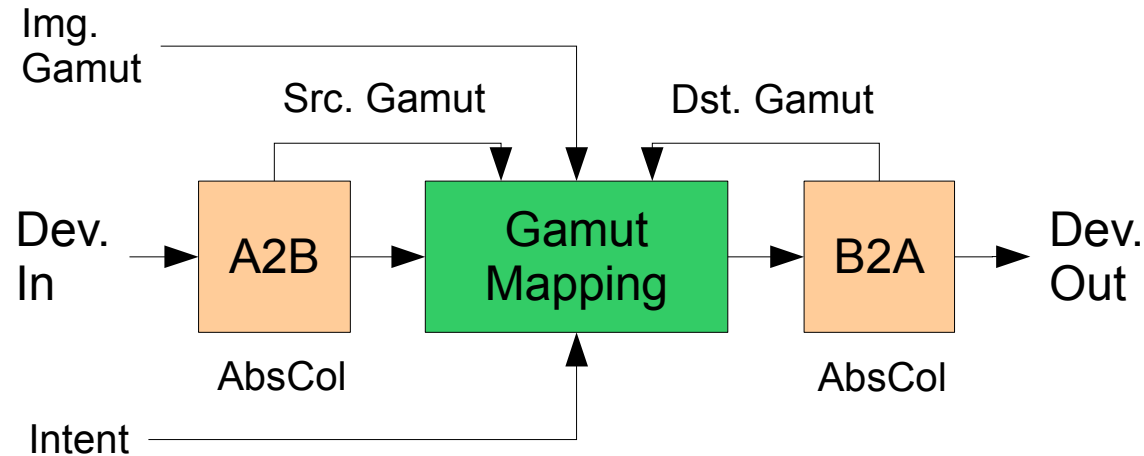
Filling in Lut table

100%

Writing out file

# Device Link Creation - 2

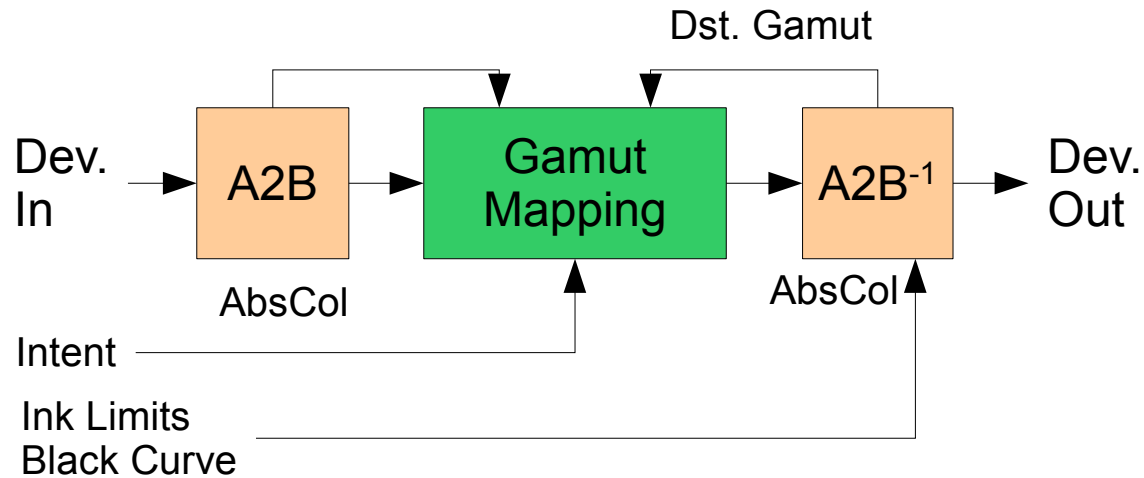
## Gamut Mapping Mode



```
collink -v -qm -g -ip -cmt -dpp sRGB.icc PrinterA.icc sRGB2PrinterA.icc
```

# Device Link Creation - 3

## Gamut Mapping using inverse A2B Mode



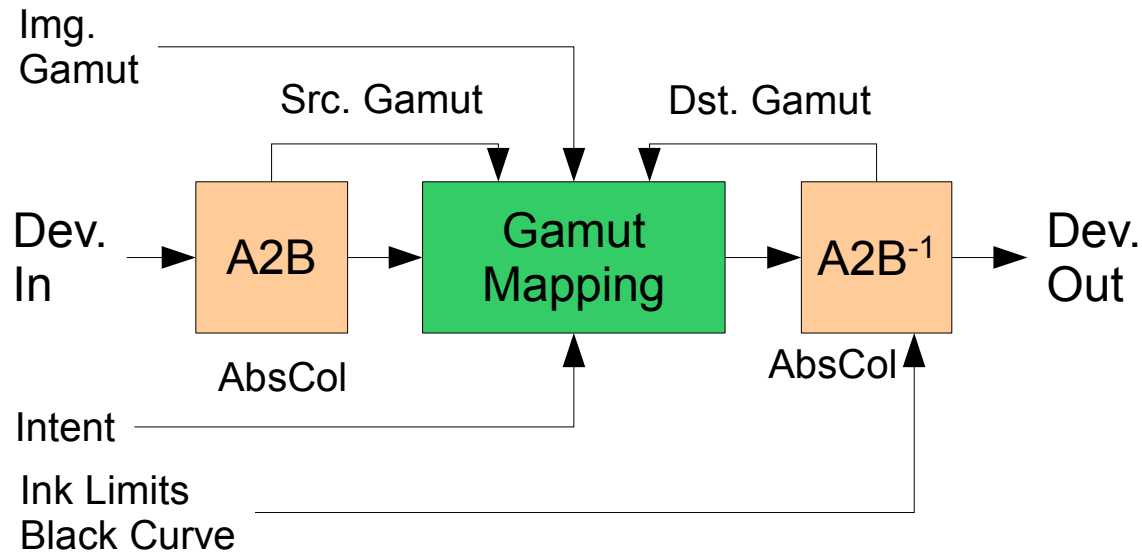
```
collink -v -qm -G -ip -cmt -dpp -kp 0 0 .87 .80 .65 -1270  
sRGB.icc PrinterB.icc sRGB2PrinterB.icc
```

# Device Link Creation - 4

## Gamut Mapping &

### Gamut Mapping using inverse A2B Modes

can also be used with an Image specific source gamut:



```
collink -v -qm -G Image.gam -ip -cmt -dpp -kp 0 0 .87 .80 .65 -1270  
sRGB.icc PrinterB.icc sRGB2PrinterB.icc
```



# Device Link Creation - 5

Special functions options for Gamut Mapping using inverse A2B Mode with CMYK output:

- |       |   |                                   |
|-------|---|-----------------------------------|
| -kt   | Transfer K from source to destination           | Black separation is maintained.   |
| -ke   | Retain K of destination B2A table               | Emulate destination behaviour.    |
| -f    | Force neutral colors to be K only output        | Good for RGB text.                |
| -fk   | Force K only neutral colors to be K only output | Maintain K only purity.           |
| -F    | Force all colors to be K only output            | Good for a monochrome conversion. |
| -fcmy | Force 100% C,M or Y only to stay pure           | Maintain other inks purity.       |

# Linking intents - 1

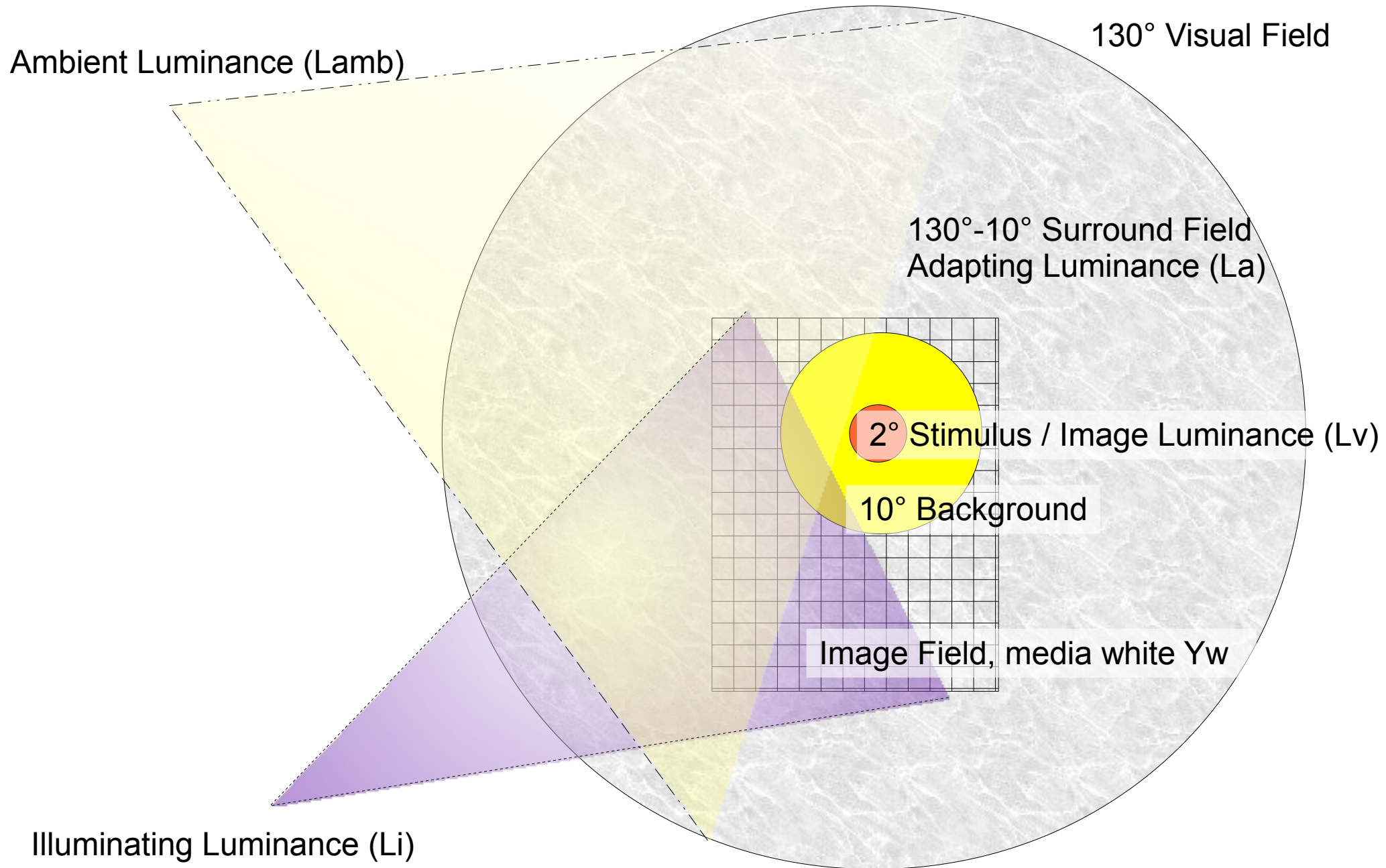
- a** Absolute Colorimetric (in Jab) [ICC Absolute Colorimetric]:  
Map absolute Jab to Jab and clip out of gamut.
- aw** Absolute Colorimetric (in Jab) with scaling to fit white point:  
scale source to avoid clipping the white point, and  
map absolute Jab to Jab and clip out of gamut.
- aa** Absolute Appearance:  
Map Jab to Jab and clip out of gamut.
- r** White Point Matched Appearance [ICC Relative Colorimetric]:  
Align neutral axes and linearly map white point, then  
map Jab to Jab and clip out of gamut.
- la** Luminance axis matched Appearance:  
Align neutral axes and linearly map white and black points, then  
map Jab to Jab and clip out of gamut.

[“Jab” is CIECAM02 analog of  $L^*a^*b^*$  colorspace.]

# Linking intents - 2

- p** Perceptual (Preferred) (Default) [ICC Perceptual]:  
Align neutral axes and perceptually map white and black points, perceptually compress out of gamut and map Jab to Jab.
- ms** Saturation:  
Align neutral axes and perceptually map white and black points, perceptually compress and expand to match gamuts and map Jab to Jab.
- s** Enhanced Saturation [ICC Saturation]:  
Same as "ms" but enhance saturation.
- al** Absolute Colorimetric (Lab):  
Map absolute  $L^*a^*b^*$  to  $L^*a^*b^*$  and clip out of gamut.
- rl** White Point Matched Appearance (Lab):  
Align neutral axes and linearly map white point, then map  $L^*a^*b^*$  to  $L^*a^*b^*$  and clip out of gamut.

# CIECAM02 Viewing conditions - 1



# CIECAM02 Viewing conditions - 2

s:surround	a = average, m = dim, d = dark, c = transparency (default average)
w:X:Y:Z	Adapted white point as XYZ (default media white)
w:x:y	Adapted white point as x, y
a:adaptation	Adaptation luminance in $\text{cd.m}^2$ (default 50.0)
b:background	Background % of image luminance (default 20)
f:flare	Flare light % of image luminance (default 1)
f:X:Y:Z	Flare color as XYZ (default media white)
f:x:y	Flare color as x, y

Assume Lambertian reflectance, so  $\text{Luminance} = \text{Illuminance} / \pi$

Often assume  $L_a = (L_{amb} \text{ or } L_i) \times 20\%$  due to grey world assumption.

For print,  $L_v = L_i \times Y_w$

$L_a/L_v == 0\%$  dark surround

$L_a/L_v 0 - 20\%$  dim surround

$L_a/L_v > 20\%$  average surround

Background relative luminance is typically assumed to be  $\approx 20\%$  (grey world)

Flare is stray light reflection.

# CIECAM02 Viewing conditions - 3

Preset viewing conditions:

Key	Description	View Cond.	La (cd/m <sup>2</sup> )	Yb (%)	Yf (%)
pp	Practical Reflection Print (ISO-3664 P2)	Avg.	32	20	1
pe	Print evaluation environment (CIE 116-1995)	Avg.	64	20	1
pc	Critical print evaluation environment (ISO-3664 P1)	Avg.	127	20	1
mt	Monitor in typical work environment	Avg.	22	20	2
mb	Bright monitor in bright work environment	Avg.	42	20	2
md	Monitor in darkened work environment	Dim	4	20	1
jm	Projector in dim environment	Dim	10	20	1
jd	Projector in dark environment	Dark	10	20	1
pcd	Photo CD - original scene outdoors	Avg.	320	20	0
ob	Original scene - Bright Outdoors	Avg.	2000	20	0
cx	Cut Sheet Transparencies on a viewing box	Cut Sheet	53	20	1

# Transforming raster files – cctiff - 1

The **cctiff** tool is capable of linking an arbitrary sequence of device profiles, device links, abstract profiles and calibration curves.

By default an 8 bit or 16 bit integer transform will be created to implement the overall transform with a very fast conversion.

Conversion using device link:

```
cctiff Source2Dest.icc infile.tif outfile.tif
```

Conversion with calibration:

```
cctiff -ip Source.icc -ip Dest.icc Dest.cal infile.tif outfile.tif
```

Conversion with abstract profile:

```
cctiff -ip Source.icc abstract.icc -ip Dest.icc infile.tif outfile.tif
```

Embed an ICC profile in a TIFF file:

```
cctiff -e profile.icc infile.tif outfile.tif
```

Convert from RGB to L\*a\*b\* with CIE encoding:

```
cctiff -t1 -ip Source.icc infile.tif outfile.tif
```



# Transforming raster files – cctiff - 2

```
usage: cctiff [-options] { [-i intent] profile.icm | calbrtn.cal ...} infile.tif outfile.tif
-v          Verbose.
-c          Combine linearisation curves into one transform.
-p          Use slow precise correction.
-r n        Override the default CLUT resolution
-t n        Choose TIFF output encoding from 1..n
-a          Read and Write planes > 4 as alpha planes
-I          Ignore any file or profile colorspace mismatches
-D          Don't append or set the output TIFF description
-e profile.[icm | tiff]  Optionally embed a profile in the destination TIFF file.
```

Then for each profile in sequence:

```
-i intent    p = perceptual, r = relative colorimetric,
              s = saturation, a = absolute colorimetric
-o order     n = normal (priority: lut > matrix > monochrome)
              r = reverse (priority: monochrome > matrix > lut)
```

```
profile.[icm | tiff]  Device, Link or Abstract profile
                      ( May be embedded profile in TIFF file)
```

or each calibration file in sequence:

```
-d dir       f = forward cal. (default), b = backwards cal.
calbrtn.cal  Device calibration file.
```

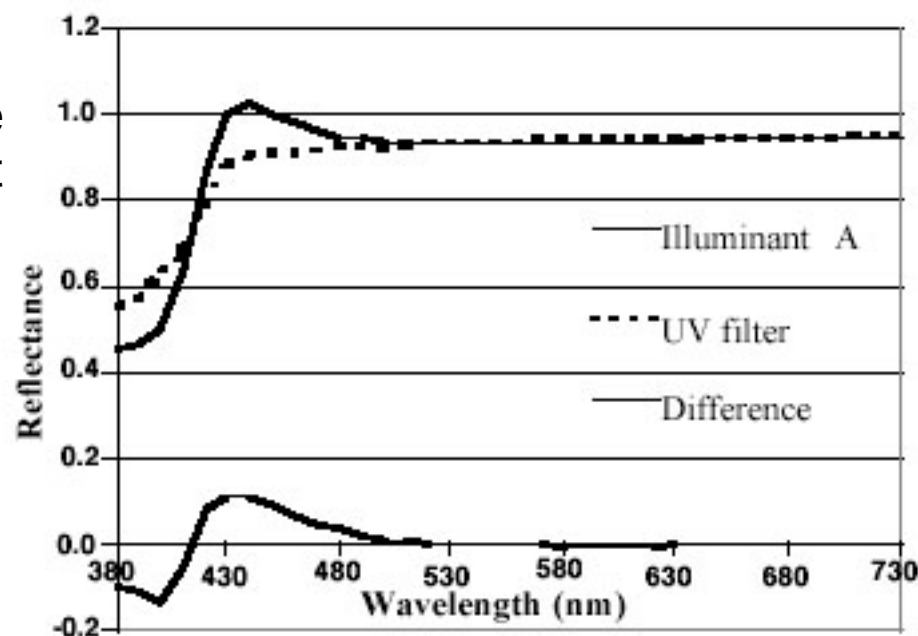
```
infile.tif    Input TIFF file in appropriate color space
outfile.tif   Output TIFF file
```

# Fluorescent Whitener Additive Compensation - 1

Everywhere that Argyll accepts spectral reflectance measurements and converts them into CIE XYZ values, both the observer model and illuminating spectrum can be chosen for this conversion. For maximum ICC compatibility the 1932 2° observer and D50 illuminant would be chosen, but other choices are possible to account for real world viewing conditions.

The presence of Fluorescent Whitener Additive (FWA) makes computing the XYZ from reflectance and illuminant inaccurate because FWA emits light at a different wavelength from which it absorbs it, breaking the simple reflectance model used. The shape of the instrument illuminant spectrum affects the apparent reflectance during measurement due to the level of Ultra Violet irradiation and resulting FWA response.

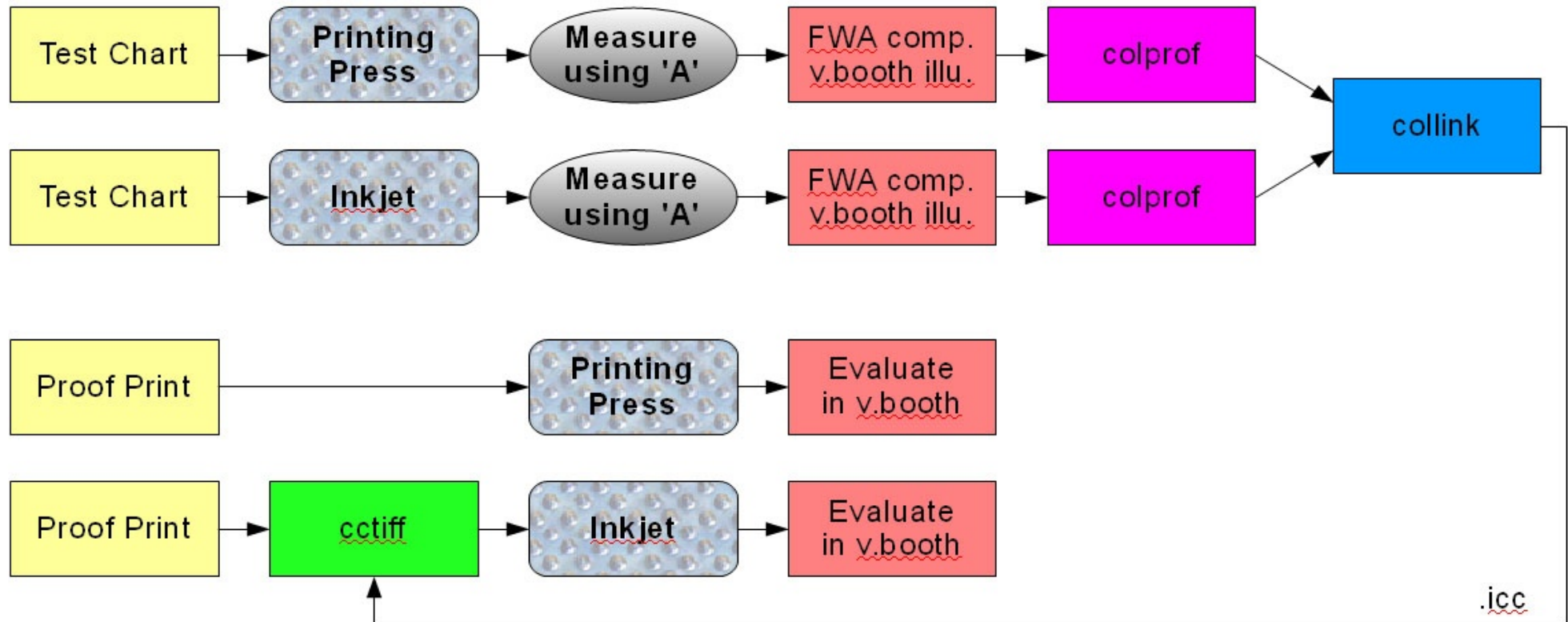
FWA compensation computes the XYZ as if the instrument had measured the samples under the target illumination, by creating a model of FWA response and using it in the calculation. Spectral measurements from a non-UV filtered instrument are required for this.



Inkjet paper, spectral reflectance under illuminant A, with and without instrument UV filter.

# Fluorescent Whitener Additive Compensation - 2

The most common situation for employing FWA compensation, is in proofing.

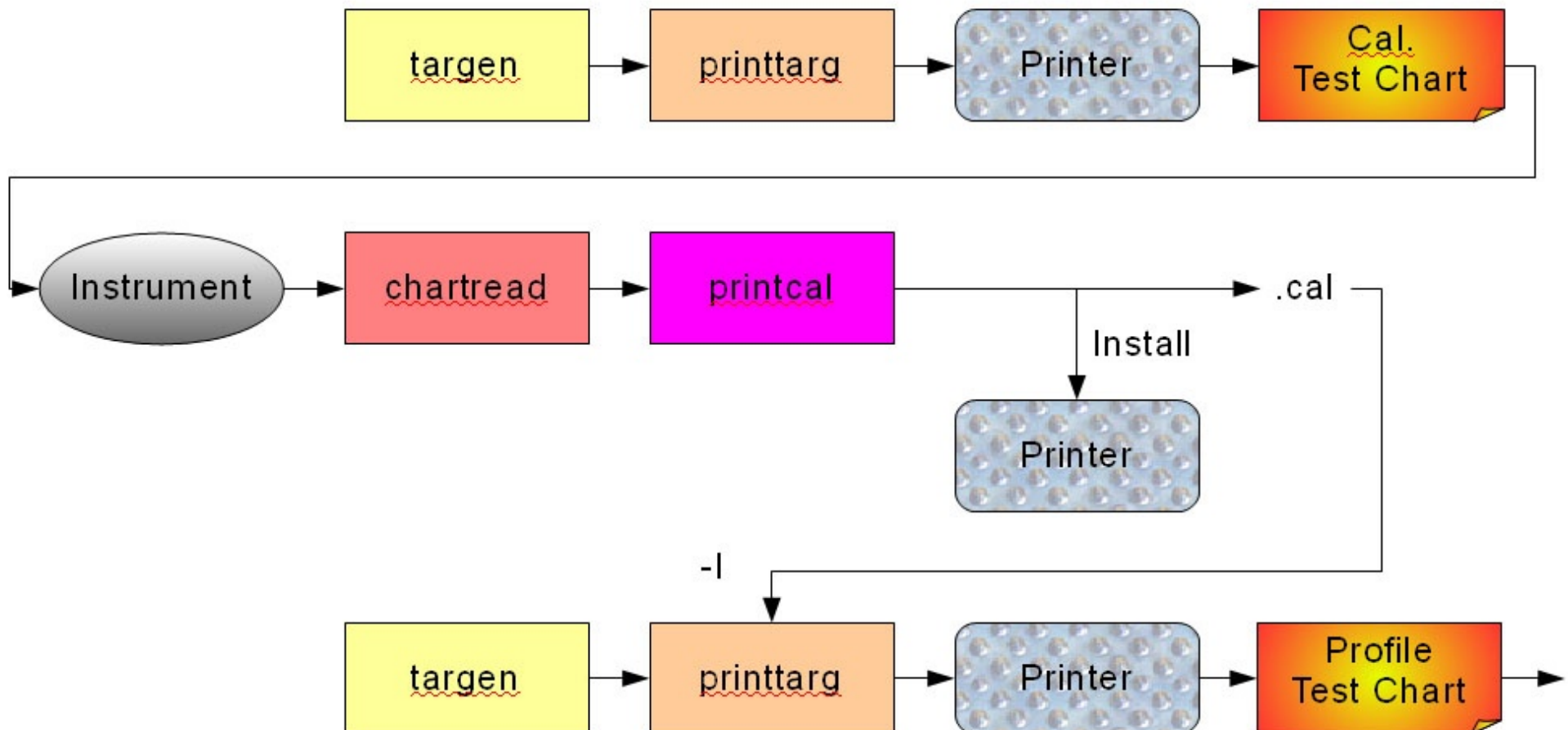


More details at <<http://www.argyllcms.com/doc/FWA.html>>

# Printer Calibration - 1

Argyll has the ability to create per-channel device space calibration curves for print devices, that can then be used to improve the behaviour of the device, making a subsequent profile fit the device more easily and also allow day to day correction of device drift without resorting to a full re-profile.

Printer with native calibration capability



# Printer Calibration - 2

Creating a calibration is similar to profiling: Create test target, print it, measure it, then create the calibration.

Let's consider two devices in our examples, "PrinterA" which is an "RGB" printer device, and "PrinterB" which is CMYK. We'll create a 50 steps per channel calibration test charts for our devices:

```
targen -v -d2 -s50 -e3 -f0 PrinterA_c
```

```
targen -v -d4 -s50 -e4 -f0 PrinterB_c
```

Then use **printtarg** and print to the printer in its un-profiled and un-calibrated mode, then use **chartread** read the chart and create the .ti3 file. The **printcal** tool then creates the initial calibration file from the .ti3 file:

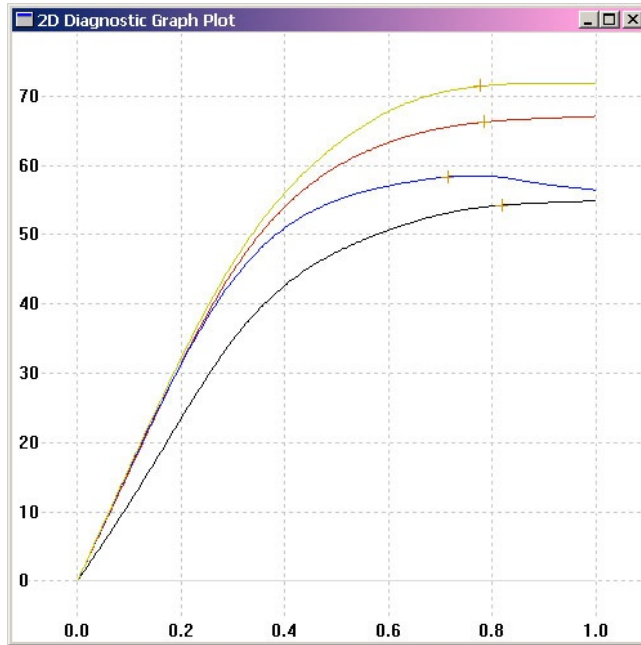
```
printcal -v -p -i PrinterB_c
```

The resulting PrinterB\_c.cal file can then be installed into the printer.

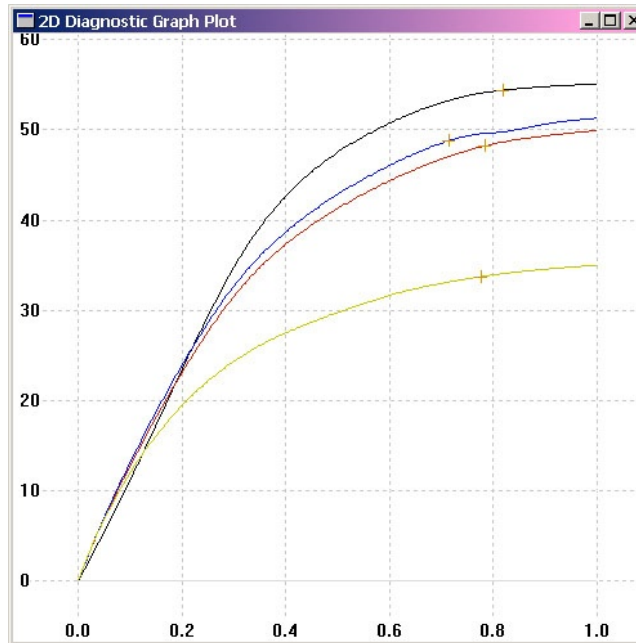
To make subsequent profiling aware of the calibration for correct ink limit calculations, we add it to the profiling printtarg using the **-I** option:

```
printtarg ... -I PrinterA_c.cal PrinterA
```

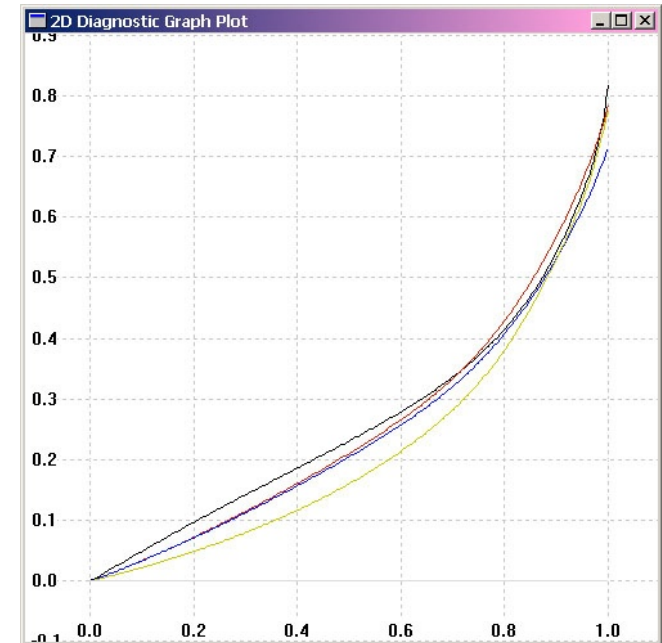
# Printer Calibration - 3



Absolute DE



Relative DE

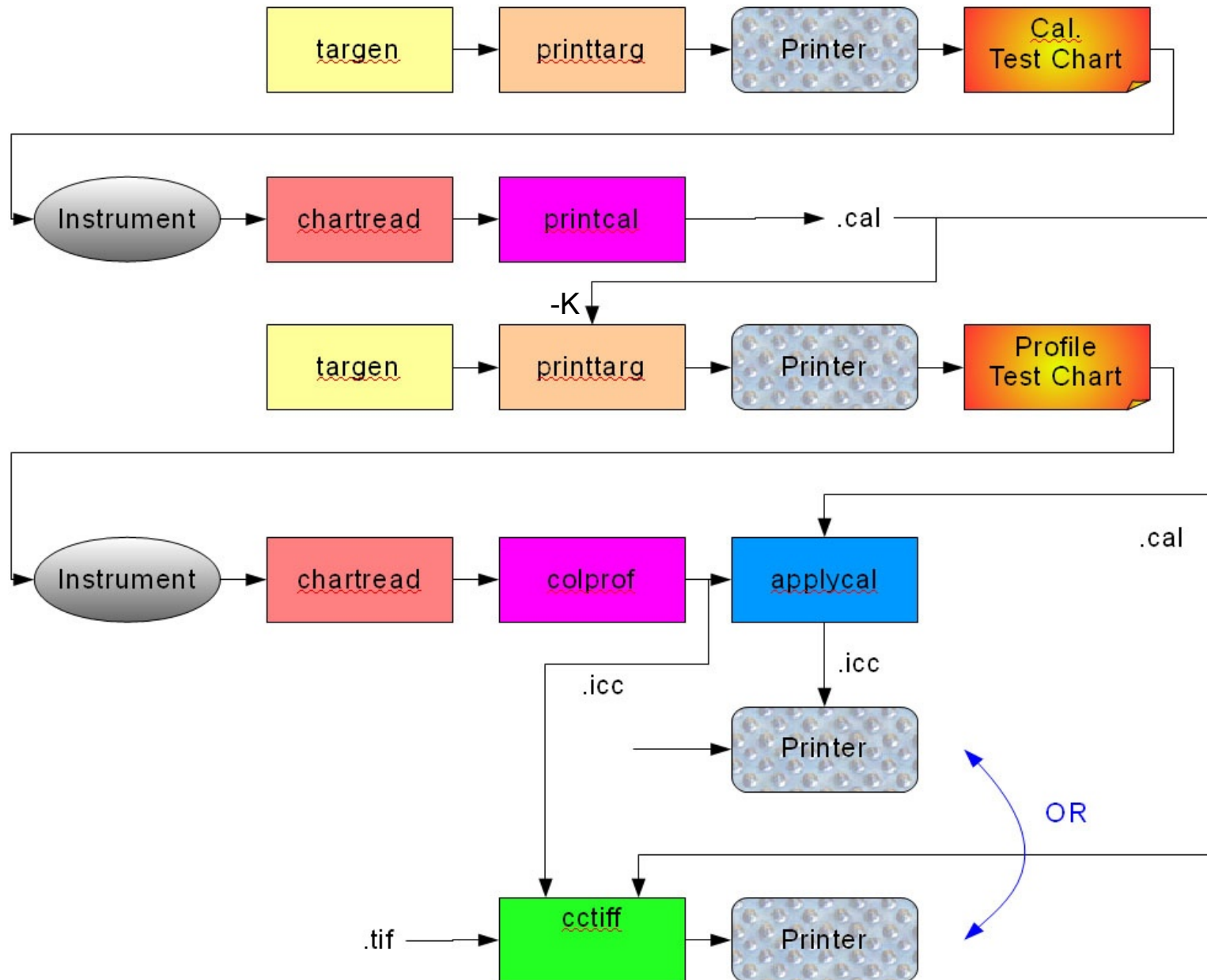


Calibration curves

```
usage: printcal [-options] [prevcal] inoutfile
-v verbosity      Verbose mode
-p                Plot graphs.
-i                Initial calibration, set targets, create .cal
-r                Re-calibrate against previous .cal and create new .cal
-e                Verify against previous .cal
-d                Go through the motions but don't write any files
-x# percent       Set maximum device percentage target
-n# deltaE        Set white minimum deltaE target
-t# percent       Set 50% transfer curve percentage target
                  # = 0123..., rgb, cmyk etc.
-a                Create an Adobe Photoshop .AMP file as well as a .cal
prevcal           Base name of previous .cal file for recal or verify.
inoutname         Base name of input .ti3 file, output .cal file
```

# Printer Calibration - 4

Printer without native calibration capability





# Printer Calibration - 5

In a workflow without native calibration capability, the calibration curves would be used with **printarg** to apply the calibration to the test patch samples during subsequent profiling, as well as embedding it in the resulting .tif to allow all the tools to be able to compute final device value ink limits:

```
printtarg -v -ii1 -pA4 -K PrinterA_c.cal PrinterA
```

To apply calibration to an ICC profile, so that a calibration unaware CMM can be used:

```
applycal PrinterA.cal PrinterA.icm PrinterA_cal.icm
```

To apply colour management and calibration to a raster image instead:

```
cctiff Source2Destination.icm PrinterA_c.cal infile.tif outfile.tif
```

Another useful tool is **synthcal**, that allows creating linear or synthetic calibration files for disabling calibration or testing.

# Gamut visualization and comparison - 1

First requirement: a VRML viewer:

Often web browser plug-ins, although standalone application exist. X3D capable viewers often support VRML97 too.

MSWindows : The Cosmo plugin is still one of the best. It may not work with recent browsers though (I use it in a copy of Netscape 4.73)  
Gview – standalone, still available if you look for gview.zip

MAC OS X:

Linux: FreeWRL

There are many others, but all have their frustrations. (We're still waiting for the day when every browser comes with 3D capability built in.)

[As well as tools specifically used to create & view gamuts (**iccgamut**, **tiffgamut**, **viewgam**) some other tools create diagnostic **VRML** output, such as **collink -P**.]

# Gamut visualization and comparison - 2

**iccgamut** to create a gamut and optional VRML file from an ICC profile.

Need to choose level of detail, ICC table, intent, colorspace (L\*a\*b\* or CIECAM02 Jab), ink limits for CMYK forward lookups.

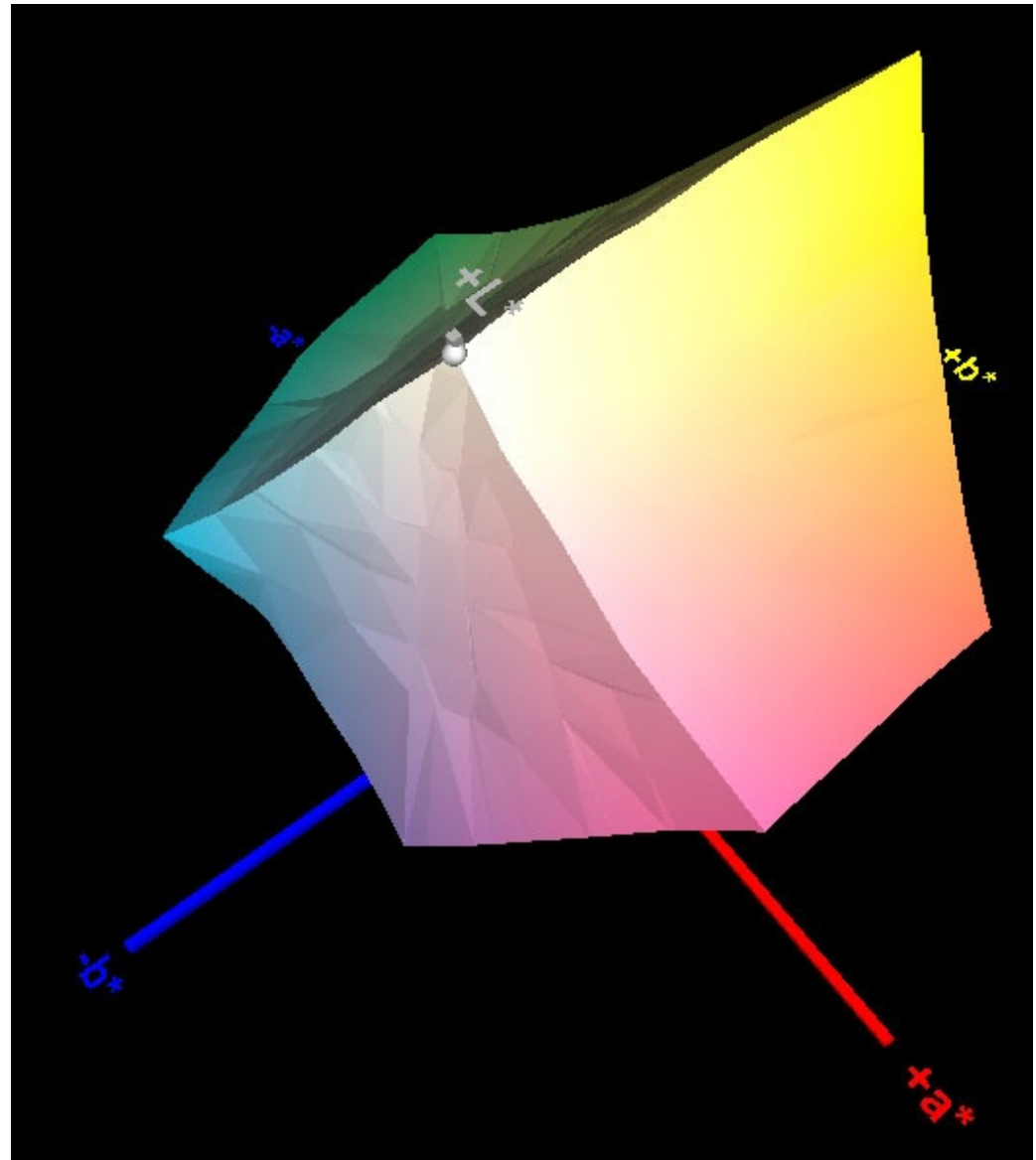
```
iccgamut -w -ff -ia -pl -l270  
PrinterB.icc
```

Results in PrinterB.gam and PrinterB.wrl

Similarly, **tiffgamut** is used to create a gamut from a TIFF image:

```
tiffgamut -w -ia -pl  
PrinterB.icc Image.tif
```

Results in Image.gam and Image.wrl



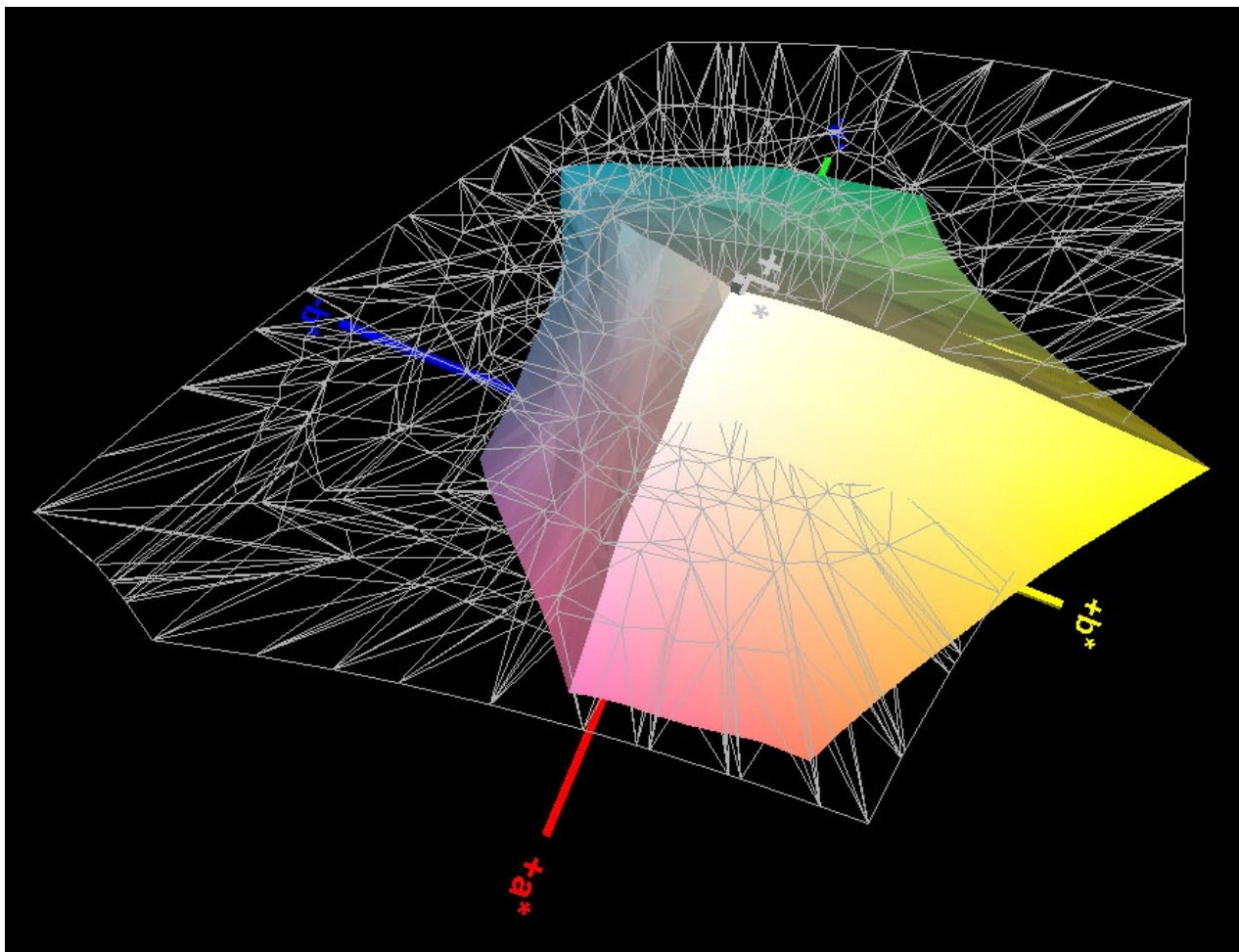
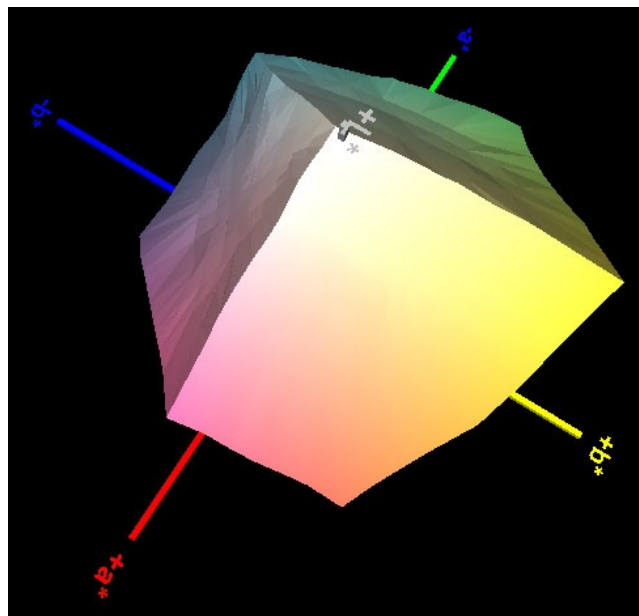
# Gamut visualization and comparison - 3

**viewgamut** can then be used to view multiple gamuts:

```
viewgamut PrinterB.gam PrinterA.gam outfile.wrl
```

There are options to set type of rendering (solid, wireframe), colour, transparency. The intersecting volume of two gamuts can also be computed and visualized:

```
Intersecting volume  
  = 352085.2 cubic units  
'PrinterB.gam' volume  
  = 385388.5 cubic units,  
    intersect = 91.36%  
'PrinterA.gam' volume  
  = 899261.2 cubic units,  
    intersect = 39.15%
```



Quick overview  
of other tools and topics

# Diagnostic Tools

- spotread** - Use an instrument to read and save individual readings. Show spectral plots, compute colour temperature, CRI. Supports transmission, emission display, projector, flash modes.
- iccdump** - Show tag contents of an ICC profile. Selectable detail.
- xicclu** - Transform colours through an ICC profile, including inverse A2B.
- profcheck** - Check profile colorimetric forward transform against measurements.
- invprofcheck** - Check profile forward to backwards accuracy.
- verify** - Check two sets of measurements against each other.

## Other Tools

- extracticc** - Extract ICC profile from a TIFF file.
- extracttttag** - Extract text tag from ICC profile (e.g. 'targ' tag).
- fakeread** - Can be used for testing or turning profile behaviour back into test readings for re-profiling – e.g. create a cLUT based sRGB profile that does perceptual and saturation mapping.

# Display calibration & Profiling

**dispcal** - Calibrate and simple matrix/shaper profile a display.

Assist setting display controls.

Can set transfer curve shape, white point, brightness.

Ambient light adjustment using CIECAM02.

Black point options.

Multi-monitor support.

Can use external tools to interact with display and/or instrument allowing remote display and/or unsupported instruments to be used.

(Is only capable of setting graphic card Lookups)

**dispread** - Use a .ti1 target and read the response from a display into a .ti3, that can then use the flexibility of **colprof**.

Can use external tools to interact with display and/or instrument allowing remote display and/or unsupported instruments to be used.

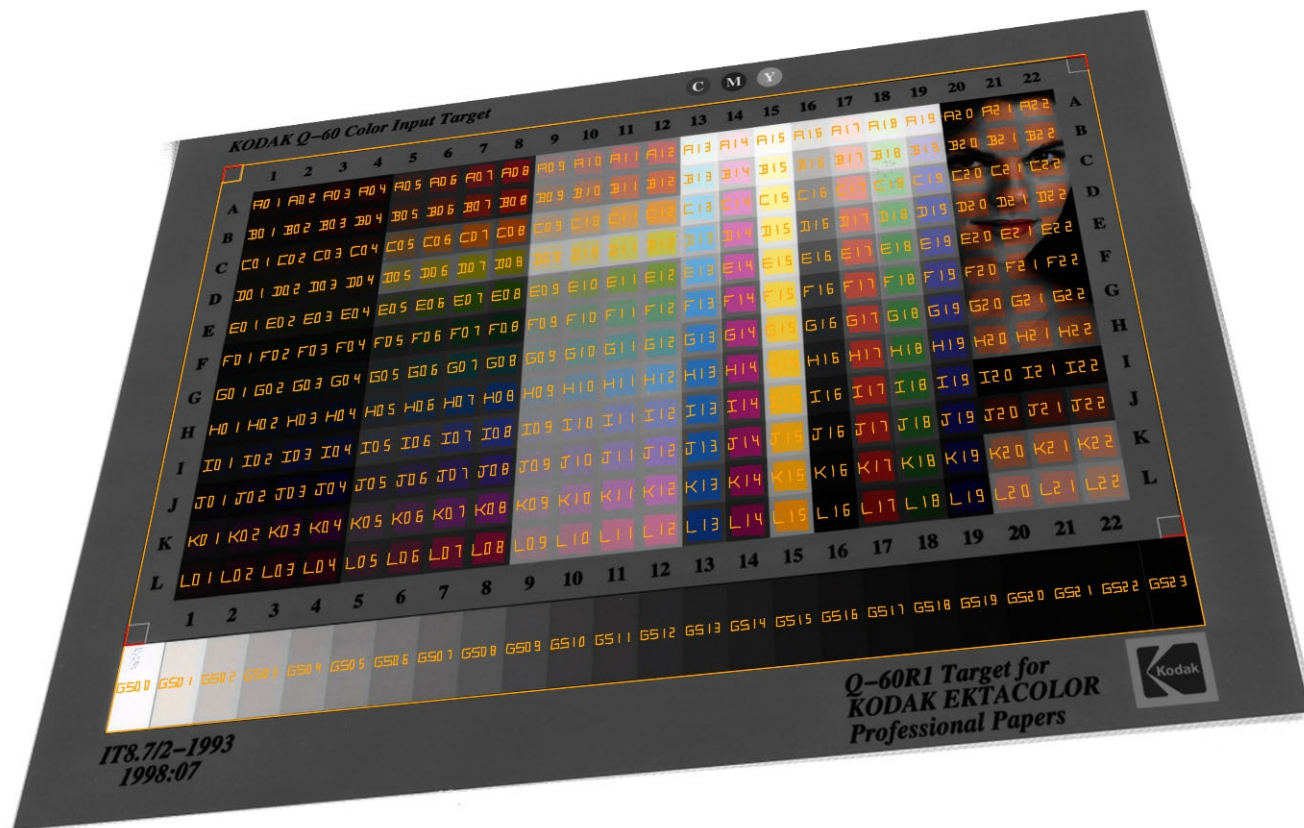
**dispwin** - General display test utility.

Can also install and/or load display calibration and/or ICC profiles.

# Camera & Scanner profiling

## chartread

- Tool that auto-recognizes an image of a test chart and converts it into a .ti3 file that can then be use with **colprof**.  
Can also be used for other similar purposes such as obtaining general raster patch values, emulating a colorimeter using a scanner etc.  
Optional perspective distortion correction.





# Softproofing link

Often it is desirable to get an idea what a particular device's output will look like using a different device. Typically this might be trying to evaluate print output using a display. Often it is sufficient to use an absolute or relative colorimetric transform from the print device space to the display space, but while these provide a colorimetric preview of the result, they do not take into account the subjective appearance differences due to the different device conditions. It can therefore be useful to create a soft proof appearance transform using collink:

```
collink -v -qm -G -ila -cpp -dmt -t250 CMYKDest.icm Monitor.icm  
SoftProof.icm
```

We use the Luminance matched appearance intent, to preserve the subjective appearance of the target device which takes into account the viewing conditions and assumes adaptation to the differences in the luminance range, but otherwise do not attempt to compress or change the gamut.

# Tailoring test charts

## Chart pre-conditioning:

**targen** can use a previous or similar profile for a device to better distribute the test points so as to balance exploring device space, perceptual space and curvature.

## Adding specific test colours:

The .ti1 file can be manually edited to add specific device colours to test, such as proof print test colours. Using **xicclu** with a preliminary profile can be used to obtain device values from critical CIE values.

The .ti3 file for a device can also be augmented manually with spot measured colours that are particularly critical, such as proof print test colours.

## Importing readings:

Test chart results from other colour management systems can be imported using the **text2ti3** tool.

## Converting spectral to tri-stimulus:

**spec2cie** tool.

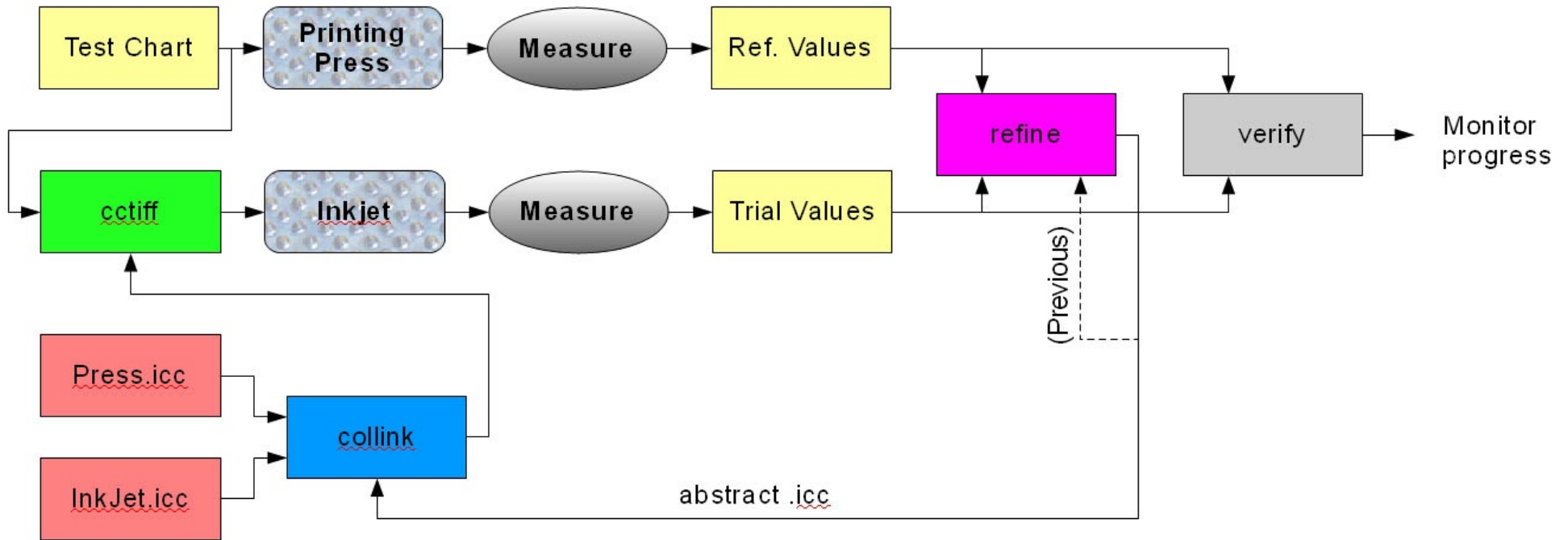
## Combining multiple readings:

**average** tool.

## Using spread sheet:

Spreadsheet programs such as Microsoft Excel and Open Office Calc can be used to import (as Text CSV) manipulate and export .ti1, .ti2 and .ti3 files.

# Refining proofing profiles



Direct profile workflows as well as device link workflow is supported as **colprof** also takes an abstract profile as a parameter.