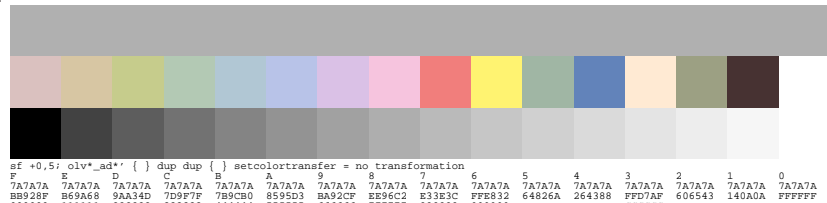


Technical information: http://o2.ps.bam.de

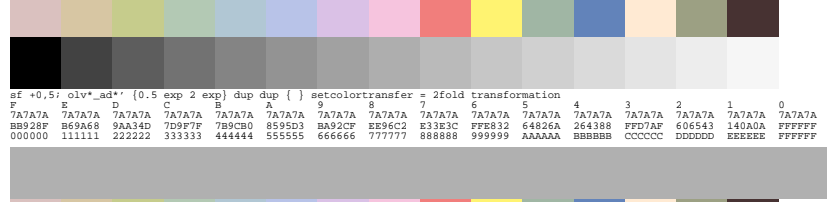
Image file version 1.4, 20010101, D8650E00

BAM registration: 20010101-D8650E00

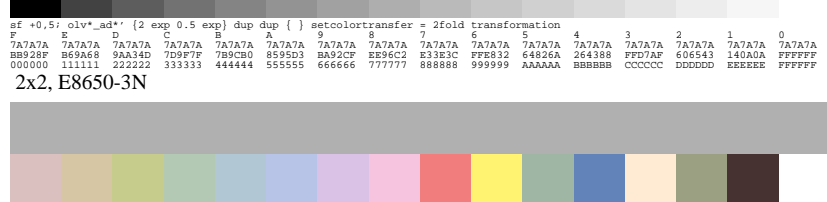
BAM-Reference material: code=rh4ra-D8650E00



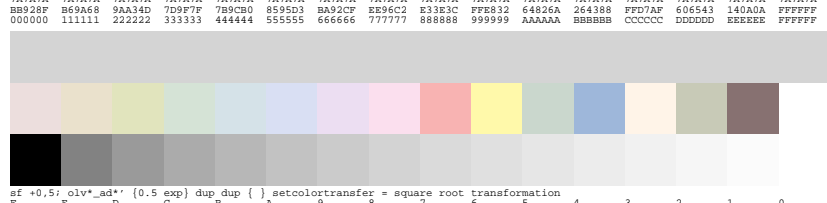
```
sf +0.5; olv*_ad* ( ) dup dup ( ) setcolortransfer = no transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A
BB928F B69A68 9AA34D 7D9F7F 7B9CB0 8595D3 BA92CF E996C2 E33E3C FFE832 64826A 264388 FFD7AF 606543 140A0A FFFFFF
000000 111111 222222 333333 444444 555555 666666 777777 888888 999999 AAAAAA BBBBbb CCCCcc DDDDDD EEEEE EEEEE FFFFFF
```



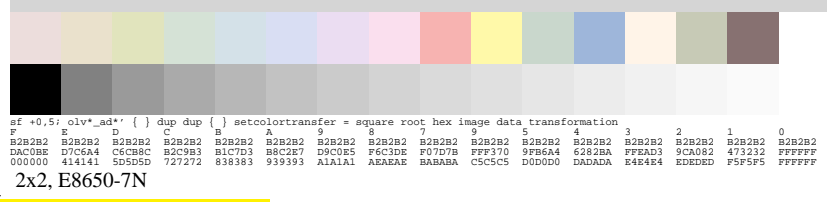
```
sf +0.5; olv*_ad* (0.5 exp 2 exp) dup dup ( ) setcolortransfer = 2fold transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A
BB928F B69A68 9AA34D 7D9F7F 7B9CB0 8595D3 BA92CF E996C2 E33E3C FFE832 64826A 264388 FFD7AF 606543 140A0A FFFFFF
000000 111111 222222 333333 444444 555555 666666 777777 888888 999999 AAAAAA BBBBbb CCCCcc DDDDDD EEEEE EEEEE FFFFFF
```



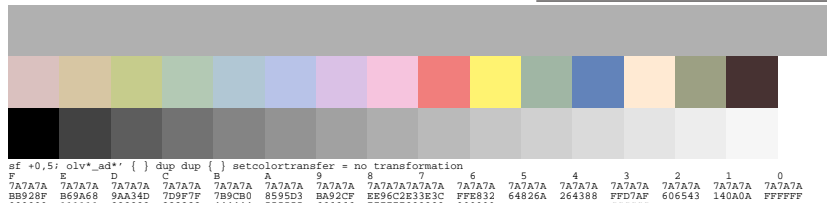
```
sf +0.5; olv*_ad* (2 exp 0.5 exp) dup dup ( ) setcolortransfer = 2fold transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A
BB928F B69A68 9AA34D 7D9F7F 7B9CB0 8595D3 BA92CF E996C2 E33E3C FFE832 64826A 264388 FFD7AF 606543 140A0A FFFFFF
000000 111111 222222 333333 444444 555555 666666 777777 888888 999999 AAAAAA BBBBbb CCCCcc DDDDDD EEEEE EEEEE FFFFFF
```



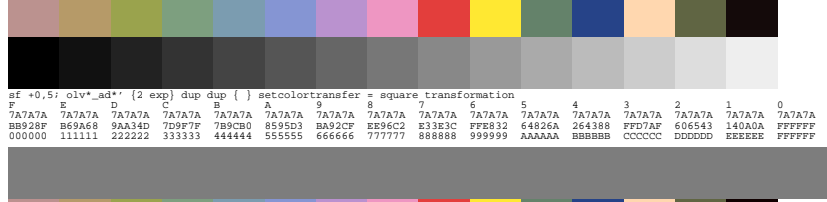
```
sf +0.5; olv*_ad* ( ) dup dup ( ) setcolortransfer = no transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A
BB928F B69A68 9AA34D 7D9F7F 7B9CB0 8595D3 BA92CF E996C2 E33E3C FFE832 64826A 264388 FFD7AF 606543 140A0A FFFFFF
000000 111111 222222 333333 444444 555555 666666 777777 888888 999999 AAAAAA BBBBbb CCCCcc DDDDDD EEEEE EEEEE FFFFFF
```



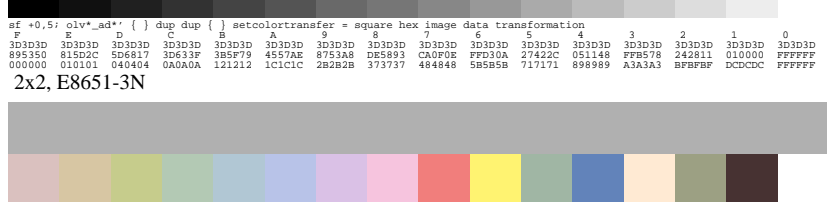
```
sf +0.5; olv*_ad* ( ) dup dup ( ) setcolortransfer = square root hex image data transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2
DAC08E D7C6A4 C6CB8C B2C9B3 B1C7D3 B8C2E7 D9C0E5 F6C3DE F07D7B FFF370 9FB6A4 6282BA B2EAD3 9CA082 473232 FFFFFFFF
000000 414141 5D5D5D 727272 838383 939393 A1A1A1 AEAEA EABABA C5C5C5 D0D0D0 DADADA E4E4E4 EDEDED F5F5F5 FFFFFFFF
```



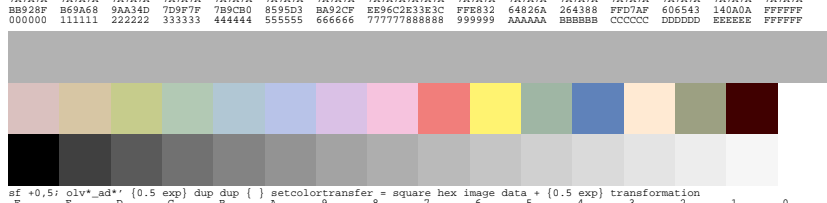
```
sf +0.5; olv*_ad* ( ) dup dup ( ) setcolortransfer = square transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A
BB928F B69A68 9AA34D 7D9F7F 7B9CB0 8595D3 BA92CF E996C2 E33E3C FFE832 64826A 264388 FFD7AF 606543 140A0A FFFFFF
000000 111111 222222 333333 444444 555555 666666 777777888888 999999 AAAAAA BBBBbb CCCCcc DDDDDD EEEEE EEEEE FFFFFF
```



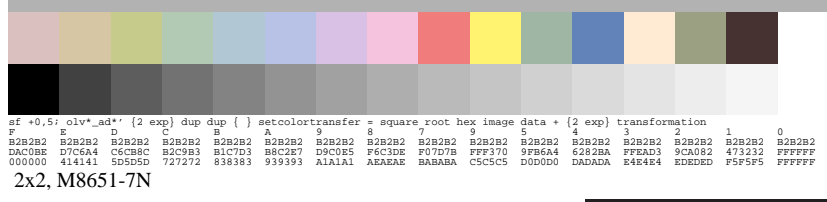
```
sf +0.5; olv*_ad* (2 exp) dup dup ( ) setcolortransfer = square transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A 7A7A7A
BB928F B69A68 9AA34D 7D9F7F 7B9CB0 8595D3 BA92CF E996C2 E33E3C FFE832 64826A 264388 FFD7AF 606543 140A0A FFFFFF
000000 111111 222222 333333 444444 555555 666666 777777 888888 999999 AAAAAA BBBBbb CCCCcc DDDDDD EEEEE EEEEE FFFFFF
```



```
sf +0.5; olv*_ad* ( ) dup dup ( ) setcolortransfer = square hex image data transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D
89535D 815D2C 5D6817 3D633F 3B5F79 4557AE 8753A8 DE5893 CA0F0E FFD30A 27422C 051148 FFB578 242811 010000 FFFFFFFF
000000 010101 040404 0A0A0A 121212 1C1C1C 2E2E2E 373737 484848 5B5B5B 717171 898989 A3A3A3 BFBFBF CDCDCD FFFFFFFF
```



```
sf +0.5; olv*_ad* (0.5 exp) dup dup ( ) setcolortransfer = square hex image data + (0.5 exp) transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D 3D3D3D
89535D 815D2C 5D6817 3D633F 3B5F79 4557AE 8753A8 DE5893 CA0F0E FFD30A 27422C 051148 FFB578 242811 010000 FFFFFFFF
000000 010101 040404 0A0A0A 121212 1C1C1C 2E2E2E 373737 484848 5B5B5B 717171 898989 A3A3A3 BFBFBF CDCDCD FFFFFFFF
```



```
sf +0.5; olv*_ad* (2 exp) dup dup ( ) setcolortransfer = square root hex image data + (2 exp) transformation
F E D C B A 9 8 7 6 5 4 3 2 1 0
B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2 B2B2B2
DAC08E D7C6A4 C6CB8C B2C9B3 B1C7D3 B8C2E7 D9C0E5 F6C3DE F07D7B FFF370 9FB6A4 6282BA FFEAD3 9CA082 473232 FFFFFFFF
000000 414141 5D5D5D 727272 838383 939393 A1A1A1 AEAEA EABABA C5C5C5 D0D0D0 DADADA E4E4E4 EDEDED F5F5F5 FFFFFFFF
```

Test chart no. 00 for Colour Management: No, square and square root + square root transfer