Altona Test Suite 2.0 · Technical Page · Evaluation template						
Patch	No issues	Very small deviations, but still acceptable	Deviations that may or may not be acceptable	Clearly unacceptable deviations	Some or all of the patch not rendered at all	Remarks
A	X					
B	X					
C	X					
D	X					
E	X					
F	X					
G				X		multiple blending color spaces
H	X					
K	X					
L	X					
M	X					
N	X					
P		X				different rendering intents (possible reason)
Q	X					
R	X					
S	X					
T	X					
V	X					
W	X					
X				X		Blending color space wrong (known Lim.)
Z				X		Blending color space wrong (known Lim.)
Label	X					
Page frame		X				
Σ:						

Altona Test Suite 2.0 – Technical 2

December 2011

Background

When version 1.0 of the Altona Test Suite was released, it addressed the all new PDF/X standard – most importantly how to find out whether output from a PDF/X file was correct or not. The Altona Test Suite turned into a powerful tool for many participants of the print production food chain:

- to find out whether a given product would live up to their expectations
- for users to ensure their entire workflow could and did handle PDF/X files correctly
- for associations to develop best practice guidelines

The first international PDF/X standard was released in 2001 (PDF/X-1a, followed by PDF/X-3 in 2002). PDF/X has since turned into the predominant file exchange format in the printing industry. A decade later we find ourselves in a situation, where substantial new features have been introduced into the PDF format. A new part of the PDF/X standard has been released in 2008: PDF/X-4, finally supporting transparency, OpenType fonts, optional content (aka layers), page sizes beyond 5 by 5 meters, JPEG2000 compression and more. Its companion standard, PDF/X-5, was developed in the same time, standardizing incomplete or partial file exchanges, targeting very specific needs in niche markets.

Some of the features defined in newer versions of the PDF syntax and standardized in PDF/X-4 have proven to be more demanding for implementers than others:

- **Transparency** introduced a completely new dimension of complexity – any combination of object type, color space, blend mode, opacity, nesting of transparency groups and so forth can be used, and actually is used by adventurous and demanding designers who in turn are driven by their insatiable customers longing for ever more enticing designs. Quite a number of developers found it less than trivial to cope with the challenges of such added complexity, both in terms of performance as well as correctness of the produced results.
- **Optional content** (often referred to as layers) has been around in the PDF format for a number of years but so far is only supported by a small number of PDF vendors. It allows the creation of PDFs for which a user can switch between views, for example between different language versions of a text document, or between different instances of a PDF for packaging, where each instance reflects a different flavor or color of a product package, while all instances share the common graphic elements. In addition, some page content can be turned on or off on an as needed basis – like measurement information, cut lines, identifying information or human readable instructions in a packaging file.
- The **OpenType** font format – an industry standard that has also been adopted as an ISO standard that uses the term Open Font Format (ISO/IEC 14496-22) – constitutes a consolidated font format building on both the PostScript and TrueType font formats, fully supporting Unicode as well as allowing intelligently for even the most complex scripts like Thai or classical Mongolian .
- **JPEG2000** (ISO 15444) is an extremely flexible compression format for continuous tone images. It is not to be mixed up with JPEG, as it uses a completely different architecture and algorithms. While JPEG is always lossy, JPEG2000 can be used in a lossless way, on average offering better compression than ZIP. In addition it supports a wide variety of color spaces and bit depths. Finally, it also provides options to decode images progressively – very convenient for very large and data intensive images.
- **Page size** in PDFs can now easily exceed the original implementation limit of roughly 5 by 5 meters – using a parameter called UserUnits current implementations support page sizes of up to 381 kilometers by 381 kilometers. While rarely in use so far, this very specific new feature is a nice approach to make handling of large page sizes that are not uncommon in very large format printing more elegant.

Quite a number of new features or extensions of existing features tend to have a less drastic impact but might easily be overlooked, like the support for version 4 ICC profiles or NChannel color.

Not only is the development work to support the features introduced between PDF/X-1a and PDF/X-3 on one side and PDF/X-4 on the other side far from trivial – it is also a challenge to do quality assurance and testing. Vendors tend to have their own set of carefully develped test files, but even they might not have thoroughly designed test files for each aspect of a PDF/X-4 implementation. Not to speak of their customers – how do they find out whether the PDF/X-4 support claimed by a vendor is actually as extensive and accurate as suggested by the vendor's product marketing? Last but not least there will be a number of scenarios where two experts may disagree about what the correct result should look like.

In addition to targeting features introduced in PDF/X-4, the Altona Test Suite 2.0 Technical Page also extends a number of tests that would have already applied to PDF/X-1a and PDF/X-3, for example more extensive coverage for smooth shades or PostScript and TrueType fonts than was available in Altona Test Suite 1.0 through 1.2.

The Technical Page of the Altona Test Suite 2.0 aims to address the needs of all parties involved in the production and processing of PDF/X-4 files. The test page contains a sequence of patches where each patch allows a relatively easy and straightforward assessment of the quality of output generated from a PDF/X-4 file. All patches have been developed by members of the European Color Initiative – all of them experts familiar with PDF standards development, and with a solid background in software development and print production – over a period of about three years. Vendors have had the option for about eighteen months to access a beta version of the test page before it was released to the public.

Who should use the Altona Test Suite 2.0 Technical Page, and how?

Vendors: It is expected that any vendor developing tools or solutions that process or output PDF/X-4 files in any way will find the Altona Test Suite 2.0 Technical Page useful for:

- quality assurance for the rendering results of a RIPs, printer, proofing system or output on screen
- quality assurance for products that process PDF/X-4, for example color converters, ink optimizers, or tools that assemble complete or partial pages
- quality assurance for PDF consuming tools, like for example PDF import in authoring applications

Output service providers: any organization offering printing services in the broadest sense, that does accept PDF/X-4 files will find the test page useful to determine to which degree their workflow processes those PDF/X-4 files correctly, and to identify processing steps that provide less than perfect handling of PDF/X-4 files. Very often all tools involved in the processing of PDF/X-4 files fully support PDF/X-4, but are either not configured correctly, or are combined into a processing sequence that is not PDF/X-4 compliant. Thus output service providers can use the Altona Test Suite 2.0 Technical Page to

- test the complete workflow processing chain
- test individual components for suitability
- check the configuration for a component
- track conformance of a setup on an ongoing basis through regularly repeated control output, for example in the form of a weekly process validation

Producers of print ready PDF files: it may sound as if only organizations or persons processing PDF/X-4 file have a need for a decent PDF/X-4 test page. Rather to the contrary: a producer of print ready PDF/X-4 files will need a means to judge the appearance to be

expected from a compliant output system. In order to achieve this, already the authoring application must support PDF/X-4 correctly, or at least a verification tool – like a general purpose PDF viewer – that correctly simulates the expected output based on the PDF/X-4 file that is going to be sent to a printer.

Repurposing: While the PDF format is often considered as a final form format, where the only remaining processing steps are display or creation of printed output, reality in the communication industry at large teaches us that a substantial portion of PDF files originally intended just for a single print run tends to be reused and repurposed in numerous ways. Whether a file for gravure printing is to be sent to a digital printing device, or to be used for online advertising or on a tablet PC – no thinkable option exists that is not actually used at some point by quite a number of users.

Where appearance of content or brand colors is important to those initiating the communication, it usually becomes an absolute requirement to get the rendered output right regardless of what the ultimate communication channel is. Though a smartphone will typically not render a non-trivial PDF/X-4 file correctly, precautions need to be taken that whatever is sent to such a smartphone is reflecting the originally created appearance as correctly as possible.

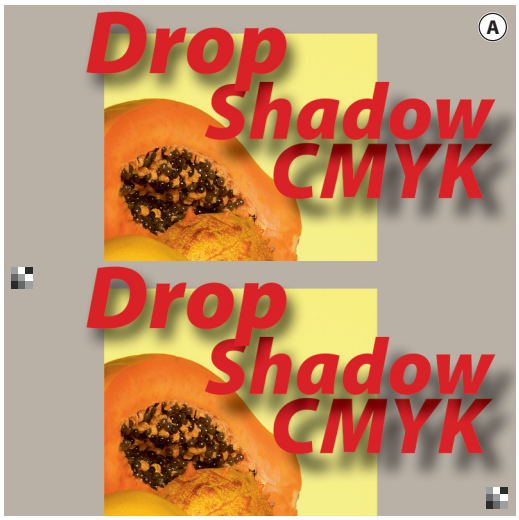
Overall structure

The Technical page consists of an overall structure as listed below:

- 15 patches that reflect real world use of PDF/X-4 features, each identified by a single uppercase character (A through R)
- 6 blocks to systematically test the main transparency parameters, identified by a single uppercase character (S through Z):
  - blend modes
  - blend color spaces
  - source color spaces
- page label with color conversion indicators
- page frame with different variants producing a uniform gray visual appearance.

In addition to being contained in the test page, the patches A through R, a block consisting of patches S through Z, as well as the label are also provided individually, with each of these contained on its own in a separate page.

Patch A



Patch design, practical relevance

Patch A reflects a very common use of transparency: drop shadows. In addition it addresses the potential impact of the layout and design applications' "Transparency Flattener Presets" (resolution settings in particular) on the output quality. These settings often define the image resolution of the so called soft masks which are created during the PDF creation to achieve the transparent effect (here: drop shadow).

Regardless of the transparency handling by the output device (transparency flattening with higher resolution or transparency rendering on e.g. the Adobe PDF Print Engine) the soft mask's resolution may limit the quality under certain circumstances. For that reason it is important, that the creator of the PDF chooses the "Flattener Presets" properly.

Technical parameters

All object colors are defined as process colors. Three lines of text are placed on top of an image and a gray background. The drop shadow (100% black ink, using blend mode "Multiply", with an opacity of 75%) is divided into two sections to demonstrate the potential impact of the transparency processing settings.

The part on the left hand side has a significantly higher resolution than the the part on the right hand side (200 ppi). The two drop shadow sections touch each other in the middle of the patch (under the letters "p" and "h").

Example for incorrect output

Patch B

Patch design, practical relevance

Patch B addresses possible issues from the interaction of color conversions with processing transparency or with transparency flattening, in particular the conversion of spot colors (here: orange colored text objects) and RGB images to process colors. Digital printing processes often require the conversion of spot colors to process colors.

Technical parameters

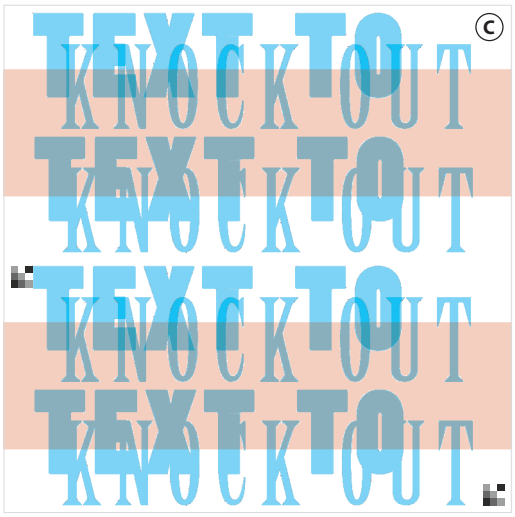
The text color is a spot color named "Orange". The drop shadow is defined in the same way as in patch A (100% black ink, using blend mode "Multiply", with an opacity of 75%). The background image is divided into two sections: the lower left section is an RGB image (eciRGB v2) while the part in the upper right is a DeviceCMYK image. The color properties assigned to the RGB image have been used to create the CMYK image in Photoshop: rendering intent "perceptual", target profile "ISO Coated v2 300% (ECI)".

Examples for incorrect output

If the process color conversion of the spot color takes place after transparency flattening, white boxes may occur underneath the drop shadow (upper example).

A distinct color difference between the upper right and lower left section of the background image would indicate color management deficiencies (lower example).

Patch C



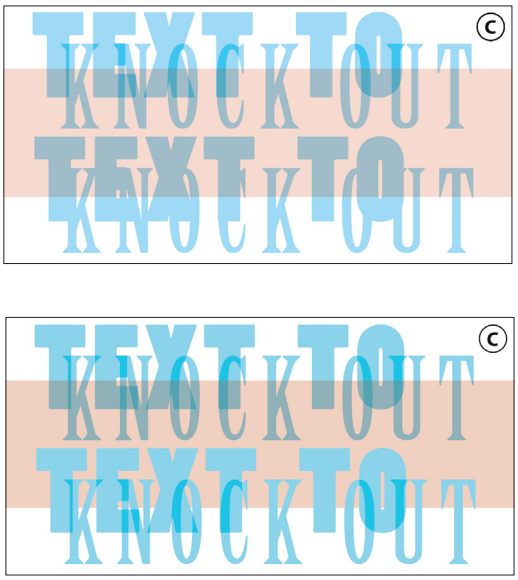
Patch design, practical relevance

This patch focuses on a property determining the interaction of objects within a transparency group: The group may be "knockout" or "non-knockout", which determines whether each of the objects within its stack are composited with one another and then into the group's backdrop, or only with the group's backdrop.

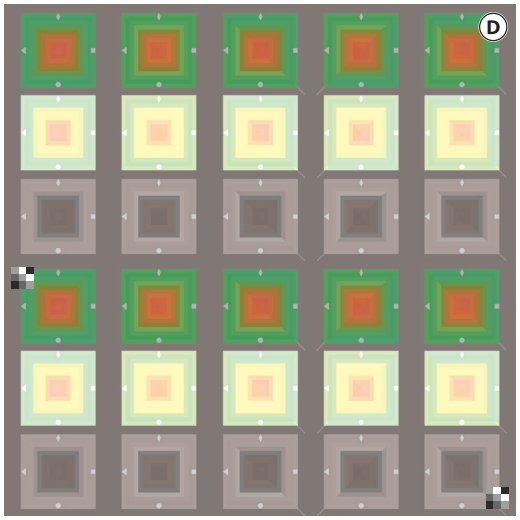
Technical parameters

The upper two lines of text belong to a transparency group set to non-knockout. As shown in the reference image (lower half of the patch), the transparency effect (blend mode "Multiply") results in a darker color in the intersecting parts of the two text lines. The lower two lines are set to knockout and therefore shall be rendered without a color difference in the intersecting parts. Both groups interact with the backdrop (rectangle filled with a light red process color).

Example for incorrect output



Patch D



Patch design, practical relevance

Patch D is designed to systematically assess the color management capabilities in conjunction with transparency handling. Proper color rendering requires that all properties such as source color profile, rendering intent and target color profile (here the destination profile in the OutputIntent) are honored. The respective test objects are taken from the Visual page of the Altona Test Suite v1.1.

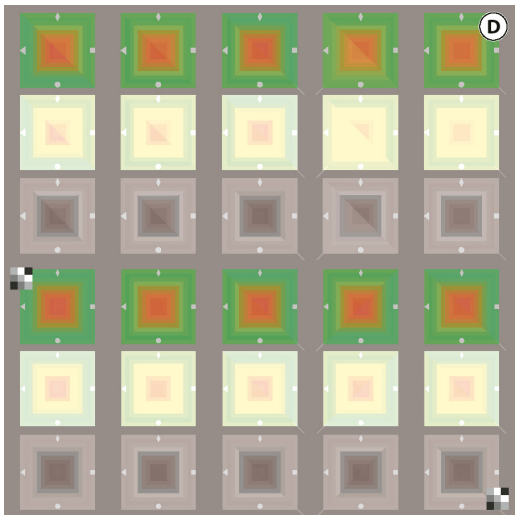
Technical parameters

The patch contains three lines of objects, each line consisting of the five color management patches from the Visual page of the Altona Test Suite v1.1. The upper row's transparency setting is blend mode "Normal" with an opacity of 70%. The middle row is set to the blend mode "Color Dodge" with an opacity of 100%, while the bottom row's blend mode is "Luminance" with an opacity of 100%. All objects share the same gray background (DeviceCMYK 0.32/0.32/0.32/0.32).

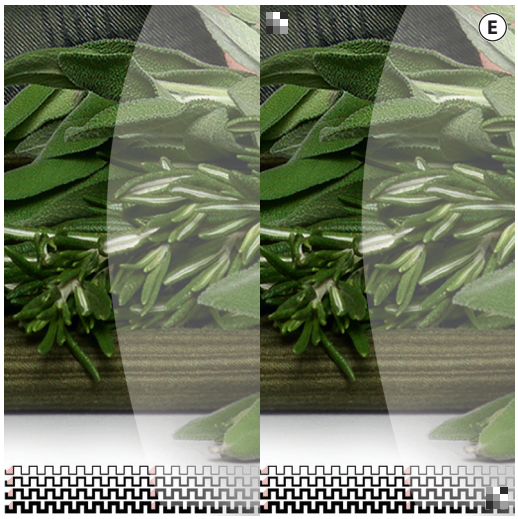
Expected output

Note: For a quick check the following hint may help: the first two color management squares per row (from left to right) shall be colored identically in all four sections. The sections are indicated by small icons. The three squares at the right hand side shall show a distinct color difference along the edges indicated by small lines underneath the squares.

Examples for incorrect output



Patch E



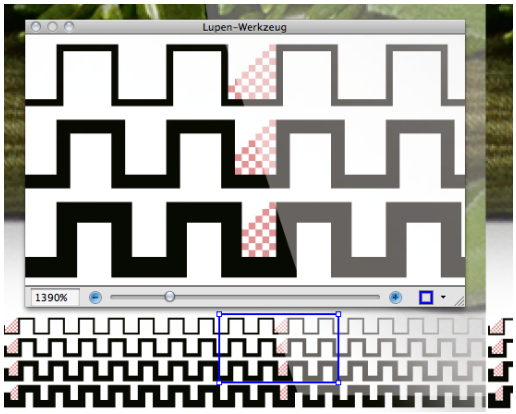
Patch design, practical relevance

This patch is designed to detect possible issues with fine image details in conjunction with transparency processing or flattening.

Technical parameters

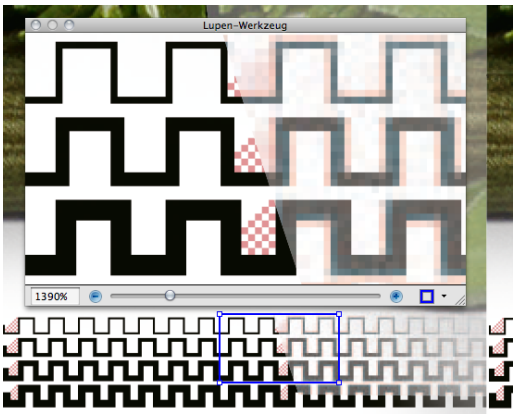
A curve shaped, white (DeviceCMYK 0/0/0/0) object is placed on top of an image, using the blend mode "Normal" with an opacity of 42%. In the lower part of the image crisp black lines and red checker boards have been added. Each of the black lines is either one, two, three or four pixels wide. Each small square of the red checker board is sized at exactly one pixel.

Expected output



Enlarged view of fine image details

Examples for incorrect output



Inadequate image compression settings applied during transparency flattening may cause disturbing artifacts, e.g. ragged lines along the black strokes.

Patch F

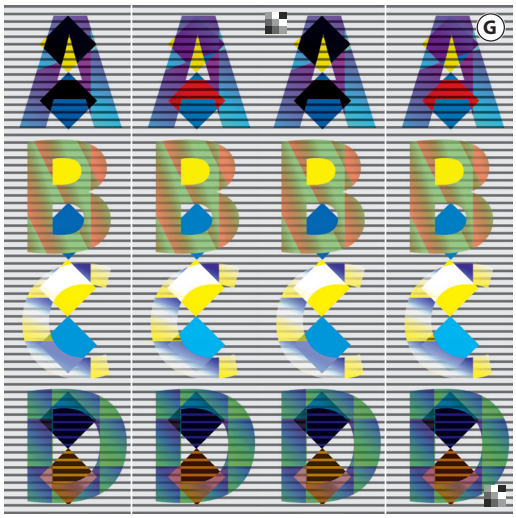
Patch design, practical relevance

Proper font handling is a challenge whenever text is affected by transparency flattening or trapping. PDF/X-4 extends the number of valid font types allowed in previous PDF/X version by adding support for Type 1 and TrueType flavored OpenType fonts. In addition issues may occur with text converted to outlines.

Technical parameters

The fonts used in this patch comprise all relevant font types such as PostScript Type 1, TrueType and OpenType, the latter in both TrueType and PostScript flavored versions. The fonts chosen for the patch cover different scripts such as Latin, Asian or Hebrew and Arabic typefaces. Besides that, the characters were selected to address issues that may occur with different font encodings and certain characters such as ligatures, as well as diacritic marks like the French *accent aigu*. The glyphs of the two words at the bottom of the patch have been converted to outlines in Adobe Photoshop ("Muchas") and Adobe InDesign ("Gracias").

Patch G

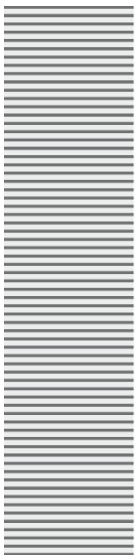


Patch design, practical relevance

The patch combines smooth shade effects with transparency effects, using combinations of device independent and device dependent color definitions and blending spaces. Designers will in some cases play with both kinds of effects until they arrive at an appearance that is to their liking. As can be seen already from comparing the same characters in the left vertical character sequence and the right vertical character sequence, small but sometimes significant differences in appearance occur depending on exactly how the effects are used.

Technical parameters

Background: very wide dashed line alternating between 10%K and 70%K.

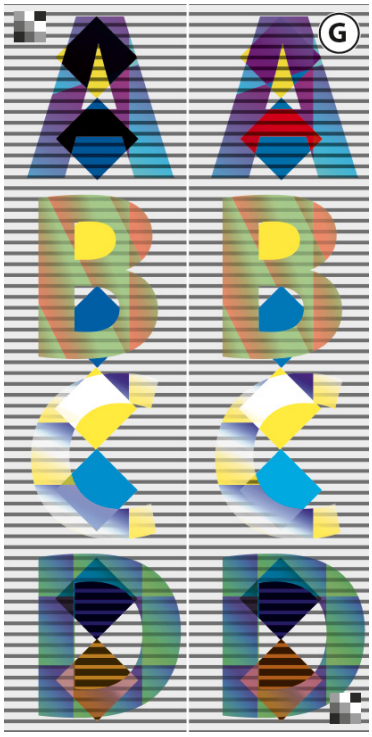


Foreground: Characters A, B, C and D, in Myriad Bold 48pt, filled with axial smooth shading in the form of Type 1 patterns:

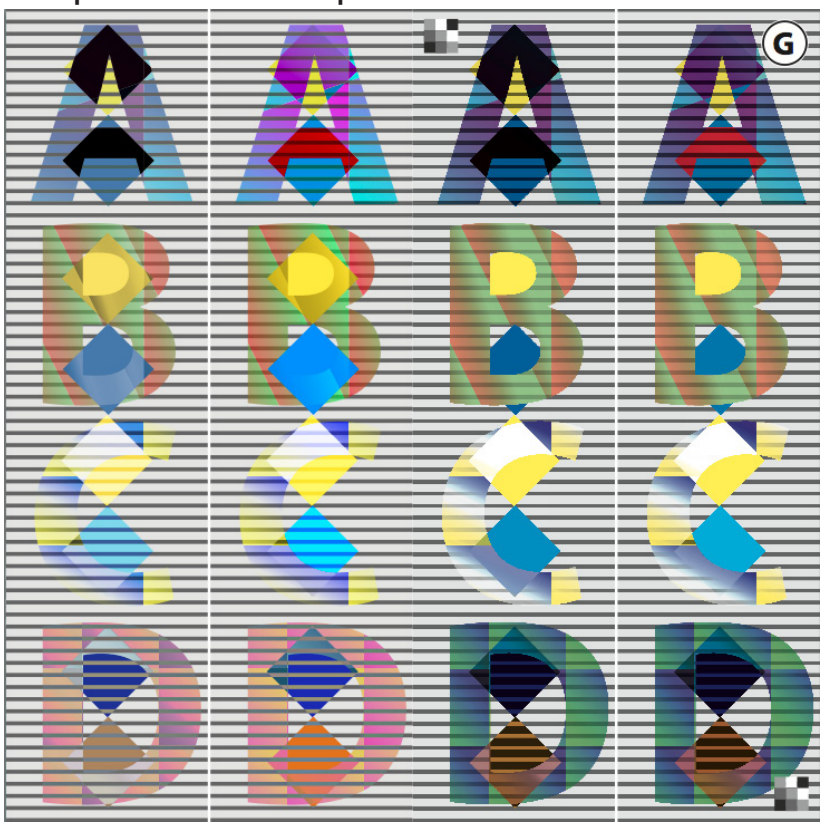
- Each smooth shade is built using Wide Gamut RGB ICC based color space
- each character sits on top two yellow and blue squares, these squares are colored using Lab
- the characters on top of these squares use transparency
  - A: Difference 100%
  - B: Luminosity 60%
  - C: Colour 100%
  - D: Exclusion 75%
- each character is grouped together with the yellow and blue square underneath it, where some groups use an isolated group, whereas others don't, and for some the group is isolated and for others it is non-isolated.

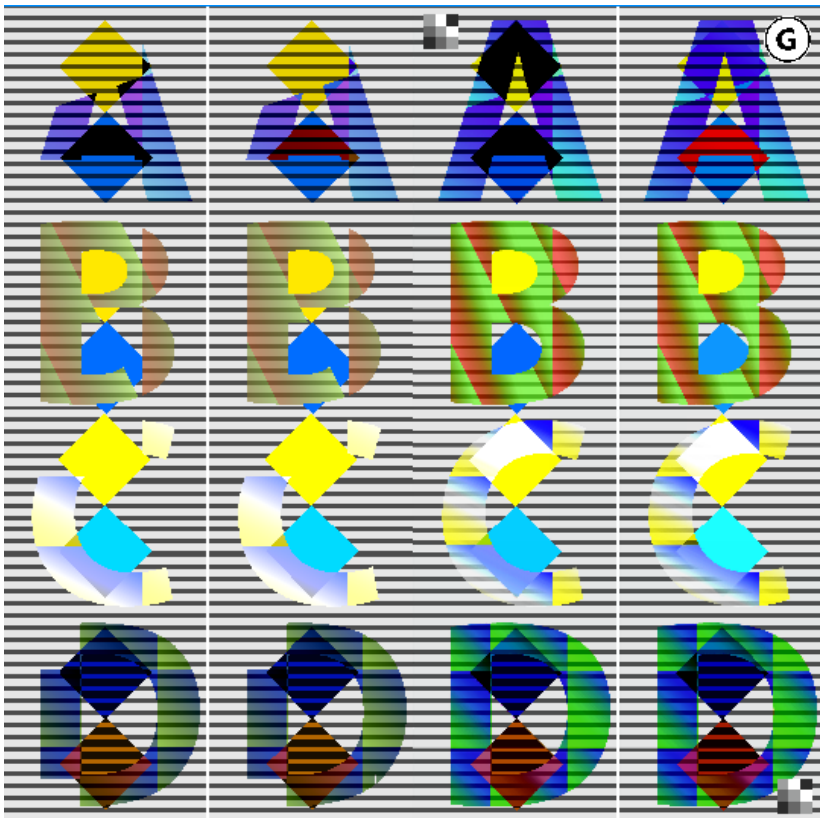
- the vertical A-B-C-D sequence on the left uses DeviceCMYK as blending space for the transparency group, whereas the vertical A-B-C-D sequence on the right uses the Wide Gamut RGB ICC based color space as its blending space.

Expected output

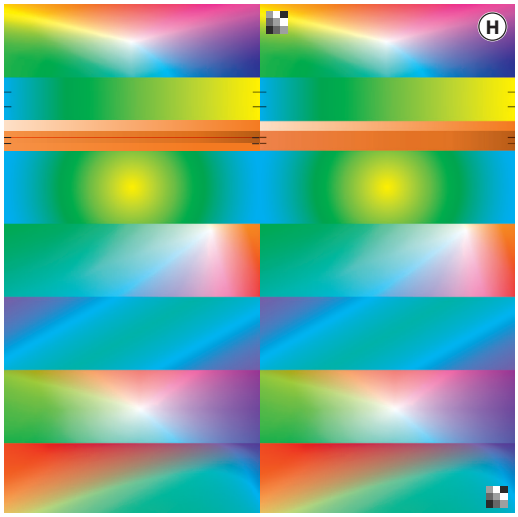


Examples for incorrect output





Patch H



Patch design, practical relevance

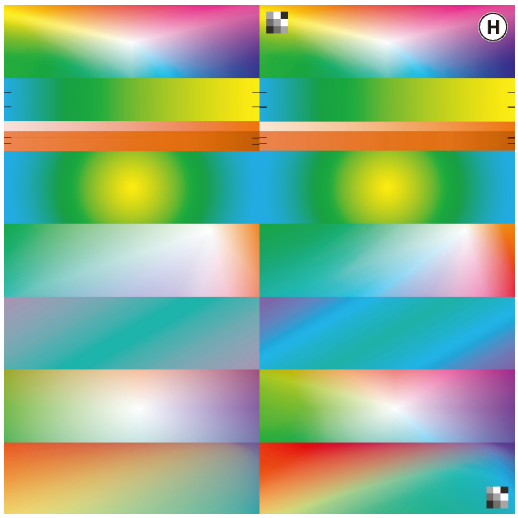
The patch is intended to test proper rendering of all types of gradients, including the seven smooth shade types defined in the PDF language, conventional "PostScript 1 style" gradient built by vector elements as well as pixel based gradients. Common smooth shade types like axial and radial shades usually are rendered properly, while more complex types such as the so called "Illustrator gradient mesh" are sometimes not processed correctly. Another issue are gradients containing a spot color where overprinting and transparency properties may not be handled properly.

Technical parameters

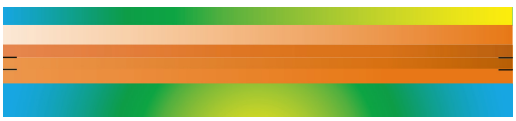
The smooth shades are arranged from top to bottom in this sequence: function-based shading (type 1), axial shading (type 2), radial shading (type 3), free-form Gouraud-shaded triangle mesh (type 4), lattice-form Gouraud-shaded triangle mesh (type 5), Coons patch mesh (type 6) and tensor-product patch mesh (type 7), better known as "Illustrator gradient mesh".

The black lines mark variants of gradients representing different approaches in the design stage. The cyan to yellow gradient contains (from top to bottom): a smooth shade (axial shading), a conventional vector gradient and a pixel gradient. The four spot color gradients comprise axial shadings using the spot color Orange in different ways (from top to bottom): spot color Orange only, DeviceN (black and Orange), and two axial shadings created by two pieces placed on top of each other: Orange on black (transparency, blend mode "Multiply" with an opacity of 1.0) and at the bottom Orange on black (set to overprint).

Examples for incorrect output

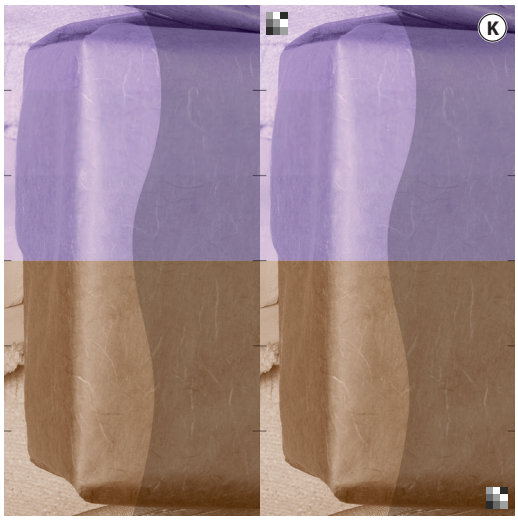


The four shades in the lower section show significantly less saturated colors compared to the reference image.



In this case the lowermost spot color gradient is too light as the black gradient underneath the orange gradient is ignored (overprinting not honored).

Patch K



Patch design, practical relevance

The patch shows several variants that go back to colorizing a grayscale image in an authoring application. Designers often use different ways to achieve the same visual appearance, in this case a violet tint in the upper part and a brownish tint for printing with black and a spot color Orange in the lower part. The PDF properties of the different sections depend on the file formats of the source images and how the grayscale images have been colorized.

Technical parameters

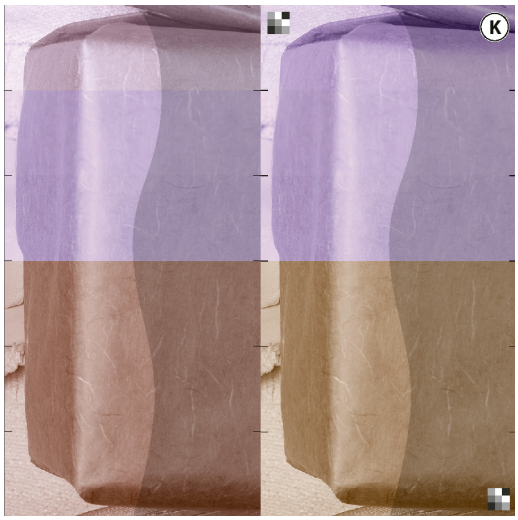
Indicated by black lines, the six sections reflect the following ways of colorizing:

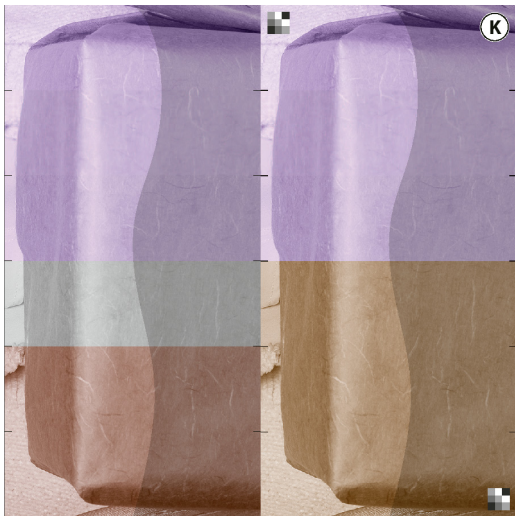
- the topmost segment is an Lab image created in Photoshop
- the 2nd segment consists of three grayscale images placed on top of each other, each colored differently, and the top ones set to overprint in InDesign.
- the 3rd segment has been created as a duotone image in Photoshop.
- the 4th segment was saved as a Photoshop PDF (resulting in two DeviceN images),
- the 5th segment was saved as TIFF (one indexed DeviceN image)
- the 6th segment was saved as 16bit Photoshop (PSD) file (one DeviceN image).

The three brownish image segments have been created in Photoshop based on the same file as for the first three segments to reflect the impact of the source file type on the PDF property. The Photoshop source image contains two channels colorized with process black ink (top) and Orange ink (bottom):

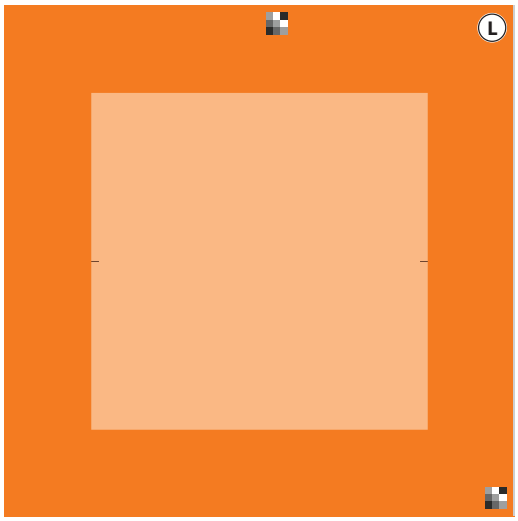
Each of the six segments was created as a separate file and placed on an InDesign page which was then exported as a PDF file.

Examples for incorrect output





Patch L



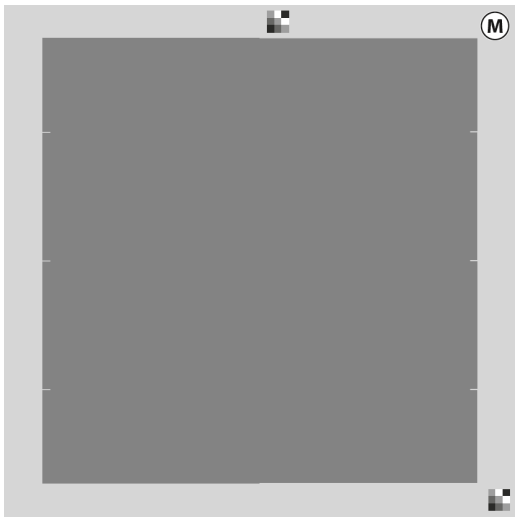
Patch design, practical relevance

This patch deals with the use of transparency and white page objects to lighten a given background.

Technical parameters

The background color is a 100% tone value of the spot color "Orange", partially covered by two "white" squares (0% of the respective color). The color space of the upper rectangle is a separation color space named "PANTONE Warm Gray 6C" using a tint value of 0.0. The lower rectangle is colorized with DeviceCMYK 0/0/0/0. Both squares use blend mode "Normal" with an opacity of 50%.

Patch M



Patch design, practical relevance

This patch consists of four gray images with different color properties, placed on top of a gray background. The practical relevance of the different color properties are idiosyncrasies of common design and workflow applications used along the workflow from design to output. These applications sometimes tend to alter the color specification of a given grayscale image.

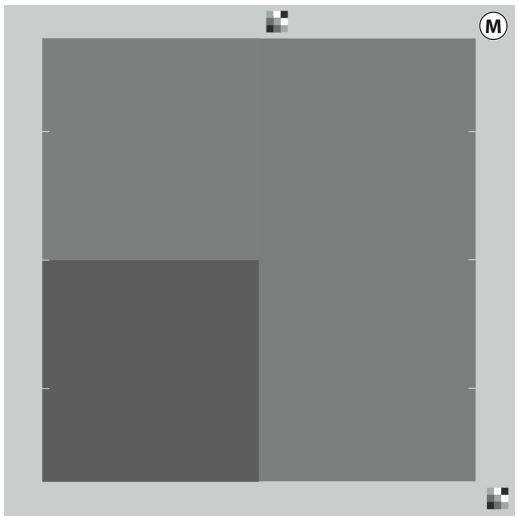
Technical parameters

The background is a vector object, colorized with 22% of process black (DeviceCMYK 0/0/0/22%). The four rectangles in the foreground are images containing 100% black pixels only using the following color spaces (from top to bottom):

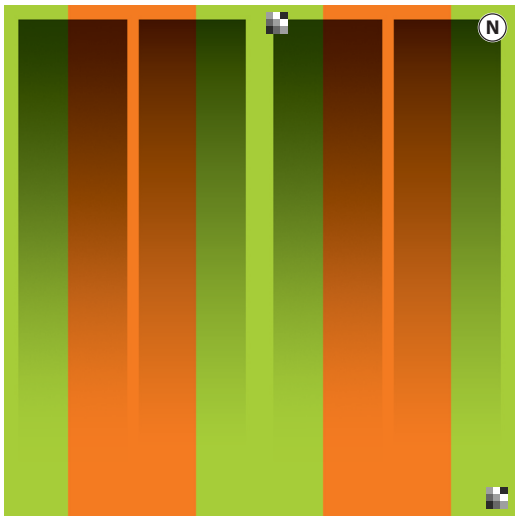
- IndexedDeviceCMYK
- IndexedDeviceN
- Separation Black, and
- DeviceGray.

The image's blend mode is "Normal", with an opacity of 50%.

Examples for incorrect output



Patch N



Patch design, practical relevance

This patch demonstrates two popular methods used by designers to darken a background with a gradient.

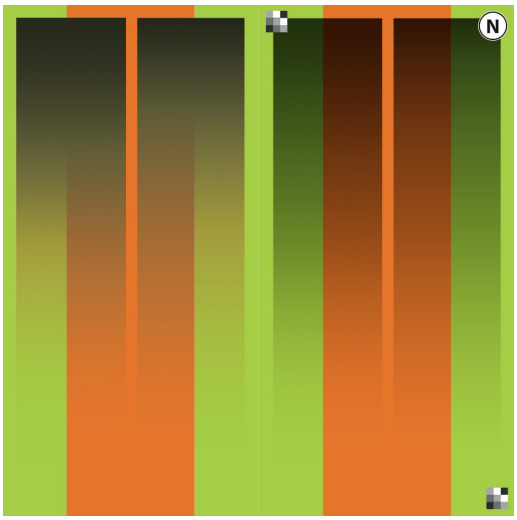
Technical parameters

The background is divided into the two sections “green” (DeviceCMYK 40/0/100/0) and “orange” (spot color “Orange”). The blend mode used for the two gradients is “Darken”. The gradient on the left hand side is a grayscale image, and the gradient on the right hand side is a smooth shading colorized with process color black (coded as a single channel DeviceN color space).

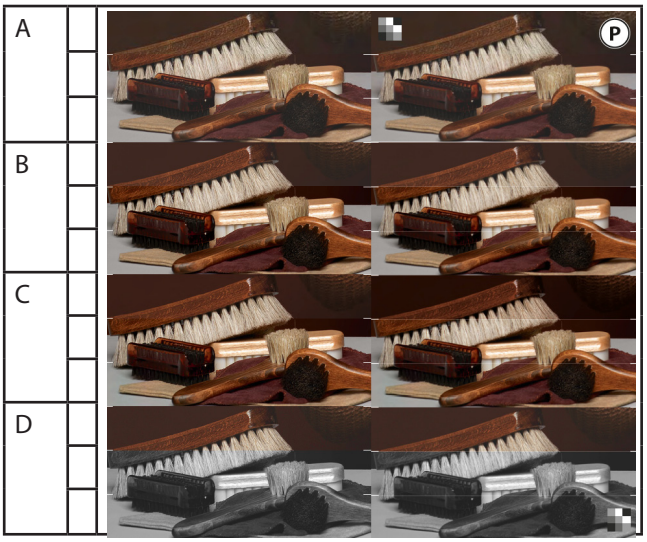
Expected output

Proper rendering puts the gradients into the process color black channel only and does not affect the color values of the green and orange background areas.

Examples for incorrect output



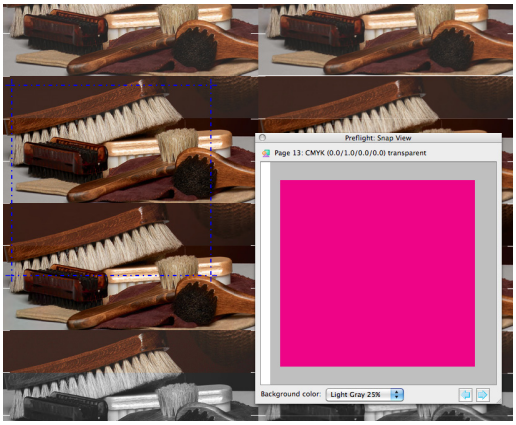
Patch P



Patch design, practical relevance

JPEG2000 image compression as well as a bit depth of 16 bit have been part of the PDF syntax for a number of years already but are only slowly being adopted in day to day production. While using just JPEG2000 or just a bit depth of 16 bit usually works well in up to date technology, problems might occur when these are used in combination, or together with not so common color spaces, as well as v2 versus v4 ICC profiles. This patch uses the same base image in a number of bit depths, color spaces, and use of JPEG2000.

In order to trigger processing for the patch with transparency involved, a Magenta colored rectangle has been positioned on top of the image slices, with an alpha value of 0.0, making it invisible.

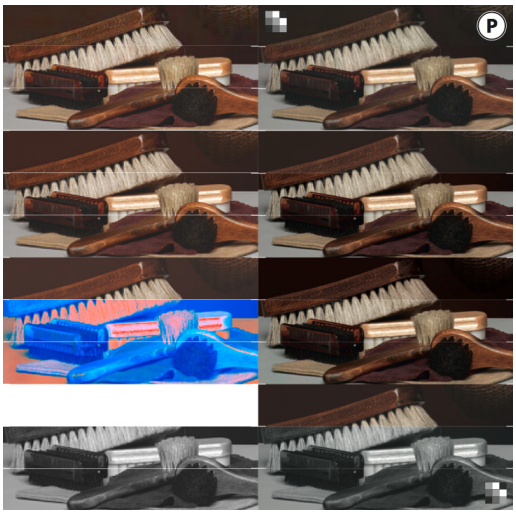


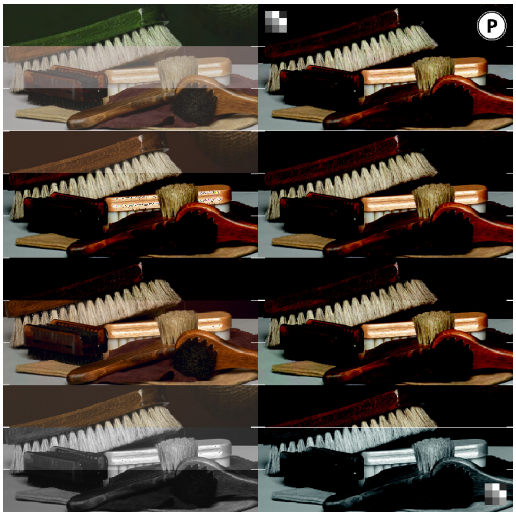
Technical parameters

The base image shows up four times on the patch (referred to as A, B, C and D from top to bottom), and each instance is subdivided into three portions (referred to by 1, 2 and 3 from top to bottom):

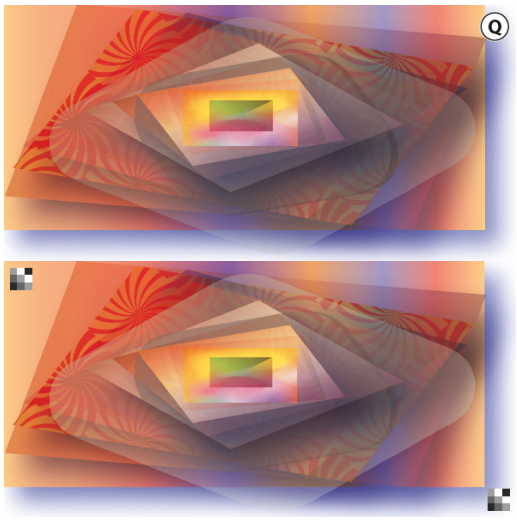
- A1: 8bit RGB profile (i.e. an RGB profile where the Red and Green channels have been swapped) using JPEG2000 with lossy compression (max. quality)
- A2: 8bit RGB as a v4 ICC profile, using JPEG2000 with lossy compression
- A3: 16bit RGB as a v2 ICC profile, using JPEG2000 with lossy compression
- B1: 16bit RGB as v2 profile, using lossless JPEG2000 compression
- B2: 16bit CMYK as a v2 ICC profile, using JPEG2000 with lossy compression
- B3: 16bit CMYK as a v2 ICC profile, using lossless JPEG2000 compression
- C1: 8bit CMYK as a v4 ICC profile, using JPEG2000 with lossy compression
- C2: 16bit using Lab color space, using JPEG2000 with lossy compression
- C3: 8bit using Lab color space, using JPEG2000 with lossy compression
- D1: 8bit Indexed RGB color based on a v2 ICC profile, using JPEG2000 with lossy compression
- D2: 16bit gray using a 30% dotgain grayscale v2 ICC profile, using JPEG2000 with lossy compression
- D3: 8bit gray using a 10% dotgain grayscale v2 ICC profile, using lossless JPEG2000 compression

Examples for incorrect output





Patch Q



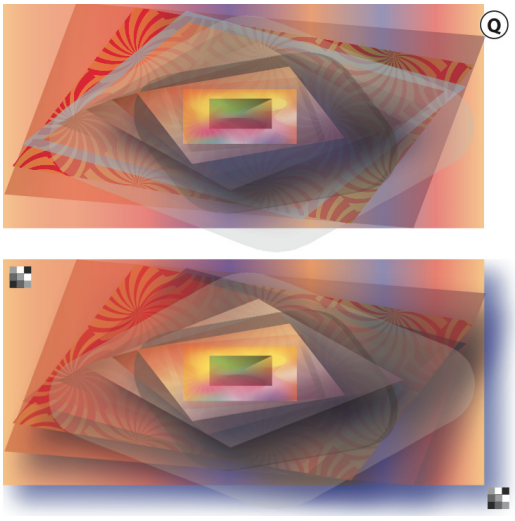
Patch design, practical relevance

This patch makes use of patterns, smooth shades, drop shadows, transparency effects and nested grouping of objects such that in some instances isolated transparency groups were created and in other instances not. The patch does not follow any specific logic, but rather aims at being an 'unplanned' arrangement of graphic objects using all kinds of graphic effects available in today's software products.

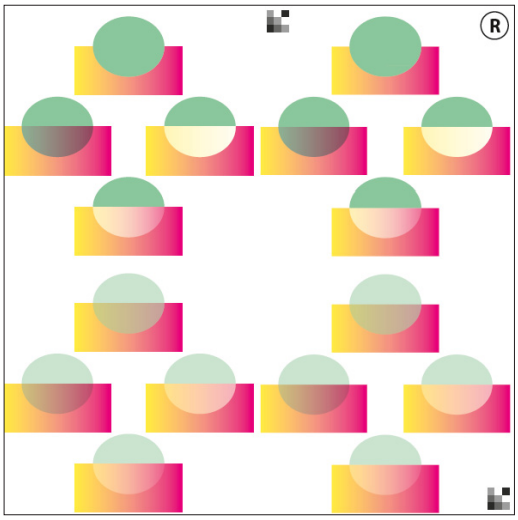
Technical parameters

Arbitrary, partially grouped arrangement of vector objects making use of various effects.

Example for incorrect output



Patch R



Patch design, practical relevance

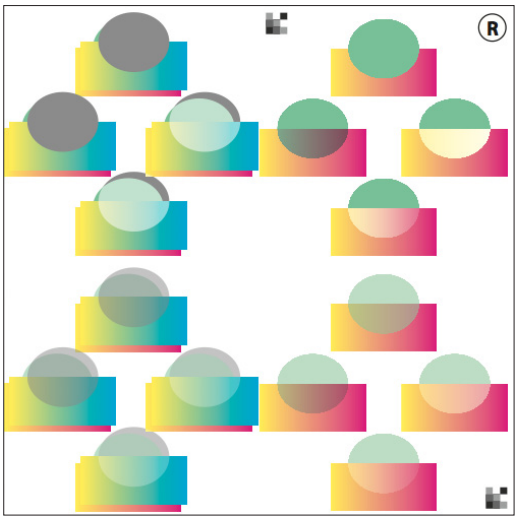
This patch explores effects that result from combining transparency with overprint. In addition it makes use of optional content.

Technical parameters

The patch is built according to the information given in ISO 32000-1 – Document management – Portable document format – Part 1: PDF 1.7, page 744, Figure L.20 – Blending and overprinting (11.7.4.3, "Compatibility with Opaque Overprinting").

In addition, the contents of the patch is repeated as optional content set to be non-visible, where this optional content uses different coloring, and is offset slightly to the right and to the top.

Example for incorrect output



Block with patches S-Z – Structure

Patch design, practical relevance

The main goal of patches S through Z is to completely iterate through certain combinations of graphic state parameters that affect the appearance of transparent objects directly or indirectly. The graphic state parameters that are being permutated are:

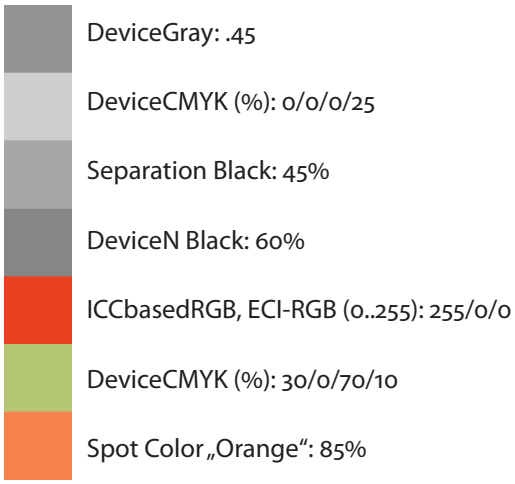
- type of object: on the lowest level, the patch uses miniature elements consisting of a filled object, a stroked object (line) and a text (a minus sign); all three are arranged to form a rectangular shape; as all are colored using the same color space, all three should have identical appearance
- color space: each object triple appears using several color spaces:
- variations of black or gray, using DeviceGray (using a value of 0.45, equivalent to 55% black), DeviceCMYK (CMY channels are zero, K has a value of 25%), Separation Black (with a value of 45%) and a single channel DeviceN with Black as its only colorant (with a value of 60%).
- ICC based RGB (1.0/0.0/0.0 RGB tagged with ECI-RGB.V1.0 profile)
- DeviceCMYK (light pale green, 30%/70%/0%/10%)
- spot color by means of a Separation color space named "Orange"
- 16 different blend modes for the transparent 'green' object (using a DeviceCMYK fill of 85%/15%/80%/25%) on top of the background with the object triples in the various color spaces described above.
- 3 different blend color spaces – DeviceCMYK, ICC based RGB and ICC based Gray
- the 3 different blend color spaces are used by a non-isolated transparency group (on the left) and by an isolated transparency group (on the right).

The inclusion of all the combinations implied by this setup makes it possible to systematically check that should any of these occur in real world workflows they are handled correctly. It is obvious that it is more likely for some than for others to happen in real world documents, but on the background that modern authoring tools enable users to actually create documents that use arbitrary combinations of object types, color spaces and other graphic state parameters it is to be expected that any of the combinations found in patches S through Z will one day be encountered in a real world workflow.

In order to make it easy to identify incorrect rendering in a complex patch like this one, it was decided to mirror each patch in the form that a correct appearance is shown directly to the right of the actual patch segment. With the areas of the various objects touching each other, it is relatively easy to spot deviations between neighboring objects where they should actually look the same.

Each segment of a column's background is a combination of three basic objects:

Each segment of a column's background uses a different color space:



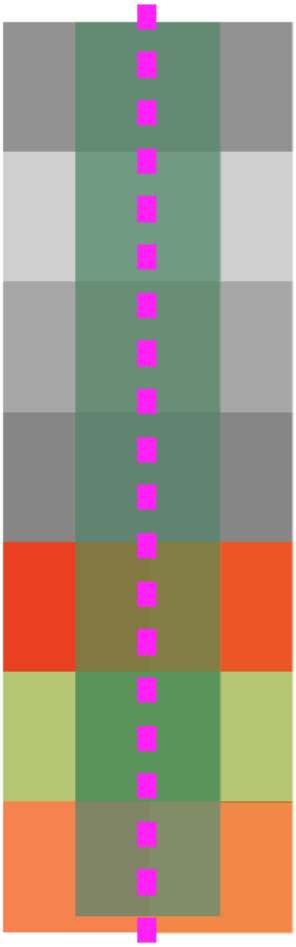
A vertical green bar set to using transparency sits on top of the background in a column

The vertical green bar uses DeviceCMYK (Cyan: 0.85, Magenta: 0.15, Yellow: 0.8, Black: 0.25) and a **ca** value of 0.6 (opacity). In each of the column's occurrences a different blend mode is used for the vertical green bar (see below).

Each of the columns is mirrored to the right, such that the mirrored part consisting of an image reflects the expected appearance

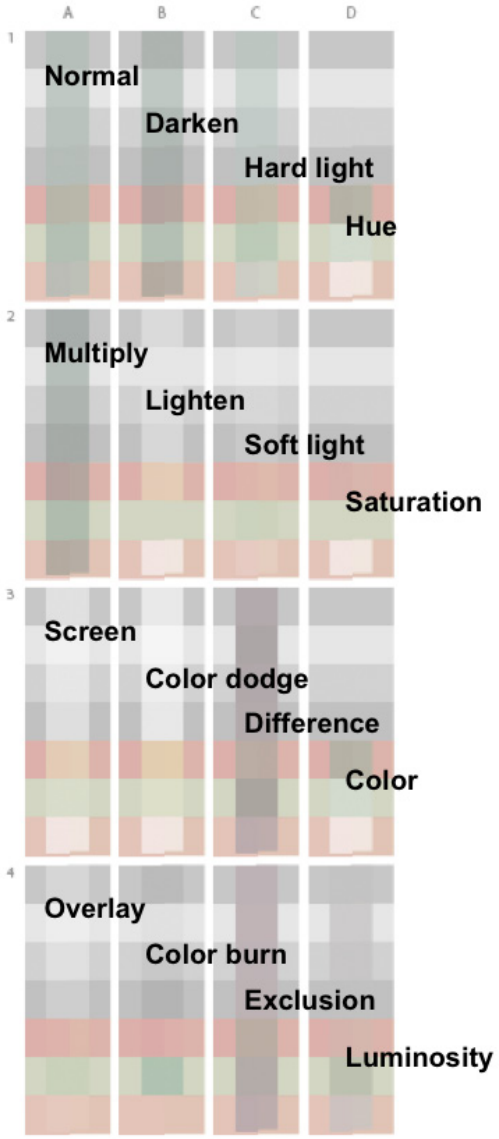
The left part consists of "live objects", whereas the right part is an image reflecting the expected correct appearance for the part on the left.

Note: Due to the nature of this setup, it may easily occur that the left part and the right part of a column do not match exactly. For the purposes of determining correct transparency rendering, common sense should be used to decide whether left and right part are considered to match or not. Small visual differences between left and right part can usually be ignored.



A 4 by 4 matrix of columns iterates over the 16 blend modes available in PDF

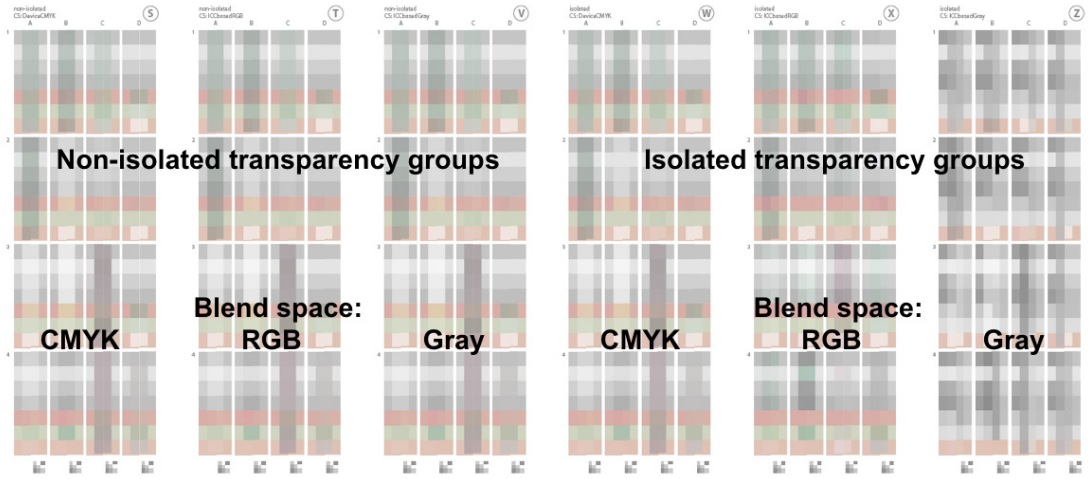
The columns are organized in a 4 by 4 matrix, where each column uses a different blend mode for the vertical green bar. All other objects of a column are identical across all column occurrences.



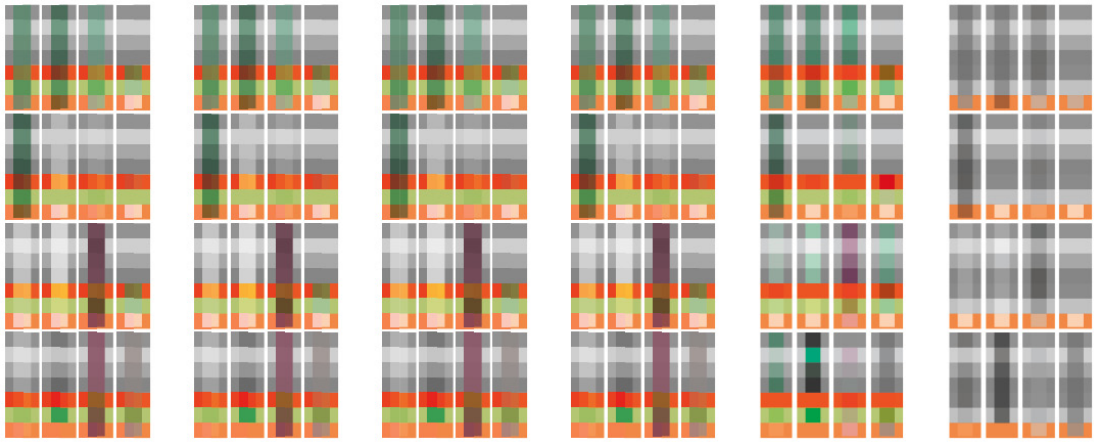
Six blocks iterate over CMYK, RGB and gray blend color space, for both non-isolated and for isolated transparency groups

Each 4 by 4 matrix as a whole uses different combinations of isolated versus non-isolated transparency group and blending color space, from left to right:

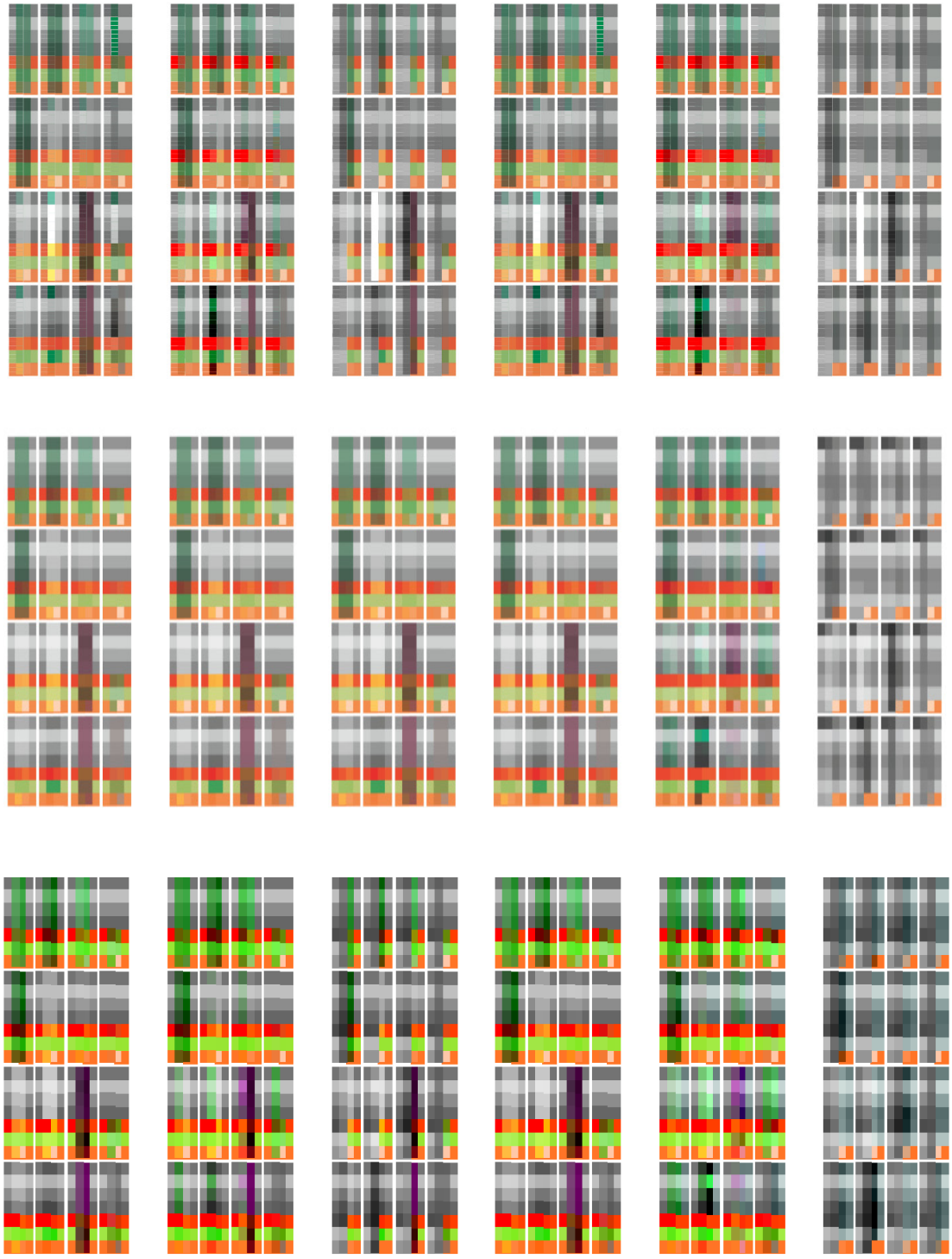
- non-isolated, DeviceCMYK (patch S)
- non-isolated, ICC based RGB (patch T)
- non-isolated, ICC based gray (patch V)
- isolated, DeviceCMYK (patch W)
- isolated, ICC based RGB (patch X)
- isolated, ICC based gray (patch Z)



Patches S - Z



Examples for incorrect output



Page label

Patch design, practical relevance

The page label (on the right hand side of patch "R") contains two small gray squares in its lower right corner. The purpose of these objects is to indicate issues that may arise after color conversions or the exchange of the OutputIntent profile (OI).

The practical relevance of the upper square ("Color Conversion Indicator"; lighter gray) are e.g. ink saving color conversions prior to transparency rendering or transparency flattening. Depending on the blend mode, color conversions of objects with live transparency may significantly affect the visual appearance of these objects and their background.

The bottom square ("OI Swap Indicator", darker gray) addresses the importance of the OutputIntent profile for the visual transparency effect of objects colored with output independent colors, e.g. RGB images.

Technical parameters

The "Color Conversion Indicator" consists of three page objects. The background is a vector element using DeviceCMYK with the color values 80/57/67/10. The background object is fully covered by an object using DeviceCMYK with the color values 0/0/0/66, and the transparency blend mode "Hard Light" with an opacity of 100%. The top most element is "X" shaped using DeviceCMYK with the color values 0/0/0/39 – the expected appearance for the area outside the "X" shape.

The design of the "OI Swap Indicator" is almost identical, except for two differences: The background object is an image object using the Lab color space with the color values 60/0/0 and an assigned perceptual rendering intent. The transparency blend mode is "Color Burn".

Expected output

Proper rendering results in two uniformly gray colored squares. The upper square, colorized with 39% of process black ink and the lower one colorized with the CMYK values 38/30/30/53.

Examples for incorrect output

Gray page frame

Patch design, practical relevance

Along the edges of the page, a 5 mm frame creates a gray page border. The border reflects different technical ways to achieve a certain gray appearance. The technical properties are indicated on the sections of the page frame. Depending on the capabilities and settings of the processing software, the sections may be rendered incorrectly, creating distinct visual differences.

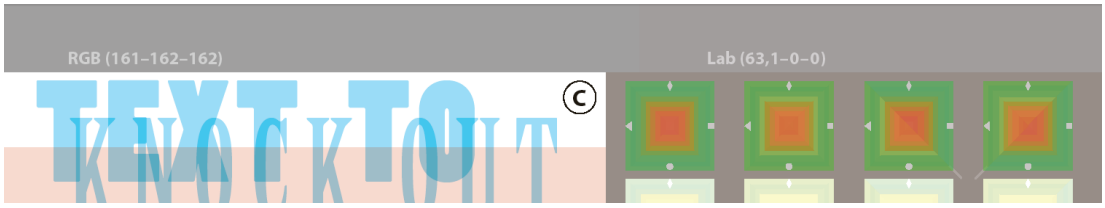
Technical parameters

From the upper left corner to the upper right corner:

- the first section consists of a black vector object (DeviceCMYK 0/0/0/100) placed with 50% opacity in "Normal" blend mode on top of a white background (DeviceCMYK 0/0/0/0). Correct transparency rendering creates 50% process ink black.
- the second section is a vector object using DeviceCMYK with the values 0/0/0/50. This section is the reference for correct processing of the first section.
- the third section is a vector object using an ICC based RGB color space ("eciRGB v2"). When properly converted to the destination profile in the OutputIntent, the RGB color values 161/162/162 and the rendering intent "Relative Colorimetric" result in a gray tone visually matching the visual appearance of 50% process black ink.
- the fourth section is colorized in the color space "Lab" with the values 63.1/0/0. Properly converted to the output intent profile, honoring the rendering intent "Relative Colorimetric", this section again matches the visual appearance of 50% process black ink.

Expected output

Correct rendering results in a uniformly colored gray border.
Note: Slight visual differences between the black-ink only and the RGB and Lab sections do not indicate problems as they may be caused by rounding effects or color deviation of the printing.



Examples for incorrect output

In this case, a color conversion using a device link profile with color purity properties (leaving solid ink tones untouched) caused the distinct visual difference between the first two sections:

Evaluating the test page

As it is not trivial to provide a "correct result" of this test page that will look correct under all circumstances, and can be achieved in exactly the same way by an output device, evaluating output of the test page most often will not be a yes or no test. Rather, under many circumstances very small variations will have to be considered acceptable and equivalent to an officially correct result.

In order to make evaluations as comparable as feasible, the evaluation table on the next page is recommended. For carrying out an evaluation of the output of the test page, the schematic structure of the test page on the page after the evaluation table may be used.

The following classification should be used for tracking evaluation results in the table

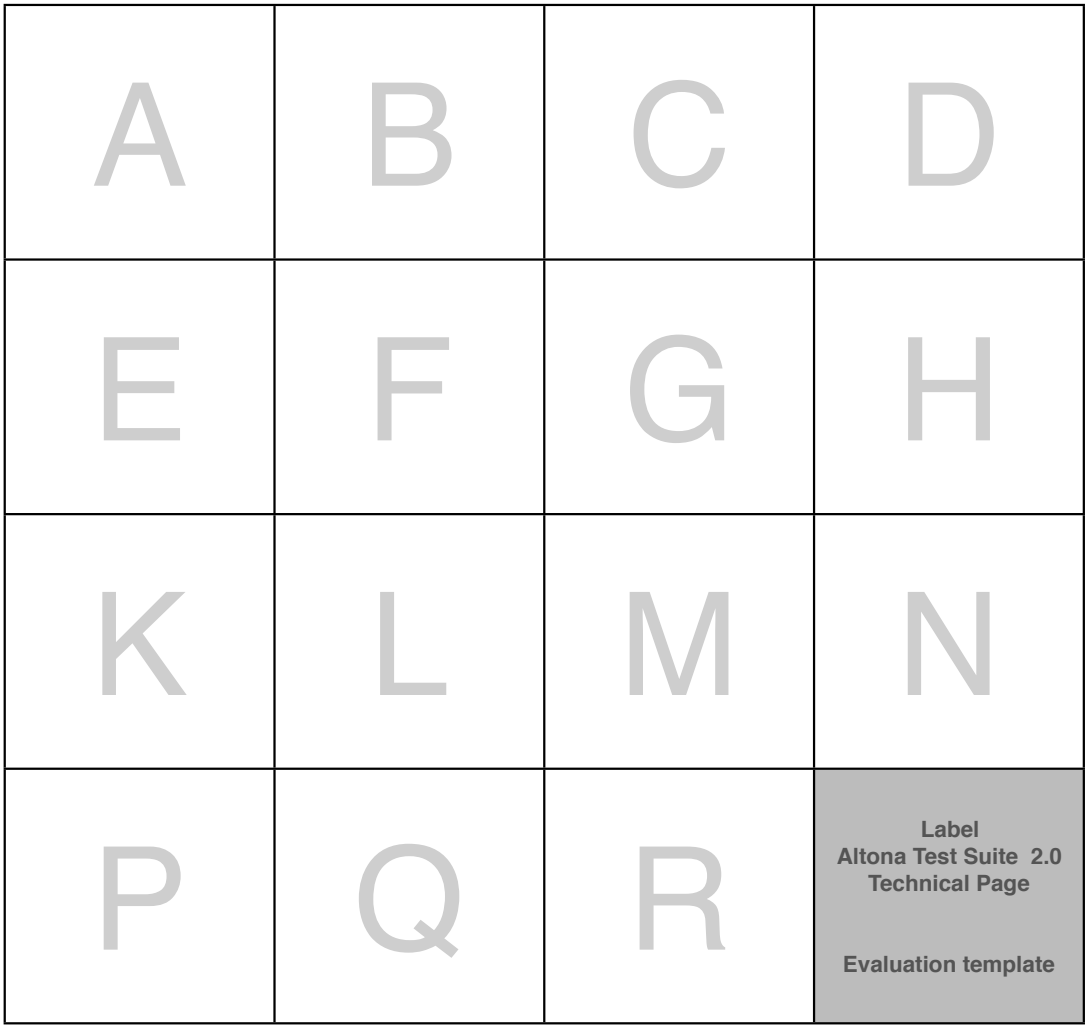
- no issues
- Very small deviations, but still acceptable
- Deviations that may or may not be acceptable
- Clearly unacceptable deviations
- Some or all of the patch not rendered at all

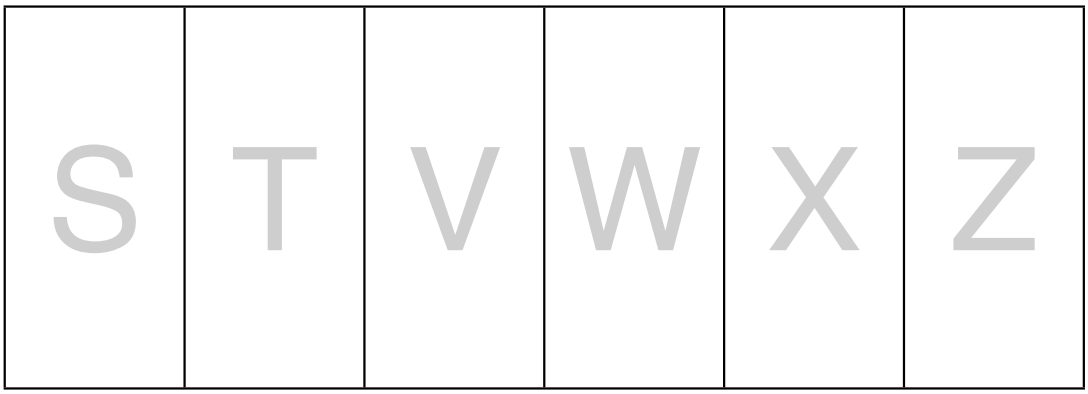
When carrying out an evaluation using the schematic structure, issues should be indicated by using "O" or an appropriate number of "X":

- O - no issues
- X - Very small deviations, but still acceptable
- XX - Deviations that may or may not be acceptable
- XXX - Clearly unacceptable deviations
- XXXX - Some or all of the patch not rendered at all

Note: When doing a visual assessment, slight color differences between spot color objects in the live part of a patch versus their appearance in the reference image do not indicate an error.

Altona Test Suite 2.0 · Technical Page · Evaluation template						
Patch	No issues	Very small deviations, but still acceptable	Deviations that may or may not be acceptable	Clearly unacceptable deviations	Some or all of the patch not rendered at all	Remarks
A						
B						
C						
D						
E						
F						
G						
H						
K						
L						
M						
N						
P						
Q						
R						
S						
T						
V						
W						
X						
Z						
Label						
Page frame						
Σ:						





Page frame

O *no issues* — **X** *Very small deviations, but still acceptable* — **XX** *Deviations that may or may not be acceptable* — **XXX** *Clearly unacceptable deviations* — **XXXX** *Some or all of the patch not rendered at all*