PDF Reference

sixth edition

Adobe® Portable Document Format

Version 1.7

November 2006

Adobe Systems Incorporated

Contents

CHAPTER 1

Introduction

The Adobe Portable Document Format (PDF) is the native file format of the Adobe® Acrobat® family of products. The goal of these products is to enable users to exchange and view electronic documents easily and reliably, independently of the environment in which they were created. PDF relies on the same imaging model as the PostScript® page description language to describe text and graphics in a device-independent and resolution-independent manner. To improve performance for interactive viewing, PDF defines a more structured format than that used by most PostScript language programs. PDF also includes objects, such as annotations and hypertext links, that are not part of the page itself but are useful for interactive viewing and document interchange.

1.1 About This Book

This book provides a description of the PDF file format and is intended primarily for developers of *PDF producer* applications that create PDF files directly. It also contains enough information to allow developers to write *PDF consumer* applications that read existing PDF files and interpret or modify their contents.

Although the *PDF Reference* is independent of any particular software implementation, some PDF features are best explained by describing the way they are processed by a typical application program. In such cases, this book uses the Acrobat family of PDF viewer applications as its model. (The prototypical viewer is the fully capable Acrobat product, not the limited Adobe Reader® product.) Appendix C discusses some implementation limits in the Acrobat viewer applications, even though these limits are not part of the file format itself. Appendix H provides compatibility and implementation notes that describe how Acrobat viewers behave when they encounter newer features they do not understand and specify areas in which the Acrobat products diverge from the specification presented in

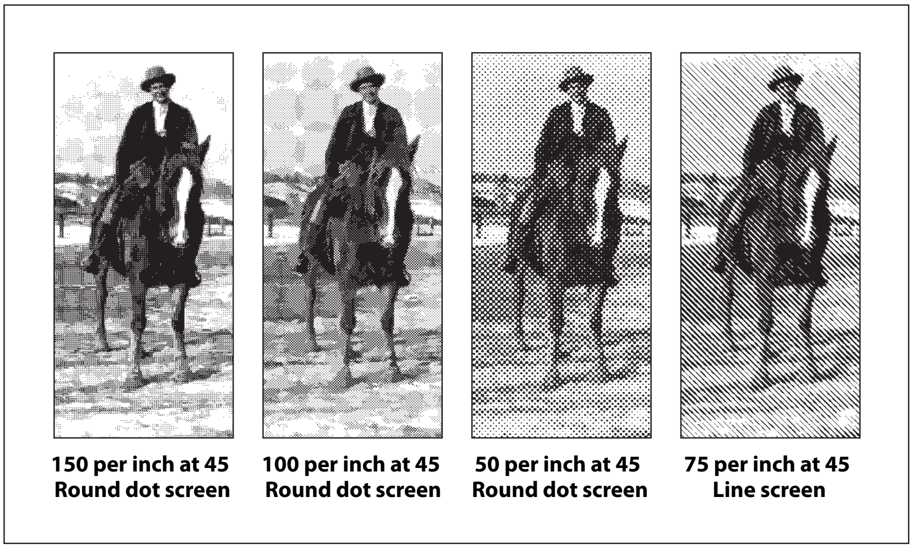


FIGURE 6.1 *Various halftoning effects*

6.4.3 Threshold Arrays

Another way to define a halftone screen is with a *threshold array* that directly controls individual device pixels in a halftone cell. This technique provides a high degree of control over halftone rendering. It also permits halftone cells to be arbitrary rectangles, whereas those controlled by a spot function are always square.

A threshold array is much like a sampled image—a rectangular array of pixel values—but is defined entirely in device space. Depending on the halftone type, the threshold values occupy 8 or 16 bits each. Threshold values nominally represent gray levels in the usual way, from 0 for black up to the maximum (255 or 65,535) for white. The threshold array is replicated to tile the entire device space: each pixel in device space is mapped to a particular sample in the threshold array. On a bilevel device, where each pixel is either black or white, halftoning with a threshold array proceeds as follows:

1. For each device pixel that is to be painted with some gray level, consult the corresponding threshold value from the threshold array.

2. If the requested gray level is less than the threshold value, paint the device pixel black; otherwise, paint it white. Gray levels in the range 0.0 to 1.0 correspond to threshold values from 0 to the maximum available (255 or 65,535).

Note: *A threshold value of 0 is treated as if it were 1; therefore, a gray level of 0.0 paints all pixels black, regardless of the values in the threshold array.*

This scheme easily generalizes to monochrome devices with multiple bits per pixel. For example, if there are 2 bits per pixel, each pixel can directly represent one of four different gray levels: black, dark gray, light gray, or white, encoded as 0, 1, 2, and 3, respectively. For any device pixel that is specified with some in-between gray level, the halftoning algorithm consults the corresponding value in the threshold array to determine whether to use the next-lower or next-higher representable gray level. In this situation, the threshold values do not represent absolute gray levels, but rather gradations between any two adjacent representable gray levels.

A halftone defined in this way can also be used with color displays that have a limited number of values for each color component. The red, green, and blue components are simply treated independently as gray levels, applying the appropriate threshold array to each. (This technique also works for a screen defined as a spot function, since the spot function is used to compute a threshold array internally.)

6.4.4 Halftone Dictionaries

In PDF 1.2, the graphics state includes a *current halftone* parameter, which determines the halftoning process to be used by the painting operators. The current halftone can be specified as the value of the **HT** entry in a graphics state parameter dictionary; see Table 4.8 on page 220. It may be defined by either a dictionary or a stream, depending on the type of halftone; the term *halftone dictionary* is used generically throughout this section to refer to either a dictionary object or the dictionary portion of a stream object. (The halftones that are defined by streams are specifically identified as such in the descriptions of particular halftone types; unless otherwise stated, they are understood to be defined by simple dictionaries instead.)

Every halftone dictionary must have a **HalftoneType** entry whose value is an integer specifying the overall type of halftone definition. The remaining entries in the dictionary are interpreted according to this type. PDF supports the halftone types listed in Table 6.2.

TABLE 6.2 PDF halftone types

TYPE	MEANING
1	Defines a single halftone screen by a *frequency, angle,* and *spot function.*
5	Defines an arbitrary number of halftone screens, one for each colorant or color component (including both primary and spot colorants). The keys in this dictionary are names of colorants; the values are halftone dictionaries of other types, each defining the halftone screen for a single colorant.
6	Defines a single halftone screen by a threshold array containing 8-bit sample values.
10	Defines a single halftone screen by a threshold array containing 8-bit sample values, representing a halftone cell that may have a nonzero screen angle.
16	*(PDF 1.3)* Defines a single halftone screen by a threshold array containing 16-bit sample values, representing a halftone cell that may have a nonzero screen angle.

The dictionaries representing these halftone types contain the same entries as the corresponding PostScript language halftone dictionaries (as described in Section 7.4 of the *PostScript Language Reference*, Third Edition), with the following exceptions:

- The PDF dictionaries may contain a **Type** entry with the value **Halftone**, identifying the type of PDF object that the dictionary describes.
- Spot functions and transfer functions are represented by function objects instead of PostScript procedures.
- Threshold arrays are specified as streams instead of files.
- In type 5 halftone dictionaries, the keys for colorants must be name objects; they may not be strings as they may in PostScript.

Halftone dictionaries have an optional entry, **HalftoneName**, that identifies the halftone by name. In PDF 1.3, if this entry is present, all other entries, including **HalftoneType**, are optional. At rendering time, if the output device has a halftone with the specified name, that halftone is used, overriding any other halftone parameters specified in the dictionary. This provides a way for PDF documents to select the proprietary halftones supplied by some device manufacturers, which would not otherwise be accessible because they are not explicitly defined in PDF. If there is no **HalftoneName** entry, or if the requested halftone name does not exist on the device, the halftone's parameters are defined by the other entries in the dictionary, if any. If no other entries are present, the default halftone is used.

See Section 7.6.4, "Rendering Parameters and Transparency," and in particular, "Halftone and Transfer Function" on page 573, for further discussion of the role of halftones in the transparent imaging model.

Type 1 Halftones

Table 6.3 describes the contents of a halftone dictionary of type 1, which defines a halftone screen in terms of its frequency, angle, and spot function.

TABLE 6.3 Entries in a type 1 halftone dictionary

KEY	TYPE	VALUE
Type	name	*(Optional)* The type of PDF object that this dictionary describes; if present, must be **Halftone** for a halftone dictionary.
HalftoneType	integer	*(Required)* A code identifying the halftone type that this dictionary describes; must be 1 for this type of halftone.
HalftoneName	byte string	*(Optional)* The name of the halftone dictionary.
Frequency	number	*(Required)* The screen frequency, measured in halftone cells per inch in device space.
Angle	number	*(Required)* The screen angle, in degrees of rotation counterclockwise with respect to the device coordinate system. (Most output devices have left-handed device spaces. On such devices, a counterclockwise angle in device space corresponds to a clockwise angle in default user space and on the physical medium.)
SpotFunction	function or name	*(Required)* A function object defining the order in which device pixels within a screen cell are adjusted for different gray levels, or the name of one of the predefined spot functions (see Table 6.1 on page 489).
AccurateScreens	boolean	*(Optional)* A flag specifying whether to invoke a special halftone algorithm that is extremely precise but computationally expensive; see below for further discussion. Default value: **false**.

KEY	TYPE	VALUE
TransferFunction	function or name	*(Optional)* A transfer function, which overrides the current transfer function in the graphics state for the same component. This entry is required if the dictionary is a component of a type 5 halftone (see "Type 5 Halftones" on page 505) and represents either a nonprimary or nonstandard primary color component (see Section 6.3, "Transfer Functions"). The name **Identity** may be used to specify the identity function.

If the **AccurateScreens** entry has a value of **true**, a highly precise halftoning algorithm is substituted in place of the standard one. If **AccurateScreens** is **false** or not present, ordinary halftoning is used. Accurate halftoning achieves the requested screen frequency and angle with very high accuracy, whereas ordinary halftoning adjusts them so that a single screen cell is quantized to device pixels. High accuracy is important mainly for making color separations on high-resolution devices. However, it may be computationally expensive and therefore is ordinarily disabled.

In principle, PDF permits the use of halftone screens with arbitrarily large cells—in other words, arbitrarily low frequencies. However, cells that are very large relative to the device resolution or that are oriented at unfavorable angles may exceed the capacity of available memory. If this happens, an error occurs. The **AccurateScreens** feature often requires very large amounts of memory to achieve the highest accuracy.

Example 6.1 shows a halftone dictionary for a type 1 halftone.

Example 6.1

```
28 0 obj
   << /Type /Halftone
      /HalftoneType 1
      /Frequency 120
      /Angle 30
      /SpotFunction /CosineDot
      /TransferFunction /Identity
   >>
endobj
```

Type 6 Halftones

A type 6 halftone defines a halftone screen with a threshold array. The halftone is represented as a stream containing the threshold values; the parameters defining the halftone are specified by entries in the stream dictionary. This dictionary can contain the entries shown in Table 6.4 in addition to the usual entries common to all streams (see Table 3.4 on page 62). The **Width** and **Height** entries specify the dimensions of the threshold array in device pixels; the stream must contain **Width** × **Height** bytes, each representing a single threshold value. Threshold values are defined in device space in the same order as image samples in image space (see Figure 4.26 on page 338), with the first value at device coordinates (0, 0) and horizontal coordinates changing faster than vertical coordinates.

Type 10 Halftones

Although type 6 halftones can be used to specify a threshold array with a zero screen angle, they make no provision for other angles. The type 10 halftone removes this restriction and allows the use of threshold arrays for halftones with nonzero screen angles as well.

TABLE 6.4 Additional entries specific to a type 6 halftone dictionary

KEY	TYPE	VALUE
Type	name	*(Optional)* The type of PDF object that this dictionary describes; if present, must be **Halftone** for a halftone dictionary.
HalftoneType	integer	*(Required)* A code identifying the halftone type that this dictionary describes; must be 6 for this type of halftone.
HalftoneName	byte string	*(Optional)* The name of the halftone dictionary.
Width	integer	*(Required)* The width of the threshold array, in device pixels.
Height	integer	*(Required)* The height of the threshold array, in device pixels.
TransferFunction	function or name	*(Optional)* A transfer function, which overrides the current transfer function in the graphics state for the same component. This entry is required if the dictionary is a component of a type 5 halftone (see "Type 5 Halftones" on page 505) and represents either a nonprimary or nonstandard primary color component (see Section 6.3, "Transfer Functions"). The name **Identity** may be used to specify the identity function.

Halftone cells at nonzero angles can be difficult to specify because they may not line up well with scan lines and because it may be difficult to determine where a given sampled point goes. The type 10 halftone addresses these difficulties by dividing the halftone cell into a pair of squares that line up at zero angles with the output device's pixel grid. The squares contain the same information as the original cell but are much easier to store and manipulate. In addition, they can be mapped easily into the internal representation used for all rendering.

Figure 6.2 shows a halftone cell with a frequency of 38.4 cells per inch and an angle of 50.2 degrees, represented graphically in device space at a resolution of 300 dots per inch. Each asterisk in the figure represents a location in device space that is mapped to a specific location in the threshold array.

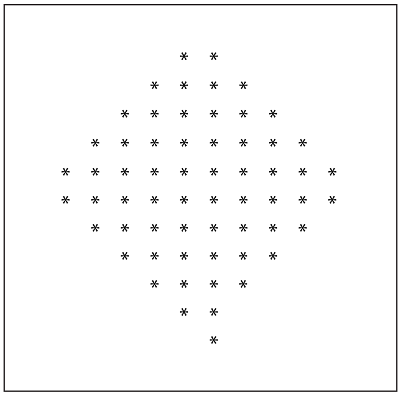


FIGURE 6.2 *Halftone cell with a nonzero angle*

Figure 6.3 shows how the halftone cell can be divided into two squares. If the squares and the original cell are tiled across device space, the area to the right of the upper square maps exactly into the empty area of the lower square, and vice versa (see Figure 6.4). The last row in the first square is immediately adjacent to the first row in the second square and starts in the same column.

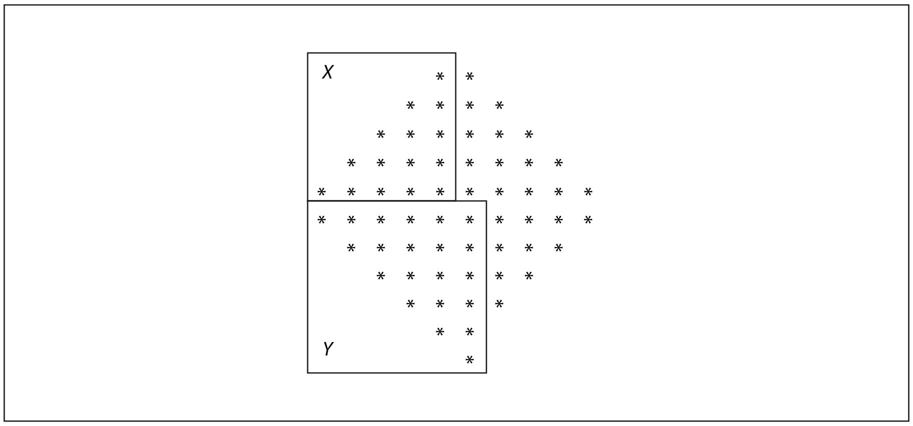


FIGURE 6.3 *Angled halftone cell divided into two squares*

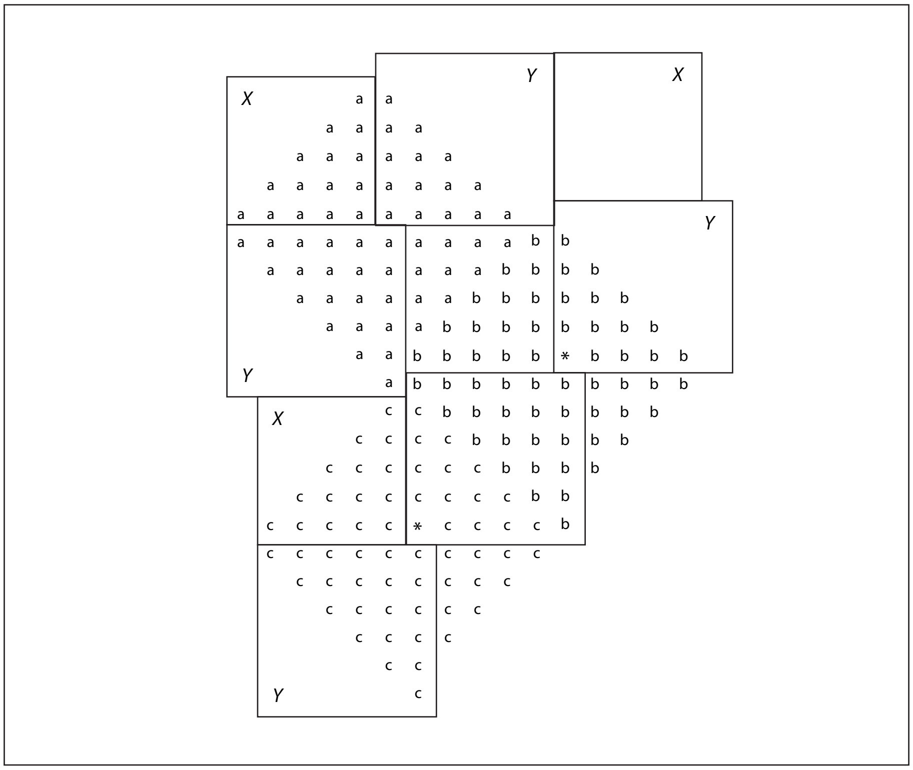


FIGURE 6.4 *Halftone cell and two squares tiled across device space*

Any halftone cell can be divided in this way. The side of the upper square *(X)* is equal to the horizontal displacement from a point in one halftone cell to the corresponding point in the adjacent cell, such as those marked by asterisks in Figure 6.4. The side of the lower square *(Y)* is the vertical displacement between the same two points. The frequency of a halftone screen constructed from squares with sides *X* and *Y* is thus given by

$$\textit{frequency} = \frac{\textit{resolution}}{\sqrt{X^2 + Y^2}}$$

and the angle by

$$\textit{angle} = \operatorname{atan}\left(\frac{Y}{X}\right)$$

Like a type 6 halftone, a type 10 halftone is represented as a stream containing the threshold values, with the parameters defining the halftone specified by entries in the stream dictionary. This dictionary can contain the entries shown in Table 6.5 in addition to the usual entries common to all streams (see Table 3.4 on page 62). The **Xsquare** and **Ysquare** entries replace the type 6 halftone's **Width** and **Height** entries.

TABLE 6.5 Additional entries specific to a type 10 halftone dictionary

KEY	TYPE	VALUE
Type	name	*(Optional)* The type of PDF object that this dictionary describes; if present, must be **Halftone** for a halftone dictionary.
HalftoneType	integer	*(Required)* A code identifying the halftone type that this dictionary describes; must be 10 for this type of halftone.
HalftoneName	byte string	*(Optional)* The name of the halftone dictionary.
Xsquare	integer	*(Required)* The side of square *X*, in device pixels; see below.
Ysquare	integer	*(Required)* The side of square *Y*, in device pixels; see below.
TransferFunction	function or name	*(Optional)* A transfer function, which overrides the current transfer function in the graphics state for the same component. This entry is required if the dictionary is a component of a type 5 halftone (see "Type 5 Halftones" on page 505) and represents either a nonprimary or nonstandard primary color component (see Section 6.3, "Transfer Functions"). The name **Identity** may be used to specify the identity function.

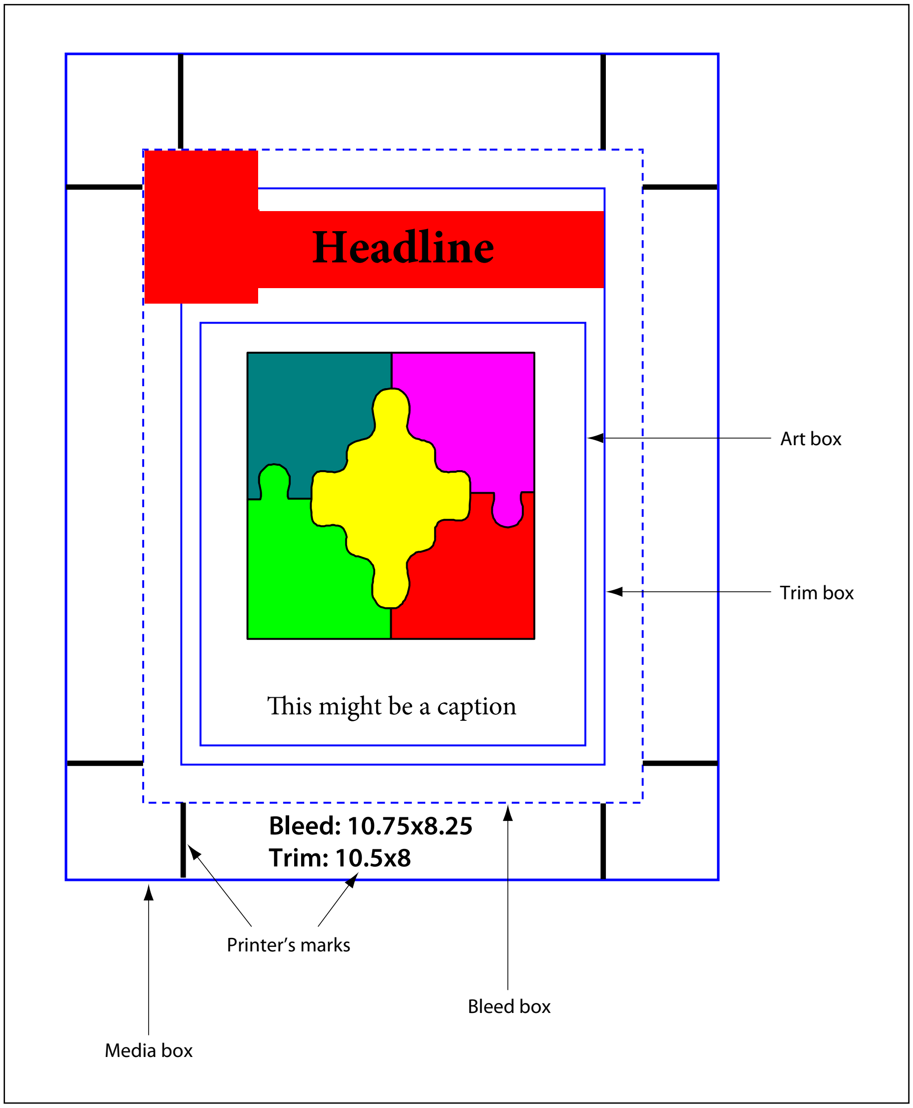


FIGURE 10.3 *Page boundaries*

How the various boundaries are used depends on the purpose to which the page is being put. The following are typical purposes:

- *Placing the content of a page in another application.* The art box determines the boundary of the content that is to be placed in the application. Depending on

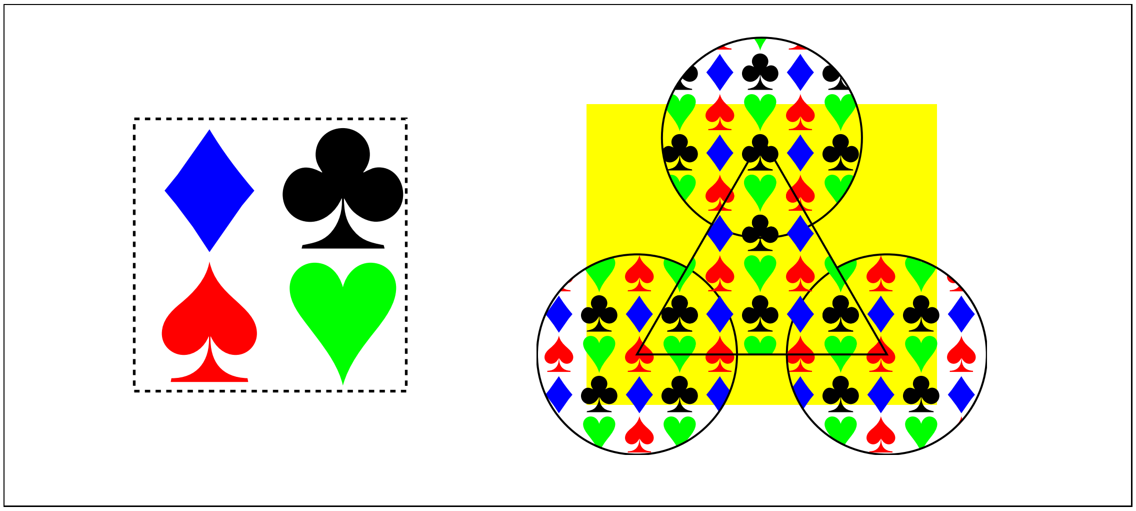


PLATE 8 *Colored tiling pattern ("Colored Tiling Patterns," page 295)*

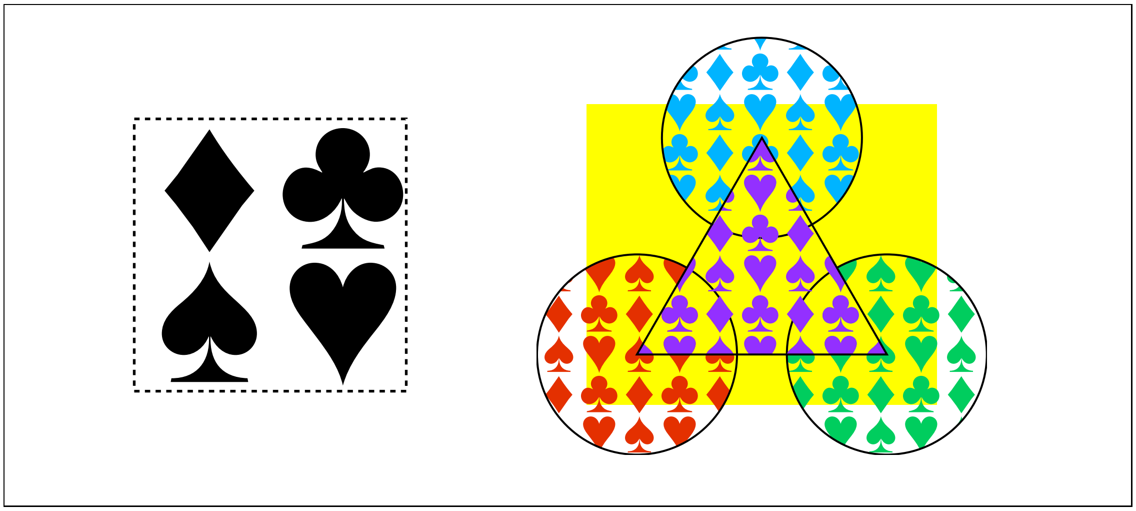
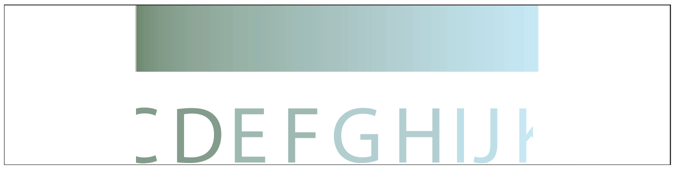
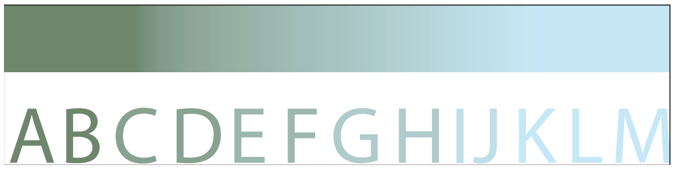


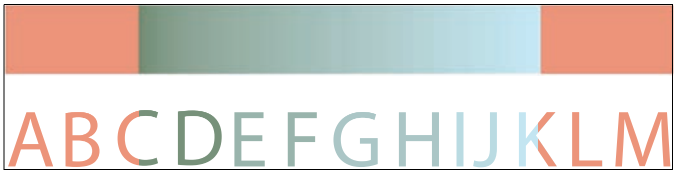
PLATE 9 *Uncolored tiling pattern ("Uncolored Tiling Patterns," page 299)*



Extend = [false false], **Background** not specified



Extend = [true true], **Background** not specified



Extend = [true true], **Background** not specified

PLATE 10 *Axial shading ("Type 2 (Axial) Shadings," page 310)*

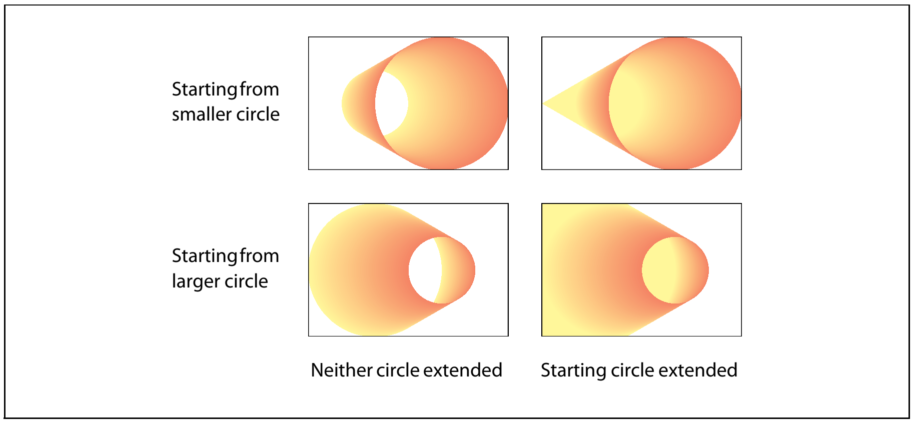


PLATE 11 *Radial shadings depicting a cone ("Type 3 (Radial) Shadings," page 312)*

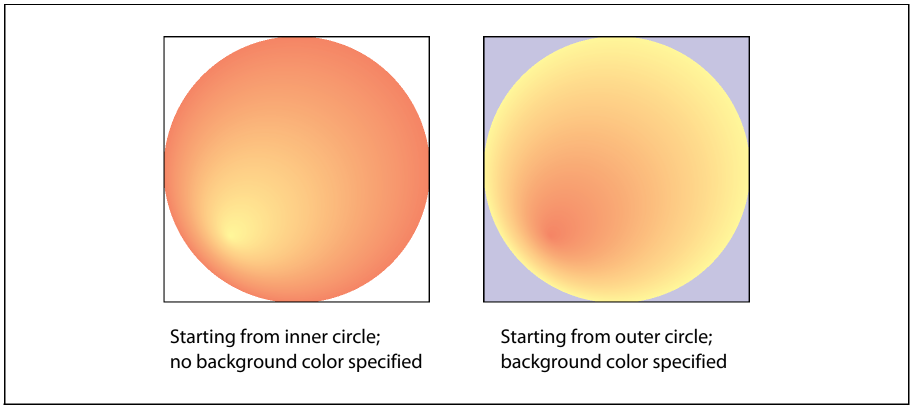

Starting from inner circle;
no background color specified
Starting from outer circle;
background color specified

PLATE 12 *Radial shadings depicting a sphere ("Type 3 (Radial) Shadings," page 313)*

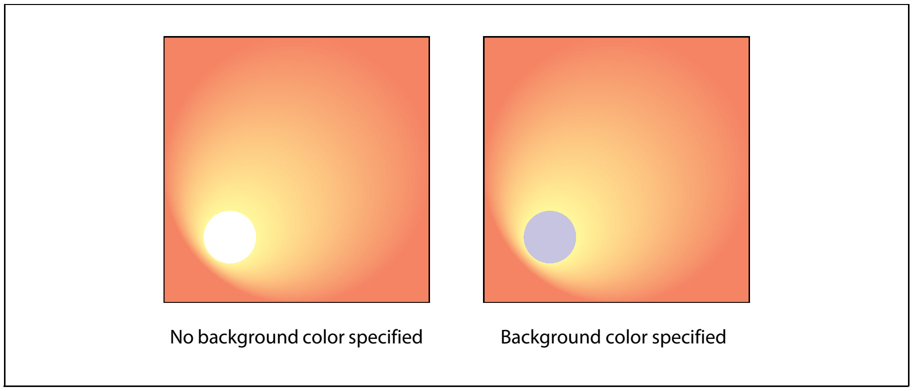


PLATE 13 *Radial shadings with extension ("Type 3 (Radial) Shadings," page 313)*

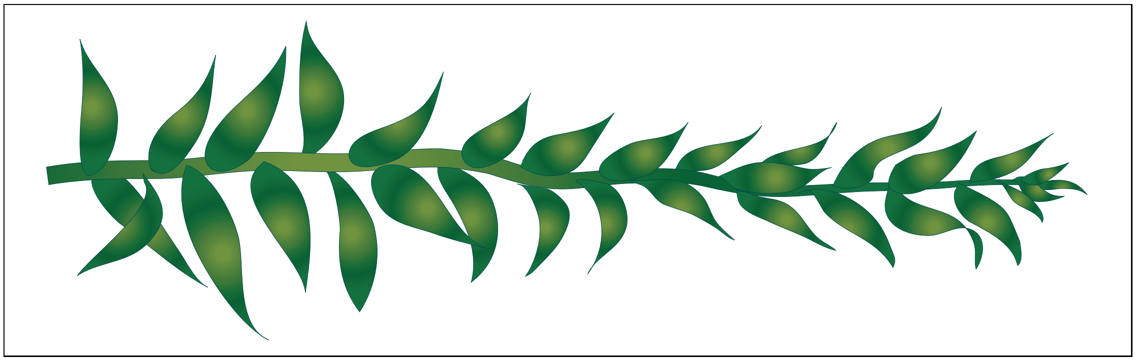


PLATE 14 *Radial shading effect ("Type 3 (Radial) Shadings," page 313)*