

TABLE 4.22 Entries in a type 3 image dictionary

KEY	TYPE	VALUE
ImageType	integer	(Required) A code identifying the image type that this dictionary describes; must be 3 for explicit masking.
DataDict	dictionary	(Required) A modified type 1 image dictionary defining the contents of the image (see Table 4.23).
MaskDict	dictionary	(Required) A modified type 1 image dictionary defining the image's mask (see Table 4.24).
InterleaveType	integer	(Required) A code indicating how the image and mask samples are organized:

- 1 *Interleaved by sample.* Image and mask samples are combined into a single data source, identified by the **DataSource** entry in the image data dictionary; the mask dictionary must contain no **DataSource** entry. Components are interleaved sample by sample, with the mask component preceding all color components; for example, in the **DeviceRGB** color space, each sample would consist of a mask component followed by three color components (red, green, and blue). The mask sample must have the same number of bits as each color component of the image sample, with all bits set to the same value (that is, either all 0 or all 1); any other value will be treated as if the bits were all 1.
- 2 *Interleaved by row.* Image and mask samples are combined into a single data source, identified by the **DataSource** entry in the image data dictionary; the mask dictionary must contain no **DataSource** entry. Mask and image data are organized into *interleave blocks* whose format is determined by the **Height** entries in the image data and mask dictionaries. The heights given in the two dictionaries may differ, with the restriction that one must be an integral multiple of the other. Each interleave block thus consists of either one row of mask data followed by one or more rows of image data, or one or more rows of mask data followed by one row of image data, according to the ratio of the heights specified in the two dictionaries.

All interleave blocks have the same format. Within each block, all of the mask data precedes all of the image data. Mask data is always one bit per sample, regardless of the number of bits per sample in the image data. Within the image data, color compo-

nents are interleaved on a sample-by-sample basis. Each row of mask and image samples is padded separately to byte boundaries.

- 3 *Separate data sources.* Image and mask samples are provided through separate data sources, identified by the **DataSource** entries in the image data dictionary and the mask dictionary, respectively. The color components of the image samples themselves may in turn be interleaved or separate, depending on the value of the **MultipleDataSources** entry in the image data dictionary. The width and height of the mask are independent of those of the image, but the image and mask must have the same orientation and placement.

TABLE 4.23 Entries in an image data dictionary

KEY	TYPE	VALUE
ImageType	integer	<i>(Required)</i> A code identifying the image type that this dictionary describes; must be 1 for an image data dictionary.
Width	integer	<i>(Required)</i> The width of the source image, in samples. In interleave type 1, this value must equal that of the Width entry in the mask dictionary.
Height	integer	<i>(Required)</i> The height of the source image, in samples. In interleave type 1, this value must equal that of the Height entry in the mask dictionary. In interleave type 2, the image and mask heights may differ, with the restriction that one must be an integral multiple of the other. In interleave type 3, the heights of the image and mask are independent.
ImageMatrix	array	<i>(Required)</i> An array of six numbers defining a transformation from user space to image space.
MultipleDataSources	boolean	<i>(Optional)</i> A flag indicating whether the image samples are provided through a separate data source for each color component (<i>true</i>) or packed into one data stream, interleaved sample by sample (<i>false</i>). If this entry is <i>true</i> , the interleave type in the main image dictionary (Table 4.22) must be 3; that is, the image's mask data must also be provided through a separate data source, designated by the DataSource entry in the mask dictionary (Table 4.24). For interleave types 1 and 2, MultipleDataSources must be <i>false</i> . Default value: <i>false</i> .
DataSource	(various)	<i>(Required)</i> The source from which image samples are to be taken. If MultipleDataSources is <i>false</i> or absent, DataSource must be a single file, procedure, or string. If MultipleDataSources is <i>true</i> , DataSource must be an array of n such data sources, where n is the number of color components in the current color space. For interleave types 1 and 2, the designated data source will also include mask samples interleaved with the

source samples in the manner implied by the interleave type (see Table 4.22); in this case, the mask dictionary (Table 4.24) must contain no **Data-Source** entry.

BitsPerComponent	integer	<i>(Required)</i> The number of bits used to represent each color component. Only a single number may be specified; the number of bits is the same for all color components. Allowed values are 1, 2, 4, 8, and 12.
Decode	array	<i>(Required)</i> An array of numbers describing how to map image samples into the range of values appropriate for the current color space; see “Sample Decoding” on page 299. The length of the array must be twice the number of color components in the current color space.
Interpolate	boolean	<i>(Optional)</i> A flag indicating whether image interpolation is to be performed on the source data; see “Image Interpolation” on page 301. Default value: <i>false</i> .

TABLE 4.24 Entries in a mask dictionary

KEY	TYPE	VALUE
ImageType	integer	<i>(Required)</i> A code identifying the image type that this dictionary describes; must be 1 for a mask dictionary.
Width	integer	<i>(Required)</i> The width of the mask, in samples. In interleave type 1, this value must equal that of the Width entry in the image data dictionary.
Height	integer	<i>(Required)</i> The height of the mask, in samples. In interleave type 1, this value must equal that of the Height entry in the image data dictionary. In interleave type 2, the image and mask heights may differ, with the restriction that one must be an integral multiple of the other. In interleave type 3, the heights of the image and mask are independent.
ImageMatrix	array	<i>(Required)</i> An array of six numbers defining a transformation from user space to image space. This matrix must align the corners of the mask with the corresponding corners of the image, so that they coincide in user space, and must perform any scaling needed to compensate for differences in the dimensions of the mask and the image.
MultipleDataSources	boolean	<i>(Optional)</i> If present, must be <i>false</i> . Default value: <i>false</i> .
DataSource	(various)	<i>(Required for interleave type 3)</i> The source (a single file, procedure, or string) from which mask samples are to be taken. This entry must be absent for interleave type 1 or 2.
BitsPerComponent	integer	<i>(Required)</i> The number of bits used to represent each color component. In interleave type 1, this value must equal that of the BitsPerComponent

		entry in the image data dictionary. In interleave type 2 or 3, the value of this entry must be 1.
Decode	array	<i>(Required)</i> An array of two numbers describing how to map mask samples into the appropriate range of values; see “Sample Decoding” on page 299. A decoded value of 0 designates a sample to be painted; a decoded value of 1 designates a sample that is to be masked out (not painted).
Interpolate	boolean	<i>(Optional)</i> A flag indicating whether image interpolation is to be performed on the mask; see “Image Interpolation” on page 301. Default value: <i>false</i> .

Color Key Masking

Type 4 image dictionaries (Table 4.25) are identical to type 1 with one additional entry, **MaskColor**, specifying a color or range of colors to be masked out. Samples in the image that match this color or fall within this range are not painted, allowing the existing background to show through. The effect is similar to that of the video technique known as *chroma-key*.

TABLE 4.25 Entries in a type 4 image dictionary

KEY	TYPE	VALUE
ImageType	integer	<i>(Required)</i> A code identifying the image type that this dictionary describes; must be 4 for color key masking.
MaskColor	array	<p><i>(Required)</i> An array of integers specifying the color to be masked. The array may contain either n or $2 \times n$ integers, where n is the number of components required to specify a color in the current color space. If n integers are given, they specify the masked color exactly; any image sample whose color components match the values in the array will be masked out (not painted). If the array contains $2 \times n$ integers, each pair of consecutive integers specify a range of values for the corresponding color component; an image sample will be masked if each of its component values falls within the specified range.</p> <p>Image samples are compared with the mask color as they are read from the data source. This comparison occurs before the application of the decode mapping specified by the Decode array.</p>
Width	integer	<i>(Required)</i> The width of the source image, in samples.
Height	integer	<i>(Required)</i> The height of the source image, in samples.