7.6.5 PostScript Compatibility

Because the PostScript language does not support the transparent imaging model, PDF 1.4 consumer applications must have some means for converting the appearance of a document that uses transparency to a purely opaque description for printing on PostScript output devices. Similar techniques can also be used to convert such documents to a form that can be correctly viewed by PDF 1.3 and earlier consumers.

Converting the contents of a page from transparent to opaque form entails some combination of shape decomposition and prerendering to flatten the stack of transparent objects on the page, perform all the needed transparency computations, and describe the final appearance using opaque objects only. Whether the page contains transparent content needing to be flattened can be determined by straightforward analysis of the page's resources; it is not necessary to analyze the content stream itself. The conversion to opaque form is irreversible, since all information about how the transparency effects were produced is lost.

To perform the transparency computations properly, the application needs to know the native color space of the output device. This is no problem when the application controls the output device directly. However, when generating PostScript output, the application has no way of knowing the native color space of the PostScript output device. An incorrect assumption will ruin the calibration of any CIE-based colors appearing on the page. This problem can be addressed in either of two ways:

- If the entire page consists of CIE-based colors, flatten the colors to a single CIE-based color space rather than to a device color space. The preferred color space for this purpose can easily be determined if the page has a group attributes dictionary (**Group** entry in the page object) specifying a CIE-based color space (see Section 7.5.5, "Transparency Group XObjects").
- Otherwise, flatten the colors to some assumed device color space with predetermined calibration. In the generated PostScript output, paint the flattened colors in a CIE-based color space having that calibration.

Because the choice between using spot colorants and converting them to an alternate color space affects the flattened results of process colors, a decision must also be made during PostScript conversion about the set of available spot colorants to assume. (This differs from strictly opaque painting, where the decision can be deferred until the generated PostScript code is executed.)