

進 明

化 日

す は

る ど

D つ

T ち

P だ

、

上高地 仁

incunabula

目次



貼り込み画像の標準は EPS から Photoshop 形式にシフトする 6

- EPS が貼り込み画像のスタンダード
- ハンドリングし易く安定した EPS 形式
- モニタブルーフができない EPS 画像
- ロスレスでクリッピングマスク対応は Photoshop 形式のみ

劣化する JPEG 圧縮は最後に適用するのが賢い使い方 12

- ファイルサイズの大きな画像は扱いにくい
- カラープリンタから実画像をプリントできない OPI や DCS
- 圧縮率は高いがオリジナル情報が失われる JPEG 圧縮
- 普及した「JPEG エンコーディング」形式の EPS 画像
- JPEG は圧縮を重ねると劣化が進む

フォントのダウンロードは PDF のエンベットの仕組みで 18

- 制限の多い日本語フォントはトラブルが多い
- 低くなるフォントのプロテクト
- アウトライン化すると太って出力される
- アプリケーションや OS でするフォントのラスタライズ
- PDF でするフォントのダイナミックダウンロード
- ダイナミックダウンロードが常識になればフォントトラブルは減少する

フォントフォーマットの改変はどこまで続く 24

- DTP のコストの要はフォントにあった
- 安全性のためにはフォント環境は変えられない
- CID と CMap を使うと文字化けしない
- アプリケーションではあまり使われてない CMap
- フォントフォーマットはこれからも変わり続ける

PostScript で扱えない透明効果はどこで分割する 30

- いまだに Illustrator のスタンダードバージョンは 8.0
- 9.0 から採用された PDF の透明機能
- 透明が適用される部分は別のオブジェクトに作り直す
- 透明が分割されるのはいつ



ベクトルを維持して統合の解像度を指定する
透明の分割・統合は出力前が理想

PDF の編集を Illustrator ですれば PDF の入稿は怖くない 38

出力前の訂正は Illustrator で開けば対処できる
PDF は Illustrator で正しく開けない
TouchUp テキストツールでする PDF のテキストの修正
Illustrator CS ではレイアウトを維持して PDF が開く

PDF の普及はアプリケーションの PDF 保存が決めて 44

バージョンアップした Acrobat 6.0 Professional
出稿側にある PDF で出稿するメリット
Distiller でプリプレス用 PDF を作成するのは難しい
普及していないアプリケーションでの PDF 保存
アプリケーションでの PDF 保存はいずれ当たり前になる

Photoshop のカラー設定でするカラーマネージメント（前編）..... 50

コストのかかるカラーマネージメントは普及しない
カラーマネージメントに必要な 3 つの要素
Adobe Color Engine のできるカラーマネージメント
RGB を統一する RGB 作業用スペース
最終ターゲットを指定する CMYK 作業用スペース

Photoshop のカラー設定でするカラーマネージメント（後編）..... 56

「カラーマッチングモジュール」は Adobe(ACE) を選択する
印刷で利用する「知覚的」と「相対的な色域を維持」
異なるアプリケーションでカラー設定を共通にする
DTP で必要不可欠になる ACE のカラー設定

モニタで校正する印刷のシミュレーション 64

印刷するときのカラーをモニタに表示する
標準的なプロファイルで CMYK をマネージメントする
洗練された「Japan Color 2001 Coated」
モニタの固体差はモニタプロファイルで吸収する



モニタカラーマッチングで印刷のシミュレーション	
プリンタで校正する印刷のシミュレーション	70
カラープリンタで手軽にするプリンタカラーマッチング	
プリンタでも必要なプリンタのデバイスプロファイル	
ACE だけでカラーマネージメント	
カラープリンタでカラーマッチングの注意点	
OpenType のコピー＆ペーストはいつでもできる	76
いままでの JIS 規格だけでは足りなくなった文字コード	
字形として定義されていなかった JIS 規格	
JIS にはない文字は CID と CMap で扱う	
現在は OS の制限を受ける OpenType フォント	
Mac OS X ではユニコードで文字情報を交換できる	
番外編 「InDesign ソリューションセミナー」をレポートする	84
文書構造がわからないと利用できなかった XML	
フィードバックが難しい AppleScript の自動処理	
プログラムの改変とデータベースとの連携	
InDesign に CSV を読み込み XML で自動組版する	
XML データのマルチユースを提案する	
本書をもっと理解するための参考書籍	89
あとがきにかえて	90

			明	進
			日	化
			は	す
			ど	る
			っ	D
			ち	T
			だ	P
				、

第一回

貼り込み画像の標準は EPS から Photoshop 形式にシフトする

EPS が貼り込み画像のスタンダード

DTP アプリケーションを扱ってデータを作成するときには、いくつかのルールがあります。誰かが勝手に決めたわけではありませんが、多くのユーザーが「こうした方がいい」というやり方が、不文律といっていいルールになっています。

もちろんこれらの不文律は、さまざまな試行錯誤の中から生まれました。失敗を重ねる中で、もっとも的確なやり方が経験的に導き出されて、不文律となったのです。

そういう不文律の中には、画像フォーマットのスタンダードは「EPS」というフォーマットで貼り込むというものがあります。QuarkXPress にしても、Illustrator にしても、Photoshop で保存するときは、Photoshop EPS で保存してレイアウトソフトに貼り込むのです。

切り抜きのない角版ばかりを大量に処理するときは、TIFF を使うこともあります。OPI サーバというサーバを使って、レイアウト時は低解像度画像、出力前に高解像度画像に差し替えて出力する場合は、EPS ではなく TIFF を使うことがよくあります。これは、EPS より TIFF の方が画像がすこしばかり軽いので、大量に処理するときは TIFF を使うようになったものです。ファイルサイズの少しの違いでも、何百点、何千点となると、膨大なデータ量になるので、少しでも軽い方がいいわけです。

とはいえ、一般的な DTP の画像データでは EPS が基本です。ではなぜ EPS がスタンダードになったのでしょうか。

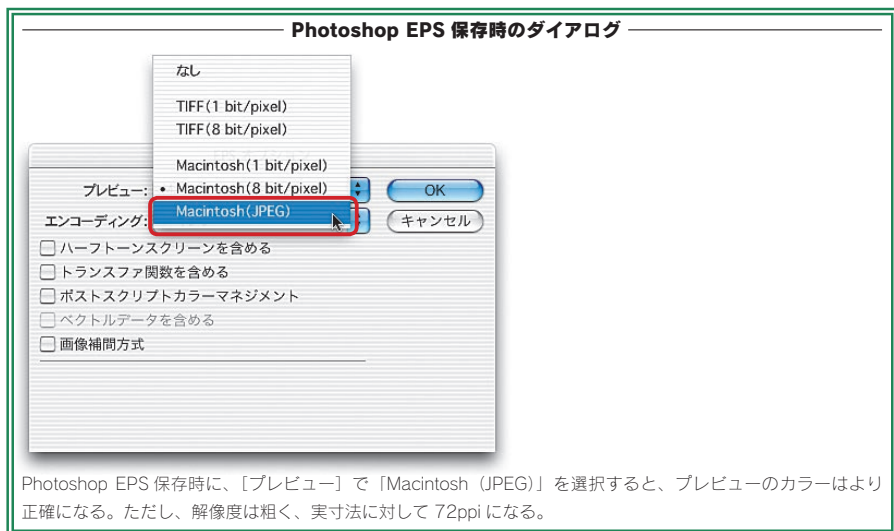
ハンドリングし易く安定した EPS 形式

EPS が画像のフォーマットとして認知された理由は、なんといっても、扱いやすさではないでしょうか。EPS は画像ファイルとしては珍しく二重構造のファイルになっています。

二重構造というのは、実画像のデータとプレビューデータの二つのデータを併せ持っていることです。このプレビュー画像がある分だけ、EPS ファイルはファイルサイズが肥大化するのですが、プレビューがあるからこそレイアウトソフトに貼り込みやすいという特質がありました。

レイアウトソフトに実画像を貼り込むと、これが重たいわけです。スクロールするだけでも、モニタの描画をやり直す必要がありますから、力士が四股を踏んでいるような悠長さで画像が動いたりします。重たい実画像を再演算するだけで、マシンには負担になるわけです。

ですから、レイアウトソフトにはプレビューが貼り込める EPS が重宝したのです。ただそうはいつても、ネックがありました。普及し始めていた QuickDraw プリンタでは、モニタ描画用のデータを利用してプリントしていましたから、プリントするとプレビューで印刷されてしまうのです。これでは DTP の世界では実質的にプルーフ代わりにはならないわけです。



それで、プレビューをもっと高精度に作る裏技がありました。プレビューは Photoshop で保存するときに実寸法のサイズから 72ppi のサイズで作成されますから、実寸法を大きくして、レイアウトソフトで縮小したうえで、プレビューの形式を「8 bit/pixel」から「JPEG」にすると、QuickDraw プリンタでもそこその品質でプリントアウトできました。もっともカラーの再現は当てにはなりませんでしたがね。

モニタプルーフができない EPS 画像

ただし、ここにくて実画像のデータを扱うことが難しくなってきました。マシンの CPU が高速化したことで、アプリケーションが実画像のデータをそのまま貼り込んでも、ストレスなく表示できるわけです。つまり、アプリケーションのドキュメントに実画像を貼り込んでも十分使えるわけです。

しかしなぜドキュメントで実画像を表示させないといけないのでしょうか。い



ままだレビューで利用できていたわけですから、レイアウトデータを貼り込むだけであればレビューでもかまわないわけです。

それは、モニターでのカラーマッチングができるようになってきたからです。モニターに実画像を表示すると、Photoshop と同じカラーを再現できて、なおかつ、そのカラーがオフセット印刷をシミュレーションしたものになっているわけです。つまり、印刷したそのカラーをモニターで再現できるようになってきているのです。

モニターで印刷結果を再現するためには、そりゃレビューではいけません。正味のデータを貼り込んでレイアウトするべきです。

つまり、モニターで色校正できるわけです。本当にできます。もちろん、そのためには印刷用の CMYK のプロファイルが必要になりますし、モニタープロファイルも必要です。しかし、それがあれば、Illustrator や InDesign ではモニター上のソフトプルーフが可能なのです。

ですから、色校正も、クライアントのマシンでデータを開いて、モニター上で最終確認ということができる環境は整いつつあるわけです。もっともそのためには、揃えないといけないことが多々ありますが、DTP の方向としては、そういう方向に行くだろうとっていいでしょう。

レイアウトソフトには実画像を貼り込むには、残念ながら、DTP のレイアウトソフトのスタンダードである QuarkXPress では無理です。QuarkXPress は相変わらずレビューの貼り込んだ EPS が基本です。

だから、そういう使い方ができるアプリケーションは、Photoshop 6.0 以降、Illustrator 9.0 以降、InDesign 2.0、Acrobat 5.0 ということになってしまいます。そういう意味ではアプリケーションやバージョンの制限はありますが、モニターで印刷したときのカラーが本当に再現できるのであれば、その壁は決して高くないと私とは思っています。

—— ロスレスでクリッピングマスク対応は Photoshop 形式のみ ——

実画像を貼り込む場合、Photoshop 形式 (PSD)、JPEG、TIFF などがあります。あまり使われませんが、Photoshop で PDF を書き出して Illustrator や InDesign に貼り込むこともできます。

これらの中でもっとも実用的なのが、なんといっても Photoshop 形式です。Photoshop がもっとも扱いやすいのは、クリッピングマスクをレイアウトソフトに反映できるということでしょう。JPEG とか TIFF はできませんし、JPEG は非可逆圧縮されているのでこれからの DTP ワークフローでは扱わない方がいいでしょう。

それと Photoshop 形式では、ロスレスの可逆圧縮で保存されますので、ディテールの情報量が少ない画像ではファイルサイズは圧縮されます。

レイアウトソフトに貼り込んだとき、実画像で貼り込めクリッピングマスクに対応し、画質は劣化せずにファイルサイズは圧縮されているということ考えると、Photoshop 形式がもっとも便利なフォーマットであることがわかります。

ただし、Photoshop 形式では、レイヤーを含み描画モードや透明を適用したままの画像も貼り込むことができますが、DTP で使うには、CMYK にして画像を背景一枚に統合したものを使う必要があります。レイヤーの多いものは出力トラブルの原因になります。

DTP 環境が変化する中、画像フォーマットの標準も変わりつつあります。あと数年すると、Photoshop 形式が常識になるような気がします。そうなったら、モニタ上のソフトブルーが当たり前になって、DTP のワークフローも大きく変わるのではないのでしょうか。もっともそのやり方が本当に理解されていれば、プロファイルを埋め込んだ PDF でもモニタ校正は可能なんですけどね。

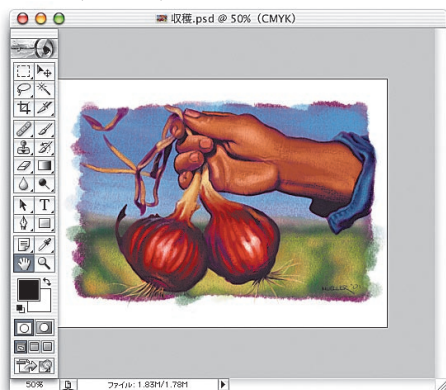
画像フォーマットにみる対応の違い

	プレビュー	実画像	可逆圧縮	クリッピングマスク
EPS(バイナリ)	○	×	×	○
EPS(JPEG)	○	×	×	○
JPEG	×	○	×	×
TIFF	×	○	○	×
PSD	×	○	○	○

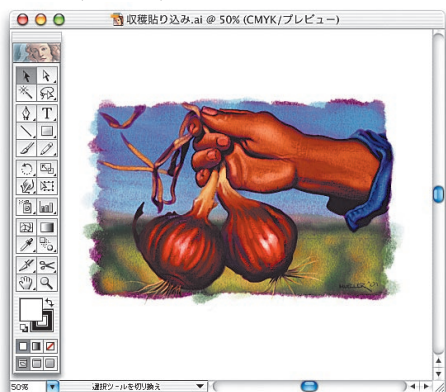
EPS(バイナリ) は画質は劣化しないが、圧縮されないで「×」になっている。実画像を貼り込め、劣化せず圧縮を行い、クリッピングパスも反映できるのは、Photoshop 形式のみとなっている。

Photoshop 形式の画像を Illustrator に貼り込む

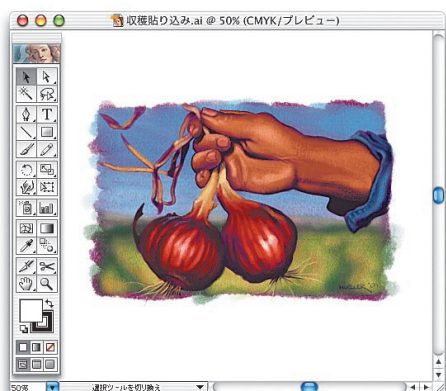
Photoshop 7.0 (Japan Standard v2)



Photoshop 7.0 (Japan Standard v2)



Illustrator 10 (Illustrator 5.5J をエミュレート)



画像を、Illustrator 10 に貼り込む。Illustrator 10 のカラー設定で「Illustrator 5.5J をエミュレート」を選択していると Photoshop と同じカラーにならないが、カラー設定の CMYK 作業スペースを「Japan Standard v2」にしていると、同じカラーで表示される。

第二回

劣化する JPEG 圧縮は 最後に適用するのが賢い使い方

ファイルサイズの大きな画像は扱いにくい

DTP の初期の頃、なんといってもやっかいなのは、画像データの扱いでした。画像データというより、画像データのファイルサイズがやっかいなのです。

画像データというのはピクセルの集まりです。CMYK モードの場合、1 つのピクセルに 4 つのチャンネルがあり、1 つのチャンネルが 256 階調ありますから、1 バイトのデータ容量になります。CMYK で 4 バイト必要となります。つまり 32 ビットというわけです。

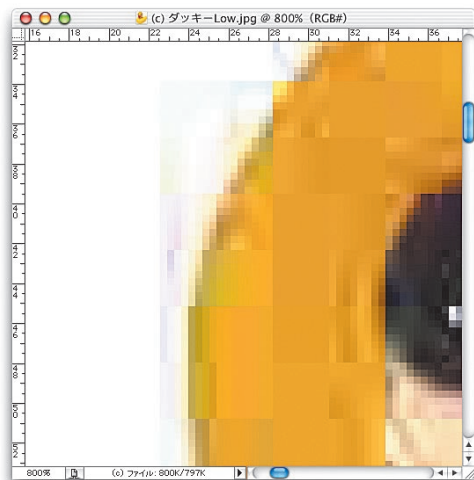
印刷では高密度の画像が要求されます。1 インチに 350ppi の解像度が必要といわれています。1 インチ平方では 122,500 ピクセルになります。データ容量としては、およそ 500 キロバイト弱といったところでしょう。これを A4 で換算すると、だいたい 40 メガバイト程度になってしまうわけです。

当時はハードディスクの容量も 1 ギガバイトを超えることはなく、DTP データの大半は画像データだったのです。画像データのファイルサイズが大きいと、ハードディスクの占有率もさることながら、ネットワークでの転送にも支障をきたすことがありました。

そこで、考えられた方法が、OPI (Open Prepress Interface) であり、DCS (Desktop Color Separation) でした。サーバに実画像をおいてレイアウトワークでは低解像度の画像を利用するのが OPI で、画像の分版を最初からしておいて、出力時に分版されたデータのみを転送するのが DCS という方法です。いずれも高速に処理するために編み出されたのです。

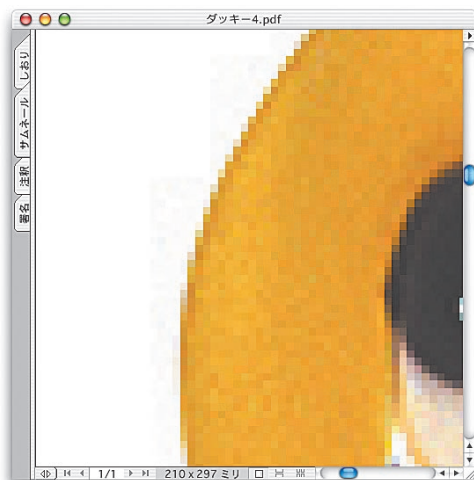
JPEG 圧縮の比較でみる劣化によるノイズ

低品質高圧縮の JPEG 画像



最低の画質 (Photoshop 7.0 で画質「0」で保存) で JPEG 保存した画像を拡大したもの。JPEG はブロックに分けて圧縮するので、低品質を選択するとブロックノイズが現れる。800%に拡大してある。

高品質低圧縮を三回重ねた JPEG 画像



Photoshop で JPEG で書き出した画像を Illustrator 10 に貼り込んで PDF 保存で JPEG 圧縮したものを、InDesign 2.0 に貼り込みさらに JPEG 圧縮して書き出したもの。すべて最高品質を選択しているが、エッジにモスキートノイズが現れている。三回重ねただけではわかりにくい、度重なるともっと目立つようになる。

カラープリンタから実画像をプリントできない OPI や DCS

とはいえ、OPI や DCS が万能というわけではありません。OPI はサーバで低解像度ファイルを作成しなければならないので、手軽に扱えるものではありませんし、DCS はカラー画像ファイルが 5 つに分割されるので、取り扱いが煩雑に

なるということがありました。

それよりも、これらのやり方はカラープリンタからプリントアウトしたときに、実画像で出力できないというのがネックでした。レイアウトソフト上には実画像がありませんから、当然実画像では出力できないのです。

DTP の環境が向上にするにつれ、カラープリンタでのカンパにより最終の仕上がりに近いものが求められるようになったのです。そのため、OPI や DCS といった方式は、時代の要求にマッチしなくなっていったのです。

フロントエンドでも実画像を扱いたいと望むのは、DTP ユーザーにとっては、当然のことでしょう。そこで、OPI でもなく DCS でもなく、高速に出力できて、なおかつカラープリンタなどでも実画像を出力できる方法として、採用されたのが JPEG を使うという方法だったのです。

———— 圧縮率が高いがオリジナル情報が失われる JPEG 圧縮 ————

JPEG はデジタル静止画像の圧縮方式をいいます。正確にはその規格を定めた委員会の名称ですが、いまではその委員会の略称がファイルフォーマット名になっています。

JPEG にはいくつかの方式がありますが、基本的には階調の変化を波形と見なして圧縮します。画像の記述をピクセルの数値で行わずに、波形の関数で表現することで、圧縮を行うのです。

画像の階調の変化と、関数化された波形は同じものではありませんので、圧縮前の完全なオリジナルに戻すことができません。そのため「非可逆圧縮」と呼ばれています。

しかし、その分高い圧縮率を誇ります。ファイルサイズをかなり小さくできるのです。また、圧縮の方法も劣化が大きくても圧縮率の高いものと、劣化が小さくて圧縮率が小さいものと使い分けることができます。劣化の小さい方式で圧縮しても、写真画像では4分の1から5分の1程度のファイルサイズになるのです。

JPEG で圧縮すると、当然画質は劣化します。低品質高圧縮するとブロックノイズが目立ち、高品質低圧縮ではモスキートノイズが発生しています。しかも不

可逆圧縮なので、もとに戻すことはできません。

しかしそれでも、画像を JPEG で圧縮するとカラープリンタでは実画像で出力できるのです。加えて高品質低圧縮を使用したときは、ほとんど劣化がわかりません。さらに出力時にネットワークに負荷をかけることもありません。

「非可逆の圧縮はいやだが、実用上問題なければ、JPEG もやむを得ない」という理解が広がって、JPEG を画像に使うようになっていったのです。

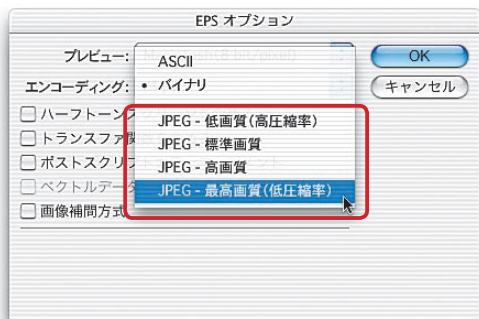
普及した「JPEG エンコーディング」形式の EPS 画像

ただしそうはいつても、実際に JPEG そのものを貼り込むのではなく、EPS フォーマットのオプションで「JPEG エンコーディング」という方式で貼り込んでいたのです。つまりファイルフォーマットは EPS でも、画像自体は JPEG になっているという形式でした。

この JPEG エンコーディングフォーマットを使うときに、このフォーマットに対応していない RIP がありました。すべての PostScript RIP は JPEG には対応していても、EPS 内の JPEG 画像を正しく扱うことができなかったのです。これらの RIP では JPEG エンコーディング画像を出力しようとすると、カラーがグレースケールになってしまうのです。

こうした中、JPEG エンコーディングに対応した RIP の普及が進みました。主

Photoshop の EPS 保存で JPEG エンコーディングを選択する



EPS 保存するとき「エンコーディング」から「JPEG」を選択すると、EPS 内の画像データのみを JPEG で圧縮できる。RIP さえ対応していれば、もっとも実用的なフォーマットといえる。

には Harlequin という会社の互換 RIP でした。もっとも Harlequin は RIP のコアだけを提供し、ベンダーが RIP を仕上げるという形になっていましたので、ユーザーが知らないだけで Harlequin RIP はよく使われていました。

過渡期には JPEG エンコーディングが出力できる RIP とできない RIP が併存し、JPEG エンコーディングの利用は評価が大きく分かれました。JPEG エンコーディングが出力できない RIP をもっているところになると、これほど忌まわしいものもなかったのかもしれませんが。いまの PDF での出力と似たような関係にあるのかもしれませんが。

———— JPEG は圧縮を重ねると劣化が進む ————

現在では JPEG エンコーディングを使用するのは「常識」になっています。もちろん必要に応じてですが、ほとんど画像データは JPEG で圧縮され出力されているといえるでしょう。

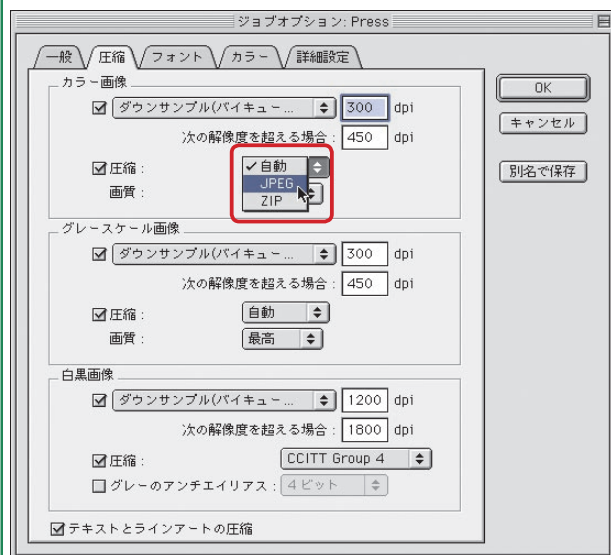
JPEG を使うときに注意したいのは、度重なる JPEG 圧縮は確実に画像を劣化させるということです。一度や二度であれば高品質で圧縮しても、ほとんど劣化は目立たないでしょう。しかし画像データを再度利用すれば、劣化の度合いは進むのです。そういうことを考慮せずに多用していると、あるとき気が付くとノイズにまみれた画像を使わなくてはならないことになってしまいます。

できればレイアウトソフトに貼り込むオリジナルの画像は JPEG を利用せず、ロスレスの画像フォーマットを利用する方が間違いありません。以前であれば、ロスレスの実画像はハードディスクを占有するだけの悪者でしたが、いまではハードディスクの容量も数十ギガバイトが普通で、もう少しするとテラバイトにも手が届くかもしれません。ですから、レイアウトソフトに実画像を貼り込んでも、必ずしも DTP ワークの妨げになるわけではありません。

できれば、最終出力のファイルフォーマットを PDF にし、PDF 化するときに一気に画像を圧縮するのがいいでしょう。そうすれば、圧縮しないオリジナル画像も残りますし、PDF 作成時に圧縮の度合いをコントロールできるからです。

これからインターネットなどでデータをやり取りすることが多くなるに違いありません。電子送稿の時代になると、JPEG をかけないわけにはいきません。イ

Distiller で画像を JPEG 圧縮する



最終の出稿ファイルフォーマットで PDF を選択し、PDF 作成時に JPEG で圧縮すると、圧縮率が高く高品質な JPEG 圧縮ができる。Illustrator や InDesign で保存する場合も保存時にのみ JPEG 圧縮するほうがよい。ただし、Illustrator 9.0 や 10.0 では不具合があるので JPEG は使わない。

インターネットでの入出稿でも PDF が利用されるでしょう。そうなりとやはり、JPEG の圧縮は PDF 保存時に適用するのがもっとも適切なやり方といえそうです。

第三回

フォントのダウンロードは PDF のエンベットの仕組みで

制限の多い日本語フォントはトラブルが多い

フォントといえば、私は「プロテクト」という言葉をすぐに思い起こしてしまいます。DTP で利用するフォントにはなにがしかの「プロテクト」がかかっていることがよくあります。

なぜプロテクトがかかっているのかというと、「ユーザーは信用ならない。プロテクトしておかないと、不正利用するからだ」ということがあります。不正利用されると、フォントビジネスが成り立たないので、プロテクトをかけるしかないということはあるでしょう。

DTP 初期の頃は、DTP で利用できるフォントも少なく、フォントの財産を守るために「プロテクト」が必要だったといえるでしょう。「プロテクト」には、アウトラインプロテクト、インストールプロテクトとありますが、プリンタフォントが必要なのも、「出力プロテクト」といってもいいでしょう。

フォントベンダーにとって、プロテクトは必要不可欠のものでしたが、ユーザーにとってはフォントの扱いを複雑にしトラブルを招く原因になるものでした。なんといっても、出力トラブルの中で、フォントに起因するものももっとも多いのですから。

日本語フォントでは TrueType フォントは RIP から出力できず、PostScript フォントでもプリンタフォントがなければ出力できません。また、欧文フォントは同じフォント名なのにフォントフォーマットが違ったりバージョンが違ったりして正しく出力できないことがありました。つまり、出力環境を考慮せずにデータを作成すると、モニタ表示と同じようには出力できないわけです。また、不用意にフォ

トフォーマットを変更すると、外字部分が文字化けしてしまうという問題もありました。

低くなるフォントのプロテクト

ただそうはいっても、日本語 DTP が始まって十年以上が経ちました。DTP で利用できるフォントが増え、フォントも大競争時代にはいりました。プロテクトの垣根は少しずつ低くなり、いまではほとんどなくなりつつあります。アウトラインプロテクトがなくなり、最近ではインストールプロテクトもついていない PostScript フォントもあります。そして、ダイナミックダウンロードすることでプリンタフォントも不要になりつつあります。

プリンタフォントを必要とせずに出力する方法は、必ずしもダイナミックダウンロードだけではありません。広い意味でいえば、フォントをアウトライン化して出力することや画像化して出力する方法もあります。

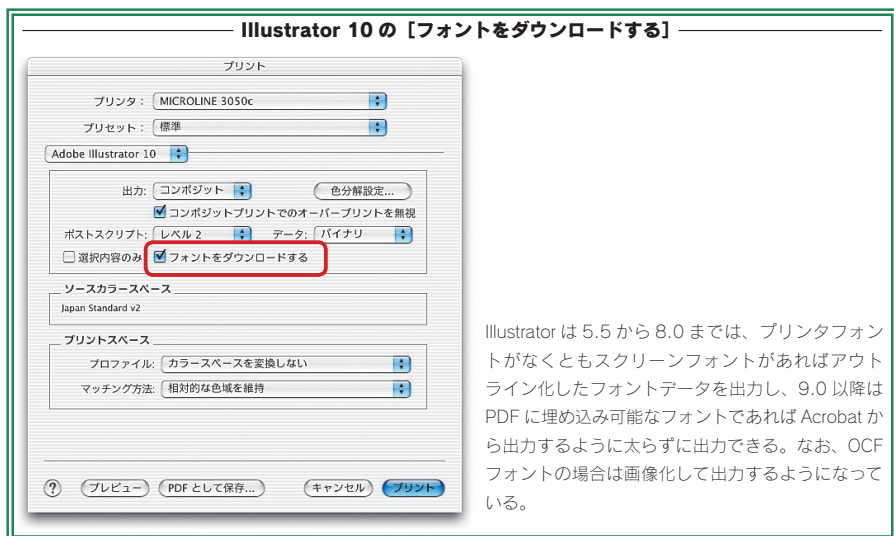
そういえば、かつての CEPS はフォント部分は画像化して処理していました。最近の RIP でも内部的には PDF で処理しているにもかかわらず、PDF 内の文字をアウトライン化して利用するものもあります。画像化したりアウトライン化しても、データ処理の能力が高ければ、実用上はネックにならないということでしょう。もちろん CEPS や高品位な RIP では、PostScript ファイルから画像化したりアウトライン化したりするとき、プリンタフォントを利用していました。プリンタフォントという出力のプロテクトは不要というわけではありませんでした。

しかし、出力する前に、同じように画像化したりアウトライン化したりすることで、フォントの制限はほとんどなくなるのです。フォントの制限がなくなれば、フォントにまつわるトラブルも少なくなるといえるでしょう。

アウトライン化すると太って出力される

アプリケーションソフトで、プリンタフォントを利用しないでフォントを処理する方法はいくつかの方法があります。

Illustrator では、かつてバージョンが 3.2 のときに ATM フォントを画像化し



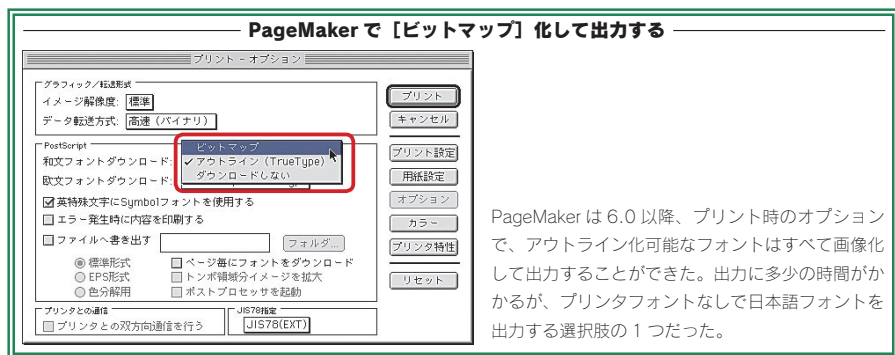
て出力する仕組みがありました。Illustrator が日本語フォントを画像するので、この方法を利用すると高解像度で出力できました。ところが、5.0 にバージョンアップしたときにこの機能は失われ、Illustrator はプリンタフォントに依存しなければ日本語フォントは出力できなくなったのです。

5.5 になったときユーザーの要望で、[フォントをダウンロードする] の機能が追加されました。これは 3.2 とは違い、日本語フォントをアウトライン化して出力する方法です。アウトライン化すると、600dpi 程度の低解像度プリンタでは、フォントが太るという現象が occurred。アウトライン化すると、PostScript の描画形式では輪郭線上のドットをすべて塗りつぶすので、低解像度プリンタでは太って見えてしまうのです。

とはいえ、2400dpi 以上の高解像度で出力するとほとんど見分けが付きません。いまでも Illustrator のデータは、フォントをはアウトライン化することで出力されています。おそらくみなさんも、Illustrator データ出力の「常識」として活用されているでしょう。

アプリケーションや OS でするフォントのラスタライズ

また、PageMaker や QuarkXPress では、スクリーンのアウトラインフォントを



PageMaker は 6.0 以降、プリント時のオプションで、アウトライン化可能なフォントはすべて画像化して出力することができた。出力に多少の時間がかかるが、プリンタフォントなしで日本語フォントを出力する選択肢の 1 つだった。

画像化して利用することができます。ATM の描画データを利用してそれを元にフォントを画像化します。昔は、フォントを画像化すると、出力に時間がかかりましたが、いまでは CPU の処理速度が向上し、ネットワークも高速化していますから、画像化してもそれほど出力にストレスを感じるわけではありません。

ただし、QuarkXPress 3.3 の ATM 出力では文字送りが正確ではなく、カンパ程度のものであればともかく実質的には使い物になりません。3.3 よりも 4.x の方がましですが、残念ながら最終出力に耐えうる品質は期待できません。

PageMaker ではモニターで見たとおりの出力が可能です。こちらの方は文字送りが狂うこともなく、正確なプリンタ出力が可能です。PageMaker には簡単な面付けの機能もあり、軽オフセットの業界では、この機能を版下出力に利用していたところも少なくありませんでした。

また、TrueType フォントについては、Macintosh も Windows も OS がフォントを PostScript Type42 フォントなどに変換してダウンロードする機能を持っていましたが、これらの機能は DTP アプリケーションでは利用できず、Windows の MS Word などの出力で利用されているだけでした。

PDF でするフォントのダイナミックダウンロード

プリンタフォントに依存せず、フォントを出力する方法にはいろいろな方法がありますが、最近ではフォントを画像にもせず、アウトライン化もせず、「ダイナミックダウンロード」という方法で日本語フォントをそのまま出力する仕組みが普及しつつあります。

たとえば、Illustrator 9.0 からは「フォントをダウンロードする」と、フォントが太らずに出力できるようになりました。これは今までのようにフォントをアウトライン化せずに、別の方法で処理しているからです。

Illustrator 9.0 ではドキュメントのファイルフォーマットが PDF になり、PDF のフォントエンベットの機能を利用できるようになったのです。そのため、PDF で埋め込みフォントやスクリーンフォントをダウンロードするように、プリンタフォントを全く必要とせず出力できるようになったのです。Acrobat や Illustrator だけでなく、InDesign でも同じようにフォントをダウンロードすることができます。

PDF のフォントのダウンロードは、RIP でフォントを処理する方法の一部をアプリケーションで行ってしまうものです。RIP での PostScript フォントの出力では、プリンタフォントからフォント辞書というものを作成して行います。しかし、このフォント辞書の作成を、PostScript ファイルを作成するときにアプリケーションが行ってしまえば、プリンタにフォントは不要になります。

PDF ネイティブの RIP の場合は、PDF に埋め込まれたまま出力されます。PDF に埋め込まれたフォントもフォント辞書によって記述されています。PostScript のフォント辞書と PDF のフォント辞書は全く同じものではありませんが、いずれにしてもプリンタフォントに依存せずに出力するため、グリフデータを含むフォントのすべての情報が含まれています。

ダイナミックダウンロードが常識になればフォントトラブルは減少する

PDF のフォントエンベットの（埋め込み）での出力は、まだまだ一般的にはいえないかもしれません。しかし今後利用が増えていくことも確かです。

出力の安定性・安全性という点では、信頼を勝ち得ていない部分もあります。古い RIP では、PDF の埋め込みフォントが出力できないこともあるのです。ですから、Illustrator でアウトライン化したり、QuarkXPress でプリンタフォントを利用して出力した方が勝手がわかっていて扱いやすいということはあるでしょう。

とりわけ、安価な TrueType フォントでは、扱い方によっては文字が化けたりすることがあります。たとえば、Distiller で作成した PDF を InDesign に貼り込んで、もう一度 PDF 化すると、フォントによっては文字化けする場合があります。

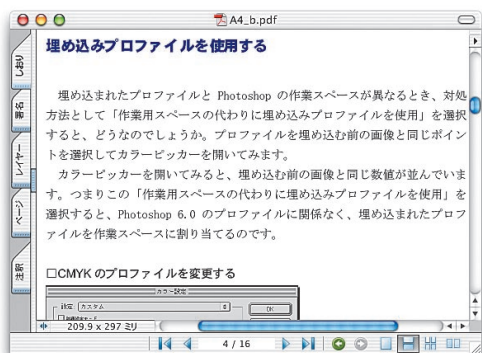
す。PostScript CID フォントではそのようなことはありませんが、フォントの埋め込みに際に要求されるスペックを完全に満たしていない TrueType フォントもあるでしょう。

アウトライン化したり画像化したりするより、PDF にエンベッドして出力する方が、出力ドキュメントは扱いやすく、ファイルサイズも小さくなります。一度使い慣れれば、PDF にしてフォントを埋め込むという便利さから後に戻ることはできません。

フォントの利用を制限せず、自在に利用し、そのまま出力するという点では、間違いなく PDF のフォントの埋め込みが「常識」になるでしょう。それができれば、DTP の出力トラブルで、フォントが制限されていたことで引き起こされる数多の問題は少なくなるのではないのでしょうか。

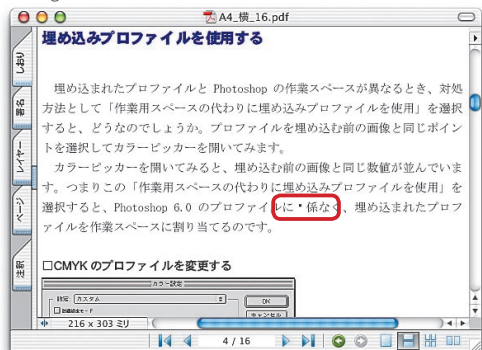
文字化けした Windows の TrueType フォント

オリジナルの PDF



MS Word から Distiller 5.0 で PDF にしたときは、正しい PDF が作成されている。

InDesign に貼り込んで書き出した PDF



MS Word から Distiller で PDF にし InDesign に貼り込んで、さらに PDF 書き出したもの。MS 明朝の日本語の文字の一部が化けている（本文下から二行目の「関」の文字）。PostScript CID や OpenType フォントを利用したときは、InDesign で PDF 書き出しを行っても文字化けすることはない。なお、Distiller 6.0 で作成した PDF ではこの現象は発生しない。

番外編 「InDesign ソリューションセミナー」を レポートする

去る 4 月 15 日 (2003 年) に大崎ゲートシティホールでアドビシステムズ主催の「InDesign ソリューションセミナー」が行われた。ここでは番外編としてこのセミナーのレポートをお届けする。

このセミナーは「ソリューションセミナー」と銘打たれているように、簡単にいうと「InDesign は何ができるか」というテーマのセミナーだ。今回のキーワードは「XML」で、InDesign は「XML」を利用することで、自動組版処理が簡単にできてしまうという内容である。

文書構造がわからないと利用できなかった XML

印刷業界はここ数年、文書処理の方法として SGML や XML を取り込もうとしてきた。しかし SGML も XML も文書型定義を決めるという工程があり、誰でも簡単に利用できるものではなかった。ドキュメントの構造を決めて、タグを独自で作成しなければならないのだ。XML は SGML に比べて自由度が増し、タグ付けが簡単になったとはいえ、実際にはどのような用途でマルチユースするのかを決めないと XML 文書の構造をフィックスできないのである。

また、もともとこうした文書を標準化するフォーマットは、マルチユースできなければ意味がないが、いままでは、マルチユースしなければならないものが限られており、一部の大企業や官庁などでしかその需要がなかった。

そうした中で、印刷業での XML は話題になることはなっても、実務に反映されることが少なく、ハイタッチという視点ではまだまだ十分なソリューションとはいえなかったのである。

しかしもともと XML とは、その仕組みを一般のユーザーが理解して XML タグを入力する利用するものではなく、XML を活用するソフトウェアがバックグラウンドで動作し、それをユーザーが利用するものでなければならない。CPU や OS の仕組みがわからなくてもパソコンが利用できるように、XML の文書構造を知らなくても、XML を使ってできることがあればそれでいいのである。

フィードバックが難しい AppleScript の自動処理

DTP で XML を利用するとき、考えないといけないことには、まず、自動処理がある。XML を使って、自動処理での分野でいままでのシステムより抜きんでいなければ、既存のシステムから移行するだけのメリットは少ない。

現在の DTP の自動組版処理は、QuarkXPress 3.3J をベースに AppleScript で動作させるもののがもっとも多い。定型のフォーマットに対して AppleScript でテキストや画像を貼り込んでいくわけである。

AppleScript を利用するメリットは、なんといっても、長年蓄積されたノウハウがあり、安定した動作を見込めることにある。誰にでもできるというわけではないが、定型の大量のページを処理するのであれば、もっとも現実的な方法だろう。また、QuarkXPress のプラグインなどを利用して自動処理を行うケースもある。

ただ、こうした方法のネックは最終的にレイアウトソフト上で訂正を加えることになったとき、オリジナルのデータベースにそれを戻すことが難しいことである。本来であれば訂正はオリジナルのデータに行い、再度自動処理を行うべきかもしれないが、納期に追われる場合が多く、現実的にはレイアウトソフト上での訂正となることもしばしば起こることである。

プログラムの改変とデータベースとの連携

現状の自動組版処理、バッチ処理のボトルネックになっているもっとも大きな課題は 2 つあると考えられる。

1 つは自動処理プログラムの改定の容易さである。レイアウトを新しいフォー

マットにしたとき、Script を変更しないとイケないのであれば、それに精通した技術者が必要になり、外部に発注すると、開発費が結構かかるということである。

いままではそういった開発費が発生しても、それによってコストダウンが図れる印刷物だけがその対象になってきていた。しかし、開発費のコストが下がれば、さらに自動処理可能な印刷物が増えることは間違いない。高度なシステムを組んだり、プログラムを書き直す必要がなくなり、DTP のオペレータが簡単に扱うことのできる仕組みが希求されていることは確かなのである。

もう1つは、データベースとの完全な連動性である。つまり、レイアウトソフト上での変更が確実にデータベースに反映される仕組みである。レイアウトソフト上であれ、Web 上であれ、書き直したデータがすぐさますべてに反映され、なおかつその履歴が残るようなシステムであろう。

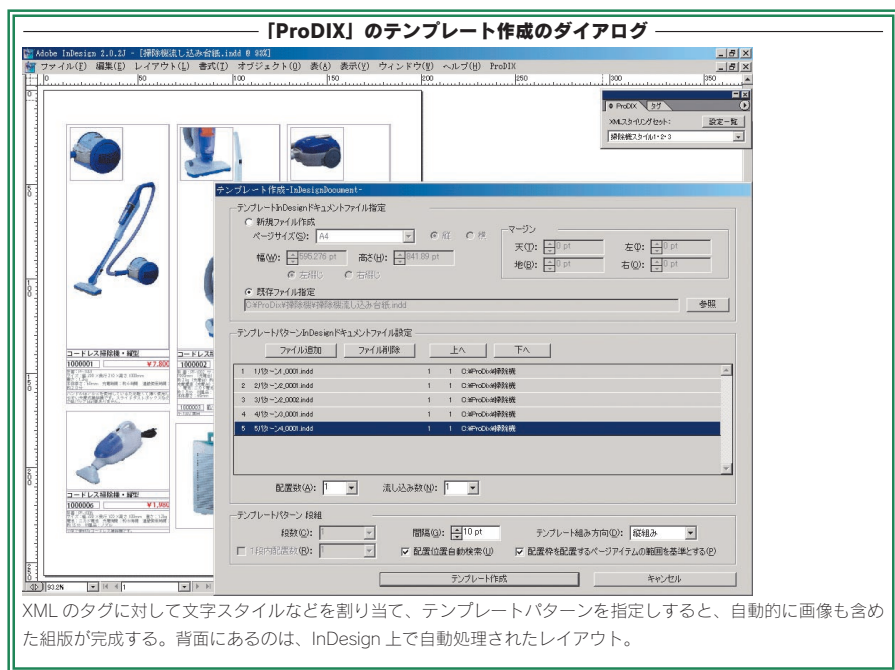
これらの課題を解決したときに、DTP の自動組版処理は次のステップに進むと考えてよい。

———— InDesign に CSV を読み込み XML で自動組版する ————

「InDesign ソリューションセミナー」で提示されたものは、XML を利用した自動組版処理のソリューションであった。プロフィールド [<http://www.profield.jp/>] の「ProDIX」という製品で、エクセルなどから生成した CSV (コンマ区切りテキスト) ファイルから XML ファイルを生成する「ProXCreator」と、InDesign 上に取り込んだ XML ファイルで組版処理する「ProDIX」というプラグインから構成されている。

つまり普通のデータベースソフトで管理されたデータにタグ付けした XML ファイルを作成して、それを InDesign 上で、文字スタイルや位置情報、またプライオリティやレイアウト時の条件分岐処理を行いつつ、自動組版を行うのである。

定型レイアウトであれば、レイアウト要素を一つ一つ項目にわけてエクセルやファイルメーカーなどに入力し、それを項目ごとにタグ付けし、XML ファイル化するだけでよい。あとは、テンプレートに流していけば、自動組版処理が簡単にでき上がるというわけだ。



XML のタグに対して文字スタイルなどを割り当て、テンプレートパターンを指定すると、自動的に画像も含めた組版が完成する。背面にあるのは、InDesign 上で自動処理されたレイアウト。

ベースは XML なので、レイアウト上で直した情報をもう一度 XML ファイルに戻すことも簡単で、データベースの一貫性も維持できるのである。

最近は QuarkXPress でも、バックグラウンドで XML を利用したソリューションがあり、技術的には自動組版処理に XML を活用することは実現されている。ただし、そういうソリューションはワークフローに合わせてカスタマイズされたものであるため、一般に数千万円単位の開発フィーがかかることは珍しくない。

もちろん、自動組版処理のシステム開発に数千万円かかっても、印刷物の売上が年間で数億円とか数十億円あるものであれば、制作費のコストダウンでその費用は十分回収でき、その上制作にかかる納期も短縮されるのである。

単にレイアウトを自動処理するだけであれば、導入費用が一桁違う「ProDIX」などで自動組版処理を行うことができるようになれば、DTP の自動組版処理の世界は大きく変化するような気がする。つまり、いままでボリュームがないばかりに自動組版処理できなかった印刷物がさらに省力化・合理化されていくのである。

XML データのマルチユースを提案する

当然そうすると、自動組版処理できるものの価格はさらに下がっていく可能性はある。また、「ProDIX」以外にも XML を活用したソリューションが現れれば、自動組版処理のマーケットは確実に XML に移行していくに違いない。

すぐにそうなることはないにしても、「ProDIX」は特殊な技能を持った技術者でなくても、オペレータレベルで導入できることがミソで、予想するより普及が早い可能はあるだろう。数年前に比べれば、印刷会社のフットワークは軽くなり、印刷という業態に縛られなくなっているのだから、自動組版処理可能な印刷物を抱えている場合は、具体化しやすいように思う。

そうして導入が進むと、またまた競争が起こることになるだろうが、そのときに問われるのが、XML データをどのように活用するのかということではないか。本当の意味でのマルチユースを果たすためには、データベースの活用を既存のメディアに収めるのではなく、今までは違ったメディアを創出するという試みも必要ではないか。

今回のセミナーでは、事例として太平印刷のカタログが紹介されていた。あるカタログで1つのXMLデータから、HTMLと組版データへのマルチユースでの利用である。XMLを利用すると、HTMLと組版データへのマルチユースがいとまたやすいことがよくわかる。

いまのところは、それでも十分であるにしても、いずれもっと多様なマルチユースが要望されるように思うのだ。おそらくデータベースのマルチユースだけでなく、さらに踏み込んでデータベースそのものの活用まで提案していくようになっても不思議ではない。つまり、印刷会社といえども「データベースコンサルティング」といったスタンスで取り組む時代に移って行くように思うのである。

データベースのXML化というソリューションが、Macintosh上で使われるようになったのは、Mac OS XというUNIXをベースとした堅牢なOSが登場したことに密接な関係がある。おそらく、いままでのMac OSではデータベースを本格的に運用することは難しかったかもしれない。そういう意味では、InDesignとMac OS Xで時代は大きく変わりつつあるといってもいいだろう。

本書をもっと理解するための参考書籍

インクナブラの書籍



■ 透明効果や Illustrator の PDF 保存を知るためには

Illustrator は保存するだけで PDF が作成できるが、PDF の作成には注意が必要だ。とりわけ、9.0 以降の透明効果や書類のラスターライズ効果などの新機能は、正しい扱いが必要となる。『失敗しない Illustrator 出力講座』は、Illustrator 8.0 から CS までの、プリプレス出力するために知っておきたいことと EPS や PDF 保存についてわかりやすく解説している。Illustrator 使いは是非手元に置きたい 1 冊だ。



■ 確実に出力できる PDF を作成するためには

そのままでは RIP から正しい PDF 出力ができないことがある。RIP の機能に依存する部分があると予想した出力ができないのである。しかし Acrobat 6.0 Professional を利用すると、PDF をリメイクし確実に出力可能な PDF に作りかえることができる。本書では PDF 内のブレンドをスムーズシェーディングにしたり、フォントをアウトライン化することで、RIP に依存しない PDF の作り方を解説する。



■ アプリケーションでカラーマッチングの仕組みとは

Adobe 社のアプリケーションの最近のバージョンは、Adobe Color Engine という仕組みが内蔵され、アプリケーションでカラーがマネージメントできるようになっている。Photoshop で開いて確認したカラーを、Illustrator や InDesign、Acrobat などでも同じカラーで表示し、印刷時のカラーに合わせてブルーフの出力も可能だ。本書では Adobe Color Engine の仕組みを詳細にかつ具体的な解説している。



■ DTP を Mac OS X 環境に移行するためには

Adobe Creative Suite や QuarkXPress 6.0 など、DTP でも Mac OS X でしか動作しないアプリケーションが増えてきた。いずれ Mac OS から Mac OS X への移行は避けられない。『Mac OS X で DTP はどう変わる・改訂新版』では、Mac OS X での DTP を Classic 環境での DTP、ネイティブ DTP アプリケーションで DTP、そして Mac OS X の Quartz での DTP に分類し、Mac OS X での DTP の現実的な対応を探る。

あとがきにかえて

DTP の環境は進化したかという、これがなかなか進化してしない。むかし、『DTP 出力講座 (翔泳社)』という本を書いたとき、Illustrator は 5.5.J で QuarkXPress は 3.3J がスタンダードだった。1997 年に書いたものののだが、DTP 環境はいまでもそれほど大きく変わっていない。

アプリケーションのバージョンアップが激しく、こうした解説本は陳腐化が激しいにも関わらず、DTP のスタンダードバージョンが進まないために、予想より延命した。Illustrator や QuarkXPress のバージョンが上がっても、InDesign がリリースされても、ユーザーの多くはそれらを使わなかったのである。

こうしたバージョンアップや新しいアプリケーションが利用されないのは、DTP が成長期を過ぎてしまったためだろう。DTP のユーザーが増えようとしてつあるときは、新しい技術は積極的に受け入れられるが、成長が止まり成熟期になると、作り上げられたスキームを変革する力が失われてしまう。そのため、積極的に新しいものにシフトしていくことがなかった。

また、バージョンアップしたアプリケーションにはさまざまな新機能が追加されたが、それらは DTP にとって必ずしも必要なものではなかった。新機能は十分条件であっても、必要条件ではなかったのだ。

しかしだからといって、DTP 環境が変わっていかないというわけではない。新しいものを取り入れていくためには、大きなエネルギーが必要で、新しいものに十分な魅力がなければならないが、必要性があれば変わっていくことも確かである。

必要性は DTP 業界の内部から発生するわけではなく、世の中が変わっていくことで、DTP にも新しい側面が求められるためだろう。世の中が変貌すること

で、DTP もまた変身するのである。

従来の写植・製版をリプレースするだけの価値しかないのであれば、Illustrator は 5.5J、QuarkXPress は 3.3J でも十分かもしれないが、時代はそれを許さなくなっている。

たとえば、DTP を変革するものにカラーマネジメントがある。カラーマネジメントはここ数年でスキームが完成しつつあり、実用的なレベルに達している。ここでいうカラーマネジメントは、DTP というプリプレスでのカラーマネジメントではなく、その後工程にあるプレス、つまり印刷でのカラーマネジメントである。

印刷でのカラーマネジメントの確立はなにを意味するのかというと、これは、印刷のさらなるコストダウンなのである。プリプレスからプレスまでのカラーマネジメントを行うことで、色校正という工程をスキップし、コストダウンを図るのである。これらはモニタでのソフトプルーフだけでも、印刷時のカラーをコントロールして印刷可能なのである。

もちろん、すぐさま色校正が無くなるということはないだろう。平台の校正は少なくなっていくにしても、DDCP やカラープリンタでのハードプルーフは当然必要である。

しかし、今までオペレータが色校正紙を見ながらカラーを合わせていた職人技の「印刷技術」が、数値化されることで、職人技的スキルの価値は確実に減少する。それによって安定して低コストの印刷が実現するのである。

印刷時にカラーマネジメントを行うには、アプリケーションでデータメイキングする際にもカラーマネジメントが必要になる。カラーマネジメント可能なア

アプリケーションとバージョンを利用しなければ印刷のコストダウンは果たせないことになる。つまり、印刷機のカラーマネジメントが普及することが、DTP 環境を変貌させる推進力となるのである。

また、もう1つは PDF が出力用フォーマットとして認知されることにある。PDF での出力はほとんど問題のないレベルになってきており、運用時の問題がまったくないわけではないが、すでに現実利用されている。

しかし PDF が DTP ユーザーによって選択されるのは、電子入稿が否応のない現実となるからだろう。いままでように MO や CD といったメディアからインターネットなどを経由して電子送稿するのが当たり前になれば、アプリケーションのネイティブ形式のファイルは扱いづらい。ファイルサイズが大きくなるだけでなく、複数のファイルを圧縮して解凍する手間も必要となる。

電子送稿に最適化された PDF では、1 ファイルにまとめることができる上に、画像もフォントも埋め込んでおけば、デバイス・インディペンデントになり、出力機のフォント環境などに依存せずに出力することが可能になるのだ。PDFこそが、電子送稿時代のファイルフォーマットの最右翼となる。

カラーマネジメントされたデータを PDF 化して電子送稿すれば、入稿して印刷するまでに必要としていたコストの多くが削減される。DTP ももともと、写植・製版のコストダウンから導入されたように、カラーマネジメントや PDF もコストダウンという果実があれば、間違いなく浸透していくのである。

そうして変貌する環境の中で DTP に関するさまざまな事柄をテーマにしたものが本書である。画像のファイルフォーマットやフォント、透明効果や PDF の出力、そしてアプリケーションでのカラーマネジメントなどの現状とこれからの行

方を取り上げた。

本書はもともと「未来の見える DTP 進化論」というタイトルで、ジーエーシティの Web サイトに連載したものに加筆修正したものである。ジーエーシティの萩原和夫さんからお誘いを受け書き始めたもので、2003 年の 5 月から 12 月までアップしたものをまとめてある。ともすれば予定が遅れる私が最後までたどり着けたのは、柔和で粘り強い萩原さんと編集を担当された加藤小波さんの気遣いのおかげである。お二人にはここで改めて感謝申し上げたい。

incunabula で



LETS

Leading Edge Type Solution

を お申し込みされる方に

フォントワークスジャパンの包括的フォントサポートプログラム『LETS』へのご入会は、いずれの販売代理店からご入会いただくことができますが、インクナブラよりお申し込みいただくと、フォントワークスジャパンのサポート以外にさまざまな特典があります。

● DTP-S 倶楽部にそのまま入会 ●

DTP-S 倶楽部は、上高地仁の DTP-S 内にある会員制の情報サイトです。DTP の関する新しい話題や Tips など、いち早くお届けする有料のコンテンツサービスです。DTP-S 倶楽部にご入会いただくと下記のような特典があります。この DTP-S 倶楽部は有料の会員制を取っていますが、インクナブラより『LETS』にお申し込みいただくと、LETS 入会期間が同時に DTP-S 倶楽部への入会になります。

会員のみがアクセスできる Web でのコンテンツ	メールマガジン「DTP-S 倶楽部 BCC マガジン」の配信	ペーパーマガジン誌 『DTP オカジオネル』の配布	インクナブラ発行の書籍や 素材集のご優待
会員のみがアクセスできるページを用意しています。一般に公開されないコンテンツがアップされています。また、会員のみがダウンロード可能な PDF などのコンテンツも用意しています。	「DTP-S 倶楽部 BCC マガジン」は、インクナブラより書籍化されるコンテンツや DTP 倶楽部のみの記事をメールマガジンで配信します。表の DTP-S では書けない話題もあります。	Web ページやメールマガジンに掲載したコンテンツをペーパーマガジンとしてお届けするものです。『DTP オカジオネル』の一部は、DTP-S よりフリーコンテンツとして配布されますが、完全版を手にとってごらん戴けます。予定では隔月もしくは季刊程度で配布します。	DTP-S 倶楽部にご入会いただくと、インクナブラ発行の書籍・素材集・ソフトなどがご優待価格でお申し込みいただけます。また、インクナブラ主催のセミナーにもご優待価格で参加していただけます。

● ご入会方法 ●

■すでに LETS の見積りをされた方

LETS の見積りをされ、お申し込みされるだけになっている場合は、「LETS 見積書兼申込書」をインクナブラまでお送りください。また同時に、ライセンス契約書をフォントワークスジャパンにお送りください。フォントをより早くお使いになりたいときは、「LETS 見積書兼申込書」を先に FAX いただくと、製品をお届けするのがより早くなります。

■ LETS エントリーバックをお持ちの方

エントリーバックをお持ちの場合は、「LETS 見積依頼書」に必要事項をご記入の上、インクナブラもしくはフォントワークスジャパンまで FAX してください。折り返し、お見積りの内容とライセンス契約書をお届けいたします。すでにエントリーバックをお持ちの場合は、ライセンス契約時には入会金は不要です。

■これから LETS の見積りをされる方

いまだ LETS に入会される場合は、ライセンス使用料以外に入会金（30,000 円）が必要となります。お急ぎの場合は「LETS 見積依頼書」をインクナブラまでお送りください。もし、多少時間があるのであれば、エントリーバックをお買い上げいただき、そのシリアル番号を「LETS 見積依頼書」にご記入いただくと、入会金は不要になります。インクナブラで LETS エントリーバックをお申し込みいただくと、インクナブラの書籍を一冊プレゼントいたします。

■すでに LETS に入会されている方

すでに LETS に入会されていても、年ごとの更新時に販売会社を変更することができます。更新時にインクナブラで更新していただくことで、インクナブラの特典が受けられます。詳しくはインクナブラまでお問い合わせ下さい。

使用したフォント

FOT- 筑紫明朝 Pro

FOT- マティス Pro

FOT- ロダン Pro

FW- ニューロダン Pro

進化する DTP、明日はどっちだ

発 行 2004 年 3 月 31 日 初版発行

著 者 上高地 仁

発行人 田中 清

発行所 有限会社 インクナブラ

〒 540-0025 大阪市中央区徳井町 2-2-11 LM 東本町第三 405 号

TEL:06-6966-4468

FAX:06-6966-4469

製 本 石川特殊特急製本株式会社

©2004 by Jin Kamikochi

本書の内容の一部もしくは全部を著作権法の定める範囲を超え、有限会社インクナブラおよび著者に無断で複製、複写、転載することはご遠慮下さい。

本書の内容に関するお問い合わせもしくは質問は、E メール（incun@incunabula.co.jp）もしくは文書で、小社までお問い合わせください。

本書は Mac OS X 10.3 環境の InDesign CS でレイアウトしたものを、データ書き出しで PDF 化したものです。その PDF を InDesign に貼り込み、さらに PDF 化して印刷したものです。

インクナブラの書籍について

インクナブラでは、従来の出版の枠を越えて、少部数で書籍を発行する方法を模索してきました。既存の書籍流通では、大手の出版社と書籍流通の間で書籍を発行する流れができ上がっています。しかし、その仕組みの中では発行できないが、書籍にしたいものが多いです。

読者の興味はパーソナライズされ、より詳しく専門的な知識を要求するようになってきています。書籍の内容が専門的になるにつれ、その書籍を必要とする読者の数は少なくなります。書籍が少部数化する流れはこれからも変わっていかないでしょう。

そうした中で、個人の持つ情報はインターネットを通じて広く公開され、専門的な情報といえども、インターネットを通じて手に入れることができるようになってきました。

しかし、インターネットでは情報のプライオリティが明確ではなく、また、整理され分かりやすい情報になっているのかというと、必ずしもそうではないでしょう。

的確に情報を伝達するには、やはりまだまだ「紙」というメディアの方が分かりやすく理解しやすいのではないのでしょうか。そのためには、今までとは異なった方法で書籍を発行することしかありません。

従来の書籍発行の枠を越えて、さらに少部数で発行することができれば、出版のあり方も変わってくるでしょう。しかしだからといって、書籍をオンデマンドで発行することを読者が望んでいるわけではありません。いままでと同じような書籍として形で発行しながらも、少部数に対応した出版こそが求められているように思います。

インクナブラでは、少部数の読者ニーズに応えるために、さまざまな方法で書籍を制作し発行しています。少部数に対応するために、従来の書籍と比較してページ数が少なかったりします。しかし、300 ページある書籍でも必要な箇所はたったの 10 ページであることも少なくありません。必要なページがたった 10 ページであれば、総ページ数が 300 ページであっても 100 ページであっても、読者にとってはそれほどの違いはないのではないのでしょうか。

皆さんが手にしたこの書籍のうちの数ページでも、皆さんにとって必要かつ有用な情報であることを望んでいます。

インクナブラ代表 上高地仁