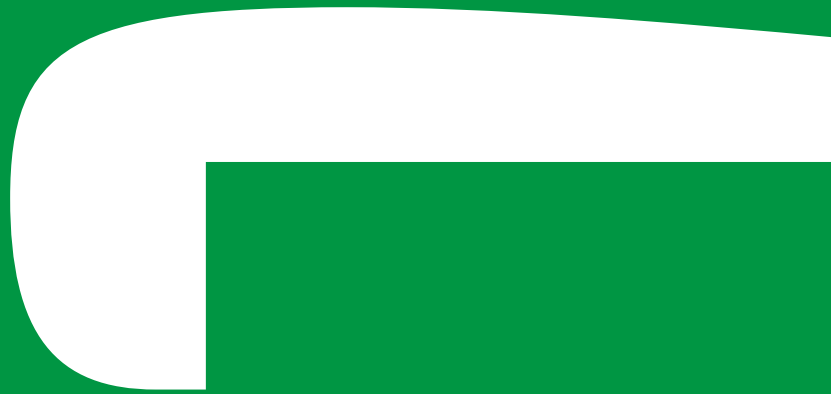


ハンドブック2016年11月版



Glyphs 2.3

デザインからリリースまで統合する
万人のためのフォントエディター

本書はGlyphsバージョン2.3用のハンドブック2016年6月版（英語）を
日本語翻訳したものです。

最新版は以下のウェブサイトダウンロードしてください。

glyphsapp.com/get-started

©2011-2016 glyphsapp.com

原著執筆：Rainer Eric Scheibelbauer、Georg Seifert

協力：Jeff Kellem、Rob Keller、Claus Eggers Sørensen、大曲都市

日本語版翻訳：大曲都市

日本語版協力：照山裕爾、丸山邦明

目次

1	Glyphs	8			
1.1	OpenType フォントを作るためのツール	8	3.3.11	アウトラインの再接続	24
1.2	インストール	8	3.3.12	パス方向の管理	24
1.2.1	必要インストール環境	8	3.3.13	極点と変曲点	25
1.2.2	インストール	8	3.3.14	重なったポイント	26
1.2.3	製品版へのアップグレード	9	3.4	アンカー	26
1.3	コミュニティ	9	3.4.1	合成用と位置合わせ用アンカー	26
1.4	アップデート	9	3.4.2	リガチャー用キャレット	27
1.5	Glyphs Mini	9	3.4.3	アンカーの追加、編集、削除	27
1.6	キーボードショートカット	9	3.4.4	ベースグリフ + 発音記号の位置合わせ	28
1.7	クラッシュレポート	10	3.4.5	発音記号 + 発音記号の位置合わせ	29
			3.4.6	連綿の位置合わせ用アンカー	29
2	環境設定	11	3.5	ガイドライン	29
2.1	環境設定の開き方	11	3.5.1	スマートガイドライン	29
2.1.1	アップデート	11	3.5.2	ローカルガイドラインとグローバルガイドライン	30
2.1.2	ユーザー設定	11	3.5.3	グリフごとの取り消し履歴	31
2.1.3	サンプルテキスト	13	3.6	グリフの表示	31
2.1.4	アドオン	14	3.6.1	ズーム	31
2.1.5	編集ビュー共有	15	3.6.2	スクロール	32
			3.6.3	表示オプション	32
			3.6.4	グリフとレイヤーの色	33
3	編集ビュー	16	3.7	背景レイヤー	34
3.1	グリフの編集	16	3.8	テキスト入力	34
3.2	パスの描画	16	3.8.1	サンプルテキスト	35
3.2.1	ペンツール	16	3.8.2	テキストツール	35
3.2.2	鉛筆ツール	17	3.8.3	書字方向	36
3.2.3	基本図形	17	3.9	計測	36
3.3	パスの編集	18	3.9.1	情報パネル	36
3.3.1	ポイントとパスの選択	18	3.9.2	ものさしツール	38
3.3.2	ポイントとパスの移動	18	3.9.3	計測ガイドライン	39
3.3.3	ポイントとセグメントのタイプ変換	19	3.9.4	補助線	39
3.3.4	アラインメントゾーン内のポイント	19	3.10	注釈	40
3.3.5	拡大縮小と回転	20	3.10.1	注釈カーソル	40
3.3.6	整列	21	3.10.2	注釈テキスト	40
3.3.7	パスの複製	21	3.10.3	注釈矢印	41
3.3.8	ポイントの削除	22	3.10.4	注釈円	41
3.3.9	パスの開閉	23	3.10.5	注釈プラスマイナス	41
3.3.10	パスの切断	23	3.11	画像	41

3.11.1	画像の追加	41	5.2.6	角丸	58
3.11.2	画像の調整	42	5.2.7	スマート角丸	59
3.12	プレビューとテスト	43	5.2.8	変形	59
3.12.1	カーニングのプレビュー	43	5.3	サードパーティー製のフィルタ	61
3.12.2	マスターのプレビュー	43			
3.12.3	パスのオフセットプレビュー	43	6	フォントビュー	62
3.12.4	OpenType フィーチャーの プレビュー	44	6.1	グリフの表示	62
3.12.5	インスタンスのプレビュー	44	6.1.1	グリッド表示	62
3.12.6	OS Xでのプレビュー	45	6.1.2	リスト表示	63
3.12.7	Adobe アプリケーションでの プレビュー	46	6.1.3	グリフの検索	64
3.12.8	Web ブラウザでのプレビュー	46	6.2	グリフセットの管理	64
			6.2.1	新しいグリフの生成	64
			6.2.2	ファイル間のグリフのコピー	65
			6.2.3	グリフの削除	67
4	パレット	48	6.3	グリフのプロパティ	67
4.1	パレットサイドバー	48	6.3.1	グリフ名	68
4.2	線幅メモ	48	6.3.2	字幅とサイドベアリング	68
4.3	自動カーブ	48	6.3.3	カーニンググループ	68
4.4	レイヤー	49	6.3.4	フォント出力時に含める	68
4.4.1	レイヤーパレットの操作	49	6.3.5	カラーラベル	68
4.4.2	特別なレイヤー	50	6.3.6	Unicode	68
4.5	変形	50	6.3.7	メモ (リスト表示)	69
4.5.1	変形の基準点	51	6.3.8	コンポーネント (リスト表示)	69
4.5.2	反転	51	6.3.9	リスト表示での表示専用の項目	69
4.5.3	双方向適用可能な変形	51	6.4	複数グリフの処理	70
4.5.4	整列	51	6.4.1	グリフの選択	70
4.5.5	ブーリアン処理	52	6.4.2	レイヤーコマンドとフィルタ	70
			6.4.3	グリフ名の一括変更	70
5	フィルタ	53	6.4.4	フィルタ	71
5.1	フィルタ	53	6.4.5	パレット操作	71
5.1.1	フィルタメニュー	53	6.5	絞り込みとソート	72
5.1.2	カスタムパラメータとしてのフィルタ	53	6.5.1	検索ボックス	72
5.2	標準で付属するフィルタ	54	6.5.2	カテゴリ	73
5.2.1	マスター互換性を手動修正	54	6.5.3	文字体系	73
5.2.2	ハッチング	55	6.5.4	スマートフィルタ	74
5.2.3	パスをオフセット	56	6.5.5	リストフィルタ	75
5.2.4	重なったパスを合体	57	6.5.6	フィルタの管理	75
5.2.5	ラフ	57	6.5.7	glyphOrder カスタムパラメータ	75

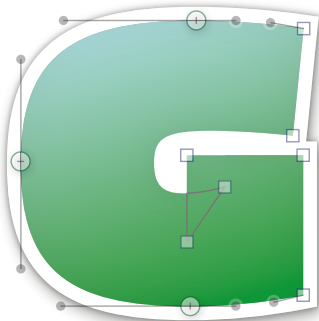
6.6	グリフ名と Unicode	76	7.5.1	グリッドの間隔とグリッド細分	94
6.6.1	グリフ情報のデータベース	76	7.5.2	標準外のグリフ名を使用	94
6.6.2	グリフの命名	78	7.5.3	コンポーネント自動整列を解除	95
6.6.3	グリフ名の変更	80	7.5.4	オルタネートを元のグリフの 横に並べる	95
6.6.4	CID マッピング	80	7.6	メモ	95
6.7	画像	80	7.6.1	フォント全体用メモ	95
6.7.1	画像の追加	80			
6.7.2	画像の表意	80			
7	フォント情報	82	8	図形の再利用	97
7.1	フォント	82	8.1	コンポーネント	97
7.1.1	ファミリー名	82	8.1.1	コンポーネントグリフの生成	97
7.1.2	ユニット数 (UPM)	82	8.1.2	パスをコンポーネントに変換	98
7.1.3	デザイナーとデザイナーの URL	83	8.1.3	レシピ	98
7.1.4	製造者と製造者の URL	83	8.1.4	コンポーネントの編集	99
7.1.5	著作権	83	8.1.5	ベースグリフと コンポーネントグリフとの関係	100
7.1.6	バージョン	83	8.1.6	アンカー	100
7.1.7	日付	84	8.1.7	自動整列	102
7.1.8	カスタムパラメータ	84	8.1.8	コンポーネントのロック	104
7.2	マスター	84	8.1.9	コンポーネントの分解	104
7.2.1	プロポーシオン：ウエイトとグリフ幅	85	8.1.10	パスとコンポーネントの混合	104
7.2.2	メトリクス	85	8.1.11	コンポーネントの ネスティング (入れ子)	105
7.2.3	ステム	86	8.1.12	優先される発音記号のグリフ名	105
7.2.4	アラインメントゾーン	87	8.1.13	アンダースコアで始まる グリフ名のコンポーネント	105
7.2.5	カスタムパラメータ	87	8.2	スマートコンポーネント	106
7.3	インスタンス	88	8.2.1	スマートグリフの設定	106
7.3.1	アクティブ	88	8.2.2	スマートコンポーネントの追加	108
7.3.2	スタイル名	88	8.2.3	スマートコンポーネントの補間	108
7.3.3	ウエイトとグリフ幅	89	8.2.4	自動整列用アンカー	109
7.3.4	スタイルリンク	90	8.3	コーナーとキャップコンポーネント	109
7.3.5	補間	91	8.3.1	コーナーコンポーネント	109
7.3.6	カスタムパラメータ	91	8.3.2	キャップコンポーネント	111
7.3.7	インスタンスプレビュー	91			
7.4	フィーチャー	91	9	スペーシングとカーニング	113
7.4.1	OpenType フィーチャーコード	91	9.1	スペーシング	113
7.4.2	フィーチャーコードの自動生成	92	9.1.1	スペーシングのショートカット	113
7.4.3	フィーチャーコードの手動編集	92			
7.5	その他の設定	94			

9.1.2	メトリクスキー	114	11.3.8	斜線	137
9.1.3	メトリクスキーと自動整列	115	11.3.9	未編集ポイントの自動補間	138
9.2	カーニング	115			
9.2.1	カーニングの方法	115	12	マルチプルマスター	139
9.2.2	カーニンググループ	116	12.1	概要	139
9.2.3	カーニングペアの確認	118	12.2	マスターのセットアップ	140
9.2.4	カーニングペアの削除	118	12.3	インスタンスのセットアップ	141
9.2.5	カーニングペアのコピー	119	12.4	互換性の修正	141
9.2.6	カーニングのクリーンアップ	119	12.5	マスターレイヤーの比較	144
9.2.7	カーニングの圧縮	119	12.6	複数ファイル間でのファミリーの 一貫性の確保	145
9.2.8	kern フィーチャー手動追加	120	12.7	ブレースレイヤー	146
			12.8	ブラケットレイヤー	146
10	PostScript ヒンティング	121	12.9	開いたブラケットレイヤー	147
10.1	ヒンティング	121			
10.2	フォント全体のヒンティング	122	13	カラーフォント	149
10.2.1	スタンダードステム	122	13.1	複数レイヤーでの作業	149
10.2.2	アラインメントゾーン	123	13.1.1	全レイヤー選択ツール	149
10.2.3	カスタムパラメータ	124	13.1.2	メトリクスの同期	149
10.3	オートヒンティング	124	13.1.3	カラーフォントの出力	149
10.3.1	フレックス・ヒント	125	13.2	複数レイヤーフォント	149
10.4	グリフレベルの手動ヒンティング	126	13.2.1	マスターのカスタム名と プレビュー色	150
10.4.1	ステムヒント	126	13.2.2	すべてのマスター用のインスタンス	150
10.4.2	ゴーストヒント	128	13.2.3	複数レイヤーフォントの プレビューと使用	150
10.4.3	マルチプルマスターのヒンティング	129	13.3	Microsoft カラーフォント	151
			13.3.1	CPAL テーブル	151
11	TrueType ヒンティング	130	13.3.2	カラーレイヤーの作成	152
11.1	インストラクション	130	13.4	Apple カラーフォント	152
11.2	オートヒンティング	130	13.4.1	ビットマップ画像ファイル	153
11.2.1	手動または自動の インストラクション	131	13.4.2	画像サイズごとのレイヤー	153
11.3	手動インストラクション	131	13.4.3	Apple カラーフォントの使用	153
11.3.1	水平ステムとゾーン	132	13.5	SVG カラーフォント	154
11.3.2	ラスターライザープレビュー	133	13.5.1	SVG ファイル	154
11.3.3	アンカー	133	13.5.2	SVG レイヤー	154
11.3.4	整列	134	13.5.3	SVG カラーフォントの使用	154
11.3.5	ステム	134			
11.3.6	トリプルヒント	136			
11.3.7	補間	136			

14	エラー対処	156	16	機能拡張	171
14.1	グリフ名	156	16.1	スクリプト	171
14.2	フォント名	156	16.2	プラグイン	172
14.3	Unicode 値の重複	157	16.2.1	プラグインの手動インストール	172
14.4	OpenType のフィーチャーコード	157	16.2.2	プラグインマネージャー	172
14.5	アウトライン	159	16.3	SDK	173
14.5.1	開いたパス	160			
14.5.2	不正なパスの方向	160	17	補足資料	174
14.5.3	重なり合った同一のパス	160	17.1	フィーチャー自動生成	174
14.5.4	アウトラインの非互換	160	17.2	クラス自動生成	184
			17.3	カスタムパラメータ	184
15	読み込みと書き出し	161	17.4	グリフの各種属性データ	208
15.1	ベクトルベースの 描画アプリケーション	161	17.4.1	グローバルなグリフデータの変更	208
15.1.1	Adobe Illustrator	161	17.4.2	ファイルごとグリフデータの変更	210
15.1.2	Sketch	161	17.5	キーボードショートカットの変更	211
15.1.3	パスのコピーペースト	161			
15.2	FontLab Studio	162			
15.2.1	FontLab から Glyphs へ	162			
15.2.2	Glyphs から FontLab へ	162			
15.3	Unified Font Object	162			
15.3.1	UFO 形式でファイル保存	162			
15.3.2	UFO 形式でファイル書き出し	162			
15.3.3	UFO ファイルの読み込み	163			
15.4	Type1、OpenType、TrueType	163			
15.4.1	既存のフォントを開く	163			
15.4.2	OpenType/CFF フォントの出力	163			
15.4.3	OpenType/TT フォントの出力	164			
15.5	Web フォント	165			
15.5.1	WOFF、WOFF2、 EOT フォントの出力	165			
15.6	メトリクス	166			
15.6.1	メトリクスデータの読み込み	166			
15.6.2	メトリクスデータの出力	167			
15.7	プロジェクト	168			
15.7.1	プロジェクトのセットアップ	168			
15.7.2	プロジェクトの出力	169			

1 Glyphs

1.1 OpenType フォントを作るためのツール



Glyphs (グリフス) は基本的には新しいフォントをデザインし出力するためのアプリケーションです。その根幹にあるのは、単語を組みながらアウトラインを編集できるというアイデアです。すべてのツールは書体デザインの各作業をできるだけ直感的かつ簡単に遂行できるように最適化されています。

私たちはGlyphsを使えば、本当に必要なとき以外は技術的なことに手を煩わせることなく、デザインに集中できると信じています。たとえばGlyphsはフォント出力時に様々な技術的な処理（重なったパスの結合など）をまとめて自動的に実行します。ですからユーザーは、従来のようにデザイン用のバージョンとフォント出力用のバージョンを個別に保存し管理する必要がありません。もちろん、こういった自動的な処理は必要に応じて解除できます。

Glyphsはすべての種類のOpenTypeフォントの出力に対応しています。OpenType/CFF (Compact Font Format、拡張子.otf)、OpenType/TT (TrueType、拡張子.ttf)、そしてWebフォント形式ではWOFFとWOFF2 (Web Open Font Format、拡張子.woffと.woff2)、EOT (Embeddable OpenType、拡張子.eot) が書き出せます。なお、最近のモダンなソフトウェア (Webブラウザを含む) の多くはOpenTypeをサポートしていますが、旧式のソフトウェアではOpenType以前またUnicode以前のTrueTypeフォントやPostScript Type1フォントを必要とするものも存在します。それらはGlyphsでは出力できません。

Glyphsは既存のフォントを開くことができます。しかしすべてのフォント情報が読み込まれるとは限りません。詳細は「15 読み込みと書き出し」(p.161)を参照してください。

1.2 インストール

1.2.1 必要インストール環境

GlyphsはMac専用のアプリケーションであり、Glyphs 2.xの動作にはOS X 10.9.5 Mavericks以降の環境が必要です。

1.2.2 インストール

試用する場合も購入を希望する場合も、まずはglyphsapp.com/buyの「Glyphs 2 Trial」から試用版をダウンロードしてください。アプリケーションフォルダにドラッグするだけでインストールは完了します (アプリケーションフォルダはFinderの「移動 > アプリケーション」メニュー (ま

たはcommand+shift+A) でアクセスすることができます)。試用期間は30日間で、全機能を使うことができます。

1.2.3 製品版へのアップグレード

試用版を製品版にアップグレードするには glyphsapp.com/buy の「Glyphs Full License」で製品版ライセンスを購入します。ダウンロードした「.glyphs2license」ファイルをダブルクリックするか、Glyphs アプリケーション（試用版）のアイコンにドラッグ&ドロップすることでアップグレードが完了します。

1.3 コミュニティ

ご質問やご意見があれば、Glyphsのフォーラムに登録また投稿することができます（英語、forum.glyphsapp.com）。スパム投稿防止のため、ユーザーはTwitter アカウントを保持している必要があります、最初の投稿はフォーラム管理権限者の承認を待つ必要があります。バグレポートに関しては、bugreport.glyphsapp.com に投稿できます。

1.4 アップデート

Glyphsは頻繁にアップデートされます。新機能はglyphsapp.com/blog で紹介されます。細かい更新履歴は「ヘルプ > 更新履歴」で確認できます。お使いのバージョン番号をチェックするには「Glyphs > Glyphsについて」を選びます。バージョンアップは「Glyphs > 更新をチェック...」を選ぶか、環境設定で自動チェックをオンにしておくことで自動的に実行できます。詳細は「2.1.1 アップデート」(p.11) を参照してください。

1.5 Glyphs Mini

Glyphs Miniは廉価版アプリケーションです。ここで紹介されているフル版の多くの上級機能がカットされています（例：プラグイン、レイヤー、マルチプルマスター、Pythonスクリプト、OpenTypeフィーチャー手動編集、カスタムパラメータ編集、手動ヒンティング）。Glyphs Miniはシンプルで安価な入門用アプリケーションで、趣味の書体デザインやアイコンフォントの作成に向いています。

1.6 キーボードショートカット

ワークフローを効率化するため、Glyphsにはたくさんのキーボードショートカットが用意されています。これらのカスタマイズはシステム環境設定の「キーボード > ショートカット > アプリケーション」で行います。

Glyphsにデフォルトで設定されているいくつかのショートカットはシステム標準のものと競合しています（例：Spotlight (command+space)、Spaces (control+矢印)）。Glyphsを快適に使用するために、これらシ



ステム用ショートカットを変更されることをお勧めします。(なお、本ハンドブックではショートカットなどの同時押しはshift+F などのように+を使って表記します。)

ツールバーのショートカットの変更方法については「17.5 キーボードショートカットの変更」(p.211)を参照してください。

1.7 クラッシュレポート

Glyphsがクラッシュすると2種類のクラッシュレポートダイアログが表示されます。1つはAppleに送られるものであり私たち開発者には届きません。もう1つはGlyphsを再起動したときに現れるもので、下図のようにGlyphsのアイコンが左上に表示されています。このレポートはクラッシュした原因を突き止めるためにとっても重要なものです。コメントをご記入いただかなくても、お使いのハードウェア、OS、Glyphsのクラッシュ寸前の状態、またどの時点でクラッシュしたかなどの細かな技術的情報が送信されます。Glyphsの品質を向上させるため、常に送信していただけるようお願いします。もしクラッシュを確実に起こす手順が分かっているのであれば、コメント欄にて説明していただくと助かります(トラブルの内容についてはフォーラムまたはbugreport.glyphsapp.comにご報告いただいても結構です)。

初めてレポートを送信する際、Glyphsはお名前とメールアドレスの入力を促します。入力された内容は次回以降、自動的に入力されることとなります。これを拒否し、後でレポートダイアログに手動入力することも、またはフィールドが空白のまま送信することも可能です。なお、あなたの環境で起きたクラッシュの原因を究明するために、その時点で作業していたGlyphsのファイルの提供をお願いすることがあります。この場合、ファイルは問題解決後に消去いたします。あなたの明確な同意がないかぎり他者に送られることはありません。

Glyphs障害報告

Glyphsが予期しない理由で終了。問題が修正できるよう、レポートを送信しますか？

お名前: Toshi Omagari

メールアドレス: tosche@mac.com

▼ コメント欄

再現手順がわかりましたらご記述ください。

詳細を表示 キャンセル 送信

報告内容に個人情報は含まれておりません。

2 環境設定

2.1 環境設定の開き方

環境設定ウィンドウは「Glyphs > 環境設定」(command+,) で開けます。

2.1.1 アップデート

Glyphsのアップデート方針を設定できます。「今チェックする」ボタンを押すとただちに最新バージョンを確認し、もしアップデートがあればダウンロードボタンと更新履歴を表示します。最後に更新をチェックした日時はボタンの上に表示されます。

「更新を自動的にチェック」がオンになっている場合は定期的かつ自動的にGlyphsサーバーへ接続し、最新バージョンの存在を確認できます。このオプションはオンにしておくことをお勧めします。

「開発中バージョンも表示」をオンにしておくことで安定したバージョンだけでなくβ版にもアップデート可能になります。Glyphsのβ版は頻繁に更新され、バグ修正もこちらの方が迅速に行われます。もし安定版を待ってられない場合や新機能をいち早く試したい場合はこちらもオンにしてください。β版は安定版と比べてテストが行き届いていませんので、更新前に安定版の複製を取っておくことをお勧めします。最新の安定版は glyphsapp.com/buy でダウンロード可能です。



2.1.2 ユーザー設定

「読み込み元のファイルのグリフ名を保持」オプションでは、既存のフォントファイルや他のアプリケーションで作られた作業ファイルを開く際のグリフ名について設定します。自動的にGlyphs標準のグリフ名に変更してかまわない場合はオフ、他のアプリケーションとの連携のためなど非標準のグリフ名を使う必要がある場合はオンにします。デフォルトではオンになっています。

「読み込んだファイルではコンポーネントの自動整列を解除」がオフの場合、既存のフォントファイルや他のアプリケーションで作られた作業ファイルを開くとコンポーネントの自動整列が実行され、コンポーネントの位置が統一されます（読み込み元のファイルとは体裁が変わる可能性があります）。オンにすると自動整列はおこなわず、読み込み元のファイルの各コンポーネントの位置が保持されます。自動整列についての詳細は「8.1.7 自動整列」(p.102)を参照してください。

編集ビューでは複数のグリフを並べた状態でデザイン作業などをおこなうことができます。ここで表示できるテキストの行長を「編集ビュー幅」で設定します。入力する数値はフォントのユニット数です。1ユニットはフォントのボディを細分化した単位で、フォント情報で設定するUPMに関連しています。詳細は「7.1.2 ユニット数 (UPM)」(p.82)を参照してください。

「ハンドルサイズ」は編集ビューで表示されるすべてのポイント（ベジェ曲線のポイント、アンカー、ヒント）のサイズです。サイズが大きいと視認性が向上しクリックしやすくなります。小さいとパスのデザインが見やすくなります。

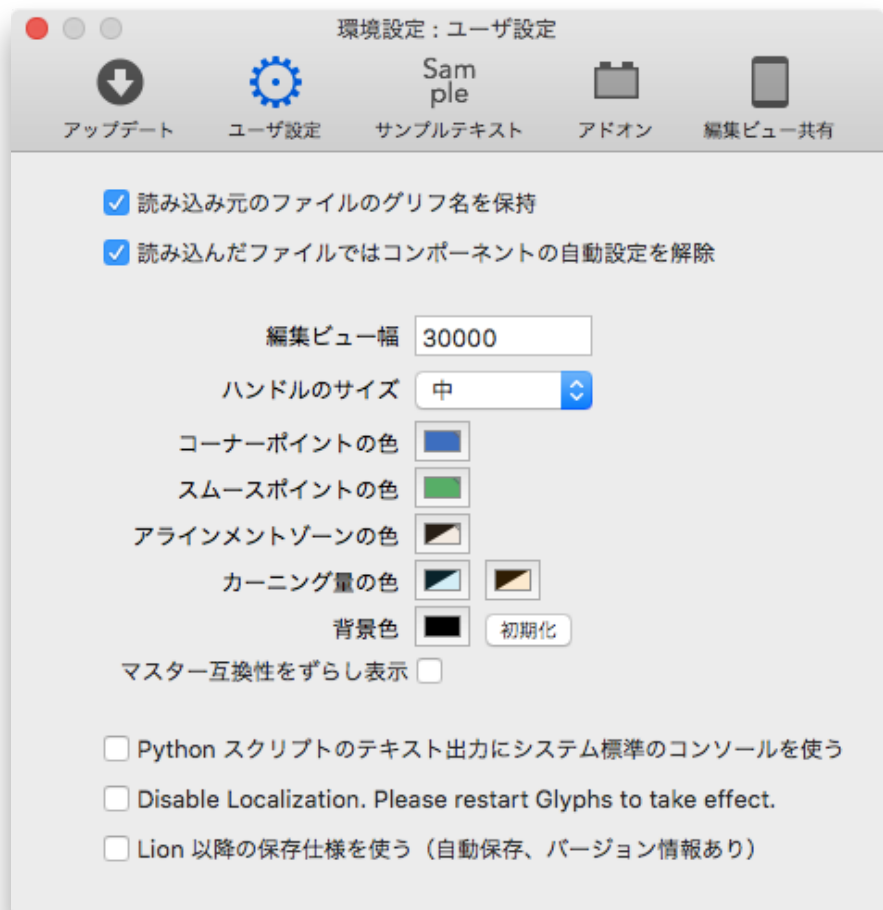
コーナーポイント、スムーズポイント、アラインメントゾーン、カーニング量、そして背景色もユーザー設定で変更できます。「初期化」ボタンをクリックするとすべての色が初期化されます。

「マスター互換性をずらし表示」は「表示 > マスター互換性を表示」(control + option + command + N) を有効にした状態でのアウトライン表示に関する設定です。オフの場合、各マスターの原点を揃えた上で重ねて表示します。オンの場合は斜めにずらして表示します。詳細は「12.4 互換性の修正」(p.141)を参照してください。

Python スクリプト API のテキスト出力は Glyphs のマクロパネル（ウインドウ > マクロパネル）で行われます。オンにすると、OSX 付属のコンソール .app で行われます。スクリプトが原因で Glyphs がクラッシュするなど、マクロパネルで出力結果が確認できない時に便利です。

「Disable Localisation. Please restart Glyphs to take effect.」をオンにするとアプリケーションの言語を英語に変更できます。アプリケーションの再起動後に効果が現れます。

「Lion 以降の保存仕様を使う」をオンにすると、OS X 10.7 Lion で実装されたファイル保存方式が使われます。ファイルは自動保存され、過去のバージョンを閲覧できるようになります。ただし独自のバージョンコントロールシステムを持ったクラウドに保存する場合などはこの機能を切っておいた方がよいでしょう。



2.1.3 サンプルテキスト

ここでは付属のサンプルテキストを編集、ペースト、初期化したりすることができます。テキストフィールドはUnicodeに完全対応しており、発音記号や欧文以外の文字もそのまま入力できます。各サンプルは1行になっている必要があるため、編集ビュー内で複数行表示させたい場合はバックslashと小文字のn (\n) を改行マークとして挿入してください。グリフ名を指定したい場合はslashで始めて (例: /quoteright)、グリフ名の末尾はスペースまたはslashで区切ります。これはタイピングしにくい文字やUnicodeにない文字、例えばOpenTypeのオルタネート (/a.sc) などを表示させたいときに便利です。/Placeholder (他のグリフを続けたいときはやはりスペースかslashで区切る) は現在選択中のグリフを繰り返し表示してくれます。slashそのものを入力したいときは二度打ちしてください (//)。

「ファイルを開く」ボタンで外部のテキストファイルを読み込みます。「初期化」ボタンを押すとサンプルテキストがGlyphs標準のものに戻ります。サンプルテキストをより使いこなすには、「3.8.1 サンプルテキスト」(p.35) を参照してください。

2.1.4 アドオン

「プラグイン」タブの左のリストにはインストールされているすべてのプラグイン名が表示されます。プラグイン開発者が自動更新機能を設定をしている場合、アップデートがあった際は自動で通知され、直接ダウンロードするためのダイアログかプラグインをダウンロードできるWebサイトへのリンクが表示されます。

「自動的に更新をチェック」がオンになっているとGlyphsは起動時にプラグインの更新を確認し、更新があれば環境設定のこのタブに案内します。新バージョンがあるプラグインにはアスタリスクマークが表示され、選択すると「ダウンロード」ボタンが右下に表示されます。これをクリックすることでプラグインのホームページにジャンプすることができます。なお、アップデートの有無を手動で確認したい場合はプラグインフィールドでプラグイン名を選択した上で「更新をチェック」ボタンをクリックします。

ダウンロードしたプラグインをインストールする場合、プラグインをFinderでダブルクリックして表示されたダイアログの指示に従ってください。アプリケーションの再起動後にプラグインが有効になります。プラグインをアンインストールする際は、環境設定のこのタブで「プラグイン」フィールドのプラグイン名を右クリックして「Finderに表示」を選択します。プラグインが表示されたらアイコンをゴミ箱や他のフォルダに移動し、アプリケーションを再起動してください。

「Python モジュール」タブでは、サードパーティー製のPython ライブラリであるVanilla (Tal Leming作)、Robofab (Tal Leming、Erik can Blokland、Just van Rossum作)、FontTools (Just van Rossum作、Behdad Esfahbod管理) をインストールできます。Glyphsのスク립トの中にはこれらのライブラリを使うものがあります。サードパーティー製のスク립トをインストールする場合や独自のPython スクリプトを書く場合は、これらのライブラリをインストールしておくことを強く推奨します。また、RobofabのPython ラッパーであるObjectsGS.pyがScriptsフォルダにインストールされます。Robofab スクリプトは他のPython ライブラリと同様にインポートして使います (例: `from robofab.world import CurrentFont`)。



2.1.5 編集ビュー共有

「外部プレビューを有効化」がオンの場合、編集ビューの内容を iPhone や iPad などの iOS デバイスにリアルタイム表示することができます。iOS デバイスには Glyphs Viewer をインストールしておいてください (AppStore から無料で入手可能)。iOS デバイスで Glyphs Viewer アプリを開始し、表示したい Glyphs を選択すると編集ビューの内容が即座に表示されます。メニューに戻りたいときは画面の任意の場所を長押しします。

iOS デバイスと Glyphs を実行中の Mac は同じ WiFi ネットワークに存在する必要があります。もし Glyphs Viewer から対象の Glyphs を見つけられない場合、両者が同じネットワークに繋がっていることを確認してください。解決しない場合は両方のマシンを再起動してみてください。それでも解決しない場合は別のネットワークを試すかルーターの設定を調整してください。なお、Mac と iOS が別々のネットワーク規格で同じルーターに接続している場合、Glyphs Viewer は Mac を発見できません。これはルーターが様々な無線規格 (例: 802.11g や 802.11n) に対応しており、それらを同時に有効化している場合などに起こります。ルーターを g または n のいずれか一方だけに設定すると問題が解決できるかもしれません。

3 編集ビュー

3.1 グリフの編集


編集ビューではグリフを編集できます。編集ビューにアクセスするには、「表示 > 新規タブ」(command+T) で新しい編集タブを開きます。すでに開かれているタブにアクセスするには当該のタブをクリックするか、またはoption+command+数字キーで、任意のタブを選択します。例えばoption+command+3だと、開かれている複数のタブのうち左から3番目のタブがアクティブになります。最初のタブ(option+command+1)は必ずフォントビューです。詳細は「6 フォントビュー」(p.62)を参照してください。

編集ビューは編集モードとテキストモードの2つのモードがあります。テキストツール(ショートカットはT)を選択するとテキストモードになります。詳細は「3.8 テキスト入力」(p.34)を参照してください。他のツールを選択するか、任意のグリフをダブルクリックするか、またはグリフ選択中(任意のグリフの手前にテキストカーソルがある状態)にescキーを押すと編集モードに切り替わります。

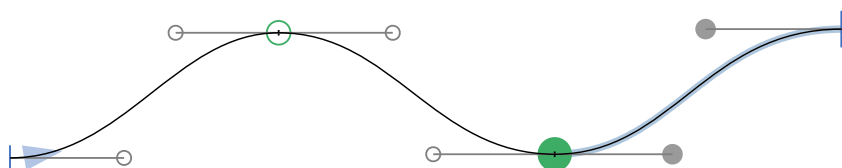
3.2 パスの描画

ペンツール、フリーハンドツール、基本図形ツールなどで新しいパスを描画します。

3.2.1 ペンツール

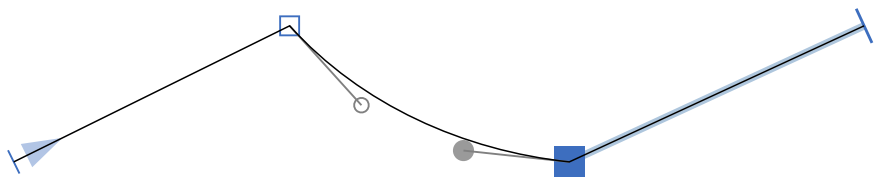
ペンツール (ショートカットはP) が選択された状態では、クリックすると直線が描け、ドラッグすると曲線が描けます。ドラッグ中にスペースバーを押すとポイントを動かします。アウトラインを閉じるには、書き始めのオンカーブポイントをクリックします。

スムーズに繋がっているオンカーブポイントは緑色の丸で表示されず(この色は「Glyphs > 環境設定」のユーザー設定で変更できます)。スムーズな繋がりとはカーブ(オンカーブポイントの両側が曲線で、2本のハンドルが一直線上にある状態)、またはタンジェント(オンカーブポイントの片方が直線、もう片方が同角度の曲線)の状態を指します。なお、ポイントの表示、非表示は「表示 > ポイントを表示」メニューから切り替えられます。



カーブの描画中にハンドルを折りたい（コーナーポイントにしたい）場合はドラッグ中にoptionキーを押します。コーナーポイントは青い正方形で表示されます（この色も環境設定で変更できます）。コーナーポイントから伸びるハンドルはスムーズポイントと違い他方のセグメントの角度の影響を受けないため、独立して動かすことができます。

ハンドルは灰色の丸で表示され、その角度と長さでセグメントの曲線を表現します。開いたパスでは開始点と終了点はパス方向に直角の短い線が表示されます。パスの開始点は3角形で表示され、パスの描画方向（右周り・左回り）も示しています。パスの描画方向は特定のパス機能およびグリフのレンダリングにおいて、とても重要な役割を果たします（詳細は「3.3.12 パス方向の管理」(p.24)を参照)。




ここで用語の説明をしておきます。Glyphsでは、アウトライン上にあるポイントを「オンカーブポイント (on-curve point)」、アンカーポイントから伸びて曲線を定義するポイントを「ハンドル」または「オフカーブポイント (off-curve point)」と呼びます。両者を区別する必要のない際はどちらも「ポイント」と呼びます。

オンカーブポイントは、他のグラフィックソフトでは通常「アンカーポイント」と呼ばれています。しかし「アンカー」はフォント製作ソフトでは他の用語として用いられるため、曖昧さ回避のためGlyphsでは「オンカーブポイント」（または「ポイント」）と呼ぶことにしています。アンカーの詳細は「8.1.6 アンカー」(p.100)を参照してください。

オンカーブポイントとオンカーブポイントの間に描かれる曲線または直線を「セグメント」と呼びます。ポイントとセグメントから構成される図形を「アウトライン」または「パス」と呼びます。

3.2.2 鉛筆ツール

鉛筆ツール  (B) はマウスドラッグによるフリーハンドで素早くカーブを描画するためのツールで、特にペンタブレットを使用する際に便利です。鉛筆ツールで描かれたパスはユーザーの描いた線を忠実に再現するため、多くのポイントを挿入します。そのため、描画後はおそらく修正作業が必要になるでしょう。

3.2.3 基本図形

基本図形ツールでは4角形と楕円形の描画が可能です。4角形と楕円形を選


択するには基本図形ツールのボタンをクリックします (F)。4角形と楕円形を切り替えるには、基本図形ツールをクリックしたままにするかショートカット shift+Fで切り替えるか、一方のツールを選択中にコンテキストメニューで「楕円を描画」または「長方形を描画」で切り替えられます。



ツール選択後、描画エリアを斜めにドラッグすることで4角形または楕円を描画します。shiftキーを押しながらドラッグすれば正方形または真円を描けます。またoptionを押しながらドラッグすると、ドラッグ開始地点を中心として図形を描くことができます。サイズを数値で指定して図形を描くには、描画エリアをクリックして現れるダイアログに数値を入力します。

3.3 パスの編集

3.3.1 ポイントとパスの選択

選択ツール  (V) ではクリックすることで個別のポイントを、ドラッグすることで範囲内の全ポイントを選択することができます。optionキーを押しながらドラッグすると、ハンドルは無視され範囲内のオンカーブポイントのみが選択されます。なお、ポイントを追加選択したい場合、あるいは一部のポイントの選択を解除したい場合はshiftキーを押しながらクリックまたはドラッグします。セグメントをダブルクリックするとパス全体が選択されます。

Glyphsではオンカーブポイントを選択することなくハンドルだけを選択することができます。またshiftを押しながら追加選択する際、ポイントの前後関係に依存せず全く離れたハンドルも追加できます。ポイントが1つだけ選択されている状態では、tabキーを押すとパス上の次のポイントに移動し、shift+tabキーで前のポイントに移動できます。

3.3.2 ポイントとパスの移動

選択したポイントやパスはマウスによるドラッグまたはキーボードの矢印キーで移動できます。マウスによるドラッグ中にshiftを押すと、移動方向が水平、垂直に限定されます。なお、shift+矢印キーでは10ユニット単位、command+矢印で100ユニット単位の移動が可能です。

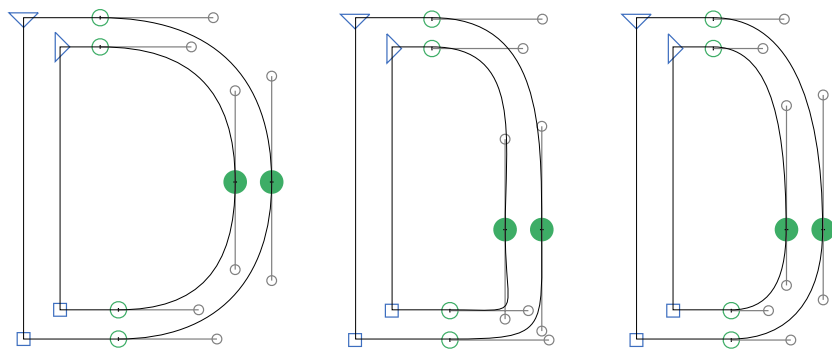
オンカーブポイントを移動すると、付随するハンドルは選択されていなくても移動します。optionキーを押しながら移動すると、ハンドルの位置が固定されたままオンカーブポイントが移動します。ポイントの移動中にcontrolとoptionを押すと「ナッジ移動」ができます。これは周囲のセ

グメントの曲線率を保ったままの移動です。

左：元のアウトライン。2つのポイントを選択中。

中央：選択したポイントを左下に移動。ハンドルは変化しないまま追従。

下：選択したポイントを左下にナッジ移動。曲線率を保つため、ハンドルは変化。



同様にハンドルもマウスによるドラッグまたは矢印キーで移動することができます。複数のハンドルが選択されている場合はすべて同時に移動できます。ここでは shift + 矢印キーでは10ユニット単位、command + 矢印で100ユニット単位の移動が可能です。option キーを押しながらだとハンドル角度を固定したままで移動させることができます。

スムーズポイント（緑色の丸）に隣接するハンドルを、control + option を押しながら移動すると反対側のハンドルも対称移動します。

曲線セグメントをドラッグすると両端のハンドルが移動します。この際、コーナーポイントから生えているハンドルは角度も変更されます。角度を固定したい場合は option を押しながらドラッグを開始してください（ドラッグ中に option を押しても角度は固定されません）。

3.3.3 ポイントとセグメントのタイプ変換

オンカーブポイントの接続タイプをスムーズポイント（緑色の丸または3角で表示）からコーナーポイント（青色の4角または3角で表示）に切り替える際は、そのポイントをダブルクリックします。ポイントを選択して return キー、または enter キーを押すことでも切り替えることができます。コーナーポイントからスムーズポイントに切り替える際も同様です。

「パスのクリーニング」を実行する時は注意してください。この操作ではほぼ無意味なポイントが取り除かれますが、他のマスターとの互換性を保つためには、そのポイントが必要なかもしれません。

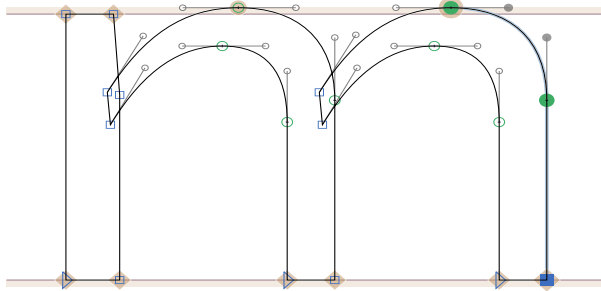
「パス > パスをクリーニング」(option + shift + command + T) を実行すると、すべてのオンカーブポイントのタイプが適切に設定されます（コーナーポイントの両側のハンドルが一直線上に並んでいる場合はスムーズポイントに変更されます）。パスの一部を選択中はその範囲のみが対象となります。直線セグメント上にあるハンドルなどの無駄な要素も取り除きます。

直線のセグメントを option キーを押しながらクリックすると曲線に切り替えられます（両端にハンドルが発生します）。曲線セグメントを直線に切り替えたい場合は、ハンドルを選び delete キーで削除します。

3.3.4 アラインメントゾーン内のポイント

縦メトリクス線（「7.2 マスター」、p.84）に整列しているポイントはベー

ジュ色の菱形、アラインメントゾーン内にあるポイントはベージュ色の円形でハイライトされます。フォントを小サイズで表示している際でも、ポイントが正確に揃っているか否かを一目で確認できます。



3.3.5 拡大縮小と回転

選択したポイントやアウトラインの状態は灰色の情報パネルに表示されます（表示 > 情報パネルを表示、shift + command + I）。

Tip: Glyphs内のすべてのテキストフィールドは矢印キーの上下で数字を増減させられます。shiftキーを押しながらで10単位で変化します。

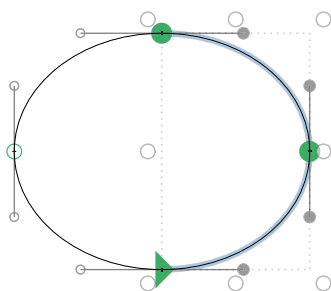




ポイントを1つ選択している時は、情報パネルのx欄とy欄で位置を数値指定することができます。上下の矢印キーで数値を変更することもできます。

複数のポイントを選択している時は、情報パネル左の9つの点で基準点を設定した後、x欄とy欄で基準点の位置を数値指定します。幅と高さを数値指定することもできます。なお、幅と高さの比率を固定したい場合は🔒をクリックしてロック状態🔒にしておきます。幅や高さを個別に指定したい場合はロックを解除します。

複数のポイントが選択されている場合、情報パネルの右端に数字が表示されます。これは選択されているパスのポイント総数（□）と、実際に選択されているポイントの総数です（■）。

複数のポイントが選択されている場合、バウンディングボックスを表示することができます（表示 > バウンディングボックスを表示、option + shift + command + B）。ボックスの白丸のいずれかをドラッグすることで選択内容を拡大縮小することが可能です。shiftキーを押しながらドラッグすると、縦横の比率を保持したまま変形します。



このほか回転ツール  (R) で選択部分を回転したり、拡大縮小ツール  (S) で大きさを変更することも可能です。一回クリックして変形の基準点を決めた後、ドラッグすることで回転や拡大縮小を実行できます。shift キーを押しながらドラッグすることで90度ごとの回転、または比率を固定した拡大縮小ができます。

他にもパレットやフィルタによる変形が可能です。それぞれパレットとフィルタの章を参照してください。

3.3.6 整列

「パス > 選択ポイントを整列」を選ぶと、選択されているポイントがすべて一列に並びます。この機能はオンカーブとオフカーブ両方のポイントに作用します。Glyphsは選択範囲の状況から判断して自動的に水平または垂直にポイントを整列します。「選択ポイントを整列」は灰色の情報パネルにある変形基準点に従います。

原点とは $x=0, y=0$ の地点のことです。イタリック角度が 0° 以外の場合はサイドベアリングの線がxハイトの中心を通過する斜め線になるため、原点はそれによって斜め線とベースラインの交点になります（傾斜がある場合、原点のx座標は0ではなくなります）。

2つのポイントと1つのアンカーを選んだ状態で「選択ポイントを整列」を実行すると、アンカーは両ポイントの中間地点に水平移動します。これはアンカーを素早く選択範囲の中央に整列させたい場合に有効です。また1つのポイントと1つのコンポーネントを選んだ状態で同機能を実行すると、ポイントを原点と捉えてコンポーネントだけが移動します（ポイントは移動しません）。あるポイントを基準としてコンポーネントを動かすときに便利です。もしコンポーネントが「origin」という名前のアンカーを含んでいる場合は、Glyphsは原点ではなくそのアンカーを使ってコンポーネントを移動します。

「パス > 選択ポイントを整列」を1個のポイントに対して実行すると、背景レイヤーにある最も近距離のポイントと同じ位置に移動します。

「パス > 選択ポイントを整列」は個別のポイントに対して適用する機能です。アウトラインの一部または全体やコンポーネントを整列させたいときはパレットサイドバー (option + command + P) の「変形」機能を使用してください。詳細は「4.5 変形」(p.50) を参照してください。

3.3.7 パスの複製


パス全体を複製するには、まずパスをダブルクリックして選択します（全セグメントが青く表示されます）。パスを完全に囲むようにドラッグすることでパス全体を選択することも可能です。必要に応じてshift キーを押しながら他のパスも追加選択してください。その後、option キーを押しながら選択したパスをドラッグすることで任意の位置に新しいパスを複製することができます（ドラッグ開始前からoption キーを押してください）。

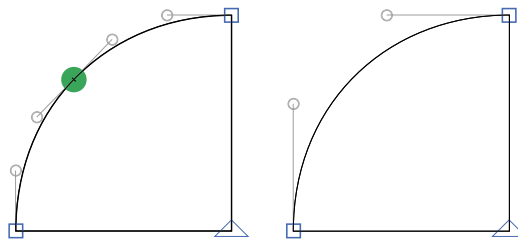
コピー (command + C) とペースト (command + V) も使えます。この場合、新しいパスは元のパスと同じ位置に複製されます。ペースト直後は新しいパスのみが選択された状態になっています。複製後にマウスで

なくキーボードで移動させたい場合はこちらの方が便利でしょう。

セグメントを複製するには、まずセグメントを定義しているオンカーブポイントを選択します。クリックでもドラッグで囲んでも選択できます(選択されたセグメントは青く表示されます)。その後、option を押しながらからドラッグすると選択されたセグメントが複製されます。セリフなど特定の部分を複製したいときに便利です。

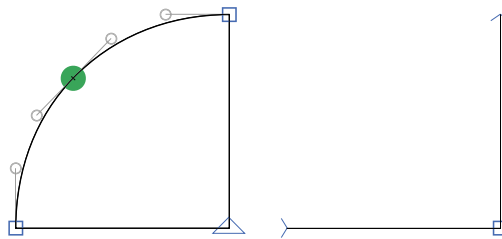
3.3.8 ポイントの削除



ポイントを削除するには任意のポイントを選んでdeleteキーを押してください。または削除ツール  (E または shift+E) でクリックして削除することもできます。Glyphs はパスを閉じたまま削除し、なるべく削除前のカーブの状態を維持するために隣接するハンドルを修正します。

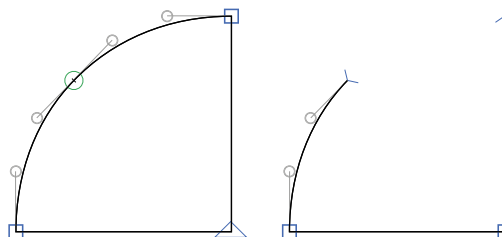


Tip: セグメントを削除する手っ取り早い方法は、ペンツール (P) でポイントを挿入し、それを即座に option+delete で削除することです。



option キーを押しながら削除すると、パスを閉じずにセグメントごと削除することができます。

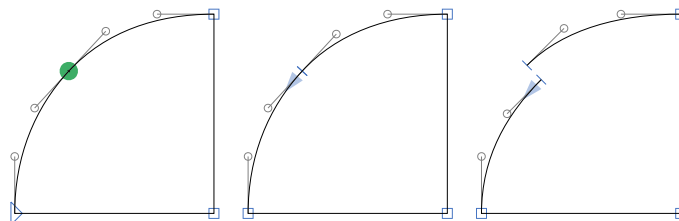


また、削除ツール  で option を押しながらからセグメントをクリックする方法、あるいは選択ツール  でハンドルを選び option+delete のショートカットを実行する方法でもセグメントを削除できます。またはオンカーブポイントを含めたセグメント全体を選択し、option+delete で削除します。これは隣接する、しないに関わらず複数のセグメントに適用できます。



3.3.9 パスの開閉


ペンツール  で既に存在するポイントをクリックするとパスが開きます。開いたパスの末端のポイントは青い線分で表示されます。この状態になると選択ツール  でポイントをバラバラに動かすことができます。

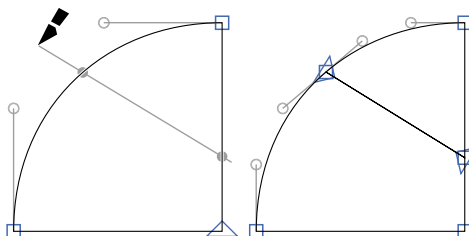



パスをふたたび閉じるには、開いているポイント同士をドラッグで重ね合わせます。ポイントの形状は線分ではなくなり（3角形、丸など）、パスが閉じたことが視覚的に伝わるようになっていきます。開いたパスのあるグリフを編集集中にコンテキストメニュー（右クリックまたはcontrol+クリックで表示）から「開いたパスを閉じる」を実行することで閉じることもできます。または開いた2つの末端ポイントを選択してコンテキストメニューから「ポイントを接続」を選択します。

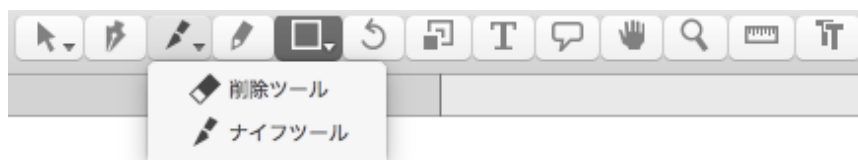
3.3.10 パスの切断

Tip: ナイフツールと削除ツールのように複数のツールが1つのボタンを共有している場合は、ショートカットにshiftキーを追加することで切り替えることができます。

ナイフツール  (Eまたはshift+E) でパスを横断するようにドラッグすると、2つの閉じたアウトラインにすることができます。もし複数のパスを同時に切った場合は、セグメントを繋ぎ直します。

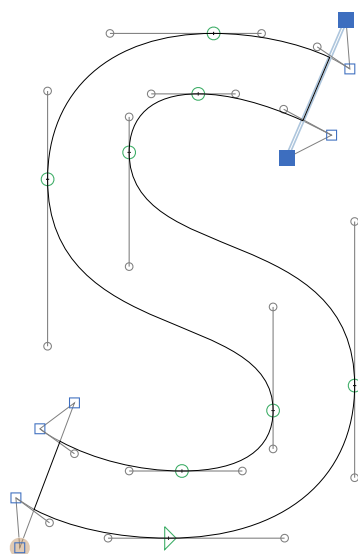
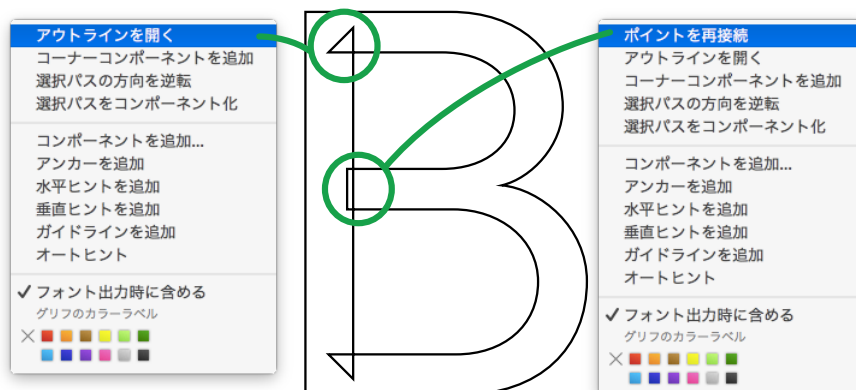


ナイフツールが表示されていない時に選択するには、削除ツール  のボタンを押しっぱなしにしてポップアップメニューから選ぶか、Eまたはshift+Eで選びます。



3.3.11 アウトラインの再接続

「アウトラインを開く」と「ポイントを再接続」の結果。「アウトラインを開く」機能はコーナーポイント(青い点)にのみ有効です。



偶数個のポイントを選択してコンテキストメニューの「ポイントを再接続」を実行すると、選択範囲内の最も近い2つずつのポイントをペアとして、ポイントの接続状態を変更します。コンテキストメニューから「アウトラインを開く」を選択すると、コーナーの延長線上に2つのポイントが生成されパスがループした状態になります。これらの状態であれば、他セグメントの角度やポイントの位置に影響を与えることなく目的のセグメント、ポイントを調整することが可能です。また、マルチプルマスターによる補間の際も適切な結果を得やすくなります。

なお、「アウトラインを開く」のオーバーラップの大きさは「ファイル > フォント情報 > マスター」(command+I)の「垂直ステム」欄に入力した1番目の値の約半分です(オーバーラップを常にステム内に収めるため)。

ループしたコーナーは、3角形の面積が近隣のアウトラインセグメントより小さい場合は透明扱いとなります。そのため、パスの外側にループを作ることも可能です。もしオーバーラップのサイズが許容範囲を超えると、ふたたび可視化されます。これは欧文のsなど、カーブした線の末端を編集する際に便利です。

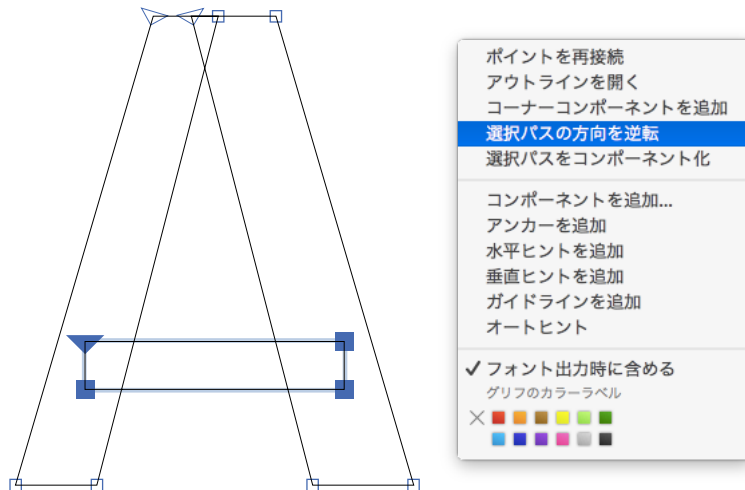
3.3.12 パス方向の管理

パスの開始点は3角形で表示され、その向きによってパスの方向がわかります(緑色はスムーズポイント、青色はコーナーポイント)。任意のオンカーブポイントを選んでコンテキストメニューから「開始点の設定」を選べば、そこを開始点にすることができます。なお、開いたパスではどちらかの末端のポイントに表示される矢印で方向を判断できます。

外周のパスなど黒塗りで表示されるべきパスは反時計回り、カウンター(空洞)のあるパスを手っ取り早く描きたい場合は、方向を気にせず外側と内側のアウトラインを描いてからshift+command+R(パス方向を修正)を適用します。

Tip: ドーナツ状などカウンター(空洞)のあるパスを手っ取り早く描きたい場合は、方向を気にせず外側と内側のアウトラインを描いてからshift+command+R(パス方向を修正)を適用します。

を逆転」を選択すると、すべてのパス方向が逆転されます。



「パス > パス方向を修正」(shift + command + R) を選ぶと、パス選択の有無に関係なく Glyphs は自動ですべてのパスの正しい方向を判別し、必要に応じて修正します。同機能はパスの重なり順も変更し、すべてのパスの開始点を左下のポイントにリセットします。

option を押すと同メニュー項目は「全マスターのパス方向を修正」(option + shift + command + R) に変わります。項目名の通り、この機能は選択中のグリフのすべてのマスターのパス方向を修正します。もしブレースレイヤー (12.7、p.146) やブラケットレイヤー (12.8、p.146) がある場合、これらにも適用されます。この機能はマスターでないレイヤーには影響しません。マルチプルマスターのファイルでは便利な機能です。

マルチプルマスターで補間する際は、パスの重なり順、開始点の位置、そしてパスの方向はどのマスターでも統一されてなければいけません。詳細は「12 マルチプルマスター」(p.139) を参照してください。

3.3.13 極点と変曲点

極点とはハンドルの角度が完全に水平または垂直になるポイントのことです。パスには極点を置くのが慣例となっています。フォント技術の中にはヒンティングなど極点が置かれていることが前提になっているものもありますし、ナッジ移動 (「3.3.2 ポイントとパスの移動」、p.18) でも極点がある方が容易に制御できます。変曲点とはカーブ方向が途中で入れ替わるセグメント (2つの弧からなる波型セグメント) の方向転換点のことです。パスのオフセット (5.2.3、p.56) などのフィルタでは変曲点があった方が美しい結果を得られます。フォントレンダラーの中にはこのようなポイントがない場合、パスが正確に描画されない可能性があります。また、マルチプルマスターで変曲点がない場合、スムーズポイントであるはずのポイントが折れることもあります。なお、Web フォントの製作などではファイルサイズを最小限に保つためにポイント数を少なめに抑えた方がいい場

合もあります。

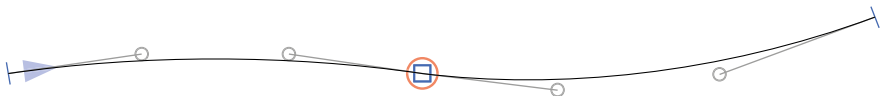
本来ならば極点が存在するべき曲線のセグメント上をペンツール (P) で shift + クリックすることで極部分、つまりアウトラインの上下左右の末端に極点を打つことができます。また、2つの弧からなる波型セグメント (ダブルカーブ) の中間付近を shift + クリックすると変曲点を追加し、2つの弧 (シングルカーブ) のセグメントに分割できます。例えば欧文の S の中央のスプラインと呼ばれるカーブ部分は1つのセグメントで描くことはできませんが、(例: 3.3.11の挿絵)、元々のPostScriptの仕様ではダブルカーブ (S字) は推奨されていないので、それに従ってカーブを分割したい場合に Shift + クリックが有効になるのです。

「パス > 極点を追加」を選ぶと、現在のレイヤーのすべての極部分にポイントが追加されます (パスやセグメントを選択しておく必要はありません)。ただし、あまりにも小さなセグメントができると予想される箇所には、Glyphsは極点を追加しません。このような場合、ポイントの配置は意図的なものであると解釈するからです。逆に極点から微妙に外れているポイントが存在した場合、Glyphsはカーブ形状を保ちつつそのポイントを極点位置に動かそうとします。

極点は「AddExtremes」カスタムパラメータを使ってフォント出力時に追加することもできます。詳細は「17 補足資料」(p.174)の「フィルタ」の章を参照してください。この機能はマルチプルマスターの互換性維持のため、フォント出力時まで極点追加を後回しにしたいときに便利です。

3.3.14 重なったポイント

2個の隣り合ったオンカーブポイントが同一の座標に存在し、結果としてセグメントの長さがゼロになっている場合、それらのポイントは赤で表示されます。これらのポイントは片方が削除可能であり、「パス > パスをクリーニング」(option + shift + command + T) で自動的に削除できます。



3.4 アンカー

3.4.1 合成用と位置合わせ用アンカー

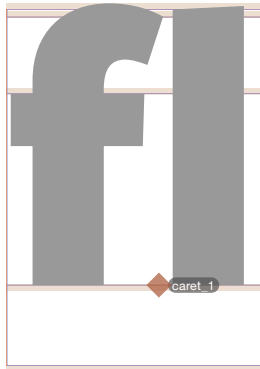
アンカーはGlyphsで様々な役割を果たす特別な種類のポイントで、赤で表示されます。基本的にアンカーはコンポーネント、コーナーコンポーネント、キャップコンポーネントなどの位置合わせ、また発音記号や連綿機能 (「8 図形の再利用」、p.97) の位置合わせに使われます。これらのアンカーには命名規則が存在します。名前についての詳細は「8.1.6 アンカー」

(p.100) を参照してください。「origin」という名前のアンカーがレイヤー内にある場合、その位置が通常のパスのポイントとの整列の基準座標として使われます（「3.3.6 整列」、p.21）。

サードパーティー製のスクリプトやプラグインの中にはアンカーを多用するものもあります。詳細はそれぞれに付属の解説を参照してください。

3.4.2 リガチャー用キャレット

リガチャー（合字）用のキャレット位置とは、合字が表示された時にテキストカーソルが表示されるべき位置のことです。2文字のリガチャーのグリフでは「caret_1」と名をつくアンカーがベースライン上に置かれ、3文字以上のリガチャーでは「caret_2」「caret_3」などが追加されていきます。アンカー名は同名のものが複数存在してはいけませんが数字の順番は関係ありません。GDEFOpenTypeテーブルのLigatureCaretByPosに従って、座標のみを参照してキャレット位置と順番が決まります。



Glyphsがリガチャーとして認識するグリフに「グリフ > アンカーを自動設置」(command+U) を実行すると、必要なアンカーが挿入されます。リガチャーとして認識するグリフとは、s_tやf_lなど、複数のグリフ名がアンダースコア区切りで連なったグリフのことです。詳細は「6.6 グリフ名とUnicode」(p.76) を参照してください。

このハンドブックの執筆時点ではCocoaのテキストエンジンを使用するMacアプリケーションのみがキャレットアンカーを参照します。AdobeとMicrosoftのアプリケーションはこの情報を無視します。

3.4.3 アンカーの追加、編集、削除

編集中のグリフ内で右クリック（またはcontrol+クリック）して表示されるコンテキストメニューから「アンカーを追加」を選択すると、そこに「new anchor」という名前のアンカーが追加されます。追加時にはテキスト入力モードになっており、即座に名前を入力できます。選択解除後でも、あらためてアンカーをダブルクリックして名前を入力できます。また、アンカーを選択すると情報パネルにアンカー名が表示されますので、これ

をクリックすることでも名前の入力が可能です。

Glyphsには各グリフの詳細情報のデータベースがあり、どのグリフにどんなアンカーが必要なかが把握されています。この情報をもとにアンカーを追加するには「フォント > アンカーを自動設置」(command+U)を実行します。optionを押しながらだと項目名が「アンカーをリセット」となり、これを実行すると既に存在するアンカーは破棄されてすべてのアンカーが再配置されます(データベースにないアンカーは削除されます)。

アンカーはクリックで選択できます。選択状態でTabキーを押すと次のアンカー、shift+tabで前のアンカーに移動することができます。また、shiftを押しながらクリックすると複数のアンカーを追加選択できます。アンカーの名前は選択時のみ表示されます。すべてのアンカーを選択したい場合は「編集 > すべてを選択」(command+A)を二度実行します。一度実行するとすべてのパスが選択され、もう一度実行するとコンポーネントとアンカーもすべて選択されます(パスが存在しない場合は一度目の実行でコンポーネントとアンカーが選択されます)。

アンカーはマウスによるドラッグ、またはキーボードの矢印キーで移動することができます。矢印キーで移動する際にshiftキーを押しておくことで10ユニット単位、commandキーで100ユニット単位の移動が可能です。また、情報パネルに座標を入力して移動することもできます。アンカーは通常赤い丸で表示されますが、ベースラインやxハイトに揃っている時は菱形で表示されます。

アンカーを複製するにはoptionを押しながらドラッグしてください(ドラッグ前からoptionを押しておく必要があります)。複製されたアンカーは名前の末尾にアンダースコアが追加されます。これは各レイヤー内でアンカー名が重複してはいけないためです。

アンカーを削除するには、アンカーを選択した状態でdeleteまたはbackspaceキーを押します。

3.4.4 ベースグリフ+発音記号の位置合わせ

Glyphsは結合用発音記号(字幅ゼロ)のアンカー(_topや_bottomなど)と、それに対応するアンカー(topやbottomなど)を持ったベースグリフから、自動的にOpenTypeのmarkフィーチャー(ベースグリフ+発音記号の結合機能)をフォント出力時に生成します。なお、欧文用の結合用発音記号(字幅ゼロ)はグリフ名の最後にcombが付きます(例:acutecomb、macroncomb)。

例えば小文字mに「top」というアンカーがあり、結合用発音記号acutecombに「_top」というアンカーがあった場合、それぞれの位置が参照され、元々フォントには存在しなかったアキュート付きのmがフォント使用時に自動的に合成されて使用できるようになります。

結合用発音記号はそれぞれ固有のUnicode値を持っており、テキスト

入力時にタイプしたりペーストしたりできます。mark フィーチャーがある
と、まずベースとなる文字、次に結合用発音記号を入力することで、フォ
ントのユーザーはあらゆる文字とあらゆる発音記号の組み合わせを入力す
ることができます。

mark フィーチャーと併せて、もしフォント内に idotless や jdotless
がある場合、Glyphs は自動的に ccmp フィーチャー（グリフの結合と解
体機能）もフォント出力時に生成します。詳細は「17.1 フィーチャー自動
生成」(p.174) を参照してください。

3.4.5 発音記号 + 発音記号の位置合わせ

結合用発音記号にアンダースコア付きアンカー（_top など）と普通のアン
カー（top など）の両方がある場合、Glyphs は自動的に mkmk フィー
チャー（mark to mark、発音記号 + 発音記号の結合機能）をフォント出
力時に生成します。これがあると、1つの文字に二重（またはそれ以上）
の発音記号を記すことができます。

例えば acutecomb の下部に通常通り _top アンカーを置いた後、上部
に top アンカーを置いた場合、フォント使用時に「文字 + アク्यूト + も
う1つ発音記号」という文字を入力することができます。

3.4.6 連綿の位置合わせ用アンカー

アラビア文字ではベースラインが常に水平とは限らず、書風によっては徐々
にベースラインが下がっていきます。これをフォントで実現するためには各
文字の連綿の位置を正確に合わせる必要がありますが、そのためには、そ
れぞれの文字の連綿の終わり（語頭形、語中形の左端）に「exit」という
アンカーを、始まり（語中形、語尾形の右端）に「entry」というアンカー
を置きます。テキスト入力時には entry アンカーの位置と exit アンカーの
位置が合わせられ、完全に水平なデザインだけでなく徐々に線が下って
いようなデザインが可能になります。右書きプレビューがオンになって
いると、連綿アンカーの結果は自動的に反映されます。詳細は「3.8.3 書字
方向」(p.36) を参照してください。

3.5 ガイドライン

3.5.1 スマートガイドライン

ポイントをドラッグ移動する際、他のポイントと整列しているか水平垂直
に移動している際に赤い線（スマートガイドライン）が表示されます。一
時的に表示を無効化するには control キーを押してください。

ドラッグ中のオブジェクトは他のオンカーブポイントやハンドルにス
ナップします。これはコンポーネントでも同様で、コンポーネント内のポ
イントにスナップする際はそのポイントが小さく表示され、周囲に赤い丸

が表示されます。これもドラッグ中にcontrolキーを押すことで一時的に無効化できます。

3.5.2 ローカルガイドラインとグローバルガイドライン

右クリック（またはcontrol+クリック）で表示されるコンテキストメニューから「ガイドラインを追加」を選択すると、表示中のレイヤーに水平のガイドラインが追加されます（青い線で表示されます）。なお、2点のポイントが選択された状態で「ガイドラインを追加」を実行すると、その2点を貫通するガイドラインが追加されます。

上級Tip: 計測ガイドラインを素早く追加するには、物差しツール中でドラッグしている時にGを押します。その直線に従って計測ガイドラインが追加されます。なお、control+option+commandキーを同時押しすることでいつでもものさしツールにアクセスすることができます。


この青いガイドラインは現在のレイヤーでしか表示されないローカルガイドラインです。フォント内のすべてのグリフに現れるグローバルガイドラインも存在し、これは赤色で表示されます。既存のローカルガイドラインのタイプをグローバルガイドラインに変更するにはガイドラインを選択し、コンテキストメニューから「グローバルガイドラインにする」を選んでください。グローバルガイドラインをローカルガイドラインに戻す際の動作も同様です。

ガイドラインはクリックで選択できます。選択されるとガイドライン上の小さな丸が塗りで表示されます（選択されていない時は小さな丸の内側が白く表示されています）。ガイドラインを洗濯中にtabキーで他のガイドラインを順に選択、shift+Tabキーで逆順に選択することができます。

ガイドライン上の小さな丸は任意の位置にドラッグ移動することができます。小さな丸をドラッグ移動するとガイドラインは角度を保ったままで移動します。他のポイントやアンカーなどと同様に、小さな丸を矢印キーで移動することも可能です（shift+矢印やcommand+矢印により大きな単位で移動できます）。小さな丸をダブルクリックすると、そこを中心にガイドラインが90度回転します（うまくいかない場合は他の場所をクリックして選択を解除した後、あらためてダブルクリックしてください）。小さな丸以外の場所（線）をクリックして選択した後、線をドラッグすると小さな丸を中心にガイドラインを回転することができます。

optionキーを押しながら小さな丸をドラッグすることでガイドラインを複製することができます。ガイドラインのコピーペーストも可能です。


ガイドラインを選択し、情報パネル（shift+command+I）で位置や角度を数値指定することもできます。x欄、y欄で小さな丸の座標を、Gマークの欄で角度を指定します。なお、初期状態（情報パネルの◀が選択された状態）では左のサイドベアリングが座標指定の基準となっています（xの値が大きくなるほど小さな丸が右方向へ移動）。情報パネルの整列アイコン▶をクリックすると基準は右サイドベアリングとなります（xの値が大きくなるほど小さな丸が左方向へ移動）。これはアラビア文字やヘブライ文字などの右基準の文字のガイドラインが必要な局面、あるいは右サイドベアリングの変更に合わせてガイドラインの位置を追従させたいときに

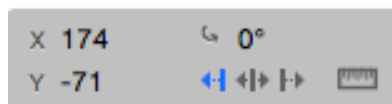
有効です。なお、整列アイコン  をクリックすると基準が左右サイドベアリングの中央となり、xの値を変更しても小さな丸は移動しません。

小さな丸を選択してコンテキストメニューから「ガイドラインをロック」を選ぶとガイドラインの位置が固定され、クリックやドラッグなどの動作に反応しなくなります。ロックされたガイドラインには小さな丸の代わりに錠前のアイコンが表示されます。ロックを解除するにはアイコン上で右クリック（またはcontrol+クリック）によりコンテキストメニューを表示し、「ガイドラインをアンロック」を選んでください。

上から順：未選択と選択状態のローカルガイド、未選択と選択状態のグローバルガイド、ロック状態のローカルガイド、ロック状態のグローバルガイド



ものさしツール選択中にGを押すと計測ガイドライン（距離を計測できるガイドライン）が作成されます。情報パネル右下のものさしアイコン  をクリックしても計測ガイドラインに変換できます。詳細は「3.9 計測」(p.36) を参照してください。



3.5.3 グリフごとの取り消し履歴

編集ビューおよびグリフビューでは、取り消し用の履歴はグリフごとに保管されています。


つまり、フォント全体やマスター単位に関わる操作（グローバルガイドラインの編集など）については取り消しができません（グローバルガイドラインは特定のグリフではなくマスターに帰属するため）。

3.6 グリフの表示

3.6.1 ズーム

編集ビューでは様々な方法でズームイン、ズームアウトができます。

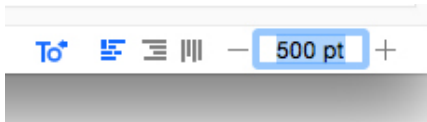
MacBookや外付けのトラックパッドをお使いの方は、2本指によるピンチのジェスチャーが使えます。optionキー + スクロールジェスチャー、あるいはoptionキー + マウスのスクロールでもズームが可能です。

ズームツール  (Z) で編集エリアをクリックするとズームイン、optionを押しながらクリックするとズームアウトできます。また、ズームツールでドラッグすることでエリアをズームアップできます。なお、他ツールを使用している際にcommand+spaceを押すと一時的にズームツールが使用できます。command+space+クリックでズームイン、option+command+space+クリックでズームアウトできます。

Tip: command+spaceのショートカットを有効にするには、Spotlightまたはキーボード切り替えのショートカットを停止しておく必要があります。

「表示 > ズームイン」(option+ プラス) および「表示 > ズームアウト」(option+ ハイフン) によるズームイン、ズームアウトも可能です。

「アクティブなレイヤーにズーム」(command+0) を実行すると、現在選択中のグリフのアセンダーからディセンダーまでが編集ビューに収まるように表示します。「原寸大にズーム」(option+command+0) ではユニットのサイズが画面のピクセルに揃うように表示されます(標準解像度のスクリーンでは1ユニット=1ピクセル、Retina スクリーンでは1ユニット=2ピクセル)。



ウインドウ右下のズームボタンによるズームイン、ズームアウトも可能です。なお、ズームインボタンとズームアウトボタンの間に表示される数字は1000ユニットを何ピクセルで表示しているかを示しています。この数値を変更することで表示サイズを指定することができます。1000と入力した場合、標準解像度のスクリーン(MacOSXは標準スクリーン解像度を72ppiと仮定します)では1ユニットが1ピクセルで表示されることとなります。Retinaスクリーン(スクリーン解像度を144ppiとみなします)であれば1ユニットが2ピクセルで表示されます。

3.6.2 スクロール

トラックパッドを使用している場合、2本指スクロールのジェスチャーで画面をスクロールできます。マウスではホイールを回転させると縦方向のスクロール、shiftを押しながらホイールを回転させると横にスクロールできます。また、編集ビュー右端と下端のスクロールバーをドラッグしてもスクロールすることが可能です(スクロールバーの挙動はシステム環境設定の「一般」で変更できます)。手のひらツール H によるドラッグでもスクロールできます。なお、他ツールを使用している際にspaceを押して一時的に手のひらツールが使用することも可能です(ただしテキスト入力モードのときはspaceバーを押すとスペースが入力されてしまいます。これを避けるにはcommand+spaceを押してからcommandのみを放してください。またはescキーでテキスト入力モードを解除します)。

3.6.3 表示オプション

表示メニューでは編集ビューに表示される内容を変更できます。

「ポイントを表示」ではオンカーブポイント、オフカーブポイントの表示の有無を設定できます。

「メトリクスを表示」ではグリフのメトリクス線の表示の有無を設定できます。

「ヒントを表示」ではPostScriptのヒンティングの表示およびキャップコーナーコンポーネントの表示の有無を設定できます。

「プレビューを黒塗り」では非アクティブなレイヤーの閉じたパスを黒で塗りつぶすか否かを設定できます。

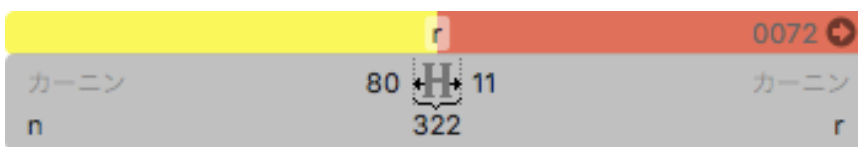
他にも編集ビューを切り替えるサードパーティー製のプラグインが数多く用意されています。これらをインストールすると表示メニューに項目が追加されます。詳細は「16.2 プラグイン」(p.172)を参照してください。

3.6.4 グリフとレイヤーの色

フォントビューでは選択したグリフに対してカラーラベルを設定できますが、編集ビューではグリフのカラーラベルだけではなくレイヤー独自のカラーラベルも設定することができます。編集ビューでレイヤー選択中に右クリック（またはcontrolクリック）でコンテキストメニューを開き、「レイヤー用の色を設定」オプションにチェックマークをつけると、グリフではなくレイヤーのカラーラベルを設定できます。



コンテキストメニューを表示させた状態で option キーを押すと「レイヤーのカラーラベル」オプションに切り替わり、グリフではなくレイヤーに色をつけられます。情報パネルにはグリフのカラーラベルとレイヤーのカラーラベルの両方が表示されます（情報パネルの左がグリフのカラーラベル、右がレイヤー用のカラーラベル）。



また、フォントビューでも同様に2色で表示されるようになります。詳細は「6.3.5 カラーラベル」(p.68)を参照してください。なお、情報パネルの表示は「表示 > 情報パネルを表示」(shift+command+I)です。

3.7 背景レイヤー

編集ビューは背景レイヤーを必ず持っています。背景レイヤーは一時的にパスを保管したり変更内容を比較確認したりするのに便利です（たとえばフィルタの中には「フィルタ > ハッチング」など現在作業中のアウトラインをバックアップのため背景レイヤーに自動的に複製するものもあります）。通常レイヤーで作業している際、背景レイヤーに存在するパスやコンポーネントのアウトラインは薄い灰色で表示されます。

「パス > 背景レイヤーを編集」(command+B) にチェックをつけると背景レイヤーに移動し、背面が薄い茶色で表示されるようになります（環境設定で色は変更可能）。なお、背景レイヤーで作業している間は通常レイヤーのパスが薄い灰色で表示されます。

通常レイヤーを表示している時に「表示 > 背景レイヤーを表示 (shift+command+B)」のチェックを外すと背景レイヤーが表示されなくなります。反対に、背景レイヤーを表示している時に「表示 > 背景レイヤーを表示 (shift+command+B)」のチェックを外すと通常レイヤーが表示されなくなります。

通常レイヤーでパスやコンポーネントなどのオブジェクトを選択し、「パス > 選択内容を背景レイヤーに複製」(command+J) を実行すると、選択したオブジェクトが背景レイヤーに複製されます（背景レイヤーで作業している時は逆に作用します）。この時、それまで背景レイヤーに存在したオブジェクトは削除されます。option を押しながらだと「パス > 選択内容を背景レイヤーに追加」(option+command+J) という項目名になり、それまで背景レイヤーに存在したオブジェクトを削除することなく選択中のオブジェクトを追加できます。

「パス > 背景レイヤーと入れ替え」(control+command+J) を実行すると通常レイヤーと背景レイヤーの内容を入れ替えることができます。また、「パス」メニューを開いている際にoption キーを押すと「背景レイヤーを消去」という項目が現れ、背景レイヤーの内容を空にできます。これらはフォントビューで複数グリフ選択時に実行することも可能です。

編集ビューで作業している時に「レイヤー > 背景レイヤーを割り当てる…」を選択すると、現在開いている他のGlyphs ファイルの当該グリフを背景レイヤーに割り当てることができます。フォントビューで複数のグリフを選択した状態で実行することも可能です。また作業中のGlyphs ファイル自身を選択することで、現在のパスを背景に複製することもできます。

3.8 テキスト入力

Tip: テキスト入力モードからグリフ編集に素早く移るには、esc キーを押してください。

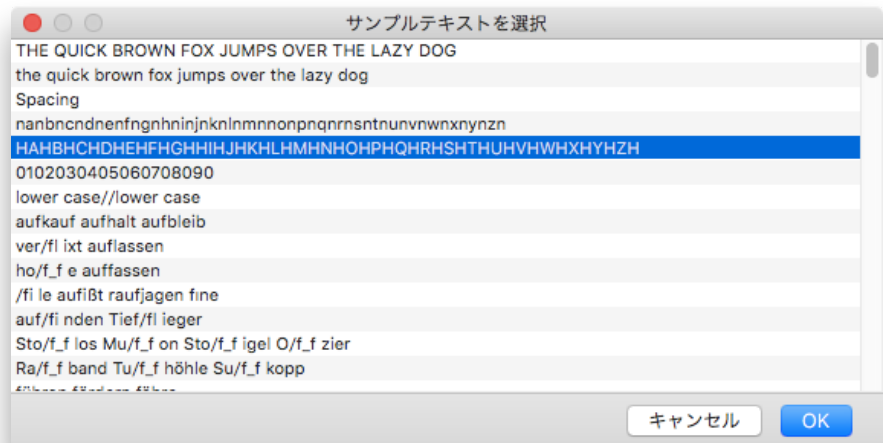
編集ビューはシンプルなテキストエディターとして機能し、単語または文章を見ながらアウトライン編集することが可能です。テキストツール (T) が選ばれている状態で入力します。システム環境設定で選択されている入力モードでの入力、キーボードビューアや文字ビューアからの入力、他ア

アプリケーションからのコピー＆ペースト、「編集 > 特殊文字...」からの入力など、様々な方法による文字入力を受け付けます。音声認識でのテキスト入力も可能です。

編集ビューの横幅の最大幅は「Glyphs > 環境設定」のユーザー設定で設定できます。計算の基準はフォントのユニット数 (UPM) です。例えば1000UPMのフォントを制作する場合、編集ビュー幅を20000と設定すると1行に全角20文字が表示できることになります。

編集ビューのテキスト量が多いとアプリケーションが遅くなることに注意してください。プレビューが開いていると更にスピードが落ちます。制作中のフォントで量の多いテキストを組んでテストしたい場合は、フォントファイルを出力することをお勧めします (出力先はAdobeのFontsフォルダを推奨)。詳細は「3.12.7 Adobe アプリケーションでのプレビュー」(p.46) をご覧ください。

3.8.1 サンプルテキスト



環境設定にはサンプルテキストを数の上限なく保存しておくことができます。サンプルテキストの内容は「Glyphs > 環境設定」の「サンプルテキスト」タブで設定してください (「2.1.3 サンプルテキスト」、p.13)。サンプルテキストを使用する場合は、編集ビューで「編集 > サンプルテキストを選択...」(option + command + F) を選びます。ダイアログで矢印キーの上下 (またはマウスクリック) で選択し、returnキーを押します (OKボタンをクリックでも可)。

「編集 > その他 > 前のサンプルテキスト」(または「次のサンプルテキスト」) を選択すると、ダイアログを開かずに前後のサンプルテキストを使用することができます。これらにショートカットを割り当てたい場合はシステム環境設定から設定を行ってください。

3.8.2 テキストツール

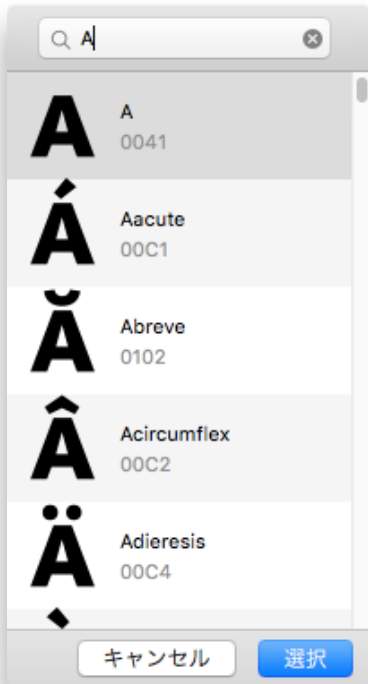
テキストツール (T) 選択中はキーボードで文字入力することができます。

単語や文章の入力はもちろん、改行も可能です。コピーペーストすることもできます。矢印キーや「編集」メニューの各種機能、OSXの「サービス」機能も他のMacアプリケーションと同様に機能します。

カーソルの右側が現在アクティブなグリフです。フォントビュー上の前のグリフまたは後のグリフに移動するにはhomeまたはendキーを押してください。これはフォントビュー上の表示グリフを絞っている場合に便利です。

「編集 > プレースホルダーを挿入」(option+shift+command+P)を選ぶとプレースホルダーを入力することができます。これは現在のグリフと同じものを自動的に表示する仮想グリフのようなものです。例えばnがプレースホルダーで「ononnoon」と表示していた場合、プレースホルダーをkに変更するだけで表示を「okokkook」に切り替えることができます。サイドベアリング編集中に同じグリフを繰り返し表示させたいときなどに便利です。

キーボードで入力することができないグリフや入力方法を知らないグリフを入力する時は「編集 > 検索」のサブメニューから「検索...」(command+F)を選びます。ダイアログ上部の検索欄にグリフ名の一部を入力するとその文字列が含まれるグリフがすべて表示されますので、グリフ名をダブルクリックするかreturnキーを押す、あるいは「選択」ボタンをクリックすることでグリフを決定します。「dier 01」など、スペースで区切った複数のキーワードで検索すると、adieresis.ss01やedieresis.cv01など、キーワードをすべて含むグリフが表示されます。複数のグリフを選択するには、shiftを押しながらクリックで範囲選択するか、commandを押しながらで個別追加します。



3.8.3 書字方向

編集ビューではウィンドウ右下のメニューから左書き、右書き、そして縦書きを選べます。



3.9 計測

Glyphsではいくつかの方法で座標、またポイントやカーブ間の距離を計測することができます。

3.9.1 情報パネル

灰色の情報パネルの表示／非表示は「表示 > 情報パネルを表示」(shift+command+I)で切り替えられます。



情報パネルの右側には現在選択しているものに関連した情報が表示されま
す。オンカーブポイントが1つ選択された場合はその座標が表示されます。





オフカーブポイント（ハンドル）を選ぶと情報パネルにはデルタ値（オン
カーブポイントからの水平距離 Δx と垂直距離 Δy ）および直線距離（L）
が表示されます。

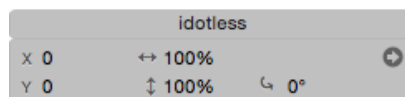
Tip: 簡単かつ正確にステムやボウ
ルの幅を調べるには、該当する2つ
のポイントを選び、選択範囲の幅の
値を確認してください。

オフカーブ、オンカーブに関わらず複数のポイントを選ぶと、情報パ
ネルには選択範囲の幅（ \leftrightarrow ）と高さ（ \updownarrow ）が表示されます。xとyは左の
マトリクスで選択した基準点に基づいた座標となっています。つまりマト
リクスの中心を選ぶと、現在選択している範囲の中央の座標が表示されま
す。一番右の数字はレイヤー内のポイントの総数（□）と、選択している
ポイントの総数（■）です。




情報パネルに表示される値は、ポイント数を除いて数値入力により変更す
ることが可能です（tabキーまたはshift+tabで入力欄を移動できます）。
新しい入力値はreturnキーを押すか、tab（またはshift+tab）で違う入
力欄に移ったときに適用されます。矢印キーの上下でも数値を増減するこ
とができます。この場合、shiftを押しながら矢印キーを押すことで変化量
が10単位になります。なお、数値入力で拡大縮小する場合は、 アイコン
で比率固定の有無が切り替わることと、左のマトリクスで指定した位置が
基準点になることに注意してください。

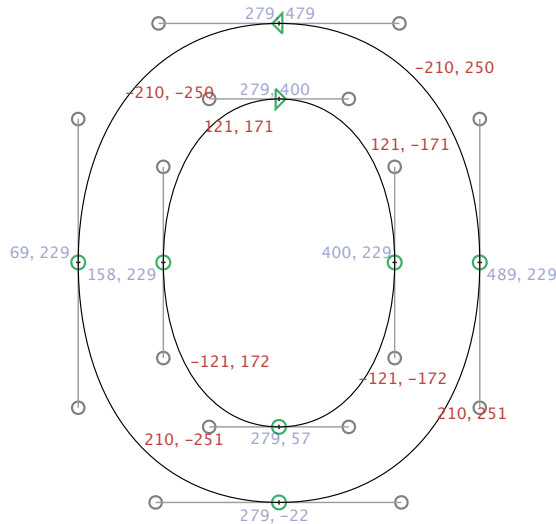
コンポーネントが選択されている場合、情報パネルにはコンポーネン
トのベースグリフ名、ベースグリフからの相対位置、拡大縮小率、角度が
表示されます。右上にある小さな ボタンをクリックすると、現在のグリ
フの左にベースグリフが挿入され、編集可能な状態になります。コンポー
ネントの操作の詳細は「8 図形の再利用」（p.97）を参照してください。



コンポーネントのベースグリフを変更するには、情報パネルに表示されて
いるベースグリフ名をクリックし、表示されたダイアログで新しいグリフ
を選びます。コンポーネントの位置、拡大縮小率、角度を変更するには情
報パネルの各欄に数値を入力します（コンポーネントの位置は自動整列
がオフになっている場合のみ変更可能です。詳細は「8.1.7 自動整列」
（p.102）を参照してください）。

3.9.2 ものさしツール

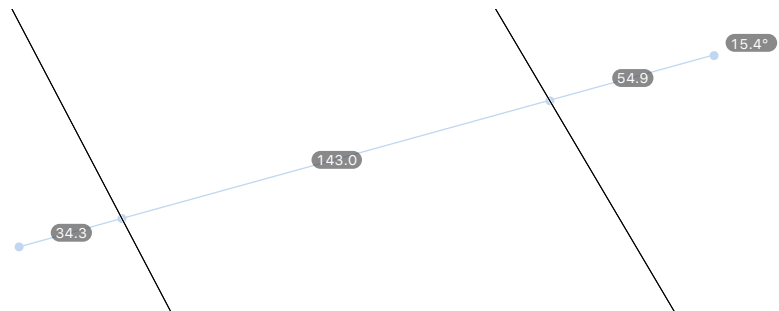
ものさしツール  (L) に切り替えるとすべてのポイントの座標が表示されます。一時的にものさしツールを使用するには control + option + command を同時押しします。



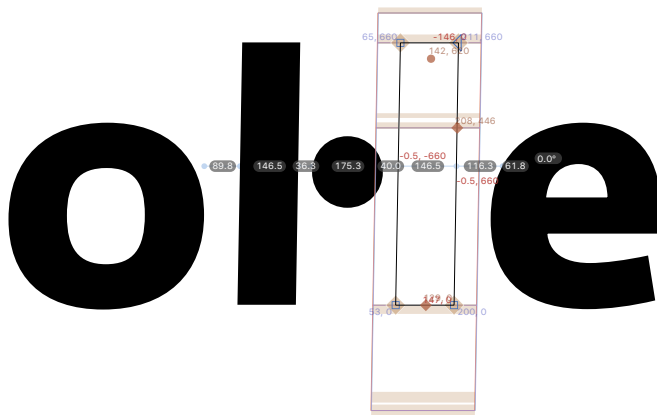
xのデルタ値はフォント情報のイタリック角度を参照します。xデルタが0の場合、イタリック角とセグメントが完全に一致しているということになります。


ものさしツール選択時に表示される青い数字はオンカーブポイントのx座標とy座標、赤い数字は曲線セグメントのオンカーブポイント同士のデルタ値です。これらの値を見ることでアウトラインの対称性などを確認することができます。

ものさしツールでドラッグすると、アウトライン間の直線距離を測ることができます。ドラッグ中のポイントの先には角度が表示されます。shiftを押しながらドラッグすると角度を水平または垂直に固定することができます。



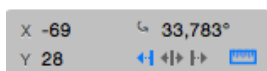
現在のグリフだけでなく編集ビューのすべてのグリフを横断して計測することも可能です。



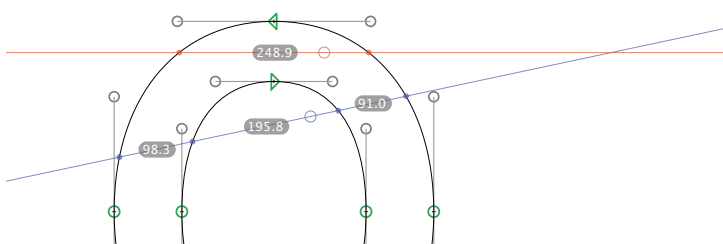
ドラッグ中にGを押すと、計測オプション付きのガイドラインが追加されます。これはガイドラインを選択して情報パネルから計測オプション  を有効化する操作と同じです。

3.9.3 計測ガイドライン


グローバルガイドラインとローカルガイドラインは、いずれも計測ガイドラインにすることができます。ガイドラインをクリックして選択し、情報パネルに表示されるものさしアイコンをクリックしてください。計測ガイドラインは他のガイドラインと同様に角度や位置などを変更することが可能です。詳細は「3.5 ガイドライン」(p.29)を参照してください。



ものさしツールと同様、計測ガイドラインはアウトライン間の距離を表示します。ものさしツールとの違いは、どのツールを選んでいても値が表示されるという点です。

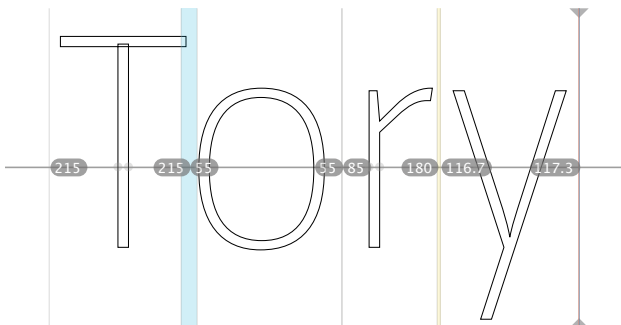


3.9.4 補助線


テキスト入力モードで「表示 > 補助線を表示」を選ぶと補助線を表示することができます。補助線はそのy位置でのグリフの左右端までの距離を表示します。補助線の表示位置(上下位置)はcontrol+option+commandを押しながらクリックまたはドラッグすることで変更できます。ものさしツール  でドラッグ移動することも可能です。

細い灰色の縦線はグリフの左右端を表しています。補助線に表示される数字はグリフの端からアウトラインまでの距離です。

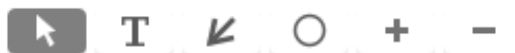
補助線を表示している時、カーニングされている部分は有彩色になります。詰めのカーニングの場合は水色で、空けのカーニングの場合は黄色で表示されます。この色は「Glyphs > 環境設定」のユーザー設定で変更できます。



3.10 注釈

注釈ツール  (A) を使うと作業中のアウトラインにコメントや印を付けることができます。このツールを選択すると情報パネルは注釈関連のツールパレットになります。

注釈ツール選択時に「編集 > すべてを選択」を実行すると、作業中レイヤーの注釈をすべて選択することができます。選択した注釈はキーボードの矢印キーで移動できます (shift を押しながらかで10単位、command で100単位)。また、delete キーで選択した注釈を削除することができます。

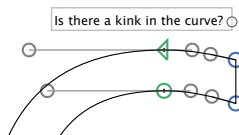


3.10.1 注釈カーソル

注釈ツールパレットの一番左にあるポインタのボタンは注釈の選択ツールです。これを使って注釈を選択、移動、拡大縮小することができます。選択した注釈はdelete キーで削除できます。

3.10.2 注釈テキスト

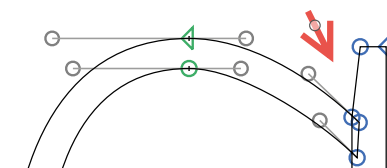
左から2番目にあるTのボタンはテキストツールです。これを選択して任意の場所をクリックすると空のテキストボックスを配置できます。ボックスをダブルクリックすると文字を入力できます。入力が終わったらreturn キーで確定します (確定させなくても他のツールに移って作業を続行することが可能です)。



ボックスの右にあるポイントをドラッグするとテキストボックスの幅を調整できます。ボックスの高さは文章量によって自動的に変化します。

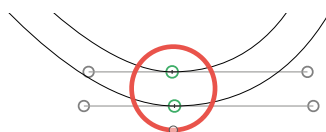
3.10.3 注釈矢印

ボックスの3番目にあるのは矢印ツールです。任意の場所をクリックして矢印マークを追加します。根元にあるポイントをドラッグして角度を変えることができます。長さは変更できません。



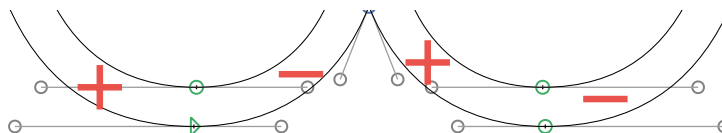
3.10.4 注釈円

4番目は円ツールです。クリックして円を作成します。大きさを変更するには下端にあるポイントをドラッグします。



3.10.5 注釈プラスマイナス

プラスボタンを選ぶとプラス記号 (+) を配置でき、マイナスボタンを選ぶとマイナス記号 (-) を配置できます。多くのデザイナーがカウンターやボウル、またステムなどの加減を説明するためにプラスとマイナスの記号を使うため、この注釈が用意されています。



3.11 画像

3.11.1 画像の追加

グリフのレイヤーには、OSX でサポートされているあらゆる形式 (PDF 含む) の画像を、編集ビューまたはフォントビューにドラッグして配置することがで

きます。また、「グリフ > 画像を追加...」(option+shift+command+I)で表示されるダイアログでは複数の画像を選択し、選択中のマスターの各グリフに配置することも可能です。この場合、用意した画像を目的のグリフに配置するためには画像のファイル名がグリフ名と一致している必要があります。例えば2つの画像のファイル名がThorn.pngとthorn.jpegだった場合、それぞれ自動的に大文字のThornと小文字のthornのグリフに配置されることとなります(ただしOSXのファイルシステムは大文字と小文字の区別を付けないため2つのファイル名をThorn.png、thorn.pngとすると同じフォルダに保存できません。先の例のように異なるファイル形式で保存するか、フォルダを分ける必要があります)。

Glyphs ファイルには画像は含まれず、画像ファイルへの相対パスのみが保存されます。そのため、画像はGlyphsファイルとなるべく近い階層のフォルダに保存しておくのが理想です。画像が移動したり改名されたりなどの理由でパスが不正の場合、画像不明のアイコンが表示されます。



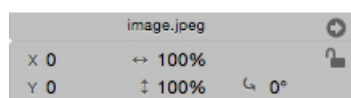
画像の表示/非表示を切り替えるには「表示 > 画像を表示」をチェックします。ただしレイヤーにアウトラインもコンポーネントもない状態では、「表示 > 画像を表示」がチェックされていなくても画像は必ず表示されます。

OpenType フォント出力時に画像ファイルは無視されますが、Apple 方式のカラーフォントを出力する場合は使用されます。詳細は「13.4 Apple カラーフォント」(p.152)を参照してください。



3.11.2 画像の調整

初期状態では、画像はDTPポイントがフォントのユニットと合致するサイズで、左下端が原点に揃うように配置されます。大量の画像を扱う場合は前もって画像サイズを計算して統一しておきましょう(例えば72dpiで1000ピクセル四方の画像は無調整で全角サイズになります)。

画像はドラッグ移動することができます。また、バウンディングボックスをドラッグして、あるいは拡大縮小ツール(S)により画像サイズを変更することができます(shiftキーを押しておくことで縦横比が保持されます)。回転ツール(R)で画像を回転することもできます。パレットの「変形」操作は画像に対しても適用できます。



情報パネルで画像の位置 (x y)、拡大率 (↔ ↓)、角度を数値指定することも可能です。

情報パネル右端の  アイコンをクリックすると元の画像ファイルを Finder 上に表示します。また、 アイコンをクリックすると画像はロックされます。画像の Finder での表示およびロックはコンテキストメニューからでも行えます。ロック解除にはコンテキストメニューの「画像のロック解除」を選択します。

コンテキストメニューの「レイヤー境界に合わせてトリム」を選択すると、グリフのバウンディングボックスをはみ出している部分をトリミング (切り抜き) することができます。

3.12 プレビューとテスト

3.12.1 カーニングのプレビュー

カーニング結果のプレビューは初期状態ではオンです。これを切り替えるにはウインドウ右下のカーニングプレビューボタンを押します。「**T**o」はカーニング表示なし、「**T**o[•]」はカーニング表示あり、「**T**o[•]」サイドベアリング固定です。カーニングなしの設定ではカーニングが一切表示されません。残りの2つではどちらもカーニングを表示しますが、最後のオプション (サイドベアリング固定) ではサイドベアリング値を編集できなくなります。このモードはカーニング作業中に誤ってサイドベアリングを編集してしまうのを避けたいときに有効です。

3.12.2 マスターのプレビュー

編集ビューはマスターのプレビューそのものです。現在選択中のマスターがアンチエイリアス付き、カーニング有効状態でプレビューされます。

マスターが2つ以上ある場合は、それぞれのマスターを表す太さの異なる「n」ボタンがウインドウの左上に現れます。マスターを切り替えるには任意のボタンをクリックするか、command+ マスター番号のショートカットを押してください。例えばマスターが2つある場合は左のマスターがcommand+1、右がcommand+2 になります。



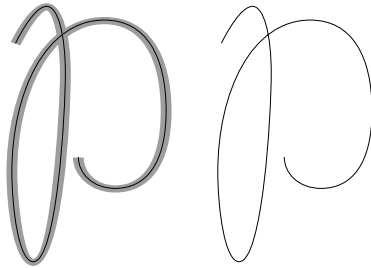
複数のマスターでの作業工程についての詳細は12章「マルチプルマスター」(p.139) を参照してください。

3.12.3 パスのオフセットプレビュー

「表示 > オフセットプレビューを表示」を選ぶと、「カーブをオフセット」フィルタの実行結果をプレビュー (シミュレーション) することができます。

す。「カーブをオフセット」フィルタは開いたパスを後から肉付けしたい場合に便利です。なお、このプレビューを有効化するには、少なくとも1回「カーブをオフセット」フィルタを使う必要があります。このプレビューは現在編集中のグリフのみに適用されます。

オフセットプレビューが適用された小文字のp (左) と適用されていないもの (右)。オフセットの値は前回に「カーブをオフセット」フィルタを実行した時のものが使われます。



3.12.4 OpenType フィーチャーのプレビュー

OpenType フィーチャーをプレビューするにはウインドウ左下の「フィーチャー▼」メニューから任意のフィーチャーを選んでください。フィーチャーは複数同時に有効化できます。フィーチャーが1つでも選択されていると、「フィーチャー」メニューがハイライトされます。ボタンには現在有効化されている4文字のフィーチャー名が表示されます。すべてのフィーチャーを一度に無効化したい場合は一番上の「ー」を選んでください。

初めてフィーチャーをプレビューする場合、またフィーチャーやグリフ名などを変更した場合には、必ず「フォント情報」の「フィーチャー」タブ内でフィーチャーをコンパイルしてください。コンパイル成功しなければフィーチャーはプレビューできません。

特定の言語でのみ作用するフィーチャーを有効化させたい場合は「Localized Forms」のフィーチャーを選択したのちに、フィーチャーのリストの下にある言語のリストから任意の言語を選びます。これを試すには、まずフォント情報の「フィーチャー」タブに「locl」フィーチャーがあり、特定言語専用のルールが用意されている必要があります。

編集ビューでは、Glyphsはグリフ置き換え (GSUB) 系のフィーチャーしか表示することができません。グリフのポジショニング (GPOS) 系のフィーチャーは、各OSやアプリケーションによって挙動が全くが変わるからです。Adobe系のアプリケーションでポジショニング系のフィーチャーをテストするにはAdobeのFontsフォルダーを活用するのがいいでしょう。詳しくは「3.12.7 Adobeアプリケーションでのプレビュー」(p.46)を参照してください。なお、InDesignでプレビューする場合、OpenTypeフィーチャーの挙動はコンポーザーの選択によって変わる場合があることに留意してください。

3.12.5 インスタンスのプレビュー

ウインドウ左下のフィーチャーメニューの隣にあるプレビューボタン (👁)


をクリックすると、ウインドウ下部にプレビューが表示されます。プレビューと作業領域の境目を上下にドラッグするとサイズを変えられます。

「ウインドウ > プレビューパネル」を選択するとプレビューを別のウインドウで表示することができ、マルチモニター環境でも快適に作業することができます。このウインドウでの表示は、マウスカーソルが編集ビューのウインドウ上にある時のみ有効になります。アクティブな編集ビューが存在しない場合は空白になることがありますが、この場合は編集ビューをクリックすることで表示内容が更新されます。

プレビューボタンの右隣のインスタンスメニュー（複数のインスタンスを作成している場合のみ表示されます）をクリックして任意のインスタンスを選ぶと、その場でインスタンスが生成されてプレビューできます。「すべてのインスタンスを表示」を選ぶと、現在のグリフのすべてのインスタンスがプレビューに一覧表示されます。インスタンスはフォント情報で設定できますが、この詳細は「7.3 インスタンス」(p.88)を参照してください。インスタンスメニューの一番上の「一」を選ぶと、現在のマスターのプレビューに戻ります。

プレビューには編集ビューに打ち込んだテキストが一行で表示され、現在選択中のグリフが中央に表示されます（「すべてのインスタンス表示」の場合を除く）。プレビュー領域に表示されている任意の文字をダブルクリックすると、編集ビューでそのグリフが選択された状態になります。なお、プレビュー領域の表示には、カスタムパラメータおよび「4.4.2 特別なレイヤー」(p.50)で説明するブラケットを使った設定が反映されます。

プレビュー領域上でコンテキストメニューを呼び出すと、「アクティブなグリフを常に中央に表示」が選択できます。これがオンの状態では、編集ビューで選択中のグリフが常に表示されるようになります。オフの状態ではGlyphsはプレビュー領域を行頭揃えでなるべく埋めるようにテキストを移動します。

インスタンスメニューの隣にある  ボタンをクリックするとプレビューの色（白黒）を反転することができます。その隣のFボタンをクリックすると、プレビューを上下反転できます。これはスペーシングのテストなどに便利です。また、その隣のスライダーを移動させるとプレビューがぼけて表示され、視力低下時の読みやすさを簡易的に確認することができます。

3.12.6 OS Xでのプレビュー

ここで書かれている通りOS Xには複雑なフォントキャッシュの問題がありますので、未完成のフォントをFontBookにインストールすることはお勧めしません。

OS Xは複雑なフォントキャッシュのメカニズムを導入しているため、単純に元フォントを同名のフォントファイルで上書きするだけでは様々な問題が起きてしまいます。大きな問題としては、変更内容が確認できなかったり、入力した文字と違うものが現れたり、フォントメニューから消えるなどがあります。このようなキャッシュの問題を避けるには、フォントを出力するたびにファミリー名に新しい番号を付けるか（例：MyFont01、

MyFont02、MyFont03...）、ファミリー名に文字を追加するといいでしょう（例：MyFontAA、MyFontAB...）。沢山のバージョンのフォントを出力するときには文字を使った方が正確な順番に並びます。

キャッシュの問題に遭ってしまったときは、すべてのアプリケーションを閉じて出力したフォントすべてをFontBookまたはFontsフォルダからいったん削除してください。次に、「ターミナル」アプリケーションを起動して以下のコマンドを一行ごとに入力しreturnキーを押して実行してください。最初のコマンドでは管理者ユーザーのパスワードが必要です：

```
sudo atsutil databases -remove
atsutil server -shutdown
atsutil server -ping
```

これらのコマンドを実行した後、マシンを再起動します。問題が解決しない場合は、Macをセーフモードで起動してください（command+Sを押しながら起動、Appleロゴが出るまで押したまま）。そしてshiftキーを押しながら再起動、ログインすることでキャッシュを再生成してください。

3.12.7 Adobe アプリケーションでのプレビュー

ポジショニングのフィーチャーやメニューの挙動など、フォントの動作をしっかりと確認するには「/ライブラリ/ApplicationSupport/Adobe/Fonts/」を保存先を選んでOTFフォントを出力するのがいいでしょう。このフォルダにインストールされたフォントはAdobeアプリケーションで即座に使用可能になります。また、同名のフォントを上書き保存した場合、即座に新しいフォントの内容が有効になります。このフォルダはOS Xのシステムのフォントフォルダから独立しているため、保存したフォントはAdobeアプリケーション以外では使えませんが、OS Xのキャッシュ問題を回避するのに非常に有効です。

Fontsフォルダが存在しない場合は、上記のAdobeフォルダ以下にshift+command+Nショートカットで新規フォルダを作ってください。Adobeアプリケーション起動中にFontsフォルダを作った場合は、アプリケーションを再起動してください。


3.12.8 Web ブラウザでのプレビュー

「ファイル > 出力 > ウェブフォント」を選ぶことでウェブフォントを出力することができます。何度も同じフォルダに出力する場合は、出力先フォルダを同ダイアログで指定できます（「出力先」をチェックした後、その下に表示されるパスをクリックして保存先を設定します）。ウェブフォントを使うHTMLテストファイルを用意し、そのファイルをウェブブラウザで開いてフォントを出力するたびにページを更新してください。この方法だとウェブフォントは出力のたびに上書きされますので、ページを更新すれば必ず最新のフォントが使われるはずですが、しかしウェブブラウザの中には

フォントキャッシュを使うものもあり、ページを更新しても最新のフォントが表示されないケースがあるかもしれません。この場合、ほとんどのMacのブラウザはshiftキーをしながら更新（たいていの場合command+R）するとページを完全に更新することができます。

4 パレット

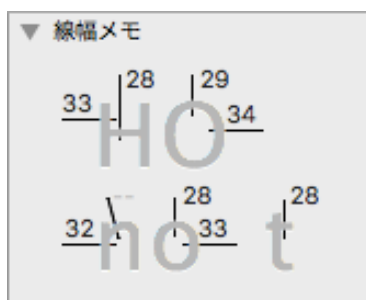
4.1 パレットサイドバー

メインウィンドウ右上の  ボタン、または「ウインドウ > パレット」(option+command+P) でパレットサイドバーを呼び出すことができます。初期状態ではパレットサイドバーは4つのセクションに別れており、各セクション左上角の3角形をクリックすることで展開/収納することができます。プラグインで拡張機能を追加することも可能です。

4.2 線幅メモ

線幅メモセクションは単純にデザイン時のメモとして使うものであり、フォントの設定や各グリフのパスには影響を及ぼしません。

すぐに参照できるよう、ここに重要な線幅の情報を入力しておくといいでしょう。この値はマスターごとに保持されます。なお、タイ文字やアラビア文字など欧文以外の文字に関しては、グリフ選択時に専用の寸法が現れます。



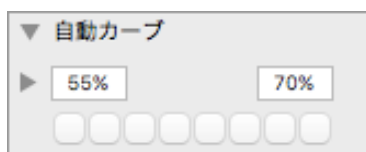
4.3 自動カーブ

自動カーブセクションではセグメントの湾曲率を簡単に設定することができます。ほとんどの場合、初期状態の1列表示で事足りるでしょう。まず2つのテキストフィールドのうち左側に最小値、右側に最大値を入力します。その後、任意のセグメントを定義しているハンドル(1つでも複数でも可)を選択し、8つのボタンのいずれかをクリックするとハンドル角度が固定されたままセグメントの湾曲率が調整されます(セグメントの一方のハンドルのみを選択して実行した場合でも他方のハンドルも調整されます)。別セグメントを定義するハンドルを選択して同じボタンを押すと、そのセグメントにも同じ湾曲率が適用されます。こうすることで、違うセグメントの湾曲率を揃えることができるわけです。

なお、8つのボタンには、テキストフィールドに入力した最小値から最大値までを均等割にした値が割り当てられます。ボタンをクリックする以外に、control+option+1から8のショートカットでもこれらの値を適用することができます。

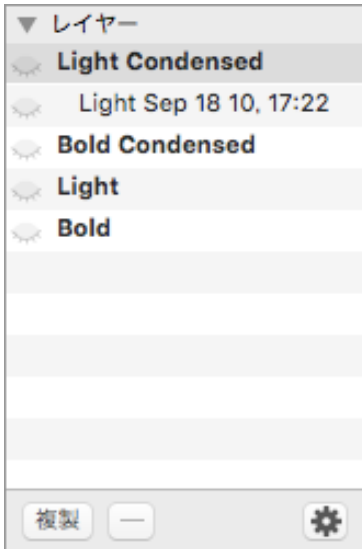
テキストフィールドに入力する数字(%)はハンドルの長さを意味します。100%ではセグメントを定義する2本のオフカーブポイントがぴったり重なり、1%だとハンドルが極めて短い状態になります。また、楕円形の湾曲率にするのであれば56%とします。テキストフィールドに1%より小さい値は入力できません。また入力可能な最大値は100です。セグメントを直線にしたい場合は自動カーブを使うのではなく、ハンドルを選んでdeleteキーで削除してください。

通常、小さいカーブほど高い湾曲率が必要です。例えばoの場合、外側よりも内側のアウトラインの湾曲率を高くしないと同じカーブに見えません。




稀にセグメント両端のハンドルを個別に調整する必要が生じるかもしれませんが。その場合はテキストフィールドの左にある3角形をクリックしてボタンを2次元に展開します。この場合、セグメントを定義している最初のハンドルの長さをx軸で、2つ目のハンドルがy軸で制御できます。ハンドルの順番は選択しているパスの方向で決まります。

4.4 レイヤー



Glyphsには2種類のレイヤーがあります。1つはマスターレイヤー、もう1つは従属レイヤーです。マスターレイヤーはインスタンスの補間やフォントのバリエーションを保管するのに必要です。すべてのグリフは全マスターレイヤーを自動的に保持しており、削除はできません。マスターに関する各種設定は「ファイル > フォント情報 > マスター」でおこないます。「レイヤー」パレットではマスターレイヤーは従属レイヤーとは区別され、レイヤー名が太字で表示されます。


各レイヤーには目玉アイコンが存在し、クリックすることで表示/非表示を切り替えられます。選択されていないレイヤーを表示状態  にすると、そのパスとコンポーネントが選択中のレイヤーの背面に灰色のアウトラインで表示されます。目玉を閉じた状態にすると、そのレイヤーは表示されません。

従属レイヤーのレイヤー名は通常の太さで、関連するマスターレイヤーの下に表示されます。従属レイヤーは個々のグリフに対して作成するもので、無制限に作ることができます。従属レイヤーはデザインのバリエーションを保管するのに使ったり、複数のウエイトを作成する際にブラケットレイヤー、ブレースレイヤーとして使ったり、カラーフォントを作るのに使ったりします。カラーフォントについての詳細は「13 カラーフォント」(p.149)を参照してください。

4.4.1 レイヤーパレットの操作

マスターレイヤー名の編集はフォント情報 (command+I) の「マスター」パネルでおこないます。レイヤーパレットでは編集できません。

新規従属レイヤーを追加するには、フォントビューか編集ビューでグリフを選択した上でレイヤーパレットでマスターレイヤーを選択し、「複製」ボタンをクリックします。その結果、新しい従属レイヤーがマスターレイヤーの下に追加されます。従属レイヤー名はマスターレイヤー名と作成日時を使って自動的に命名されます。従属レイヤー名を変更したい場合はレイヤーパレットに表示されている名前をダブルクリックして書き換えてください。従属レイヤーを消去するには、フォントビューか編集ビューでグリフを選択した上でレイヤーパレットで従属レイヤー名を選択し、「-」ボタンをクリックします。

従属レイヤーを選択した状態で  ボタンを押して「マスターとして使

用」を選択すると、その従属レイヤーの内容（パスやコンポーネント）がマスターレイヤーとなり、現在のマスターレイヤーは従属レイヤーとして入れ替わります。

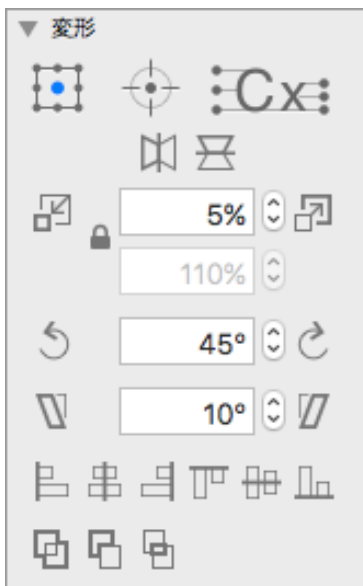
✳ ボタンの「再補間」は選択したレイヤーの内容を他のマスターレイヤーから再生成する機能で、最低でも合計3つのマスターが存在している必要があります。特にブレースレイヤーを使用する際に便利な機能です。詳細は「12.7 ブレースレイヤー」(p.146)を参照してください。

4.4.2 特別なレイヤー

従属レイヤーに特定の名前をつけると特別な機能を持つレイヤーになります。ブレースレイヤー (12.7、p.146) とブラケットレイヤー (12.8、p.146) は、マルチプルマスター環境においてマスターレイヤーと連携します。カラーフォントを作成する際も特別なレイヤー設定が必要になります。詳細は「13 カラーフォント」(p.149)を参照してください。

サードパーティー製のプラグインの中には従属レイヤーを活用するものがあるかもしれません。この場合は、それぞれに付属する資料を参照してください。

4.5 変形



パレットサイドバー最下部のセクションには基本的なパスの変形機能が用意されています。まず最上部で変形の基準点を指定します。選択したパスの相対的な位置、任意の座標、フォントのメトリクス情報（ベースラインやxハイト、キャップハイトなど）に基づく点のいずれか1つを基準点に指定することができます。

基準点の設定以降は、上から順に以下の変形操作が可能です。

- ㊄ ㊅：水平または垂直に反転
- ㊆ ㊇：拡大または縮小（パーセント指定）
- ㊈ ㊉：時計回りまたは反時計回りの回転（角度指定）
- ㊊ ㊋：左または右への傾斜（角度指定）
- 複数のパスの整列

- ブーリアン処理（複数のパスの合成、切り抜き、差分抽出）

パレットサイドバーの変形機能の大半はフォントビューでも編集ビューでも機能します。その時に選択している内容によってパス全体、パスの一部、またはレイヤー全体に作用することになります。複数グリフを選択して適用することも可能です。


変形処理を実行する際はグリッドと分割の単位が反映されますので、オンカーブポイントは自由変形ではなくグリッドに吸着します。詳細は「7.5.1 グリッドの間隔とグリッド細分」(p.94)を参照してください。


4.5.1 変形の基準点

パスの整列とブーリアン処理を除く各変形機能では、基準点に基づいてパス（一部のパス、あるいはすべてのパス）やコンポーネントを変形します。なお、パスやコンポーネントを選択せずに変形を実行した場合、すべてのパスとコンポーネントが変形の対象となります。

基準点はセクション最上部で設定します。

選択したパスの相対的な位置を基準点とする場合は左側のマトリクスを使用します。情報パネルと同様に「右下」や「中心」などの9箇所から基準点を選択してください。

編集ビューの任意の座標を基準点とする場合は、的のアイコン  を選択します。この結果、編集ビュー内に基準点が表示されます。基準点を任意の位置に置くためには、回転ツール (R) または拡大縮小ツール (S) に切り替えて編集ビューの任意の地点をクリックしてください。

フォント情報のメトリクス情報（ベースラインや、ハイト、キャップハイトなど）に基づく点を基準点とする場合、 アイコンから任意の点をクリックして選択します。





4.5.2 反転

Tip: 小文字 n のコンポーネントを水平垂直反転するか180°回転することで小文字 u をすぐに作ることができます。このとき、選択範囲の中心を基準点とするとオーバーシュートによるズレが生じます。これを避けるにはxハイトの中心を基準点とします。

パスやコンポーネントを、基準点を軸として垂直または水平に反転します。コーナーコンポーネントを反転させると、吸着先のコーナーの左右関係が入れ替わります。

4.5.3 双方向適用可能な変形

拡大縮小、回転、傾斜は、それぞれ数値入力欄の左右にボタンがあります。これらは互いに逆方向に作用するボタンです。これにより、一度適用した変形を逆のボタンで戻すことができます。ただしグリッドが有効になっている場合は（「7.5.1 グリッドの間隔とグリッド細分」、p.94）、変形値が四捨五入されるので逆作用ボタンを押しても全く同じ状態には戻りません。

拡大縮小ではx方向、y方向の値を設定することができます。 アイコンをロック  すると、yの値は自動的にxと同じになります。 ボタンを押すと、設定値どおりに拡大します（100だったものを設定値20%で拡大すると120になります）。一方、 ボタンを押した場合は設定値どおりに縮小するわけではなく、拡大ボタンの逆の操作が実行されます（120だったものを設定値20%で縮小すると100となります）。これにより、双方のボタンで拡大の取り消しとやり直しが可能になるわけです。

回転と傾斜ボタンもどんな選択対象にも適用できます。ここでも左右のボタンは互いの逆の働きをします。

4.5.4 整列




整列ボタンはパス（一部のパス、あるいはすべてのパス）およびコンポー

選択範囲を半分に縮小したい場合、100と入力して縮小ボタンをクリックするか、50と入力して拡大ボタンをクリックします。

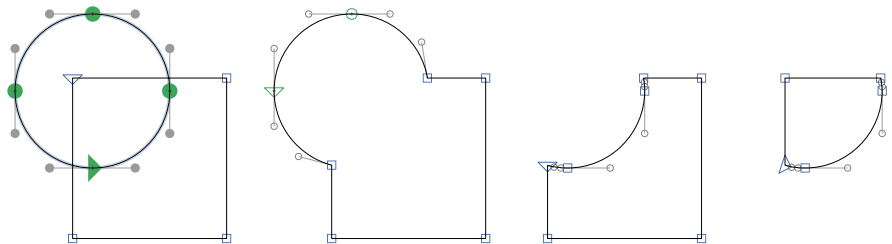
ネットに作用します。パスを部分的に選択した場合はアクティブなポイントが対象になります。なお、整列は基準点ではなく選択内容の相対位置に基づいて機能します。

特定のポイントを整列させたい場合は「パス > 選択ポイントを整列」(shift+command+A)の方が便利です。詳細は「3.3 パスの編集」(p.18)を参照してください。

4.5.5 ブーリアン処理

セクション最下部には複数パスの合体 、切り抜き 、差分抽出  という3つのブーリアン処理が用意されています。合体と差分抽出は選択したパスに作用しますが、何も選択されていない場合はすべてのパスに作用します。切り抜きは、選択したパスが他のパスを切り抜きます。パスが選択されていない場合は、最前面のパス他のすべてのパスを切り抜きます。また切り抜き処理では結果を安定化させるため、切り抜かれるパスが複数ある場合は事前に合成されます。

左から右の順：1. 2つの重なったパスで、ブーリアン処理をしていない状態。2. 合体の結果。3. 切り抜きの結果。4. 差分抽出の結果。



5 フィルタ

5.1 フィルタ

フィルタはレイヤーに適用される画像処理の機能です。グリフ幅を変えるだけのものからアウトラインのラフ処理をおこなうものまで、様々なものがあります。

5.1.1 フィルタメニュー

フィルタの適用先（対象）は状況によって異なります。編集ビューの場合、何も選択していない場合は現在表示しているレイヤー上のすべてが、選択ツールでパスを選択している場合はそのパスが対象となります。また、テキストツールを使用している場合は選択したグリフ（複数選択も可能）が対象となります。フォントビューの場合は選択したグリフ（複数選択も可能）が対象となります。


フィルタは現在表示されているレイヤーにのみ作用し、他のレイヤーには一切影響を与えません。グリフのすべてをマスターに適用させたい場合は各マスターを順に選び、それぞれにフィルタを適用する必要があります。

5.1.2 カスタムパラメータとしてのフィルタ

ほとんどのフィルタは、カスタムパラメータ機能によりフォント出力時にインスタンスに対して自動的に適用することができます。カスタムパラメータはフォント情報（command+I）の「インスタンス」タブで設定します（インスタンスを設けていない場合はパラメータは追加できません）。手動で設定する方法の他、値をコピー＆ペーストする簡単な方法もあります。後者はダイアログが表示されるフィルタのみで可能です。

手動で設定する場合、カスタムパラメータの「+」ボタンをクリックして新規パラメータを追加し、プロパティを「Filter」と設定します。プロパティを追加した時点では値の欄に「新規値」と表示されています。ここにセミコロン区切りで「フィルタ名; 値1; 値2...」などと入力します。なお、フィルタによっては引数名が必要なものもあります。その場合は「フィルタ名; 引数1の名称: 値1; 引数2の名称: 値2」という形式で設定します。詳しくは5.2の各フィルタの項を参照ください。

特定のグリフにのみフィルタを適用したい場合は末尾にincludeを付け、コンマ区切りでグリフ名を列挙します。逆に特定のグリフを対象外にしたい場合はexcludeを付けます。例えば「GlyphsFilterOffsetCurve; 5; 5; 1; 0.5; exclude:a,b,c」などのように入力します（「パスをオフセット、横5、縦5、線幅を与える、位置50%、a, b, cを除いたグリフに適用」という意味です）。セミコロン後などのスペースは必須ではありませんが、入力した方が視認性が向上するでしょう。

ダイアログが表示されるフィルタでは、ダイアログ左下の  アイコンのメニューでカスタムパラメータ用の値をコピーすることができます。コピー後にフォント情報のインスタスタブを開いてカスタムパラメータの欄の空白部分をクリックし、ペーストすることでプロパティ「Filter」が追加され、値も自動的に入力されます。必要に応じてincludeやexcludeを追加設定してください。

フィルタのカスタムパラメータは複数設定することができます。上から順に実行されます。複数設定した場合は必要に応じてドラッグして順番を入れ替えてください。

5.2 標準で付属するフィルタ

5.2.1 マスター互換性を手動修正

「マスター互換性を手動修正」はマスター間で各オブジェクトが正しく対応しているかどうかをグラフィカルインターフェイスで確認し、簡単に修正できるフィルタです。アウトラインやコンポーネントが多く見た目が複雑で「パス > 全マスターのパス方向を修正 (option+shift+ command+ R)」で期待通りの結果が得られないとき、あるいは「表示 > マスター互換性を表示 (control+option+command+N)」では互換性を確認しづらいときに便利です。

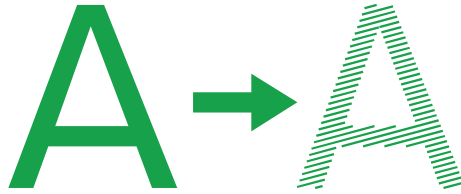
ダイアログでは、行がオブジェクトの順番、列が各マスターを示します。パスは青、コンポーネントは茶色で表示されます。オブジェクトを縦にドラッグすることで順番を変更することができます。作業が完了したら「確定」ボタンを押してください。

なお、ブレースレイヤー (12.7、p.146) 使用時など、複数の補間設定が存在する場合はダイアログを区切る灰色の縦線が表示されます。また、同一のグリフに複数の独立したデザインスペースがある場合など、より高度なマスター設定をおこなっている場合などにも表示されます。例えば長体のライトとボールド間で補間すると同時に平体のライトとボールド間も補間するが長体と平体との間では補間しない、などといったケースです。



5.2.2 ハッチング

「ハッチング」はアウトラインパスから複数の平行線を生成するフィルタです。ダイアログでは平行線の間隔、角度、オフセット幅などを設定することができます。



「開始点」では2つのテキストフィールドでハッチングのストロークを開始する地点の座標 (xとy) を設定します。

「ステップ幅」ではストロークの間隔をユニット単位で設定します。「角度」ではストロークの角度を設定します。「パスをオフセット」ではストロークを太らせる幅をユニット単位で設定します (たとえば40と設定した場合は中心軸の両側に40ずつ太くなり、最終的な幅は80となります)。なお、チェックボックスを外すと直線の開いたパスを得られます。ハッチングフィルタ以降に別の処理をする場合はこちらの方が便利なことでしょう。




「ハッチング」では、基本的には背景レイヤー上のパスを元として、通常レイヤー上に平行線を生成します。もしも背景レイヤーが空だった場合は、通常レイヤー上のパスから平行線が生成されると同時に元のパスは背景に複製されます。元のパスが破棄されず背景に保管されるため、いつでも再利用が可能です。

なお、このフィルタはカスタムパラメータとして利用することができます。手動で設定する場合はカスタムパラメータのプロパティに「Filter」を追加した後、値の欄に以下のフォーマットで入力してください (<>は入力時には除外) :

```
HatchOutlineFilter;OriginX:<x>;OriginY:<y>;StepWidth:<間隔>;Angle:<角度>;Offset:<太さ>
```

パスをオフセットを使用しない場合はOffsetの値を0とするかOffsetの項目を省きますが、初期設定ではフォント出力時に閉じていないアウトラインは消去されます。


具体例（開始点 X=-20、Y=0、ステップ幅=8、角度=25、パスをオフセット=6の場合）：`HatchOutlineFilter;OriginX:-20;StepWidth:8;Angle:25;Offset:6`

ダイアログ左下の  アイコンのメニューでカスタムパラメータ用の値をコピーすることが可能ですので、この表記法を必ずしも覚える必要はありません。

5.2.3 パスをオフセット

「パスをオフセット」はパスの厚みを増やしたり、また逆に減らすことのできるフィルタです。



「横」と「縦」のテキストフィールドでは水平方向と垂直方向のオフセット値を設定します。  アイコンで縦横の値を同じにすることもできます。なお、正の値では外側に、負の値では内側にオフセットします。ただしパス方向が時計回りの場合は正の値では内側に、負の値では外側にオフセットします。

「線幅を与える」オプションがチェックされていない場合、パスはそのままオフセットされます。「線幅を与える」オプションをチェックすると、パスを基準に幅を持ったストロークが生成されます。

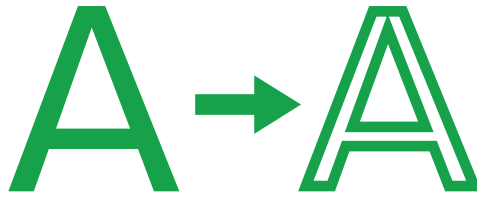


Tip: 文字の骨格の中心線だけを描き、「ストロークにする」を利用して肉付けすることもできます。例えばサンセリフなデザインのアウトラインを手早く描きたいときにとても便利です。

「線幅を与える」オプションをチェックした場合は元のパスからの位置も指定できます。「位置」を50%と設定している場合は元のパスの両側に等しく肉付けされます。0%とするとストロークはパス方向の右側（パス方向が反時計回りなら外側、時計回りなら内側）のみに生成されます。100%では逆に左側のみに作られます。

「縦メトリクスに整列」オプションにチェックを入れると、元のパスがベースラインやxハイトなどに揃っていた場合は生成後のストロークも


揃った状態になるように調整されます。このオプションがオンの場合、「位置」は50%に固定されます。



このフィルタはカスタムパラメータとして利用することができます。手動で設定する場合はカスタムパラメータのプロパティに「Filter」を追加した後、値の欄に以下のフォーマットで入力してください：

OffsetCurve;<横>;<縦>;<線幅を与える>;<位置/縦メトリクスに整列>
<線幅を与える>は0でオフ、1でオンになります。位置は%でなく0.0から1.0までの小数で入力する必要があります。もし「縦メトリクスに整列」を使いたい場合は「auto」と入力します。

具体例：OffsetCurve;10;15;1;auto（横 = 10、縦 = 15、線幅を与える、縦メトリクスに整列）。

ダイアログ左下の  アイコンのメニューでカスタムパラメータ用の値をコピーすることも可能です。

5.2.4 重なったパスを合体

「重なったパスを合体」は選択されたパスの重なった部分を合体するフィルタです。開いたパスや孤立したポイントは除去します。このフィルタにはダイアログはありません。なお、実行時にパスが選択されていない場合は編集集中のグリフ全体に適用されます。

このフィルタを適用する際は、事前にアウトライン方向が意図した通りになっているか確認してください（「3.3.12 パス方向の管理」、p.24）。

最終的にフォントを出力する際は必ずパスが合体されている必要があります。このフィルタを使って合体しない場合、出力時のダイアログ設定で自動的に適用させられますし、初期状態ではそのように設定されています（「15.4 Type1、OpenType、TrueType」、p.163）。

なお、出力時に合体するタイミングを制御したい場合はカスタムパラメータを設定します（パスのオフセットの実行後にパスを合体するなど）。フォント情報のインスタスタブにカスタムパラメータ項目「Filter」を追加し、値を「RemoveOverlap」と入力してください。


Tip: 1部分のみ合体させたい場合はナイフツールで該当箇所をまたいで切り離してください。

5.2.5 ラフ

「ラフ」はアウトラインを大量の小さなセグメントに分割し、与えられた値に従ってポイントをランダムに動かすフィルタです。ダイアログでは「セグメントのサイズ」、「横」および「縦」のブレの最大値、そして「角度」

を設定することができます。控えめな値に設定するとラフなテクスチャを適用したような見た目になるのでこのようなフィルタ名になりました。



このフィルタはカスタムパラメータとして利用することができます。その場合、まずはダイアログ左下の  アイコンのメニューでカスタムパラメータ用の値をコピーします。続いてフォント情報のインスタスタブを開き、カスタムパラメータの欄の空白部分をクリックします。ここにペーストすることでカスタムパラメータ項目「Filter」が追加され、値も自動的に入力されます。これを手動で設定する場合はカスタムパラメータ項目「Filter」を追加した後、値の欄に以下のフォーマットで入力してください：

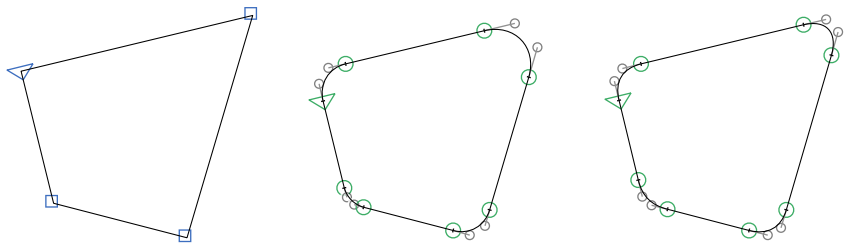
`Roughenizer;<セグメントのサイズ>;<横>;<縦>;<角度>`

5.2.6 角丸

「角丸」は選択した角に対して丸みを与えるフィルタです。何も選択しない状態で適用すると反時計回り（黒塗り）のパスは外向きの角だけが、時計回り（白抜き）のパスは内向きの角だけが丸くなります。


ダイアログの「半径」で角の丸さを指定します。「視覚調整」をチェックすると、鋭角は半径が小さく、鈍角は大きくなるように調整されて、より自然な結果を得られます。

左から右：元の形、通常の角丸、視覚調整された角丸。



このフィルタはカスタムパラメータとして利用することができます。手動で設定する場合はカスタムパラメータのプロパティに「Filter」を追加した後、値の欄に以下のフォーマットで入力してください：

`RoundCorner;<半径>;<視覚調整>`

<半径>を負の値にすると逆向きの角も同時に丸くできます（ダイアログでは負の値を入力できませんが、カスタムパラメーターでは有効です）。<視覚調整>は0でオフ、1でオンとなります。ダイアログ左下の  アイコンのメニューでカスタムパラメータ用の値をコピーすることも可能です。

5.2.7 スマート角丸

「スマート角丸」は、すべての角をきれいな丸にするフィルタです。例えば角ゴシックのデザインを元に丸ゴシック書体を作りたい場合、各アウトラインの先端に適用しなければいけない角丸半径は線の太さによって異なります。スマート角丸は正しい半径を自動的に検出して適用するフィルタです。「編集 > フォント情報」の「マスター」タブの「垂直ステム」の最初の値を読み取って使用するため、事前に設定しておく必要があります。

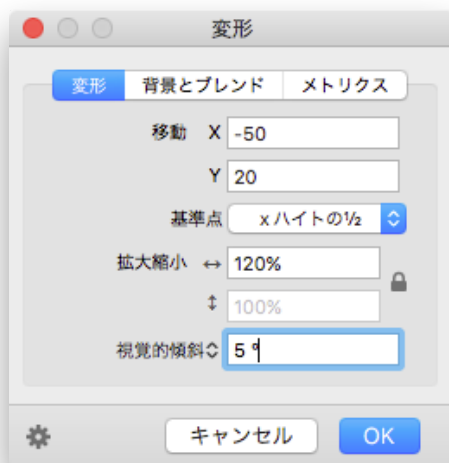
通常、理論的（数学的）にステムの端を丸くすると適用前より短くなったように感じられてしまいますが、このフィルタでは視覚的な長さと同じになるように丸の先端を少し突き出すよう補正します。このフィルタにはダイアログはありません。

「スマート角丸」フィルタをカスタムパラメータとして利用する場合は、カスタムパラメータのプロパティに「Filter」を追加した後、値の欄に「RoundedFont」と入力します。フィルタの通常利用時と同様に「垂直ステム」が設定されている必要がありますが、「RoundedFont;<ステム幅>」と入力することでフォント情報を無視し、こちらの値を使用することができます。

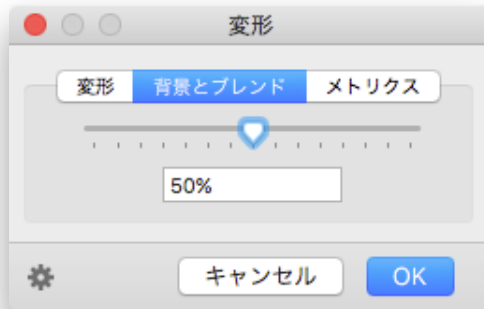
5.2.8 変形

「変形」フィルタには「変形」、「背景とブレンド」、「メトリクス」という3つのタブがあります。

「変形」タブではパスやコンポーネントを水平や垂直方向に移動、拡大縮小、傾斜させることができます。拡大縮小および傾斜の基準点はキャップハイト、キャップハイトの½、xハイト、xハイトの½、ベースラインから選ぶことができます。普通に傾斜させたい場合は「傾斜」を、傾斜した際に起こる線幅の差を補正した傾斜結果を得たい場合は「視覚的傾斜」を選んでください（設定可能な値は-40°から40°まで）。「視覚的傾斜」オプションを利用するには「編集 > フォント情報」の「マスター」タブにて垂直ステムと水平ステムを正しく設定しておく必要があります。



「背景とブレンド」タブでは、スライダーを使って背景とのブレンド（補間）を行うことができます。線の長さなど、パス形状の実験をする際に便利です。背景には互換性のあるアウトラインが配置されている必要があります（「レイヤー>選択パスを背景レイヤーへ」（command+J）により現在のパスを背景に複製すると手早く準備することができます）。



「メトリクス」タブでは、選択したグリフの字幅やサイドベアリングを一度に変更することができます。「相対的」オプションにチェックを入れると、入力値は現在の値との差として扱われます。



「変形」フィルタのカスタムパラメータには「変形」タブと「メトリクス」タブ用の2つのカスタムパラメータ用のフォーマットが用意されています。以下は「変形」タブのもので：

Transformations;OffsetX:<横移動量>;OffsetY:<縦移動量>;
ScaleX:<横の拡大縮小率>;ScaleY:<縦の拡大縮小率>;Slant:<傾斜
角度>;Origin:<基準点の種類>

横移動量、縦移動量の単位はユニットです。

横の拡大縮小率、縦の拡大縮小率の単位は小数ではなくパーセントです（設定値100だと変化なし）。

基準点の種類はメニューの項目の順番に従って0から4までを入力します（0：キャップハイト、1：キャップハイトの½、2：xハイト、3：xハイトの½、4：ベースライン）。

視覚的傾斜の有無は0で「なし」、1で「あり」です。

「メトリクス」のカスタムパラメータのフォーマットは以下のようになっています：

Transformations;LSB:<左サイドベアリング値>;RSB:<右サイドベアリング値>;Width:<レイヤー幅値>;ScaleX:<横拡大率>;ScaleY:<縦拡大率>;Slant:<傾斜角>;SlantCorrection:<視覚的傾斜の有無>;
サイドベアリングとレイヤー幅は+か-を付けると相対値、何も付いていない場合は絶対値とみなされます。

5.3 サードパーティー製のフィルタ

Tip: 以下のサイトでサードパーティー製のフィルタのいくつかを閲覧できます：

glyphsapp.com/extend

追加のフィルタをインストールすることでGlyphsの機能を拡張することができます。`.glyphsfilter`の拡張子を持ったファイルをダブルクリックすると、Glyphsは自動的にフィルタを適切なフォルダにインストールします。その後、再起動すると「フィルタ」メニューに追加されたフィルタ項目が表示されるようになります。サードパーティーのフィルタに関してはそれぞれに付属する説明を参照してください。

フィルタは~/Library/Application Support/Glyphs/Pluginsに格納されます。アンインストールの際には同フォルダーを開いて任意のフィルタを削除してください。なお、「Glyphs > 環境設定」の「アドオン > プラグイン」でプラグインを選び、右クリックで現れる「Finderに表示」を選択することで素早くアクセスすることができます。



「ウインドウ > プラグインマネージャー」から呼び出すプラグインマネージャーで、様々な外部製プラグインをインストール、更新、削除することができます。詳細は「16.2 プラグイン」(p.172)を参照してください。

6 フォントビュー

6.1 グリフの表示

メインウィンドウのフォントビューは、フォント内のグリフセットを一覧表示し、管理するエリアです。複数のタブが開かれている場合、最も左のタブをクリックするかoption+command+1でフォントビューに移動することができます。フォントビューにはグリッドビューとリストビューの2つがあります。

6.1.1 グリッド表示



Unicode値とはUnicodeの汎用の文字コードの値です。コードポイントまたは符号位置とも呼ばれます。通常は先頭に「U+」を付けて表記します。フォントビューに表示されるUnicode値は「U+」が省略されます。

フォントビューをグリッド表示に切り替えるには、ウィンドウ左上にあるフォント情報 **i** の隣のボタンにある2つのボタンのうち左側 **品** をクリックします。



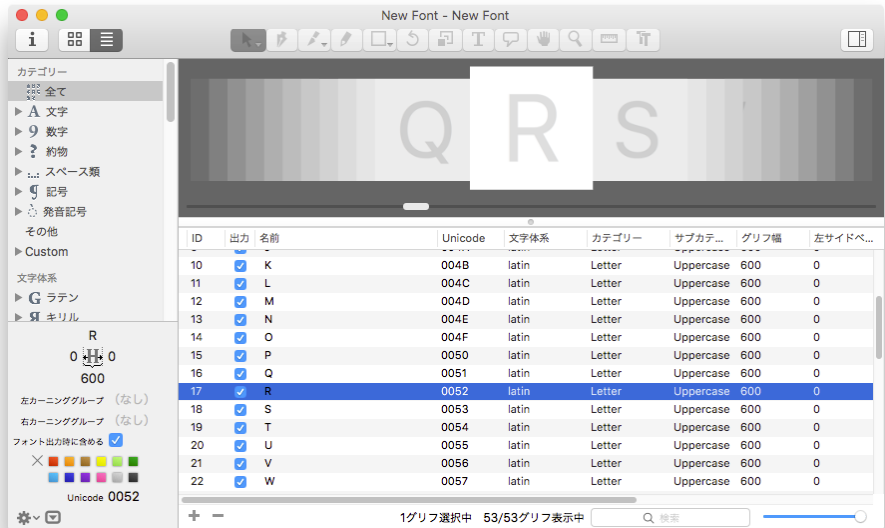
フォントビューの右下にあるスライダーでは表示されるグリフのサイズを変更できます。表示サイズが大きいときにはグリフのアウトライン形状と名前、Unicode値が表示されます。縮小されるに従って文字情報は省略されます。

グリフセルには他にも様々な情報が表示されます。右上に表示される黄色の警告アイコンは、グリフに設定されたメトリクスキーの値が反映されていない状態を表します。詳細は「9.1.2 メトリクスキー」(p.114)を参照してください。グリフに色が設定してある場合は、その通りの色で表示されます。未保存の変更内容があるグリフは薄い灰色でマークされ、ファ

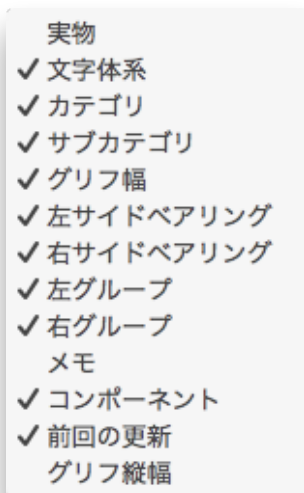


イルを保存すると消えます。左上に表示される赤い3角形は、マルチプルマスターの互換性が取れていないレイヤーがあり、補間に失敗することを表します。アウトラインの互換性の詳細については「12.4 互換性の修正」(p.141)を参照してください。

6.1.2 リスト表示



フォントビューをリスト表示に切り替えるには、ウィンドウ左上のフォント情報 **i** の隣にある2つのボタンのうち右側 **≡** をクリックします。



リスト表示では、それぞれの列がグリフのプロパティを表しています。各列のヘッダー（項目名部分）をクリックすると、そのプロパティの順で整列し、もう一度クリックすると逆順になります（ヘッダー右側に▲が表示されていると昇順、▼だと降順）。ヘッダーを左右にドラッグすると列の順番を変更できます。なお、ヘッダーを右クリックまたはcontrol+クリックするとコンテキストメニューが表示され、以下の各項目を表示するか否かをチェックマークにより設定することができます。

- ・実物：Unicode値のあるグリフの場合、その形状がサンプルとして表示されます。
- ・文字体系：グリフの属する文字体系が表示されます（例：latin、greek、arabic）
- ・カテゴリ／サブカテゴリ：カテゴリは文字（letter）、数字（number）、記号（symbol）など、サブカテゴリは大文字（Uppercase）、小文字（Lowercase）などが表示されます。
- ・グリフ幅／左サイドベアリング／右サイドベアリング：それぞれの横

方向のメトリクス値が表示されます。

- 左グループ／右グループ：それぞれ左と右のカーニンググループ名が表示されます。
- メモ：それぞれのグリフについてのメモが表示されます。この内容は検索フィールドから検索することも可能です。
- コンポーネント：コンポーネントグリフのコンポーネント構成が表示されます。コンポーネントのグリフ名を書き換えて構成を変更することができます。
- 前回の変更：最後にグリフが編集された日時が表示されます。
- グリフ縦幅：グリフの縦方向のメトリクスが表示されます。

6.1.3 グリフの検索

フォントビューでは、ウインドウ左のサイドバーにあるカテゴリ、文字体系、フィルタなどのいずれを選択している場合でも特定のグリフを検索して絞り込み表示できます。グリフを検索するにはウインドウ右下にある検索フィールド (command+F) を使用します。詳細は「6.5.1 検索ボックス」(p.72) を参照してください。

目的のグリフが見つからない場合、何らかのカテゴリを選択しているのが原因かもしれません。その場合、サイドバー最上部の「すべて」を選択してください。検索フィールドに何らかの文字列が入力されており、まだ検索中の可能性もあります。その場合はフィールドを空にしてください。

6.2 グリフセットの管理

6.2.1 新しいグリフの生成

フォントに新しいグリフを追加する方法はいくつかあります。

- 「グリフ > 新規グリフ」(option+shift+command+N) を実行、またはフォントビュー下の「+」ボタンをクリックすることで新規グリフを追加できます。追加した時点ではグリフ名が「newGlyph」(2つ目以降は「newGlyph.001」「newGlyph.002」.....) となっており、グリフ名をクリックして名称を変えることができます。
- グリフを複製する場合は、複製したいグリフを選択して「フォント > 複製」(command+D) を実行します。グリフ名は重複してはいけないので、新しいグリフ名は元のグリフ名の末尾に「.001」「.002」というように番号が振られたものになります。
- 「グリフ > グリフを追加...」(command+shift+G) を実行すると、ダイアログが現れます。ここに任意の文字 (例：あ) を入力し、「生成」ボタンをクリックすることでグリフを生成することができます。スペースや改行でグリフ名を区切ることで一度に複数を生成することが可能です (例：あいうえお)。Unicode のある範囲を一括で追加し

「グリフを追加」は編集ビューでも使用できます。この場合、新しいグリフは現在のグリフの左に追加されます。

たい場合は、半角コロン区切りで最初のグリフと最後のグリフを入力してください（例：uniE000:uniE0FF）。または文字をそのまま入力することもできます（例：Ä:ñ あ:け）。コンポーネントグリフを生成する際、グリフ追加ダイアログに任意のレシピを入力すると、Glyphs標準のレシピではなく入力したレシピを優先します。コンポーネントのレシピについては「8.3.1 コーナーコンポーネント」（p.109）を参照してください。なお、いずれの方法でもフォント内にすでに存在するグリフを重複して生成することはできません。生成しようとした場合はダイアログで警告されます。



Glyphsが認識する全グリフ名のリストは「ウインドウ > グリフ情報」で確認することができます。

- ・「ウインドウ > グリフ情報」でウインドウを開き、グリフを選択して「フォントに追加」ボタンをクリックすることでもグリフを追加できます。複数のグリフを選択して追加することも可能です。
- ・フォントビュー左にあるサイドバーの項目には数字のバッジが付いたものがあり、グリフ総数と生成されたグリフ数が表示されています（例：0/10）。バッジの付いた項目を右クリックすると、まだ生成されていないグリフがすべて表示されます。任意のグリフを選び（command+Aですべてを選択することもできます）、「生成」ボタンをクリックすることでグリフを追加することができます。



6.2.2 ファイル間のグリフのコピー

フォントビューでは任意のグリフ（複数も可）を選択してコピーし、他のファイルにペーストすることができます。コンポーネントグリフをペース

トすると、ペースト先のファイルのグリフ情報と再リンクされます。なお、ペースト先のファイルに同名のグリフが既に存在する場合、上書きを避けるため3桁の数字が末尾に追加されます（例：A.001、A.002、A.003.....）。

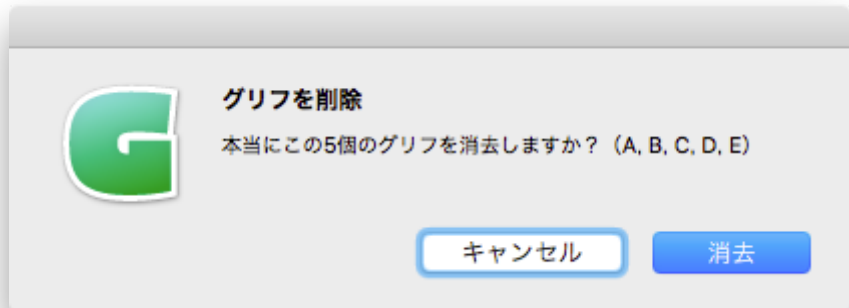
もし元から存在するグリフを上書きペーストしたい場合は「特殊なペースト（optionを押しながら「編集 > ペースト」、またはoption+command+V）」でダイアログを表示し、ペースト方法を設定した上で実行してください。

- ・「グリフを同じ名前の上書き」では、クリップボードにあるグリフと同名のグリフを上書きペーストします。同名のグリフがなければ、同名のグリフを新たに設けた上でそこにペーストします。
- ・「選択したグリフを上書き」では、ペースト先で選択しているグリフを上書きペーストします（ペースト先のグリフ名は保持され、それ以外のデータが置き換えられます）。なお、クリップボードのグリフ数とペースト先の選択グリフ数が異なる場合、余分にコピーされたグリフ、または余分に選択されたペースト先グリフについては無視されます。ダイアログの下部では、コピーしたデータのうち何をペーストするかを選択します。
- ・「すべてのデータ」はパス、アンカー、コンポーネント、レイヤー、グリフとレイヤーの属性、メトリクスを置き換えます。
- ・「現在のレイヤーの中身」は「すべてのデータ」と似ていますが、現在選択されているレイヤーのみ置き換えられます。
- ・「カーニンググループ」は左および右のカーニンググループを置き換えます。別のフォントからグループを移植したい場合に便利です。
- ・「メトリクスキー」はサイドベアリングやグリフ幅のキーを置き換えます。ペースト後には「グリフ > メトリクスを更新（control+command+M）」する必要があります。メトリクスキーの詳細は「9.1.2 メトリクスキー」（p.114）を参照してください。
- ・「左サイドベアリング」「グリフ幅」「右サイドベアリング」はグリフのメトリクス情報を置き換えます。なお、グリフ幅と右サイドベアリングを同時に選んだ場合はサイドベアリング情報が優先されます。



6.2.3 グリフの削除

フォントビューや編集ビューで選択したグリフは「フォント > グリフを削除」(command + delete) で削除することができます。フォントビューではウインドウの左下の「-」ボタンをクリックすることでも削除できます。どちらの方法でも確認のためダイアログが表示されます。



6.3 グリフのプロパティ

ウインドウ左下の ⚙ の隣の 📄 ボタンをクリックするとグリフプロパティの表示/非表示を切り換えることができます。プロパティを編集する際は、単体または複数のグリフを選択した上でおこないます。なお、プロパティのいくつかはコンテキストメニューから編集することもできます。



フォントビューのリスト表示ではグリフプロパティの各項目がリスト内にもコラム表示されており、ここで編集することができます。プロパティの中にはリスト表示でのみ表示されるものもあります。リスト表示では各コラムの情報を昇順または降順で表示できますし、コラムのタイトル部分を右クリックすることで各項目の表示の有無を選択することもできます。

6.3.1 グリフ名

グリフ名は、グリフが1つだけ選択されているときに編集することができます。複数選択したグリフの名前を一度に変更したい場合は「編集 > 検索 > 検索と置換...」(shift+command+F)を使用します。詳細は「6.4.3 グリフ名の一括変更」(p.70)を参照してください。

6.3.2 字幅とサイドベアリング

グリフの字幅や左右のサイドベアリングを編集することができます。複数のグリフについて一度に編集することも可能です。字幅の変更は左側が基準であり、変更すると右サイドベアリングが変化します。

6.3.3 カーニンググループ

カーニンググループを編集できます。複数のグリフについて一度に編集することも可能です。詳細は「9 スペーシングとカーニング」(p.113)を参照してください。

6.3.4 フォント出力時に含める

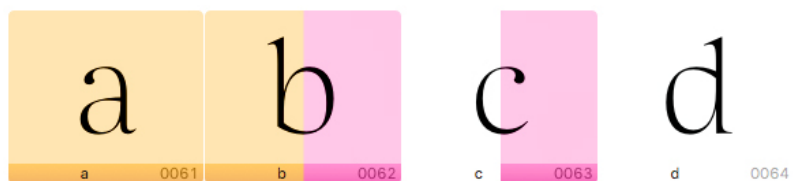
チェックを外すと、そのグリフは出力されません。デザインのバリエーションやセリフなどのパーツをグリフ単位で保管している時など、不要なグリフをフォント出力時に除外したいときに有効です。なお、新規で追加されたグリフは名前がアンダースコア () で始まる場合を除いて初期状態では出力される設定になっています。

6.3.5 カラーラベル

より簡便なソートやフィルタリングのため、もしくは見た目の整理のために、グリフに10色のカラーラベルを付けられます。この色は出力されるフォントには全く影響を及ぼしません。

option キーを押しながらカラーラベルのボタンをクリックすることで、グリフではなく現在表示中のレイヤーにラベルを設定できます。この場合、グリフに設定した色は左半分、特定レイヤーに設定した色は右半分に表示されます。詳細は「3.6.4 グリフとレイヤーの色」(p.33)を参照してください。

左から右：グリフのラベルをオレンジに設定した状態 (a)、グリフのラベルをオレンジ、レイヤーのラベルをピンクに設定した状態 (b)、レイヤーのラベルをピンクに設定した状態 (c)、ラベルが一切ない状態 (d)。



6.3.6 Unicode

Unicode 値はグリフ名に紐付いて定められており、グリフ名を変えずに

Unicode値を変更することはできません。そのためUnicode値は表示専用となっています。しかしフォント情報の「その他の設定」タブにて「標準外のグリフ名を使用」が有効になっている場合はUnicode値を手動で設定することができます。

Unicodeのプライベート領域 (PUA) のグリフに関しては自由に命名することができます。まずは新規グリフのグリフ名にUnicode値をそのまま入力し (4桁の場合は「uniXXXX」、5桁の場合は「uXXXXX」という形式で入力)、そのグリフのUnicode値が定まった後にグリフ名を任意に変更するのが良いでしょう。ただし通常の命名規則はここでも適用されます (「6.6.2 グリフの命名」、p.78)。すでにデータベースに存在するグリフ名と重複しないよう注意してください。もし既存の名前を設定した場合は、そのグリフのUnicode値になります。重複を避けるには「ウインドウ > グリフ情報」でグリフ名を検索して確認してください。なお、グリフ名を変更せずUnicode値を強制的に変更したい場合は「編集 > 選択内容のプロパティを編集」を選択します。詳細は「17.4.2 ファイルごとグリフデータの変更」(p.210) を参照してください。

6.3.7 メモ (リスト表示)

各グリフにはメモを書くことができます。メモの編集・閲覧はリスト表示のみで可能です。メモはフォントビューの検索フィールドで検索できます (ただし検索対象となるのは英数字のみです)。

6.3.8 コンポーネント (リスト表示)

各グリフに使われているコンポーネントを確認することができます。また、項目を直接編集することで使用コンポーネントを変更することができます (例: Amacronのコンポーネント構成「A, macron」を「A, macron. case」に変更)。

6.3.9 リスト表示での表示専用の項目

以下のプロパティはリスト表示でのみで表示されます。これらの項目は編集不可であり、グリフをソートするのに便利です:

- ・「ID」はフォントのグリフの順番に対応しています。これを変更するにはフォント情報の「フォント」タブでカスタムパラメータの「glyphOrder」を編集してください。詳細は「6.5.7 glyphOrderカスタムパラメータ」(p.75) を参照してください。
- ・「実物」はUnicode値のある文字をOSXシステムの標準フォントで表示します。
- ・「文字体系」は各グリフの属する文字で (例: Latin、Greek、Cyrillic)、グリフ名によって決まります。
- ・「カテゴリ」 (例: Letter、Number) と「サブカテゴリ」 (例:

- Uppercase) はグリフ名によって決まります。
- ・「前回の更新」は最後にグリフに変更を加えた日時です。

6.4 複数グリフの処理

6.4.1 グリフの選択

複数グリフを選択するために、さまざまな方法が用意されています。表示中のグリフをすべて選択するには「編集 > すべてを選択 (command+A)」、解除するには「編集 > すべてを選択解除 (option+command+A)」を実行します。選択中のグリフと選択外のグリフを逆転したい場合は「編集 > 選択範囲を反転 (option+shift+command+I)」を実行します。またFinderでのファイル選択と同様、グリフ選択後にcommandを押しながら別のグリフをクリックするとグリフを追加選択、shiftを押しながらクリックだと範囲選択が可能です。

6.4.2 レイヤーコマンドとフィルタ

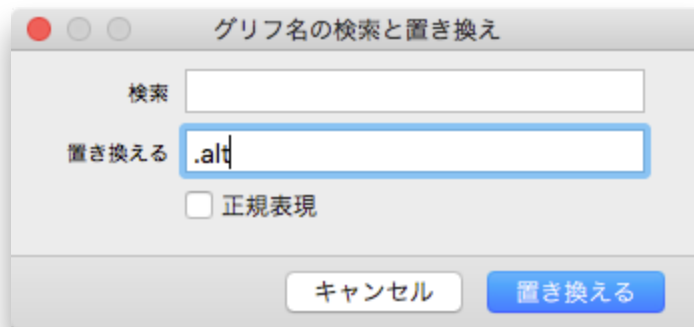
アクティブなレイヤーで選択している複数のグリフに対して、以下の処理を一括で行うことができます：

- ・グリフ > グリフを複製 (command+D)
- ・グリフ > グリフ情報を更新
- ・グリフ > コンポーネントグリフを生成 (opt+shift+cmd+C)
- ・グリフ > コンポーネントを分解 (shift+command+D)
- ・グリフ > 画像を追加... (「6.7 画像」、p.80)
- ・グリフ > メトリクス情報を更新 (control+command+M)、全マスターのメトリクス情報を更新 (control+option+command+M)
- ・グリフ > アンカーを設置 (command+U)、アンカーをリセット (option+command+U)
 - ・パス > 選択内容を背景レイヤーに複製 (command+J)、選択内容を背景レイヤーに追加 (option+command+J)、背景レイヤーを消去、背景レイヤーを割り当てる...、背景レイヤーと入れ替え (control+command+J)
 - ・パス > パス方向を修正 (shift+command+R)、全マスターのパス方向を修正 (option+shift+command+R)
 - ・パス > 座標を整数にする
 - ・パス > パスをクリーニング (option+shift+command+T)
 - ・パス > 極点を追加

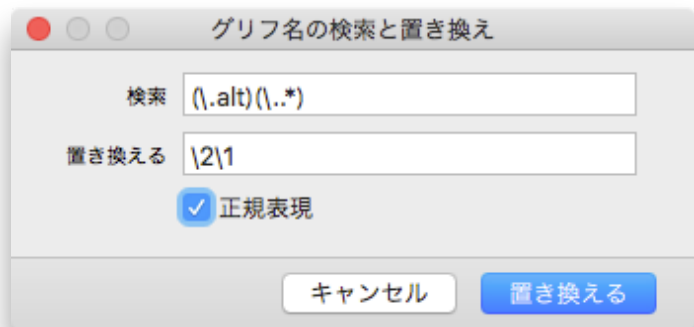
6.4.3 グリフ名の一括変更

「編集 > 検索 > 検索と置換」(shift+command+F) では複数のグリフの名前を一括変更できます。

「検索」フィールドを空にして「置換」フィールドだけに文字列を入力すると、選択グリフの末尾に置き換え文字列が追加されます。



「正規表現」チェックボックスをオンにすると、検索フィールドと置き換えフィールドの両方で正規表現が使えます。正規表現のパターンが無効である場合は赤で表示されます。



例えばグリフ名に沢山のサフィックス（拡張子）が付いていて、その中の「.alt」を末尾に移動させたいとします。その場合、検索フィールドに「\\.alt)(\\..*)」、置き換えフィールドに「\\2\\1」と入力します。正規表現では「\\.」は文字どおりのピリオドで、「.*」は数も内容も不明の、どんなパターンにもマッチする文字列です。「\\1」は括弧で囲まれた最初のパターンを表し、「\\2」は2つ目です。他の具体例は「6.5.1 検索ボックス」(p.72)を参照してください。

6.4.4 フィルタ

付属の全フィルタ、およびサードパーティー製フィルタのほとんどは複数のグリフに一括適用できます。前者に関しては「5 フィルタ」(p.53)を、後者に関してはそれぞれ付属の説明を参照してください。

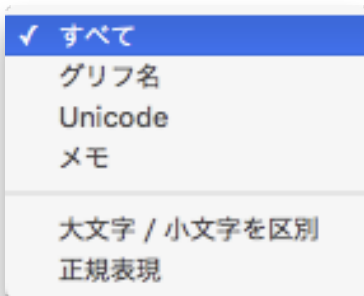
6.4.5 パレット操作

パレット (option + command + P) からは複数のグリフに対して反転、拡大縮小、回転、傾斜の変形処理を一括適用することができます。変形の基準点は設定した位置が使用されます。

ブーリアン処理も複数のグリフに一括適用できます。ただし処理結果が予想しづらいため、複数グリフに適用した場合は処理後によく確認してください。

6.5 絞り込みとソート

6.5.1 検索ボックス



ウインドウ右下の検索ボックス (command+F) は、フォントビューに表示されるグリフを素早く絞り込むのに便利です。虫眼鏡アイコンをクリックすると以下のオプションを選ぶことができます。

- すべて：グリフ名、unicode、メモのすべてが検索対象となります。
- グリフ名：グリフ名だけが検索対象となります。
- unicode：unicode値だけが検索対象となります。
- メモ：メモだけが検索対象となります。
- 大文字/小文字を区別：有効にすると、その名の通り検索内容の大文字と小文字が区別されるようになります。
- 正規表現：検索内容が正規表現パターンとして処理されます。なお、「正規表現」オプションではグリフ名だけが検索対象となります。

正規表現を使用すると曖昧で広義な表現での検索が可能になります。例えば「`[Lldt]caron(\..+)*`」と検索すると、`Lcaron`、`lcaron`、`dcaron`、`tcaron`と、それらのオルタネートまでマッチできます。正規表現に関しては他のウェブサイトなどの参照を推奨しますが、ここでも例をいくつか紹介します。

- **[abc]** a、b、cのいずれか
- **[^abc]** a、b、c以外のいずれか
- **.** アルファベット、数字、記号のどれでも
- **\d** 数字
- **\.** ピリオド
- **a?** 0個または1個のa (aが1個存在するかもしれないが、無くてもマッチする)
- **a+** 1個以上のa (a、aa、aaa、aaaa...)
- **a*** 0個以上のa (aが1個以上存在するかもしれないが、無くてもマッチする)
- **a{3}** 3連続のa (aaa)
- **a{2,5}** 2連続から5連続のa (aa、aaa、aaaa、aaaaa)
- **a.*** aから始まるグリフ名すべて
- **.*a** aで終わるグリフ名すべて
- **.*-.*** ハイフンが途中に含まれているグリフ名すべて
- **.*\..*** ピリオドが途中に含まれているグリフ名すべて
- **.*\d** 数字で終わるグリフ名すべて

・`[^\\d]*\\d{3}` ちょうど数字3桁で終わり、それ以前のどこにも数字がないグリフ名

6.5.2 カテゴリ

カテゴリはGlyphsが定めているグリフのサブセットのみを表示するオプションで、フォントビューの左側上部に表示されます。カテゴリーをクリックして選択すると、フォントビューではそのサブセットのみが表示されます。各カテゴリの隣にある3角形をクリックするとサブカテゴリにアクセスできます。commandを押しながらクリックすると複数のカテゴリを選択することが可能です。例えば「大文字」のサブカテゴリを選んだ後、「小文字」をcommand+クリックするとフォントビューに大文字と小文字の両方が表示されます。

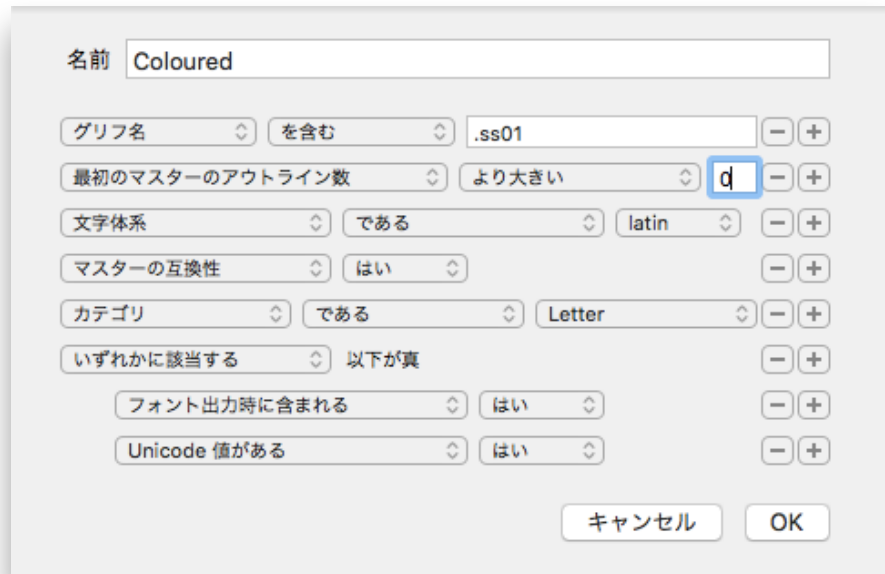
カテゴリを選択した後、command+クリックでフィルタの項目を選ぶと絞り込み表示ができます。例えば「大文字」のサブカテゴリを選んだ後、「mac roman」command+クリックすると、フォントビューにはmac romanのセットに入っている大文字のみが表示されます。


これらの表示を解除してすべてのグリフを表示したい時は、最上部にある「すべて」をクリックしてください。

6.5.3 文字体系

文字体系はカテゴリの下に表示されます。ある文字体系に属するグリフのみを表示させたい場合は、文字体系名（ラテン、キリル、ギリシャなど）をクリックしてください。文字体系名の隣にある3角形をクリックすると、より細かなサブセットを選択することができます。ここでもcommand+クリックでサブセットを複数選択できます。

6.5.4 スマートフィルタ



より特殊な条件で表示させたい場合は、ウインドウ左下にある  ボタンをクリックし「スマートフィルタを追加」を選んでください。スマートフィルタはiTuneのスマートプレイリストのようなもので、以下の項目を用いて絞り込み条件を自由に設定することができます：

- ・グリフ名：任意のグリフ名（グリフ名に特定の文字列を含むかなど）
- ・アウトラインの数：最初のマスターにあるベジェ曲線のアウトラインの数
- ・文字体系：欧文、アラビア文字などの文字体系
- ・カテゴリとサブカテゴリ：グリフ情報データベースに基づいた、グリフのカテゴリ
- ・マスターの互換性：マスターが補間可能な状態か否か
- ・出力フォントに含まれる：出力フォントに含まれるか否か
- ・Unicode 値：Unicode 値が設定されているか否か
- ・コンポーネントを含む：最初のマスター内にコンポーネントが配置されているか否か
- ・ヒンティングされている：手動のヒンティングが施されているか否か
- ・アンカーが設置されている：グリフ内にアンカーが存在するか否か
- ・特殊なレイヤーを含む：ブラケットレイヤーまたはブレースレイヤーがあるか否か
- ・カスタムのグリフ情報を含む：デフォルトのグリフ情報データベースに手動の変更が加えられているか否か（編集 > 選択内容のプロパティを編集）
- ・この名前のコンポーネントを使用：任意のグリフ名のコンポーネントが最初のマスターに配置されているか否か
- ・メトリクスキーがある：グリフのメトリクスにキー（例：「=n + 10」

など) が指定されているか否か

- ・メトリクスキーが未アップデート：グリフに設定されたメトリクスキーの値が反映されているか否か
- ・カーニンググループに属する：カーニンググループに属する否か
- ・カラーラベル：カラーラベルの設定の有無、または設定色
- ・注釈を含む：注釈ツール (A) で設置する注釈がグリフ内に存在するか否か

「+」ボタンをクリックすることで条件を追加できます。初期状態ではAND検索（絞り込み）となっています。optionキーを押しながらボタンをクリックするとOR検索が設定でき、入れ子状態にすることもできます。

フィルタ名をダブルクリックすると再度ダイアログが表示され、設定内容を編集することができます。

6.5.5 リストフィルタ

ウインドウ左下の ✳ ボタンから「リストフィルタを追加」を選択するとユーザー独自のグリフのリストを追加できます。表示されるダイアログに、1行につき1グリフでグリフ名を入力してください。通常、リストフィルタを追加する時にフォントビューで選択していたグリフは自動でリストに追加されます。設定後、リストフィルタを選ぶとリストに入力した順番にグリフが表示されます。

フィルタ名をダブルクリックすると再度ダイアログが表示され、設定内容を編集することができます。

6.5.6 フィルタの管理

フィルタはcommandキーを押しながら複数選択して同時に適用することができます。

✳ ボタンの「フィルタを編集」メニューを選ぶと、選択中のフィルタを編集できます。またフィルタをダブルクリックすることでも編集することが可能です。フィルタを削除したい場合は歯車ボタンの「フィルタを削除」を選んでください。

フィルタはドラッグすることで任意の順番に変更することができます。またフィルタをフォルダにまとめて管理することもできます。フォルダを追加するには ✳ ボタンから「フォルダを追加」を選択してください。

フィルタやフォルダの名前だけを編集したい場合は、選択した状態でreturnキーを押してください。

6.5.7 glyphOrder カスタムパラメータ

カテゴリとフィルタの情報は最終的に出力されるフォントのグリフ順には影響を与えません（リストフィルタでは表示する順番をコントロールできますが、これは出力されるフォントのグリフ順を制御するものではありません）。

せん)。出力されるフォントのグリフ順を制御するには、フォント情報の「フォント」タブで「glyphOrder」カスタムパラメータを追加し、1行につき1グリフでグリフ名を入力してリストを定義します。Glyphsがこのカスタムパラメータを検知すると、フォントビューにおける表示順もこれに従うようになります（リストフィルタを適用しない場合）。

環境設定の「読み込み元のファイルのグリフ名を保持」がオンになった状態でOTFやTTFフォントを開くと、Glyphsはフォント内のグリフ順を維持するため、自動的にこのカスタムパラメータを追加します「2.1.2 ユーザー設定」(p.11)。

6.6 グリフ名と Unicode

6.6.1 グリフ情報のデータベース

Glyphsは以下に示す独自のグリフ情報のデータベースを内包しています。

- ・人間が読みやすいGlyphs独自の標準グリフ名(英語版ではNice name)
- ・製品用のグリフ名（フォント出力時に適用されるグリフ名）
- ・Unicode値（存在しない場合もあります）
- ・使用が想定されるコンポーネント
- ・配置が必要と思われるアンカー
- ・グリフと関連するアクセント（主にアンカー選択時の仮表示用）
- ・グリフの属する文字体系
- ・カテゴリとサブカテゴリ（文字、大文字/小文字、約物など）

「標準グリフ名(Nice name)」には独自の命名規則があり、おおむねAdobe Glyph List (AGL)の仕様に沿っています。しかしAGLではすべてが人間にとって読みやすい名前になっているわけではなく、例えば「uni042F」や「afii10049」などといった覚えにくいグリフ名も混じっています。そこでGlyphsではより分かりやすくするために、文字体系を示す文字を末尾に加えたグリフ名を使用することにしています（文字体系はUnicodeの規格に則ったものです）。例えば「CYRILLIC CAPITAL LETTER A」（キリル文字のA）はGlyphsでは「A-cy」としています。なお、この情報のデータベースはアプリケーションのパッケージ内のXMLファイルに保存されています。GlyphsのApplication Supportフォルダに同じ規則で書かれたXMLを追加することにより、このグリフ情報を拡張または強制変更することも可能です。詳細は補足資料「17.4.1 グローバルなグリフデータの変更」(p.208)を参照してください。XMLファイルを追加、保存した後はGlyphsを再起動すると変更内容が反映されます。グリフ情報の概要を知るには「ウインドウ > グリフ情報」メニューを選んでください。

Adobe Glyph Listの詳細は以下のページをご覧ください：

github.com/adobe-type-tools/agl-aglfn

名前	Unicode	実物	グループ	サブグループ	文字体系	コンポーネント
A-cy	0410	А	Letter	Uppercase	cyrillic	A
Be-cy	0411	Б	Letter	Uppercase	cyrillic	
Ve-cy	0412	В	Letter	Uppercase	cyrillic	B
Ge-cy	0413	Г	Letter	Uppercase	cyrillic	
Gje-cy	0403	Ғ	Letter	Uppercase	cyrillic	Ge-cy acutecomb
Gheupturn-cy	0490	Ғ	Letter	Uppercase	cyrillic	
Ghedescender-cy	04F6	Ғ	Letter	Uppercase	cyrillic	Ge-cy toothright-cy
ghedescender-cy	04F7	ғ	Letter	Lowercase	cyrillic	ge-cy toothright-cy
De-cy	0414	Д	Letter	Uppercase	cyrillic	
le-cy	0415	Е	Letter	Uppercase	cyrillic	E
legrave-cy	0400	È	Letter	Uppercase	cyrillic	le-cy gravecomb
lo-cy	0401	Ë	Letter	Uppercase	cyrillic	le-cy dieresiscomb
Zhe-cy	0416	Ж	Letter	Uppercase	cyrillic	

また、XMLを変更しなくても、そのGlyphsファイルに限ってグリフ情報を強制変更することが可能です。グリフを選んで「編集 > 選択内容のプロパティを編集」(option + command + I) でダイアログで表示し、グリフ名を変更してください(グリフ名を変更するとUnicode値も変更あるいは削除されますので、あらためて元のUnicode値を入力します)。このメニューは複数のグリフ情報を使い分ける必要がある場合に便利です。詳細は補足資料「17.4.2 ファイルごとグリフデータの変更」(p.210)を参照してください。

ピリオド付きサフィックス(拡張子)のついたグリフは、最初のピリオドまでのグリフ名の情報を自動的に受け継ぎます。例えば「X」と「X.alt」はどちらも欧文の大文字Xとして認識され、コンポーネントの自動整列や「グリフ > アンカーを自動設置」したときの必要なアンカーに関する情報も共有されます。もしピリオドをつけずに「Xalternate」と命名すると、Xと関連したグリフだとは認識されません。Glyphsによるグリフのオルタネートやバックアップ、複製などの正しい管理を望む場合は、必ず最初のピリオドまでのグリフ名が一致するようにしてください。

6.6.2 グリフの命名

グリフ名はグリッド表示では各グリフの下、またリスト表示では初期状態で左から3番目の列に表示されます。

グリフ名をクリックすると名前を変更することができます。なお「ä」あるいは「uni00E4」などを入力した場合、Glyphsは自動的に標準グリフ名の「adieresis」に変更します。

また、Glyphsはグリフ名を元に自動的にUnicode値、カテゴリとサブカテゴリ、文字体系の情報を適用します。グリフによってはOpenTypeフィーチャーのコーディングも自動的に準備されます（適用するためには「フォント情報 > フィーチャー」の「自動」ボタンをクリックする必要があります。詳細は「7.4 フィーチャー」（p.91）を参照してください）。

グリフ名の入力には3種類の方式があります。

- ・標準グリフ名。例「Ia-cy」
- ・「uni」または「u」で始まる16進法のUnicode値。例「uni042F」「u10120」
- ・文字を直接入力。例「Я」「あ」

グリフ名の自動変更はファイル単位でオフにすることができます。「ファイル > フォント情報 > その他の設定」で「標準外のグリフ名を使用」にチェックしてください。この場合、コンポーネント生成やOpenTypeフィーチャーなどの自動生成も同時にオフになります。

Unicode値を入力する場合は「uni」で始まる16進法の4桁のコード、例えば「uniE002」などを入力してください。基本多言語面（BMP, Basic Multilingual Pane）以外のUnicode値に関しては「u」で始まる5桁または6桁のコード、例えば「u10015」などを入力してください。

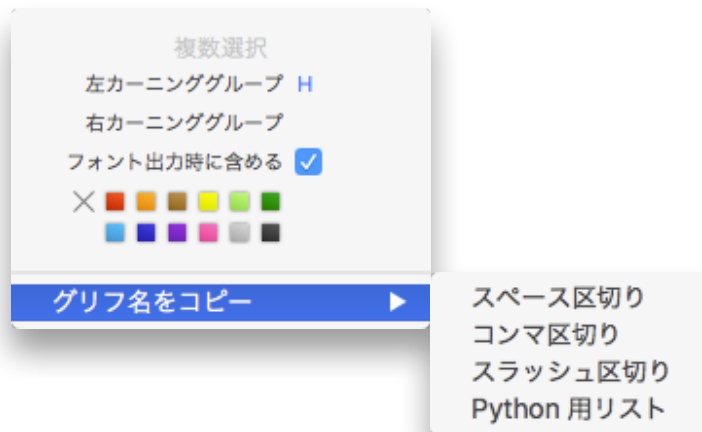
グリフ名は必ずA～Zまたはa～zで始まる必要があります。他に使える文字は数字（0～9）、アンダースコア（_）、ピリオド（.）、ハイフン（-）のみです。スペースなどその他の文字は使えません。使用禁止文字をグリフ名に使おうとすると警告ダイアログが表示されます。この規則の唯一の例外のグリフ名は「.notdef」です。またアンダースコアでグリフ名を始めることも可能ではありますが「出力時に含める」オプションが初期状態でオフになります。

あるグリフのバリエーションのグリフを作りたい場合は、末尾にピリオドで始まるサフィックス（拡張子）を追加してください。例えばnのsmallキャップを作りたい場合は「n.sc」、上付きの5を作りたい場合は「five.sups」と名付けます。サフィックスのなかにはGlyphsが自動的に認識してOpenTypeフィーチャーのコードを自動で生成するものもあります。本ハンドブックの「17章 補足資料」に、標準でサポートされているサフィックスのリストがあります。

複数のサフィックスを使い、OpenTypeフィーチャーの自動生成を高度に制御することもできます。たとえばnのsmallキャップのスタイリ

スティックオルタネートを作りたい場合、通常スモールキャップは先に発動されるため、n.sc.ss01とします（n.ss01.scだとnの異体字のスモールキャップとなり、意味が異なります）。

ハイフンはグリフが属する文字体系を示すために使用します。例えば「alef-ar」はアラビア文字のアレフ、「alef-hb」はヘブライ文字のアレフです。平仮名の「た」の場合は「ta-hira」とします。リガチャーを命名するには、それぞれのグリフ名をアンダースコアで繋げます。例えば「ffl」のリガチャーであれば「f_f_l」とします。なお、文字グループはグリフ名の最後に一度だけ表記します（アラビア文字のラム + アレフのリガチャーは「lam_alef-ar」であり「lam-ar_alef-ar」ではありません）。バリエーションのサフィックスも末尾に一度だけ表記します。サフィックスはリガチャーのグリフ全体に影響します。例えば、ラムの語頭形「lam-ar.init」とアレフの語中形「alef-ar.medi」のリガチャーは「lam_alef-ar.init」とします。「l_a.sc」はスモールキャップの「LA」リガチャーとなります（lだけが小文字にはなるわけではありません）。



フォントビューで任意のグリフを選択し、コンテキストメニューから「グリフ名をコピー」を選ぶと、グリフ名をクリップボードにコピーできます。コピーするグリフの書式には4種類あります：

- ・スペース区切り：グリフ名はスペースで区切られます（例：a adieresis b c）。OpenType フィーチャーのコードを書くときに重宝します。
- ・コンマ区切り：グリフ名はコンマとスペースで区切られます（例：a, adieresis, b, c）。これはカスタムパラメータ用のリストをペーストする時に便利です。また英語として自然な書式なので他の人とのやりとりにも便利です。
- ・スラッシュ区切り：グリフ名の前にスラッシュが付きます（例：/a/adieresis/aacute/b/c）。これはサンプルテキストを作るとき、特に Unicode 値のないグリフのサンプルグリフを作るときに便利です。編集ビューにテキストをペーストするとき、Glyphs はスラッシュ付きのグリフ名をグリフそのものとして処理します。

- ・Python用リスト：Pythonのリスト形式のグリフ名がコピーされます（例：`["a", "adieresis", "aacute", "b", "c"]`）。スクリプトを書くときに便利です。

6.6.3 グリフ名の変更

グリフ名を変更するには、名前をクリックして編集可能になったら新しい名前を入力します。繰り返しになりますが、ピリオドより前の部分を書き換えると、そのグリフの付帯情報まで変更されます。つまりUnicode値、カテゴリ、サブカテゴリ、文字体系、デフォルトのアンカーやアクセントの状態が変わるということです。

グリフ名の検索と置き換え、また多数のグリフの名称変更に関しては「6.4.3 グリフ名の一括変更」(p.70)を参照してください。

6.6.4 CID マッピング

CJK系（中日韓、漢字圏）のOpenTypeフォントは、グリフ名ではなくCID (character identifier) 番号を使ってグリフにアクセスする技術を用いています。Glyphsは標準グリフ名 (Nice name) をCID番号と関連付けるためのマッピングファイルを内蔵しています。そのマッピングファイルに含まれないグリフは、フォント出力時に末尾に追加されますが、ROSの種類によっては不正な結果になるので注意してください。ROSの選択方法についての詳細は補足資料の「17.3 カスタムパラメータ」(p.184)の該当するカスタムパラメータの記述を参照してください。

6.7 画像

6.7.1 画像の追加

フォントビューでも編集ビューと同様、表示中のレイヤーの各グリフに画像をドラッグ&ドロップして画像を追加できます。画像を一括追加するには「フォント > 画像を追加...」を選び、複数の画像を選択してください。各画像のファイル名がグリフ名と一致していればGlyphsは自動的に適切なグリフに画像を配置します。例えばntilde.pngはntildeのグリフに、iacute.pdfはiacuteのグリフに配置されます。なお、ファイル名がグリフ名と一致しない場合は、あらかじめフォントビューで選択しておいたグリフに順番に配置します。

6.7.2 画像の表意

編集ビューと同様、フォントビューでの画像の表示は「表示 > 画像を表示」で設定します。チェックした場合、そのグリフにパスが描かれているか否かに関わらず画像が表示されます。チェックを外した場合、パスが描かれていると画像は表示されません（パスが描かれていなければ画像は表

示されます)。

その他の画像の操作についての詳細は「3.11 画像」(p.41)を参照してください。

7 フォント情報

フォント情報を開くには「ファイル > フォント情報」(command+I) を選ぶか、メインウィンドウ左上の「i」ボタンを押してください。

7.1 フォント

「フォント」タブはフォント全体に適用される情報を格納しています。

7.1.1 ファミリー名

フォントメニューに表示されるファミリー名です。同一のファミリー名を持つフォントはグループ化され、サブメニューで選択できるようになります。ファミリー名にスペースは使用できますが、ASCII以外の文字を使うとフォントの出力に成功しない可能性があります。欧文と数字、簡単な記号のみにすべきでしょう。和名を付けたい場合など、特別な文字をファミリー名に使うにはlocalizedFamilyName カスタムパラメータの利用を検討してください。

Glyphsはファミリー名をもとに出力フォントのファイル名およびOpenTypeのName IDの1(Font Family)、3(Unique font identifier)、4(Full font name)、6(PostScript name)を設定します。CFFベースのOpenTypeフォントではCFFテーブルのFontNameとFullNameもファミリー名を使って設定されます。なお、インスタンスタブでfamilyNameカスタムパラメータを設定している場合はそちらが優先され、ここで定めたファミリー名は無視されます。


Name IDはOTFおよびTTFフォントに内蔵されたOpenType Naming Tableを指します。細かい仕様については以下のページを参照してください：


www.microsoft.com/typography/otspec/name.htm

7.1.2 ユニット数 (UPM)

全角あたりのユニット数 (Units per eM) のことで、1000が標準的な値です。UPMを上げることでデザインのディテールをより細かく表現することができるようになります。OpenTypeは16~16,384UPMに対応していますが、実際は5,000を超えるとInDesignやIllustratorで問題が起こる可能性があります。また、他のアプリケーションでも3,000を超えると問題が出やすいという報告があります。なお、±32,768を超えたところにポイントを置いてはいけません。これらを考えると、高い精度を求める場合はユニット数の設定値を大きくするのではなく、フォント情報の「その他の設定」でグリッドのユニット間隔を0に設定する方がいいでしょう(小数座標が有効になります)。詳細は「7.5.1 グリッドの間隔とグリッド細分」(p.94)を参照してください。

Tip: フォント情報の「その他の設定」でグリッドのユニット間隔が1(小数座標が無効)と設定されたまま、この方法で拡大縮小を行うと、座標値の端数が四捨五入されて劣化が生じる可能性があります。劣化を防ぐために、Scale to UPMパラメータをインスタンスに追加し、フォント出力時のみに適用させる方がいいかもしれません。

テキストフィールドの右にあるアイコンを押し、ダイアログでUPMの数値を変更すると、フォント全体のグリフやメトリクスなどの数値を拡大縮小することができます。UPM設定を1000で保持したままフォントのグラフィックデータを拡大したいのであれば、まずUPMを1000より小さ

い値に設定し、その後に  をクリックして拡大縮小ダイアログに1000と入力してください。2つの値の比率が拡大率となります（たとえばUPMを500と設定した後、拡大縮小ダイアログに1000と入力すると2倍に拡大されます）。同様に、縮小したい場合はUPMを1000より大きい値に設定した後、拡大縮小ダイアログに1000と入力します。


7.1.3 デザイナーとデザイナーの URL

ここにはデザイナーの名前と URL を入力します（http://やftp://などのプロトコル含む）。

例：

デザイナー：Jessica Doe

デザイナーの URL：http://www.example.com

これらの値はOpenType のName ID の9と12に相当します。テキストフィールドの右にある  をクリックするとリンク先がウェブブラウザで開かれ、URLが正しいかどうかを確認することができます。なお、これらのフィールドには日本語を使うこともできますが、基本的には英語またはローマ字の情報を入力することをお勧めします（日本語による情報はカスタムパラメータのlocalizedDesignerNameに入力します）。


7.1.4 製造者と製造者の URL

ここには製造者または販売元の名前と URL を入力します（http://やftp://などのプロトコル含む）。名前には日本語を使うこともできます。


例：

製造者：Super Font Store

製造者の URL：http://www.example.com

これらの値はOpenTypeのName IDの8と11に相当します。テキストフィールドの右にある  をクリックするとリンク先がウェブブラウザで開かれ、URLが正しいかどうかを確認することができます。

7.1.5 著作権

著作権の情報を入力します。デザイナーの名前が先に入力されている場合、テキストフィールドの右の  をクリックして自動で生成することもできます。この値はName ID 0に相当します。

7.1.6 バージョン

フォントのバージョン情報を入力します。この内容を元にGlyphsはバージョン情報（Name ID 5）を生成します。出力されたフォントファイルでは、Glyphsが内部で使用するmakeotfやttfautohintのバージョンも付記されます（例：Version 1.010;PS 001.010;hotconv 1.0.70;makeotf.lib2.5.58329）。これらの追加情報を外して完全に手動でバージョン表記

をしたい場合はversionStringカスタムパラメータを使用します。カスタムパラメータのリストについては補足資料の「17.3 カスタムパラメータ」(p.184)を参照してください。

この自動追加情報はフォントのレンダリングエンジンにとっては有意な情報になり得ます。例えば古いバージョンAFDKOのmakeotfでは欠陥のあるGSUBテーブルを出力するケースが知られています。InDesignはこれらフォントのバージョン情報を認識し、適切に対処します。makeotfのバージョン情報を外すときは、このことに留意してください。

7.1.7 日付

Headテーブルについての詳細は以下を参照してください：

www.microsoft.com/typography/otspec/head.htm

フォントの作成日を入力します。📅をクリックして現在の日時を自動で入力することもできます。この日付がOpenType Font Headerテーブル、別名headテーブルの作成日、修正日として記録されます。

7.1.8 カスタムパラメータ

カスタムパラメータには、より細かい設定を記述することができます。パラメータを追加するには⊕ボタンをクリックして新規値を追加し、右側の⇅からプロパティを選択するかテキストフィールドにプロパティ名を直接入力します。続いて値を設定します。プロパティの種類によってテキストフィールドに直接入力するか、値の欄をクリックして表示されるダイアログで値を設定します。作成したパラメータはコピーし、他のGlyphsファイルにペーストすることもできます。なお、フォントタブのカスタムパラメータではGlyphsの標準挙動をオーバーライドすることができますが、マスターやインスタンスに重複するパラメータがあればそちらにオーバーライドされます（インスタンスのカスタムパラメータが最優先されます）。

フォントに適用できるカスタムパラメータ名はプロパティ右の⇅ボタンをクリックして表示されるリストで確認できます。各カスタムパラメータの詳細については補足資料の「17.3 カスタムパラメータ」(p.184)を参照してください。

7.2 マスター

マスターのセットアップ方法の詳細は以下を参照してください（英語）：

glyphsapp.com/tutorials/multiple-masters-part-1-setting-up-masters

（日本語、動画）：[youtube.com/watch?v=mgLV7rA8s6Y](https://www.youtube.com/watch?v=mgLV7rA8s6Y)

「マスター」タブでは、マルチプルマスターで使われるマスターに関する設定を行います。詳細は「12 マルチプルマスター」(p.139)を参照してください。マルチプルマスター機能を利用しない場合でも、フォント全体に関わる設定として使われます（フォント出力時、最終的に得たいインスタンスについては後述する「インスタンス」タブで設定します）。

「マスター」タブで左下の+ボタンをクリックすると、以下の3種類のオプションが表示されます。

- 空のマスターを追加：何の情報も持たない初期状態のマスターが追加されます（例：ウエイト100、字幅100、カスタム0）。すべてのグリ

フは空になっています。

- 他のフォントから追加：同時に開いている Glyphs ファイルのマスターを選択するダイアログが表示されます。マスターを選ぶと現在のフォントファイルに追加されます。すべてのグリフは読み込み元のフォントに入っていた内容がそのまま引き継がれ、読み込み元に存在していなかったグリフは空になります。
- 選択中のマスターを複製：現在選択されているマスターと全く同一の内容のマスターが追加されます。すべてのグリフは読み込み元のマスターと同じ内容になります。

Tip: option を押しながらマスターをドラッグ移動すると、マスターを手早く複製することができます。

複数のマスターを設定すると、ドラッグで順番を変えられるようになります。最上位のマスターはグリフレベルのヒンティングにとって特に重要な意味があります。詳細は「10.4 グリフレベルの手動ヒンティング」(p.126)と「11.3 手動インストラクション」(p.131)を参照してください。また、スマートフィルタのパラメータには最上部のマスターしか認識しないものもあります。詳細は「6.5.4 スマートフィルタ」(p.74)を参照してください。各マスターのすべての値は補間されますが、同数揃っている必要があり、並んでいる順番も一致しなければいけません。

マスターを shift + クリック または command + クリック で追加選択することで、同時に複数のマスターを編集することができます。

7.2.1 プロポーション：ウエイトとグリフ幅

ポップアップメニューで選択するウエイトとグリフ幅の項目はツールバー左側の n のアイコン、そしてパレット (command + option + P) に表示されるレイヤー名に反映されます。また、両項目の右のフィールドに入力した値はインスタンスで設定する補間値の基準となります。

最終的に出力されるフォントやその補間値は「インスタンス」タブで設定します。もしマスターの内容をそのまま出力したい場合（何も補間しない場合）は、インスタンスの補間値をマスターと同一にしてください。詳細は「12.2 マスターのセットアップ」(p.140)を参照してください。

プロポーション	
ウエイト	Regular 45
グリフ幅	Regular 100
カスタム	0
垂直ステム	45
水平ステム	41

7.2.2 メトリクス

ここではフォントの縦方向のメトリクス（ハイト設定）を行います：

- アセンダー：欧文小文字の最高点 (b d f h k l p などの上端。発音記号は除く)

- ・キャップハイト：大文字の高さ（オーバーシュートや発音記号は除く）
- ・xハイト：小文字の高さ（オーバーシュートや発音記号は除く）
- ・ディセNDER：欧文小文字の最低点（g j p q の下端）

これらは「表示 > メトリクスを表示」が有効になっていると編集ビューでのガイドラインとして全グリフに表示されます。値を決めるときはオーバーシュート（oの上端など、視覚調整のためにはみ出す部分）は考慮しません。例えばxハイトの候補が490、496、502であれば、ベースラインに最も近い値、すなわち490を採用すべきでしょう。

アセンダーとディセNDERは編集ビューでの行送り量にも影響します。これらの値を変えることなく行送り量を変更したい場合は、フォントタブでEditView Line Heightカスタムパラメータを使用してください。

Glyphsはこれらのメトリクスの値からOS/2とhheaテーブルを算出します。もしアセンダーとディセNDERを全マスターで統一していれば、フォント使用時にウエイトを変えただけで行送りが変わってしまう現象を避けることができます。

smallCapHeightカスタムパラメータが設定されていると、スモールキャップグリフの表示時にはxハイトではなくこちらが表示されます。インド系文字、ヘブライ文字、アラビア文字ではshoulderHeightを使うと、同様に専用のハイトが表示されます。

また縦メトリクス情報を正しく入力しておけば、Glyphsはアラインメントゾーンを自動算出することができます。アラインメントゾンの詳細は「7.2.4 アラインメントゾーン」(p.87)を参照してください。

イタリック角度はGlyphsの様々な場面に影響します。イタリック角度を参照する機能は、メトリクス表示時のボディの傾斜角、アンカーの2点間整列、計測ツールなどでのx方向の数値調整、サイドベアリング計算などです。

メトリクス	
アセンダー	720
キャップハイト	700
xハイト	490
ディセNDER	-210
イタリック角度	0°


7.2.3 ステム

垂直ステム、水平ステムに適切な値を入力しておく、これを基準にオートヒンターがステムを探して自動的にヒンティングを施します。マルチブルマスターのファイルの場合はステムが同じ数、同じ順番で設定されていることを必ず確認してください。これが不適切な場合は補間されたインスタンスにおけるステム設定が意図しないものになります。詳細は「10

PostScript ヒンティング」(p.121) を参照してください。

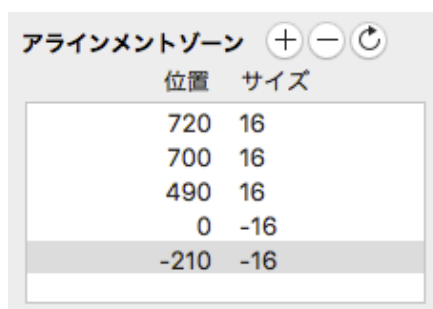
7.2.4 アラインメントゾーン

アラインメントゾーンは、低解像度でフォントが表示される場合に些細な高低差が強調されないよう整列させるために存在します。アラインメントゾーンの設定はフォントが出力される際のオートヒンティング処理に重要な役割を果たします。アラインメントゾーンは低解像度表示時に整列すべきディテールを包含している必要があります。詳細は「10 PostScript ヒンティング」(p.121) を参照してください。

縦メトリクス情報を入力した後は、 をクリックすると自動でアラインメントゾーンが算出されます。自動算出機能では、まずメトリクス情報からそれぞれのゾーンの基準の高さを決めます。smallCapHeight や shoulderHeight のカスタムパラメータが存在する場合は、それも参照されます。その後、特定のグリフの形状をもとにゾーンの幅を決定します：


- ・アセンダーでは、fの上端 +1ユニットをゾーンの上端とします。
- ・大文字では、Oの上端 +1ユニットをゾーンの上端とします。
- ・スモールキャップでは、o.sc (oのスモールキャップ) の上端 +1ユニットをゾーンの上端とします。
- ・xハイトでは、oの上端 +1ユニットをゾーンの上端とします。
- ・ベースラインでは常に-15をゾーンの下端とします。
- ・ディセNDERではgの下端-1をゾーンの下端とします。

計測に用いられるグリフが存在しない場合は、上向きゾーンを16、下向きゾーンを-16、ベースラインゾーンを-15と設定します。重要度の高い文字がきちんとアラインメントゾーン内に収まっているかどうかは十分にチェックしてください。



位置	サイズ
720	16
700	16
490	16
0	-16
-210	-16

7.2.5 カスタムパラメータ

マスター用のカスタムパラメータは、 ボタンをクリックして追加することができます。パラメータはコピーおよびペーストすることも可能です。すべてのマスターに同一のパラメータと値があれば、それは補間されます(例：全マスターにsmallCapHeightが入力されている場合)。マスターのパラメータはフォントタブの設定やパラメータより優先されますが、インスタンスに重複するパラメータがある場合はそちらがさらに優先されます。

「17.3 カスタムパラメータ」(p.184)には各カスタムパラメータの詳細や値のサンプルが載っています。

7.3 インスタンス

インスタンスはフォント出力時に実際に生成されるフォントです。インスタンスを追加するには「インスタンス」タブでウインドウ左下にある + ボタンをクリックし、以下のオプションから任意のものを選択します：

- ・新規インスタンスを追加：Regularと名前の付いたインスタンスを追加します。追加した時点の補間値はウエイト100、字幅100、カスタム軸0です。
- ・全マスターをインスタンスとして追加：フォントに含まれている各マスターと同名で同じ補間値のインスタンスをすべて追加します。

なお、インスタンスを選択した状態で - ボタンをクリックするとインスタンスは削除されます。

shift + クリックまたはcommand + クリックで追加選択し、同時に複数のインスタンスを編集することができます。順番を変更する場合はインスタンスを上下にドラッグします。ただし、ここでの表示順は出力されるフォントには一切影響しません。

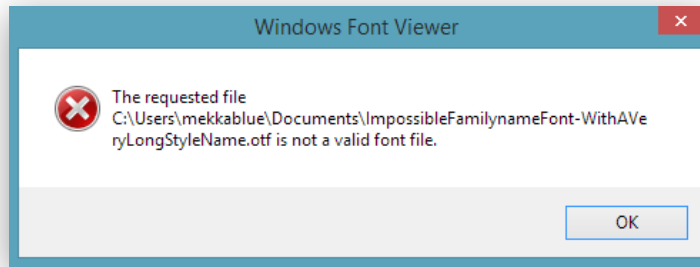
7.3.1 アクティブ

「アクティブ」のチェックボックスは、そのインスタンスをフォント出力時に生成するか否かを設定するものです。アクティブでないインスタンスもサイドバーに表示され、プレビューでも選択できますが、出力時には無視されます。

7.3.2 スタイル名

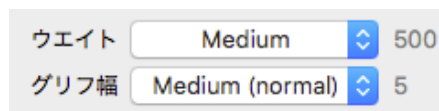
ここに入力するスタイル名はInDesignなどフォントを使用するアプリケーションのフォントメニューにそのまま表示されることとなります(例：Bold CondensedやLight Italic)。スペースは使用できますが、ASCII以外の文字を使うとフォントの出力に成功しない可能性があります。和名を付けたい場合など、特別な文字をファミリー名に使うにはlocalizedFamilyNameカスタムパラメータの利用を検討してください。カスタムパラメータの詳細は「17 補足資料」(p.174)を参照してください。

環境によっては長いスタイル名を嫌うものもあるため、スタイル名は短い方が安全です。Microsoft Windowsはファミリー名とスタイル名が合計20文字を超えると不正なフォントと認識する可能性があります。このような場合、スタイル名には省略した名称を入力し、preferredFamilyNameとpreferredSubfamilyNameカスタムパラメータにフルネームを入力することを検討してください。詳細は「17.3 カスタムパラメータ」(p.184)を参照してください。



7.3.3 ウェイトとグリフ幅

「ウェイト」と「グリフ幅」のドロップダウンメニューでは、ウェイトとグリフ幅のクラスを選択することができます。これらの値はアプリケーションがフォントのリストを適切な順番で並べるために使われたり、フォントの自動選択、あるいはウェブページのスタイルシートでフォントを指定する場面などで使われます。



ウェイトのメニュー項目には値（ドロップダウンメニューの右に示される数値）が同じものがあります。たとえばThin、ExtraLight、UltraLightはどれも値が250になっています。このような場合、weightClassカスタムパラメータを使って、より詳細に定義する必要があります。詳細は「17.3 カスタムパラメータ」(p.184)を参照してください。

Microsoft製アプリケーションの中には、ウェイトクラスの値が250を下回ると強制的に機械的ボールドを適用するものがあります。また、多くのブラウザはウェイトクラスの最初の桁しか参照しません。

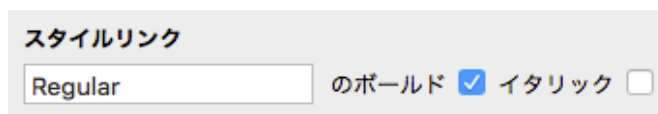
ウェイトとグリフ幅のクラスはどちらもAdobeアプリケーションのフォントメニューの表示順に影響します。まず同じグリフ幅がグループ化され、次にウェイト順に並べられることになります。

例

- Condensed Light
- Condensed Regular
- Condensed Bold
- Light
- Regular
- Bold
- Extended Light
- Extended Regular
- Extended Bold

7.3.4 スタイルリンク

スタイルリンクでは同じファミリー（同じファミリー名を持ったフォント）の他のインスタンスにレギュラー、イタリック、ボールド、ボールドイタリックの関係を紐付けることができます。アプリケーションによっては太字や斜体のボタン、あるいはキーボードショートカットなどを使ってフォントを指定するものがありますが、これは自動でフォントを判断する機能ではありません。この機能を使用するためにはデザイナーがあらかじめ紐付けを設定しておく必要があります。フォント同士を紐付けるには元となるインスタンス名を入力し、ボールドまたはイタリック、または両方のチェックボックスをクリックします。



Adobe アプリケーションはフォントスタイルのメニューにファミリー内の全フォント（ファミリー名が共通しているすべてのフォント）を表示し、shift+command+Iでイタリックのリンクが設定されているフォント、shift+command+Bでボールドのリンクが設定されているフォントを呼び出すようになっています。

テキストエディットなどのMacアプリケーションではcommand+Iとcommand+Bが、それぞれイタリックとボールドのショートカットになっています。

Microsoft Windowsのアプリケーションは1つのフォントファミリーにつき4つまでのスタイルしかサポートしません（Regular, Italic, Bold, Bold Italic）。これらは例えばOfficeのツールバーのIボタンやBボタンをクリック（またはCtrl+IとCtrl+B）することでアクセスできます。もしスタイルリンクが設定されていない場合はフォントメニューに個別のファミリーとして表示されてしまいます。この場合、アプリケーションによっては擬似ボールドや擬似傾斜が使われてしまう可能性があります。そのような状況を鑑みて、以下の手法をお勧めします。

- Bold、Italic、Bold Italicは必ずRegularとリンクさせる。他のフォント同士でボールドのリンクを作らない。あるグリフ幅クラスの中で、常にこの4フォントの対応関係を作り、それ以外にボールドのリンクを貼らない。
- 他のItalicは同じウエイトのローマン（正立体）とリンクさせる。例えばMedium ItalicはMediumとリンクさせる。その際、ファミリー名は必ず同じにする。

デザイナーによってはLightとSemiboldをリンクしようとするかもしれませんが、しかしその場合、特にWindowsアプリケーションではLightしかフォントメニューには表示されず、Semiboldの存在がユーザーに分か

らないということには留意すべきです。Bボタンを押せばSemiboldにアクセスできますが、その仕組みはユーザーには分かりにくいかもしれません。それでもLightとSemiboldをリンクさせたいのであれば、Semiboldのインスタンスのスタイルリンク部分に「ボールド」のチェックを入れ、テキストフィールドには対応させたいインスタンスである「Light」と入力します。さらにSemibold ItalicがLight用のボールドイタリックになるように設定すべきでしょう。

7.3.5 補間

ウエイトとグリフ幅、そしてカスタム軸の補間値は、あるマスターと別のマスターとの間に設定されます。詳細は「12 マルチプルマスター」(p.139)を参照してください。

7.3.6 カスタムパラメータ

インスタンスのパラメータを使うと、同じリソース（フォントやマスター）からでも異なるバージョン（著作権情報、UPM、ファミリー名など）のフォントを生成することができます。インスタンスのパラメータはマスターやフォントのパラメータよりも優先されます。

「17.3 カスタムパラメータ」(p.184)には各カスタムパラメータの詳細や値のサンプルが載っています。

7.3.7 インスタンスプレビュー

インスタスタブの下部には、インスタンスのサンプルテキスト「Aang126」が表示されます。もし他のテキストを表示させたい場合はInstance Previewカスタムパラメータを追加し、テキストを入力してください。

7.4 フィーチャー

7.4.1 OpenType フィーチャーコード

フォント情報の「フィーチャー」タブではOpenTypeフィーチャーのうち全インスタンスに一律で適用するものについて設定を行います。個々のインスタンスごとに数値を調整する必要がある「カーニング」(kern)フィーチャーなど、位置調整系（GPOS）フィーチャーについては、このタブには表示されません。

フィーチャータブではOpenTypeクラス、OpenTypeフィーチャー、先頭（prefix）を任意に追加できます。クラスとはスペースや改行で区切られたグリフ名の集合で、フィーチャーはAFDKO構文に則ったフィーチャーコードになります。先頭とは、後ほど再利用を予定しているルックアップの定義など、実のコード部分の外部に記述しておく内容です。またフィーチャーを自動生成した場合、フィーチャーに関わる言語システム

(LanguageSystems) は自動的に先頭に記述されます。

7.4.2 フィーチャーコードの自動生成

一般的なフィーチャーについては、ウインドウ左下にある「自動」ボタンを押すことでGlyphsが自動的に生成して順番を整理します。このフィーチャー生成機能は特定のグリフ名や、そのサフィックス（拡張子）に反応します。例えば「.sc」というサフィックスで終わるグリフは「スモールキャップ」(smcp)と「大文字からスモールキャップ」(c2sc)に使われるようにフィーチャーが生成されます。また、基本的なルールとしてフィーチャー名を末尾に付けると該当するフィーチャーが自動生成されます。例えばx.ss01というグリフがあれば自動的に「デザインのセット01」(ss01)フィーチャーが生成されます。すべてのサフィックスについては「17.1 フィーチャー自動生成」(p.174)を参照してください。

フィーチャーやクラスによっては半自動で生成できるものもあります。ウインドウ左下の+ボタンをクリックして表示されるメニューにあるAllクラス、AllLettersクラス、Capital Spacingフィーチャーがそれです。Allクラスは現在のフォント内のすべてのグリフのクラス、AllLettersは現在のフォント内のグリフのうち文字カテゴリに分類されるグリフのクラスです。Capital Spacingとはオールキャップ組みの際にスペーシング増加するためのフィーチャーです(cpsp)。cpspフィーチャーを追加するとUppercaseという大文字だけのクラスも自動的に追加されます。

使用が廃止された「数学用ギリシャ文字」(mgrk)フィーチャーも自動生成可能ですが、+ボタンのメニューからはあえて除外されています。追加したい場合、「フィーチャー」を選択して「xxxx」を「mgrk」に変更し、「OpenTypeフィーチャーを自動生成」をチェックします。

7.4.3 フィーチャーコードの手動編集

Adobe FDKのフィーチャーの構文は以下のページにて詳細に解説されています：

www.adobe.com/devnet/opentype/afdko/topic_feature_file_syntax.html

既存のフィーチャーを手動で編集する場合は「OpenTypeフィーチャーを自動生成」のチェックを外します。手動で新規フィーチャーを追加する場合は、ウインドウ左下の+ボタンをクリックし、表示されるメニューで「フィーチャー」を選択します。先頭とクラスの追加手順も同様です。フィーチャー名、クラス名、先頭名を編集する場合は、左側のリストに表示されている名称（追加時点ではxxxxと表示されています）をクリックします。右側のエリアにはコードを記述します。なお、GlyphsはAdobe FDKの構文を使用します。「コンパイル」ボタンをクリックするとフィーチャーはコンパイルされ、編集ビューでテストできるようになります。

すべてのフィーチャーとクラスはフィーチャータブに記されたまま、フィーチャーファイルに書き出されます。唯一の例外はカーニング(kern)フィーチャーで、これはGlyphsが書き出すkernフィーチャーの末尾にルックアップとして追加統合されるようになっています。詳細は「9.2.8

kern フィーチャー手動追加」(p.120) を参照してください。

Glyphsはフィーチャーコードの識別性向上のため、構文に則ってコードを着色します。またAFDKOで使われるキーワード (substitute、ignore、lookup など) は、入力中にescまたはF5キーを押すことで残りを自動入力することができます。もし複数の候補が存在する場合、変換候補が単語の下に表示されます。上下キーで任意の候補を選択し、returnキーで決定します。

先頭、クラス、フィーチャーを削除するには、削除したい項目を左のサイドバーで選択し、-ボタンをクリックするかdeleteキーを押します。フィーチャーをOpenTypeフォント出力時に含めたくないが残しておきたいという場合、項目を選んでウインドウ上部にある「無効」にチェックを入れると、無効化することができます。

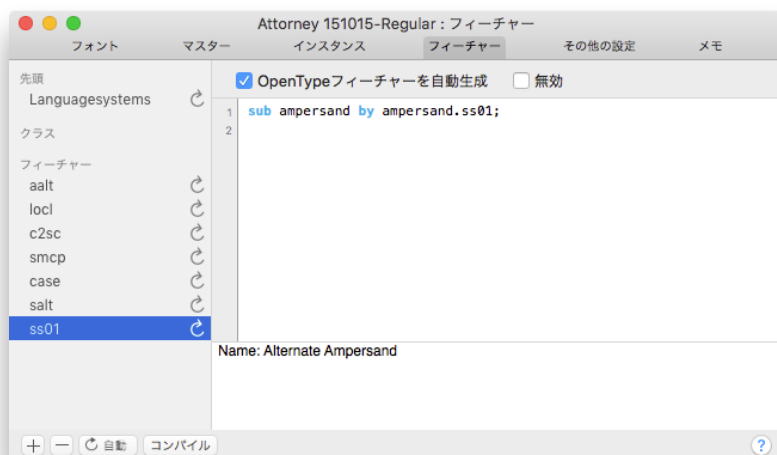
ウインドウ右側下部はメモ用のエリアです。「デザインのセット」(Stylistic Sets、ss01~ss20) フィーチャーでは、この欄で各フィーチャーに名前を設定することができます。下図のように「Name:」に続いて何のためのセットなのか名前や説明を書いてください。もしくはAdobe FDK 構文に従ったfeatureNamesルックアップも受け付けられます：

```
featureNames {  
  name <platformID> <scriptID> <languageID> "text";  
};
```

プラットフォーム、文字、言語のIDについてはAdobeのnameテーブルの仕様を参照してください：

partners.adobe.com/public/developer/opentype/index_name.html

featureNamesのブロックではそれぞれのプラットフォーム、文字、言語に合わせて複数の名前を登録できます。platformID、scriptID、languageIDはデフォルトでよければ空にしても構いません(platformID=3 Microsoft、scriptID=1 Latin、languageID=0x409 American English)。このハンドブックの執筆時点では、デザインのセットの名前はOS X 10.11 El Capitan以降でCocoaテキストエンジンを使うアプリケーションとAdobe Creative Cloud 17のみです。



Remove Features カスタムパラメータを使えば、インスタンスごとに特定

のフィーチャーをオフに設定することもできます。また Replace Feature を使えばフィーチャーの内容を変更することができます。各パラメータの詳細は「17.3 カスタムパラメータ」(p.184) を参照してください。

7.5 その他の設定

7.5.1 グリッドの間隔とグリッド細分

一般的な認識とは裏腹に、実は小数単位の座標を Postscript OpenType フォントに書き出すことが可能です。

この値はオンカーブポイント、オフカーブポイントの座標に影響します。標準値は1です。0にすると座標に小数を含めることができるようになり、Illustratorの「ピクセルグリッドに整合」オプションがオフの時のように自由な位置に動かせます。ハッチングのフィルタをかける際や細かい絵柄のフォントを作る際、またグリフを拡大縮小する際の劣化を最小限に止めたい時など、高精細なディテールを保持したいときに有効です。逆に、この値を高めにするとピクセルフォントを作る際に便利です。

すべてのツールや処理結果は最終的にグリッドに整列します。グリッドの間隔の設定に関わらず、「パス > ポイントをグリッドに整列」を選ぶことで、すべてのポイントの座標を強制的にグリッドに揃えることができます。

「グリッド細分」オプションは、グリッドの間隔は大きくても、その途中の段階を細かくグリッドとして表示して欲しいときに有効です。この値は上記のユニット間隔を細分化するマス目の数を表します。例えばグリッドの間隔を100、グリッド細分を5とした場合、1つのマスの大きさは20ユニットとなります。グリッドの間隔を1にしてグリッド細分を10にすると小数点第1位までの小数を使えるようになります。

7.5.2 標準外のグリフ名を使用

このオプションをオンにすると、Glyphs が自動的に標準グリフ名 (Nice name) へ変更するのを止めることができます。特殊なワークフローで作業する場合は、これが必要になるかもしれません。なお、環境設定のユーザー設定で「読み込み元のファイルのグリフ名を保持」(2.1.2、p.11) がオンになっている場合、既存のフォントや UFO ファイルを初めて開いたときは、自動的にそのファイルにおける「標準外のグリフ名を使用」オプションがオンになります。

編集集中にこのオプションをオフにしても即座にグリフ名が Nice name に変わるわけではありません。Nice name に自動変更するには、「グリフ > グリフ情報を更新」を選んでください。注意：この際に OpenType フィーチャーまでは変更されないので、食い違いが生じる可能性があります。

「標準外のグリフ名を使用」オプションがチェックされているときに限り、各グリフの Unicode 値を自由に変更することができます。それ以外の場合は、Glyphs はグリフ名から自動的に Unicode 値を割り当てます。

この部分についてはやや分かりにくいと思いますが、「環境設定 > ユーザー設定」にある「読み込み元のファイルのグリフ名を保持」オプションはGlyphs形式以外のファイルを開く際の挙動を決めるためのオプションであり、Glyphsで始めから作られたファイルに影響を与えるものではありません。

7.5.3 コンポーネント自動整列を解除

このオプションをオンにするとコンポーネントの自動整列が解除され、コンポーネントを使用しているグリフのサイドベアリングなどが自動で更新されなくなります。個々のグリフでコンポーネントが勝手に動いてしまうのを防ぎたい場合は、コンポーネントを選択してコンテキストメニューから自動整列をオンにしてください。環境設定で「読み込んだファイルではコンポーネントの自動設定を解除」オプションがオンになっていると、新しく読み込んだフォントについては初期状態でコンポーネントの自動整列が解除されています。詳細は「2.1.2 ユーザー設定」(p.11)を参照してください。

7.5.4 オルタネートを元のグリフの横に並べる

このオプションをオンにすると、フォントビュー内でサフィックスが付いているグリフは元のグリフのすぐ隣に表示されるようになります。例えば h.ss16、h.alt、h.loclENG は通常だと文字カテゴリの最後に並びますが、「オルタネートを元のグリフの横に並べる」がチェックされた状態ではすべて h の横に並びます。

7.6 メモ

7.6.1 フォント全体用メモ

メモタブに記述する内容はGlyphsファイル内でのみ閲覧可能で、フォント出力には一切影響を及ぼしません。メモの内容はフォントタブのnoteカスタムパラメータと同価です。詳細は「17.3 カスタムパラメータ」(p.184)の当該項目を参照してください。

テキストはMarkdown記法を採用しており、それによってボールド、イタリック、ハイパーリンク、コード、タイトルなどの見た目を整えることができます。



8 図形の再利用

8.1 コンポーネント

コンポーネントは同じアウトラインを複数のグリフで効率よく使い回すことのできる機能で、主な目的は発音記号付きのグリフを作ることにあります（例：AからÄ Å Á À Â Ã）。コンポーネントの参照元となるグリフはベースグリフと呼ばれ、コンポーネントから構成されるグリフはコンポーネントグリフ（またはコンパウンドグリフ、コンジットグリフ）と呼ばれます。コンポーネントはベースグリフと常に同期しているため、ベースグリフを変更すればすべてのコンポーネントグリフに即座に反映されます。なお、編集ビューでコンポーネントは灰色で表示されます。

CFFはコンポーネントをサポートしていないため、OpenType/CFFフォントを書き出す場合、コンポーネントは分解されて普通のアウトラインに変換されます。TrueTypeベースのフォントを書き出す場合、コンポーネントは保持されますが他のアウトラインとのオーバーラップがあるグリフでは分解されます。ただし、Keep Overlapping Componentsカスタムパラメータが設定されていれば、それらのグリフでもコンポーネントのまま出力されます。詳細は「17.3 カスタムパラメータ」(p.184)を参照してください。

8.1.1 コンポーネントグリフの生成

グリフにコンポーネントを自動的に生成するためには、あらかじめベースグリフを用意しておきます。たとえばAcute (Á) にコンポーネントを生成する際はベースグリフのAとacutecombが必要です。その上でAcuteのグリフを選び、「フォント > コンポーネントグリフを生成」(option + shift + command + C) を実行すると、Glyphsは内蔵データベースを参照して必要なコンポーネントであるAとacutecombのアウトラインを呼び出し、両者をコンポーネントとして配置します（後述のレシピでコンポーネントを使わないグリフでは何も起こりません）。

もしも既にAcuteにアウトラインを作成していたとしても「フォント > コンポーネントグリフを生成」(option + shift + command + C) を実行すればアウトラインを消去した上で、Aとacutecombのアウトラインをコンポーネントとして使用するよう再構築されます。

ベースグリフにアウトラインが存在しない場合、コンポーネントグリフの編集ビューには「empty base glyph」のマークが表示されます。

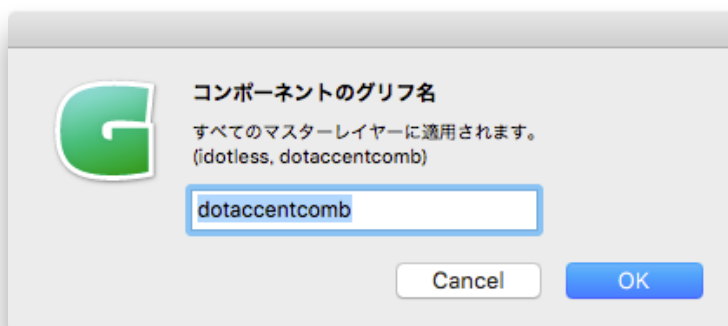
コンポーネントを手動で追加するには、編集ビューで「フォント > コンポーネントを追加」(shift + command + C) を実行、あるいは右クリックで表示されるメニューで「コンポーネントを追加」を選択し、ダイアログで任意のグリフを選択して配置します（ダイアログ上部の検索欄にグリ

複数のグリフが選択されている場合、同じコマンドで選択されているグリフすべてに同一のコンポーネントを配置することができます。

フ名を入力して検索することも可能です)。

8.1.2 パスをコンポーネントに変換

ベースグリフがまだ存在していない場合、現在選択中のパスを「グリフ > 選択パスをコンポーネント化」またはコンテキストメニューから「選択パスをコンポーネント化」を選ぶことで、パスをコンポーネントへ変換できます。現れたダイアログでは新しいコンポーネントにつける名前を決定します。Aacute (Á) などコンポーネントグリフとなるべきグリフ上のパスを選択していた場合、Glyphsは内蔵のデータベースを元に、新しいコンポーネントにふさわしいと思われる名前を提案します。



名前を決めると、その名前の新しいグリフがフォントに追加され、そのコンポーネントが配置されます。マルチプルマスターのファイルでは、他のマスターでも該当するアウトラインが自動的に置き換えられます（ただしアウトラインの互換性が取れている場合に限りです）。この機能は例えばiをidotlessとdotaccentcombから再構成したい場合などに便利です。

8.1.3 レシピ

レシピとはGlyphsに内蔵されているコンポーネントグリフ生成のためのコンポーネントのリストで、例えばAdieresis (Ä) のレシピはAとdieresiscombです。

新規にコンポーネントグリフを作成する際、標準のレシピを無視したベースグリフを使用したい場合は「フォント > グリフを追加...」のダイアログで等号 (=) を使って任意のレシピを以下の例のように記述します（左辺をベースグリフ、右辺をコンポーネントグリフとします）。なお、+ボタンでグリフを追加した場合はレシピを記述できません。

- A=a（左辺は文字グリフ単独）→複製レシピです。新規作成したaにAがコンポーネントとして配置されます。
- x+dieresiscomb=xdieresis（左辺は文字グリフ + 発音記号グリフ）→発音記号レシピです。新規作成したxdieresisにxとdieresiscomb（2点アクセント）がコンポーネントとして配置され、発音記号付きの文字になります。なお、アンカーが両グリフに設置されている場合は、

アンカー位置が採用されます。

- s.sc+s.sc=germandbls.sc (左辺は文字グリフ + 文字グリフ) リガチャーレシプです。新規作成したgermandbls.scに2つのs.scがコンポーネントとして配置され、リガチャー (ドイツ語用エスツェットのsmallキャップ) になります。

8.1.4 コンポーネントの編集


編集ビューではコンポーネントをクリックして選択することができます。選択後、tabキーで次のコンポーネントに移動します。同様にshift+tabで前のコンポーネントに移動します。選択ツールでドラッグ選択する際はoptionキーを押しながらコンポーネントを囲むようにドラッグします(optionキーを押さないとアウトラインしか選択できません)。

コンポーネントを選択すると情報パネルが右側に拡張されてコンポーネントの情報 (ベースグリフ名、相対位置、拡大率、回転角) が表示されます。この数値を変更するとコンポーネントの形状が変わります。また、パレットでもコンポーネントを変形したり鏡像反転したりできます。




発音記号のコンポーネントを垂直方向に鏡像反転すると、コンポーネントの吸着先がtopアンカーからbottomアンカー、またはその逆に変更されます。これは例えば下付き用発音記号を上付きのために再利用する際に便利です。その典型例はgcommaaccent (ğ) です。commaaccentは本来下付きですが、小文字のgcommaaccentに関しては上付きにしなければいけません。この場合、gのためだけにcommaabovecombを作るよりもcommaaccentcombを水平リフレクトおよび垂直リフレクトするだけで済むことになります。

コンポーネントの自動整列がオンになっているとコンポーネントは移動できません。コンポーネントを選択し、右クリックで表示されるメニューで「自動整列を無効化」を選択することでマウスや矢印キーでコンポーネントを移動できる状態になります。shiftキー + 矢印キーで10ユニット単位、commandキー + 矢印キーで100ユニット単位の移動が可能です。また、optionキーを押しながらコンポーネントをドラッグすると複製することができます。deleteまたはbackspaceキーを押すとコンポーネントを削除できます。

なお、ベースグリフのアウトラインを編集する場合は、コンポーネントをダブルクリックするか右側情報パネルの右端にある  ボタンを押します。コンポーネントグリフの左隣にベースグリフが表示され、編集することができます。

ベースグリフの参照元を変更するには、情報パネル右側のグリフ名をクリックしてダイアログを表示し、目的のグリフを選択します（たとえばベースグリフを `acutecomb` から `dieresiscomb` に変更する場合）。

8.1.5 ベースグリフとコンポーネントグリフとの関係

編集ビューでコンポーネントをダブルクリックするか、コンポーネントを選択してから右側情報パネルの右端にある  ボタンを押すと、ベースグリフが編集可能な状態になります。

コンポーネントの代わりに警告マークが表示されている場合があります。これはベースグリフに問題があることを示しています。警告マークには3種類あります。

- ・ `no base glyph` : ベースグリフが存在しない場合に表示されます。ダブルクリックするとベースグリフがフォントに追加され、編集可能な状態になります。
- ・ `empty base glyph` : ベースグリフは存在するもののアウトラインがない場合に表示されます。ダブルクリックすると編集可能な状態になります。
- ・ `bad reference` : コンポーネントの参照先に問題がある場合に表示されます。例えばベースグリフが現在のグリフをコンポーネントとしており、参照関係が循環してしまっている場合などです。



あるグリフが他のどのグリフでコンポーネントとして使われているかを調べたい場合は、編集ビューでコンテキストメニューを開き「このグリフをコンポーネントとして使用する全グリフを表示」を選びます（コンポーネントとして使われていない場合、コンテキストメニューに同項目は表示されません）。するとそのグリフをベースグリフとして使用している全グリフが表示されます。また、コンポーネントグリフ側でコンポーネントを選択して（あるいはコンポーネントの上で）コンテキストメニューの同項目を選んだ場合もこのコマンドは有効で、同じコンポーネントが配置されている他のグリフが現在のグリフの後に表示されます。

8.1.6 アンカー

発音記号を含むコンポーネントグリフでは、発音記号の位置は文字グリフと発音記号グリフそれぞれのアンカーの位置で決まります。上付き発音記号については、まず文字のベースグリフ内に「`top`」という名前のアンカーが必要で、次に発音記号のベースグリフにはアンダースコア付きの同名の

アンカー「_top」が必要です。コンポーネントグリフの方では、これら2つのアンカーの名前と位置に基づいて発音記号のコンポーネントの位置が定まります。

多くのグリフではGlyphs内のデータベースで必要なアンカーが定められています。これらのアンカーを追加するには任意のグリフを選択し、「フォント > アンカーを自動設置」(command + U)を実行してください。optionキーを押しながらだと「アンカーをリセット」(option + command + U)になり、既存のアンカーが一度すべて削除されて規定のアンカーが自動で再設置されます。なお、これらの操作は複数のグリフを選択した状態でも実行することが可能です。自動で設置されるアンカーはおおよそ適切と思われる位置に配置されます。また、イタリック角度が設定されている場合はその角度に合わせて配置位置が調整されます。

独自のアンカーを追加するには編集ビューにおいて右クリックで表示されるコンテキストメニューから「アンカーを追加」を選んでください。すると「new anchor」という名称のアンカーが配置されますので、ASCII文字で名前を決定します（スペースは使えません）。なお、アンカー名を確定した後でも、アンカーを選択してreturnキーを押すかダブルクリックするとアンカー名を変更できる状態になります。


new anchor

編集ビューにおいて文字のベースグリフ内のアンカーを選択すると、そのグリフに通常付帯するすべての発音記号がプレビュー表示されます。また編集ビューで文字のベースグリフと発音記号のベースグリフを表示している時、発音記号のアンカーをクリックすると、すべての文字グリフにその発音記号がプレビューされます。

アンカーはクリックして選択することができます。なお、複数のアンカーのうちの1つが選択されている場合、tabキーまたはshift + tabキーを押すことで前後のアンカーを選択することができます。なお、選択したアンカーはdeleteまたはbackspaceキーで削除できます。

アンカーはドラッグまたは矢印キーで移動することができます。矢印キーの場合、shiftを押しながらだと10ユニット単位、commandを押しながらでは100ユニット単位の移動となります。アンカーと他の2つのポイント（オンカーブまたはオフカーブ）の合計3つを選択して「パス > 選択したポイントを整列」を選ぶと、アンカーをその2点の中心に揃えることができます。ここでもフォント情報にイタリック角度がある場合は傾斜角が計算に含まれます。

時として「top」アンカーだけでは不十分な場合があります（例：リガチャーの各文字に必要な場合、組み合わせに応じて位置の微調整が必要な場合、ベトナム語用の二重の発音記号など吸着位置を大きく変更しな

ればいけない場合)。このように同名のアンカーのバリエーションを作りたい場合は、アンダースコアを使ったサフィックスで命名してください。サフィックスの内容は自由です（例：top_1、top_2のように番号で管理する、top_viet、top_acuteのように名前で管理する）。このように複数のバリエーションを作った場合、コンポーネントグリフで発音記号のコンポーネントを選択し、情報パネルにある  のアイコンから吸着先のアイコンを選ぶことができます。なお、このアイコンは吸着候補が複数ある場合にのみ表示されます。



Tip: 「編集 > 全てを選択」 (command+A) を実行するとすべてのアウトラインを選択しますが、アンカーやコンポーネントは選択されません。もう一度実行すると (command+Aを2回)、すべてのアンカーとコンポーネントも選択されます。

標準で自動設置されるアンカーの内容を変更したい場合や仮表示されるアンカーの内容を変更したい場合は「17.4.1 グローバルなグリフデータの変更」 (p.208) を参照してください。

8.1.7 自動整列

フォント情報の「コンポーネントの自動整列を解除」 (7.5.3、p.95) がオフになっている場合、コンポーネントのみで構成されたグリフでは各コンポーネントが自動整列します。自動整列とは、ベースグリフとアンカーの情報に基づいてコンポーネントの位置とグリフの幅が自動的に同期、固定されるということです。すべてのコンポーネントには自動整列の有無が内部情報として保持されています。

文字グリフ1つだけで構成されている場合：ベースグリフのカテゴリが「文字」、グリフ幅ありの発音記号、または分母や化学用下付きなどの小さな数字であれば、コンポーネントの位置とサイドベアリングはベースグリフと同じになります。

文字グリフ + 発音記号グリフで構成されている場合：最初のコンポーネントのカテゴリが「文字」で、それ以降のコンポーネントのカテゴリが「発音記号」の場合、コンポーネントの位置とサイドベアリングはベースグリフと同じになります。また、発音記号の位置はアンカーによって自動的に定められ、字幅は無視されます。この際、位置調整の基となるグリフには名前付きアンカー（例：上付き発音記号用には伝統的に top と名付ける）、発音記号側にはアンダースコアから始まる同名のアンカーが必要になります（例：_top）。

文字グリフ + 文字グリフで構成されている場合：使われているベースグリフの位置情報に基づき、自動的にリガチャーが構成されます。字幅はすべてのベースグリフの合計になり、カーニングが設定されている場合はそれも加味されます。例えば、one.numr (分子用の1)、fraction (分数用スラッシュ)、two.dnom (分母用の2) を使って onehalf (1/2) を作る場合、それぞれのグリフがタイプされた状態と同一の体裁になるわけです。

なお、発音記号のコンポーネントがある場合、位置はアンカーによって自動的に定められ、字幅は無視されます。

また、exit アンカーと entry アンカーを使って位置調整をすることもできます。開始元となるグリフの終端（出口側）に exit、それに続くグリフの始端（入口側）に entry アンカーを配置すると、それらのコンポーネントを使って作られたグリフは位置関係がアンカーで定められ、またサイドベアリングは最初のグリフと最後のグリフから採用されます。

発音記号用のアンカーはマーク吸着フィーチャー (mark と mkmk) の自動生成に使われます。アラビア文字など自動で線が繋がる文字では、exit と entry アンカーは連綿位置調整フィーチャー (curs) の自動生成に使われます。これら位置調整系 (GPOS) フィーチャーの自動生成を避けて純粋にデザインのためだけにアンカーを使用したい場合、文字以外の ASCII 記号をアンカー名の最初に付けます (例: #top、_#top、#exit、#entry)。

コンポーネントの自動整列が有効の場合、自動的にメトリクス情報を使うこととなりますので、メトリクスキーとの併用は基本的にできませんが、例外が1つあります。= + または = - を使って、自動整列グリフにサイドベアリングを足し算または引き算での加減をすることができます (例: = + 30)。詳細は「9.1.3 メトリクスキーと自動整列」(p.115) を参照してください。

数字など、グリフのカテゴリによっては初期状態では自動整列しないようになっています。これは数字グリフは等幅やプロポーションなど、スペーシングの異なる異体字を作るためにコンポーネントを利用する機会が多いため、初めから整列がオフになっている方が便利だからです。コンテキストメニューから強制的に自動整列させることができます。

また、コンポーネントグリフ内にアウトラインが存在すると自動整列は機能しません。アウトラインが削除されると、その瞬間から自動整列は復活します。コンポーネントの自動整列を強制で解除するには、コンポーネント上で右クリックまたは control + クリックすることで表示されるコンテキストメニューから「自動整列を無効化」を選びます。

文字グリフ + 発音記号グリフのコンポーネントグリフの場合、文字コンポーネントの自動整列だけ解除すると、そちらは自由に動かして字幅も変更できるようになり、発音記号コンポーネントはそれに追従します。発音記号コンポーネントの自動整列だけ解除した場合は、発音記号コンポーネントだけが自由に動かせるようになります。

コンポーネントの自動整列を有効化するには、コンポーネント上で右クリックまたは control + クリックすることで表示されるコンテキストメニューから「自動整列を有効化」を選びます。元の状態（自動整列の無効化を実行する前の状態）に戻すには、コンポーネントグリフ内に存在するすべてのコンポーネントの自動整列が有効化される必要があります。

フォント全体でコンポーネントの自動整列を無効化したい場合は、フォ

ント情報の「その他の設定」で「コンポーネント自動整列を解除」を有効にしてください。なお「コンポーネント自動整列を解除」を有効にしてコンポーネントを移動した後、「コンポーネント自動整列を解除」を無効にした場合、そのコンポーネントの移動はキャンセルされ、自動整列しますので注意が必要です。

8.1.8 コンポーネントのロック

コンポーネント上で右クリックまたはcontrol+クリックすることで表示されるコンテキストメニューから「コンポーネントをロック」を選ぶと、そのコンポーネントの位置が完全に固定されます（個々のコンポーネントに対して個別に設定します）。移動も選択もできないため不慮の変更を避けることができます。特に自動整列が通常無視されるコンポーネントにおいて便利です。ロックを解除するには、同様にロックされたコンポーネント上でコンテキストメニューを表示して「コンポーネントをロック解除」を選択します。

8.1.9 コンポーネントの分解

「フォント > コンポーネントを分解」(shift+command+D) を実行すると、現レイヤーの全コンポーネントがアウトラインに分解され、編集できる状態になります。入れ子になっているコンポーネント（コンポーネントのコンポーネントのコンポーネントなど）も分解されます。

特定のコンポーネントのみを分解したい場合は、そのコンポーネント上で右クリック（またはcontrol+クリック）でコンテキストメニューを表示して「分解」を実行します。この場合、入れ子になっているコンポーネントについてはベースグリフを一段階だけ遡ります。さらに奥の階層までアウトラインに分解したい場合は、階層の分だけ重ねて分解を実行する必要があります。

8.1.10 パスとコンポーネントの混合

レイヤーにアウトラインとコンポーネントが混在している場合、コンポーネントの自動整列は無効化されます。そのため、発音記号付きのグリフなどでアウトラインを併用している場合、予期せずコンポーネントを動かしてしまったり、メトリクスの同期を忘れてしまうことがあるかもしれません。このような事態を避けるためにも、特別な理由がなければコンポーネントのみでグリフを構成したほうがいいでしょう。

レイヤーにアウトラインとコンポーネントが混在している場合、アウトラインをドラッグするとコンポーネント内のポイントにスナップします。またアウトラインの1ポイントとコンポーネントが選択された状態で「パス > 選択したポイントを整列」を実行すると、そのポイントを基準としてコンポーネントが移動します。基準は原点 (x=0、y=0の地点) ですが、

originという名前のアンカーがあればそこが基準になります。コンポーネントを使ってセリフを作るときには、この手法を使うと便利です。

8.1.11 コンポーネントのネスティング (入れ子)

Glyphsではコンポーネントを入れ子にすることが可能です。例えば2つのdotaccentcombをコンポーネントとしてdieresiscombのグリフを作成し、それを使ってadieresis (ä)などの発音記号付き文字を作ることができます。

入れ子になったコンポーネントは、アンカーの情報を大元のベースグリフから引き継ぎます。発音記号を組み合わせる際なども、特に新しいアンカーを設置する必要はありません。例えばoslashacute (アキュート付きø)を作る場合、oslashがoとslashlongcombをコンポーネントしているのであれば、acuteの位置合わせにはoのアンカーが再利用されるということです。そのためoslashにあらためてアンカーを設置する必要はありません。アンカーを設置した場合は、そちらが優先して使われます。

8.1.12 優先される発音記号のグリフ名

発音記号のコンポーネントグリフを生成する際、Glyphsは文字グリフと同じサフィックスを優先します。例えばadieresiscomb.sc (Åのスマールキャップ)を生成する場合、dieresiscomb.sc (スマールキャップ専用でデザイン調整されたdieresiscomb)がフォント内に存在する場合、通常のdieresiscomb (スマールキャップ用ではないもの)ではなくそちらを優先して使います。

大文字用の発音記号には.caseのサフィックスが使われます (例: Öを生成する際、dieresiscomb.caseがあればdieresiscombではなく前者が使われます)。

iやjの発音記号付きグリフでは、通常ドットが外されます。このため、合成用にidotless (ドット無しi)とjdotless (ドット無しj)を必ず用意しておいてください。また、iとjの発音記号付き文字グリフでは、その字幅の狭さゆえに発音記号が隣り合う文字とぶつかってしまうことがあります。それを避けるために幅の狭い発音記号を用意しておくことがあります。Glyphsはこれを理解しており、「.i」または「.narrow」というサフィックスの付いた発音記号が存在すれば、それをiとj専用のものと判断して優先的に使用します。また狭いバージョンの発音記号グリフはOpenTypeによるccmp (ダイナミック合成)のフィーチャーにも自動的に使用されるようになります。

8.1.13 アンダースコアで始まるグリフ名のコンポーネント

部品用コンポーネントなど、出力時に含めたくないグリフの名前はアンダースコアで開始することをお勧めします (例: _serifBottomLeftや_

A.ogonek)。こうするとスマートフィルタ (6.5.4、p.74) を設定しやすくなったり、グリフ名で検索 (6.1.3、p.64) する効率がよくなります。

なお、アンダースコアで始まるグリフを追加すると「出力時に含む」が初期状態でオフになります。

8.2 スマートコンポーネント

スマートコンポーネントは東アジアの文字向けに開発された機能です。もちろん他の言語でも、文字のパーツを調整しながら繰り返し再利用する場面で効率が大幅に上がる便利な機能です。

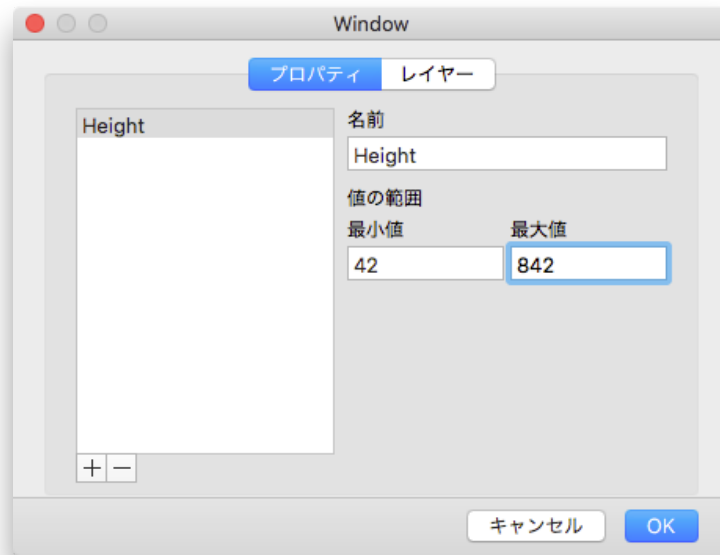
漢字やハングル、チベット文字などでは似た形状の構成要素が繰り返し用いられます。ただ、そのサイズやプロポーシオンは完全に同じではなくグリフごとに調整が必要です。このような需要に応えるため、スマートコンポーネントではグリフ単位での小規模なマルチプルマスターを使います。スマートコンポーネントを駆使するにはマルチプルマスターへの理解が必要なので、先に「12 マルチプルマスター」(p.139) を読むことをお勧めします。

8.2.1 スマートグリフの設定

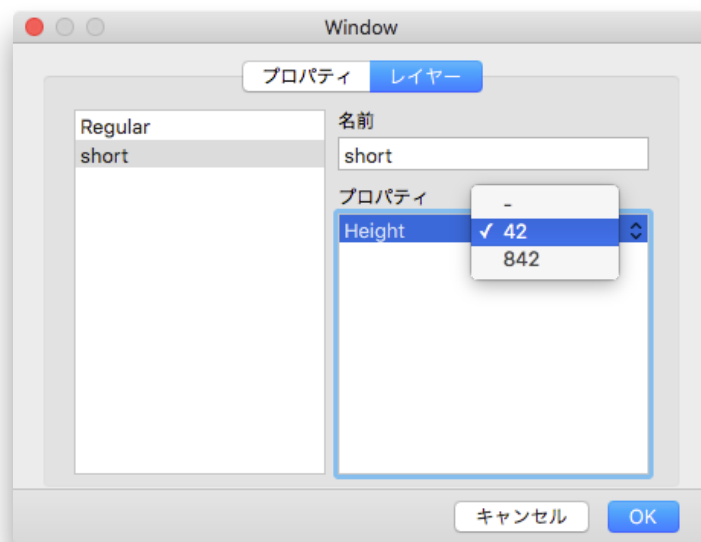
スマートグリフは複数の補間用レイヤーによって構成されます。スマートグリフをスマートコンポーネントとして別のグリフに配置すると、スライダや数値入力によって補間された形状に調整することができます。ハングルの部首に指定されているグリフは、初期状態でスマートグリフとして使用できます。それ以外の場合、名前に `_part.` というプレフィックスが付いたグリフ (例: `_part.001`、`_part.arch`) であればスマートグリフと認識されます。

スマートグリフにはアウトライン互換の取れた複数のレイヤーが必要です。各レイヤーには `Wide`、`Deep`、`Narrow` など管理しやすい名前をつけると良いでしょう (日本語の入力も可能です)。

次に右クリックで表示されるコンテキストメニューから「スマートグリフ設定」を選択して (または `option + command + I` で) ダイアログを表示します。「プロパティ」タブでは「+」で補間軸を追加した後、「名前」テキストフィールドに軸の名称を入力し (日本語の入力も可能です)、「最小値」「最大値」欄で値の範囲を設定します。例えばディセンダーが調整可能なスマートグリフでは、名前を `decender`、最低値を `-200`、最高値を `0` にするなど分かりやすく設定すると良いでしょう。h、m、nなどのショルダーの幅を調整したい場合は最小値を `0`、最大値を `100` にするといいかもありません。また、アセンダーの高さを調整する場合であれば最低値を `x` ハイト、最大値をアセンダーハイトにするなど、実際の値を用いると良いでしょう。なお、「プロパティ」タブで設定した軸は作業中のスマートグリフのみで有効です。



「レイヤー」タブでは、各レイヤーのアウトラインの形状と、先に設定した軸の値の対応関係を設定します（フォント情報のマスター設定で軸と数値を設定するのと同じような要領です）。左側の枠でレイヤーを選択した後、右側の枠で各軸の値を設定します（プルダウンメニューに先に設定した最低値か最大値が表示されますのでいずれか一方を選択します）。全レイヤーに対して、すべての軸に関する設定をおこなってください。このグリフをスマートコンポーネントとして使用すると、その形は「プロパティ」タブで設定した軸に沿って「レイヤー」タブで設定したレイヤーの間を補間します。



例えば幅と高さの2軸があり、いずれも最小値0、最大値600というケースを考えてみましょう。この場合であればレイヤーAは幅600/高さ600と設定、レイヤーBは幅0/高さ600と設定、レイヤーCは幅600/高さ0、レイ

ヤーDは幅0/高さ0と設定すると良いでしょう。ただし2軸補間の場合、どれか1つの頂点は欠けていても自動で補うことができますので3レイヤー構成でも機能します（3軸の場合、最低必要なレイヤー数は4です）。ただし自動補間した場合とデザインスペースの各頂点のレイヤーをちゃんと用意した場合には結果が異なりますので、確実に形状をコントロールしたい場合はなるべく多くの最小値/最大値のパターンのレイヤーを作るようにしましょう。

軸に「Width」と「Height」と名付けた場合、スマートコンポーネントのバウンディングボックスでサイズ変更する際に、通常の変形方式ではなく補間を使った拡大縮小が可能になります。これは上記の通り正確に軸を命名した場合のみ使用できる機能です（例：全小文字のheightや全大文字のWIDTHでは無効です）。バウンディングボックスを使用するには、「表示 > バウンディングボックスを表示」（option + shift + command + B）を選択します。

8.2.2 スマートコンポーネントの追加

スマートコンポーネントを他のグリフに追加する手順は「8.1.1 コンポーネントグリフの生成」（p.97）と同じで、右クリックで表示されるコンテキストメニューから「コンポーネントを追加」（shift + command + C）を選択します。違いはスマートグリフ（漢字やハングルの部首、または `_part.` で始まるグリフ）を選ぶという点だけです。`_part.` のグリフであればダイアログにアンダースコアを入力するだけですぐにスマートグリフを絞り込むことができます。

8.2.3 スマートコンポーネントの補間

スマートコンポーネントが配置されたグリフで、コンポーネント上でコンテキストメニューを表示して「スマートコンポーネント補間」（option + command + I）を選択します。表示されたダイアログでスライダーを動かしたり数字を入力するとコンポーネントを補間することができます。



テキストフィールドには最低値未満の値または最大値より大きい値を入力し、補外することも可能です。

テキストフィールドでは矢印キーの上下で数字の増減を、shift キーを

押しながらで10単位の増減をすることができます。

8.2.4 自動整列用アンカー

コンポーネントグリフ内に複数のコンポーネントがある場合、アンカーが埋め込まれていればその位置情報を使って自動整列させることができます。例えばベースグリフに connect という名前のアンカーがあり、追従するベースグリフに _connect という名前のアンカーがあれば、そこを起点に後者のコンポーネントは吸着します。このコンポーネントにも connect アンカーがあれば、次のコンポーネントはやはりそこに吸着します。スマートグリフでは通常のマルチプルマスターと同様にアンカーの互換性も取らなければいけないため、このテクニックを使用する際はすべてのレイヤーに同数かつ同名のアンカーが存在することを確認してください。

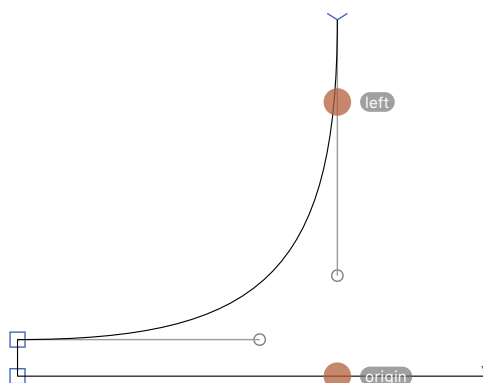
8.3 コーナーとキャップコンポーネント

8.3.1 コーナーコンポーネント

コーナーコンポーネントは主にセリフを作るために用意された機能です。開いたアウトラインのコンポーネントであり、他のアウトラインの角に吸着しスムーズに接続します。

Tip: 凹んだセリフのコーナーコンポーネントに left と right アンカーの両方を配置すると、フレックスヒントが自動生成されます。

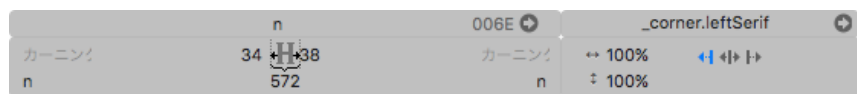
コーナーコンポーネントのベースグリフは、名前が _corner. で始まる必要があります (例: _corner.leftSerif)。このグリフに、原点を基準に開いたパスを描きます。なお、origin と名の付いたアンカーを配置した場合は、そこが原点として機能します。始点は必ず垂直、終点は必ず水平になるようにパスを作成してください。また、どちらかの点 (通常は始点) の横位置が原点または origin アンカーと揃うように作図してください。left や right アンカーを追加すると、吸着先のパスに合わせたコンポーネントの変形具合を制御できます。V など斜めのステムに垂直ステム用のコーナーコンポーネントを吸着させたい場合に便利です (詳細は後述)。



パスの方向は、吸着先のパスの方向と一致していなければいけません。つまりステムの左下の角に吸着するセリフを作るのであればパスが上から開

始している必要があります。パスの方向を変更する必要がある場合はパスを選択し、コンテキストメニューから「選択パスの方向を逆転」を実行してください。

コーナーコンポーネントを利用するには、コンポーネントを吸着させたいパスのポイントを1つ選択した状態で右クリックで表示されるコンテキストメニューの「コーナーコンポーネントを追加」でダイアログを表示し、コンポーネントを選択します。コーナーコンポーネントが追加されるとコンポーネントの開始点と終点が選択したポイントに揃うように吸着します。吸着先の角が直角になっていない場合、自動的に末端の角度が調整されます。



コーナーコンポーネントを削除したい場合は灰色の●を選択してdeleteまたはbackspaceキーを押します。また、コンポーネントを選択した状態で「編集 > コピー」(command+C)を実行し、他のオンカーブポイントを1つ選択した上で「編集 > ペースト」(command+V)を実行するとコーナーコンポーネントをペーストすることができます。

●をクリックして選択すると、情報パネルが右側に拡張され、コーナーコンポーネントの詳細が現れます。ここでパネル右端の🔄をクリックするとコンポーネントのベースグリフが表示され、編集できる状態になります。コンポーネント名をクリックするとダイアログが表示され、コーナーコンポーネントを選択し直すことができます。また水平方向と垂直横行の拡大率や、変形サイドを選択できます。変形サイドとは、吸着先のパスの変形に合わせてコンポーネントのどちら側を湾曲させるかを選択するものです。

- 🔄: コーナーコンポーネントの開始側が湾曲し、終了側は固定されます。ステムの左下や右上のセリフにはこのオプションが最適でしょう。
- 🔄: コーナーコンポーネントの終了側が湾曲し、開始側は固定されます。ステムの右下や左上のセリフにはこのオプションが最適でしょう。
- 🔄: コーナーコンポーネントのどちら側も回転湾曲します。インクトラップ（墨取り）のデザインにスマートコンポーネントを使用するときなど、どちらの角度も参照したい場合に有効です。

吸着先パスへのコンポーネントの位置や角度は、多くの場合、left、right、origin アンカーの存在、パスの方向、および変形基準によって定まります。これらのパラメータを使って、例えば斜めのステムに吸着した場合のセリフの長さを制御できます。そのためには、スマートコンポーネント内でorigin アンカーの真上にleftまたはright アンカーを配置しておくことが大切です。左下および右上の角に吸着させる場合はleft、逆の場合にはright アンカーを配置してください。両方順応または凹型セリフの場合は

両方配置します。これらのアンカーと origin アンカーの距離に応じてセリフの長さが変化します。origin アンカーから上に離すと、鋭角コーナーに吸着したセリフは短くなり、鈍角コーナーに吸着している場合は長くなります。origin アンカーに近づけると逆の効果を得られます。カーブのないスラブセリフの場合、水平と垂直セグメントの交点とちょうど同じ位置にこれらのアンカーが必要になります。

Tip: コーナーコンポーネントを反転する最も早い方法は、ウインドウ右のパレットの変形オプションから反転ボタンをクリックすることです。

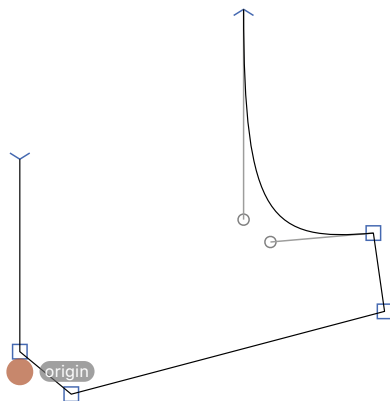
コーナーコンポーネントは反転して逆サイドの角にも使用できます。つまり水平拡大率を -100% にするとコンポーネントは反転され、左用のコンポーネントを右側のコーナーにも再利用できます。

コーナーコンポーネントをすべて分解してアウトラインを得たい場合、「フィルタ > 重なったパスを合体」(shift+command+O) を選択します。コーナーコンポーネントを個別に分解したい場合、コーナーコンポーネントを選択してコンテキストメニューから「コーナーコンポーネントを分解」を選択します。

8.3.2 キャップコンポーネント

キャップコンポーネントはステムの先端にセリフを付けたりスワッシュを付けたりするために用意された機能です。コーナーコンポーネントとほぼ同じですが、1ポイントではなく2ポイントに吸着します。

キャップコンポーネントとなるグリフは、名前が `_cap.` で始まる必要があります (例: `_cap.headSerif`)。このグリフに、原点を基準に開いたパスを描きます。なお、origin と名の付いたアンカーを配置した場合は、そこが原点として機能します。左側の端点は横位置が原点または origin アンカーと揃うように作図してください。また、パスの下端はベースラインに揃うように作図します。パスの方向を変更する必要がある場合はパスを選択し、コンテキストメニューから「選択パスの方向を逆転」を実行してください。

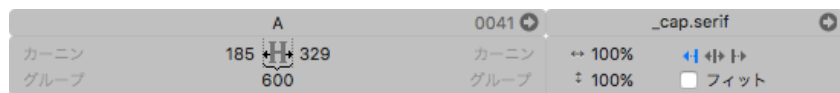


キャップコンポーネントを利用するには、コンポーネントグリフ上の連続する2つのポイントを選択した状態で右クリックで表示されるコンテキストメニューの「キャップコンポーネントを追加」でダイアログを表示し、コ



ンポーネントを選択します。キャップコンポーネントが追加されるとコンポーネントの開始点と終点が選択したポイントに揃うように吸着します。適切な状態で吸着させるためには、キャップコンポーネントの開いたパスの幅と、配置先のステムの幅が一致しているのが理想です。

キャップコンポーネントの存在を示す灰色の●をクリックして選択すると、右側情報パネルで水平方向と垂直横行の拡大率を変更することができます。また、「フィット」を選択すればステムに合わせてコンポーネントが自動的に変形します（この際、水平拡大率は変更不可となります）。右側情報パネルでは整列基準を変えることもできます。左基準または右基準にすると、コンポーネントの吸着先もそれに応じて変化します。また、leftとrightのアンカーがキャップコンポーネントにあれば、それらのアンカーの原点（またはoriginアンカーのy座標）からの垂直距離と、開始点および終点の水平距離が参照され、どのようにパスが湾曲するかを決定します。別の言い方をすれば、leftアンカーとrightアンカーはそれぞれ配置先の2つのポイントの想定位置を示したものです。この効果は特に傾いたステムにおいて顕著です。



右側情報パネルに表示されているコンポーネント名をクリックするとダイアログが表示され、キャップコンポーネントを選択し直すことができます。また、右側情報パネル右端の●をクリックするとコンポーネントのベースグリフが表示され、編集できる状態になります。

灰色の●を選択して「編集 > コピー」(command + C)を実行し、他の連続する2つのポイントを選択した上で「編集 > ペースト」(command + V)を実行するキャップコンポーネントをペーストすることができます。また、灰色の●を選択してdeleteまたはbackspaceキーを押すとキャップコンポーネントを削除できます。

コーナーコンポーネントをすべて分解してアウトラインを得たい場合、「フィルタ > 重なったパスを合体」(shift + command + D)を選択します。コーナーコンポーネントを個別に分解したい場合、コーナーコンポーネントを選択してコンテキストメニューから「キャップコンポーネントを分解」を選択します。

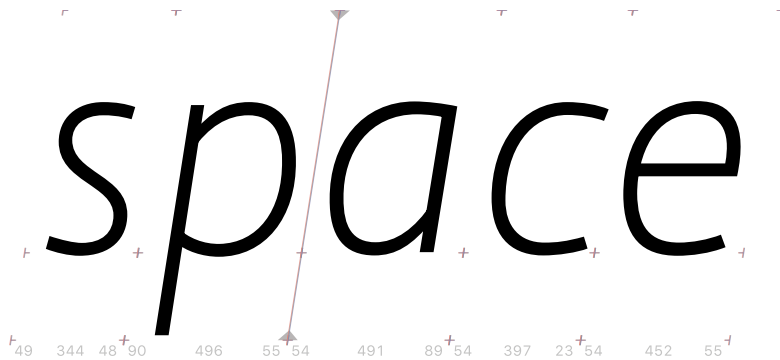
9 スペーシングとカーニング

9.1 スペーシング

左右のサイドベアリング値は英語版ではleft sidebearingとright sidebearingの略としてLSB、RSBと表記しています。日本語版では左サイドベアリング、右サイドベアリング、または省略表記として左SB、右SBとします。

スペーシングとはテキストの均一なリズムを生み出すためにサイドベアリングを調整する作業のことです。スペーシングには特に決まったルールはありません。しかし等幅フォントなどの例外を除けば、形が似たグリフではサイドベアリングも近い値になるべきです。例えば通常Dの左側はHと同じ、右側はOのようなサイドベアリング値にすべきです。

編集ビューでは、高さや幅の情報を確認できるボディの表示／非表示を「表示 > メトリクスを表示」(shift+command+M)で切り替えられます(テキストツール、てのひらツールの場合を除き、初期状態では表示されています)。



9.1.1 スペーシングのショートカット

左側にあるcontrolキーで左サイドベアリング、右側にあるcommandキーで右サイドベアリングを調整する、と覚えると良いでしょう。

テキストモード(T)では、キーボードショートカットで現在のレイヤーのサイドベアリングを調整することができます。controlキーを押したまま右や左の矢印キーを押すと左サイドベアリング値を増減することができ、commandキーと矢印キーで同様に右サイドベアリング値を増減できます(これに伴いグリフ幅も増減します)。なお、shiftキーも押したまま操作すると10単位での調整が可能です。controlキーとcommandキーの両方を押しながら右の矢印キーを押した場合は左サイドベアリング値が増加、右サイドベアリング値が減少し、実質的にグリフが右方向に移動することになります(グリフ幅は保たれます)。左の矢印キーだと逆方向への移動となります。

なお、左サイドベアリングを調整するショートカット(controlキー+左右矢印キー)は、OSX標準の「左(右)の操作スペースへ移動」ショートカットと競合しています。これはシステム環境設定で変更または無効化できます。

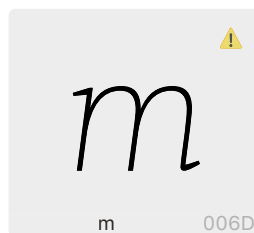
9.1.2 メトリクスキー

メトリクス同士を関連付けるため、数字の代わりにメトリクスキーと呼ばれる情報を入力することもできます。参照したいグリフ（ベースグリフ）の名前を任意のグリフのサイドベアリングまたはグリフ幅のフィールドに入力すると、サイドベアリングやグリフ幅の値がベースグリフと同一になります（例えば、bやpの右サイドベアリングにoと入力する）。メトリクスキーは左サイドベアリング、右サイドベアリング、グリフ幅のうち2項目まで入力することができます。

メトリクスキーは常時自動的に更新されるわけではないことに注意してください。ベースグリフのサイドベアリングやグリフ幅の値が変更された場合、「フォント > メトリクス情報を更新」(control+command+M)を実行、もしくは「フォント > 全マスターのメトリクス情報を更新」(control+shift+command+M)を実行することでリンクされたグリフのサイドベアリングやグリフ幅の値も更新されます（更新されていない場合、フォントビューでは各グリフに黄色い警告マークが表示されます）。なお、編集ビューの情報パネルでは、数値が最新でない項目は赤い文字で表示され、横に⌘マークが表示されます。このアイコンをクリックするか、いったん入力欄をクリックした後他の場所をクリックすることで値が更新されます。



フォントビューでは、メトリクスキーの更新が反映されていないグリフの右上に黄色の警告アイコンが表示されます。マルチプルマスター環境では、どれか1つでもインスタンス生成に関わるレイヤーが未同期であれば警告アイコンが出ます。

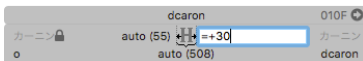


サイドベアリング入力欄には単純な計算式を記述することもできます。この場合、計算式は等号(=)で始まる必要があります。例えば「=n+10」はnのグリフのサイドベアリング値に10を足した値にするということです。同様に「=n-10」ならnのグリフのサイドベアリング値から10を引いた値になります。「=g/2」はgのグリフのサイドベアリング値を2で割った

値、「 $=v*2$ 」は v のグリフのサイドベアリング値を2倍した値になります。なお、「 $=n$ 」は単純に「 n 」と入力した場合と同じです。また「 $=20$ 」などとした場合、必ずその数値になるように指定できます（グリフを編集してサイドベアリング量が変わった場合、情報パネルには赤い文字で警告が表示されます）。これは等幅フォントなどのようにグリフ幅を固定したい場合、またはストロークが繋がるスクリプト書体などサイドベアリングを固定させたい場合に便利です。

等号を二重（ $=$ ）にするとキーの有効範囲が現在のマスターのみに限定されます。これはマルチプルマスターのフォントにおいて特定のマスターで違うキーを使いたい場合に便利です。

垂直棒の記号（ $|$ 、日本語キーボードでは $\text{shift}+\text{¥}$ 、USキーボードでは $\text{shift}+\text{\}$ ）を使うと逆側のサイドベアリングを参照することができます。例えば u の左サイドベアリング値を n の右サイドベアリング値と同じ値にしたい場合、 u の左サイドベアリング値入力欄に「 $=|n$ 」と入力します。



9.1.3 メトリクスキーと自動整列

コンポーネントの自動整列が有効になっているグリフでさらにサイドベアリングを調整したい場合は、等号とプラスまたはマイナス記号（ハイフン）を使って値を入力します（例： $=+20$ 、 $=-10$ ）。こうすることで自動整列の結果からスペーシングを加減できます。

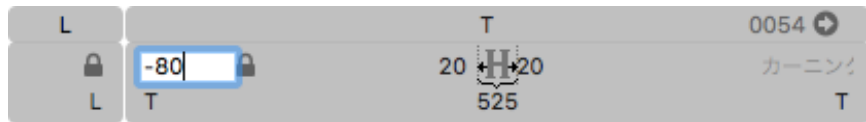
これは横方向のスペーシングの調整をしたいがコンポーネントの自動整列は解除したくない、という際に便利です。その典型的な例は欧文の dcaron (d) や lcaron (l) など、文字の横に発音記号が付くためにサイドベアリングの追加が必要なグリフです。

9.2 カーニング

多くのグリフは調整することなく並べても特に問題なく見えますが、並び方の組み合わせによっては微調整が必要になります。特に大きなスペースが空いてしまう VA、LW、T と小文字などは調整が必要な組み合わせの典型です。これらサイドベアリングだけでは解決できない個別のケースを調整する作業がカーニングです。カーニングはスペースを詰めるだけでなく広げるためにも使われます。

9.2.1 カーニングの方法

カーニングの設定値「カーニングペア」を設定するには、まず編集ビューでテキストツール（T）に切り替えて、カーニングしたい文字列を入力します。次に調整する2つのグリフの間にカーソルを置き、情報パネル左端・上段の入力欄（「カーニング」と表示されている部分）に、この2つのグリフのカーニング値を入力します。値をマイナスにすると文字間が詰まります。



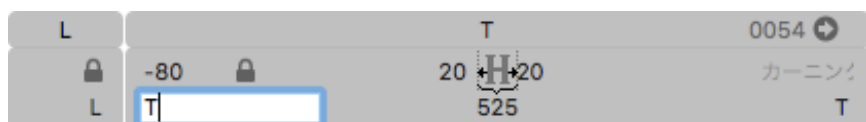
メトリクスの調整と同様、左側にあるcontrolキーで左側のカーニング値、右側にあるcommandキーで右側のカーニング値を調整する、と覚えると良いでしょう。

メトリクス調整と同じように、キーボードショートカットを使ってカーニング値を調整することもできます。option+control+左右の矢印キーで現在の文字（カーソルの右にある文字）の左側のカーニング値、option+command+左右の矢印キーで現在の文字の右側のカーニング値が増減します（右側のカーニング値は、現在の文字の次にも文字が入力されている時のみ調整することができます）。shiftキーも押して操作すると10単位での調整が可能です。

9.2.2 カーニンググループ

形状が似ているグリフには同じカーニング値を適用するのが理にかなっています。カーニンググループはこれらを一括設定する機能です。カーニング値が単独のグリフだけでなく、グループ内のすべてのグリフにも共通で適用されるため、手動で設定するカーニング値の数を大幅に減らすことができます。たとえばAとAの発音記号付き文字を同じグループにまとめておけば、AT間のカーニングはÁT、ÄT、ÅTなどにも自動で適用されます。

Glyphsでは、カーニンググループはグリフの値の1つとして編集されます。カーニンググループの設定欄は、編集ビューでは情報パネルの左右・下段に存在します（「グループ」と表示されている部分）。また、フォントビューのリストモードでは「左グループ」「右グループ」と表示されている列で設定することができます。こちらのほうが各グリフの設定状況が把握しやすく効率的に作業できるでしょう。








Tip: カーニングのグループを自動で設定してくれる「Set Kerning Groups」スクリプトが以下のサイトからダウンロードできます：

github.com/schriftgestalt/Glyphs-Scripts

例えばOの左側のカーニング値を使い回したいすべてのグリフ（C、Ccedilla、G、Odieresis、そしてO自身）の左グループ入力欄に共通のグループ名（例：O）を入力します。どのグリフにもなるべく左グループと右グループを設定してください（たとえグリフが1個しか含まれないグループでも作成するのが理想です。これはOpenTypeのkernフィーチャーをより効率的に生成するためです）。

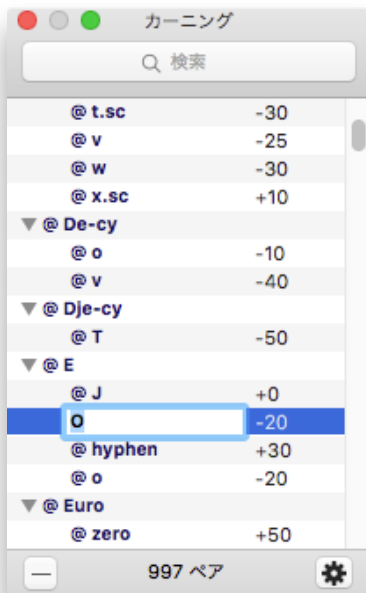
編集ビューでテキストツールを選択してカーソルを文字間に置くと、情報パネルにはカーソルの右側にあるグリフの情報（グリフ名、グループのロック、グループ名）、情報パネルの左隣の小さな情報パネルにはカーソルの左側にあるグリフの情報が表示されます。

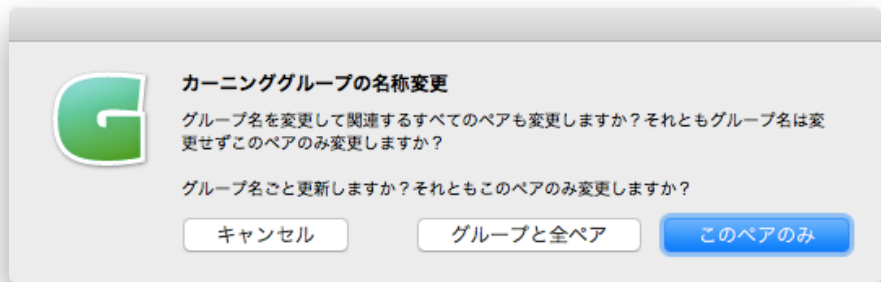
情報パネルにある  を解除して違うペア値を入力することで、グループカーニングに例外を作ることができます。例えばoとöは同じカーニンググループにしておきたいところですが、ToとTöでは違うカーニング値が必要になる場合があります（アクセントがTのバーと接触してしまう可能性があるため）。この場合、Töの間にカーソルを置き、情報パネル（öの側）の錠前を解除  した上で左カーニング値を変更します。なお、ロック  すると例外値は破棄され、グループ値に従う状態に戻ります。例外値はグループと単独グリフの組み合わせにすることもできます。たとえばTöのペアに設定した例外値はoグループの中でöにのみ適用されるので、öの錠前マークは解かれている  ことになります。しかしこの例外値はどんなTにも適用されるべきなので（T̈öやT̄öなど）、T側の錠前はロック  されたままにします。このようにペア内での両者のロックの有無により、「グループ対単独」、「単独対グループ」、そして「単独対単独」の例外を作ることができるのです。

グリフに割り当てたカーニンググループの名称を変更したい場合は、カーニングウインドウで既存のグループ名で登録したペアを探し、そのグループ名をダブルクリックして後のグループ名を書き換えます（@はグループであることを示すための便宜上の記号であり、ユーザーが入力する必要はありません）。入力内容を確定するとグループ名の変更をどこまで反映させるか確認のダイアログが出ますので、同名のすべてのペアおよびカーニンググループ名に反映させるか、当該のペアのみに反映させるかを選択してください。

当該のペアのみに反映する場合、カーニンググループ名は変更されず、編集を加えたペアのみが新しいグループに移ります。例えば@dという右グループが小文字dの右側のグループであるとして、その中にはdcaron (đ) も含まれるとします。このグループを@h（小文字hやlの左側グループ）とdcaron基準でカーニングすると、他のdhペアは大きく開いてしまうことになります。このような場合、まずはdcaronの右側グループを新しく@dcaronとします。その後に@d-@hペアを@dcaron-@hペアに変更し、入力後に現れるダイアログで「このペアのみ」を選択してください。

既存のグループ名を変更する場合には、入力する場所に注意を払う必要があります。フォントビューや編集ビューの情報パネルのグループ名を書き換える場合、そのグループ名は新規グループの作成と解釈され、既存グループの改名にはならないからです。この場合、改名後には、既存のグループ名で施していたカーニングペアまでは変更されません。グループ名と関連するペアをすべて変更したい場合であれば、カーニングウインドウから行いましょう。



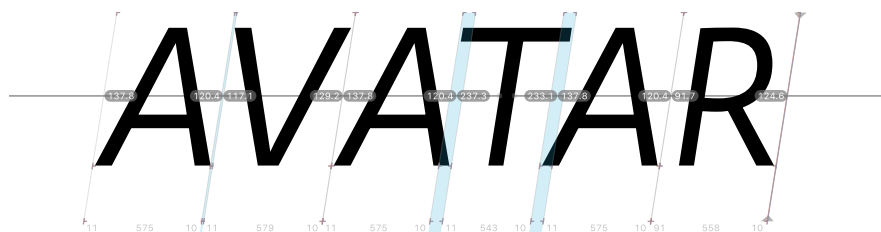


9.2.3 カーニングペアの確認

カーニングウインドウ（「ウインドウ > カーニング」）にはすべてのカーニングペアの一覧が、ペアの先（左側）のグリフまたはグループの順に表示されています。3角が付いているのが先（左側）のグリフまたはグループ、3角をクリックして展開することで表示されるのが後（右側）のグリフまたはグループです。グループはアットマーク付き（例：@latin_cap_A）で、青い文字色で表示されます。単独のグリフはベージュです。

カーニングウインドウでペアをクリックすると編集ビューにそのペアが表示されます。カーニング値を表示、編集するときは必ずウインドウ右下のカーニングボタンがオン **To** またはサイドベアリング固定 **To** になっていることを確認してください（**To** はカーニングが効いていない状態です。状態を切り替えるにはこのボタンをクリックします）。カーニングウインドウで選択中のペアがグループを含んでいる場合は、ウインドウ右下の **⚙** メニューから「すべてのグリフを表示」を選ぶと起こり得るすべてのペアが編集ビューに表示されます。

「表示 > メトリクスを表示」がオンの状態では、テキストツール選択時にカーニング状態がカラーで表示されます（詰めは水色、空けが黄色）。「表示 > 補助線を表示」をオフにすると、カーニングカラーは下端にのみ表示されます。詳細は「3.9 計測」（p.36）を参照してください。

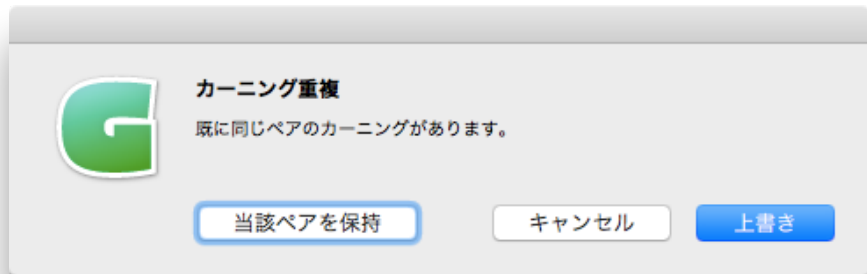


9.2.4 カーニングペアの削除

カーニングウインドウでカーニングペアを選択し、ウインドウ左下の「-」ボタンをクリックすると、選択中のカーニングペアをすべて削除します。編集ビューでは、情報パネルのカーニングペア欄を空欄にすることも削除できます。

9.2.5 カーニングペアのコピー

カーニングペアを他のマスターやフォントにコピーしたい場合、「ウインドウ > カーニング」(option+command+K)でカーニングウインドウを表示し、コピーしたいペアを選択します。すべてのペアをコピーしたい場合はカーニングウインドウがアクティブな時に「編集 > すべてを選択」(command+A)を選んですべてのペアをコピー(command+C)した後、目的のマスターのカーニングウインドウにペースト(command+V)します。もし重複がある場合は対処を促すダイアログが表示されます。



9.2.6 カーニングのクリーンアップ

グリフを消去した場合、あるいはフォントを他からインポートした場合は、ペアを構成するグリフやグループが存在しないために無効なカーニングペアが発生することがあります。これらを取り除くには、カーニングウインドウの右下の ✳ ボタンから「親字不在のペアを消去」を選んでください。

9.2.7 カーニングの圧縮

単独のカーニングペアを設定した後でカーニンググループを設定するとした場合、その時点で当該ペアはまだグループカーニングに属していません。カーニングウインドウ右下の ✳ ボタンから「なるべくグループ化」を選択することでグループカーニングに変換することができます。この機能は単独グリフに設定したペアを可能な限りグループ用ペアに変換する機能です。グループ化機能は例外値をキープしますが、グループ値と同一になっている例外値(ロックを外したが値はグループ値と同じもの)は削除されます。一度で全ペアの処理が完了するとは限りませんので、念のため2度実行してください。

例えば、Ṭ-m間のペアがフォント内にあるとしましょう。それぞれ@Tという右グループと@nという左グループに属していますが、Ṭmはお互いグループではなく単独グリフ同士のペアになっているとします。「なるべくグループ化」を一度選択すると、ペアは@Tグループと単独小文字mのペアに変化します。Tm、Ṭm、Ṭmなどはすべてカーニングが適用されますが、mに対してのみ反映されるため、@nに入っているはずのnやrなどのグリフには適用されません(Tnはカーニングされないまま)。なるべくグループ化をもう一度実行すると、ペアの後半のグリフもグループ化

されるため、@T内のどのグリフも@n内のどのグリフともカーニングされるようになります。またTñに別の値のカーニングペアを設定していた場合、そのペアは例外値として残され、@T-@nの標準値とは違って表示されます。

9.2.8 kern フィーチャー手動追加

ここまでの作業で設定したカーニング値から、フォント出力時に自動的にOpenTypeのkern フィーチャーが生成されます。これに加えてフォント情報 (command + I) ウィンドウのフィーチャータブではkern フィーチャーを手動で追加することができます。ここに書いたカーニング値は「kernCustom」という独立したルックアップとして出力時にkern フィーチャーの末尾に自動的に追加されます。ただし、ここに書くフィーチャーはファミリー全体で共有となるため、通常のカーニングと違って数値の補間はされません。

Tip: ここでカーニンググループを使用したい場合、グループ名の前に@MMK_L_ (ペアの左側の場合) または@MMK_R_ (ペアの右側の場合) を付けてコードを書きます。例えばL'Aのトリオをグループ化したい場合、以下のようなコードになります: @MMK_L_L' -50 quoteright' 60 @MMK_R_A; これはL右側のグループ名が「L」、A左側のグループ名が「A」と仮定した場合の内容です。Glyphs内部では実際のカーニンググループ名はこのような長いのですが、その前半は必要時以外はユーザーから見えないうようになっています。

この手動追加の機能が便利になる局面は、たとえば3文字が関連するなど、広範囲の前後関係に依存するカーニング値を記述したい場合です。たとえばf、スペース、Tの組み合わせ (f T) などではfとTの頭が近寄りすぎてスペースが視認し辛くなり、少し開けた方がいい場合があります。またLとアポストロフィ (L') のペアをきつく詰めた一方で、'Aもきつく詰めている場合、フランス語などで起こるL'Ameriqueなどの文字列ではLとAがくっついてしまう場合があります。このような場合には以下のように記述します。

```
pos f' 60 space [T V W];  
pos L' -50 quoteright' 60 A;
```

最初の例では「f」の後に「スペース」「T、W、Vのいずれか」の3グリフが続く時のみカーニング値 +60が設けられます (カーニングが設定されるのは'でマークされたグリフの直後、つまりこの例ではfとスペースの間です)。2つ目のL'Aでは、Lとquoterightの間は-50、quoterightとAの間は60に設定され、結果としてquoterightは左に押し込まれてLA間にスペースが少し追加されます。これはL'Aという3グリフが並んだときのみ発動し、L'Oやl'Aなどでは発動しません。

10 PostScript ヒンティング

10.1 ヒンティング

PostScript ヒンティングは低解像度下でのPostScriptフォントの見た目を改善する技術のことで、CFFベースのOpenTypeフォント（拡張子は通常.otf）に対して設定することができます。

フォントをデジタル画面上でピクセルに落とし込む役割はラスターライザーというソフトウェアが担います。PostScriptフォントでは、フォント単体があらゆる機能を担うことは困難だと考え、ピクセル化の作業についてはラスターライザーが独自に判断することになっています。しかしフォントにその判断の補助となる情報（ヒント）を埋め込むことで、ラスターライズの結果を改善することができます。

ヒンティング情報のほとんどは、グリフのどの部分が重要であるか、逆にどの部分を小サイズ時に犠牲にしてよいかを定義するものです。これを実現するために2種類のヒントが存在します。まずはフォント全体に適用されるヒントで、「フォントレベルのヒント」や「フォント全体のヒント」などと呼ばれます。フォントのステム値やアラインメントゾーンがその例です。2つ目は各グリフに埋め込まれ、アウトラインを変形させてピクセルのグリッド上に載せるための情報で、「グリフレベルのヒント」などと呼ばれます。グリフレベルのヒントにはステムヒントやゴーストヒントがあります。

もっとも効率がよい方法はフォントレベルのヒント関連情報をしっかり定義し、その後グリフレベルのヒンティングはオートヒンティングに任せることです。

ヒンティングは、フォント内のステム幅が規則的であるなど、一貫したデザインがある場合にのみ有効です。手書き風や絵文字、装飾用ボーダー、ラフなテクスチャのあるフォントのようにアウトラインがきわめてイレギュラーになる場合、ヒンティングは役に立ちません。また高解像度のディスプレイやMac OSX、iOSなどヒンティング情報が無視される環境では、ヒンティングはファイルサイズの無駄にしかありません。このような場合、ヒンティング情報を付加しないことをお勧めします。

PostScript ヒンティングの目標は低解像度下で一貫性の高いシャープな文字を表示するということを忘れないでください。つまりラスターライザーは与えられたサイズ内でアウトラインを変形させ、ピクセルグリッド上に表示させます。また別の言い方をすれば、ヒンティングの仕事とは形を守るのではなく、その真逆であるということです。もしいかなる場合でもアウトラインの変形を望まない場合（例えばストロークが連綿するスクリプト書体など）、ヒンティング機能は使用しない方が得策でしょう。

10.2 フォント全体のヒンティング

グリフレベルのヒントを設定する前に、フォント全体で使用されるいくつかのパラメータを設定する必要があります。これらフォントレベルでのヒンティング情報は通称「PostScript プライベートディクショナリー」に保管されます。詳しくは以下をご覧ください（英語）：

typophile.com/files/hinting.pdf

partners.adobe.com/public/developer/en/font/T1_SPEC.PDF
(特に35～45ページ)

vimeo.com/38364880

10.2.1 スタンダードステム

ステム幅とはフォントの文字の線幅のことです。垂直ステムとは文字の縦線の幅です（例：大文字Iの太さ、Oの左右のカーブの太さなど）。また、水平ステムとは文字の横線の幅です（例：セリフの厚さ、A H f tのクロスバーの太さ、Oの上下のカーブの太さなど）。

そして最後に、スタンダードステムです。これはステム幅の平均のことで、フォント内の様々なステム幅のうち最も代表的な値のことです。オートヒンティングのアルゴリズムがステムを認識し適正なヒントを施すには、まずスタンダードステムを適切に設定することが重要です。そうすることでラスタイザーはピクセルへのレンダリング、特に様々なステムの太さを低解像度下で整える処理を最適化できます。

スタンダードステムの値はフォント情報の「マスター」タブの垂直ステムと水平ステムの欄に入力します。代表的な値だけを、なるべく件数を絞って設定してください。値はフォント情報の「マスター」タブの垂直ステムと水平ステムの欄に入力します。もし非常に近い値がある場合は、その2つの平均値に統合するのがよいでしょう。ステムの太さを素早く測るには、ステム上の2つのポイントを選択し、情報パネルでその選択範囲の幅を確認するか、ものさしツール（「3.9 計測」、p.36）に切り替えます。

例えばフォント内の垂直ステム幅を計測した結果68、71、72、74、75、82、83、85となっている場合は、70台前半と80台前半に山があるため、それぞれの平均の72と83あたりを選ぶのがよいでしょう（このような場合、たいてい68～75は小文字、82～85は大文字のステム幅だと想像できます）。スタンダードステムの値を1つだけにすると、フォント内のすべてのステムは小ピクセルサイズ下では同じ太さに見えるようになります。

理論上は垂直ステム欄と水平ステム欄でそれぞれ12個までスタンダードステムを設定できます。しかし、なるべく代表的な値に絞っていくと、ステム値は1個か2個に収束するはずで、2個の場合は大文字と小文字の垂直ステムだったり、水平ステムの場合はセリフとそれ以外の水平線などです。2つ目や3つ目のステム値を入力する場合は、サイズによってはステム幅に差が出る可能性があることを留意しておきましょう。例えば70と80の

2つの垂直ステムがある場合、フォントサイズによっては最初のステムが2ピクセルで表示され、後者のステムが3ピクセルで表示される可能性があります。ステム数を欲張ると表示結果がばらつく可能性があるのです。

それぞれのステム幅のフィールドに入力する最初の値は最も重要です。ここにはフォント内で最も代表的なステム幅、例えば小文字のステム幅を入力しましょう。入力されたステム幅はヒンティング以外にも「変形」フィルタの「視覚的傾斜」や「スマート角丸」フィルタなど、他の機能にも使われることとなります。2つ目以降のステム値はヒンティングのみに使用されます。水平ステムの値はTrueTypeヒンティングにも使用されます。詳細は「11.3 手動インストラクション」(p.131)を参照してください。

マルチプルマスターのファイルではステム値は補間されるため、同じ個数のステム値を各マスターに入力し、その順番も必ず揃えてください。

10.2.2 アラインメントゾーン

フォントの欧文が非常に少ないピクセル数で表示されているとき、xハイト付近のアウトラインは同じ高さに整列（つまり同じ高さのピクセル数で表示）されるべきです。これはf h kのアセンダやg p yのディセンダ、また大文字の高さでも同様です。また当然ながら、低解像度下では欧文のすべての文字のベースラインが揃うべきです。

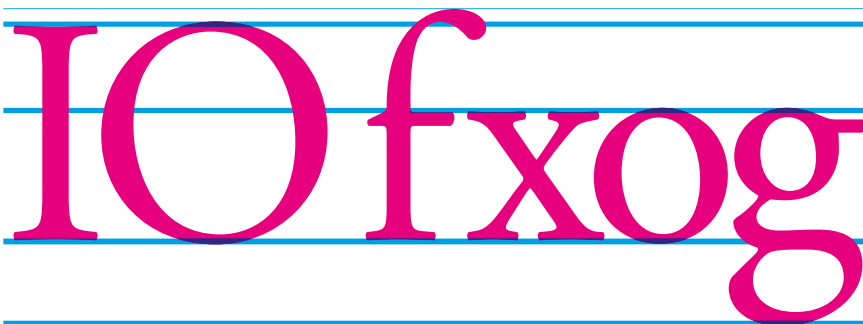
しかし通常の書体デザインではこれらの文字のアウトラインはぴったり揃っているわけではありません。例えば、小文字のoの下は視覚調整のためにベースラインよりもやや下にはみ出すようデザインしますが、小文字のnは完全にベースライン上に揃えるようデザインするのが普通でしょう。また大文字Aのアペックス（頂点）は大文字Hの高さより少し高めにしないと、背が低く見えてしまいます。これら標準の高さをはみ出る部分はよく「オーバーシュート」と呼ばれ、その度合いは大抵10から15ユニット程度でしょう。

アラインメントゾーンとはラスターライザーに伝えるオーバーシュートの情報のことです。小サイズではオーバーシュートが再現される必要はありません。このため低解像度下ではアラインメントゾーン内に存在するアウトラインは縦方向に圧縮され、ゾーンの目標位置に整列されます。

アラインメントゾーンは2つの値から構成されています。1つは開始位置、もう1つはサイズです。開始位置とはxハイトやアセンダなど、ゾーンの基準となるy位置のことです。開始位置はオーバーシュートが整列する目標の高さとなるので、時として「フラット・エッジ」と呼ばれることがあります。サイズはオーバーシュートの最大値です。もしオーバーシュートが目標の高さ（xハイトやキャップハイト、アセンダー）を上を越える場合は、サイズの値は正でなければいけません。このようなゾーンを「トップゾーン」と呼びます。もし逆にオーバーシュートが目標の高さ（ベースラインやディセンダー）を下を越える場合、サイズは負の値でなければな

らず、そのようなゾーンは「ボトムゾーン」と呼ばれます。

典型的なアラインメントゾーンの設定：アセンダー、キャップハイト、xハイトに正の値であるトップゾーン、ベースラインとディセンダーに負の値であるボトムゾーン。



正確に言えば、アラインメントゾーンの最大値はblueScale値の制約を受けていて（以下を参照）、 $240 \div (240 \times \text{blueScale} - 0.98)$ 以上のゾーン幅を設定してはいけません。

アラインメントゾーンのサイズは小さければ小さいほどよいので、「安全策として」無駄に大きく設定したりはしないでください。いかなる場合でも、最大値は25ユニットです。トップゾーンは5つまで、ボトムゾーンは6つまで設定できます。ゾーン同士は重なってはならず、最低1ユニットの間隔が必要です（より広いことが望まれます）。ベースラインのゾーンのサイズは0に設定してはいけません。

「フォント情報 > その他の設定」の「グリッドのユニット間隔」を標準以外の値（つまり1以外）に設定すると、座標が四捨五入されて整数値化するので（「7.5.1 グリッドの間隔とグリッド細分」、p.94）、アラインメントゾーンも対策として上下方向に1ユニットずつ拡張する必要があります。つまり開始位置を1ユニット広げ、ゾーンサイズは2ユニット広げます。ベースラインのゾーンだけは開始位置を必ず0にしなくてはなりません（サイズは1広げます）。

10.2.3 カスタムパラメータ

アラインメントゾーンとステムの他にも、フォント全体のプライベートディクショナリーにはヒンティングに関する値が各種あります。blueScale、blueShift、そしてFamily Alignment Zonesです。Glyphsではこれらの値をカスタムパラメータとして設定できます。これらのパラメータの詳細を知りたい場合は「17.3 カスタムパラメータ」（p.184）を参照してください。

10.3 オートヒンティング

アラインメントゾーンとスタンダードステムの2つのフォント全体のパラメータが適切に設定されれば、フォント出力ダイアログの「すべてのグリフをオートヒント」オプションをオンにしてオートヒンティングのアルゴリズムに作業を任せることができます。またAutohintカスタムパラメータにより強制発動させることができます（フォント出力ダイアログでオートヒントをしない設定にしても、このパラメータがあるインスタンスはオートヒントが実行されます）。

Adobeのアプリケーションでヒンティング結果をテストしましょう。テストのテキストを打って、少ないピクセルで表示されるようにズームアウトします「3.12.7 Adobeアプリケーションでのプレビュー」(p.46)。次にシステム環境設定の「アクセシビリティ」にあるズーム機能を使って、そのテキストを拡大してヒンティングの結果を確認しましょう。必要に応じてフォント情報から設定を変更したり、特に問題のあるグリフを手動でヒンティングして再出力、確認する作業を繰り返します。手動のヒンティングに関しては「10.4 グリフレベルの手動ヒンティング」(p.126)を参照してください。

10.3.1 フレックス・ヒント

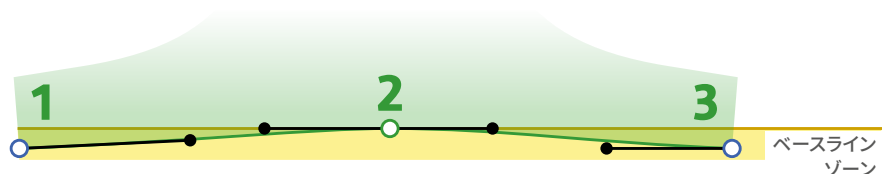
もしフォントに緩やかに凹んだセリフやステムがある場合、オートヒンターは自動的にフレックス・ヒントと呼ばれるヒントを付加し、浅いカーブを低解像度下で平坦にしてくれます。フレックス・ヒントはフォント出力時に自動で追加され、手動で追加することはできません。これが設定されるには以下の条件を満たしていなければいけません。

まず blueShift の値が、これらのディテール + 1 ユニットの値に指定されていること。例えば凹型のセリフのへこみが5ユニットであれば、blueShift は6以上にすべきです。

2つ目に、アウトラインの条件がいくつかあります。凹み曲線の形状は3つのオンカーブポイントから構成される2つのセグメントでなければいけません。2つのセグメント形状は対称である必要はありません。縦方向の場合、1つめと3つめのポイントは同じx座標 (OptimaのIのようなステム形状の上端と下端のポイントが同じx座標)、または横方向の場合はy座標 (PalatinoのTのようなセリフの右端と左端が同じy座標) に並んでいる必要があります。最大4本あるハンドルは完全に垂直や水平になっている必要はありませんが、3つのオンカーブポイントは極点でなければいけません。曲線の深さの限界は20ユニットです。

最後に曲線のセリフの場合、3つのポイントがアラインメントゾーンに確実に収まっているのが理想的です。最高の結果を得るには、2番目のポイント (中心) をフラット・エッジ、つまりゾーンの目標の高さにぴったりと乗せましょう。そして両端のポイントはアラインメントゾーン内に存在する必要があります。つまり下図のように曲線状のセリフがベースラインにある場合、両端のポイントは曲線の深さの分だけベースラインから下のボトムゾーンに存在している状態だということです。

フレックス・ヒント：ポイント1と3は全く同じ高さに揃っていてアラインメントゾーンの内側にあり、ポイント2はゾーンのフラット・エッジに乗っている必要があります。ハンドルは1~3番までのポイントの空間を超えてはいけません (1~3番のポイントはいずれも極点でなければいけません)。



10.4 グリフレベルの手動ヒントイング

GlyphsのPostScriptヒントイングでは自動と手動のヒントを共存させられるようになっています。そのため、手動でヒントを設定する前にオートヒントイングで可能な限り品質を上げておくことをお勧めします。低解像度下でうまく表示されないグリフにのみ手動調整を行きましょう。

手動ヒントイングと自動ヒントイングは、1つのグリフに共存することはできません。手動ヒントイングが施されたグリフには自動ヒントイングが施されず、出力時に手動ヒントイングデータがそのまま採用されます。つまり手動ヒントイングを行う場合は、そのグリフの面倒を最後まで見なければいけません。

グリフ単位で手動で施せるヒントには「ステムヒント」と「ゴーストヒント」の2種類があります。ステムヒントは垂直または水平のステム、またはセリフやクロスバーを明示するのに使います。ゴーストヒントはサンセリフのステムのようにセリフがなく、水平ヒントが施せない場合にアウトラインの上端、下端を明示するのに使います。

アラインメントゾーンと同様、水平ヒントはxハイトやアセンダーなどの垂直方向の整列に重要なものです。低解像度下では、ラスターライザーは水平ヒントが施されてアラインメントゾーンに存在する部分すべてを整列するよう試みます。水平ヒントの上端もしくは下端は、ポイントが揃うべき高さ（xハイトやキャップハイト、アセンダー）と一致している必要があります。ヒント同士が重なってはいけないため、離れたステムも1つのヒントが面倒を見るようになっています（例：mの2つのショルダー）。

大文字Bの右側の縦ステムや数字の8の縦ステムなどは2つのステムヒントを設置する必要があるでしょうが、これらはオーバーラップすることが多いでしょう。PostScriptヒントイングでは厳密に言えばオーバーラップは許されていませんが、重なっても不具合は生じません。この場合、Glyphsは自動的にヒントリプレースメントという情報を仕込み、それぞれのステムが描画される時に不要な方のヒントを自動的に解除するようにします。そのため、ステムヒントのオーバーラップは気にしなくてよいのです。

マルチプルマスターでは、最初のマスター（フォント情報の最も上にあるマスター）にあるヒントだけが使用され、他のマスターに流用されません。この場合、手動で置かれたヒントは必ずポイントとリンクされているようにしてください。さもないとGlyphsはどのようにヒントを補間すればよいか分からなくなります（「10.4.1 ステムヒント」、p.126）。

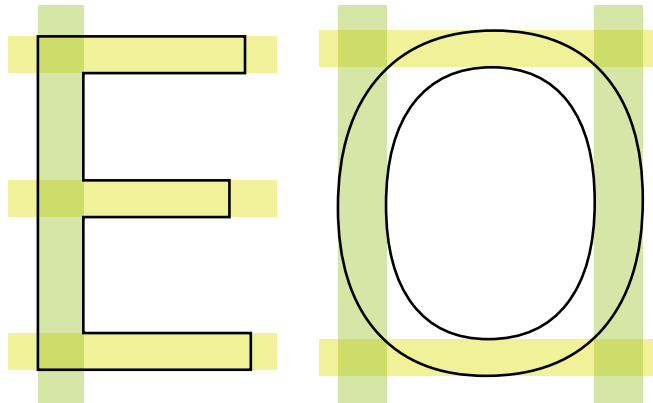
10.4.1 ステムヒント

ステムヒントをグリフに追加するには、右クリックしてコンテキストメニューから「水平ヒントを追加」または「垂直ヒントを追加」を選んでください。2つの数字が書かれたバッジの付いた灰色のバーが現れます。最

初の数字はヒントの基準点、2つ目はヒントの幅を表しています。

2つのポイントが選択された状態でヒントを追加すると、そのヒントはポイントに自動的にリンクされます。このように2つのポイントをペアにしてリンクしたヒントを追加する方法は、複数のペアを選択している場合でも（別々のアウトライン上にペアが存在する限り）有効です。最適な結果を得るためには、ヒントは常に極点にリンクさせてください（詳細は「3.3.13 極点と変曲点」(p.25)を参照)。

垂直ステム（緑）と水平ステム（黄色）の指定結果。



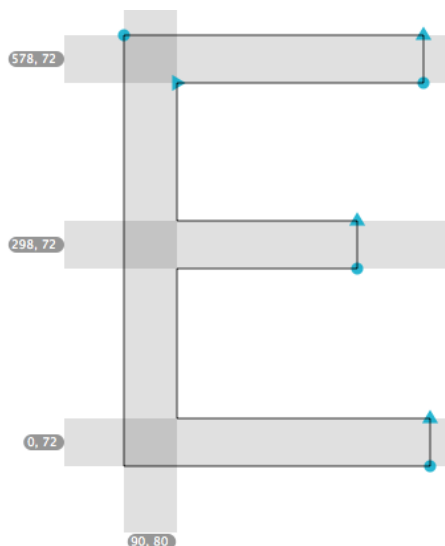
ヒントを選択するには数字バッジをクリックしてください。shiftを押しながら複数選択することもできます。ヒントが1つだけ選択されている場合、灰色の情報パネル（「表示 > 情報を表示」またはshift+command+I）にて数値を編集できます。また、tabキーとshift+tabキーで次のヒント、前のヒントに選択を移動することができます。

高さ140、幅40と書かれた数字バッジのある水平ヒント。



ヒントをグラフィカルに編集する場合、ヒントの端にある青いマークをドラッグしてください。青の丸は基準点を、3角は幅と方向を表します。なお、マークをポイントの上にドラッグすると、Glyphsはそのポイントの座標に自動的にリンクします。リンクした状態でポイントを動かすと、ヒントも自動的に追従します。ヒントを削除するには、ヒントを選んでdeleteまたはbackspaceキーを押してください。

ヒントの青い丸と三角がポイントとリンクした状態。ステムヒントの幅はすべて正の値でなければいけません。つまり水平ヒントの場合は三角が丸の右側、垂直ヒントの場合は上側になければいけません。

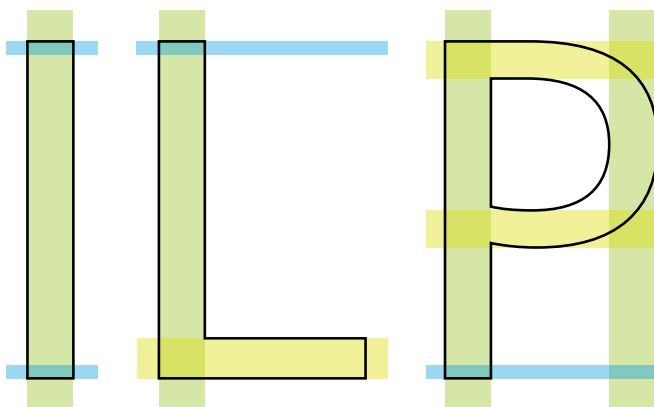


出力される OTF では、すべての PostScript ステムヒントは正、つまり 0 より大きい幅でなければいけません。もし誤って負の値で追加された場合、Glyphs は出力時にすべて正になるように修正します。

10.4.2 ゴーストヒント

アウトラインの上端と下端を垂直方向のメトリクスに整列させたくても水平ヒントを指定できない場合には、ゴーストヒントが使えます。例えばサンセリフの大文字の I です。I の上端はキャップハイトに、下端はベースラインに揃わなければいけません。セリフ付きのデザインであればセリフに水平ステムを与えれば済むことですが、サンセリフでは水平ステムを与えられるような目立った特徴がありません。この場合、I の上端に上端ゴーストヒントを、下端に下端ゴーストヒントを追加します。

通常のステムに加えて、ゴーストヒント（青）を追加した状態。



ゴーストヒントを追加するにはポイントを1つだけ選択してコンテキストメニューから「水平ヒントを追加」を選択します。もしくは水平ヒントのバッジを選択した状態で右クリックし、コンテキストメニューから「ゴーストヒントに変換」を選びます。ゴーストヒントのバッジには基準と方向

のみが表示されます。下向きの矢印が下端ゴーストヒント、上向きの矢印が上端ゴーストヒントであることを表します。青い丸をポイントにドラッグすることで、そのポイントにリンクさせることができます。上端、下端を切り替える場合は、灰色の情報パネルの「上」または「下」アイコンをクリックしてください。

y座標100に配置された上端ゴーストヒントと、50に配置された下端ゴーストヒント。



10.4.3 マルチプルマスターのヒントイング

マルチプルマスターのフォントでは、最初のマスターにのみヒントイングを行ってください。ヒントがポイントにリンクされていてパスも完全に互換していれば、フォント出力の際にヒント情報は他のマスターの対応するポイントに自動的に適用されます。最初のマスターとはフォント情報の「マスター」タブ内で最上位に位置するフォント（またはフォントウィンドウで左端のアイコンのマスター）のことです。

他のマスターに施した手動ヒントは使用されませんが、最初のマスターに手動ヒントがなければ他のマスターのヒントが採用されます。デザインによっては手動ヒントが施されたマスターを最上位にドラッグした方がよいかもしれません。例えば最初のマスターが非常に細いウエイトの場合、目的のポイントを選択して手動のヒントイングを施す作業はやりづらいかもしれません。このような場合、RegularまたはBoldマスターを最上位にした方がよいでしょう。

ブラケットレイヤーまたは逆ブラケットレイヤーを使用している場合（「4.4.2 特別なレイヤー」、p.50）、最初のマスター以外にもヒントイングを施さなければいけません。この場合、そのデザイン軸の最初のレイヤーを手動でヒントイングしなければいけません。

11 TrueType ヒンティング

11.1 インストラクション

TrueType フォントではかなり正確に各サイズでのアウトラインの変化を制御できます。この作業はよくヒンティングと呼ばれます。本書でもヒンティングと特に区別は付けませんが、正確にはインストラクションと呼ばれるものです。レンダリングの際、ピクセルがアウトラインの内側にあればそのピクセルは黒、外側にあれば白になります。インストラクションとは、低解像度なフォントサイズで表示品質を向上させるために、アウトラインそのものをサイズに合わせて変形させる技術です。そのためどのような種類のインストラクションでも、ピクセルの位置に合わせてアウトラインのポイントを操作するという点で共通しています。PostScript ヒンティングではステムの存在などだけを教えてあとはラスタイザーの性能に頼るためピクセル化はラスタイザーの作業になりますが、TrueType のインストラクションではデザイナーが自ら詳細に挙動を設定するため、ラスタイザーに高度な処理機能は必要とされません。ただし TrueType フォントのレンダリング結果は OS ごと、アプリケーションによって多少の差は生じます（同じ Windows でもバージョンが違えば結果は異なります）。

インストラクションを施すと、TrueType ベースフォントの低解像度下でのレンダリング結果が向上します。TrueType ベースフォントとは、TrueType 形式の OpenType（拡張子 .ttf または .otf）、EOT Web フォント (.eot)、TrueType ベースの WOFF フォント (.woff または woff2) です。PostScript のそれと同様、ヒンティングの目標はグリフ形状をそのまま表示するのではなく、ピクセルグリッドに合わせて適用させることにあります。これは低解像度下ではかなり激しい変形になることがあります。

TrueType フォントは2次ベジェ曲線を使っているため、カーブを描画する計算式も描画方向も違います。しかし Glyphs のインターフェイスでは PostScript 方式の3次ベジェ曲線しか扱いません。TrueType に出力する際、パスは自動的に TrueType 用のものに変換されますが、この際には手動で施されたインストラクションも含まれます。

11.2 オートヒンティング

TTFAutohint の詳細については公式 Web サイトを参照してください：
freetype.org/ttfautohint

Glyphs は Werner Lamberg 開発によるオープンソースの TTFAutohint というオートヒンティングツールを使い、TrueType インストラクションを自動でフォント出力時に施します。それぞれのインスタンスに「TTFAutohint Options」カスタムパラメータを加えることでオートヒンティングのオプションを設定できます。各パラメータの詳細は「17.3 カスタムパラメータ」(p.184) を参照してください。

11.2.1 手動または自動のインストラクション

これはグリフによっては任意のグリフに手動ヒントを仕込むことができるPostScriptのワークフローとは対照的です。

TrueTypeインストラクションはフォントのかなり深い部分まで関わってくるため、TTFAutohintとTrueTypeの手動インストラクションツール(ショートカットI)は互換性がありません。つまりフォント全体のインストラクションを手動または自動で行わねばならず、両方の混在はできないということです。TTFAutohintを使用すると手動のインストラクションはすべて無視されます。

11.3 手動インストラクション

Glyphsは4種類のTrueTypeインストラクションを用意しています。整列、ステム、補間、そして斜め線です。これらのインストラクションを挿入するにはまずTrueTypeインストラクションツール(ショートカットI)に切り替え、特定の数のポイントを選択し、コンテキストメニューから任意のインストラクションを選びます。コンテキストメニューは右クリックまたはcontrol+クリックで、ポイント上に限らず編集ビュー内のどこからでも呼び出せます。

どのようなポイントを選択しているかに関わらず、コンテキストメニューには常に「オートヒント」と「ヒントを全削除」のメニュー項目が表示されます。「オートヒント」は現在のグリフに適切と思われるインストラクションを施します。ヒントを全削除を選ぶと、現在のグリフのインストラクションをすべて削除します。これはすべてのインストラクションを選択して(command+A)、backspaceまたはdeleteキーを押しても同じ結果が得られます。インストラクションの影響で位置が動くポイントは「編集済み」、動かないポイントは「未編集」と呼びます。

マルチプルマスター環境では、最初のマスターに施したインストラクションのみが採用されます。またすべてのマスターの互換性が取れていることを前提として、そのインストラクションは全インスタンスに適用されます。


オーバーラップしているアウトラインは合成する必要はなくそのままにしておいて構いませんし、オーバーラップしている部分にインストラクションを施しても構いません。出力時に重なったアウトラインが除去される際、TrueTypeインストラクションは自動的に位置を変えて、合成後のポイントに移動します。ただしこれは合成前のポイントが合成後のそれとそれなりに近いことが条件ですので、狙い通りにインストラクションを機能させるには合成前のポイントを合成後の仮想ポイントに近づけるか、アウトラインを合成してしまう必要があります。

手動でグリフのインストラクションを施す際、まずは縦方向の整列、つまりベースラインやxハイト上の整列に注力しましょう。それが完了したら縦方向のステム幅、すなわちセリフの太さやH t fのクロスバーの太さ、Eの上下カウンターの高さなどの細かい内容に移るようにしましょう。

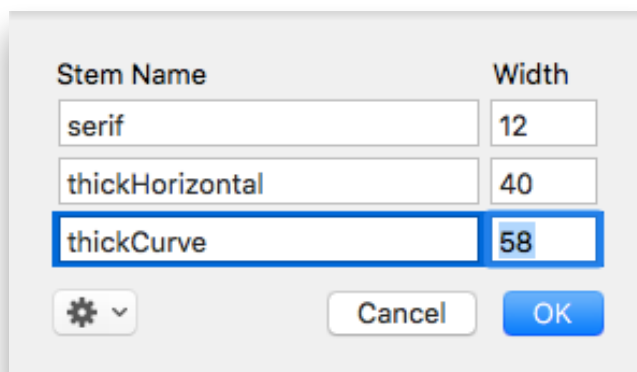
MicrosoftのClearTypeなどに代表される現代のサブピクセルレンダリングでは、横方向のレンダリングを自動的に処理するため、横方向に関するインストラクション（垂直ステムの幅など）は無視してください。TrueTypeインストラクションツールでも、設置するのは前述した縦方向に関するインストラクション、すなわち水平ステムなどに限られています。

11.3.1 水平ステムとゾーン

TrueTypeインストラクションは「フォント情報 > マスター」のアラインメントゾーンと水平ステムの情報を利用します。詳細は「10.2 フォント全体のヒンティング」(p.122)を参照してください。

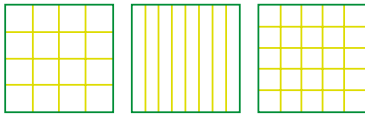
TrueTypeではPostScriptと比べてより多くのステムをサポートしています。TrueType専用で水平ステムを設定したい場合は、メニューバー「ファイル > フォント情報」のマスターにカスタムパラメータ「TTFStems」を追加します。値をクリックしてオプションを表示し、メニューでPostScript用ステムをインポートするか、または新しいステムを追加して任意の名前を設定します。マルチプルマスター環境では、必ずTTFStemsカスタムパラメータも互換するようにしてください。つまりすべてのマスターで同数かつ同名のステムが登録されている必要があります。パラメータは他のマスターにコピー&ペーストすることができます。

スタンダードなステムの値を定義する作業はPostScriptヒンティングにおけるステムの定義の仕方と非常に似ていますが、TrueTypeではステムの数を増やしたほうが安全です。稀にしか登場しない水平ステムを登録することもできます。これはステムヒントの段階で名前の付いた専用のステムを適用できるからです。ただし、登録するのは太さの差が明瞭なステムのみにするのが理想です。幅の違うスタンダードステムは、その太さがわずかな違いであっても、表示サイズによって1ピクセル違う太さで表示されることがあります。新たに水平ステムを追加する場合は十分に太さの違いがあるものに限るべきでしょう。この違いがむしろあなたのデザインで重要な場合は、それぞれにステムを設置して区別を付けるようにしてください。



11.3.2 ラスタライザープレビュー

Tip: インストラクションが施されたアウトラインとピクセルプレビューの結果を他のインスタンスで確認したい場合、ウインドウ下のプレビューから他のインスタンスを選択します。



4×4のグレースケール (左)、8×1のClearType (中)、5×5のDirectWrite (右)

TrueType インストラクションツールでグリフを編集している時は、アウトラインの背景に赤い別のアウトラインが表示されます。これはインストラクションが施されたアウトラインの変形後の結果であり、その内容はインストラクションの内容、フォントサイズ、そして想定するラスタライザーによって変化します。ラスタライザーはグレースケール、GDI ClearType、DirectWriteの3つから選択できます。また、それに加えてピクセルプレビューが背景に表示されます。実際のレンダリングの結果はフォント表示環境によって大きく変化するため、ピクセルプレビューはあくまで参考程度のもので考えてください。

グレースケールレンダリングでは4倍オーバーサンプリングの結果を表示します。つまり1つのピクセルを表示するために $4 \times 4 = 16$ 分割したサブピクセルで計算しているということです。それぞれのサブピクセルの中心がアウトラインの黒側か白側に入っているかを確認し、その個数によって16段階のグレースケールからピクセル色を決定しています。主にWindows XPで使われているGDI ClearTypeは8×1のオーバーサンプリングを使用します。つまり水平方向は8分割し、垂直方向には分割しません。Windows 7以降で使われているDirectWriteは5倍 (5×5) のオーバーサンプリングを使用しています。

これらのモードを切り替えると、アウトラインのピクセルグリッドの吸着具合に違いがあるのがわかります。インストラクションのあるポイントは、グレースケールでは4×4分割されたサブピクセルに吸着し、DirectWriteでは5×5のサブピクセルに吸着するのに対し、ClearTypeでは1ピクセルごとに吸着します。

情報パネル (shift+command+I) では、レンダリングモードを切り替えることができます。その左隣のサイズ変更フィールドではPPMでフォントサイズを変更できます。

Windowsではデフォルトで画面解像度が96ppiと設定されています。実際のPPMサイズを計算するには、Windows内のポイントサイズに4/3を掛けてください。例えば目的のサイズが12ポイントとすれば、Windowsでは $12 \div 3 \times 4 = 16$ ピクセルで表示されます。Windows画面上のピクセルサイズをポイントに逆算したい場合は $\frac{3}{4}$ を掛けます。例えば20ピクセルはWindows内部では $20 \times 3 \div 4 = 15$ ポイントです。

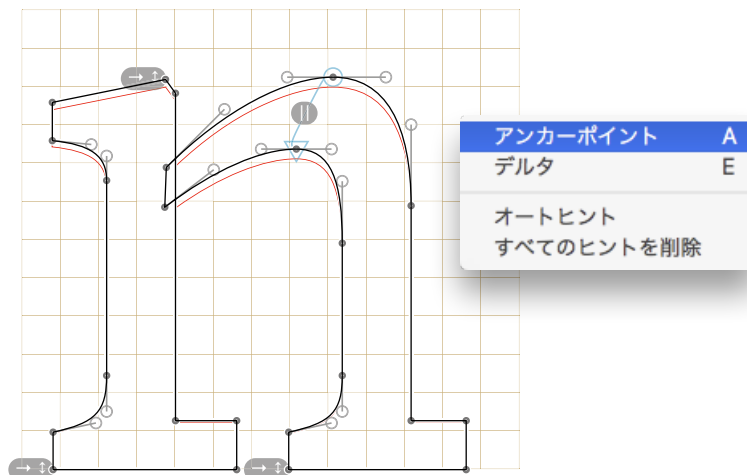
11.3.3 アンカー

Tip: TrueType インストラクションのショートカットはQWERTY配列キーボードのAからGに並んでいます。ヒンティングツール使用時はこれらのショートカットが優先されますので、Aを押しても注釈ツールには切り替わりません。

アンカーインストラクションを追加するには、ポイントを1つ選択し (なるべくアラインメントゾーン内のもの)、右クリックまたはcontrol+クリックでコンテキストメニューを呼び出して「アンカー」を選択してください (ショートカットA)。そのポイントはステムや補間など他のインストラクションの影響を受けないようにしなければいけません。

アンカーインストラクションはポイントを直近のピクセルの端に吸着させます。その方向は上または下を向いた小さい青3角形で表示されます。アンカーの方向を変更するには、バッジをクリックして選択し、情報パネルに現れるアイコンをクリックしてください。◄は必ず直近のピクセルの端に吸着（初期状態）、▲は必ずピクセルの上端に吸着、▼は必ずピクセルの下端に吸着、✖は吸着しません。

上端のセリフと下端のセリフでは、1つのアンカーインストラクションで同じ高さの他のセリフがすべてピクセルグリッドに吸着します。



もしポイントがステムヒントの基準点となる場合、Glyphsはそこに自動的に（かつ不可視の）アンカーインストラクションを挿入しますので、手動で挿入する必要はありません。

11.3.4 整列

整列インストラクションを追加するには編集済みポイントと未編集のポイントを1つずつ選択し、コンテキストメニューから「整列」を選択します (F)。整列インストラクションは未編集ポイントを編集ポイントに追従させ、編集ポイントと同量動くようにするものです。

整列インストラクションを2つの未編集ポイントに適用させると、最初の方のポイントには自動的に（かつ不可視の）アンカーインストラクションが挿入されます。通常はアラインメントゾーン内にある方のポイントが基準点になります。整列の基準を反転するにはコンテキストメニューから「反転」を選択します。

整列インストラクションは、ステムヒントで発生するポイントの移動を他のポイントでも繰り返したいときに便利です。例えばローマン書体の左側のセリフにステムヒントを施したあとは右側のセリフにも同様にステムヒントを施すのではなく、左側のセリフの編集済みポイントと右側のセリフの未編集ポイントを整列させます。

11.3.5 ステム

2つまたは4つのポイントを選択し、コンテキストメニューから「ステム」

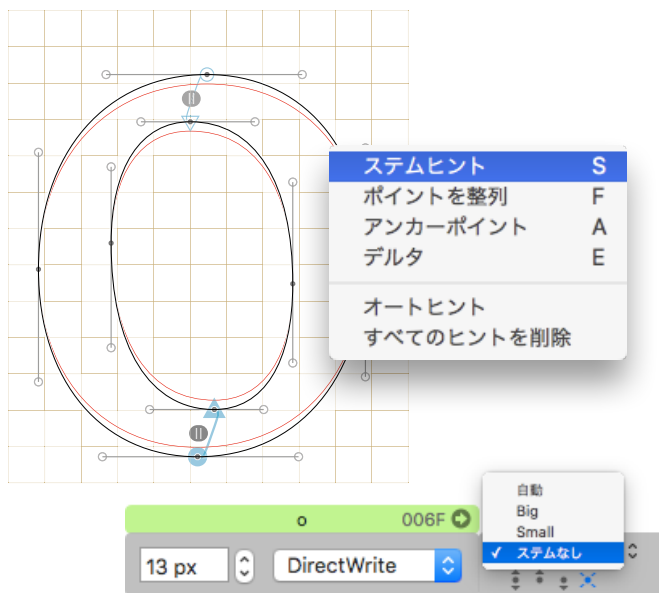
を選択します (S)。青い丸がインストラクションの基準点を表し、3角が移動対象のポイントを表します。基準点には自動的に不可視のアンカーインストラクションが挿入されます。ただし基準点がすでに他の TrueType ヒント、例えば補間などによって編集済みの場合はそちらが優先されます。

Tip: 上端吸着と下端吸着の2つのオプションは直近吸着と違ってサブピクセルではなくピクセルに整列します。そのため、オーバーサンプリングを使用するレンダリングでは結果が荒くなります。これらはXPのGDIレンダリング向けのオプションです。

ステム (またはリンク) とは2つのポイント間の距離を制御するためのものです。ステムの基準点の座標は自動のアンカーまたは任意に挿入された他のインストラクションによってまず位置が固定され、その後に2つ目のポイントの位置が決まります。最初のポイントの固定基準は直近のオーバーサンプリンググリッド (サブピクセルグリッド)、またはアラインメントゾーン内のポイントかアンカー付きのポイントであればピクセルグリッドです。ステムの吸着方針を変更するには灰色のバッジをクリックし、情報パネルに現れるアイコンのいずれかをクリックします。◄ は必ず直近のグリッドの端に吸着 (初期状態)、▲ は必ずピクセルの上端に吸着、▼ は必ずピクセルの下端に吸着、✖ は吸着しません。

基準点と対象点との距離はフォント情報のマスターの TTFStems カスタムパラメータで設定する水平ステムによって決まります。パラメータの設定方法については「17 補足資料」(p.174) を参照してください。インストラクションが参照するステムを変更するには、バッジをクリックして情報パネルから他の任意のステムを選択します。「ステムなし」を選択すると、それぞれのポイントを直近のピクセルの端に単純に吸着させるようにできます。つまりステムなし設定のヒントは整列インストラクションと同様の扱いになります。

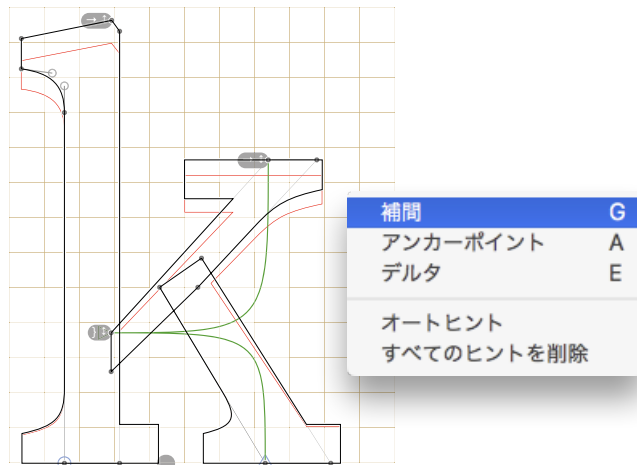
小文字 o の上端と下端のステムにヒントが配置された状態。下端ステムが選択され、情報パネルで TTF 用ステムが選択されています。



ステムヒントを右クリックまたは control + クリックして「反転」を選択すると、インストラクションの方向が逆転します。ステムインストラクションの基準点は、できるだけアラインメントゾーン内のポイントまたは近い

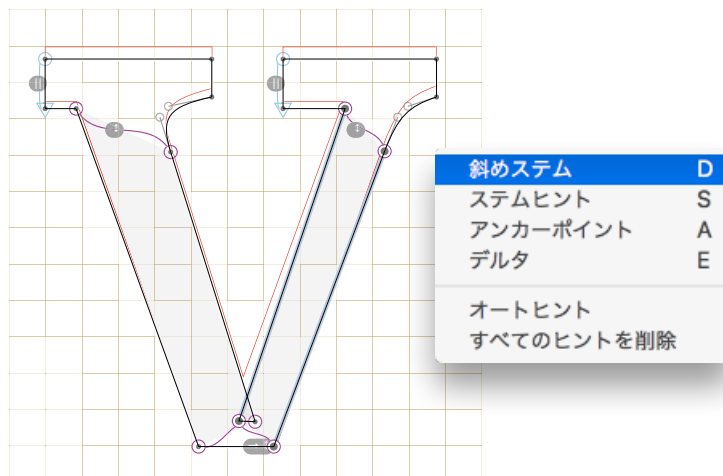
これは縦に3重の構造がある文字、例えばB E G K Rの中間のバーの高さのバランスを取りたいときに便利です。

この小文字kの例では、垂直ステムと斜めステムとの交点の位置を補間ヒントにより定めるように設定しています。基準となるのは下端の整列ポイントと、アーム上端のアンカーポイントです。



11.3.8 斜線

サブピクセルレンダリングでは赤青緑のサブピクセルの色を個々に制御することにより、縦ステムの表示精度を3倍に引き上げることができますが、横ステムの太さは1ピクセル単位のままです。そのため、斜め方向のステムは水平ステムと比べて最大3分の2ピクセル細すぎるか太すぎるように表示されることがあります。斜線インストラクションでは、指定の水平ステムに太さの変化を合わせようと調整が行われ、斜めステムと水平ステムの相対的な太さの関係が守られるようになります。斜線インストラクションを適用するには4つのポイントを選択してコンテキストメニューから「斜めステム」を選択し、情報パネルから任意の水平ステム幅を指定します。



このヒントは斜めステムが特徴的なグリフ、例えばk v w x y zなどに効果的です。

11.3.9 未編集ポイントの自動補間

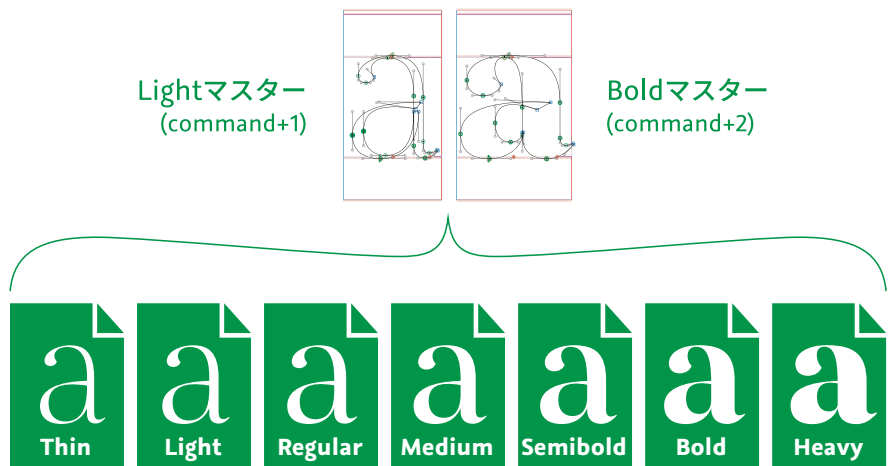
残された未編集ポイントの位置は、他の編集済みのポイントから自動的に補間計算されます。この自動補間は、編集済みポイントの位置調整が終わった後に実行されます。そのためグリフ内のすべてのポイントをヒンティングしなくても意図した結果が得られるようになっています。まずは上端と下端から始め、徐々に内側に作業を移していくようにしましょう。

12 マルチプルマスター

12.1 概要

Glyphs ファイルは、同じファミリーの複数スタイルをそれぞれ個別のマスターとして1つのファイルにまとめることができます。ここで両極のスタイル（例：LightとBold）をマスターにして、その中間のスタイルを「インスタンス」で定義すると、2つのマスターから自動的にそのスタイルを補間生成できるようになります。この状態を「マルチプルマスター環境」と呼びます。

少ない数のマスターから多数のインスタンスを補間、生成することができます。この図では、ウエイトの補間軸にある2つのマスター（LightとBold）をベースに、ThinからHeavyまでの7つのウエイトを補間生成しています。



Tip: ローマンとイタリックは形があまりにも違い補間はできないため、同じファイルに格納するのは避ける方が賢明です。詳細は「12.6 複数ファイル間でのファミリーの一貫性の確保」(p.XXX)を参照してください。

補間生成をするには「補間軸」の設定が必要です。補間軸とはマスターとマスターを結ぶ直線であり、この直線上に2つ以上のマスターを配置することでインスタンスの補間生成が行われます。補間軸として最も代表的なのがウエイト（例：LightからBold）とグリフ幅（例：CondensedからExtend）で、この2つはデフォルトで用意されています。この他にも、線幅のコントラスト、セリフの太さ、アセンダーやディセンダーの長さなど、さまざまな補間を自由に設定できるカスタム軸があります。

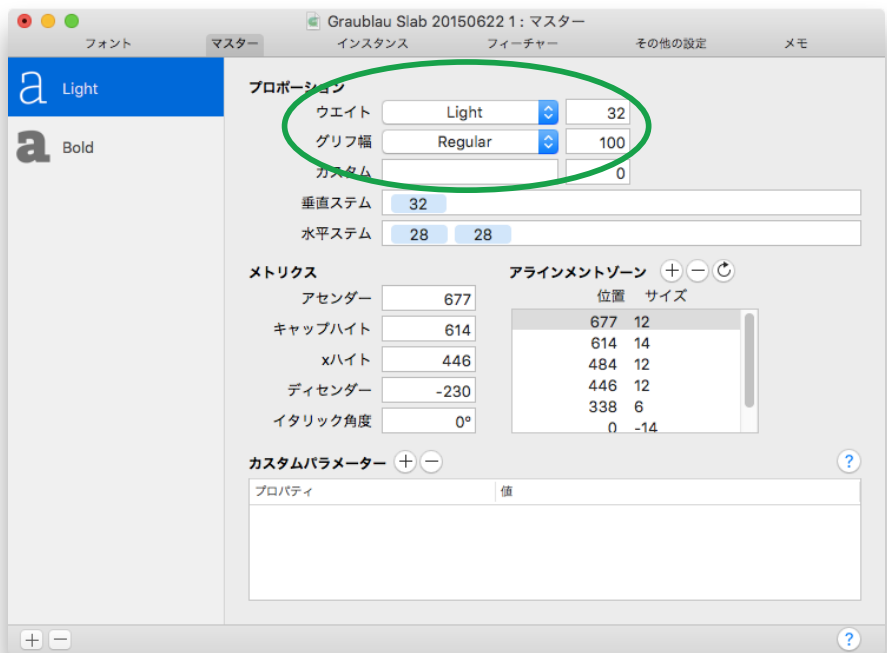
マスター同士の関係性は字幅とウエイト、カスタムの3軸の値によって定義されます。これらの軸を組み合わせることでデザインスペース（デザインの補間範囲）は直線だけでなく2次元、3次元にもなり得ます。またインスタンスをデザインスペース外に設置する、補外と呼ばれる処理も可能です。しかし現実的には補間範囲外では形の破壊が起こりやすく、マスターをかなり慎重に設計する必要があるのであまりお勧めしません。

マスターレイヤーはそれぞれ独立しています。つまり各レイヤーのポイントやアンカーが自動的にリンクされるわけではなく、あるマスターを編集している時に他のマスターが勝手に変更されることはありません。しかし最終的に補間をする際にはマスターが正しく互換している必要があります。互換とはすべてのポイント、アンカー、そしてコンポーネントの順

番や構造が他のマスターでも共通している状態です。非互換のマスターがあると補間に失敗し、出力されたフォントのグリフが空白になります。

12.2 マスターのセットアップ

マスターを追加するには、メニューバー「ファイル > フォント情報」の「マスター」タブを表示し、左下の+ボタンをクリックします。詳細は「7.2 マスター」(p.84)を参照してください。「ウエイト」「グリフ幅」「カスタム」に数値を入力することで軸を決めます。同時にデザインスペースも定義されます。



ウエイト軸の数値には、小文字nのステムの太さのような値を使用することをお勧めします。こうすることでインスタンスのステム幅を決定しやすくなります。次にウエイトのポップアップメニューからマスター名を選びます。この名前は標準で用意されているもので、レイヤーなどに表示されるマスター名と、メインウィンドウのツールバーアイコンにのみ影響するものです。フォントの書き出しには影響しません。最終的なフォントのスタイル名はフォント情報の「インスタンス」タブで定義します。

もし直線的な補間ではなく2軸、3軸の補間もしたい場合、「グリフ幅」や「カスタム」軸を使います。カスタム軸では任意のマスター名を定義できます。マスター名はウエイト、字幅、カスタム軸で定めた名前から作られます。

各マスターの軸の数値には十分注意してください。数値が全く同じのマスターがあるとインスタンスの補間に失敗してしまいます。

マスターはドラッグすることで順番を変更できます。マスターの順番は補間に影響を与えませんが、最初のマスターはヒンティング、カラーフォ

ントやマルチレイヤーフォントなどの作成で非常に大きな影響を与えます。どれを最初のマスターにするかは慎重に選択してください。

12.3 インスタンスのセットアップ

マスターに数値を与えてデザインスペースを確立したあとは、各インスタンスの数値を決定します。もしマスタータブで定めた最小値と最大値の範囲内（デザインスペースの内側）でインスタンスを作る場合、そのインスタンスは「補間」により生成されます。マスターの値の範囲を超えた数値を入力すると補外になります。インスタンスを管理するにはフォント情報の「インスタンス」タブを開きます。詳細は「7.3 インスタンス」(p.88)を参照してください。



インスタンスのスタイル名を他と重複しないように入力してください。重複があると出力時に片方が上書き削除されます。続いてウエイト、グリフ幅、カスタムの軸の数値を決定してください。これらの数字はマスタータブで入力した各軸の値と関係しています。

12.4 互換性の修正

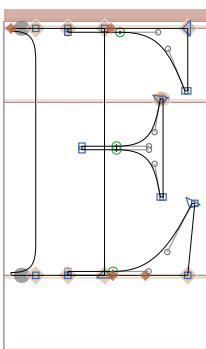
Glyphsのマルチプルマスターは、自動でアウトラインの互換性を確保しようとはしません。これは手動により設定・修正する必要があります。非互換のグリフはフォントビューの各グリフの左上に赤い三角形で表示され、編集ビューでは上に赤いストライプで表示されます。

この例では大文字のEがフォントビューで非互換と表示されています。グリフ左上にある赤い3角形がそれを示しています。

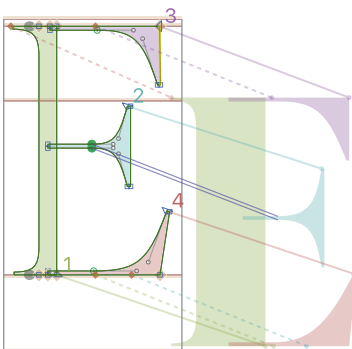


フォント制作中はパスを非互換のまま編集できますが、最終的に補間をする際には互換性のある状態にしないといけません。パスの方向、ポイントの数と構造が、どのマスターでも完全に一致していなければならないということです。

編集ビュー内では、非互換のグリフでは上部に赤いストライプがグリフ幅いっぱいに表示されます。

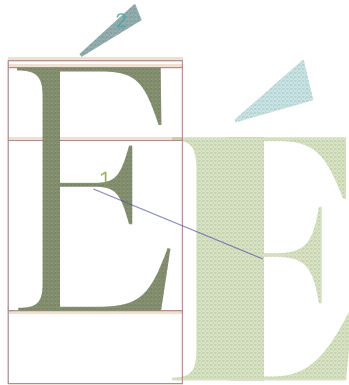


「表示 > マスター互換性を表示」(control+option+command+N) を選ぶと、各マスターに互換性があるかどうかを編集ビューで確認できます。各マスターのアウトラインが斜めに並び、閉じたパスにはレイヤー内の順番に合わせて色と数字が付けられます。また、パスの開始点は他のマスターの対応する開始点と相互に直線で結ばれます。さらに、アンカーの対応関係は点線で表示されます。もし対応するパスやアンカーが他のマスターにない場合は、これらの線が原点に向かって伸びていきます。なお、選択中のポイントは他のマスターと青い直線で結ばれて表示されます。



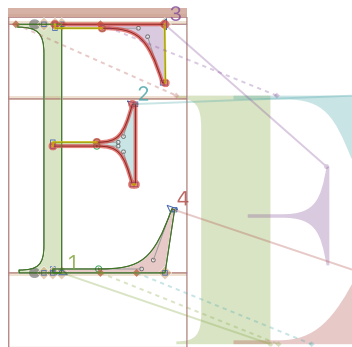
コンポーネントは市松模様のパターンになり、レイヤー内の順番に合わせて色と数字が付けられます。コンポーネントを選択すると、対応するコンポーネントの中心同士が青い線で結ばれます。コンポーネントが足りない

場合、青い線は原点に向かって伸びていきます。



アウトラインの各セグメントは他のマスターと比較され、その対応状態によって色が付けられます。赤い線は補間不可能という意味です。これは多くの場合、曲線セグメントと直線セグメントが対応させられていることが原因です。緑の線は問題なく補間可能な部分です。黄色は補間可能ではありますが、セグメントの角度が大きく変わるため注意が必要かもしれないという意味です（例：垂直から斜めに変化、右向きの斜めから左向きに変化するなど）。アウトラインの他の部分がすべて緑色であれば、黄色の線は特に問題ないケースも少なくありません。しかし黄色の線は開始点の位置に問題がある場合もあり、思わしくない補間結果が生じるかもしれません。これは特に長方形のパスによく起こりますので注意して確認してください。

左のステムは緑色で表示され補間には問題ありませんが、上段と中段のセリフはマスター間で順番が入れ替わってしまっており、セグメントが赤と黄色で表示されています。



補間生成で使用する各マスターは、すべてのパス、ポイント、コンポーネント、アンカーが互換していなければいけません。逆に言えば、補間に関わらないマスターの互換性は気にする必要がありませんし Glyphs も警告を出しません。

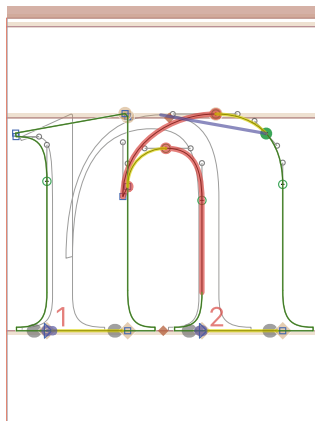
もしパスが互換していない場合、まずは「パス > パス方向を修正」(shift + command + R)、または option を押しながら「パス > 全マスターのパス方向を修正」(option + shift + command + R) を実行してみるとよいでしょう。Glyphs はパスの順番を整理し、開始点を設定し直し、パス方向を修正しようと試みます。もしポイントの数がすべてのマスターで同じでない場合、ペンツール (P) でポイントをパス上に追加したり、選択ツール (V) でポイントを選択して削除したりします。

コンポーネントグリフを標準レシピから再生成したい場合は「グリフ > コンポーネントグリフを生成」(option+shift+command+C) を実行します。もし標準以外のコンポーネントを使っている場合は、コンポーネントをすべて選択し他のマスターにコピー&ペーストします。詳細は「8.1 コンポーネント」(p.97) を参照してください。

パスやコンポーネントの順番を変更するには、「フィルタ > マスター互換性を手動修正」を実行し、表示されるウインドウ内でドラッグして並び替えてください。詳細は「5.2.1 マスター互換性を手動修正」(p.54) を参照してください。開始点を手動で設定するには任意のポイント上でコンテキストメニューを表示して「開始点の設定」を選択します。

アンカーの不一致が起きている場合は、「グリフ > アンカーを自動設置」(command+U) を実行し、標準で設置されるべきアンカーをすべて設置してみましょう。optionを押しながら「グリフ > アンカーをリセット」(option+command+U) を実行すると、現在置かれているアンカーが全削除され、標準で設置されるべきアンカーが設置されます。

マスター互換性は他の表示方法でも確認できます。「環境設定... > ユーザー設定」の「マスター互換性をずらし表示」のチェックをOFFにすると表示方法が変わります。このモードではパスとコンポーネントは色付けされませんが、数字は開始点の位置に表示されます。さらに各マスターの位置が斜めに並ばず元の位置に揃うので、デザインの細かな差を比較しやすくなります。



12.5 マスターレイヤーの比較

マルチプルマスター環境では、各マスターレイヤーの表示を頻繁に切り替えて比較することになります。その際、ツールバーにあるマスターのアイコンをクリックするか、command+1、command+2...のようにマスターの番号に対応するショートカットを利用すると表示の切り替えが素早くできます。この操作では全グリフの表示が切り替わります。

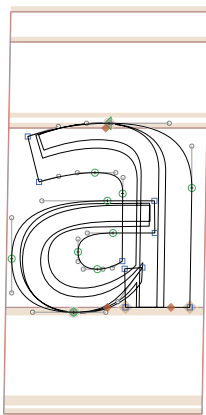


「ウインドウ > パレット」(option+command+P) で表示されるパレットのレイヤーリストでは、テキストツールなどで選択したグリフの全レイヤーが表示されます。マスターレイヤーは太字で表示されます。任意のレイヤーをマウスクリックでリスト選択すると、該当グリフはそのレイヤーで表示されます。複数のグリフが選択された状態でマスターレイヤーをマウスクリックすると、そのマスターで表示されます。これは各マスターを隣同士に並べて比較したいときに便利です。



すべてのマスターレイヤーを隣同士に並べたい場合は、「編集 > すべてのマスターを表示」を実行します。このモードではブレースレイヤーやブラケットレイヤーを表示しません。

レイヤーリストの目玉アイコンをクリックすると、そのレイヤーのアウトラインが灰色で表示されます。



12.6 複数ファイル間でのファミリーの一貫性の確保

フォントファミリーを複数の Glyphs ファイルまで拡充するときは (例: ローマンのマルチプルマスターとイタリックのマルチプルマスター)、フォント情報の「フォント」タブにあるファミリー名を完全に一致させてください。この箇所のファミリー名を変更したくない場合、カスタムパラメータ「familyName」を設定すれば、フォント書き出し時にファミリー名を変更できます。この設定はインスタンス単位でも可能です。詳細は「17.3 カスタムパラメータ」(p.184) を参照してください。イタリック体はつねに対応するウエイトのローマン (正立体) と「インスタンス」タブでスタ

イルリンクしておかなければいけません。ポールドイタリックはこの限りではありません。詳細は「7.3.4 スタイルリンク」(p.90)を参照してください。

ファイル間でグリフの状況とメトリクスを確認するには、該当する複数のファイルを開いて、「編集 > フォントを比較...」を選んでください。グリフ数、互いに不足しているグリフ、左右サイドベアリング、そしてグリフ幅をマスター別にまとめた表が表示されます。不足しているグリフを補いたい場合は表からグリフ名をコピーし、メニューバー「フォント > グリフを追加...」で表示されるダイアログにペーストしてください。

12.7 ブレースレイヤー

レイヤー名にブレース（波括弧 { }）が含まれていて、その括弧内にマルチプルマスターの補間値が記されている場合、このレイヤーは「ブレースレイヤー」と呼ばれ特殊な扱いを受けます。ブレースレイヤー内にあるパス、コンポーネント、アンカーの内容はそのグリフの中間マスターとして機能します。レイヤー名のブレース内にはウエイト軸を使った1つの数字だけでなく、字幅やカスタム軸まで使った3つの数字まで入力できます（この際、数字はコンマ区切りで入力してください）。ブレースレイヤーは補間の途中のデザインを微調整するのに便利で、特に e や s など構造が複雑なグリフに効果を発揮します。

上：ブレースレイヤーなし。

下：中間にブレースレイヤーあり。
クロスバーの太さの変化に注目してください。上の行のサンプルでは直線的な補間のため、中間のウエイトでクロスバーの太さが足りません。このような場合にクロスバーの太さが調整されたブレースレイヤーが便利です。



ブレースレイヤーを追加するには、パレットのレイヤーのリストからマスターレイヤーを複製します。デザインスペースの軸がウエイトのみの場合は例えば {100} と入力し、それ以外にも軸がある場合は {100, 90}、{100, 90, 80} などと入力してください。括弧内のそれぞれの数字はマルチプルマスターのウエイト、字幅、カスタム軸の補間値に対応しています。名前を正しく設定した後は ⚙ ボタンメニューから「再補間」を選びます。次に補間して得られたブレースレイヤーのアウトラインを編集します。ウィンドウ下のプレビューでインスタンスを表示させている場合、編集した結果はすぐに反映されます。

12.8 ブラケットレイヤー

ブラケットレイヤーとはレイヤーの名前がマスター名で始まり、スペース

とブラケット（角括弧 []）が続き、その括弧内に数字が1つだけ記されているレイヤーのことで、ブレースレイヤーと同様に特殊な挙動をします。ブラケットレイヤーは、括弧内に記された補間値からマスターと入れ替わります。例えばBold [130]という名前のレイヤーがあると、ウエイトの補間値が130以上の場合はそのレイヤーがマスターとして機能するようになります。ブラケットレイヤーによるデザイン変更は一度とは限らず、複数レイヤーを与えて複数回変更することもできます。複数のマスターを同時に切り替えることもできます（例：Light [100]とBold [100]）。補間軸が2つあるマスターでは、コンマ区切りで2つ目の軸の補間値を入力できます（例：Bold [100,300]レイヤーはウエイト値が100、字幅値が300を超えたときに使用されます。）

ブラケットレイヤーは細いウエイトと太いウエイトで全く構造の異なるデザインを両立させたいときに便利な機能です。これの一般的な例はドル記号の縦棒です。ウエイトが太くなるほど縦棒を収めるのが難しくなってくるため、多くのデザイナーは太い方のウエイトで縦棒の貫通をやめて上下に付けるだけのデザインに切り替えます。この場合、貫通している方のデザインをLightとBoldの両方のマスターに格納し、途切れている方のデザインをそれぞれLight [100]、Bold [100]のようにレイヤー名を設定して用意します。



また他の方法として、独立したオルタネートのグリフを用意し、フォント出力時に入れ替えることもできます（例：dollarとdollar.bold）。オルタネートグリフをインスタンスのRename Glyphsカスタムパラメータで名前を上書き変更し、他のインスタンスではRemove Glyphsカスタムパラメータを使ってオルタネートの方を削除します。例えばボールド用にのみ使うdollarのデザインdollar.boldを、ボールドのインスタンスにRename Glyphsでdollarに改名します。他のインスタンスではRemove Glyphsでdollar.boldをフォントから削除するように設定します。この方法だとすべてのデザイン変更がグリフとしてフォントビューに現れて見落としを防げるため、こちらを好むデザイナーもいます。

12.9 開いたブラケットレイヤー

角括弧の向きがどちらも同じになっているブラケットレイヤー（例：Bold [100]）はまた別の種類のブラケットレイヤーです。これは通常のブラケットレイヤーととても似た働きをしますが、影響範囲が入れ替わり、括弧内

の補間値「までは」そのレイヤーがマスターとして使用されます。このレイヤーの利点は、フォントビューで常に正しいマスターが表示できることです（Lightでは縦棒の繋がったバージョンのdollar、Boldでは途切れたバージョンが表示できる）。

13 カラーフォント

Glyphsはカラーフォントの効率的な編集、プレビュー、出力をサポートしています。対応しているカラーフォントは「伝統的な手法の複数レイヤーフォント」「Microsoft方式のCPALとCOLRテーブルを使ったフォント」「Apple方式のsbixテーブルを使ったフォント」「Mozilla/Adobe方式のSVGテーブルを使ったフォント」の4種類です。どの方式でも、従属レイヤーにグラフィック要素を配置する点で共通しています。

13.1 複数レイヤーでの作業

13.1.1 全レイヤー選択ツール

複数レイヤーのベクトルデータを編集するには、全レイヤー選択ツールが便利です。これは選択ツール（V）と同様の働きをしますが、表示中のすべてのレイヤーを編集できる点が違います。全レイヤー選択ツールと通常の実選択ツールは選択ツールボタンを押したままにしてポップアップメニューを表示するか、shift+Vキーで切り替えられます。

13.1.2 メトリクスの同期

カラーフォントでは、様々な内容の異なるすべてのレイヤー間でグリフ幅だけでなくカーニングも統一されてなければいけません。すべてのマスターとカラーレイヤーを最初のマスターのメトリクスと同期させるには、フォント情報の「マスター」タブでLink Metrics With First Masterカスタムパラメータを設定してください。

13.1.3 カラーフォントの出力

すべてのカラーフォントは普通のOpenTypeフォントです。これらはGlyphsが出力するどの形式のOpenTypeフォントでも機能します（OTF/ CFF、TTF、WOFF、WOFF2、EOT）。しかし、カラーフォントが表示されるのは、それに対応している環境のみです。例えば、複数レイヤーフォントは、DTPソフトのように複数のテキストを重ねて配置可能なアプリケーションでしか使用できません。そのためこの形式はテキストエディットやMicrosoft Wordなどで利用しても、望むような表示はかなり困難だったり、まったく不可能だったりします。

13.2 複数レイヤーフォント

複数レイヤーのカラーフォントは、単純にレイヤーごとに別々に出力した複数のフォントで構成されていて、各フォントをアプリケーション上でぴったり重ね、文字を着色して使うものです。そのため、使用を想定するアプリケーションはテキストの重ね表示に対応している必要があります。

13.2.1 マスターのカスタム名とプレビュー色

まずは必要な色数の分だけ、「ファイル > フォント情報」でマスターを追加してください。マスターの名前は、カスタム欄のテキストフィールドに文字入力することで任意に設定できます。また、カスタム欄の数値については、各マスターに固有の値を与える必要があります。例えば最初のマスターのカスタム名がFrontで値が1、次がSideで値が2といった具合です。

マスターにプレビュー色を設定するには、各マスターのカスタムパラメータにMaster Colorを追加します。値の箇所をクリックするとカラーピッカーが表示されるので、そこで色を設定してください。この色はGlyphs内のプレビュー色のためだけに使われ、出力されるフォントには一切影響を与えません。

13.2.2 すべてのマスター用のインスタンス

各色のレイヤーをフォントファイルとして出力するには、フォント情報の「インスタンス」タブで左下の+ボタンをクリックし、「全マスターをインスタンスとして追加」を選択してください。Glyphsはマスターと同名で同じ補間値のインスタンスを追加します。

13.2.3 複数レイヤーフォントのプレビューと使用

通常のマルチプルマスターファイルと同様、各マスターへの切り替えはアイコンまたはキーボードショートカットで行います。複数のマスターを編集ビューで同時にプレビューするには、テキストツール (T) に切り替えるかスペースバーを一時的に押したままにしてください。パレットのレイヤーのセクション (の目玉アイコン) で非表示にされていないければ、各マスターのオブジェクトはMaster Colorカスタムパラメータで設定された色で表示されます。レイヤーの表示状態の切り替えについての詳細は「4.4 レイヤー」(p.49) を参照してください。

複数レイヤーフォントのユーザーはAdobe Illustratorなどを利用するであろうことに留意してください。Illustratorでは、テキストボックスからのベースラインの距離はフォント内の小文字のdの高さで決まります。別の言い方をすると、小文字のdの最高点がテキストボックスの上端に整列するようになっています。すべてのカラーレイヤーの小文字のdが完全に統一されるというのはデザイン上可能性が低いため、Illustratorでは複数のレイヤーの位置を合わせるのがとても困難になるでしょう。この問題を回避するには、Illustratorのユーザーに「書式 > エリア内文字オプション... > オフセット」で「1行目のベースライン」を「仮想ボディの高さ」または「固定」に切り替えるようお願いするとよいでしょう。またすべてのレイヤーの小文字dの同じ位置 (他のすべてのパスよりも高い位置) に微小なサイズのパス (1×1ユニットの正方形など) を描く方法もあります。

複数レイヤーフォントを Adobe Illustrator (欧米版) の初期状態のベースラインオフセットで表示した場合 (上) と、「仮想ボディの高さ」に変更した場合 (下)。



13.3 Microsoft カラーフォント

Microsoft 方式のカラーフォントは、CPAL (color palette) と COLR (color) の2つのテーブルが追加されたフォントです。CPAL テーブルはフォントで使用される色のリストで、COLR テーブルはどのアウトラインにどの色が適用されるのかを記述します。このテーブルを利用できるのは Windows 8.1以降で、さらにアプリケーション側でも対応している必要があります。

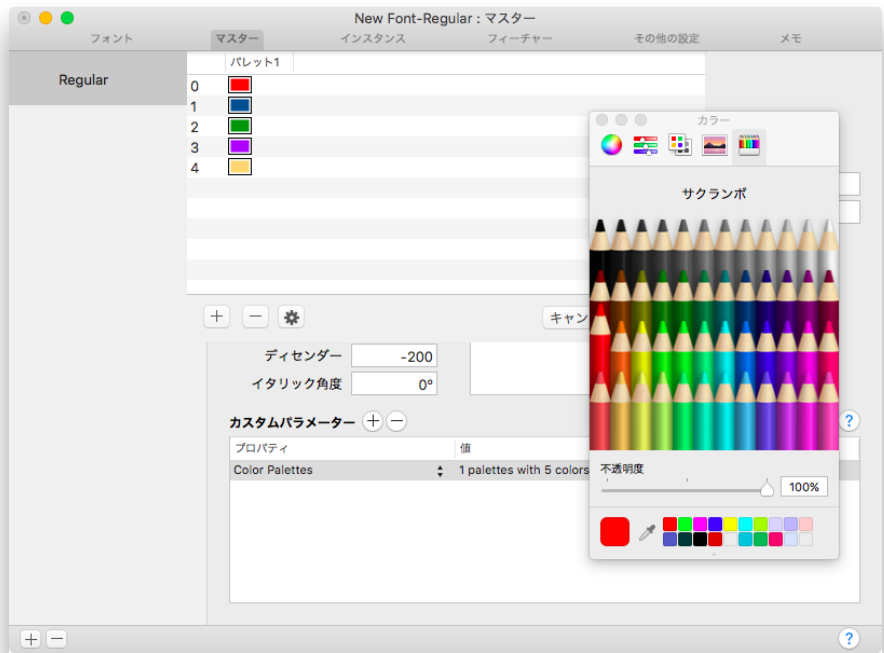
13.3.1 CPAL テーブル

カラーパレットを設定するには、フォント情報の「マスター」タブで Color Palettes カスタムパラメータを追加します。値の箇所をクリックしてカラーパレット設定ダイアログを呼び出してください。1行につき1色、1列につき1パレットが設定できます。表の左端の列には0から始まるカラーインデックスが表示されています。

ダイアログ内では + ボタンをクリックして新しい色を追加、- ボタンで色を削除します。色を編集するには色をクリックして選択し、もう一度クリックするか、またはダブルクリックします。するとシステム標準のカラーピッカーが現れますので、任意の色を選んでください。

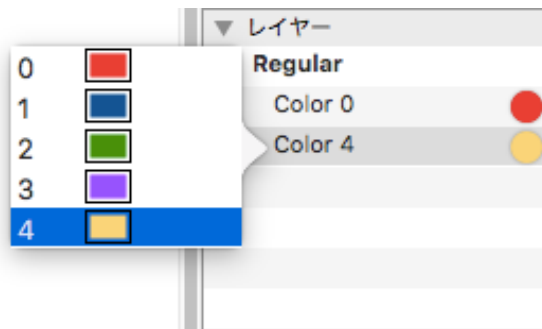
✳ ボタンではパレットを追加または削除できます。追加のパレットは暗い背景用など、カラーバリエーションを追加したいときに便利になるかもしれませんが、しかし Glyphs は最初のパレットしかプレビューできませんし、この間ハンドブックの執筆段階では2つ目以降のパレットにアクセス可能なソフトウェアの存在は知られていません。

色の設定が終わったら OK ボタンを押して決定します。するとパレットがフォントに追加されます。



13.3.2 カラーレイヤーの作成

カラーパレットで決定した色を適用するには、「Color 数字」の書式で従属レイヤーを追加します（例：3色目のレイヤーはColor 2）。このように命名されたレイヤーには右にカラーマーカーが現れます。従属レイヤーとカラーパレットの対応を変更するには、レイヤー名を入力し直すか、カラーマーカーをクリックして表示される一覧から選択します。



13.4 Apple カラーフォント

Appleのカラーフォントは複数の異なるサイズのビットマップ画像を保管するsbixというテーブルを採用しています。sbixテーブルは現在OS X 10.7以降またはiOSでのみサポートされています。sbixの機能のなかにはOSX 10.9以降またはiOS7以降でのみ動作するものもあります。現在、sbixテーブルを抱えたフォントはAdobeアプリケーションやWindowsでは表示できません。sbixについての技術的な詳細は以下のページで参照できます：

developer.apple.com/fonts/TrueType-Reference-Manual/RM06/Chap6sbix.html

13.4.1 ビットマップ画像ファイル

Glyphsは画像編集ソフトではないため、画像ファイル自体はプレビュー、Pixelmator、またはPhotoshopなど他のアプリケーションで用意してください。またGlyphsファイル内には相対パスのみが保存されることに注意してください。そのため画像ファイルはGlyphsファイルとなるべく近い階層に保存することをお勧めします。またリンク切れを防ぐため、Glyphsファイルを移動する際は必ず画像ファイルまたはフォルダも同時に移動するようにしてください。

画像ファイル形式は、PNG、JPEG、TIFFがサポートされています。PNGの透過もサポートされています。仕様によればPDFもサポートされているはずですが、このハンドブックの執筆時点ではPDF入りのカラーフォントを実際に表示できるアプリケーションの存在は知られていません。

画像ファイルは1つだけでも可能ですが、フォントサイズごとに最適化した複数の画像ファイルを扱うこともできます（例：24ピクセル用、48ピクセル用、512ピクセル用）。この場合は、フォント内のすべてのグリフに対して各サイズ用の画像をすべて用意してください。このフォントを使用できるアプリケーションは、文字サイズに合わせて自動的に適切な画像を表示します。

13.4.2 画像サイズごとのレイヤー

画像ファイルをグリフに配置するには、従属レイヤーを追加し、レイヤー名を「iColor サイズの数字」にしてください（例：48ピクセル用画像のレイヤー名はiColor 48、512ピクセル用はiColor 512）。そして、各レイヤーにFinderから画像をドラッグ&ドロップします。また、メニューバー「グリフ > 画像を追加」で表示されるダイアログでも配置できます。

画像が配置されればiColorレイヤーは完成です。画像のグリフ内での位置や拡大率は一切無視されますので、原点の近くあたりにでも置いておけば十分です。画像が表示される際には、目標のピクセルサイズに合わせて拡大縮小され、画像の位置はベースラインから15%ほど下に移動されます。グリフ幅は画像サイズではなく、グリフに設定した数値の方が使われます。マスターレイヤーには通常のアウトラインを格納できます。アウトラインは通常、拡大縮小後の画像の上に重ねて表示されます。

sbix用の画像を含むグリフは、必ずiColorレイヤーがすべて一致するようにしてください。例えば、iColor 64、iColor 256、iColor 512の3つのレイヤーをあるグリフで使用することにしたなら、他のカラーグリフもすべて同じ数のiColorレイヤーで、数字も同じになるようにしなければいけません（そして画像も配置されている必要があります）。

13.4.3 Apple カラーフォントの使用

sbixテーブルはOpenTypeフォントならどの種類にも（OTF、TTF、WOFF、

WOFF2、EOT) 格納できる仕様なのですが、現時点ではMac OSX 10.10以降のSafariとSafari上でサポートされているフォント (OTF、TTF、WOFF) でのみ使用できます。それ以前のSafariでは、マシンにインストールされたOTFまたはTTFフォントでしか使えません。ただしインストールさえされていれば、テキストエディットなどブラウザ以外のCore Textエンジンを使用したアプリケーションでも使用できます。

13.5 SVGカラーフォント

SVGカラーフォントはSVG形式のWebフォント (拡張子が.svgまたは.svgzのフォント) とは別物です。後者はGlyphsではサポートしていません。SVGカラーフォントは、SVGテーブルにSVGを埋め込んだOpenTypeフォントであり、OpenType内のSVGテーブルを使用することでその機能を実現しています。Mozillaが標準のカラーフォント規格として提唱し、Adobeがサポートを表明しているため、「SVG内蔵OpenType」や「Mozilla/Adobe式カラーフォント」と呼ばれることもあります。

仕様はこちらで参照してください：
[w3.org/community/
svgopentype/](http://w3.org/community/svgopentype/)

SVGカラーフォントが他の方式と違うのはアニメーションをサポートしている点です。現時点ではFirefox 26以降のバージョンでのみSVGカラーフォントがサポートされています。

13.5.1 SVGファイル

SVGファイルは別のアプリケーションで作成してください。Glyphsファイルに配置すると、SVG内の1ピクセルまたは1ポイントが1ユニットに相当するよう自動的に拡大縮小されます。SVGファイルの寸法は、エラー回避とフォントファイルのサイズ抑制ため、適切なものにしておきましょう。配置後も寸法はUPM (通常1000) やフォントのデザインとして決めた高さ (例: キャップハイト) を超えないようにしてください。

13.5.2 SVGレイヤー

SVGファイルをGlyphsファイルに配置するには、従属レイヤーを追加し、レイヤー名を「svg」にしてください。そのレイヤーにFinderからSVGファイルをドラッグ&ドロップします。また、メニューバー「グリフ > 画像を追加」で表示されるダイアログでも配置することが可能です。配置するとユニット単位に合わせて自動的に拡大縮小されます。サイズを調整するには、情報パネルで数値を変更するか、バウンディングボックスをドラッグします。位置の移動も可能です。

13.5.3 SVGカラーフォントの使用

「svg」レイヤーを含むグリフができたら (数は不問)、なるべくWOFFまたはFirefox用にWOFF2形式のOpenTypeを出力してください。Glyphsは配置されたSVGファイルに合わせたSVGテーブルをフォントに追加し

ます。移動や拡大縮小がされたSVGには「transform」属性が追加され、そこに変形情報が格納されます。

アニメーションSVGも格納できますが、たとえ単純なアニメーションであってもCPUに非常に大きな負荷をかけますので注意してください。

14 エラー対処

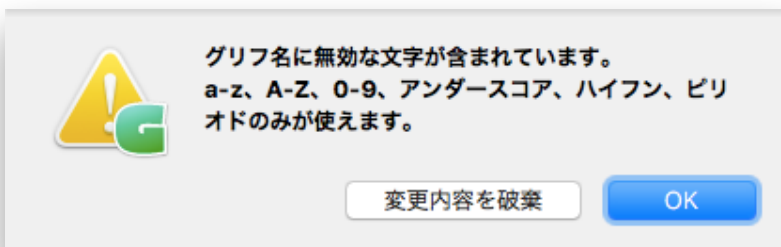
14.1 グリフ名

フォント出力時に起きるエラーの原因で最も多いのがグリフ名の不備です。すべてのグリフ名が以下の条件を満たしていることを確認してください。

- ・A～Z、a～z、数字、アンダースコア (_)、ハイフン (-)、ピリオド (.) のみを使っている。
- ・文字で始まる (A～Z、a～z)。ただし .notdef と .null だけは例外。スマートコンポーネントなどアンダースコアで始まるグリフは出力時に含まれない限りは問題ない。
- ・空白文字 (スペース、タブ、改行) を使っていない。特にグリフ名の末尾にある場合は見落としやすいので注意。
- ・ASCII文字以外 (á è ß ü や平仮名、漢字など) を使っていない。

これらを満たさない場合はフォント出力時に、グリフ名に関するエラーメッセージが、その詳細とともに表示されます (例: グリフ名にスペースは使えません)。このようなエラーメッセージはフォント情報の「フィーチャー」タブでOpenTypeフィーチャーのコードをコンパイルするときにも表示されます。

スペースなど禁止されている文字を使っているグリフを探したい場合は、フォントビュー右下の検索フィールド (command+F) でそれらの文字を検索してください。



14.2 フォント名

エラー内容がPOSIXパス (例: /Users/ユーザー名/Library/Application Support/Glyphs/Temp/フォント名/FontMenuNameDB) で記述された「FontMenuNameDB」というファイルに言及している場合、フォント名やスタイル名に問題があるのかもしれませんが。ここではスペースを使えますが、非ASCII文字は使えません。

ASCII文字以外を使う場合は、localizedFontNameカスタムパラメータを設定してください。詳細は「17.3 カスタムパラメータ」(p.184) を参照してください。スタイル名に非ASCII文字を使う場合はlocalizedStyleNameカスタムパラメータを設定します。

14.3 Unicode 値の重複

既存のフォントには間違っても同じUnicode値を持った複数のグリフが存在する場合があります。このようにダブルエンコーディングされたフォントを読み込む際、Glyphsは警告を出します。すべてのグリフを選択して「フォント > グリフ情報を更新」を選ぶと多くの場合は解消できます。

Tip: もし二つの異なるコードポイントから一つのグリフにアクセスしたい場合は、単純に二つグリフを用意したほうがベターです。

また逆に、単体のグリフが2つのUnicode値を持っているフォントを読み込む場合は、Glyphsはグリフ名から判別して自動的にこれを解消します。この問題は特にギリシャ文字の大文字デルタ Δ とオメガ Ω によく起こります。 Δ はU+3094（グリフ名Delta）であるべきですが、数学のラプラス作用素の記号U+2206（グリフ名increment）になっていることがあります。 Ω はU+03A9（グリフ名Omega）であるべきですが、オームの記号U+2126（グリフ名Ohm）が割り当てられている場合があります。

14.4 OpenTypeのフィーチャーコード

フォント書き出し時にOpenTypeフィーチャー関連のエラーが発生した場合、もし作業中のフォントのOpenTypeフィーチャーが完全に自動生成機能のみで作られていれば、フォント情報の「フィーチャー」タブの「自動」と「コンパイル」ボタンをクリックすることで再生成して解消することができます。

GlyphsはFDKでのコンパイルで起きたエラーをできるだけエラーダイアログとして表示します。また具体的にどのフィーチャーのどの行で問題が起きたのかもできるだけ報告します。この場合、エラーダイアログには「表示」ボタンが現れます。これをクリックすると、Glyphsは即座に問題のある行に案内します。例えば以下のようなエラーが起きる可能性があります：

「Contextual substitution clause must have a replacement rule or direct lookup reference.」この場合、おそらくグリフ置き換え(GSUB)系のフィーチャーでbyを書くのを忘れたかスペルミスしたのだと考えられます。

「DFLT script tag may be used only with the dflt language tag.」これは文字タグ (script tag) を先に書くのを忘れて言語タグ (language tag) を書いてしまったのでしょう。例えば、「language DEU;」と書く前に「script latn;」と書いておく必要があります（ドイツ語であることを宣言する前にラテンアルファベットであることを宣言しなければいけません。1つの言語が違う文字で書かれることもあるからです）。

「“Feature” statement allowed only in ‘aalt’ feature.」これはおそらくフィーチャーコードをプレフィックスの部分に書き、それを適切に閉じるのを忘れてに違いありません。

「Glyph xxxx not in font.」存在しないグリフをフィーチャー内で使った場合に表示され。エラーメッセージではどのグリフが存在しないの

かを示されています。おそらくグリフ名を間違えたか、フィーチャー編集時に存在していたグリフを消したか改名した、もしくは始めからそのようなグリフを作っていなかったのでしょう。自動生成されたフィーチャーが最新でないことも原因になり得ます。このような場合、フォント情報の「フィーチャー」タブのコンパイルボタンの左隣にある「自動」ボタンをクリックすることで解決できます。

異なる文字体系同士のカーニングペア（例：欧文とキリル文字のペア）は、このエラーを回避するためフォント出力時に無視されます。

「GPOS feature ‘kern’ causes overflow of offset to a subtable.」GPOSのkernフィーチャーが複雑すぎてグリフのポジショニングのテーブルが膨大になるとこのエラーが表示されます。OpenTypeのテーブルの最大容量は64KBです。カーニングをクリーンアップしたりグループ化すると解消するかもしれません。「ウインドウ > カーニング」で表示されるカーニングペアで設定を確認してください。また5ユニット以下のカーニングはほとんど意味がないため削除してよいでしょう。異なる文字体系同士のカーニングも不要ですので、例えば欧文とキリル文字のカーニングペアなどがあれば削除してください。他にはUse Extension Kerningカスタムパラメータを使って拡張カーニングを使い、上限を無視するという方法もあります。「17.3 カスタムパラメータ」(p.184)には各カスタムパラメータの詳細や値のサンプルが載っています。

「Invalid token.」クラス名がスペースで始まっているか、またはフィーチャー内でクラスを使っている際に@とクラス名との間にスペースが入っていると考えられます。またはフィーチャーのどこかに非ASCII文字が使われている可能性があります。フィーチャーのコメント欄にも使用文字制限は当てはまります。

「Lookup type different from previous rules in this lookup block.」kernなどグリフ位置調整（GPOS）系のフィーチャーで前後関係に依存するものとそうでないものを混ぜたのだと考えられます。

「MakeOTF error. GPOS feature “xxxx” causes overflow of offset to a subtable.」挙げられたフィーチャーの位置調整ルールが多すぎます。もしこれがkernフィーチャーの場合、カーニングが多すぎるということです。カーニンググループより多く使用し、ペアをグループ化し、例外を減らし、また最終手段としてUse Extension Kerningカスタムパラメータを使うことで解決を図りましょう。

「MakeOTF error. GSUB feature “xxxx” causes overflow of offset to a subtable.」挙げられたフィーチャーの置き換えルールが多すぎます。ルックアップを使用し、繰り返しになっている置き換えルールを極力まとめることでコードを短くしましょう。

「makeotfGlyphs.」最も可能性の高い原因はグリフ名に使用禁止文字が使われているを使うことです。多くの場合、ダイアログ内に詳細な情報が表示されるはずですが。

「makeotfGlyphs [FATAL] line too long.」おそらくグリフ名が長

すぎるのが原因です。グリフ名に使える字数の上限は122文字です。

「not in range -32767 .. 32767 (text was "...").」少なくとも1個のポイント、ハンドル、またはアンカーが範囲外にあります。これらの座標はすべて x、y ともに ±32767 以内に収まってなければいけません。Glyphs のファイルをテキストエディターで開き、「e+」という文字列を検索してください。こうすることで指数で表記された部分（異常に大きな数値）、例えば「-9.22337e+18」などを探し当てて、どのグリフに修正が必要なのかを確認できます。もしこれで何も見つからない場合は、エラーメッセージのブラケット内に表示されている数字で検索してみてください。

「Positioning values are allowed only in the marked glyph sequence, or after the final glyph node when only one glyph node is marked.」kern フィーチャーの前後関係依存カーニングの構文は他と違うため細心の注意が必要です。少なくとも1つのグリフが垂直のマヌケ引用符 (') でマークされ、そのグリフの次に数値が来る必要があります。他のポジショニングの構文のように数字が最後に来るではありません。

「Premature end of input.」フィーチャーで使用される記号がグリフ名に使われているために混乱が起きていると考えられます。フォント内に等号 (=)、アットマーク (@)、ブラケット ([]) を使ったグリフがないか探し、見つかった場合には改名してください。

「Target glyph class in rule doesn't have the same number of elements as the replacement class.」クラス対クラスの置き換えで、元と先のクラスのグリフ数が一致していないというエラーです。このエラーダイアログは問題のある行に直接案内します。置き換えに使われるクラスのグリフ数と順番を確実に一致させてください。

「Syntax error.」フィーチャーのコードがAFDKOの構文に従っていないエラーです。これにはいくつかの原因があります。

- ・非ASCII文字をコードに使用している。
- ・フィーチャー名がルールに準拠していない。4文字のASCIIの小文字と数字のみで、スペースやその他記号は一切使ってはいけません。
- ・フィーチャーの命令 (sub、pos、by) などを打ち間違えた。
- ・コードの各行の終わりのセミコロンを忘れた。この場合、エラーメッセージには「missing ";"」とも書かれています。
- ・ポジショニング関連のフィーチャーに数値を入れ忘れた。この場合、エラーメッセージには「missing NUM」とも書かれています。

14.5 アウトライン

フォント書き出し時にエラーが表示されなくても、アウトラインが欠損していたり、アウトラインが消えて空白のグリフになってしまったりする場合があります。

14.5.1 開いたパス

出力するときに「重なったパスを合体」オプションが入っている場合、アウトラインが一周して閉じていないものは削除され、出力されたフォントには含まれません。もしフォントファイルの中に空白のグリフがある場合は、正しくパスが閉じられているかを確認しましょう。

開いたパスをそのまま出力したい場合（カスタムパラメータでフォント出力時に自動的にパスをオフセットする時など）は、「重なったパスを合体」オプションを無効にしておきましょう。

14.5.2 不正なパスの方向

アウトラインは必ず反時計回りに描かれていなければいけません。ただしカウンター、つまり抜きの形のアウトラインは逆方向の時計回りにする必要があります。この方向が正確でないと、パスが閉じていても正常に表示されません。アウトラインの方向を逆転させるには、パスを選択して「パス > パス方向を逆転」を選びます。すべてのパスを逆転させるにはコンテキストメニュー「すべてのパス方向を逆転」を選択します。修正を Glyphs に任せるには、グリフを選択して「グリフ > パス方向を修正」(shift + command + R)を選びます。フォント全体に適用するには、option を押しながら「グリフ > 全マスターのパス方向を修正」(option + shift + command + R)を選びます。

14.5.3 重なり合った同一のパス

形が同じで方向の違うパスが重なり合っていると、出力時に互いを消去しあう可能性があります。パスをダブルクリックで選択して backspace または delete キーで消去してください。何も変化がないように見えるかもしれませんが、それは全く同じアウトラインが2つ重なっていたということです。問題は解決されています。

14.5.4 アウトラインの非互換

アウトラインの互換性が取れていなくてもフォントの出力が中止することはありません。非互換のグリフは単純に空白となって出力されます。「以下のグリフは非互換であり、アウトラインは含まれなくなります」というエラーが出ても、フォントはとりあえず出力されます。非互換のアウトラインの修正方法については「12.4 互換性の修正」(p.141)を参照してください。

15 読み込みと書き出し

15.1 ベクトルベースの描画アプリケーション

15.1.1 Adobe Illustrator

Tip: Illustratorで正しいサイズにするには、Glyphs側でボディの高さに合わせた長方形を描き、それをIllustratorにコピー＆ペーストして、その長方形の高さに収まるように拡大縮小するのが簡単です。

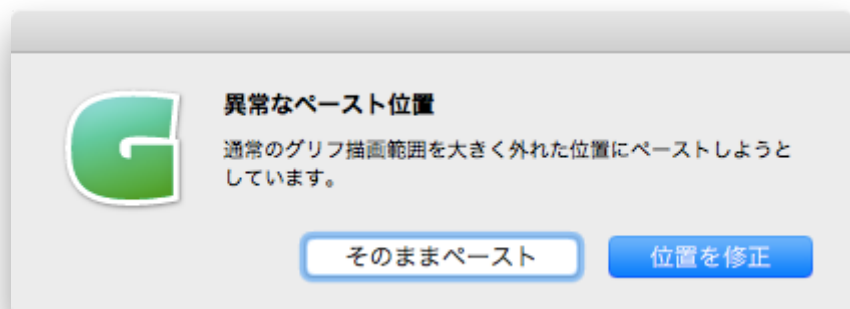
Illustratorでの1ptはGlyphsでの1ユニットに相当します。つまりIllustratorにある100pt四方の正方形をGlyphsにペーストすると100ユニット四方になります。またはIllustratorのアートボードの高さを書体のUPMに合わせて設定しましょう（1000UPMの場合はアートボードの高さを1000ptとします）。アセンダーとディセンダーの両方がアートボード内に収まるよう注意してください。ページの原点をベースライン上の左端に設定すると、Glyphsにアウトラインをペーストした時に正しい位置に配置されます。Illustrator CS4までのバージョンでは、定規の左上の角をドラッグして現れる照準を任意の位置にドロップすると、そこがページの原点になります。CS5以降はアートボードオプションで原点を設定できます。

15.1.2 Sketch

Sketchからアウトラインを読み込む場合は、アウトラインの線幅を1ptに設定し、塗りは解除してください。こうすることでGlyphsで二重のアウトラインの発生を回避できます。

15.1.3 パスのコピーペースト

上記のアプリケーションでコピーしたアウトラインをペーストする際は、コピー元のアウトラインの座標位置によって、グリフ内に収まらない可能性がある旨のダイアログが現れることがあります。「位置を修正」ボタンをクリックすると、ペーストされたアウトラインが原点（0, 0）に揃うように移動されます。



15.2 FontLab Studio

15.2.1 FontLab から Glyphs へ

FontLab から Glyphs 形式のファイルを書き出す、FontLab 用の Python スクリプトがあります。以下のアドレスに行き、Glyphs Export.py をダウンロードしてください。

github.com/schriftgestalt/Glyphs-Scripts/

インストールするには、「~/Library/Application Support/FontLab/Studio 5/Macros/」のフォルダに行き（Finder のメニューバー「移動 > フォルダへ移動」で上記のパスをコピー & ペーストすると簡単にアクセスできます）、Macros のフォルダにスクリプトを入れてください。FontLab を再起動すれば、マクロのツールバーから使えるようになります。

15.2.2 Glyphs から FontLab へ

FontLab で Glyphs 形式のファイルを読み込む、FontLab 用の Python スクリプトがあります。上記の Github リポジトリに行き、Glyphs Import.py をダウンロードして Macros フォルダに入れてください。

FontLab に存在しない機能を使っている Glyphs ファイルを読み込む場合は、結果が同じにならないかもしれません。例えば FontLab ではコンポーネントを入れ子にすることができないため、それらは分解されます。

15.3 Unified Font Object

15.3.1 UFO 形式でファイル保存

Glyphs は UFO ファイルをネイティブに読み書きすることができます。UFO 形式で保存するには、「ファイル > 別名で保存...」を選び、保存ダイアログのフォーマットメニューで「Unified Font Object」を指定します。ただし、Glyphs の全機能が UFO でサポートされているわけではありません。Glyphs 固有の機能を使ったファイルを UFO 形式で保存しようとする、サポートされていない機能が抜け落ちる旨の警告メッセージが表示されます。

15.3.2 UFO 形式でファイル書き出し

安全のため、Glyphs ファイルを基本の保存フォーマットとしておき、UFO は必要に応じて出力するのが良いでしょう。別名で保存する以外にも、「ファイル > 書き出し...」で UFO 形式を選択できます（別名で保存した場合と結果は同一です）。UFO はマルチプルマスターをサポートしていないので、各マスターを個別の UFO ファイルとして保存したい場合はこちらから行う必要があります。

15.3.3 UFO ファイルの読み込み

UFO ファイルを読み込む際、Glyphs は可能な限りアプリケーション標準のデザイン用グリフ名に変換し、コンポーネントグリフを自動整列しようとしています。この動作を阻止したい場合は、「Glyphs > 環境設定 > ユーザー設定」で「読み込み元のファイルのグリフ名を保持」と「読み込んだファイルではコンポーネントの自動整列を解除」のチェックを ON にします。これらの設定はアプリケーションのデフォルト動作となります。環境設定を変えず、読み込んだ後でファイル単位に設定変更をしたい場合は、フォント情報の「その他の設定」で「標準外のグリフ名を使用」と「コンポーネントの自動整列を解除」のチェックを ON にします。

15.4 Type1、OpenType、TrueType

15.4.1 既存のフォントを開く

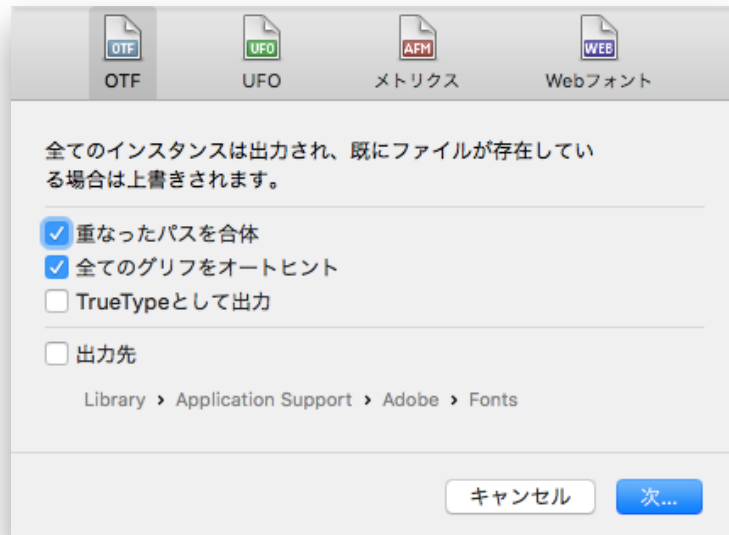
既存フォントの扱いの詳細については、以下のブログ記事をご覧ください（英語）：

glyphsapp.com/blog/importing-existing-fonts

Glyphs は既存の Type1、OTF、TTC、TTF フォントを開くことはできませんが、すべての情報をリバースエンジニアリングできるわけではありません。例えば、OTF フォントを Glyphs で開いて書き出すと、元とは異なるフォントファイルが書き出されます。ヒンティングやその他の OpenType テーブル内のメタデータも破棄されます。オリジナルの Glyphs ファイルを消失してフォントファイルを開く事態にならないよう、必ず Glyphs ファイルの複製を保管して作業するようにしてください。

15.4.2 OpenType/CFF フォントの出力

「ファイル > 出力...」を選ぶと出力ダイアログが表示されます。レイアウトソフトやワープロソフトで使用できる OpenType フォントを出力するには、OTF を選んでください。OpenType/CFF フォント (.otf) を出力する場合、「TrueType として出力」オプションは OFF にしてください。



「重なったパスを合体」オプションは、同名のフィルタを出力時に適用するものです。「4.4 重なったパスを合体」で詳細をご覧ください。製品版のCFF形式フォントでは、重なったアウトラインは必ず合体されていなければいけませんので、テスト用に出力する場合、すでに合体を済ませている場合、あるいはカスタムパラメータで合体のタイミングを変更したい場合以外はチェックをONのままにしてください。

「すべてのグリフをオートヒント」は、まだ手動のヒントングが施されていないグリフにAFDKOのオートヒントングを適用します。このオプションが正常に動作するには、ステム幅とアラインメントゾーンが適切に設定されていなければいけません（第10章 PostScript ヒンティング p.XXX）。ヒントングを施したくない場合、またはテストのために出力時間を短縮したい場合はチェックをOFFにしてください。

「出力先」オプションではフォントの出力先を指定できます。このチェックをOFFにした場合は保存ダイアログが毎回表示されます。もし同名のフォントが既に存在する場合は、警告なく自動的に以前のものが削除されますので注意してください。AdobeのFontsフォルダを出力先に指定すると便利です（「3.12.7 Adobe アプリケーションでのプレビュー」、p.46）。

15.4.3 OpenType/TTフォントの出力

TrueType形式のOpenTypeフォント（.ttf）を書き出すには、「ファイル > 出力...」で表示されるダイアログ（command+E）で「OTF」タブを選択し、「TrueTypeとして出力」のチェックをONにします。

「重なったパスを合体」オプションと「出力先」オプションはCFF形式で出力するときと同様に機能します。詳細は「15.4.2 OpenType/CFFフォントの出力」（p.XXX）を参照してください。

「すべてのグリフをオートヒント」を有効にすると、オートヒントングにWerner Lamberg作のttfautohintアルゴリズムを使用します。この

オプションは手動で設定したTrueType ヒンティングを無視します。オートヒンティングのオプションは、インスタンスの「TTFAutohint Options」カスタムパラメータを追加することで設定できます。カスタムパラメータで設定できる内容については「17.3 カスタムパラメータ」(p.184)を参照してください。もし手動のヒンティングを使いたい場合は（「11.3 手動インストラクション」、p.131）、オートヒントのチェックをOFFにするか、オートヒントを無効にしたいインスタンスのAutohintカスタムパラメータをチェックを外した状態で追加します。Autohintのカスタムパラメータの詳細は「17.3 カスタムパラメータ」(p.184)を参照してください。

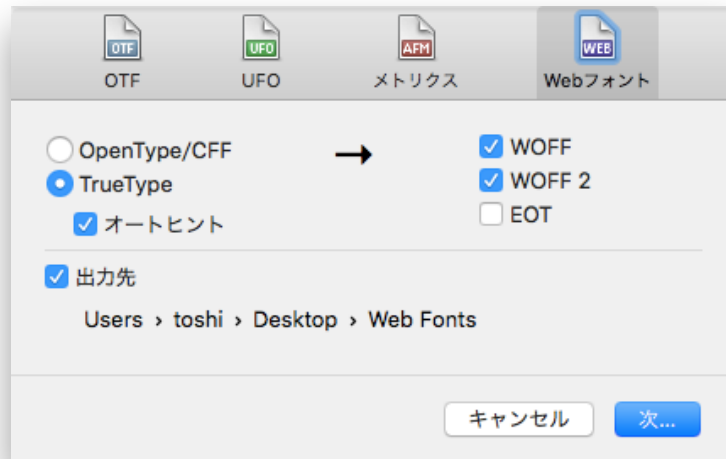
出力されたTrueType フォントでは、ファイルサイズを節約するためコンポーネントが維持されます。ただしコンポーネントが他のコンポーネントやアウトラインと重なっている場合は分解されます（例：Ccedilla、Lslash、Tbar）。ワークフローによっては、アウトライン同士の重なりは合体させたくても、上記のグリフのようなパスとコンポーネント、またはコンポーネント同士の重なりはそのままにしておきたい場合があります。その場合は他のアプリケーションでの後処理を行う必要があります。そのような重なりを実現したい場合はインスタンスのKeep Overlapping Componentsカスタムパラメータを有効にする必要があります。Keep Overlapping Componentsの詳細は「17.3 カスタムパラメータ」(p.184)を参照してください。

15.5 Web フォント

15.5.1 WOFF、WOFF2、EOTフォントの出力

OpenType フォントをWeb用書き出す場合は、「ファイル > 出力...」で表示されるダイアログ (command + E) で「Web フォント」を選択します。GlyphsではWOFF、WOFF2 (Web Open Font Format)、そして古いバージョンのInternet Explorer用にTrueType形式のEOT (Embedded OpenType) をサポートしています。

右側のチェックボックスで、必要なフォーマットを一度にすべて出力することができます。インスタンスのWebfont Formatカスタムパラメータがあると、そちらが優先されてここでのチェック項目は無視されず。Webfont Formatの詳細は「17.3 カスタムパラメータ」(p.184)を参照してください。



左にあるOpenType/CFFとTrueTypeのラジオボタンはWOFFとWOFF2形式のOpenTypeに影響を与えます。EOTファイルは常にTrueTypeのため影響はありません。

オートヒントのオプションは、CFFベースのWOFFに対してはPostScriptのオートヒントングを、TrueTypeベースのWOFFまたはEOTに対してはttfautohintを実行します。インスタンスにAutohintカスタムパラメータがある場合、そちらの設定が優先されます。Autohintの詳細は「17.3 カスタムパラメータ」(p.184)のカスタムパラメータのリストを参照してください。

「出力先」オプションはOTFを出力するときと同様の働きをします。詳細は「15.4.2 OpenType/CFFフォントの出力」(p.163)を参照してください。

15.6 メトリクス

15.6.1 メトリクスデータの読み込み

選択したグリフに対し外部のファイルからカーニングデータを読み込むことができます。「ファイル > 読み込み > メトリクス」を選んで表示されるダイアログで、.glyphs、.metrics、そして.ufoのいずれかの形式のファイルを選びます。続いて現れるダイアログでは、UFOファイルを選択した場合はカーニンググループ（またはクラス）の設定とカーニングペアを読み込むことができ、Glyphs形式のファイルからはカーニンググループのみ読み込めます。Metricsファイルからはサイドベアリングやグリフ幅を読み込むこともできます。選んだファイル形式によって現れるダイアログは違います。



カーニンググループやカーニングペアを読み込む前には、先にグリフが選択されていないとダメです。もし読み込むファイルの内容をフォント内のすべてのグリフに対して適用したい場合は、読み込みコマンドの実行前にすべてのグリフを選択してください。

Metrics ファイルからメトリクスを読み込む際、選択していないグリフで勝手にメトリクスが変わってしまうのを防ぐため、変更が生じるコンポーネントグリフの自動整列を強制的に無効化します。

カーニングペアを他の Glyphs ファイルや他のマスターから読み込みたい場合は、まずそのファイルを開き、コピー元のマスターに切り替えて「ウインドウ > カーニング」(option + command + K)を開きます。カーニングウインドウ内でコピーしたいペアを選択するか command + A ですべてを選択し、command + C でコピーします。続いてペースト先のファイルとマスターを開き、カーニングウインドウにペースト (command + V) します。カーニングの上書きが起こる場合にはダイアログが表示されます。



15.6.2 メトリクスデータの出力

サイドベアリングとカーニングのデータを出力したい場合は「ファイル > 出力...」で表示されるダイアログ (command + E) で「メトリクス」タブを選択します。「次...」をクリックして保存先を選び、表示されたダイアログで、Metrics ファイルか AFM ファイルを選択します。



Metrics ファイルは Glyphs の使用するメトリクスファイル形式でフォントのスペーシングやカーニング情報をすべて内蔵しています。拡張子は .metrics です。Metrics ファイルの読み込みは「ファイル > 読み込み > メトリクス...」から行います。

AFM ファイルは Adobe Font Metrics と呼ばれるフォーマットで、他のフォントエディターでも使用できます。しかし AFM フォーマットは旧式であり、Glyphs で使用するメトリクス情報のすべてを記述できるわけではありません。例えばメトリクスキーやグループカーニングの情報は破棄されてしまいます。

15.7 プロジェクト

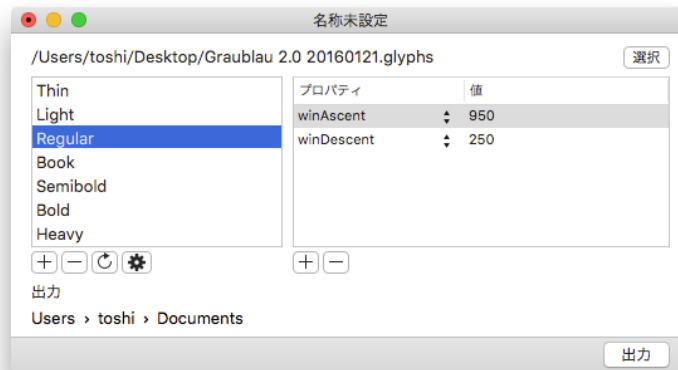
プロジェクトとは .glyphs ファイルから出力されるインスタンスの集まりですが、インスタンスの設定を .glyphs ファイルに保存するのではなく、別の「プロジェクト」と呼ばれるファイルに保存します。これは元の .glyphs ファイルに変更を加えることなくインスタンスを管理するためのもので、別バージョンのフォントを作る際に便利です。プロジェクトファイルの拡張子は .glyphsproject です。

15.7.1 プロジェクトのセットアップ

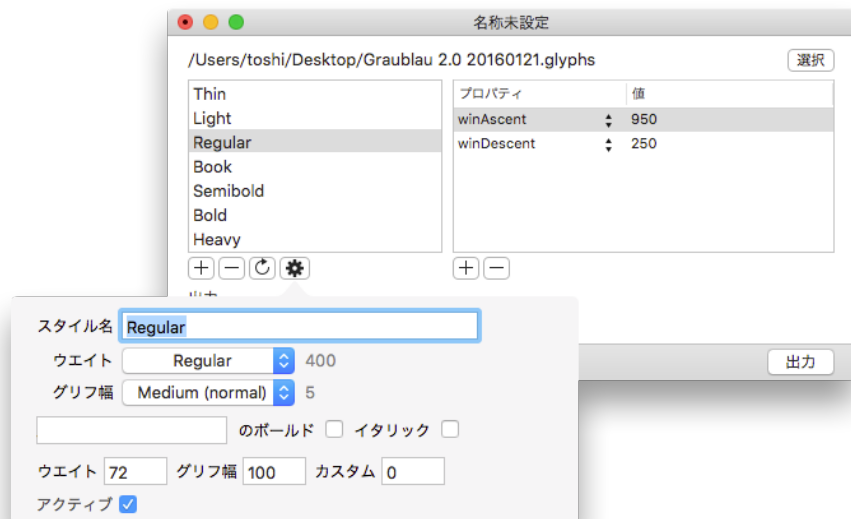
Glyphs プロジェクトを作るには、「ファイル > 新規プロジェクト」を選びます。通常の .glyphs ファイルのように保存できます。

プロジェクトファイルと .glyphs ファイルをリンクさせるには、右上の「選択」ボタンをクリックして .glyphs ファイルを選択します。 .glyphsproject ファイルには、リンク元の .glyphs ファイルの場所が保存されます。 .glyphs ファイルとのリンクが完了すると、そのファイルで設定されていたインスタンスが左のサイドバーに読み込まれます。

読み込み元の Glyphs ファイルを開く必要はありません。ディスク上に保存さえされていれば大丈夫です。



インスタンスはドラッグ&ドロップで順番を変更できます。また、option キーを押しながらドラッグすると、そのインスタンスを複製できます。コピー&ペーストでも複製可能です。+ ボタンでは新規インスタンスを追加、- ボタンでは削除することができます。.glyphs ファイルにあったインスタンスの状態にリセットしたい場合は🔄 ボタンをクリックしてください。⚙️ ボタンをクリックするとインスタンスのスタイル名、ウエイト、字幅、スタイルリンク、補間値、出力設定を変更できます。



カスタムパラメータは、任意のインスタンスを選択しウインドウ右側で編集できます。フォント情報の「インスタンス」タブのカスタムパラメータ欄と同様の感覚で使うことができます。カスタムパラメータについての詳細は「7.3.6 カスタムパラメータ」(p.91) を参照してください。カスタムパラメータのリストを確認したい場合は「17.3 カスタムパラメータ」(p.184) を参照してください。

15.7.2 プロジェクトの出力

インスタンス出力の際に元の .glyphs ファイルが開いている必要はありません。

プロジェクト内のすべてのインスタンスを出力するには、まず .glyphsproject ファイルが保存しており、リンクされている .glyphs ファイル

イルが間違いなくウインドウに表示されている通りの場所に保存されていることを確認してください。そしてウインドウ下の「出力」エリアでファイルパスを選択し、表示されるダイアログでインスタンスの出力先を選択します。その後、右下の「出力」ボタンをクリックして、出力指定されているすべてのインスタンスを生成します。

16 機能拡張

ユーザーは各種のプログラミング言語を利用して、Glyphsが標準で実装していない機能を作成したり、他のユーザーの作成した機能を追加することができます。GlyphsはPython、Objective-C、Swift、AppleScriptに対応しています。

16.1 スクリプト

Pythonスクリプトとは、Pythonというプログラミング言語で記述された.pyという拡張子の付いたファイルで、Glyphsでの作業を自動化するのに使われます。Pythonスクリプトを利用するには「~/Library/Application Support/Glyphs/Scripts」フォルダに.pyファイルをインストールします。このフォルダへは「スクリプト > Scriptsフォルダを開く」(command+shift+Y)で簡単にアクセスできます。なお、このフォルダ内にさらにフォルダを追加してファイルを整理することも可能です。.pyファイルを追加したら、Optionを押しながらメニューバー「スクリプト > スクリプトメニューを更新」(option+shift+command+Y)を選択してください。スクリプトがメニューに表示され、使用可能になります。

GlyphsのPythonスクリプトのコードは、メニューに表示される名前の表記で始まっている必要があります。

```
#MenuItem: Rotate Glyphs
```

「_doc_」で始まる文字列は、スクリプトメニューにカーソルを置くと表示されるツールチップの文章を書くところです。スクリプトの説明を書くといいでしょう。

```
_doc_="This script rotates selected glyphs."
```

Glyphs公式サイトにはGlyphsでのPythonプログラミングAPIの詳細な説明があります(英語)。

```
docu.glyphsapp.com
```

初心者の方は、三編に分かれたチュートリアルがGlyphsのWebサイトにありますので、そちらを参照するとよいでしょう(英語)。

```
glyphsapp.com/tutorials/articles/tag:scripting
```

またGlyphs用サードパーティー製のスクリプトもGitHub上で多く公開されています。Glyphsのサイトでもその中からいくつかを厳選して紹介しています。

```
glyphsapp.com/extend
```

16.2 プラグイン

16.2.1 プラグインの手動インストール

プラグインはダブルクリック、またはGlyphsのアプリケーションアイコンにドラッグ&ドロップすることで、自動的に「~/Library/Application Support/Glyphs/Plugins」フォルダにインストールされます。またエイリアスをインストールすることもできます。アンインストールしたい場合はフォルダから外してください。このフォルダに素早くアクセスする方法は2つあります。1.「スクリプト > Scripts フォルダを開く」を選び、Scripts フォルダの隣にあるPlugins フォルダを選ぶ。2. 環境設定「アドオン > プラグイン」から既にインストールされているプラグインを選択し、右クリックで「Finderに表示」を選ぶ。

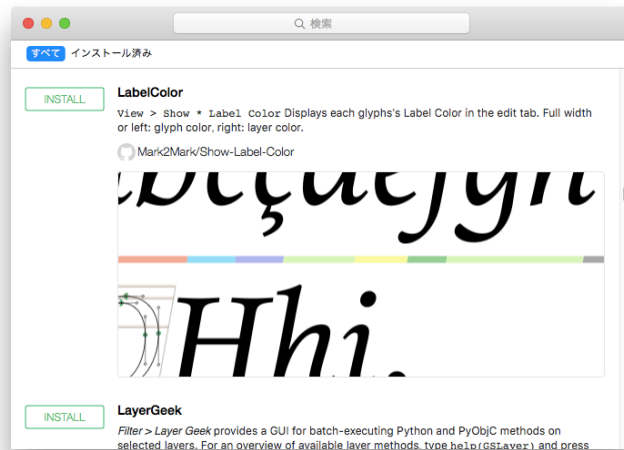
Glyphsにインストールできるプラグインには様々な種類があります。これはファイルの拡張子で判別できます。

- ・リポーター（拡張子.glyphsReporter）：表示メニューにオプションを追加し、編集ビューに表示する情報を追加します。
- ・フィルタ（拡張子.glyphsFilter）：選択したグリフに視覚的な処理をするフィルタを追加します。ダイアログが出るものも、ないものもあります。カスタムパラメータに対応しているものもあります。詳細はプラグインに付属の資料を参照してください。
- ・ファイルフォーマット（拡張子.glyphsFileFormat）：ファイルを開く際、フォントを出力する際のファイル形式の選択肢を追加します。
- ・パレット（拡張子.glyphsPalette）：パレットに機能を追加します。
- ・ツール（拡張子.glyphsTool）：ツールバーにツールを追加します。
- ・一般（拡張子.glyphsPlugin）：上記以外の機能を追加します。

16.2.2 プラグインマネージャー

プラグインをインターネット上で探したり手動でインストールする手間が煩わしいという場合には、「ウインドウ > プラグインマネージャー」をお勧めします。このウインドウでは、マネージャーが登録しているプラグインを検索したり、説明を読んだり、リンクされているウェブサイトに行くことができます。プラグインの説明に画像が含まれている場合、クリックすると原寸表示されます。

プラグインマネージャーを使用するにはXCodeのgitプラグインをインストールする必要があり、初めて開く際にはインストールダイアログが表示されます。インストール後はGlyphsを再起動するとプラグインマネージャーを使用できるようになります。



プラグインはアプリケーション開始時にのみ読み込まれるため、プラグインのインストールやアンインストール後はGlyphsを再起動する必要があります。

ここからインストールするには、インストールボタンを押します。アンインストールするには削除ボタンを押します。ウインドウ左上で「インストール済み」ボタンを押すと、マシン内にインストールされたプラグインのみが表示されます。

自作したプラグインをGlyphsのプラグインマネージャーに表示させたい場合は、GitHubのプラグインリストにpullリクエストを送ってください。詳細はGlyphsSDKのreadmeファイルに記されています。

16.3 SDK

GlyphsSDKはオープンソースのGlyphs用ソフトウェア開発キットです。これはGlyphsユーザーコミュニティ内におけるプラグイン、スクリプト、.glyphsファイル処理する独立したアプリケーションなどの開発を触発するために作られたものです。SDKにはスクリプトAPIの資料、.glyphsファイルの仕様、プラグイン開発のためのテンプレートを用意しています(XcodeのObjectiveC用とPyObjC bridgeを使ったPython用の両方が用意されています)。SDKは以下のGitHubリポジトリからダウンロードしてください：

github.com/schriftgestalt/GlyphsSDK

17 補足資料

17.1 フィーチャー自動生成

Glyphsには自動でOpenTypeフィーチャーを生成する機能があります。この自動生成を利用するには、使用されるグリフを特定の命名規則に合わせておく必要があります。

- aalt** *All Alternates* すべてのオルタネートにアクセスするためのフィーチャーです。GlyphsはGSUB（グリフ置換系）の各種フィーチャーを認識して自動でaaltを生成します。ユーザーが意図的にグリフ名で自動生成させることはできません。
- liga** *Standard Ligatures* 通常のリガチャー（合字）です。フォントを使用するアプリケーションでは、通常はデフォルトで有効になっていて、ユーザーが無効にすることもできます。リガチャー用のグリフ名は、その基になるグリフ名をアンダースコア（`_`）で繋げて書いてください。欧文以外で「`-cy`」や「`-hira`」など末尾に文字識別子が付いているグリフをリガチャーにする場合、識別子は末尾に一度明記するだけで十分です。よくあるリガチャー（`f_f_i`、`f_f_l`、`f_f`、`fi`、`fl`、`lu_lakkhangyao-thai`、`ru_lakkhangyao-thai`など）は自動的にligaフィーチャーに登録され、他はすべてdligに登録されます。また任意のリガチャーグリフをdligではなく強制的かつ自動的にligaフィーチャーに登録してほしい場合はグリフ名の末尾に「`.liga`」を付けてください。例：`f_b.liga`、`yi_yi-cy.liga`
- dlig** *Discretionary Ligatures* 任意のリガチャーです。フォントを使用するアプリケーションでligaは全グリフへ一律に適用されますが、dligはデフォルトで無効になっており、ユーザーが任意の箇所個別に有効にして適用します。一般にdligにはligaよりも装飾度の強いもの、重要度の低いものを含めます。グリフ名は、その基になるグリフ名をアンダースコア（`_`）で繋げて書いてください。例：`s_t`、`f_odieresis`
- hlig** *Historical Ligatures* 古典的な文字のリガチャーで、longs（長いs）のリガチャーが典型的な用途です。自動でフィーチャーを生成させたい場合はグリフ名の末尾に`.hlig`または`.hist`を追加してください。

	rlig <i>Required Ligatures</i>	<p>フォントを使用するアプリケーションでは常に有効であり、ユーザーが無効化の選択をできない強制発動リガチャーです。基本的にはアラビア文字の lam と alif など、ルール上絶対に合字にならないといけないものに対して使用されます。グリフ名の末尾に「.rlig」を付けてください。また以下のグリフ名に対しては自動的に rlig フィーチャーが生成されます : lam_alef-ar、lam_alefHamzaabove-ar、lam_alefHamzabelow-ar、lam_alefMadda-ar、lam_alef-ar.fina、lam_alefHamzaabove-ar.fina、lam_alefHamzabelow-ar.fina、lam_alefMadda-ar.fina、lam_alefWasla-ar、lam_alefWasla-ar.fina</p>
<p>c2sc と smcp の両方のスモールキャップを作った場合、「c2sc」系列は「A.c2sc」などのように大文字と、「smcp」は「a.smcp」などのように小文字と対応させてください。</p>	c2sc <i>Capitals to Small Capitals</i>	<p>大文字をスモールキャップにするフィーチャーです。Adobe アプリケーションでは「すべてスモールキャップス」を選ぶことで smcp フィーチャーと共にアクセスできます。グリフ名の末尾に「.sc」、「smcp」、または「.c2sc」を付けてください。</p>
	smcp <i>Small Capitals</i>	<p>小文字をスモールキャップに変換するフィーチャーです。グリフ名の末尾に「.sc」、「smcp」、または「.c2sc」を付けてください。</p>
	c2pc <i>Capitals to Petite Capitals</i>	<p>大文字をプチキャップにするフィーチャーです。グリフ名の末尾に「.pc」、「c2pc」、または「.pcap」を付けてください。</p>
	pcap <i>Petite Caps</i>	<p>通常スモールキャップでは小文字よりハイトを上げますが、プチキャップという小文字と高さが完全に揃った大文字も存在し、これには専用の pcap フィーチャーを使います。グリフ名の末尾に「.pc」、「c2pc」、または「.pcap」を付けてください。</p>
	sup <i>Superscript</i>	<p>上付き文字です。グリフ名の末尾に「.sup」を付けてください。数字に限ってはピリオドなしの「superior」をつける方法も可能です。例 : onesuperior</p>
	sub <i>Subscript</i>	<p>下付き文字です。グリフ名の末尾に「.sub」を付けてください。数字に限っては「inferior」も許可されます。</p>

もし製作中のフォントファイルで特に下付きと化学用下付きの区別をつけない場合は、単純に「.subs」のグリフだけを作っておけば、subsとsinfの両方のフィーチャーが自動的に生成されます。

sinf *Scientific Subscript*

汎用の下付き文字とは別に用意された科学用途の下付き文字です。グリフ名の末尾に「.sinf」または「.subs」を付けてください。

afrc *Alternate Fractions*

afrcフィーチャーはfracフィーチャー（後述）とは違い、数学で使用されるような水平分割の分数を表示するためのものです。始めから定義された組み合わせのみサポートし、fracとは違ってどのような数字の組み合わせにも対応しているわけではありません。このフィーチャーをサポートするには以下の分数グリフをフォント内に用意してください：

oneovertwo、zerooverthree、oneovertwo、twoovertwo、threeovertwo、fourovertwo、fiveovertwo、sixovertwo、sevenovertwo、eighthovertwo、nineovertwo、tenovertwo、elevenovertwo、oneovertwo、zerooverthree、oneoverthree、twooverthree、oneoverfour、threeoverfour、oneoverfive、twooverfive、threeoverfive、fouroverfive、oneoversix、fiveoversix、oneoverseven、twooverseven、threeoverseven、fouroverseven、fiveoverseven、sixoverseven、oneovertwo、threeovertwo、fiveovertwo、sevenovertwo、oneovernine、twoovernine、fourovernine、fiveovernine、sevenovernine、eighthovernine、oneoverten、threeoverten、sevenoverten、nineoverten、oneovereleven、twoovereleven、threeovereleven、fourovereleven、fiveovereleven、sixovereleven、sevenovereleven、eighthovereleven、nineovereleven、tenovereleven、oneovertwelve、fiveovertwelve、sevenovertwelve、elevenovertwelve、oneoveronehundred

frac *Fractions*

斜めのスラッシュで区切られた分数にするフィーチャーです。 .numr と .dnom の付いたグリフ、そして fraction（分数用スラッシュのグリフ）を組み合わせで自動生成します。もしこれらが存在しない場合、onehalf、onequarter、threequarters など始めから結合されている分数がフィーチャーに使用されます。

dnom *Denominators*

分数の分母用数字です。グリフ名の末尾に「.dnom」を付けてください。

numr *Numerators*

分数の分子用数字です。グリフ名の末尾に「.numr」を付けてください。

	<p>onum <i>Oldstyle Numerals (Oldstyle Figures)</i></p>	<p>オールドスタイル数字です。オールドスタイル数字とは、小文字の高さに合っていて（スモールキャップと同様、実際は小文字より少し高め）、かつ数字によってはアセンダーやディセンダーのあるデザインのもので、プロポーショナルのオールドスタイルにはグリフ名の末尾に「.osf」を、等幅のものには「.tosf」を付けてください。通貨記号など、数字以外のグリフにも適用できます。</p>
<p>Tip: デフォルトの数字となるグリフには「.lf」、「.tf」、「.osf」、「.tosf」などを付加しないでください。Glyphsは他にあるグリフの名前や幅の情報から、デフォルトの数字がどのタイプなのかを判別し、自動でフィーチャーを生成します。</p>	<p>tnum <i>Tabular Numerals (Tabular Figures)</i></p> <p>pnum <i>Proportional Numerals (Proportional Figures)</i></p>	<p>等幅 (tabular) な数字です。等幅のライニング数字にはグリフ名の末尾に「.tf」を、等幅のオールドスタイル数字には「.tosf」を付けてください。通貨記号など、数字以外のグリフにも適用できます。</p> <p>字幅がそれぞれ異なるプロポーショナルな数字で、tnumの対となるフィーチャーです。プロポーショナルのオールドスタイルにはグリフ名の末尾に「.osf」を、プロポーショナルのライニング数字には「.lf」を付けてください。通貨記号など、数字以外のグリフにも適用できます。</p>
	<p>lnum <i>Lining Numerals (Lining Figures)</i></p>	<p>高さの揃った数字はライニング数字と呼ばれ、オールドスタイル数字とは対となります。プロポーショナルのライニング数字にはグリフ名の末尾に「.lf」を、等幅のライニング数字には「.tf」を付けてください。通貨記号など、数字以外のグリフにも適用できます。</p>
	<p>ordn <i>Ordinals</i></p>	<p>序数です。numero (№)、ordfeminine (ª)、ordmasculine (º) がフォントに含まれていると自動で生成されます。</p>
	<p>mgrk <i>Mathematical Greek</i></p>	<p>特定のギリシャ文字 (Delta、Omega、Pi、Sigma、alpha、mu、phi、epsilonLunateSymbol、epsilonLunateReversedSymbol) に対して、数学用のオルタネート (increment、Ohm、product、summation、proportional、micro、phiSymbol、element、containsasmemberSmall) を呼び出すためのフィーチャーです。このフィーチャーは現在は使用が推奨されていないためGlyphsは自動では追加しませんが、ユーザーが手動で追加すればフィーチャーの更新は自動で行います。</p>

- ornm** *Ornaments* オーナメント（花形装飾）を使用するためのフィーチャーです。大文字A-Zと小文字a-zに「.ornm」のついたオルタネートを用意してください。また「bullet」グリフが必ずフォント内に存在するようにしてください。
- hist** *Historical Forms* 歴史的な形のオルタネートです。典型的な例は小文字のsを長いs (ſ) に切り替えるという用法です。グリフ名の末尾に「.hist」を付けてください。
- case** *Uppercase Forms* オールキャップでの文字組専用位置や大きさが調整された大文字専用オルタネートグリフです。よくある例は大文字の中心の高さに揃ったハイフン (-) や括弧 (())、発音記号などです。Adobe アプリケーションではフォントパネルから「オールキャップす」を選ぶことでアクセスできます。グリフ名の末尾に「.case」を付けてください。また「.lf」の付いた数字（ライニング数字）もここで使われます。
- cpsp** *Capital Spacing* 大文字のスペーシングを調整するためのフィーチャーです。オールキャップ組みでは字間を少し広めに組む方がよいのですが、このスペーシングを始めからフォントに組み込んでおくことができます。Adobe アプリケーションではフォントパネルなどから「すべて大文字」を選ぶことでアクセスできます。フォント情報の「フィーチャー」タブで + ボタンをクリックし、Capital Spacing を選ぶと追加されます。

- locl** *Localised Forms* このフィーチャーは特定の言語でより望ましい結果を得るために存在します。Adobe アプリケーションではテキストブロックの言語を指定すると、locl フィーチャーの当該言語のコードが実行されます。グリフ名の末尾に「.loclXXX」を付けてください。XXX の部分にはOpenType の言語タグが入ります（例：.loclENG で英語用、.loclSVE でスウェーデン語用）。他にも初期状態で自動的にフィーチャーを生成するグリフがあります：
- トルコ語、アゼルバイジャン語、クリミア・タタール語、カザフ語、タタール語（それぞれ言語タグはTRK、AZE、CRT、KAZ、TAT）ではドットのあるiとないiの区別があり、大文字との対応が他の言語とは違います（上記言語ではi→İとı→I、英語ではi→I）。正しい表示にはiと見た目が同じオルタネートとlocl

こちらで言語タグの一覧を確認できます：

microsoft.com/typography/otspec/lanagugetags.htm

フォントの各言語対応についてさらに詳細を知りたい場合は以下のページにアクセスしてください（英語）：

glyphsapp.com/tutorials/tag/languages

フィーチャーが必要です。フォント内にidotaccent、i.TRKまたはi.loclTRKがあると（グリフのデザインは小文字iと同一）、上記言語の小文字iのフィーチャーが自動的に生成されます。

- Scommaccent、Tcommaaccent、scommaccent、tcommaaccentが存在すると、ルーマニア語、モルドヴァ語（ROM、MOL）の選択時にScedilla、Tcedilla、scedilla、tcedillaは前者4つに置き換えられます。これら2つの言語のためにセディーユ付きのS/sとT/tにUnicode符号が与えられましたが、正しくはコンマアクセント（別称コンマビロー）付きであることが判明したため、後にそちらも符号化した経緯があります。セディーユ付きSとTはUnicodeへの追加以来、ある程度普及してしまっていますが、現在は上記2言語では可能な限りコンマアクセント付きのSとTを使うことが推奨されています。上記のグリフがフォント内に存在すると、必要なコードが自動で生成されます。
- カタルーニャ語（CAT）ではperiodcenteredが2連続のlの発音を区別するスペリングに使われます（例：cellaとcel·laの発音はそれぞれセーヤとセッラ、後者のl·lはella geminadaと呼ばれます）。ただしperiodcenteredをそのまま使用すると位置が低すぎるのでlの中央揃いに見えるように高さを変更し、スペーシングも空きすぎて1単語に見えなくなるのでキツめにします。大文字では最初のLのカウンターの中に浮くような状態になります。これを実現する方法は3つあります。まずはカタルーニャ語専用のperiodcentered.loclCATとperiodcentered.loclCAT.caseを用意することで、これとL、lがフォント内に存在すると必要なコードが自動で生成されます。2つめはL_periodcentered_L.loclCATとl_periodcentered_l.loclCATという始めから合字のように作ったグリフがあると、これもコードが自動で生成されます。3つめはUnicodeに登録されているLdotとldotがフォント内に存在すると、同じく必要なコードが自動で生成されます。3つ目の方法についてですが、Ldotとldotは現在使用が推奨されていないので、先の2つのいずれかを採用するのが望ましいものの、条件を満たしていない場合には3つ目のコードが最終手段として生成されます。

- ・オランダ語（NLD）ではIJとijは2重母音ですが、しばしば1文字とみなされることがあります。またオランダ語では母音にアクセント記号を付けることで発音を強調することができ、ijを強調したい場合は両方にアクセントをつけるのが望ましいのですが、Unicodeにはアクセント付きのjが存在せず、通常はijになってしまいます。Iacute_J.loclNLDとiacute_j.loclNLD、またはJacuteとjacuteがあると、オランダ語のijのアクセント付きオルタネートのフィーチャーが自動生成されます。Iacute、iacute、J、そしてjがフォントに含まれていることも生成の条件です。
- ・ウルドゥー語で使用される数字はペルシャ語用数字をベースにしていますが（ペルシャ語用数字はアラビア文字用の数字とは別です）、4、6、7などのいくつかの数字ではウルドゥー語専用のデザインのオルタネートに切り替えるのが理想です。グリフ名の末尾に.loclURDの付いたペルシャ語用数字があると、ウルドゥー語用に必要なコードが自動で生成されます。またアラビア文字用の6はウルドゥー語で受け入れられているため、six-arがあればウルドゥー語でも再利用されるようコードが生成されます（いずれかのケースで対応すれば十分です）。

cv01– *Character*
cv99 *Variants*

文字のバリエーションで、最大99個まで作れます。グリフ名の末尾に「.cv01」から「.cv99」を付けてください。

ss01– *Stylistic Set*
ss20

デザインの見た目を切り替えるセットで、最大20セットまで作れます。現状、AdobeアプリケーションではInDesignでしか利用できません（テキストエディットやPagesなどOSX標準のタイポグラフィパネルを使用できるアプリケーションは利用できます）。グリフ名の末尾に「.ss01」から「.ss20」を付けてください。なお、フィーチャー設定下部のメモ欄に「Name:」で始まる分かりやすい名前をつけておくと、Glyphsはそれをセットの名前として自動的に設定します。もしくはメモ欄にfeatureNamesのAFDKOコードを書いても有効です。

salt	<i>Stylistic Alternates</i>	デザインの見た目を切り替えるフィーチャーです。Adobe 系列のアプリケーションでは、InDesign では ss01～ss20 が使われる代わりに Illustrator と Photoshop では salt が使われます。デフォルトでは、Glyphs は ss01 フィーチャーを salt にも記述します。
swsh	<i>Swashes</i>	スワッシュのオルタネートです。グリフ名の末尾に「.swsh」を付けてください。
titl	<i>Titling</i>	タイトル用のオルタネートです。グリフ名の末尾に「.titl」を付けてください。
init	<i>Initial Forms</i>	語頭形のオルタネートです。特にアラビア文字など、単語内の位置によって字形が変化する文字によく使われます。グリフ名の末尾に「.init」を付けてください。
medi	<i>Medial Forms</i>	語中形のオルタネートです。グリフ名の末尾に「.medi」を付けてください。
med2	<i>Medial Forms 2</i>	語中形のオルタネート2です。シリア文字でのみ使用されます。グリフ名の末尾に「.med2」を付けてください。
fin	<i>Final Forms</i>	語尾形のオルタネートです。グリフ名の末尾に「.fin」を付けてください。
fin2	<i>Final Forms 2</i>	語尾形のオルタネート2です。シリア文字でのみ使用されます。グリフ名の末尾に「.fin2」を付けてください。
fin3	<i>Final Forms 3</i>	語尾形のオルタネート3です。シリア文字でのみ使用されます。グリフ名の末尾に「.fin3」を付けてください。
hwid	<i>Half Widths</i>	半角グリフ幅のオルタネートです。グリフ名の末尾に「.half」を付けてください。
vrt2	<i>Vertical Alternates and Rotation</i>	縦書き時に表示される回転字形のオルタネートです。グリフ名の末尾に「.vertical」、「.vert」または「Vertical」を付けてください。

akhn <i>Akhands</i>	アカンドと呼ばれる子音同士の合字で、デヴァナーガリー文字では最優先されるフィーチャーです。デヴァナーガリー文字のk-deva、j-deva、ssa-deva、nya-deva、そしてk_ssa-deva、j_nya-devaがあると生成されます。
blwf <i>Below Base Forms</i>	本体の子音の下に連なる子音のオルタネートです。デヴァナーガリー文字ではra-deva、halant-deva、そしてrashtrasign-devaがあると生成されます。
cjct <i>Conjunct Forms</i>	デヴァナーガリー文字や他のインド系文字の結合文字があると生成されます。
half <i>Half Forms</i>	デヴァナーガリー文字や他のインド系文字のグリフ名の末尾が「Halfform」になっているハーフフォームかhalantがあると生成されます。
nukt <i>Nukta Forms</i>	デヴァナーガリー文字や他のインド系文字でグリフ名の末尾がNuktaになっているヌクター付きリガチャ、それに対応するヌクター無しのグリフがあると生成されます。
rkrf <i>Rakar Forms</i>	デヴァナーガリー文字や他のインド系文字でラカールのリガチャ、それに対応する独立したグリフとhalantがあると生成されます。
rphf <i>Reph Forms</i>	デヴァナーガリー文字のra-deva、halant-deva、reph-deva、または他のインド系文字で同様の働きをする文字があると生成されます。
ccmp <i>Glyph Composition and Decomposition</i>	様々な文字の合体や分解処理がccmpの自動生成で行われます。例えばフォント内にi、j、idotless、jdotless、そして合成用発音記号があると、iやjの後に発音記号が続いた場合に欧文用のルックアップが呼び出され、発音記号が配置できるようにドットがないグリフに置き換えられます。

mark *Mark to Base
Positioning*

基底文字のグリフの指定した位置に結合文字のグリフを配置するフィーチャーです。まず、基底文字のグリフに top や bottom などのアンカーを追加してください。そして結合文字のグリフ (acutecomb など) にもアンカーを追加し、その名前の先頭にアンダースコアを付けてください (acutecomb などの上付き用発音記号には _top アンカーを追加)。こうすることで、フォント使用時に基底文字の上下へ任意の結合文字を配置できます。結合文字のグリフはフォント出力時に自動的にグリフ幅が 0 に変更されます。mark フィーチャーはフォント情報の「フィーチャー」タブには追加されず、フォント出力時に Glyphs 側で自動的に追加されます。

mkmk *Mark to Mark
Positioning*

結合文字のグリフの指定した位置に結合文字のグリフを配置するフィーチャーです。mark フィーチャー用のアンダースコア付きアンカーだけでなく、アンダースコアなしのアンカーも追加してください (「グリフ > アンカーを自動設置 (command + U)」を実行すれば自動で配置されます)。例えば acutecomb の上に、他の基底文字と同様に top アンカーを追加します。こうすることで、フォント使用時に基底文字へ複数の結合文字を配置できます。結合文字のグリフはフォント出力時に自動的にグリフ幅が 0 に変更されます。mkmk フィーチャーはフォント情報の「フィーチャー」には追加されず、フォント出力時に Glyphs 側で自動的に追加されます。

17.2 クラス自動生成

OpenType クラスの中には条件次第で自動的に追加、更新できるクラスがあります。

- All** フォント内のすべてのグリフのクラスです。必要に応じてユーザーが追加する必要がありますが、一度追加されれば更新は自動で行うことができます。
- AllLetters** フォント内で文字カテゴリに分類されるすべてのグリフのクラスです。必要に応じてユーザーが追加する必要がありますが、一度追加されれば更新は自動で行うことができます。
- ArabicLetters** フォント内で文字カテゴリかつアラビア文字に分類されるすべてのグリフのクラスです。
- DevaHalfForms** デヴァナーガリー文字のハーフフォーム字形のクラスです。
- Uppercase** フォント内で文字カテゴリかつ大文字サブカテゴリに属するすべてのグリフのクラスです。大文字がフォント内にあり cspc フィーチャーが追加されている場合に自動で追加されます。

17.3 カスタムパラメータ

この項で鍵括弧「」で記されている内容は、特に明記されていない限りはすべて Microsoft OpenType 仕様書からの引用です：

microsoft.com/typography/otspec/

UFO3 フォント情報のプロパティは以下のページに記されています：

unifiedfontobject.org/versions/ufo3/

カスタムパラメータはプロパティと値の2つで構成されています。この補足資料では、プロパティを太字で示し、続いて値のタイプを斜体で表記、そしてパラメータの機能を説明します。

フォント情報のフォント、マスター、およびインスタンスのタブの「カスタムパラメータ」欄では、有効なカスタムパラメータが黒で、無効なものには灰色で表示されます。カスタムパラメータを無効にするには存在しないプロパティ名に変更するのが最も手っ取り早いでしょう（名前の頭に *x* を追加するなど）。

キャメルケース（大文字と小文字の組み合わせでスペースを入れないスペリング。例えば `redFish` や `RedFish` はキャメルケースですが `Red Fish` や `redfish` は違います）で記されたカスタムパラメータ名は UFO の仕様に記載されており、フォント情報の変更にかかわるものです。Glyphs 独自のものはスペースを使用し語頭が大文字になっており、大抵これらはフォント内の他のデータを変更するものです（フィルタを実行するなど）。UFO パラメータの名前は 2012 年 3 月に発行された UFO3 仕様の命名法に則っています。Glyphs ではパラメータ名を短縮することができ、可能な場合、名前の前半部分を省略することができます。例えば「`openTypeNameDescription`」の代わりに「`description`」と記すこともできますし、「`postscriptBlueScale`」の代わりに「`blueScale`」と記すこともできます。省略した名前とフルネームは混ぜて使うことも可能です。

値のタイプはブーリアン（真か偽、有効か無効などの2択）、小数、整数、自然数（正の整数）、文字列、カラーパレット、設定、リスト（たいてい場合は1行につき1項目）のいずれかです。

Autohint ブーリアン フォント出力ダイアログのオートヒント設定の内容に関わらず強制的にオートヒントを実行または無視します。

blueScale 小数 BlueScale値です。これはType 1/CFFのBlueScaleに対応しています。オーバーシュートの押し込みが解除される（オーバーシュートが1ピクセル分表示され始める）フォントサイズを制御するプロパティであり、 $(300\text{dpiでのポイントサイズ}-0.49)\div 240$ という計算式で算出できます。例えば300dpiの10ポイント以降でオーバーシュートを可視化させる場合は0.039625です。もし値を独自に設定しない場合は、blueScaleはデフォルト値である0.037を使用し、これは300dpiにおける9.37ポイントまたは39ピクセルに相当します。その場合、オーバーシュートはフォントサイズが40ピクセルからはオーバーシュートが再現されるようになるということです。また300dpiのレーザープリンタでも9.37ポイントまでは押し潰しが発生します。blueScaleの最大値はフォントのアラインメントゾーンのサイズによります。300dpiにおける最大のポイントサイズの計算式は $0.49 + 240\div$ フォント内の最大のアラインメントゾーンのサイズであり、これは $2.04 + 1000\div$ フォント内の最大のアラインメントゾーンのサイズで算出されるppm（ピクセル単位のフォントサイズ）に対応しています。（最大のポイントサイズ-0.49）×（最大のアラインメントゾーンのサイズ）は240未満でなければいけません。

例えば、もしあなたのフォント内の最大のゾーン幅が21ユニットである場合、 $2.04 + 1000\div 21 = 49.659$ なので、最大49ppm（pixels per eM、ピクセル数で表現されるフォントサイズ）まではオーバーシュートが押し込まれるのが合理的でしょう。300dpiでそれに対応するサイズは $0.49 + 240\div 21 = 11.919$ ポイントになり、つまりblueScaleの値は $(11.919 - 0.49)\div 240 = 0.04762$ を超えてはいけません。

blueShift 整数または小数 BlueShiftの値です。これはType 1/CFFのBlueShiftに対応しています。blueScaleで設定されたフォントサイズより細かなオーバーシュートの表示を制御する値で、初期値は7です。アラインメントゾーン内のオーバーシュートは以下の条件を満たした時に表示されます。(a) オーバーシュートのはみ出し具合がBlueShift以上である場合。(b) BlueShiftよりオーバーシュート量が小さいが、0.5ピクセルよりは大きい場合。例えばblueScaleにより32ppmまでオーバーシュート不可視になっておりblueShiftが6で、アラインメントゾーンのサイズが12ユニットのケースを考えましょう。CやOのような文字は12ユニットに到達していますが、それとは別にHなどのセリフ形状が平坦ではなく緩やかな凹型で、そのオーバーシュート量は5ユニットであるとします。0～32ppmまではベースラインは完全に真っ平らに表示されます。33ppmからはCやOのオーバーシュートが表示されるようになります。しかし先に挙げた5ユニットの小さなオーバーシュートは、100ppmになるまで半ピクセルに満たないため平らなままとなります。

CJK Grid 整数 日中韓の文字（漢字や仮名、ハングルなど）を編集集中に表示される点線のグリッドの線の本数です。CJK Grid HorizontalとCJK Grid Verticalを使うと縦と横で違う分割率を適用できます。CJK Gridカスタムパラメータがない場合、グリッドは表示されません。またCJK Guideと同様、文字種専用のグリッドを設定できます。

CJK Grid Horizontal 整数 日中韓の文字を編集集中に表示される点線のグリッドの縦線の数です（水平方向の分割回数）。またCJK Guideと同様、文字種専用のグリッドを設定できます。

CJK Grid Vertical 整数 日中韓の文字を編集集中に表示される点線のグリッドの横線の数です（垂直方向の分割回数）。またCJK Guideと同様、文字種専用のグリッドを設定できます。

CJK Guide 小数または文字列 日中韓の文字を編集集中に表示される補助の正方形のサイズをパーセントで指定します。（例：10%小さい正方形は10）。ただし両端からそれぞれ10%引くので、正方形の幅および高さは合計20%小さくなります。このパラメータがない場合は表示されません。使用を想定した文字の名前を付記すると、その文字にのみ現れます（例：kana:5とすると、片仮名および平仮名にのみ5%引きの正方形が現れ、漢字など他の文字には現れません）。平仮名はhira、片仮名はkata、両方はkana、漢字はhan、ハングルはhangulとなっています。文字によってサイズの違う正方形を表示させたい場合は、その分だけCJK Guideカスタムパラメータを追加してください。

codePageRanges リスト OS/2テーブルのulCodePageRange1とulCodePageRange2を設定します。「このフィールドはフォントファイルが含むコードページをプラットフォーム3、エンコーディングID 1（Microsoftプラットフォーム）向けに知らせるためのものです。ここに記されるすべてのコードページはしっかり機能している前提となります。それぞれのページは独立したフラグであり、どのように組み合わせても構いません。しっかり機能している、とはフォントデザイナー側の解釈に任せられますが、コードページ内の文字は確実に表示されることが期待されます。」

Color Palette カラーパレット Microsoft式のOpenTypeのCPALテーブルを出力します（詳細は「13.3 Microsoft カラーフォント」(p.151)を参照してください）。このパラメータがあると、Glyphsのフォントビュー内でカラーグリフが表示されるようになります。

compatibleFullName 文字列 互換用フルネームです（旧Mac OS専用。OS Xおよび他のOSでは使われません）。OpenTypeのnameテーブルID 18に相当します。もしこれが設定されていない場合、ID 18は各インスタンスのファミリー名とスタイル名を用いて自動的に生成されます。「Macintoshでは、このメニュー名はFONDリソースを用いて構築されます。これは通常Full Nameと一致します。もしFull Nameと違うフォント名を表示させたい場合は、ID 18に任意のCompatible Full Nameを設定することが

できます。」

Compatible Name Table ブーリアン 旧式のネームテーブルを出力するよう指定します。Macのアプリケーションのなかにはこのテーブルが入っていることを想定しているものがあります(例: Quark XPress、FontExplorer)。

copyright 文字列 著作権表記です。フォント情報の「フォント」タブの著作権表記は無視され、こちらが優先されます。これはOpenTypeのnameテーブルID 0に相当します。

description 文字列 フォントの説明です。OpenTypeのnameテーブルID 10に相当します。「タイプフェイスの説明です。更新履歴、推奨用途、開発の経緯、フォントの特徴などを記述できます。」

DisableAllAutomaticBehaviour ブーリアン これが有効の場合、フォント出力時に実行されるすべての自動処理が発動しなくなります。これには非スペーシンググリフの字幅ゼロ調整、OpenTypeフィーチャーの更新、グリフ名の製品用名への変更があります。特殊なフォント製品化のワークフローが関わる場合にのみ使用するようになっています。

Disable autohinting for glyphs 文字列リスト グリフ名のリストです。このパラメータに挙げられたグリフにはPostScriptのオートヒンティングが省略されます(TrueTypeオートヒントはグリフごとに省略できません)。装飾度の強いオーナメントなど、フォント内の特定のグリフのみヒンティングを省きたいときに有効です。

Disable Last Change ブーリアン .glyphsファイルに前回更新日の情報を保存するのを防止します。SVNやGitなどのバージョンコントロールシステムの使用時、無意味なファイル内容の更新が起きてしまうのを避けるのに有効です。

Disable Masters 文字列リスト マスター名のリストです。挙げられた名前のマスターを無効化し、残りのマスターから補間を試みます。これは3つ以上のマスターがあるファイルで、任意のマスターをファイルから取り除いても良好な補間結果が得られるかどうかを試す実験用に使われます。

Disable Subroutines ブーリアン これが有効の場合、フォント出力時にアウトラインの自動処理が無効になります。フォントのアウトラインが複雑でポイント数が多く出力ができない場合、またはグリフ数がとても多いフォントをテスト目的で出力するときに使います。例えば日中韓(漢字圏)のフォントを出力するときに毎回コンパイルに時間がかかりすぎる場合に使用してください。

Don't use Production Names ブーリアン 通常Glyphsは内部のグリフ名のデータベースを元に、出力するフォントに格納されるグリフの名前をデザイン時のものから、技術的な要件を満たした製品用名に変更します。このカスタムパラメータが有効になっていると、グリフ名の変更処理が省略されます。ただしOS X 10.4のテキストエンジンなど、アプリケーションやシステムによってはAGL (Adobe Glyph List) に準拠したグリフ名

を想定したものがああります。これはフォント情報の「その他」で定義する「標準外のグリフ名を使用」と同じで、独自の命名規則をそのまま最終的なフォントに適用させたいユーザーに向けた機能です。

EditView Line Height 整数 編集ビューでのフォントの行の高さを指定します。制作中のフォントの縦メトリクスがイレギュラーであるか、その他の何らかの理由で初期状態の行間を変更したい場合に使用します。出力するフォントの情報には一切影響を与えません。

Export Glyphs 文字列リスト グリフ名のリストです。ここに挙げられたグリフは、フォントファイル内の「出力時に含める」設定を一切無視して必ず出力されるようになります。

Export Path 文字列 フォント出力先のフォルダの位置をPOSIX形式のファイルパスで明記してください。

Family Alignment Zones リスト ファミリー内でのオーバーシュートの差を抑え、低解像度下でより一貫性の高いフォントの表示結果を出すためのパラメータです。ファミリー内の最も重要なフォント（例えばRegular）のアラインメントゾーン（例えばベースライン、x-ハイト、キャップハイト）をここに記しておくとい良いでしょう。するとラスタイザは、各フォントのハイトとファミリー・アラインメントゾーンとの差が1ピクセル以内のときは、すべてのフォントの高さを揃えようと試みます。このメカニズムが機能するには、ファミリー・アラインメントゾーンが各マスターのアラインメントゾーンと数が一致しており、互換性がある必要があります。


上：ファミリー・アラインメント設定なし。
下：ファミリー・アラインメント設定あり。

family alignment.
family alignment.

familyName 文字列 ファミリー名です。フォント情報のファミリー名は無視され、こちらが優先されます（「7.1.1 ファミリー名」(p.XXX)を参照）。OpenTypeのnameテーブルID 1とID 4に対応します。ID 3、4、6を生成するのに使われます。

fileName 文字列 既に出力済みのフォントがあって、これから出力するフォントの情報を変更することなく、かつ出力済みのフォントの上書きも避けてフォント出力したい場合に、ファイル名をここで変更しておく機能です。あるいはフォントファイル内に同一名のインスタンスが複数ある場合、フォント出力時には片方が上書きされてしまいますが、ここで名前を指定しておくことで回避することができます。OTFファイルの、「.otf」拡張子を除く名前を入力します。

Filter 文字列 Glyphsのフィルタをフォント出力時に使用する機能で、コンポーネントグリフの分解後に実行されます。以下が使用可能なフィルタ値

です。手入力を必要としない便利な追加方法も用意されています。「フィルタ」メニューから実行するダイアログ内の左下には  アイコンがあり、そこからフィルタのプロパティ名と設定値をクリップボードへコピーできます。これを「インスタンス」タブのカスタムパラメータでペーストすると、プロパティと値を追加できます。

- 極点を追加：AddExtremes
- ハッチング：HatchOutlineFilter; OriginX:<x座標>; OriginY:<y座標>; StepWidth:<ステップ幅>; Angle:<角度>; Offset:<線幅>
- パスをオフセット：OffsetCurve;<横>;<縦>;<ストロークを作る>;<ポジション/自動>
- 重なったパスを合体：RemoveOverlap
- ラフ：Roughenizer; <セグメントのサイズ>; <横>; <縦>; <角度>
- 角丸：RoundCorner;<半径>;<視覚調整>
- スマート角丸：RoundedFont; <ステム値>
- 変形：Transformations; LSB:<左サイド>; RSB:<右サイド>; Width:<グリフ幅>; ScaleX:<横拡大率>; ScaleY:<縦拡大率>; Slant:<傾斜角>; SlantCorrection:<視覚的傾斜>; OffsetX:<横移動量>; OffsetY:<縦移動量>; Origin:<基準点>

Tip: 角丸フィルタで内向きの角に角丸を適用させたい場合は、<半径>を負の数にします。

<>で記された値の部分には、<>は取り除いて数字だけを入力してください。<ストロークを作る>、<視覚調整>、<視覚的傾斜>のブーリアン値には1（発動）または0（無効）を入力します。パスをオフセットの<ポジション/自動>には0.0から1.0までの小数を入力します（1.0が100%を意味します）。またはautoと入力すると自動ストロークになります。スマート角丸フィルタの<ステム値>は入力しなくても構いません。変形フィルタの<基準点>は0から4までの5つのオプションがあり、「フィルタ > 変形」の基準点の内容と一致しています。キャップハイト（0）、キャップハイトの½（1）、xハイト（2）、xハイトの½（3）、ベースライン（4）。また変形フィルタの左サイドベアリング、右サイドベアリング、グリフ幅は入力値の分だけ増減させたい場合は + または - を付け（例：LSB:+20; RSB:-10）、入力値にしたい場合は = を付けます（例：Width:=500）。

フィルタを特定のグリフにのみ適用させたい場合、末尾にinclude: またはexclude: を追加し、コンマ区切りまたはスペース区切りのグリフ名を入力します。

例：

```
RoundCorner;10;1;include:A,B,C (A B Cのみ角丸フィルタを適用)  
RemoveOverlap;exclude:a b c (a b c以外のすべてのグリフのパスを合体)
```

サードパーティー製フィルタのカスタムパラメータを使用したい場合、表記法についてはそれぞれの説明を参照してください。特にincludeとexcludeはサポートされていない場合があります。

埋め込みタイプはアプリケーションに対して使用範囲を提案するだけのもので、フォントの保護メカニズムではありません。fsTypeを無視するアプリケーションもあります。

fsType 数字リスト 埋め込みタイプを表すビット数のリストであり、OpenType OS/2 table fsType フィールドに対応しています。ビット数はOpenType OS/2仕様書に記載されています。「埋め込みタイプの表記です。フォントの埋め込みのライセンス権利を表します。埋め込み可能なフォントは書類の内部に保存することができます。目的のフォントがインストールされていないマシン（遠隔システム）でフォントが埋め込まれた書類を開いた場合、埋め込み機能を理解するシステムまたはアプリケーションによって一時的（または場合によっては恒久的）にフォントが読み込まれます。埋め込みライセンス権利の有無はフォントベンダーによって決定されます。

OpenType フォント埋め込みDLL仕様書とDLLリリースノートはOpenType フォントの埋め込みと読み込み機能をサポートするためのAPIについて説明しています。埋め込みDLLや他の方法を使ってフォント埋め込みをサポートするアプリケーションは、埋め込みを許可されていないフォントを埋め込んではいけません。さらに、フォントを一時的に読み込むアプリケーション（以下のプレビュー&印刷、編集用の埋め込みを参照）は、埋め込まれた書類が閉じられた時点で即座にフォントを削除しなければいけません。」

設定の候補は以下の通りです。

- Not set（未設定）：「未設定になっているフォントは、書類への埋め込みとアプリケーションを通して遠隔システムに恒久的にインストールすることが許可されます。遠隔システムのユーザーは、元のフォントの購入者と同じ権利、義務、そしてライセンスを得ることになり、同じユーザーライセンス契約、著作権、デザイン特許、そして商標を受け継ぎます。」
- Forbidden（禁止）：「埋め込みライセンス規制あり。この設定になっているフォントは法的な所有者の許可なしにはいかなる方法でも加工、埋め込み、受け渡ししてはいけません。注意：このオプションが機能するには、これ以外の埋め込みレベルが設定されてはいけません。」
- Preview & Print（表示と印刷のみ）：「この設定になっている場合はフォントの埋め込みが可能で、遠隔システムに一時的に読み込まれ、読み込み専用の書類の表示と印刷が可能になります。書類の編集はできません。」
- Editable（編集可）：「この設定になっている場合は、フォントは書類に埋め込み可能ですが、遠隔システムに読み込まれるのは一時的です。表示と印刷の設定とは違って、このフォントを埋め込まれた書類は編集でき、変更内容を保存することもできます。」
- Subsetting forbidden（サブセット禁止）：「この設定になっている場合は、フォントはサブセット（文字セットの分割）が禁止であり、埋め込み時には必ずフォント全体が含まれます。他の埋め込み制限も適用されます。」

GASP Table 設定 TrueType フォントの gasp テーブル (grid-fitting and scan-conversion procedure) を設定するためのパラメータです。望ましいレンダリング挙動が変化する2つの PPM の閾値を制御します。gasp テーブルはグレイスケールと ClearType サブピクセルレンダリング両方において推奨されるレンダリング情報を保有します。ただしレンダラーによってはこの情報を一切無視することがあります。「このテーブルは書体がグレイスケール表示可能なデバイスで表示される際に望ましいラスタライズのテクニックを記す情報を含んでいます。またこのテーブルは白黒2階調デバイスでとても大きい場合と小さい場合にパフォーマンスを改善するためにヒンティングを無効化するのにも使えます。」標準の閾値は8PPMと20PPMです。閾値は2つ設定するため3つのゾーンが存在し、それぞれに以下のプロパティを与えられます：

- ・ヒンティングなし&対称：最初の閾値に達するまではピクセルグリッドに合わせた変形は行われず、可能な場合アンチエイリアスが有効化されます。「グレイスケール表示可能なデバイスでは極小サイズのフォントはヒンティングを使わずグレイスケールのみで表示するのが最適です。」
- ・ヒンティング&非対称：2つの閾値の間のサイズでは、ラスタライザーはグレイスケール表示を抑え、ピクセルグリッドに合わせてアウトラインを変形するようになります。「中間のサイズでは、通常はヒンティングと白黒2階調の組み合わせのレンダリングが最も見た目が良くなります。」ClearType では縦方向にはアウトラインの変形が起きますが横方向にはサブピクセルレンダリングが適用されます。この縦横の挙動の違いが「非対称」と呼ばれる理由です。
- ・ヒンティング&対称：2つ目の閾値以降は、ラスタライザーはアウトラインの変形とグレイスケール表示の両方を行うようにプログラムされています。「大きいサイズでは、通常はヒンティングとグレイスケールの併用が最良な結果になります。」ClearType ラスタライザーは縦方向にはグレイスケールのアンチエイリアス、横方向にはサブピクセルレンダリングを適用します。これが対象と呼ばれる理由です。「デジタル画面上で大きいサイズで表示する場合、(略) Windows フォントレンダラーに追加された新機能によりスムーズで明瞭な結果が得られます」(Now Read this: The Microsoft Cleartype Font Collection, Microsoft, 2004, p.14)

glyphOrder 文字列リスト グリフの並び順です。Glyphs 内のグリフ ID 番号、出力されたフォントのグリフ ID の両方に反映されます。グリフ名を1行に1つつ記入して設定します。リストフィルタの内容をそのままコピー＆することもできます。ここにリストされていないグリフは末尾に Glyphs の標準の並び順で表示されます。

Grid Spacing 小数 出力するフォントのポイントの座標の精度をユニット数

で設定します。これはフォント情報の「グリッドのユニット間隔」を「グリッド細分」で割った値になります。0.0に設定すればインスタンスのポイントの座標が小数になり、精度が最大になります。0.01に設定すると精度を無制限ではなく1/100ユニットに制限できます。こうすることで無駄に長い小数座標を短縮することができます。このパラメータの主な目的はとも細かいフォントの補間において微細な差を残すため、整数のユニット間隔を避けることにあります。

Has WWS Names ブーリアン OS/2テーブルのfsSelectionの8ビット目の情報を設定します。OpenType仕様書によれば、このビットは「フォントがWWS（ウエイト/字幅/傾斜）の表記方式に則ったnameテーブルを内包しており、name ID 21および22を必要としないことを意味します。」このパラメータはウエイト/字幅/傾斜の順にフォントのスタイル名を設定している時のみ有効です。

hheaAscender 整数 アセンダー値です。OpenTypeのhheaテーブル(horizontal header)のAscenderフィールドに相当します。「タイポグラフィ的なアセンダー（ベースラインから一番高いアセンダーまでの高さ）の値です。」

hheaDescender 整数 ディセンダー値です。OpenTypeのhheaテーブルのDescenderフィールドに相当し、負の数で表記します。「タイポグラフィ的なディセンダー（ベースラインから一番低いディセンダーまでの高さ）の値です。」

hheaLineGap 整数 行間の値、つまりディセンダーと次行のアセンダーとの隙間の値です。OpenTypeのhheaテーブルのLineGapフィールドに相当します。「タイポグラフィ的な行間の値です。」

InterpolationWeightY 整数 縦の補間値です。これを設定すると各ポイントのX座標とY座標の補間値を意図的に違うものに設定することができ、パラメータにはY軸の補間値を入力します。これを発動させるには、インスタンスの補間値と違う値が入っていなければいけません。ただし斜め方向の線は形が崩れやすいので注意してください。InterpolationWeightYの値はウエイトの補間値からあまり差を大きくしないことをお勧めします。

例えば、ウエイト20と100のマスターがあるとして、ウエイト補間値50のインスタンスがあるケースで、水平方向の線が細すぎて見えなくなります。補間値60ではちょうどいいのですが、それでは縦線も同時に太ってしまい、縦線が太くなりすぎます。このような場合に補間値は50にしておき、InterpolationWeightYを60に設定すれば理想の結果が得られます。横方向（縦線の幅）は補間値50、縦方向（横線の幅）は補間値60として補間、出力されます。

isFixedPitch ブーリアン postテーブルのisFixedPitchフラグを設定します。これはフォントが等幅かどうかを明示するもので、ラスライザーはすべての文字の幅を同じピクセル数で表示するようになります。

CFFとpostテーブルではイタリック角度は90°から反時計回りで表記されます。Glyphsは入力値を自動的に解釈して最適な値を適用します。

italicAngle 整数または小数 イタリックの角度です。90°から反時計回りの傾斜角を設定します。このパラメータを設定するとフォント情報の「マスター」タブのイタリック角度の情報は無視され、こちらが優先されます。これが便利になる状況は、たとえば少し傾斜のあるローマン書体をデザインするときです。「マスター」タブで傾斜角を入力するとデザイン時に楽ですが、OS Xはイタリック角度が0でない場合は即座にイタリックであると判断するため、フォント出力時にこのカスタムパラメータを使って強制的に0に戻すことができます。このパラメータの存在はCFFのItalicAngle、postテーブルのitalicAngle、OS/2の上付き文字とした付き文字の横方向のオフセット値、そしてhheaテーブルのcaretSlopeRiseおよびcaretSlopeRunに影響を与えます。

Keep Glyphs リスト 出力するフォントに含めるグリフのリストです。ここに含まれないグリフはすべて削除され、カーニングとOpenTypeフィーチャーも自動で更新されます。Remove Glyphsカスタムパラメータと逆の働きをします。Webフォント用のサブセット版フォントを作る時に便利です。Remove GlyphsとKeep Glyphsは互いに排他的な関係にあり、共存はできません。

Keep Overlapping Components ブーリアン Ccedilla (Ç) コンポーネントを使ったグリフに複数のコンポーネント同士やコンポーネントとアウトラインとの重なりがあった場合には通常コンポーネントを分解しますが、このパラメータが有効になっているとスキップします。他アプリケーションで追加処理することを想定したTrueTypeフォントを出力する場合などに便利です。

license 文字列 ライセンス情報です。OpenTypeのnameテーブルID 13に相当します。「フォントが法的にどのような使用が許可されているかの記述、または具体的な合法使用例の記述です。このフィールドは法律的な言葉遣いを避け、分かりやすく書かれていなければいけません。」

licenseURL 文字列 ライセンスのURLです。OpenTypeのnameテーブルID14に相当します。「より詳細なライセンス情報が確認できるURLです。」必ず通信プロトコルから始まるようにしてください(典型的な例はhttp://)。

Link Metrics With First Master ブーリアン これが有効の場合、サイドベアリングとカーニングをすべてのマスターとカラーレイヤーにおいて最初のマスターの値に同期します。こうすることで、スペーシングとカーニング作業は最初のマスターのみに行うだけで済みます。各レイヤーの字幅が一致している必要のある複数レイヤー構成のカラーフォントをデザインする場合に便利です。

Local Interpolation 文字列 指定したグリフに異なる補間値を適用します。フォーマットは<ウエイト値>; <字幅値>; <カスタム値>; include: <コマ区切りのグリフ名>です。最初の3つの値は補間値で、最後は適用させるグリフ名のリストです。1軸のみ使用しているファイルでは、値は1つだ

けで十分です。例えば「120; include: a, g, s」とすると、a g sの3グリフにウエイト値120で補間し、他のグリフはインスタンスの設定値通りに補間されます。

localizedDesigner 文字列 英語以外のシステム言語を使用時に表示される非ASCII文字を使用したデザイナー名です (name ID 9)。値をダブルクリックして言語を選び、名前を入力します。複数の言語に対応させたい場合はその数だけlocalizedDesignerカスタムパラメータを追加します。

localizedFamilyName 文字列 英語以外のシステム言語を使用時に表示される非ASCII文字を使用したフォントファミリー名です。値をダブルクリックして言語を選び、名前を入力します。複数の言語に対応させたい場合はその数だけlocalizedFamilyNameカスタムパラメータを追加します。nameテーブルID1とID4用にエンコードされたローカライズ名を自動生成します。

localizedStyleMapFamilyName 文字列 英語以外のシステム言語を使用時に表示されるStyleMapFamilyNameです。詳細はそちらのカスタムパラメータの記述を参照してください。

localizedStyleName 文字列 英語以外のシステム言語を使用時に表示される非ASCII文字を使用したフォントスタイル名です。例えば英語のBoldをドイツ語ではFett、フランス語ではGrasにできます。

Apple Font Toolsは以下のWebサイトからダウンロードできます：
developer.apple.com/fonts

Make morx table ブーリアン AATテーブルのmorx (extended glyph metamorphosis)をフォント出力時に追加します。フォント情報の「フィーチャー」タブに「more」というフィーチャーを作りMIF (metamorphosis input file)形式のコードを書いてください。このカスタムパラメータを機能させるには、Apple Font Toolsのftxenhancerというコマンドラインツールがインストールされている必要があります。詳細はApple Font Toolsに付属の資料を参照してください。

AFDKOはこちらからダウンロードできます：
adobe.com/devnet/opentype/afdko

makeOTF Argument 文字列 セミコロン区切りのmakeotfターミナル引数です (例: -ns;-dcs;-ni') 使用できる引数の詳細についてはAFDKOのMakeOTFユーザーガイドを参照してください。

Master Background Color カラー マスターの背景色を設定します。当該マスターが編集ビューで選択された状態で、背景が標準の白ではなく指定した色になります。

Master Color カラー 複数レイヤーのカラーフォント用のカスタムパラメータです。詳細は「13.2 複数レイヤーフォント」(p.149)を参照してください。

Master Name 文字列 カスタムのマスター名です。Glyphsが標準で用意している名前の組み合わせを避けたいときに便利です。

Name Table Entry 文字列 OpenTypeのnameテーブルのカスタム表記です。表記法は以下の3つのいずれかです (<>は表記の一部ではないので取り除いてください)。

- ・ <名前ID>; <名前>
- ・ <名前ID> <プラットフォームID>; <名前>
- ・ <名前ID> <プラットフォームID> <エンコーディングID> <言語ID>;
<名前>

ここで使用できる値のリストは
OpenTypeのnameテーブルの仕
様を参照してください：

[www.microsoft.com/
typography/otspec/name.htm](http://www.microsoft.com/typography/otspec/name.htm)

プラットフォームIDの指定がない場合は3として扱われ、またエンコー
ディングIDがない場合は1 (Unicode)、言語IDがない場合は0x0049
(Windows English) として扱われます。プラットフォームIDが1と指
定された場合のみ、エンコーディングIDと言語IDは0 (Mac Romanと
Mac English) と解釈されます。

ネームIDは1 2 3 5 6以外でならどのような値でも構いません。プラッ
トフォームIDは1 (Mac) または3 (Windows) です。任意表記のエン
コーディングIDと言語IDは、プラットフォームID次第でWindowsまた
はMacのそれを表します。これらは0~65536までの値で、8進数、10進
数、16進数のいずれかで入力します。AFDKOの構文仕様によれば「10
進数の場合は0以外の数字から始まる必要があり、8進数は0で始まる表記、
16進数は0xで始まり、A-Fまたはa-fを使用」することが要求されます。

note 文字列 フォントに関する任意の内容のメモです。これは出力フォントに
含まれる情報ではなく、.glyphsファイルにのみ残ります。このカスタム
パラメータとフォント情報のメモ欄は繋がっています。

openTypeHheaAscender hheaAscenderと同様

openTypeHheaDescender hheaDescenderと同様

openTypeHheaLineGap hheaLineGapと同様

openTypeNameCompatibleFullName compatibleFullNameと同様

openTypeNameDescription descriptionと同様

openTypeNameLicense licenseと同様

openTypeNameLicenseURL licenseURLと同様

openTypeNamePreferredFamilyName preferredFamilyNameと同様

openTypeNamePreferredSubfamilyName

preferredSubfamilyNameと同様

openTypeNameSampleText sampleTextと同様

openTypeNameWWSFamilyName WWSFamilyNameと同様

openTypeNameWWSSubfamilyName WWSSubfamilyNameと同様

openTypeOS2Panose panoseと同様

openTypeOS2Type fsTypeと同様

openTypeOS2TypoAscender typoAscenderと同様

openTypeOS2TypoDescender typoDescenderと同様

openTypeOS2TypoLineGap typoLineGapと同様

openTypeOS2UnicodeRanges unicodeRangesと同様

openTypeOS2VendorID vendorIDと同様

openTypeOS2WeightClass weightClassと同様

openTypeOS2WidthClass widthClassと同様

openTypeOS2WinAscent winAscentと同様

openTypeOS2WinDescent winDescentと同様

openTypeVheaVertTypoAscender vheaVertTypoAscenderと同様

openTypeVheaVertTypoDescender vheaVertTypoDescenderと同様

openTypeVheaVertTypoLineGap vheaVertTypoLineGapと同様

Optical Size 文字列 最適サイズを定義するOpenTypeのsizeフィーチャーと、サイズを基にしてMacおよびWindows用にエンコードされたメニュー項目名を生成します。値にはセミコロン区切りの5つの項目が必要です。

- ・デザインサイズ：理想のフォントサイズ×10
- ・サブファミリー識別子：任意の整数。識別子が同じの別フォントは、アプリケーションがサポートしている場合は最適サイズのサブメニューにグループ化できます。
- ・許容サイズ最小値：フォントが使われる想定サイズの範囲の最小値を、デザインサイズの時と同様に10倍の値を入力します（値そのものは含まれず、値より大きい数字が許容範囲に含まれます）。
- ・許容サイズ最大値：フォントが使われる想定サイズの範囲の最小値をやはり10掛けで入力します（最大値は範囲に含まれます）。
- ・サイズのメニュー名：インスタンスのフォントメニュー名です（例：Display、Subhead、Small、Caption）。

例えば「120; 1; 79; 140; Twelve」と入力すれば、12ポイントを最も理想のフォントサイズとし、8ポイントから14ポイントまでを想定使用範囲とする「Twelve」という理想サイズ名のsizeフィーチャーを生成します。他に1をサブファミリー識別子とするフォントはグループとしてまとめられます（例：Twelve Italic）。

panose 数字リスト Panoseの設定をするための10個の自然数のリストで、値をクリックするとダイアログが現れて各カテゴリの設定をすることができます。これはOpenTypeのOS/2テーブルのPanoseフィールドに対応します。「この10バイトの数字列は書体の見た目を記述するために使われます。これで記述されたフォントの性格は、他の見た目の近いフォントと関連させるのに使われます。（中略）Panose値は現在Monotype Imagingの保有している『グレイ・ブック』に載っています。PANOSEでは各個16までのバリエーションを記述できる数字が10あります。Windowsはフォントマッパー内でファミリーのタイプを定義するためにbFamilyType、bSerifStyle、そしてbProportionを使用します。等幅フォントを定義する際にもbProportionが使われます。もしフォントが絵記号フォントである場合、PANOSEの最初のバイト（bFamilyType）は“pictorial”に設定されていなければいけません。」

postscriptBlueScale blueScaleと同様

postscriptBlueShift blueShiftと同様

postscriptFontName 文字列 Type 1/CFFのFontNameフィールドに使われる名前です。許可されている文字はASCII文字のみで、スペースの使用は禁止されています（例：MyFont-BoldCndIt）。

「FontNameの構成要素は通常ファミリー名（正確に言えばFamilyNameに使われる文字列）、その次にハイフン、そしてスタイル名をFullNameと同様の順番で記します。最初期のPostScriptインタプリタやOSのファイルシステムと互換性を保つため、AdobeはFontNameに使用できる文字数を29文字に制限しています。他のPostScript言語の名称と同様、FontNameにはスペースは使用できず、ASCII文字のみ使用できます。29文字制限に対応するために名称の省略が必要な場合、その省略名はファミリー全体で使用されなければいけません。」（Adobe Technote #5088）

Adobe Technote #5088 :
partners.adobe.com/public/
developer/en/font/5088.
FontNames.pdf

Adobeはスタイル名の省略表記に以下を推奨しています：Bd = Bold、Bk = Book、Blk = Black、Cm = Compressed、Cn = Condensed、Ct = Compact、Dm = Demi、DS = Display、Ex = Extended、Hv = Heavy、Ic = Inclined、It = Italic、Ks = Kursiv、Lt = Light、Md = Medium、Nd = Nord、Nr = Narrow、Obl = Oblique、Po = Poster、Rg = Regular、Sl = Slanted、Su = Super、Th = Thin、Up = Upright。さらに省略するためには以下の表記がおすすめです：Dm = Demi、Sm = Semi、Ult = Ultra、X = Extra

postscriptFullName 文字列 Type 1/CFFのFullNameフィールドに使われる名前です。これはユーザーが目にする理想のフォントの名前で、スペースの使用が許可されています（例：My Font Bold Condensed Italic）。

システムの中には「ファミリーのグループ化とフォントの整列のためにファミリー名とFullNameを比較」します。そのため、ファミリー名は「FullNameの当該部分と一致している必要があります、フォントメニューに表示されても問題ない内容である必要があります。あるデザインから派生したすべてのフォントは同じFamilyNameを共有する必要があります。（中略）FullNameはFamilyNameのコピーで始まり、スタイル名を追加して完成します。その順番は一般的に以下の通りです：ウエイト、字幅、傾斜（イタリックか否か）、最適サイズ」（Adobe Technote #5088）

postscriptIsFixedPitch isFixedPitchと同様

postscriptUnderlinePosition underlinePositionと同様

postscriptUnderlineThickness underlineThicknessと同様

Post Table Type 整数 インスタンスに組み込まれるpostテーブルのバージョンで、通常はTTFだと2、CFFだと3になります。

preferredFamilyName 文字列 推奨されるファミリー名です。これはOpenTypeのnameテーブルID 16に相当します。「歴史的な理由で、フォントファミリーは最大4つまでのスタイルでしか構成できませんでしたが（日本語版注：典型的にはレギュラー、イタリック、ボールド、ボールドイ

タリック)、デザイナーは4つより多いスタイルを1つのファミリーに含めたがるかもしれません。Preferred Familyを使えば4フォント構成より大きいファミリーの構築が可能になります。このIDは、ID 1の内容と違う場合にのみ必要です。」ID 1とはフォント情報で設定するファミリー名です。

preferredSubfamilyName 文字列 推奨されるサブファミリー名です。これはOpenTypeのnameテーブルID 17に相当します。「Preferred SubfamilyはID 2より更に詳細なサブファミリーの構成を可能にします。このIDは、ID 2の内容と違う場合にのみ必要で、Preferred Familyも必ず存在していなければいけません。」

PreFilter 文字列 Filterと同様の機能ですが、コンポーネントの分解やアウトラインの合成以前に実行されます。使用できるフィルタの内容に関してはFilterカスタムパラメータの説明を参照してください。

Preview Ascender 小数 マスター用のパラメータで、ベースラインからフォントプレビューの上端までの確保したい幅をユニット数で指定します。編集ビュー下部のプレビューパネルで文字の上部が収まっていない場合に使用します。初期値は1000です。

Preview Descender 小数 Preview Ascenderと同様の機能ですがディセンダーに作用し、プレビューパネルのディセンダー以下の幅を確保するために使います。このカスタムパラメータがない場合はフォント情報に入力するディセンダー値を使用しますが、winAscentカスタムパラメータの値があればそちらが優先されます。

Reencode Glyphs リスト コンマ区切りの「グリフ名=Unicode値」のリストを入力します(例:「smiley=E100, logo=E101」)。各グリフはフォント出力時に、その新しいUnicode値が割り振られます。もしUnicode値がすでに他のグリフに割り振られている場合は、カスタムパラメータにある方のグリフが優先され、元のグリフはUnicode値を失いますが、製品用名への変換は行われます。もしUnicode値が書かれていない場合、そのグリフはUnicode値を失うようになっています(例:「f_f_i=, f_f_j=」と記された場合、f_f_iとf_f_jのUnicode値は削除されます)。

Remove Classes リスト 値に記されているOpenTypeクラスを削除します。フォントのサブセット化が理由でグリフをRemove GlyphsやKeep Glyphsカスタムパラメータで削除する場合、手動で書かれたクラスは自動で更新されないで、それらのクラスを削除するのに便利です。自動生成されるクラスに関しては特に手動で指定しなくてもクラスから自動で削除されます。

Remove Features 文字列 値に記されているOpenTypeフィーチャーを削除します。グリフ名の拡張子のせいで不要なフィーチャーが自動生成されてしまう場合や、特定のインスタンスで特定のフィーチャーを除去したい場合に便利です。ちなみに自動生成されるフィーチャーに関しては、Remove GlyphsやKeep Glyphsカスタムパラメータが原因でフィーチャー内に関

与するはずのグリフがすべて消えている場合はフォント出力時に自動で削除されます。

Remove Glyphs 文字列 値に記されているグリフを出力するフォントから削除します。自動生成されたOpenTypeフィーチャーはグリフの削除を認識し自動更新します。例えばスモールキャップのグリフをすべて削除すると、smcpとc2scフィーチャーは自動的に削除されます。サブセットされたフォントを出力する場合に便利です。

Remove post names for webfonts ブーリアン Webフォントの出力用に、ファイルサイズ削減のためにグリフ名を削除します。

Rename Glyphs 文字列 値に記されているグリフの名称を変更します。コマ区切りの、「変更前=変更後」のグリフ名のリストを認識します（例：e.bold=e, g.alt=g）。同名のグリフがすでに存在する場合は名前が入れ替われます。その既存のグリフがコンポーネントとして使用されている場合は新しいグリフがまずコンポーネントとして配置されてから必要に応じてフィーチャーの書き換えやアウトライン分解が始まります。また「出力時に含める」オプションの状態も入れ替わりますので、これらを確実に制御するには、この現象を考慮して各グリフの出力状態をしっかりと確認するか、Export Glyphsカスタムパラメータを使用します。

Replace Class 文字列 OpenTypeクラスのコードを独自のコードに置き換えます。値の先頭には@なしのクラス名で、その次にセミコロン、そしてクラスのコードが続くようにします。当該のクラスが既に存在している場合にのみ実行されます。これは手動でクラスを設定するときのみ必要です。自動に設定されているクラスは特に何もしなくても自動で更新されます。

Replace Feature 文字列 OpenTypeフィーチャーのコードを独自のコードに置き換えます。値の先頭には4文字のフィーチャー名で、その次にセミコロン、そしてクラスのコードが続くようにします。当該のクラスがすでに存在している場合にのみ実行されます。これは手動でクラスを設定するときのみ必要です。

ROS 文字列 グリフのマッピング表 (cmap) のある特殊なフォントには Adobe,Identity,0のROS (Registry, Ordering, Supplement) を使えます。公開されている汎用のROSは以下の通りです：

Adobe-CNS1-6

Adobe-GB1-5

Adobe-Japan1-6

Adobe-Korea1-2

Adobe-Identity-0

Adobe-Identity-0を使用する場合、GSUBテーブルはフォント情報の「フィーチャー」タブの内容から生成されます。それ以外の場合はAdobeから提供されたGSUBリソースが使用されます。

sampleText 文字列 サンプルテキストです。OpenTypeのnameテーブル

ID 19に相当します。「これはフォント名でもいいですし、デザイナーがフォントの見本として最適だと考えるどんな文章でも結構です。」このサンプルテキストは例えばAppleのFont Bookでフォントが選択された時のサンプルとして表示されます。

Save as Truetype ブーリアン 有効になっている場合、フォント出力ダイアログの設定を無視して強制的にTrueTypeとして出力されます。

Scale to UPM 整数 入力されたUPM値に合わせてフォント全体を縮小します。例えば2048UPM（または16から16,364までの2の累乗数）のTrueTypeフォントを出力したい時、もしくはデザイン時のUPMが1000以外の場合に有効です。

shoulderHeight 整数 アラビア文字、ヘブライ文字、インド文字用の縦メトリクス値です。xハイトでないメトリクス線が必要な文字に新たな線を表示させるためのものです。

smallCapHeight 整数 スモールキャップの高さです。スモールキャップ用のアラインメントゾーン生成アルゴリズムは、この値を優先します。また編集ビューでスモールキャップのグリフを表示中はx-ハイトではなく、この値のハイトが表示されます。

styleMapFamilyName 文字列 ボールド、イタリック、ボールドイタリックとのスタイルマッピングに使われるためのファミリー名です。これを用いて大きなフォントファミリー内にサブファミリーを作ることができます。「4つまでのフォントが同じFont Family名を共有でき、スタイルリンクのグループを構成することができます。」Glyphsはフォント情報のインスタンスタブにあるスタイルリンクのスタイル名を使用して各スタイルをリンクさせます。

trademark 文字列 商標の記述です。OpenTypeのnameテーブルID 7に相当します。Microsoftによると、「ここにはフォントについてのあらゆる類の商標事項や情報が入ります。これらの情報は法律的なアドバイスを得た上で書かれるべきです。これは著作権とは明確に別のものです。」

TrueType Curve Error 小数 TrueType変換時のカーブのずれの許容値をユニットで指定します。特に指定がない場合は0.6です。この値が大きくなるほどエラーがより許容されるようになり、TrueTypeコンバーターはカーブの再構成により少ないポイントを使用するようになります。そのため二次ベジェ曲線データを保存するglyphテーブルが小さくなり、ファイルサイズの節約になります。つまりアウトライン品質とファイルサイズのバランスを調整するためのカスタムパラメータです。

TTFAutohint binary path 文字列 既にコンパイルされたTTFAutohintのバイナリーファイルのファイルパスです。アプリケーション標準のTTFAutohintではなく古いバージョンを使う必要があるときや、より新しいバージョンを使いたいときに便利です。

TTFAutohint control instructions 文字列 TTFAutohintのインストラ

クションを制御するために使われます。コントロールコードは別のアプリケーションで書いてパラメータに後でペーストするのをお勧めします。インストラクションの構文は以下のいずれかです：

- <グリフ名> left <ポイントID> <オフセット>
 - <グリフ名> right <ポイントID> <オフセット>
 - <グリフ名> nodir <ポイントID>
 - <グリフ名> touch <ポイントID> xshift <x> yshift <y> @ <PPM>
 - <グリフ名> point <ポイントID> xshift <x> yshift <y> @ <PPM>
- <オフセット>の値は任意表記で、省略した場合は0と見なされます。touchとpointインストラクションでは、片方のみ、または両方のシフトを指定できます。<x>と<y>の値は0.0~1.0の範囲内で設定します。<グリフ名>は複数のグリフ名も可で、デザイン用名ではなく製品用名でなければいけません。<PPM>は単体のピクセルサイズ、ハイフンで指定したサイズ範囲、またはコンマ区切りで指定したサイズのリストです。ハッシュ記号 (#)で始まる行はコメントと解釈され、実行されません。インストラクションの各単語は最初の1文字に省略しても受け付けられます（例：rightの代わりにr）。

ttfautohint についての詳細は以下のサイトを参照してください：

freetype.org/ttfautohint/doc/ttfautohint.html このカスタムパラメータでの引用はすべてttfautohintからのものです。

TTFAutohint Options 文字列 TrueType オートヒントングツール「ttfautohint」のコマンドラインオプションです。値をクリックして現れるダイアログで設定してください：

- Hint Set Range：インストラクションが最適化される PPM 範囲です。範囲を広げるとファイルサイズが肥大化する可能性があります。
- Default Script：「OpenType フィーチャーのための標準の文字体系です。」
- Fallback Script：「グリフの文字体系が自動で認識されない場合に使われる標準の文字体系です。」
- Hinting Limit：「ヒントングが解除される」PPM サイズです。標準は200ピクセルで、Hint Set Rangeの最大値より大きくなければいけません。Hint Set Range最大値からHinting Limitまでのヒントングは前者の最大値のものが使われ続けます。
- x-Height Increase Limit：この値以下6PPMまではxハイトが切り上げされます。標準値は14PPMです。「通常ttfautohintはxハイトをピクセルグリッドに合わせて変形しますが、概ね切り上げるようにします。（中略）小サイズの文字の視認性を向上させたい場合は値を上げてください。」値を0にするとxハイトの切り上げを無効化します。
- x-Height Snapping Exceptions：「xハイトのピクセルグリッドへの整列を解除したいPPMの値または値の範囲をコンマ区切りで収録したリストです。」例えば「8, 10-13, 16」だと、フォントのサイズが8、10、11、12、13、16ピクセルのときにxハイトの整列が起きないようになります。範囲を指定する際はハイフンを使います。

コントロールファイルの構文の詳細は、TTFAutohintの公式文書を参照してください：

freetype.org/ttfautohint

- Windows Compatibility : 「このオプションは擬似ブルーゾーンを usWinAscent と usWinDescent (OS/2 テーブルから認識) の2か所に挿入します。この目的は、Windows ではこの範囲外のすべてのアウトラインはカットされてしまうためヒンティングされたグリフをなるべくこの範囲内に収めることにあります。」もし Windows でアウトラインのカットが起こるようであれば使用してください。
- Pre-Hinting : 「与えられたフォントに始めから存在する TrueType ヒンティングが、オートヒンティング処理前に適用すべきかどうかの設定です。(中略) 日中韓のフォントなどでもし既存のヒンティングが出力解像度と関係なしにサブグリフを移動または拡大縮小してしまう場合に使用してください。」
- Hint Composites : 「初期状態ではコンポーネントグリフないのコンポーネントは別にヒンティングされます。このオプションが有効になっていると、コンポーネントグリフが直接ヒンティングされます(コンポーネントの元のグリフのヒントを継承しません)。このオプションを使うとバイトコードのサイズを肥大化させることになりませんがヒンティング結果が向上するかもしれません。」
- Symbol Font : 「サポートされている文字体系でサンプルとなるグリフがフォント内に存在しないために ttfautohint が通常ヒンティングを拒むであろうフォントを強制でヒンティングします。参照用グリフが見当たらない文字体系に関してはステム値を読み込めないため、ttfautohint はデフォルトのステム値を使います。記号、ディングバットフォントや数学記号用にこのオプションを使用しますが(たいていは Fallback Script と併用)、小サイズでのヒンティング品質はあまりよくないかもしれません。」
- Dehint : すべての TrueType ヒンティングを無効化し、他のすべてのオプションを無視します。テスト目的にのみ使用してください。
- Add Autohint Info : 「ttfautohint のバージョンとコマンドライン情報を、フォントのネームテーブルのバージョン情報 (ID 5) に付記します。」
- Strong Stems : ステムの表示結果を最適化させたいレンダリング環境を選択します。「なるべくステムが白黒はっきり表示させるため、ステムの幅と位置をなるべく整数に落とし込み強力にアウトラインを変形させます。」最適化できるレンダリング環境は以下の通りです: グレイスケール (Android)、GDI ClearType (Win XP などが採用する v.36 および v.37 のラスタイザー)、DirectWrite ClearType (IE9 以降や Win 7 以降で採用されている v.38 以降のラスタイザー)

TTFStems リスト TrueType 専用の水平ステムリストです。それぞれの水平ステムに対して名前と平均値を決めることができます。値を編集する際は、ダイアログシートが表示されます。✳メニューからステムを追加、削除ま

たはフォント情報のステム欄に入力したPostScript ステム情報を読み込むことができます。

typoAscender 整数 アセンダーの値です。OpenTypeのOS/2テーブルのsTypoAscenderフィールドに相当します。「フォントのタイポグラフィ的なアセンダーの値です。hheaテーブルのAscenderの値はAppleにより大きく異なる方式で定義されており、同じではないことに注意してください。(中略) sTypoAscenderの主な目的はunitsPerEmの値と共に使用され、タイポグラフィ的に正しい行間スペーシングを決定することです。この目的はMacintoshやWindowsなど各々のシステムに特化して後方互換性を確保しなければならないメトリクス情報の制限を解放し、最適な設定をアプリケーションに使わせることにあります。これらの新しいメトリクスと各文字のデザイン幅を組み合わせれば、どんなアプリケーションでも正しく、そして一貫性のある書類作成が可能になります。これらのメトリクスはWindowsのAPIからアクセスされます。MacintoshアプリケーションはOS/2テーブルからこのデータを抽出するためにsfntリソースにアクセスして解析する必要があります。

縦組み（と横組み）に使われるCJK（中日韓、漢字圏）フォントに関しては、sTypoAscender値はem-box（全角の正方形）の上端の高さである必要があります。例えば、もし漢字の全角が座標0, -120から1000, 880で出来るとすれば（つまり1000×1000ユニットの正方形が120ユニット下にはみ出ている場合は）、sTypoAscenderは880に設定されなければいけません。これに従わない場合、縦書き時のレイアウト結果が不正確になるでしょう。」

typoDescender 整数 ディセンダーの値です。OpenTypeのOS/2テーブルのsTypoDescenderフィールドに相当します。「フォントのタイポグラフィ的なディセンダーの値です。hheaテーブルのDescenderの値はAppleにより大きく異なる方式で定義されており、同じではないことに注意してください。欧文ベースの書体のsTypoDescender値を決定するのに適したソースはAFMファイルに記されているDescender値です。CJKフォントに関しては以下をご覧ください。

sTypoDescenderの主な目的はunitsPerEmの値と共に使用され、タイポグラフィ的に正しい行間スペーシングを決定することです。この目的はMacintoshやWindowsなど各々のシステムに特化して後方互換性を確保しなければならないメトリクス情報の制限を解放し、最適の設定をアプリケーションに使わせることにあります。これらの新しいメトリクスと各文字のデザイン幅を組み合わせれば、どんなアプリケーションでも正しく、そして一貫性のある書類作成が可能になります。これらのメトリクスはWindowsのAPIからアクセスされます。MacintoshアプリケーションはOS/2テーブルからこのデータを抽出するためにsfntリソースにアクセスして解析する必要があります（ただし現時点ではAppleはOS/2テー

ブルを参照していません)。

縦組み (と横組み) に使われるCJK (中国、日本、韓国) フォントに関しては、sTypoDescender 値はem-box (全角の正方形) の下端の高さである必要があります。例えば、もし漢字の全角が座標0, -120から1000, 880で出来るとすれば (つまり1000×1000ユニットの正方形が120ユニット下にはみ出ている場合は)、sTypoDescender は-120に設定されなければいけません。これに従わない場合、縦書き時のレイアウト結果が不正確になるでしょう。」

typoLineGap 整数 フォント製作者の推奨する行間の値です。OpenTypeのOS/2テーブルのsTypoLineGapフィールドに相当します。「フォントのタイポグラフィ的な行間の値です。hhea テーブルのLineGapの値はAppleにより大きく異なる方式で定義されており、同じではないことに注意してください。

sTypoLineGapの主な目的はunitsPerEmの値と共に使用され、タイポグラフィ的に正しい行間スペーシングを決定することです。典型的な値は平均してUPMの7~10%です。この目的はMacintoshやWindowsなど各々のシステムに特化して後方互換性を確保しなければならないメトリクス情報の制限を解放し、最適の設定をアプリケーションに使わせることにあります。これらの新しいメトリクスと各文字のデザイン幅を組み合わせれば、どんなアプリケーションでも正しく、そして一貫性のある書類作成が可能になります。」

underlinePosition 整数または小数 アンダーラインが挿入される高さの値で、ベースラインからアンダーラインの上端までの距離を負の数で入力します。Type1/CFF/post テーブルのUnderlinePositionフィールドに相当します。初期値は-100です。

underlineThickness 自然数 アンダーラインの太さの値です。Type1/CFF/post テーブルのUnderlineThicknessフィールドに相当します。初期値は50です。

unicodeRanges 文字列リスト フォントがサポートしているUnicode範囲のリストです。OpenType OS/2テーブルのulUnicodeRange1、ulUnicodeRange2、ulUnicodeRange3、ulUnicodeRange4に相当します。通常はGlyphsが自動で設定しますが、これを強制的に設定したい場合に使用します。Glyphsは検索フィールドを提供しており、製作中のフォントの正しいサポート範囲をすぐに見つけられるように設計されています。例えば欧文をすべてカバーしているという表記にしたい場合は、「latin」と検索して該当する検索結果にすべてチェックを入れてください。

uniqueID 自然数 PostScript 言語レベル1のUniqueIDです。PostScriptプライベートディクショナリーに保存される任意のIDで、0~16777215までの整数です。PostScript Type1仕様書より：「この第一の目的はフォントプログラムにより生成され、キャッシュされたビットマップに固有の

識別番号を振ることです。UniqueIDを与えることでPostScriptインタープリターが他のタスクのためにビットマップをキャッシュできるようになります。」これは既に廃止されていますが、CJKフォントではまだ使用されています。Adobeは、特にCJKフォント以外では使用を控えるよう促しています。このパラメータの効果を把握しており、必要としている場合だけ使用してください。

unitsPerEm 自然数 全角あたりのユニット数 (UPM) です。PostScript形式のOpenTypeフォントの初期値は1000で、TrueType形式のOpenTypeフォントの初期値は16~16,384の範囲内の2の累乗 (普通は2048) です。出力されるフォントはこの値のUPMに変換されます。小さい値を設定するとフォントは大きく表示されるようになり、真逆の現象も起きます。このカスタムパラメータはUPM値を変更するのみでフォントそのものを拡大縮小するものではありません。UPM変更と共に拡大縮小をしたい場合は、Scale to UPMカスタムパラメータを参照してください。

Update Features ブーリアン 自動更新がオンになっているOpenTypeフィーチャー強制的に更新します。フォントのエンジニアリングの段階などグリフセットが頻繁に変わる場合や、フィーチャーの自動追加を避けたいときに便利です。

Use Extension Kerning ブーリアン これが有効だと、kernフィーチャーにGSUBまたはGPOS拡張ルックアップタイプのルックアップが追加されフォントにより多くのカーニングペアを保存できるようになります。フォント出力時にoffset overflow errorが出た場合はカーニングペア、特に例外ペアが多すぎるため、通常はペア数を減らさなければいけないのですが、これをやりたくない場合にUse Extension Kerningを使います。最近のOfficeやデザインアプリケーションはすべて拡張カーニングをサポートしていますが、古いソフトウェアではカーニングデータが非互換になる可能性があります。

Use Line Breaks ブーリアン これが有効の場合、.glyphsファイルに保存されるOpenTypeフィーチャーコード内の改行はエスケープされません (\012で置き換えられます)。バージョンコントロールツールを使用したいときに便利です。

Use Typo Metrics ブーリアン これが有効だとフォントを使用するアプリケーション (特に2006以降のMS Office) はフォントの縦メトリクスにwinAscentとwinDescentの代わりにtypoAscender、typoDescender、typoLineGapを優先使用するようになります。初期状態では無効です。OS/2テーブルのfsSelectionフィールドのビット7 (don't use Win line metrics) に相当します。MakeOTFユーザーガイドによれば、このビットは「Microsoftがすべてのフォントの挙動を変更したときにドキュメントのリフローがなるべく減るように」追加されました。

vendorID 文字列 フォントのベンダーを表す4文字の識別コードです。Open-

Microsoftは登録済みのベンダーIDのリストを公開しています：
microsoft.com/typography/links/vendorlist.aspx

Type OS/2テーブルのachVendIDフィールドに相当します。もし未設定の場合、Glyphsは「UKWN」(unknown)をベンダーIDとして使用します。「フォントのベンダーを表す4文字の識別コードです。これはデザインの著作権者のことではありません。これはフォントをマーケティングまたは流通させる、いわゆる販売元のことです。たとえばITC Zapf Dingbatsが将来的に6つのベンダーから販売されることも考えられます（既にそうなっているかもしれませんが）。また各ベンダーが独自の利点を持っていることも考えられます（より豊富なカーニング、フォント別の丁寧なデザイン、手作業のヒントングなど）。これを使ってベンダーごとにフォントを識別し、より劣悪な可能性のあるフォントを避けるために使われます。ベンダーIDの入力は必須ではありません。」

versionString 文字列 バージョン番号です。「Version X.XXX」という書式で書き、Xには数字を入れます（後半は必ず3桁）。Glyphsは標準状態ではnameテーブルのName ID 5に、フォント情報に入力されているバージョンからバージョン番号を生成しますが、このカスタムパラメータはそれを上書きします。

vheaVertTypoAscender 整数 縦組み用のアセンダーの値です。OpenTypeのvheaテーブルのvertTypoAscenderフィールドに該当します。

vheaVertTypoDescender 整数 縦組み用のディセンダーの値です。OpenTypeのvheaテーブルのvertTypoDescenderフィールドに該当します。

vheaVertTypoLineGap 整数 縦組み用のアセンダーの値です。OpenTypeのvheaテーブルのvertTypoLineGapフィールドに該当します。

Webfont Formats リスト フォント出力ダイアログの設定内容に関わらず、ここに記載されているフォーマットのWebフォントが出力されます。1行につき「EOT」、「WOFF」、「WOFF2」のうち1フォーマットを記載します。

Webfont Only ブーリアン これが有効だと、デスクトップ用途に必要な情報をフォントファイルから削除します。こうすることで、ダウンロードされたWebフォントをデスクトップ用に変換してWindowsやMacにインストールし、ローカルに使用するのを難しくすることができます。このパラメータは、ブラウザ上では機能するものの厳密には破損したフォントを意図的に生成するためのものです。

weightClass 自然数 ウェイトクラスの値です。OpenType OS/2テーブルのusWeightClassフィールドに相当します。「見た目のウェイト（線の黒さ、または太さ）の記述です。」このパラメータが設定されている場合、フォント情報のインスタスタブの「ウェイト」ポップアップメニューの値は無視され、こちらが優先されます。

値	説明
100	Thin
200	Extra-light (Ultra-light)
300	Light

400	Normal (Regular)
500	Medium
600	Semi-bold (Demi-bold)
700	Bold
800	Extra-bold (Ultra-bold)
900	Black (Heavy)

アプリケーションの中にはこの値を使ってフォントメニュー内でサブファミリーを整理するものもあります。ただしこの数字のもたらす効果には大きな振り幅があります。Microsoftの古いレンダラーの中には、ウエイトクラスが250未満の書体を機械的に太くするバグを抱えているものがあります。また100の位しか参照せず、例えば400と425の区別を付けないエンジンもあります。ウエイトクラスを全く参照しない環境もあります。

widthClass 整数 字幅クラスの値です。値は1~9のいずれかです。OpenType OS/2テーブルのusWidthClassフィールドに相当します。

アプリケーションの中にはこの値を使ってフォントメニュー内でサブファミリーを整理するものもあります。これが入力されている場合、フォント情報のインスタスタブの「字幅」ポップアップメニューの値は無視され、こちらが優先されます。

値	説明	字幅のパーセンテージ
1	Ultra-condensed	50%
2	Extra-condensed	62.5%
3	Condensed	75%
4	Semi-condensed	87.5%
5	Medium (normal)	100%
6	Semi-expanded	112.5%
7	Expanded	125%
8	Extra-expanded	150%
9	Ultra-expanded	200%

「フォントデザイナーによって決められた、グリフの通常の比率（字幅と高さの比率）の相関的な変化の記述です。フォント内のすべての文字は違う比率を持ち得ますが、ノーマルな字幅のフォントの文字の字幅は1とします（訳注：100%）。字幅の違う新しいフォントスタイル（手作業か自動生成かに関わらず）が作られる場合、ノーマルなフォントと比べて字幅の比率は大きく、または小さくなります。このパラメータはそれを表現するためのものです。」

ANSIとはWindowsの8ビットエンコーディング1252のことで、西欧言語を主にカバーする文字セットです。詳細は以下のウェブサイトをご覧ください：

msdn.microsoft.com/en-us/library/cc195054.aspx

winAscent 自然数 Windowsで用いられるフォント内の最高点の値です。winAscentは大文字とそのアクセントが収まる高さでないとはいけません。OpenType OS/2テーブルのusWinAscentフィールドに相当します。

「usWinAscentは、Windows ANSI文字セットですべてのグリフのyMaxと解釈されます。usWinAscentはWindowsのフォントの高さと

行間の初期値を決定するために使われます。platform 3 encoding 0 フォントに関しては、yMaxと同じ値になります。Windowsはこの値より高い位置にあるアウトラインのビットマップをすべて非表示にします。」

winDescent 整数 Windowsで用いられるフォント内の最低点の値です。winDescentはg p q y等のディセンダを含めた高さでないとはいけません。OpenType OS/2テーブルのusWinDescentフィールドに相当します。

「usWinDescentは、Windows ANSI文字セットですべてのグリフの-yMinと解釈されます。usWinDescentはWindowsのフォントの高さと行間の初期値を決定するために使われます。platform 3 encoding 0 フォントに関しては、-yMinと同じ値になります。Windowsはこの値より低い位置にあるアウトラインのビットマップをすべて非表示にします。」

Write Kern Table ブーリアン フォントの出力時、OpenTypeのGPOSテーブルに加えて旧式のkernテーブルを追加します。グループカーニングは分解され、起こりうるすべての組み合わせが格納されます。そのため、テーブルのオーバーフローを回避するためにこのカスタムパラメータを使用する前にフォントをサブセット化する必要があるでしょう。

WWSFamilyName 文字列 WWSファミリー名です。WWSは「Weight Width Slope (ウエイト、字幅、傾斜角)」のことです。OpenTypeのネームテーブルID 21に相当します。「ID16とID17がWWSに従っていない場合に使われるWWS形式ファミリー名です(つまりID 17がウエイト、字幅、傾斜角以外の属性も使っている場合)。もしfsSelectionフィールドの8ビット目が設定されている場合はWWSFamilyNameは必要がなく、記述するべきではありません。逆に、もしこのIDが設定されている場合は8ビット目が設定されていないはいけません(詳細はOS/2 fsSelectionフィールドを確認してください)。(中略) ネームID 21の例: “Minion Pro Caption” と “Minion Pro Display” (この場合、ID 16は “Minion Pro” です)。」

WWSSubfamilyName 文字列 WWSサブファミリー名です。OpenTypeのネームテーブルID 22に相当します。「ID 21と併用される、ID 16と17がWWSに従っていない場合に使われるWWS形式サブファミリー名です(ウエイト、字幅、そして傾斜角しか記述しません)。ID 21と同様、fsSelectionフィールドの8ビット目と共存はできません。(中略) ネームID 22の例: “Semibold Italic”、“Bold Condensed” (ID 17には “Semibold Italic Caption” や “Bold Condensed Display” など、WWS以外の属性も記述可能です)。」

17.4 グリフの各種属性データ

17.4.1 グローバルなグリフデータの変更

Glyphsはグリフデータ(グリフの各種属性)をGlyphData.xmlファイルとして保持しており、これがデフォルト設定としてグローバルに機能し

ます。ユーザーはこのファイルを独自に書き替えることで、グリフの各種属性を変更できます。

オリジナルのGlyphData.xmlファイルはGlyphs.appのアプリケーションパッケージ内にあります (Glyphs.app/Contents/Frameworks/GlyphsCore.framework/Versions/A/Resources/GlyphData.xml)。各種属性を変更したい場合はこのGlyphData.xmlファイルを複製し、「Application Support/Glyphs/Info」フォルダーに置いて編集してください (InfoフォルダーにアクセスするにはGlyphsのメニューバー「スクリプト > Scriptsフォルダを開く」を選択し、表示されたScriptsフォルダの上の階層へ移動するのが簡単です。そこにInfoフォルダが存在しない場合は新規作成してください)。オリジナルのファイルを直接編集するのは避けてください。

オリジナルのGlyphData.xmlとInfoフォルダー内のGlyphData.xmlが存在する場合、GlyphsはまずオリジナルのGlyphData.xmlを読み込み、その後、Infoフォルダー内のGlyphData.xmlを読み込んで設定を上書きします。したがってオリジナルのGlyphData.xmlファイルは編集する必要がありません (いつでも初期設定に戻せるよう編集は避けるべきです)。

カスタムのGlyphData.xmlには、全グリフが記述されている必要はありません。ユーザーの変更したいグリフだけで十分です。書き替える予定のない部分は削除して構いません。書き替えた後はアプリケーションを再起動する必要があります。

XMLにはロングタグであるglyphData要素が1つ存在している必要があります。glyphDataの開始タグから終了タグの間に任意の数のglyph要素を記述します。glyph要素には最低でも以下の3つの属性が必要です。

- name : Glyphs内で使われるグリフ名
- category : グリフのカテゴリ名。フォントビューまたは「ウインドウ > グリフ情報」に表示されます。letterやdigitなどがその例です。3つ目の属性は以下のリストから任意のものを入力します (複数可)。
- accents : グリフのアンカー選択時などに現れる半透明の発音記号のリスト。複数ある場合はコンマ区切りで入力します。
- altNames : グリフ名の別名。Glyphsが旧バージョンで使っていた古いグリフ名などが入ります。メニューバー「グリフ > グリフ情報を更新」を実行すると、この別名がname属性のグリフ名に変換され、各種属性も更新されます。複数ある場合はコンマ区切りで入力します。
- anchors : 「グリフ > アンカーを自動設置」 (command+U) を実行したときに設置されるアンカーの名前のリスト。複数ある場合はコンマ区切りで入力します。
- decompose : 複数のコンポーネントで構成するグリフのレシピ。コンポーネントのグリフ名を正しい順番で入力します (例: 文字グリフ

- の後に発音記号グリフ)。複数ある場合はコンマ区切りで入力します。
- description：グリフの説明。通常はUnicodeの文字情報を使用しています。
 - production：出力されるフォントファイル内に格納される製品用のグリフ名。
 - script：グリフの属する文字体系。arabic、latin、greekなどです。
 - sortName：フォントビューでのグリフの並び順。例：d1はdのすぐ後に並びます。
 - sortNameKeep：sortNameと似ていますが、フォント情報の「オルタネートを元のグリフの横に並べる」が有効な場合に使われます。
 - subCategory：カテゴリの中のさらに細かい分類。例えばletterカテゴリにはuppercaseとlowercaseがあります。
 - unicode：Unicode値。4桁または5桁の16進数で入力します。

17.4.2 ファイルごとグリフデータの変更

グリフの属性を特定の.glyphsファイル内でのみ変更したい場合、複数の解決法があります。1つ目は.glyphsファイルと近い階層にGlyphData.xmlを配置する方法です。.glyphsファイルと同階層、.glyphsファイルと同階層に作られたInfoフォルダーの中、または.glyphsファイルの1つ上の階層に置いた場合に認識されます。このため複数の.glyphsファイルが1つのGlyphData.xmlを共有できます。他のデザイナーにファイルを送る際は、このxmlファイルも併せて送るようにしましょう。

2つ目は.glyphs内で特定のグリフのみを変更する方法です。フォントビューでグリフを選択し、「編集 > 選択内容のプロパティを編集」(option+command+I)を実行します。現れたダイアログで変更したいフィールドのチェックボックスを有効にして、データを変更します。グリフを1つだけ選択した場合、グリフ名、Unicode値、製品用名、文字体系、カテゴリ、サブカテゴリを変更できます。グリフを複数選択した場合、文字体系、カテゴリ、サブカテゴリを一括変更できます。



17.5 キーボードショートカットの変更

ツールバー上のショートカットはターミナルのコマンドで変更できます。Finderの「アプリケーション/ユーティリティ」フォルダ内の「ターミナル.app」を起動し、以下のように入力し、returnキーで実行します：

```
defaults write com.GeorgSeifert.Glyphs2 <ツール> "<キー>"
```

ショートカットをデフォルトに戻すには以下のように入力し、returnキーで実行します：

```
defaults delete com.GeorgSeifert.Glyphs2 <ツール>
```

変更内容は即座に反映されるので、Glyphsを再起動させる必要はありません。<キー>には使いたいキーを小文字で入力します。<ツール>には以下のいずれかを入力します：

- AnnotationTool.Hotkey (注釈ツール、標準：a)
- DrawTool.Hotkey (ペンツール、p)
- HandTool.Hotkey (手のひらツール、h)
- MeasurementTool.Hotkey (計測ツール、l)
- OtherPathsTool.Hotkey (ナイフ/消しゴムツール、e)
- PrimitivesTool.Hotkey (幾何図形ツール、f)
- SelectTool.Hotkey (選択ツール、v)
- SelectAllLayersTool.Hotkey (全レイヤー選択ツール、v)
- TextTool.Hotkey (テキストツール、t)
- RotateTool.Hotkey (回転ツール、r)
- ScaleTool.Hotkey (拡大縮小ツール、s)
- TrueTypeTool.Hotkey (TrueTypeヒンティングツール、i)
- ZoomTool.Hotkey (ズームツール、z)

hotkey IDはインターフェイスのツール名と一致しています。唯一の例外は1つにまとめられているナイフと消しゴムツールです。例：注釈ツールのショートカット（初期状態ではA）をQに変更したい場合、ターミナルに以下のように1行で入力し、returnキーで実行します（使用したいキーを小文字で入力してください）：

```
defaults write com.GeorgSeifert.Glyphs2 AnnotationTool.  
Hotkey "q"
```

使用したいキーを小文字で入力してください。returnキーで実行します。これでqを押すと注釈ツールに変更するようになりました。もし初期状態に戻すには、再びターミナルに以下のように1行で入力、再びreturnキーで実行します：

```
defaults delete com.GeorgSeifert.Glyphs2 AnnotationTool.  
Hotkey
```

これでショートカットはAに戻ります。