

**dtl FontMaster Konferenz  
Hamburg, 19. November 2002**

- . Einführung
- . Was ist OpenType?
- . Wer braucht OpenType?
- . Aufbau von OpenType Fonts
- . Open Type Layout Model
- . Status und offene Punkte
- . Open Type mit FontMaster

# 1.

- Vor** : Proprietäre Maschinenformate (Bitmap, Vector)
- : Ikarus Format  
(Offen, Hardware unabhängig, Database)
- Mitte** : Skalierbare FontFormate (Outline + Hints)
- URW (VS, BS, Nimbus)
  - T1 (Bezier Outlines und Hints)
  - F3, Bitstreams Speedo...
- Späte** : Entwicklung von TrueType bei Apple  
(Implementation Mac 1990, Windows 3.1 1991)

Offenlegung T1 Format (Adobe) ( 1-Byte)  
T0 FontFormat ( 2-Byte Fonts)

CID FontFormat für CJK (2-Byte)  
Brauchte etwa 5-6 Jahre auf dem Markt

TrueType GX (Advanced Layout features)  
Nicht erfolgreich

TTO (Multilingual Support , Layout features for Arabic)  
TTC (TrueType Collection Files for FarEast)

Sfnt-Wrapped CID Fonts (Adobe, MAC platform)

OpenType Specification (Adobe, MS, nicht Apple)

## Treibende Kräfte der Entwicklung :

Graphik, DTP (80er Jahre)

Globalisierung

- Neue Märkte Asien, Indien, Arabischer Sprachraum mit komplexen Schreibsystemen

Internationalisierung( ) und Lokalisierung( )

- Standardisierung (Unicode)
- elektronischer Dokumentenaustausch

# Konsequenz

Fonts werden immer komplizierter

Fonts erfordern mehr Aufwand bei der Herstellung

Fonts erlauben dafür aber

- exzellente Bildschirmdarstellung
- größere Zeichensätze
- typographische Feinheiten
- Multilinguale Zeichensätze

2.

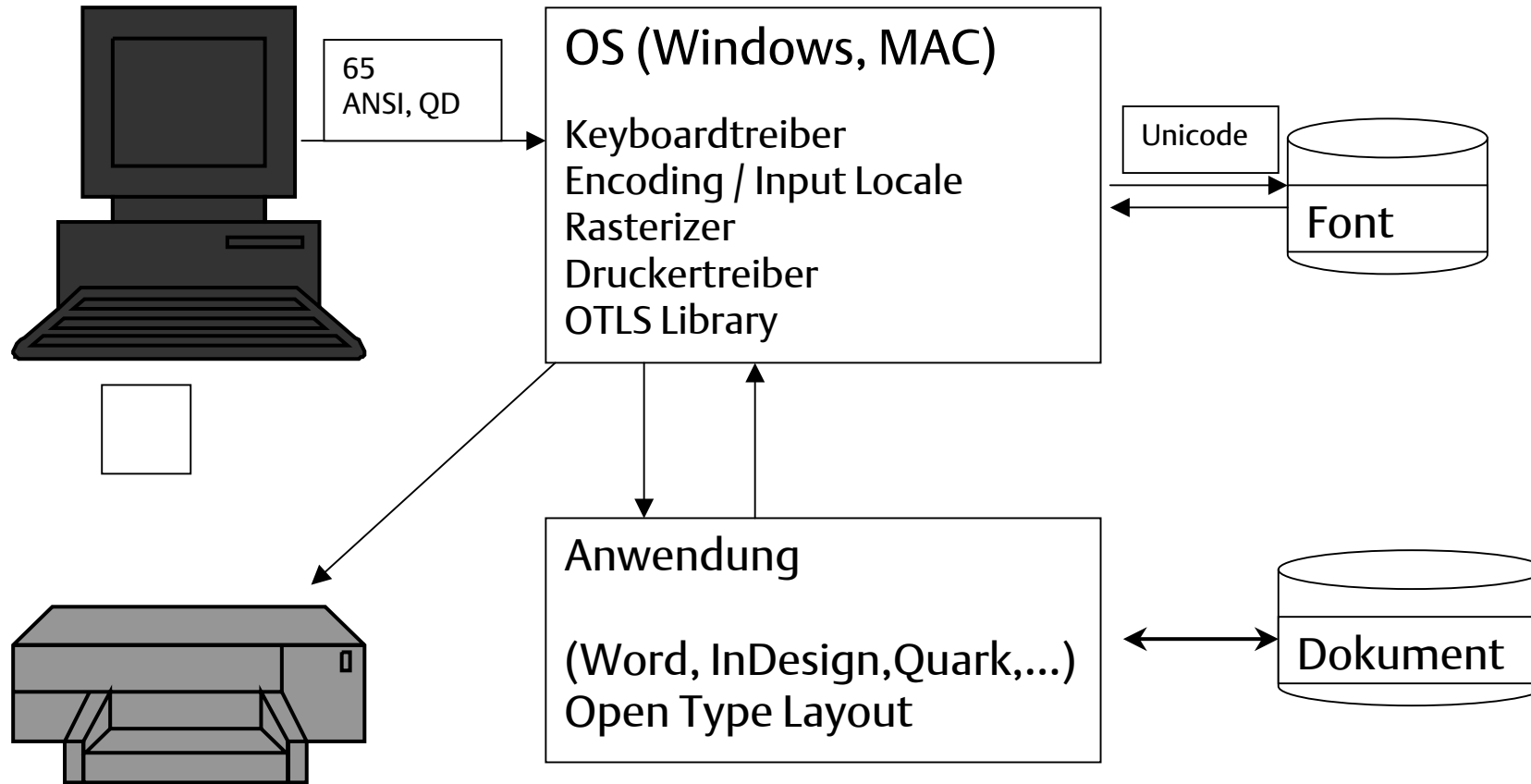
OpenType ist mehr als ein einfaches Fontformat, es ist eine  
Architektur

Open Type Fonts

Operating System

Anwendungen

Drucker, Belichter Unterstützung



# OpenType vs. TrueType/Type1

## OT Fonts...

sind plattformunabhängig

enthalten alle Daten in einem FontFile  
(Metrik, Kerning, Outline, Hints, Bitmaps...)

bieten einen erweiterten Zeichensatz für  
typographische Feinheiten

erlauben Unterstützung aller Sprachen und Schriftsysteme

sind nicht standardisiert (Anzahl Glyphs, Features...)

Type1  
24 Files  
504 Kbyte

Opentype  
1 File  
42 Kbyte

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

## OS Support für Opentype

OT wird auf allen Plattformen unterstützt (mehr oder weniger)

Windows 2000 / xp

Mac OS (9.2, x) eingeschränkt,  
keine Unterstützung der Features  
Eigene Technologie (AAT, ehemals GX)

Linux teilweise (Freetype, XFree86)

# Applikationen und Opentype

Adobe Applikationen (InDesign, PhotoShop, Illustrator...)

Office 2000/XP  
multilingualer Support  
Typographische Features

Quark XPress

### 3.

- . Millionen von Computernutzern in Asien (Indien, Tibet, Birma, Laos, Thailand..) mit komplexen Schriftsystemen
- . Millionen von arabisch schreibenden Nutzern
- . Milliarden von Chinesen, Japanern, Koreanern
- . Einige Zehntausend anspruchsvolle Typographen und Schriftdesigner in Europa und den USA

# 4.

OT Fonts haben eine 'SFNT-Tabellen-Struktur'

OT Fonts gibt es in zwei Varianten (.otf, .ttf)

Outline Beschreibung

- TT-Outlines und Instruktionen(Glyf table) (.ttf)
- Postscript Outlines und Hints (CFF table) (.otf)

Hinting Information (Hints, Instructions)

- TT Hinting überlegen für den Bildschirm

## Character/Glyph mapping tables

CMAP :	Zuordnung Glyphs <—> Unicode
HEAD,	HHEA: Schriftspezifische Informationen
NAME:	Schriftnamen, Copyright etc.
OS/2 :	Allgemeine Informationen
GASP :	Antialiasing
DSIG:	Digital Signature

## Features (for glyph substitution and positioning)

GSUB :	Glyph Substitution
GPOS:	Glyph Positionierung
BASE	Baseline Positioning
GDEF	Glyph Definition
JSTF	Justification

# Apples OS-X Systemfonts Hiragino Mincho OpenType Fonts with CFF outlines and AAT support tables:

tag	offset	length	checksum
<u>BASE</u>	364	456	6962C672
<u>CFE</u>	820	6720412	D234DEBC
<u>DSIG</u>	10240852	5788	EADEC4BC
<u>EBDT</u>	6721232	1636487	32BDCD3
<u>EBLC</u>	8357720	67148	883E371E
<u>GPOS</u>	8424868	14600	DD21703D
<u>GSUB</u>	8439468	185706	7F930AE3
<u>OS/2</u>	8625176	96	3814B65D
<u>VORG</u>	8625272	812	2BE8ACA
<b>Zapf</b>	<b>8626084</b>	<b>442236</b>	<b>2736C019</b>
cmap	9068320	276664	E31BA3BF
<b>feat</b>	<b>9344984</b>	<b>340</b>	<b>81CD4A53</b>
head	9345324	54	D3061EC9
hhea	9345380	36	8B5416B
hmtx	9345416	72546	D255AEAD
maxp	9417964	6	4F485000
<b>morx</b>	<b>9417972</b>	<b>739840</b>	<b>496DB24</b>
name	10157812	5060	3F369656
post	10162872	32	FFB80032
<b>prop</b>	<b>10162904</b>	<b>3758</b>	<b>DA5761FF</b>
vhea	10166664	36	74F5311
vmtx	10166700	74152	8EFBA4CC

	TrueType (TTF)	Apples TTF (AAT/GX)	OpenType (TTF)	Opentype (OTF)	SFNT-CID (Adobe)
Required	head, hhea, hmtx name OS/2 maxp post cmap	head, hhea, hmtx name OS/2 maxp post cmap	head, hhea, hmtx name OS/2 maxp post cmap <b>DSIG</b>	head, hhea, hmtx name OS/2 maxp post cmap	cmap name post
Outline	glyf, loca cvt, fpgm, prep	glyf, loca cvt, fpgm, prep	glyf, loca cvt, fpgm, prep	CFF	CID
Optional	gasp hdmx kern LTSH PCLT VDMX vhea vmtx	gasp hdmx kern vhea vmtx	gasp hdmx kern LTSH PCLT VDMX vhea vmtx	gasp kern vhea vmtx VORG	
Bitmap	EBDT EBLC EBSC	bdat bloc	EBDT EBLC EBSC		bdat bloc
OTF			BASE Baseline data GDEF Glyph definition GPOS Glyph positioning GSUB Glyph substitution JSTF Justification	BASE Baseline data GDEF Glyph definition GPOS Glyph positioning GSUB Glyph substitution JSTF Justification data	
AAT		mort, feat, bsln, prop opdb, trak, just.. fvar, gvar, Zapf...			feat mort
Adobe					ALMX BBOX FNAM, HFMX, VFMX

## The TrueType Font File (Apple's specification AAT)

<b>'acnt'</b>	(accent attachment) table	<b>'kern'</b>	(kerning) table
<b>'avar'</b>	(axis variation) table	<b>'lcar'</b>	(ligature caret) table
<b>'bdat'</b>	(bitmap data) table	<b>'loca'</b>	(glyph location) table
<b>'bhed'</b>	(bitmap font header) table	<b>'maxp'</b>	(maximum profile) table
<b>'bloc'</b>	(bitmap location) table	<b>'mort'</b>	(metamorphosis) table
<b>'bsln'</b>	(baseline) table	<b>'morx'</b>	(extended metamorphosis) table
<b>'cmap'</b>	(character code mapping) table	<b>'name'</b>	(name) table
<b>'cvar'</b>	(CVT variation) table	<b>'opbd'</b>	(optical bounds) table
<b>'cvt'</b>	(control value) table	<b>'OS/2'</b>	(compatibility) table
<b>'EBSC'</b>	(embedded bitmap scaling control) table	<b>'post'</b>	(glyph name PostScript compatibility) table
<b>'fdsc'</b>	(font descriptor) table	<b>'prep'</b>	(control value program) table
<b>'feat'</b>	(layout feature) table	<b>'prop'</b>	(properties) table
<b>'fmtx'</b>	(font metrics) table	<b>'trak'</b>	(tracking) table
<b>'fpgm'</b>	(font program) table	<b>'vhea'</b>	(vertical header) table
<b>'fvar'</b>	(font variation) table	<b>'vmtx'</b>	(vertical metrics) table
<b>'gasp'</b>	(grid-fitting and scan-conversion procedure)	<b>'Zapf'</b>	(glyph reference) table
<b>'glyf'</b>	(glyph outline) table		
<b>'gvar'</b>	(glyph variation) table		
<b>'hdmx'</b>	(horizontal device metrics) table		
<b>'head'</b>	(font header) table		
<b>'hhea'</b>	(horizontal header) table		
<b>'hmtx'</b>	(horizontal metrics) table		
<b>'hsty'</b>	(horizontal style) table		
<b>'just'</b>	(justification) table		

Wesentlich für OpenType sind

Unterschied Glyph  $\leftrightarrow$  Character

Existenz von Glyphen ohne direkten Unicode Codepoint

Feature Tabellen (GPOS, GSUB)

- Unbedingt nötig für komplexe Scripts (Indisch, Arabisch)

**Dies erlaubt:**

**Reichhaltigen Einsatz von alternativen Zeichen wie  
Ligaturen, positionsabhängigen Formen etc.**

**Zeichenpositionierung (vertikal und horizontal)**

**Zeichenverbindung (arabisch, indisch)**

**Änderung der Zeichenreihenfolge**

**Script- und Sprachspezifisches Verhalten der Applikation**

5.



## 2. Ebene: Sprachsysteme ( )

Es gibt einen Default-Regelsatz für alle Sprachen eines Scripts

Sprachspezifische Regeln können das Aussehen von Zeichen ändern:

Im Deutschen z.B das

Im Türkischen z.B. Versales I mit Punkt  
und Minuskel i ohne Punkt

Im arabischen gibt es unterschiedliche Glyphen für Farsi und Urdu

**Bestimmte Formen im serbischen differieren von russisch  
in kyrillischen Fonts :**

**Links: Standard Form  
Rechts: Serbische Form**

**(Beispiel aus: OpenType - Windows Glyph Processing, John Hudson, 2001)**

Ein weiteres Beispiel ist der CJK-Zeichensatz:

Script Tag:

Language Tag:

# Beispiel Arial Unicode: Scripts, Languages, Features:

### 3. Ebene: Features

Features sind typographische Regeln

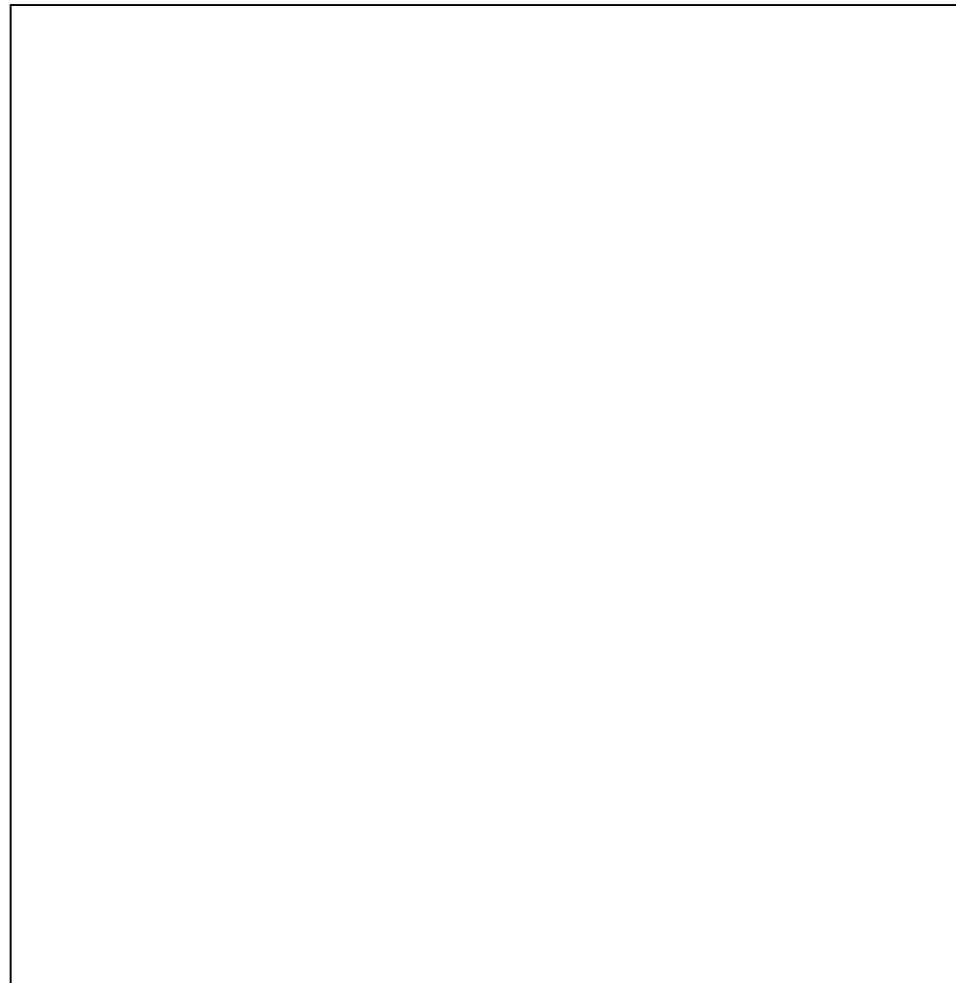
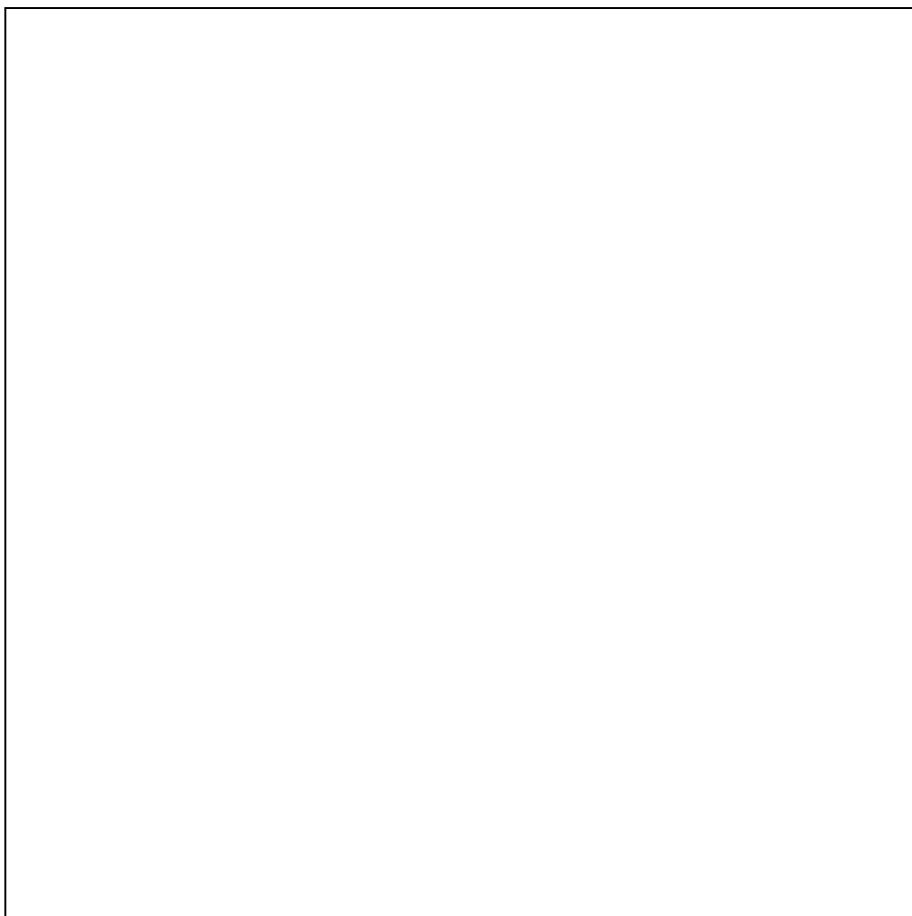
Features sind sprachabhängig

Features bestimmen, wie und welche Glyphen für eine bestimmte Sprache benutzt oder verändert werden.

Features sind registriert (Open Type Layout Tag Registry)

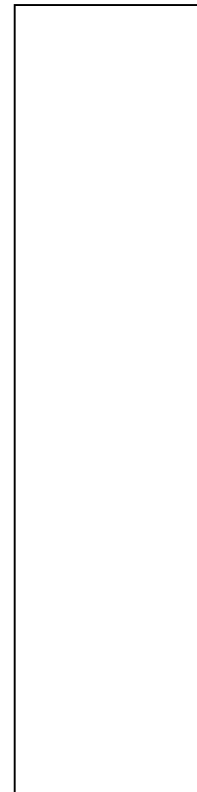
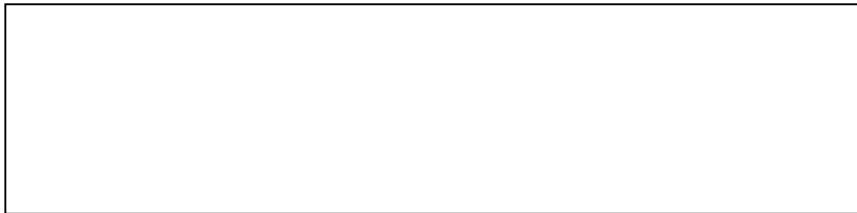
Features sind definierbar für Fontentwickler(!)

## Beispiel Kozuka Mincho :



## Features: Beispiele

substitutes vertical glyphs in  
Japanese (MS Mincho)



# Features: Beispiele

## (Arabische Sprachen)

## Features: Beispiele

Diese Features lassen sich gut auch für Schreibschriften nutzen, werden leider z.Z. noch nicht für lateinische Scripts unterstützt !

Beispiel: Schreibschrift Kolibri

Isolated, Initial and final forms, Swash alternates

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

## Features: Beispiele

: Ersetzt zwei oder mehrere Glyphs durch ein anderes

# Features: Beispiele

(discretionary ligatures)

# Features: Beispiele

## Kontextabhängige Ligaturen

Ersetzt f und t nur dann durch eine Ligatur, wenn kein Glyph mit Unterlänge vorhergeht.

# Features: Beispiele

# Features: Beispiele

## (historische japanische Formen)

Features: Beispiele

(alternative Zeichen)

*DTL Elzevir Book Italic*

*A B C D E F G H I K*

*A B C D E F G H I K*

*a b c d e h i k l m n p u z*

*a b c d e h i k l m n p u z*

## GPOS Features:

### Beispiel:

Kerning ist ein GPOS Feature, es modifiziert die Position

In OpenType Fonts kann Kerning sehr umfangreich werden:

Horizontales Kerning (keine Begrenzung der Anzahl Paare)

Vertikales Kerning

Crossstream Kerning (in Y-Richtung bei horizontalem Satz)

Klassenbasiertes Kerning

Punktgrößen abhängige Kernwerte zur Verbesserung der  
Bildschirmdarstellung

## Weitere GPOS Features:

Capital Spacing (Alternative Dicken für Versalsatz)

Proportional Alternates (für jap. Vollgeviertzeichen)

Optical Bounds (Verschiebung am Zeilenanfang/Ende !)

Vertical Alternate Metric  
(Neupositionierung lateinischer Zeichen bei vert. Satz)

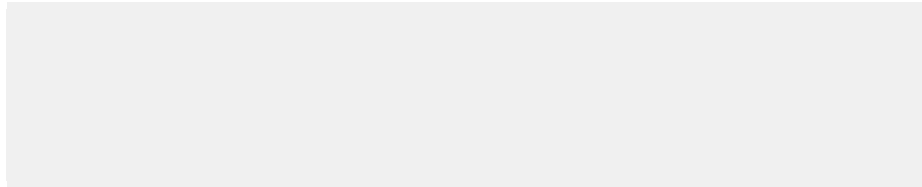
Beispiel:

sensitive Forms

Positionierung einiger Zeichen im Versalsatz:

**Beispiel:**

**Urdu erfordert neben der Kontrolle der Position auch die kontextabhängige Glyphsubstitution**



## 4. Ebene:

Lookups enthalten eine Liste von Glyphs, die durch eine Transformation modifiziert werden, die Art dieser Transformation und das erwünschte Ergebnis, d.h. ein oder mehrere Glyphs.

## GSUB Table

Ersetzt ein Zeichen durch ein anderes (vert, salt, ...)

Ersetzt ein Zeichen durch mehrere  
(ligature decomposition)

Ersetzt ein Zeichen durch eins von vielen (crcy)

Ersetzt mehrere Zeichen durch eins (liga, dlig,...)

Ersetzt ein oder mehrere kontextabhängig (clig...)

Ersetzt ein oder mehrere Zeichen im  
mehrfachen Kontext (Swash alternates)

# GPOS

GPOS Lookups beschreiben die Positionierung von Zeichen

Relativ zu ihrem eigenen Ursprung

Durch die Verbindung von Zeichen an vordefinierten Punkten

Es gibt acht verschiedene Lookups:

Single Adjustment

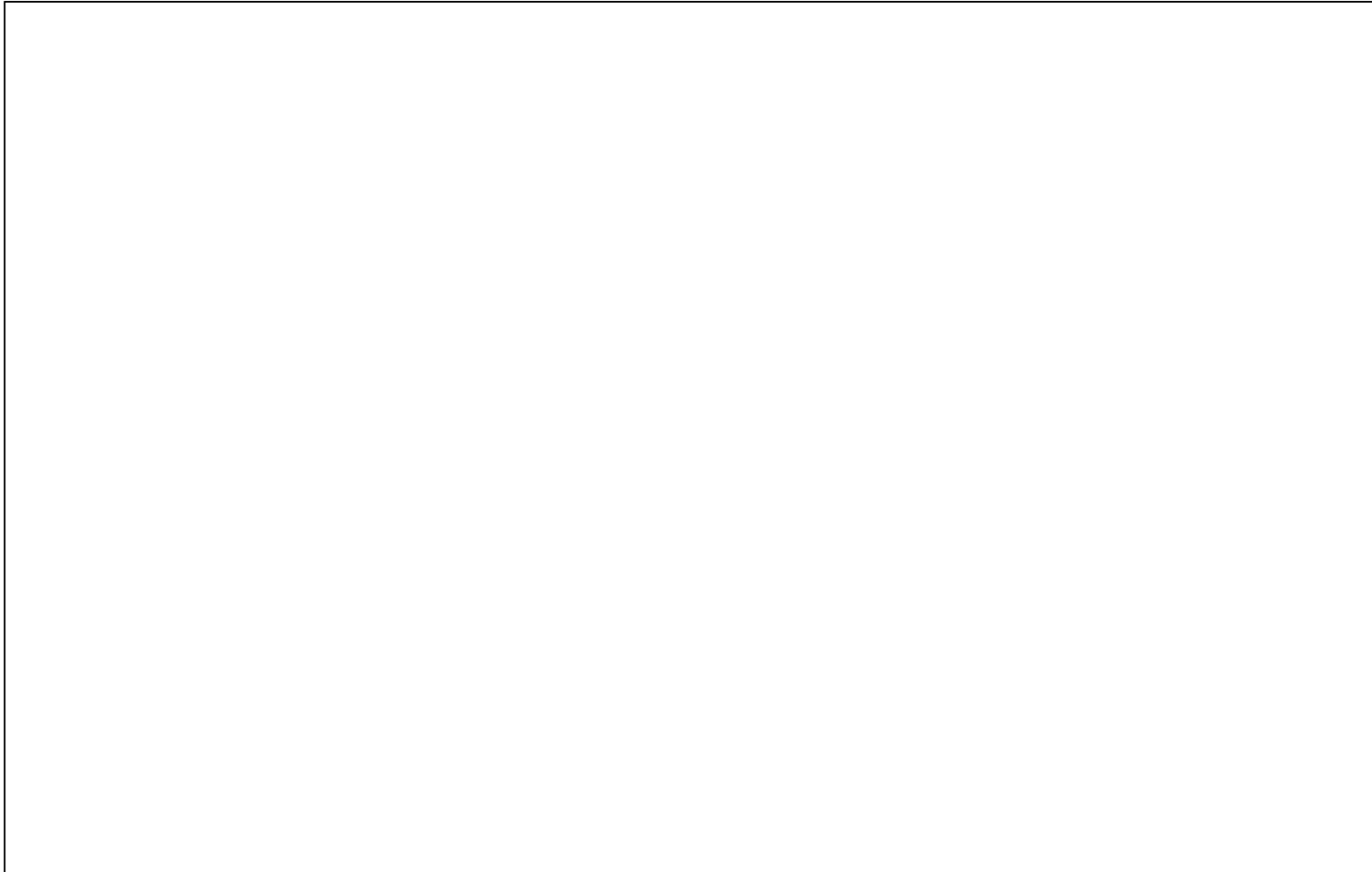
Pair Adjustment

Mark to Base Adjustment

...

**Beispiel:**

**Kontextabhängige Akzentpositionierung nach  
einem Versalbuchstaben mit Überhang (W,V,...)**



**Beispiel:**

**Verschiedene Lookups aus MS Mangal (Devanagari)**

## Verarbeitung von Features und Lookups:

Die Verarbeitung macht die Applikation  
entweder selbst oder über OS-Unterstützung (OTLS)

Die Reihenfolge der Lookups ist wichtig

GSUB Lookups werden immer vor GPOS Lookups ausgeführt

Der Font muss die Lookups in der richtigen Reihenfolge  
enthalten

## Beispiel: Reihenfolge

Ligaturen TA, AE. AE ist die erste Ligatur in der Liste:

Ligature ffi, fi: ffi muss zuerst gelistet werden:

# Beispiel: Sprachabhängigkeit

fi, ffi Ligaturen gibt es nicht im Türkischen :

```
feature liga {  
    sub f f i by ffi;  
    sub f i by fi;  
    lookup NOFI {  
        sub f f l by ffl;  
        sub f f by ff;  
        sub f l by fl;  
        sub f f j by f_f_j;  
        sub f j by f_j;  
    } NOFI;  
    language TUR excludeDFLT;  
    lookup NOFI;  
} liga;
```

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

## 6. Status und offene Punkte

InDesign hat die meisten Features plattformunabhängig implementiert, andere Adobe Applikationen werden folgen.

MS Office unterstützt noch keine lateinischen Features.  
Ändert sich (vielleicht) mit der nächsten Version.  
Daraus resultieren Probleme beim Übertragen von Dokumenten:

InDesign:

Cut + Paste in Word:

Das Feature 'Kapitälchen' wird elektronisch umgesetzt:

EIN BEISPIEL AUS DER  
FLEISCHMANN

InDesign:

Cut + Paste in Word:

Es wird der zu Grunde liegende Unicode String übertragen:

Ligaturen:

spezial, traffic

- ↪ **Apple unterstützt in OS X OpenType nur eingeschränkt AAT ist implementiert, aber ohne Applikationen und Fonts.**
- ↪ **Quark Xpress ?**
- ↪ **Es gibt noch Detailprobleme wie Kerning (Kern vs GPOS)**

## 7.

- Die OpenType Produktion basiert auf Adobe's SDK
- FontMaster erzeugt so viele Features wie möglich automatisch
- Fortgeschrittene Benutzer können eigene Feature Files benutzen
- Es gibt keine graphische Bedienoberfläche
- FontMaster läuft auf PC und MAC

Für die OpenType Produktion werden zwei Files benötigt:

- Layout file (Zuordnung Nummer, Unicode, PS name)
- Feature file (Glyph Substitution, Kerning,...)

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

The layout File determines the character set and relates the character data base numbers (BE-numbers) to Postscript names.

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

```

C urwotf.cha
Version 002.000
Starttable
URWNum;urwcOMP;urwcOMP;UNINum;ANNum;QDNum;PSNum;PSName;KernClass;StatClass
101;;;x0041;65;65;65;A;V;
102;;;x0042;66;66;66;B;V;
103;;;x0043;67;67;67;C;V;1
104;;;x0044;68;68;68;D;V;1
105;;;x0045;69;69;69;E;V;1
106;;;x0046;70;70;70;F;V;
107;;;x0047;71;71;71;G;V;
....
201;101;701;x00C4;196;128;;Adieresis;V;
202;101;704;x00C1;193;231;;Acute;V;
203;101;705;x00C0;192;203;;Agrave;V;
204;101;706;x00C2;194;229;;Acircumflex;V;
....
301;;;x0061;97;97;97;a;M;
302;;;x0062;98;98;98;b;M;
...
336;;;xEA00;;;ff;;
337;;;xEA01;;;fi;;
338;;;xEA02;;;fl;;
341;;;xEA03;;;ffi;;
342;;;xEA04;;;ffl;;
399;;;x207D;;;parenleftsuperior;;
....
543;;;;;;oneoldstyle;;
544;;;;;;twooldstyle;;
...
4308;;;;;;Hsmall;;
....
4309;;;;;;Ismall;;
Endtable

```

## Adobe feature File

```

feature smcp {
  sub @LETTERS_LC by @LETTERS_SC;
  sub @PUNCT_DEFAULT by @PUNCT_SC;
}

```

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

```

    sub @FIG_TAB_LINING by @FIG_TAB_OLDSTYLE;
    sub @FIG_FIT_LINING by @FIG_FIT_OLDSTYLE;
    sub @CURRENCY_LINING by @CURRENCY_TAB_OLDSTYLE;
    sub @ACCENTS_LC by @ACCENTS_SC;
    sub [numbersign percent perthousand] by [numbersign.taboldstyle percent.oldstyle perthousand.oldstyle];
} smcp;

```

```

feature crcy {
  script latn;
  lookup ALL_CRCY {
    substitute \5 from [\61 \102 \103 \107];
    substitute \61 from [\5 \102 \103 \107];
    substitute \102 from [\5 \61 \103 \107];
    substitute \103 from [\5 \61 \102 \107];
    substitute \107 from [\5 \61 \102 \103];
    substitute \235 from [\291 \608 \609];
    substitute \291 from [\235 \608 \609];
    substitute \608 from [\235 \291 \609];
    .....
    substitute \714 from [\711 \712 \713];
  } ALL_CRCY;

```

```

  script kana;
  lookup ALL_CRCY;
  script hani;
  lookup ALL_CRCY;
  script cyrl;
  lookup ALL_CRCY;
  script grek;
  lookup ALL_CRCY;

```

```

} crcy;

```

## List of currently supported features:

```
# --- FEATURE SUMMARY
# --- Substitution
#   aalt: access all alternates
#   smcp: lc to smallcaps
#   c2sc: caps to smallcaps
#   case: uppercase, math and punct shift up
#   titl: replaces default glyphs with .titling variations
#   onum: changes to oldstyle figures
#   lnum: changes to lining figures
#   pnum: changes to proportional figures
#   tnum: changes to tabular figures
#   crcy: replaces any currency char with any other
#   sups: makes numbers superior
#   sinf: makes numbers inferior
#   numr: replaces selected figures with numerators
#   dnom: replaces selected figures with denominators
#   frac: substitutes the existing fractions
#   dpng: diphthong subs
#   liga: standard lig replacement
#   salt: replaces glyphs with alternate variations
#   dlig: discretionary ligatures
#   ordn: subs ordinal glyphs after numbers
#   ornm: allows access to ornaments
#   zero: slashed zero
#
# --- Positioning
#   kern: pairs in simple and class form
#   csp: wider spacing for caps in all-cap setting
```

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

**Dr. Jürgen Willrodt**  
*Open Type Font Technologie*

## Zusammenfassung:

OpenType wird sich weiter durchsetzen.

2002 ist noch keine komplette Unterstützung vorhanden.

Die Vorteile von OpenType überwiegen schon heute.

FontMaster ist ein ideales Tool um OT Fonts zu erzeugen.