

Latin Modern Nederlands

De Latin Modern-fonts zijn inmiddels geen onbekende meer en zijn al aardig geïntegreerd in de $\text{T}_{\text{E}}\text{X}$ -distributies. Deze fonts zijn de uitkomst van het Latin Modern-project dat door de $\text{T}_{\text{E}}\text{X}$ gebruikersgroepen, waaronder de NTG, wordt gefinancierd. De doelen van dit project laten zich als volgt samenvatten:

- het combineren van de tot dan toe gebruikte $\text{T}_{\text{E}}\text{X}$ -pe1-fonts in slechts een font (voor $\text{pdf}_{\text{L}}\text{M}_{\text{TeX}}$)
- het voorzien in de behoefte aan een OpenType font (voor $\text{X}_{\text{L}}\text{TeX}$ en $\text{Lua}_{\text{L}}\text{TeX}$)
- het opschonen van de shapes en corrigeren van bekende bugs
- het aanvullen van het repertoire aan Latin glyphs

De fontset vervangt in $\text{T}_{\text{E}}\text{X}$ -distributies de traditionele Computer Modern-fonts en biedt daarbij wat extra monospaced varianten. Over de jaren heen waren nogal wat aanvullingen ontstaan op CM en geen ervan was compleet zodat schrijvers van macropakketten zich nog wel eens van rommelige constructies moesten bedienen om de gewenste karakters in de uitvoer te krijgen.

Aansluitend is het $\text{T}_{\text{E}}\text{X}$ Gyre-project gestart, dat dezelfde doelen verwezenlijkt voor tien veel gebruikte fonts in de $\text{T}_{\text{E}}\text{X}$ -distributies. In de wandelgangen wordt daarbij vaak gesproken over het ‘lm-ineren’ van fonts, wat dan zowel duidt op het synchroniseren van de beschikbare glyphs als het verminderen van het aantal fontfiles. Wat geldt voor LM, dat geldt ook voor Gyre.

Het verder opwerken en completeren van de standaard LM-fonts resulteert in regelmatige updates van de resulterende files. Eind januari is versie 1.05 uitgebracht. In deze versie zijn weer wat extra glyphs beschikbaar (vooral ten behoeve van transliteraties) maar de meest drastische wijzigingen zitten in de OpenType varianten:

- nieuwe filenamen zodat in andere applicaties dan $\text{T}_{\text{E}}\text{X}$ -gebruik handiger is
- een meer robuuste ondersteuning van features zoals ligaturen
- een andere systematiek van interne naamgeving

Er is door een groep betrokkenen zo goed en kwaad als dat kan gekeken in hoeverre de verschillende besturingssystemen en grafische applicaties fonts presenteren en de huidige interne naamgeving is

het best mogelijke compromis. Natuurlijk kunnen $\text{T}_{\text{E}}\text{X}$ -macropakketten er hun eigen laag overheen leggen en veel voor de gebruiker verbergen.

Er is verrassend veel tijd gaan zitten in een voor de Nederlandstaligen specifiek probleem: de ij. Om een of andere reden vinden we het vanzelfsprekend als we in een font een fi, fl, ffi en ffi ligatuur aantreffen. Overigens, zulke ligaturen zijn vooral bedoeld om dichtlopen door inkt te voorkomen en niet zozeer esthetisch. Als gevolg van meer moderne druktechnieken vinden we in hedendaagse fonts minder vaak zulke ligaturen. Echter, wat te denken van onze zogenaamde ‘lange ij’. Op school leren we deze combinatie van een i en een j als een karakter te schijven en het zal dan ook niemand verbazen dat er in Unicode twee slots zijn gereserveerd: 0x0132 voor IJ en 0x0133 voor ij. Zowel in Unicode als in de glyphname worden dit ligaturen genoemd, maar de meeste Nederlandstaligen zullen het als een letter zien. Het is de vraag of bij het opdreunen van het alfabet op school kinderen denken aan deze ij of aan de zogenaamde griekse y.

Hoe dan ook, met de komst van OpenType zullen we de ligatuur vaker zien verschijnen. De reden hiervoor is niet dat plotseling iedereen het gereserveerde Unicode slot gaat gebruiken (we blijven gewoon i+j intypen), maar dat fonts ingebouwde regels hebben voor het vervangen van combinaties van karakters. Dit is niet nieuw, denk maar aan de fi ligatuur die opduikt als we een f en een i intypen. Echter, waar deze ligatuur in vrijwel alle talen opduikt, is de ij ligatuur alleen in het Nederlands van belang. Dit betekent dat in bijvoorbeeld Latin Modern, waar we een ij-ligatuur hebben, de OpenType-fonts extra slim moeten zijn als het onze taal betreft. Vergelijk de voorbeelden in figuur 1 maar eens.

In een OpenType font zijn features gekoppeld aan een script (zoals latin) en een taal (zoals Nederlands). Beiden hebben standaard de waarde ‘default’. Een van de beslissingen die een fontontwerper (of degene die het font technisch afhandelt) moet nemen is waar precies zo’n ligatuur moet worden ondergebracht: is het een ‘required’ ligatuur (rlig), een ‘discretionary’ ligature (dlig), een gewone ligatuur (liga) of misschien een ‘historic’ (hlig)? De praktijk leert dat fonts op dit vlak niet consistent zijn, iets wat niet verwonderlijk is, omdat de OpenType-standaard minder open is dan de naam suggereert. We zullen er mee moeten leren leven dat weliswaar veel font-gerelateerde zaken (om er een

script	taal	liga feature	resultaat
dflt	dflt	disabled	fijn ijs eten in de file
dflt	dflt	enabled	fijn ijs eten in de file
latn	dflt	enabled	fijn ijs eten in de file
latn	nld	enabled	fijn ijs eten in de file
latn	eng	enabled	fijn ijs eten in de file

Figuur 1.

te noemen: installatie) eenvoudiger worden maar dat de eindgebruiker nog steeds voor beslissingen wordt gesteld.

In Latin Modern is onze ij beschikbaar als ligatuur waarvan de i en de j wat dichterbij elkaar staan dan normaal. Hier zou dus ook taal- en context-afhankelijke kerning kunnen worden gebruikt.

Wat is dat, een context-afhankelijkheid? Ook in het geval van een ligatuur moeten we bij vervanging van de i en j door een ij kijken naar meer dan alleen deze twee letters. Immers, wat verwacht men in het woord *fijn*? Een *fi+j+n*, of een *f+ij+n*. In Latin Modern zijn regels opgenomen die inhouden dat in geval van een *f* gevolgd door een *i*, gevolgd door een *j* eerst de laatste twee karakters worden vervangen door een *ij*. Vervolgens hebben we een *f* gevolgd door een *ij* en dat is geen paar dat tot een ligatuur leidt. Dit soort truukjes kan alleen door $X_{\text{e}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ en $\text{L}^{\text{u}}\text{a}\text{T}_{\text{E}}\text{X}$ worden

uitgevoerd, op basis van de in het font opgeslagen informatie. Let wel: op zich zou je dit soort regels kunnen vastleggen onafhankelijk van het font (en dus ook kunnen toepassen bij gebruik van Type1-fonts) maar in dit geval liggen de regels vast in het OpenType font.

Gezien het aantal talen en gezien de taalspecieke font gerelateerde (al dan niet historische) zaken, kan men zich voorstellen dat fonts en fontontwerpen er niet eenvoudiger op worden. Ook kunnen we uitzien naar bugs in fonts, incomplete feature-specificaties, beperkte omvang van het aantal regels, enzovoort. Het is verder de vraag of en wanneer we gaan afwijken van zulke regels, immers, $\text{T}_{\text{E}}\text{X}$ -gebruikers hebben vaak zo hun eigen idee over wat mooi en wenselijk is. Hoe dan ook, het blijft opletten geblazen.

Hans Hagen