

*Nu de PC niet meer weg te denken is en een integraal onderdeel vormt van de werkzaamheden van vele mensen, worden de eisen die gebruikers aan applicaties stellen steeds hoger. Door de steeds verdergaande globalisering zien we de laatste jaren dat er behoefte is aan op maat gemaakte applicaties, die ook qua taal aansluiten op de eisen van de gebruikers.*

*Met de komst van het .NET Framework maakt Microsoft het de ontwikkelaars mogelijk op eenvoudige wijze de nieuwe generatie applicaties 'voor te bereiden' op internationalisatie.*

Een LCID is een 32 bits getal dat vroeger (in Win32) gebruikt werd om de taal en sub-taal mee aan te geven. Windows heeft een standaard LCID dat wordt toegekend op het moment dat Windows geïnstalleerd wordt. Elke gebruiker erft deze standaard LCID en geeft deze door aan alle resources die voor de gebruiker benodigd zijn. Het LCID kan per gebruiker worden ingesteld via de Regionale Instellingen in het Configuratie Scherm. Applicaties kunnen het LCID programmatisch veranderen door gebruik te maken van een API. In .NET wordt het LCID beschikbaar gesteld via de CultureInfo klasse, die gedefinieerd is in de System.Globalization namespace.

Visual Studio .NET maakt het eenvoudig om een geïnternationaliseerde forms-based applicatie te ontwikkelen. Het bouwen van een geïnternationaliseerde applicatie is op te delen in twee stappen. De eerste stap is het bouwen van de applicatie, die eenvoudig kan worden uitgebreid om andere talen te ondersteunen. De tweede stap is het feitelijke vertaalwerk van alle schermen van de applicatie naar de gewenste taal. Met Visual Studio .NET worden zogenaamde resource files, die de data in verschillende talen bevat, vanuit de form designer automatisch aangemaakt. Door op

het form de Localizable property op true te zetten en de Language property op bijvoorbeeld Dutch, wordt er een additionele resource file aan het form toegevoegd met de naam FormName.nl.resx. Er zijn nu twee aparte visuele instanties van het form, de standaard instantie (FormName.resx) en de taal specifieke instantie (FormName.nl.resx), die geheel apart van elkaar kunnen worden bewerkt. Voor elke gewenste taal wordt er dus een nieuwe taal-specifieke instantie van het form gegenereerd. Nadat de vertaling gereed is, worden alle FormName.xxx.resx resource files toegevoegd aan het form. Afhankelijk van de standaardtaalinstellingen (LCID), wordt de juiste resource file gebruikt en zullen alle controls op het form in de gewenste taal worden weergegeven.

De inhoud van een .resx file bestaat uit een XML structuur die gebruikt wordt om alle elementen op het form te beschrijven. Gelukkig bestaat er een tool in de .NET SDK, dat ons kan helpen bij het bewerken van de XML resource files, zonder dat kennis van de XML structuur noodzakelijk is. De Windows Resource Localization Editor (WinRes.exe) is eigenlijk de standalone versie van de Visual Studio .NET form designer.

Voor elke resource file (.resx) die wordt toegevoegd aan het form, wordt een aparte assembly aangemaakt met de volgende naamgeving: ProjectName.resources.dll. Deze assembly wordt een satellite assembly genoemd, bevat geen code en is terug te vinden in een subdirectory met de naam van de gewenste culture. Als de culture is ingesteld op nl-NL, dan zal er onder de debug/release directory een subdirectory te vinden zijn met de naam nl-NL, met daarin de satellite assembly (ProjectName.resources.dll).

Het .NET Framework en de Common Language Runtime maken het ontwikkelen van een geïnternationaliseerde applicatie erg eenvoudig. Door gebruik te maken van Visual Studio .NET en WinRes, wordt het mogelijk een Winforms applicatie te ontwikkelen die een adaptive user interface kan aanbieden. Bovendien kan door het model van satellite assemblies en het zoek algoritme van de runtime nieuwe taalondersteuning worden toegevoegd aan voorbereide geïnternationaliseerde applicaties, zonder code te veranderen of een nieuwe versie van de applicatie te installeren.

*Xander Buffart is trainer en ICT-Architect van Info Support*