

Doe meer met Visual Studio .NET

VERKEN JE ONTWIKKELOMGEVING

Microsoft heeft een ontwikkelomgeving gemaakt die out-of-the-box al heel wat functionaliteit biedt. Een grote troef van Visual Studio.NET is de mogelijkheid om alles aan te passen naar je eigen smaak, en eventueel uit te breiden. Dit artikel zal een aantal minder bekende tips aanhalen waarmee je nog meer plezier zult beleven bij het coderen.

Vaak moet je in je code op zoek gaan naar een bepaald stukje tekst. Een handigheidje hierbij is het incrementeel zoeken. Door de sneltoets Ctrl+I te drukken, kom je in de zogenaamde 'Incremental Search'-mode. Als je nu tekst intypt zal de cursor tijdens het typen verspringen naar de eerstvolgende plaats waar de tekst is gevonden. Typfouten worden vermeden, omdat je onmiddellijk kan zien of je zoekopdracht iets oplevert. Met de F3-toets verspringt de cursor naar de volgende plaats waar het gezochte tekstfragment voorkomt. Ctrl+Shift+I zoekt in de omgekeerde richting, namelijk van beneden naar boven. Veel VB6-ontwikkelaars zullen al gemerkt hebben dat de functie 'Last Position' is verdwenen in Visual Studio .NET. Deze functie liet je bijvoorbeeld toe om na het navigeren in je code terug te keren naar de oorspronkelijk positie. Deze navigatie is in Visual Studio .NET vervangen door de navigation-knoppen in de standaard toolbar. Teruggaan naar de laatste positie kan ook met de sneltoets Ctrl+-.

Kopiëren en plakken (Ctrl+C en Ctrl+V) zal bij iedere ontwikkelaar wel bekend zijn. Minder bekend is dat de laatste twintig gekopieerde items worden onthouden. Om snel door de lijst van gekopieerde items te lopen, kan je de sneltoets Ctrl+Shift+V gebruiken. Bij het herhaaldelijk indrukken van de V-toets (Ctrl+Shift blijven indrukken) zullen één voor één de voorheen gekopieerde items worden getoond. Nog een handigheidje bij het kopiëren: als er geen tekst is geselecteerd en je kiest Copy (Ctrl+C), dan wordt de hele lijn waarop de cursor staat gekopieerd.

Door de vele toolbars en vensters van Visual Studio .NET is de plaats waar code wordt geschreven wel eens beperkt. Naast een grotere monitor kopen, kan je de werkruimte ook vergroten door een of meer van deze toolbars tijdelijk te minimaliseren. Dit doe je door op het punaise-icoontje te klikken. Je toont de geminimaliseerde items opnieuw door de muis boven het icoontje te bewegen. Om heel het scherm vrij te maken voor de code-editor, kan je ook de sneltoets Alt+Shift+Enter gebruiken. Met deze combinatie kan je wisselen tussen een al dan niet full-screen code-editor.

Het selecteren van code die verspreid is over meer regels zorgt ervoor dat ook de tabs die voor de coderegels staan worden geselecteerd. Om een blok code te selecteren zonder deze voorafgaande tabs, kan je de code selecteren met de Alt-toets ingedrukt. Op deze manier kan je een rechthoek selecteren waaruit de code zal worden gekopieerd.

Vaak ben je als ontwikkelaar op zoek naar de juiste naam van een of andere class. Visual Studio .NET heeft hiervoor het 'Find Symbol' venster. Hier kan je zoeken aan de hand van de volledige of een gedeeltelijke naam van een Class of Class Member. De

sneltoets om dit venster te tonen is Alt+F12. In dezelfde categorie hoort ook de Object Browser thuis die is op te roepen met de sneltoets Ctrl+Alt+J. Sneltoetsen kan je ook zelf toewijzen aan commando's. Dit kan door in het Options-scherm, General/Keyboard te selecteren; zie afbeelding 1. Aan ieder beschikbaar commando in Visual Studio .NET en aan alle macro's kan je sneltoetsen toewijzen. Deze instellingen zijn per gebruiker en kunnen, indien gewenst, worden opgeslagen in verschillende schema's.

Mooiere code

Ontwikkelaars spenderen waarschijnlijk het grootste gedeelte van hun tijd aan het schrijven van code in de Visual Studio.NET code-editor. De nieuwe editor heeft een aantal verbeteringen meegekregen. Een van de verbeteringen die het meest in het oog springt, is de mogelijkheid om de code te laten samenklappen. Dit wordt aangegeven in de code-editor door een plus- of minteken aan de linkerkant, juist voor de coderegels. Door het samenklappen van code, die bijvoorbeeld reeds klaar is of van minder belang, krijg je een beter overzicht in de code-editor. Standaard is enkel samenklappen per codeblok mogelijk, bijvoorbeeld per class, per functie, per property, enzovoort. Niets belet je echter om grotere of kleinere codeblokken samen te klappen, zoals het 'Windows Form Designer generated code' codeblok. Dit kan je gemakkelijk krijgen door de #region RegionName ... #endregion regels toe te voegen aan je code. In Visual Basic is de syntax iets anders: #Region "Region-Name" ... #End Region. De RegionName kan ook spaties en andere karakters bevatten en wordt getoond in het label van de region. Het resultaat van het toevoegen van regions en het samenklappen ervan wordt getoond in afbeelding 1. Om snel alle mogelijke regions samen te klappen, kan de sneltoets Ctrl+M, Ctrl+O gebruikt worden.

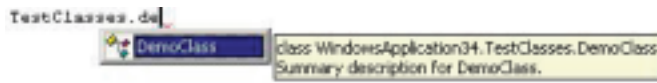
Een andere verbetering moeten we zoeken op het gebied van commentaargebruik. In de vorige versies van Visual Studio waren com-



Afbeelding 1. Region statements en samengeklapte code <<collapsing.bmp>>



mentaarlijnen louter informatief en enkel voor de ontwikkelaar. Maar sinds de komst van Visual Studio .NET is hier verandering in gekomen; commentaarlijnen kunnen nu extra functies hebben. Zoals te zien is in afbeelding 1 in de constructor van de DemoClass staat er op een commentaarlijn 'TODO'. Dit keyword zorgt ervoor dat een item wordt getoond in de Task List van Visual Studio .NET. Hiervoor moet de Task List in Policy-modus staan. Deze modus wordt geactiveerd door rechts te klikken op de Task List en onder het 'Shows Tasks'-item voor de 'Policy'-optie te kiezen. De sneltoets om de Task List te tonen is Ctrl+Alt+K. Het resultaat kan je zien in afbeelding 2.



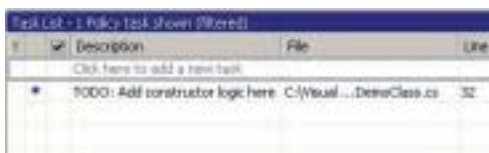
Afbeelding 2. Task List in Policy-modus <<tasklist.bmp>>

Een andere interessante mogelijkheid voor commentaarlijnen in Visual Studio.NET zijn de zogenaamde XML Comments. Zoals je kunt zien in afbeelding 1 worden bij het aanmaken van een nieuwe class automatisch drie commentaarlijnen toegevoegd, juist boven de class-definitie. De eerste en laatste lijn zijn XML-tags, de middelste lijn is de feitelijke waarde voor de XML-tag. Deze XML Comments kunnen worden toegevoegd aan ieder codeblok. Je zult merken dat wanneer je drie slashes (///) typt, Visual Studio automatisch de overeenkomstige XML-tags zal toevoegen. Het gebruiken van deze XML Comments heeft op het eerste gezicht weinig extra functionaliteit, maar er zijn toch twee grote voordelen. Wanneer je code gebruikt die voorzien is van XML Comments, zal Intellisense dit commentaar oppikken en tonen in de code-editor; zie afbeelding 3. Wanneer het project wordt gecompileerd, wordt een extra XML-bestand aangemaakt dat de XML Comments bevat, zodat deze ook kunnen getoond worden in andere projecten die jouw assembly gebruiken. Ten tweede is het zeer gemakkelijk wanneer XML Comments worden gebruikt om documentatie te laten genereren met behulp van bijvoorbeeld NDoc. In Visual Studio .NET 2002 en 2003 zijn XML Comments enkel beschikbaar voor C#. Maar in de op stapel staande versie van Visual Studio, codenaam Whidbey, zullen XML Comments ook beschikbaar zijn voor VB .NET-ontwikkelaars.

Macro's

Macro's zijn waarschijnlijk welbekend uit de Office-omgevingen, maar ook in Visual Studio.NET kunnen macro's gebruikt worden om repetitieve taken uit te voeren. Net zoals in Office kan je macro's opnemen, bekijken, editen en opnieuw laten uitvoeren. De beschikbare macro's kan je bekijken en laten uitvoeren in de Macro Explorer (sneltoets Alt+F8). Bovendien kunnen veel gebruikte macro's ook gekoppeld worden aan zelfgedefinieerde sneltoetsen; via het Options-scherm: Environment/Keyboard.

Een heel handige macro is de ConvertToProperties-macro, die gedownload kan worden via de link onder aan dit artikel. Deze macro kan gebruikt worden om makkelijk en snel code te genereren voor properties. Het enige wat je moet doen om een property aan te maken is de private variabele declaratie typen en selecteren, en daarna de macro laten uitvoeren. Het resultaat is een volledige property-definitie, compleet met getters en setters, gebaseerd op de declaratie van de interne variabele. Zie ook de



Afbeelding 3. XML Comments in actie <<intellisense.bmp>>

```
C#:
string _test;
//Converteert naar:
public string Test
{
    get
    {
        return _test;
    }
    set
    {
        _test = value;
    }
}
```

Codevoorbeeld 1: C#

Converteren van private variabelen naar properties

```
VB .NET:
Private _test As String
'Converteert naar
Public Property Test() As String
    Get
        Return _test
    End Get
    Set(ByVal value As String)
        _test = value
    End Set
End Property
```

Codevoorbeeld 1: VB .NET

Converteren van private variabelen naar properties

voorbeelden van het aanroepen van deze macro in codevoorbeeld 1. Om deze macro te gebruiken, download het Macroproject via de link onder aan dit artikel, en laad dit project via het Tools-menu, Macros, Load Macro Project in Visual Studio. Je kunt de macro dan uitvoeren via de Macro Explorer (Alt+F8) of door er een sneltoets aan te koppelen.

Nuttige links

ConvertToProperties macro

Broncode: <http://xxx/ConvertToProperties.txt>

Macro Project: <http://xxx/ClassMacros.vsmacro>

Meer Visual Studio.NET tips

<http://msdn.microsoft.com/vsdata/tips/showtips.asp?group=1&area=1&ID=1>

XML Documentation Tutorial

<http://msdn.microsoft.com/library/en-us/csref/html/vcwlxmldocumentationtutori>

Automating Repetitive Actions by Using Macros

<http://msdn.microsoft.com/library/en-us/vsintro7/html/vxoriatomatingrepetitivea.asp>

Jan Tielens is als .Net-architect werkzaam bij Ordina en onderhoudt ook een weblog die je kunt vinden op: <http://weblogs.asp.net/jan>

