

# IntelliTrace: Sneller en effectiever debuggen

Tom de Koning

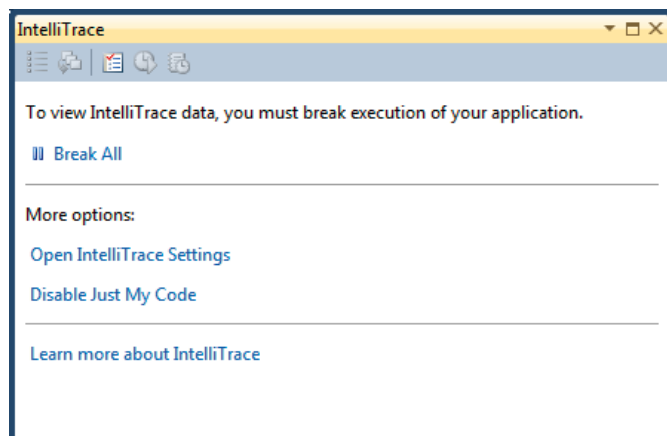
Als ontwikkelaar besteed je veel tijd in de debugger en iedereen heeft daar zijn eigen tips en trucs voor. Het hoort er bij, maar het kost erg veel tijd, is een pijnlijk proces en het levert geen business waarde op. IntelliTrace van Visual Studio 2010 bied je nieuwe debugfunctionaliteit, waarmee je op elk moment een debugsessie kan pauzeren om vervolgens met een soort video 'playback'-mechanisme door de lijst van events heen te lopen.

Deze functionaliteit biedt enorme voordelen voor de productiviteit van een developer, aangezien een applicatie veel minder vaak herstart hoeft te worden. In dit artikel gaan we dieper in op IntelliTrace, logbestanden, en opties die je helpen in een debugsessie.

Om de werking van IntelliTrace te illustreren gebruiken we verschillende voorbeeldapplicaties, waaronder Contoso Automotive uit de Visual Studio 2010 Training Kit en de voorbeeldapplicatie behorende bij de IntelliTrace labs.

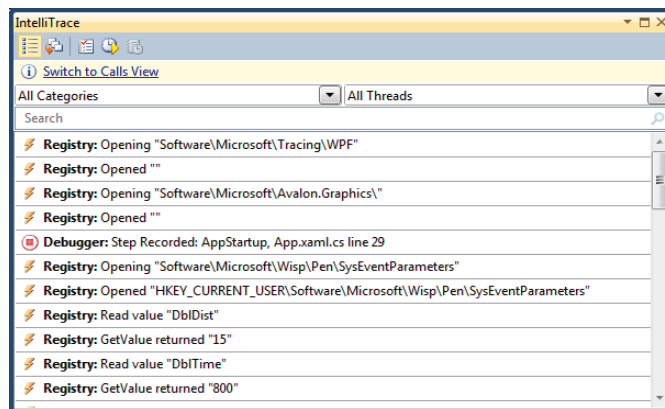
## IntelliTrace in het kort

Het leren werken met bestaande applicaties heeft een steile leercurve en vereist een groot uithoudingsvermogen, want je besteedt de meeste tijd in de debugger om de code te doorgronden. Met IntelliTrace kun je tijdens je debugsessie een pauze moment inlassen (zie Figuur 1), waarna je toegang krijgt tot de opgenomen lijst van events en method calls.



FIGUUR 1 BREAK ALL BIJ INTELLITRACE

Na het kiezen van 'Break All' wordt een venster geopend met daarin in chronologische volgorde alle acties tot dan toe (zie Figuur 2). Je kunt daarmee eenvoudig heen en weer stappen tussen de verschillende events, waarbij de desbetreffende source inclusief call stack automatisch getoond wordt.



FIGUUR 2 INTELLITRACE EVENTS PANE

Microsoft heeft bij de ontwikkeling van Visual Studio 2010 goed geluisterd naar de wensen van developers. Niet alleen is er een flinke inhaalslag gemaakt in refactoring tools, JavaScript ondersteuning en betere deployment mogelijkheden, maar er is nu ook een flinke efficiency slag gemaakt in het dagelijks werk van een developer. Het geschetste heen en weer springen in de debugger waarbij de applicatie keer op keer opnieuw moet worden opgestart, komt met IntelliTrace grotendeels te vervallen.

Niet alleen is het duidelijker welke events en method calls binnen een applicatie plaatsvinden, je kunt nu ook het doorlopen van je code nadoen, waarbij je elk stuk met een paar klikken opnieuw kunt laten afspelen. Dit scheelt veel tijd omdat je nu niet langer iedere keer opnieuw hoeft te bouwen om alleen de beginwaarden van variabelen te kunnen terugzien.

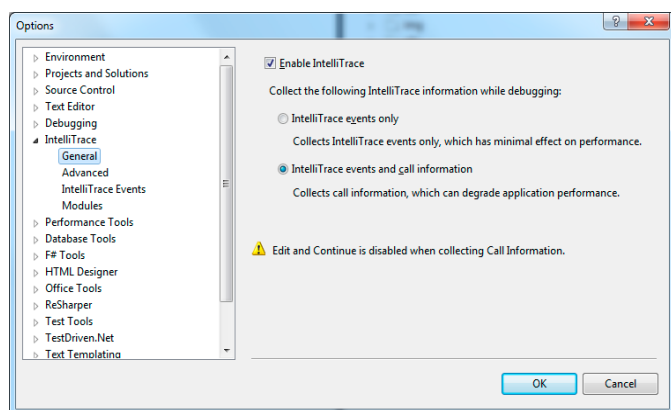
Een IntelliTrace sessie is voor één developer op zichzelf al goud waard, maar het wordt nog beter als je bedenkt dat je een IntelliTrace sessie kunt importeren en exporteren. Bug reports bevatten bijna nooit de exacte stappen die genomen zijn en bugs zijn daarom lastig te reproduceren. Testers kunnen nu een Microsoft Test Manager test uitvoeren en aan een IntelliTrace sessie koppelen. Het IntelliTrace logbestand wordt automatisch aan een work item gekoppeld op het moment dat een teststap niet kan worden door-

## Je kunt op elk moment vanuit het venster een snapshot maken met de tot dan toe opgenomen events door IntelliTrace

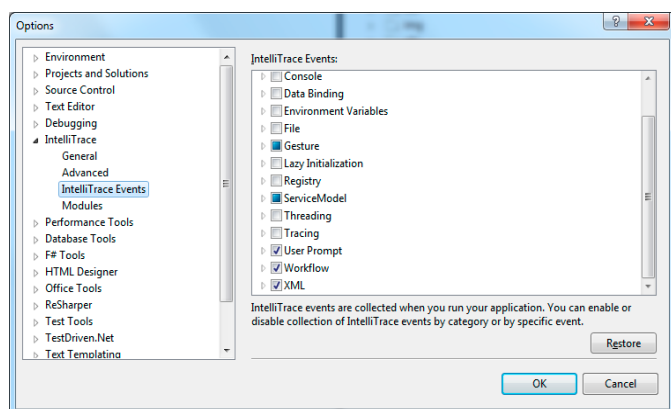
lopen. Hiermee behoort het pijnlijke ‘niet reproduceerbaar’ tot het verleden. Het reproduceren van een foutsituatie is niet langer machine afhankelijk; een developer doorloopt eenvoudigweg de lijst met events zoals ze bij de tester hebben plaatsgevonden en ziet vanzelf waar het fout gaat.

### Beginnen met IntelliTrace

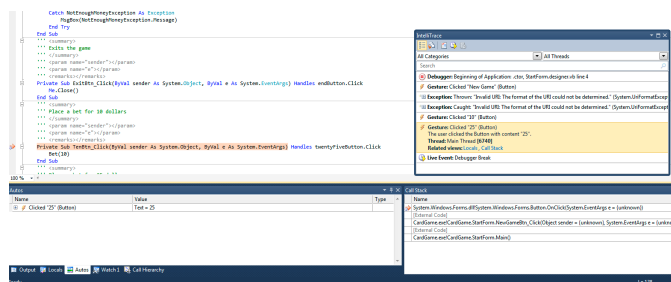
Voordat je met IntelliTrace aan de slag gaat, is het handig om even kort de opties te bespreken. De IntelliTrace opties kun je terugvinden via Tools | Options | IntelliTrace sectie. Het General menu bevat veruit de belangrijkste optie: de keuze tussen “IntelliTrace events only” en “IntelliTrace events and call information”.



FIGUUR 3 GENERAL INSTELLINGEN INTELLITRACE



FIGUUR 4 INTELLITRACE EVENTS



FIGUUR 5 SOURCE CODE HIGHLIGHTING NA KLIKKEN OP HET EVENT

Events only zal in de events historie alléén framework events laten zien. Events and call information biedt veel meer informatie waaronder de zojuist genoemde framework events, diagnostische gegevens, method calls en method-level tracing. Het kiezen van deze optie zorgt echter ook voor een grotere belasting van je machine tijdens het debuggen omdat er meer data wordt gelogd. Met de huidige kostprijs per GB hoeft dit geen probleem te zijn, maar het is wel iets om in de gaten te houden. Daarnaast kan je machine traag worden omdat er zoveel wordt weggeschreven. Je kunt goed filteren op het type events en op dll's, waardoor met wat tunen de performance prima blijft.

Het Advanced menu bevat het pad naar en de maximale grootte van de logfile. Met de check box voor “display navigation gutter” krijg je icoontjes waarmee je kan navigeren tijdens je Debug sessie. Deze optie staat default aan en dat is goed aangezien het de navigatie tussen IntelliTrace pane en code een stuk vergemakkelijkt. Het IntelliTrace Events menu bevat een overzicht van alle events die worden weergegeven in je sessie tijdens debuggen. De lijst met events is opgedeeld in de verschillende framework categorieën. Het kan handig zijn om een aantal categorieën uit te schakelen omdat dit het overzicht en snelheid enorm bevordert.

Het Modules menu is een belangrijke sectie waar je dll's kunt toevoegen waarvan je geen informatie wilt bijhouden. In het bijzonder bij applicaties waar veel dll's worden gebruikt als Unity of MEF is dat belangrijk. Stelregels hiervoor zijn er niet echt: het is een kwestie van uitproberen. Het kan handig zijn om hier alleen de dll's van een externe library toe te voegen, zodat communicatie over en weer met deze duidelijk wordt. Overigens kan dit ook in het IntelliTrace history pane door handig je filters te kiezen.

### De IntelliTrace sessie

Je roept het IntelliTrace venster op door de applicatie te starten, waarna je op elk moment een snapshot kunt maken met de tot dan toe opgenomen events door IntelliTrace. Dit doe je via Debug | IntelliTrace | IntelliTrace Events. Je krijgt dan eerst een kort status overzicht te zien, waar je kunt kiezen voor ‘break all’. Hiermee wordt de debugsessie gepauzeerd en krijg je een overzicht van alle events en details tot dan toe (zie figuur 8).

Bovenin het IntelliTrace venster kun je zoeken naar een specifiek event of lijst van events in een categorie. Vooral dit filteren en het feit dat alle events geordend onder elkaar staan, zorgt voor een veel beter overzicht van wat je code nu eigenlijk doet. Daarnaast stelt IntelliTrace je in staat per thread een overzicht van calls te geven. Door nu op het “gesture” event te klikken, wordt automatisch de source code getoond (zie figuur 5).

“Events only” modus is een erg prettige manier om met weinig overhead toch snel een overzicht van de gemaakte calls binnen je applicatie te maken. Zoals je in figuur 5 kunt zien wordt ook automatisch context informatie met betrekking tot variabelen geladen in het Autos pane.



HET REPRODUCEREN VAN EEN SITUATIE IS NIET LANGER MACHINE AFHANKELIJK; EEN DEVELOPER DOORLOOPT BIJVOORBEELD EEN LIJST MET EVENTS ZOALS ZE BIJ EEN TESTER HEBBEN PLAATSGEVONDEN.

## De diepte in

Het is eenvoudig om filmbelton vooruit te spoelen, te herhalen of gewoon nog eens terug te kijken. Met IntelliTrace kan dit nu ook, maar dan met debuggen. Wanneer je kiest voor “events and call information” word je de regisseur van je debugsessie. Je kunt switchen tussen verschillende gerezen events en de method calls die hiervoor verantwoordelijk zijn. Vanuit de method view kan je dan weer inzoomen in de code. Je krijgt daarmee ook direct toegang tot de vertrouwde debugtools als toetscombinaties (o.a. F10 en F11) aangevuld met navigatie-icoontjes in het lichtblauwe kader (zie figuur 6).

```

34 List<Car> cars = new List<Car>();
35 Stopwatch stopwatch = new Stopwatch();
36 CancellationTokenSource searchOperation = new CancellationTokenSource();
37
38 [ImportMany(typeof(ICarQuery), AllowRecomposition = true)]
39 public ObservableCollection<ICarQuery> CarQueries { get; set; }
40
41 public CashMaker()
42 {
43     this.InitializeComponent();
44
45     this.CarQueries = new ObservableCollection<ICarQuery>();
46     this.commandGrid.DataContext = this.CarQueries;
47
48     Task.Factory.StartNew(GenerateCars);
49
50 }

```

FIGUUR 6 INTELLITRACE NAVIGATION GUTTER

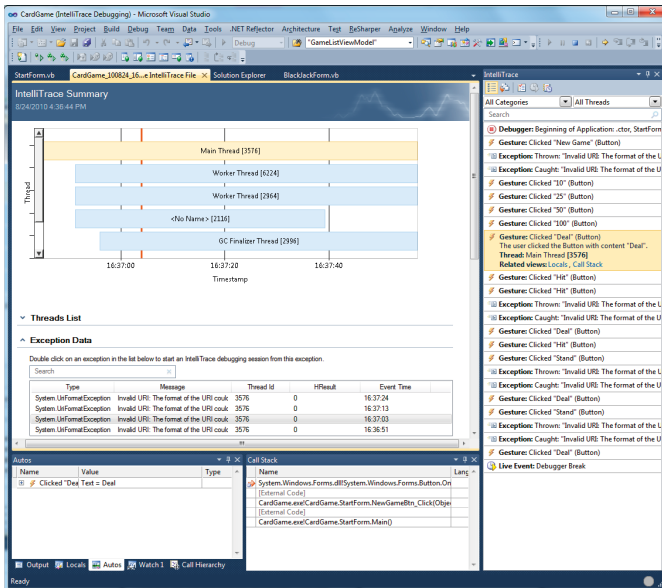
Een belangrijk verschil tussen de normale debugomgeving en IntelliTrace is het verschil in context; Bij IntelliTrace kan je alleen op en neer stappen tussen opgenomen method calls. Dit houdt in dat lambda's die verwijzen naar methodes vaak niet worden opgenomen. Een omweg is het zetten van een tracepoint of breakpoint in de onderliggende code. IntelliTrace neemt namelijk alle context op binnen een methode waar een breakpoint staat. We komen daarmee meteen op de volgende innovatie om sneller te kunnen werken tijdens debuggen; In plaats van lukraak breakpoints te zetten, kan je met IntelliTrace ook de omgekeerde weg bewandelen. Indien je het vermoeden hebt waar het probleem

```

ContosoAutomotive.CashMaker
Searching For: CashMaker 1 of 1
35 Stopwatch stopwatch = new Stopwatch();
36 CancellationTokenSource searchOperation = new CancellationTokenSource();
37
38 [ImportMany(typeof(ICarQuery), AllowRecomposition = true)]
39 public ObservableCollection<ICarQuery> CarQueries { get; set; }
40
41 public CashMaker()
42 {
43     this.InitializeComponent();
44
45     this.CarQueries = new ObservableCollection<ICarQuery>();
46     this.commandGrid.DataContext = this.CarQueries;
47
48     Task.Factory.StartNew(GenerateCars);
49
50 }

```

FIGUUR 7 RESULTATEN NAAR METHODE ZOEKEN



FIGUUR 8 OPENEN VAN ITRACE LOGFILE OM EEN TEST TE ANALYSEREN

zich voordoet maar je wilt weten vanuit welke context je in die code terecht komt, kan je door een rechtermuisklik in de code de methode of regel opzoeken in de IntelliTrace call geschiedenis. Bovenin het scherm zie je hoe vaak de methode geraakt wordt (zie figuur 7). Als je de navigatie gebruikt, wordt ook automatisch de historie in het IntelliTrace venster getoond.

Je kunt door middel van de blauwe navigatie iconen bovenin in figuur 7 gemakkelijk switchen tussen de verschillende momenten waarop je code geraakt werd. Het mooie is dat van ieder moment ook weer context meekrijgt in de "autos" en "locals" schermen. Je zult in de praktijk vaak switchen tussen het "events" en het "calls" scherm: Vanuit een event switch je naar de bijbehorende calls, doorloopt de code, inzoomen en vervolgens terugzoeken via 'Search for this Method in IntelliTrace'.

Aangezien alle informatie in een IntelliTrace sessie naar een iTrace logfile wordt geschreven stelt het je ook in staat om een eerder opgenomen debugsessie nog eens af te spelen. Dit opent meteen mogelijkheden voor testers om iTrace bestanden aan te

leveren waardoor je exact kunt nagaan wat er tijdens het aflopen van de test is gebeurd. Een tester kan met Microsoft Test Manager 2010 eenvoudig een test koppelen aan een iTrace logfile. In het kort komt het erop neer dat een iTrace file wordt gekoppeld aan een bug work item dat door een developer uit Team Foundation Server kan worden opgehaald en afgespeeld. Bij het openen van een iTrace file wordt een helder overzicht van alle threads geopend. Als je klikt op de 'Main Thread', wordt automatisch het IntelliTrace venster geopend waarna je dezelfde IntelliTrace ervaring hebt als wanneer je de applicatie gepauzeerd zou hebben.

De interactie met het testteam zal een stuk gemakkelijker verlopen doordat bugs eenvoudiger zijn te reproduceren.

## Conclusie

IntelliTrace in Visual Studio 2010 is een geweldige vooruitgang van het oplossen van problemen met ontwikkeltools. Een developer wint aanzienlijk tijd door niet iedere keer de applicatie opnieuw te hoeven afspelen. Daarnaast zal de interactie met het testteam een stuk gemakkelijker verlopen doordat bugs eenvoudiger zijn te reproduceren. Vanuit de IntelliTrace logfile kan je alle informatie afleiden om tot een snelle oplossing te komen.

## Referenties

Debugging with IntelliTrace

<http://msdn.microsoft.com/en-us/library/dd264915.aspx>

IntelliTrace blog

<http://blogs.msdn.com/b/ianhu>



**Tom de Koning**, is application development consultant bij Avanade Netherlands en is bereikbaar via [tom.de.koning@avanade.com](mailto:tom.de.koning@avanade.com).

## KORT NIEUWS

# TU Delft start pilot voor gebruik van Microsoft Online Services

In samenwerking met SURFdiensten <<http://www.surf-diensten.nl>> en Microsoft Nederland <<http://www.microsoft.nl>> start de TU Delft <<http://tudelft.nl/>> een pilot om het gebruik van Microsoft's Business Productivity Online Standard Suite (BPOS) te testen. Met deze pilot wil de universiteit ervaring opdoen met gemengde omgevingen van zowel on-premises producten als cloud producten. Ook wil de TU Delft hiermee de kosten voor software terugbrengen en de samenwerking tussen medewerkers bevorderen. De pilot loopt tot en met december 2010. De ervaringen en resultaten worden door de samenwerkende partijen gratis ter beschikking gesteld aan onderwijsinstellingen in Nederland.

TU Delft zet SharePoint Online, Office Live Meeting en Office Communications Online in om de communicatie en het online samenwerken van medewerkers binnen en buiten de TU Delft te faciliteren. De pilot bestaat uit deelprojecten op verschillende afdelingen. Zo zullen onder andere de TU Delft Library, TU Delft

Valorisatie Center, de afdeling Industrial Design en ICT-managers uit drie landelijke overleggenia deelnemen. Microsoft's verantwoordelijke voor het hoger onderwijs Michael Kleine: "Bij gebruik van online diensten hoeven onderwijsinstellingen vooraf geen investering te doen voor de aanschaf en implementatie binnen de eigen omgeving. Microsoft verzorgt de hele oplossing van hardware, stroom en koeling, tot en met installatie, updates, storage en back-up. De toepassingen zijn daarnaast overal en altijd voor medewerkers en studenten toegankelijk, waardoor zij waar en wanneer zij willen de mogelijkheden kunnen benutten." De licenties voor het gebruik van de Microsoft-producten worden afgesloten door SURFdiensten. Deze inkooporganisatie voor het hoger onderwijs werkt samen met zijn partners APS IT diensten, SLBdiensten en Microsoft Nederland om zodoende de hele onderwijssector, van primair tot universitair onderwijs, te helpen hun weg te vinden binnen het groeiende aanbod van gehoste en cloud oplossingen.