

Smart client is een door Microsoft geïntroduceerde term voor applicaties die de beste eigenschappen van thick en thin clients in zich verenigen. Een smart client applicatie gebruikt de lokale processor, gebruikt het netwerk af en toe en kan worden uitgerold en onderhouden vanaf een centrale server.

Het .NET Framework (Windows Forms) van Microsoft biedt met Visual Studio 2005 veel ondersteuning voor smart clients. In dit artikel belicht Willem Koppenol de achtergronden van smart clients in een Microsoft-omgeving.

# Smart client applicaties

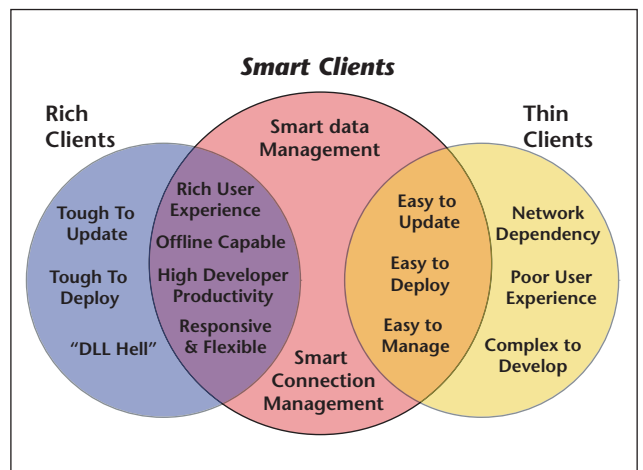
## Het beste van twee werelden

Een hoofdrol speelt de ClickOnce-technologie waarmee onder andere automatische updates van smart client-applicaties kan plaatsvinden. De smart client-architectuur is echter niet voorbehouden aan Microsoft. De Java-wereld kent met Java Web Start vergelijkbare mogelijkheden. De smart clients hebben met de opkomst van Ajax in webapplicaties concurrentie gekregen.

**THICK CLIENTS** De smart client architectuur is gebaseerd op de lessen die geleerd zijn in de recente geschiedenis van de applicatie-ontwikkeling. In de jaren negentig van de vorige eeuw zijn er veel zogeheten thick client applicaties ontwikkeld. Dergelijke applicaties hebben over het algemeen een rijke grafische user interface. Ze gebruiken de lokale rekenkracht en zijn prima geschikt voor het invoeren en raadplegen van gegevens. Vaak worden deze gegevens via het netwerk opgeslagen in een centrale database.

Het nadeel van dit type client-applicaties is echter dat de installatie en beheer in grotere organisaties problemen geeft omdat de applicatie op elk gebruikerssysteem moet worden geïnstalleerd. Bij het installeren van een nieuwe versie van de applicatie betekent dit wederom veel werk voor systeembeheer. Ook kunnen deze clients vaak onderling geen gegevens uitwisselen. Het kwam ook veelvuldig voor dat deze applicaties conflicten hadden met andere applicaties doordat ze elkaars DLL's overschreven en verschijnsel dat bekend werd onder de naam 'de DLL-hel'.

**THIN CLIENTS** Eind jaren negentig deden de webapplicaties hun intrede. Deze applicaties volgen het thin client model en zijn via een browser te gebruiken.



AFBEELDING 1. Thick, thin en smart clients.

Webapplicaties worden geïnstalleerd en onderhouden op een centrale server. Ze zijn veel eenvoudiger te beheeren omdat op een gebruikerssysteem alleen een goede browser aanwezig moet zijn. Webapplicaties geven geen conflicten met andere applicaties op een gebruikerssysteem (geen 'DLL-hel'). Over het algemeen kunnen ze goed gegevens uitwisselen door middel van de webservices. In geval van nieuwe versies van de applicatie hoeft er op de systemen van de gebruikers in principe niets te gebeuren. Als de nieuwe versie op de server wordt geïnstalleerd krijgen de gebruikers daar automatisch de beschikking over.

Het nadeel van dit type client-applicaties is dat de gebruikersinterface veel minder rijk is. Ook is de performance vaak minder want de rekenkracht van het lokale werkstation wordt niet benut. Vooral bij het opvoeren en veranderen van grote hoeveelheden gegevens zijn

thin client applicaties onhandiger in het gebruik dan thick client applicaties.

**SMART CLIENTS** Smart client applicaties combineren de voordelen van thick en thin clients. Ze hebben een rijk geschakeerde user interface en kunnen gemakkelijk geïnstalleerd en onderhouden worden. Smart client applicaties voldoen vaak aan heel verschillende eisen en zijn dus heel divers qua ontwerp en implementatie. Toch kunnen we een aantal algemene kenmerken formuleren. Smart client-applicaties hebben over het algemeen een aantal van of al deze kenmerken:

*Het gebruik van lokale resources:* smart client applicaties draaien, in ieder geval ten dele, op de client en maken daar gebruik van de hardware (CPU, de disk en het geheugen) van het lokale systeem alsmede de daarop aanwezige software.

*Het gebruik van netwerk resources:* smart client applicaties staan nooit op zichzelf, maar maken altijd onderdeel uit van een grotere gedistribueerde oplossing. Ze zijn daarmee verbonden via het netwerk. Dit kan betekenen dat de applicatie communiceert met webservices die toegang geven tot bepaalde gegevens. In ieder geval staat de applicatie in verbinding met services die zorgen voor het installeren en onderhouden van de applicatie.

*Het offline-gebruik:* één van de voordelen van smart clients is dat ze zo gemaakt kunnen worden dat de gebruiker er ook mee kan werken als hij niet met het netwerk is verbonden. Voor bepaalde toepassingen is de mogelijkheid om de applicatie ook offline te kunnen gebruiken van essentieel belang. Denk aan mobiele werkers met laptops of PDA's die over een wisselende netwerk connectiviteit beschikken en in ieder geval geen garantie hebben op de aanwezigheid van een netwerk.

*Flexibel qua client device:* Hoewel het misschien geen strikte noodzaak is, wordt van smart client applicaties vaak verwacht dat ze ook draaien op PDA's en andere draadloze systemen. Vaak zijn er dan ook verschillende versies van een smart client applicatie, die ieder gericht zijn op een specifiek device en gebruik maken van de unieke eigenschappen van dat device.

**INSTALLATIE EN ONDERHOUD** Voor het installeren van applicaties op een client locatie, zou je gebruik kunnen maken van Setup-applicaties of Microsoft Installer files (MSI). Deze moeten dan wel over de verschillende clients worden gedistribueerd en daar worden uitgevoerd. Dit is een tijdrovend proces dat bovendien bij wijzigingen moet worden herhaald. De installatie en het onderhoud van smart client applicaties is daarom heel anders geregeld dan bij traditionele thick client-

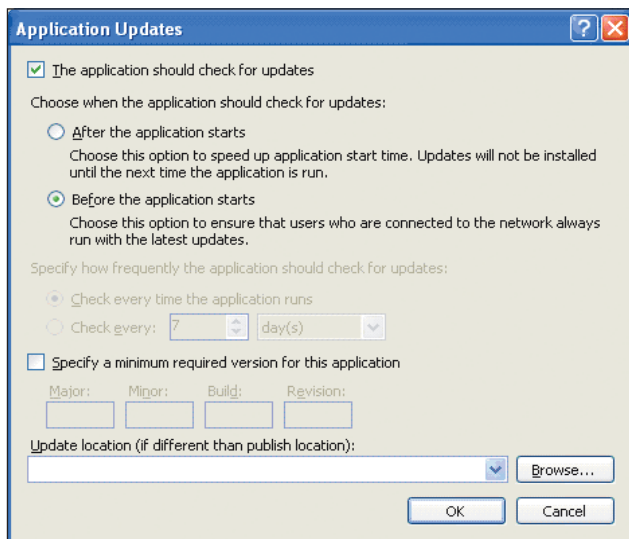
applicaties. In beginsel heeft een smart client applicatie twee componenten: enerzijds een kleine thin client applicatie die lokaal op de client aanwezig is en anderzijds de echte applicatie die door een webserver wordt gehost. Als de gebruiker de smart client-applicatie start, zal de kleine thin client verbinding zoeken met de webserver om een refererende assembly te checken. Is de gerefereerde assembly al lokaal aanwezig en is er geen nieuwere versie op de server dan wordt de lokaal aanwezige assembly gebruikt om de applicatie te starten. Is er wel een nieuwere versie van de assembly op de server dan wordt deze gedownload en wordt de applicatie daarmee gestart. Wijzigingen in de applicatie worden op deze manier in alle clients doorgevoerd zodra ze opstarten. Vandaar de naam smart clients. Voorbeeldcode waarin een dergelijk scenario wordt uitgevoerd ziet er in Visual Basic .NET als volgt uit:

```
namespace MySmartClient {
    class SmartClient {
        [STAThread]
        static void Main(string[] args) {
            SmartClient objSmartClient = new
            SmartClient();

            // Set the URL to load the Assembly
            from
            string strURL = "http://Server/Smart/
            Bin/Release/Smart.exe";

            // Set the class to call
            string sClassName = "MySmartClient.
            SmartForm";

            Assembly assemblyContent = null;
            Try {
                // Load the assembly
                assemblyContent = Assembly.
                LoadFrom(strURL);
            }
            catch(Exception e){}
            // Create a object for the Class
            Type typeContent = assemblyContent.
            GetType(sClassName);
            // Invoke the method. Here we are
            invoking the Main method.
            Try {
                typeContent.InvokeMember ("Main",
                BindingFlags.Public |
                BindingFlags.InvokeMethod |
                BindingFlags.Static, null,
                null, null);
            }
            catch(Exception e){}
        }
    }
}
```



**AFBEELDING 2.** Dialoog met opties voor het bijwerken van een smart client applicatie.

Door de ClickOnce-technologie in het .NET Framework 2.0 kunnen applicaties worden geïnstalleerd door een URL aan te klikken en worden aangepast terwijl ze draaien. Je hebt bij ClickOnce verschillende opties voor de wijze waarop aanpassingen in de applicatie worden doorgevoerd zoals te zien is in het dialoogvenster in afbeelding 2.

**SMART CLIENT TYPES** In de Microsoft Windows-omgeving kunnen we een drietal typen smart client applicaties onderscheiden die verschillen door het platform waarvoor ze bedoeld zijn:

- Windows smart client applications
- Office smart client applications
- Mobile smart client applications

Visual Studio 2005 in combinatie met Visual Studio Tools for Office dan wel het .NET Compact Framework biedt ruime ondersteuning voor de creatie van Office en Mobile smart client-applicaties. Niet zelden zijn smart client-applicaties voor verschillende platformen bedoeld. In de rest van dit artikel beschouwen we met name de Windows Smart Clients in .NET.

**WINDOWS SMART CLIENTS** Het .NET Framework 2.0 biedt, en met name de Windows Forms 2.0 bibliotheek, een platform waarop smart client applicaties gebouwd kunnen worden. Ook Visual Studio 2005 biedt de nodige ondersteuning voor deze technologie. Het deployment-model voor smart client applicaties waarbij gebruikers net als bij webapplicaties automatisch de nieuwste versie van een applicatie krijgen heet ClickOnce Deployment. Het is de opvolger van de No-Touch-Deployment (NTD) technologie die in Windows Forms 1.0 al werd geïntroduceerd. Wil een smart client-applicatie die gebruik maakt van de .NET-technologie kun-

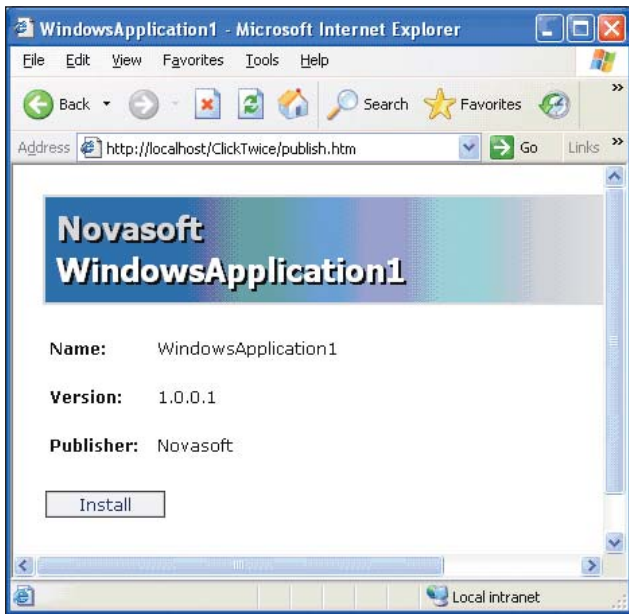
nen werken dan is er op de client wel een versie van het .NET Framework nodig. Dit kan echter meestal samen met de applicatie zelf worden gedownload.

**CLICKONCE** Met ClickOnce wordt een smart client-applicatie vanaf een URL of een UNC bestandslocatie gestart en desgewenst lokaal geïnstalleerd. Een nieuwe versie van de applicatie komt beschikbaar door de nieuwe bestanden onder de URL te plaatsen. De link naar een ClickOnce-applicatie verwijst naar een deployment-manifest met extensie '.application'. Bij activering van deze link zoekt de Deployment Framework Service (DFS) de huidige versie van de applicatie op, voorziet de client van de bestanden van de applicatie en start de applicatie. Is de applicatie eenmaal geïnstalleerd op de client, dan controleert een .NET Framework component op de client of er updates van de applicatie zijn. Dit gebeurt iedere keer wanneer een gebruiker de applicatie start. De component die dit regelt is de Updater Controller en het mechanisme kun je vergelijken met de manier waarop ook Windows Update werkt. Je kunt je laten waarschuwen voor nieuwe versies en patches en deze automatisch installeren.

**CLICKONCE OPTIES** ClickOnce is geïntegreerd in Visual Studio 2005 en biedt een reeks configuratie opties voor de installatie en het beheer van smart client applicaties. Een afgeronde Windows Form-applicatie publiceert je eerst op een webserver of bestandslocatie. Visual Studio geeft je standaard de publish-keuze onder het build-menu. Vervolgens zijn twee basale installatie-opties: online en online/offline. Bij de laatste optie kunnen ClickOnce applicaties zonder de aanwezigheid van een netwerk connectie worden opgestart mits ze tenminste reeds aanwezig zijn op het systeem van de gebruiker. Offline applicaties worden vermeld onder het Start menu van de gebruiker en kunnen via Add- en



**AFBEELDING 3.** Publicatie van een smart client applicatie met ClickOnce.

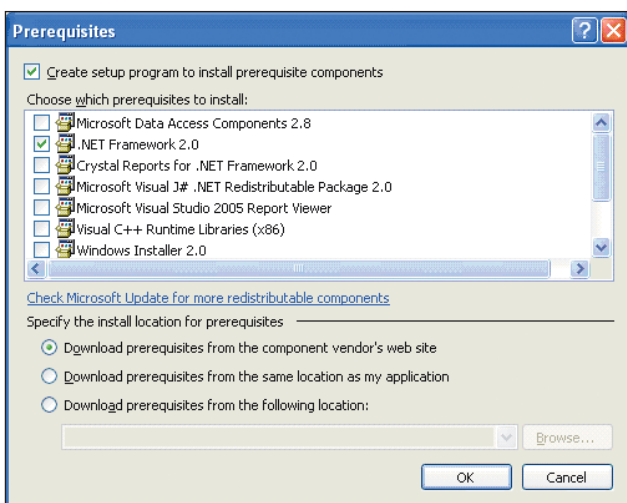


**AFBEELDING 4.** Installatie van een smart client via een URL op de webserver.

Remove-programs weer van het systeem van de gebruiker worden verwijderd.

Via Project Property's kan worden aangegeven van welke assembly's de smart client afhankelijk is, zoals bijvoorbeeld het .NET Framework 2.0, en of deze samen met de applicatiecode moeten worden gedownload en geïnstalleerd bij de client. Voorts zijn er nog allerlei publicatiedetails zoals een ondersteunende webpagina en een deployment-page die ClickOnce met default-opties genereert, maar die desgewenst aangepast kunnen worden.

**SECURITY** In het .NET Framework kan aan code in de vorm van assembly's een set permissies worden gegeven. Dit wordt Code Access Security (CAS) genoemd. Deze permissies lopen uiteen van de Internet-permissie



**AFBEELDING 5.** ClickOnce selectie dialoog van benodigde componenten.

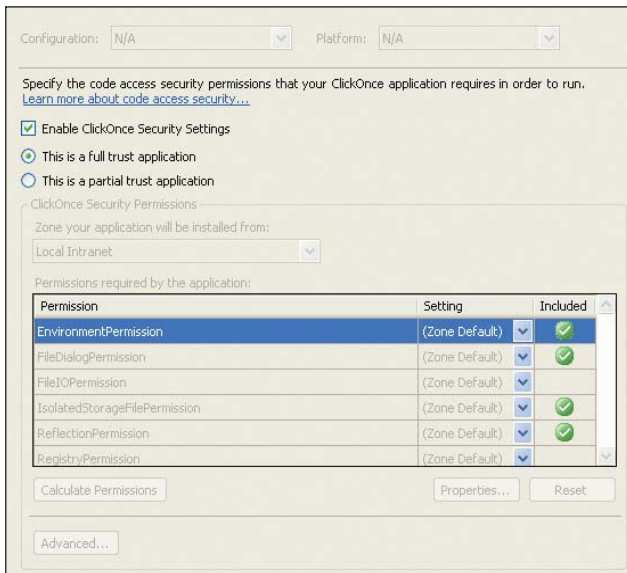
set met heel weinig rechten tot de FullTrust permissie set waarbij alles mag. Bij applicaties die met ClickOnce-technologie worden geïnstalleerd, is de default instelling de permissie set FullTrust. Dit betekent dat de applicatie allerlei activiteiten op de client kan uitvoeren. Over het algemeen is dit niet zo'n goede instelling omdat een applicatie dan niets in de weg wordt gelegd bij het benaderen van bijvoorbeeld het bestandssysteem of de registry op het client-systeem. Dit kan misbruik in de hand werken. Bovendien toont een applicatie die Full Trust permissies nodig heeft een dialoog aan de gebruiker waarin hij de gebruiker toestemming vraagt om te starten. Dit kan verwarrend werken bij de gebruikers van de ClickOnce applicatie. Het is daarom beter de applicatie precies die rechten te geven die hij nodig heeft en dit is mogelijk via het Security-tabblad van het project in Visual Studio.

**BEPERKINGEN CLICKONCE** ClickOnce is niet voor alle applicaties geschikt. Zo kunnen via ClickOnce geen drivers worden geïnstalleerd en kunnen geen bestandstypen worden geregistreerd. Ook is het niet mogelijk te installeren in de Global Assembly Cache. De applicatie wordt in de ClickOnce applicatie cache geplaatst en kan niet worden toegevoegd in de Program Files directory. ClickOnce is dus geen complete vervanger van het gebruik van Windows Installer files (MSI). Voor applicaties met dergelijke behoeften is het gebruik van MSI dus een beter alternatief, maar we hebben het dan niet meer over smart clients.

**OMGAAN MET DATA** Een complicerende factor bij smart clients is het omgaan met data. Want terwijl de applicatiedata bij thin clients in webapplicaties over het algemeen gecentraliseerd zijn in de webserver of in back-end services, zijn applicatiedata bij smart clients veelal aanwezig op de client. Deze data kunnen beschikbaar zijn gesteld via een serverapplicatie zoals een webservice. Om performance-redenen of om offline te kunnen werken, kan de smart client applicatie deze data cachen. Je zult dan moeten beslissen hoe wordt omgegaan met data die niet in de pas lopen met de serverdata. Je wilt immers geen verouderde data gebruiken. Mogelijk kan de smart client ook lokaal wijzigingen aanbrengen in data. In dat geval moet worden vastgelegd hoe de data met de server worden gesynchroniseerd, hoe wordt omgegaan met conflicterende data en hoe wordt bijgehouden welke wijzigingen naar de server moeten gestuurd. Dit zijn complicerende factoren die om een oplossing vragen. In ieder geval is het voor een goede werking van de smart client applicatie van belang te zorgen voor geldige en consistente data.

**SLOTWOORD** Smart client applications combineren de rijk geschakeerde user interface van thick client





**AFBEELDING 6.** ClickOnce Security Settings in Visual Studio Project.

applicaties met de eenvoud van installatie en onderhoud van thin client webapplicaties. Het .NET Framework biedt met de ClickOnce deployment technologie en de uitgebreide Windows Forms bibliotheek

ruime ondersteuning voor de ontwikkeling van smart client applicaties. Met de komst van Ajax krijgen webapplicaties echter een veel rijker en meer responderende user interface. Hoewel Ajax-applicaties vooralsnog niet offline gebruikt kunnen worden, wordt het voordeel dat smart client-applicaties hebben op traditionele webapplicaties daarmee ten dele teniet gedaan. In de toekomst staan ontwikkelaars en hun managers dus niet alleen voor de keuze tussen Java of .NET, maar ook voor de keuze tussen smart clients of Ajax. Mogelijk sluiten deze opties elkaar niet altijd uit en kunnen ze coëxisteren of misschien zelfs samenwerken.


*drs. Willem Koppenol is Senior Trainer en Product Specialist  
 Software Development Training bij Twice IT Training  
 (e-mail: wkoppenol@twice.nl).*

# LANSA Integrator

## Driving Down the Cost of iSeries Connectivity

**LANSA Integrator is the practical choice to transfer and convert data directly between your iSeries and any target environment. Publish and consume Web Services or format and distribute documents. LANSA Integrator is the answer to all your connectivity questions.**

**Get to know LANSA today. Call +31 (0) 20 698 0011 or visit [www.lansa.com/integrator](http://www.lansa.com/integrator)**



**LANSA**  
ADVANCED SOFTWARE MADE SIMPLE

LANSA Ltd.  
 Paasheuvelweg 15  
 1105 BE Amsterdam

