

Interop today: Host Integration Server 2004

INTEGRATIE TUSSEN APPLICATIES IN HETEROGENE OMGEVINGEN

Microsoft Host Integration Server (HIS) biedt een aantal technologieën waardoor het mogelijk is om het Microsoft Windows-platform¹ te koppelen met non-Microsoft-platformen. HIS richt zich specifiek op IBM OS/390 mainframes en IBM OS/400 midrange computers. De geschiedenis van deze systemen gaat terug tot begin jaren '70. Vanaf deze tijd is er veel geïnvesteerd in bedrijfskritische applicaties op deze host-platformen.

HIS slaat een brug tussen het Microsoft Windows-platform en genoemde IBM-platformen. Hierdoor kunnen .NET-ontwikkelaars applicaties ontwikkelen die integreren met mainframes en midrange systemen. Met behulp van HIS kunnen bijvoorbeeld web- of mobiele toepassingen ontwikkeld worden die gebruik maken van achterliggende mainframesystemen. Wat is Host Integration Server 2004? HIS bestaat uit de volgende onderdelen waardoor je integratie op de verschillende manieren kunt realiseren:

- Transaction Integrator
- Data Services
- Messaging Services

Transaction Integrator koppelt applicaties door vertaling van functieaanroepen. Hierdoor kun je applicaties op verschillende platformen gebruik laten maken van elkaars functionaliteit. Data Services maakt databases toegankelijk op verschillende platformen. Hierdoor kun je databases benaderen op ondersteunde hostplatformen vanuit .NET. Messaging Services integreert messaging-systemen. Hierdoor kun je koppelingen realiseren tussen MSMQ (Microsoft Message Queuing) en IBM MQ Series.

HIS biedt op niveau's een integratieoplossing. Zie afbeelding 1. Applicatie-integratie wordt door middel van de Transaction Integrator bereikt. Databases worden door middel van Data Services toegankelijk. Messaging-systemen worden door middel van MQ Series Bridge gekoppeld.

HIS Data Services

HIS Data Services bestaat uit een aantal hulpmiddelen en componenten; zie tabel 1. Met HIS Data Services kun je databases op IBM mainframe- en/of midrange-systemen benaderen. De HIS Data Services bestaat uit een configuratietool genaamd Data Access Tool; zie afbeelding 3 en HIS-specifieke OLE DB providers en ODBC-drivers. De Data Access Tool configureert de Data Services componenten - de OLE DB providers en de ODBC-driver. Je kunt

HIS Data Services	
Hulpmiddel:	• Data Access Tool
Componenten:	• OLE DB Provider voor AS/400 • OLE DB Provider voor VSAM • OLE DB Provider voor DB2 • ODBC Driver voor DB2

Tabel 1. HIS Data Services

¹ Met platform wordt bedoeld besturingssysteem, applicaties en databases.

door middel van HIS Data Services databases (DB2, VSAM) op het hostplatform benaderen.

Voorbeeldscenario met HIS Data Services

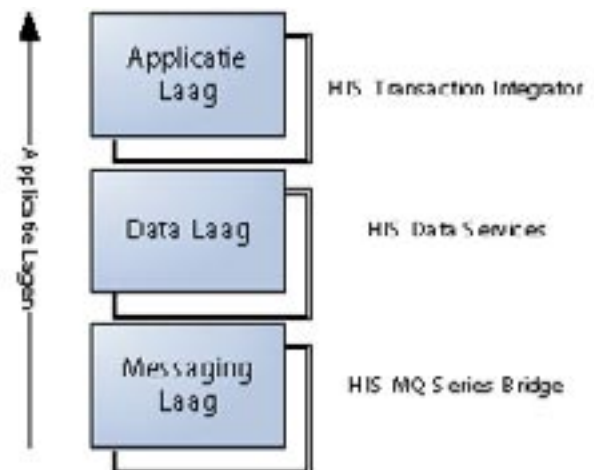
Het voorbeeldscenario bestaat uit de implementatie van een XML webservice waardoor queries op een DB2-database uitgevoerd worden. Het voorbeeld in afbeelding 2 demonstreert de functionaliteit van HIS Data Services.

Het voorbeeld implementeert een XML webservice die door middel van ADO.NET een DB2-database op het AS400-platform benadert. De uitwerking hiervan bestaat uit:

1. configuratie van de dataverbinding door middel van Data Access Tool
2. implementatie van een webservice.

Configuratie doe je met de Data Access Tool

Implementatie van data-integratie met HIS begint met het configureren van een dataprovider door middel van een connection-string. In de praktijk krijg je veelal een gedeelte van de benodigde configuratiegegevens aangeleverd vanuit beheer zoals protocol, netwerkadres, besturingssysteem en dergelijke. HIS levert de zogenaamde Data Access Tool ten behoeve van configuratie van een dataprovider. De Data Access Tool (zie afbeelding 3) is qua functionaliteit vergelijkbaar met de 'bekende' ODBC Data Source Administrator, standaard onderdeel van de Windows Administrative Tools.



Afbeelding 1. Applicatielagen HIS 2004

```

[WebMethod]
public DataSet GetDB2Data(
    string userID,
    string password,
    string databaseQuery)
{
    // 1. Connectie string, samengesteld dmv Data Access Tool
    string databaseConnection =
        "Provider=DB2OLEDB;"+
        "Initial Catalog=CONDOR;"+
        "Network Transport Library=TCP;"+
        "Host CCSID=37;"+
        "PC Code Page=1252;"+
        "Network Address=condor;"+
        "Network Port=446;"+
        "Package Collection=MSPKG;"+
        "Default Schema=MGB2004;"+
        "Process Binary as Character=False;"+
        "Units of Work=R UW;"+
        "Default Qualifier=MGB2004;"+
        "DBMS Platform=DB2/AS400;"+
        "Persist Security Info=True;"+
        "Connection Pooling=True;";

    // 2. Gebruikersnaam en wachtwoord worden
    // dmv parameters aangeleverd
    databaseConnection += "User ID = " + userID;
    databaseConnection += ";Password = " + password;

    // 3. Instantieer Db2 connectie object.
    MsDb2Connection myConnection = new MsDb2Connection(databaseConnection);

    // 4. Open verbinding naar mainframe database.
    myConnection.Open();

    // 5. Gebruik een MsDb2DataAdapter om gegevens op te halen.
    // De parameter SQLString is een embedded SQL commando waarmee
    // gegevens opgehaald worden.
    MsDb2DataAdapter myDataAdapter = new MsDb2DataAdapter(databaseQuery,
        myConnection);

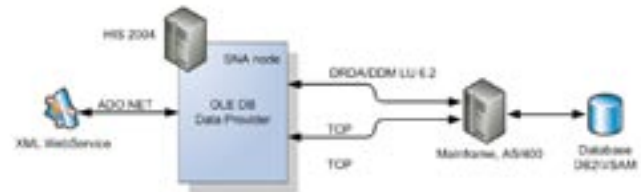
    // 6. Creeer object
    DataSet myDataSet = new DataSet();

    // 7. Vul myDataSet met gegevens vanuit mainframe.
    myDataAdapter.Fill(myDataSet);

    // 8. Sluit connection om deze vrij te geven
    myConnection.Close();
    return myDataSet;
}

```

Codevoorbeeld 1. Implementatie webservice door gebruik additionele .NET-assemblies



Afbeelding 2. Data-integratie met Host Integration Server en .NET

De linker treeview (1) toont folders met connections-strings gesorteerd op type waarmee je naar een bepaalde connection-string kun navigeren. De toolbar (2) bevat knoppen met de volgende mogelijkheden (van links naar rechts):

- Aanmaken, openen, bewerken, tonen en testen van een databaseverbinding (zogenaamde connection-string).
 - Aanmaken van een installatiepakket.
 - Uitvoeren van een voorbeeld-query op basis van de databaseverbinding.
 - Converter een specifieke dataconnectie-string naar een ander type.
- Het onderste paneel (3) toont de geselecteerde databaseverbinding 'MGB_CONDOR'.

Het configureren van de databaseverbinding

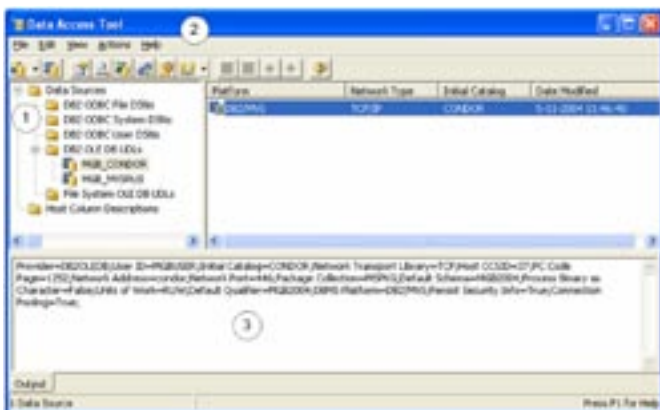
De Data Access Tool bevat een wizard voor het aanmaken en bewerken van een databaseverbinding. De configuratie van zo'n verbinding bestaat uit een aantal instellingen. De eerste configuratie betreft platform en netwerkprotocol.

In het voorbeeldscenario wordt een koppeling tussen Windows-platform en AS400-platform op basis van het TCP/IP-protocol gerealiseerd. Afbeelding 4 geeft deze keuzes weer. Naast TCP/IP kan ook SNA als protocol toegepast worden. IBM mainframesystemen zijn gebaseerd op System Network Architecture (SNA) die specifieke netwerkprotocoldefinities bevat waaronder SNA LU 6.2. Sinds 5 à 8 jaar is ook TCP/IP geïntroduceerd in de IBM mainframewereld. Vervolgens worden het netwerkadres, de database en de initiële catalogus ingesteld, vergelijkbaar met andere databaseverbindingen.

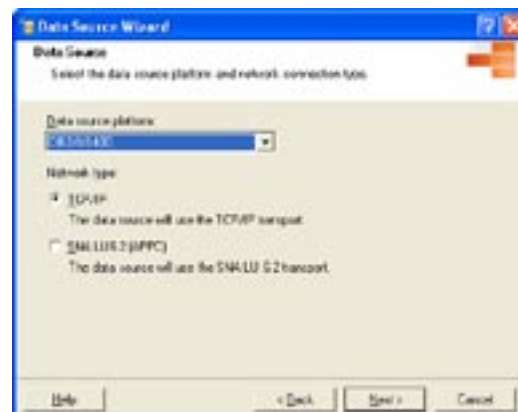
Configuratie 'Locale'

Een ander aspect van de configuratie is het instellen van de 'Locale'. Wanneer je gegevens uitwisselt tussen verschillende platformen is conversie een fenomeen waar je snel mee geconfronteerd wordt. Het Win32/x86 platform gebruikt ASCII en UNICODE voor de opslag van tekst/karaktergegevens. IBM mainframesystemen gebruiken andere typen karaktersets waardoor conversie noodzakelijk is. Het opgeven van de juiste 'locale' doe je met de Data Source Wizard, zie afbeelding 5.

Naast deze vertaling kan het ook noodzakelijk zijn om specifieke datatypes te converteren. Bepaalde datatypes die op het Windows-



Afbeelding 3. HIS Data Access Tool



Afbeelding 4. Keuze datasource-platform en netwerk

platform gebruikelijk zijn, komen niet per definitie overeen met mainframe-datatypes. De onderliggende hardware, processor-technologie en besturingssysteem zijn over het algemeen volkomen verschillend. Hierdoor is de opslag van specifieke datatypes afwijkend. Op databaserecordniveau biedt HIS een mogelijkheid om dit soort conversies te configureren. Dit aspect is verder niet uitgewerkt.

Afronding configuratie

Tenslotte wordt de databaseverbinding afgerond met configuratie van de beveiliging. Je hebt twee alternatieven om de databaseverbinding te beveiligen. Door middel van gebruikersnaam en wachtwoord of door toepassing van Enterprise Single Sign On (SSO). In dit voorbeeld is gebruik gemaakt van een eenvoudige combinatie van gebruikersnaam en wachtwoord. Een nadeel van deze aanpak is dat de opslag van de gegevens beveiligd moet worden. Een alternatief is toepassing van SSO. Dit is een onderdeel van HIS en implementeert een master-secret database. Dit is een specifieke SQL Server-database waarin accountgegevens veilig opgeslagen worden.

Implementatie van XML webservice

Na configuratie van de databaseverbinding wordt het tijd voor een stukje C#-code. Door installatie van HIS zijn er in Microsoft Visual Studio 2003 een aantal additionele .NET-assemblies beschikbaar. Deze assemblies gaan we gebruiken in codevoorbeeld 1. Door middel van C# wordt een webservice geïmplementeerd.

Met deze code is een eenvoudige webservice geïmplementeerd. In feite is het ophalen van gegevens uit een DB2-database vergelijkbaar met het ontwikkelen van een applicatie met een willekeurige andere database. Hierdoor is de leercurve om met HIS Data Services te gaan werken laag.

HIS Transaction Integrator

Naast de Data Services bevat HIS ook de Transaction Integrator. Dit onderdeel fungeert als intermediair tussen het Windows-platform en het IBM Mainframeplatform. Vergelijkbaar met HIS Data Services vertaalt ook HIS Transaction Integrator hostplatformprotocollen waardoor echte applicatie-integratie mogelijk is.

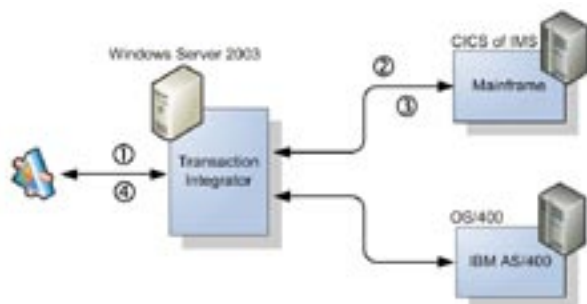
Afbeelding 6 toont de werking van de Transaction Integrator:

1. .NET-client roept een functie van een .NET-component aan.
2. De Transaction Integrator vertaalt de functieaanroep naar functie op het hostplatform.
3. Hostplatform retourneert gegevens.
4. Transaction Integrator vertaalt gegevens van hostplatform en geeft deze door aan .NET-client.

De Transaction Integrator is een intermediair tussen het Windows-platform en het hostplatform waardoor transparante interactie tussen applicaties op heterogene platformen mogelijk is.

Twee smaken: HIP of WIP

Er zijn twee manieren waarop je gebruik kan maken van de Trans-



Afbeelding 6. Werking van HIS Transaction Integrator



Afbeelding 5. Configuratie 'Locale'

action Integrator. De keuze voor één van beide manieren wordt bepaald door het platform dat het initiatief neemt. Een functieaanroep vanaf het Windows-platform verzorgt door middel van de Transaction Integrator dat de applicatieloga op een hostplatform wordt uitgevoerd. Dit scenario noemen we Windows-geïnitieerde verwerking (Windows-Initiated Processing, kortweg WIP). Andersom heet het scenario Host-geïnitieerde verwerking (Host-Initiated Processing, kortweg HIP).

Zowel HIP als WIP bieden interessante mogelijkheden. WIP kun je toepassen wanneer je gebruik wilt maken van bestaande applicaties op een hostplatform. Denk hierbij aan:

- Een ASP.NET website die gebruik maakt van rekenregels op een hostplatform.
- Een webservice die functionaliteit op een hostplatform toegankelijk maakt.

HIP is nieuwe functionaliteit die is geïntroduceerd met de release HIS 2004. Door middel van HIP kunnen bestaande applicaties op een hostplatform worden uitgebreid door middel van .NET-technologie. Voorbeelden hiervan zijn:

- Dataopslag in SQL Server 2000\2005 ten behoeve van applicaties op een hostplatform
- Implementatie van bedrijfsregels in .NET ten behoeve van applicaties op een hostplatform.

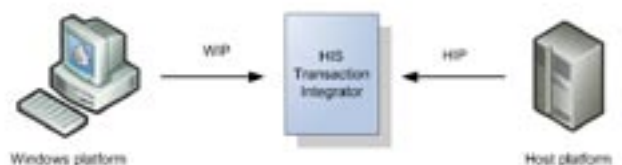
Gebruik van de Transaction Integrator is in beide scenario's vergelijkbaar. Tijdens ontwikkeling bepaal je als ontwikkelaar op welke manier je componenten gebruik maken van de Transaction Integrator.

Ontwikkelen met HIS Transaction Integrator

Gebruik van de Transaction Integrator vanuit een .NET-applicatie vereist het doorlopen van een aantal stappen.

Definitie van een TI-component

Ontwikkeling met de HIS Transaction Integrator is volledig geïntegreerd met Visual Studio 2003. Tijdens de installatie van Microsoft HIS 2004 worden er zogenaamde TI projecttemplates toegevoegd aan de aanwezige templates. Zie afbeelding 8.



Afbeelding 7. WIP- of HIP-integratie

Door middel van de HIS templates in Microsoft Visual Studio 2003 kun je je eigen TI-projecten ontwikkelen.

Een TI-project bestaat uit de volgende onderdelen:

- Een TI-project bestaat uit één of meer TI-componenten. Een TI-component definieert op welke wijze er gecommuniceerd wordt met de hostomgeving (protocol (TCP/IP et cetera.), target (CICS (2) et cetera.), programmeermodel (bijvoorbeeld TRM Link)).
- Een TI-component bestaat uit één of meer interfacedefinities. Een interfacedefinitie bevat de datastructuur van een hosttransactie.

TI-componenten maak je met de TI Designer. Dit is een wizard waardoor je via enkele schermen de communicatieparameters kunt configureren. Het resultaat is een DLL (.NET of COM) die aan je project wordt toegevoegd. Vervolgens kun je interfacedefinities aan een TI-component toevoegen. Het definiëren van een interface kun je manueel of door middel van de 'COBOL import wizard'.

De COBOL-import werkt op basis van een COBOL copybook dat definities van datastructuren bevat. Door middel van de importwizard kun je specifieke datastructuren gebruiken voor je eigen TI interfacedefinitie.

Afbeelding 10 toont een TI Interfacedefinitie ('CedarBank'). Er worden drie eigenschappen (name, ACCNUM, ACCBAL) getoond. Per eigenschap wordt de configuratie getoond; zie afbeelding 10. Deze bestaat uit:

- parameternaam
- .NET datatype
- Richting, dus lezen en/of schrijven
- hostdatatype, deze is overgenomen uit voorgenoemde COBOL-definitie.

Met deze interface kun je bij de CedarBank het saldo (ACCBAL) van een bepaalde rekening (ACCNUM) en rekeninghouder (name) kunnen opvragen. Hiervoor dien je de TI-component te koppelen aan een specifieke host.

Configuratie van TI-componenten

HIS biedt de zogenaamde TI Manager om TI-componenten te koppelen met een specifieke host. Alle TI-componenten - zowel WIP als HIP - worden door middel van deze managementconsole beheerd en geconfigureerd.

Configuratie van een TI-component begint met het toevoegen van een specifieke host die vervolgens wordt weergegeven onder

2 CICS is een transactiemonitor die wordt toegepast op host-omgevingen.



Afbeelding 9. COBOL Import Wizard

```
[WebMethod]
public decimal GetBalance(string AcctName, int AcctNum)
{
    TCP_TRMLink.CedarBank myCedarBank = new TCP_TRMLink.CedarBank();
    decimal AcctBal = 0;
    myCedarBank.CedrBank(ref AcctName, ref AcctNum, ref AcctBal);
    return AcctBal;
}
```

Codevoorbeeld 2.

'Remote Environments'. Vervolgens kun je je TI-component toevoegen onder 'objects'. Tijdens het toevoegen worden servernaam en accountgegevens geconfigureerd en wordt ten slotte een .NET Proxy gegenereerd. Door middel van deze proxy kan je vanuit .NET de transactie aanroepen.

Gebruik van een TI-component en .NET

Het programmeren met .NET en TI-componenten is vergelijkbaar met 'reguliere' .NET-ontwikkeling. Het .NET-project heeft een referentie naar System.Runtime.Remoting en de .NET Proxy van je TI-component nodig. Vervolgens kun je code schrijven die gebruikmaakt van de TI-component.

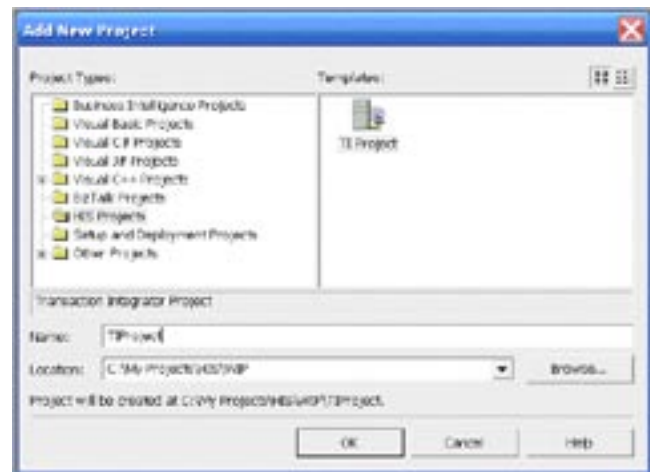
In codevoorbeeld 2 zie je een webmethod-implementatie op basis van de hiervoor geconfigureerde TI-component 'TRMLink' die de interfacedefinitie 'CedarBank' bevat. Deze code maakt door middel van de HIS Transactie Integrator eenvoudig gebruik van een CICS-transactie op een hostomgeving.

HIS Messaging Services

Een van de minst bekende Microsoft-producten is de Microsoft MSMQ-MQSeries Bridge. Message Queuing-technologieën worden vaak toegepast wanneer een betrouwbaar transportprotocol vereist is. Microsoft biedt Message Queuing als onderdeel van het Windows-platform. Door middel van deze technologie kunnen applicaties op basis van berichten informatie uitwisselen. De MSMQ-MQSeries Bridge maakt koppelingen tussen MSMQ en IBM MQ Series mogelijk. Hierdoor kunnen .NET-applicaties op het Windows-platform door middel van XML-berichten informatie uitwisselen met applicaties op andere platformen. De ontwikkelaar kan door middel van MSMQ XML berichten versturen die door middel van de MQBridge met applicaties op andere platformen uitgewisseld worden! Interop tussen platformen wordt op deze manier wel erg eenvoudig.

Integratie van applicaties

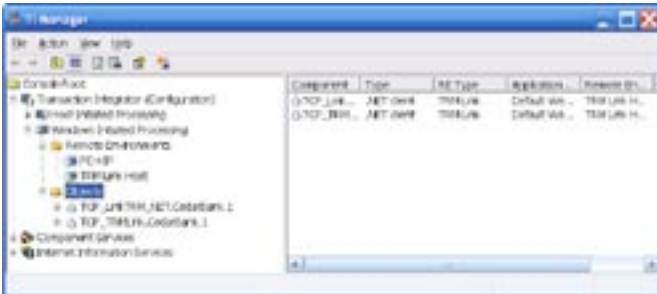
De verschillende onderdelen maken integratie van applicaties op uiteenlopende platformen mogelijk. Belangrijke integratie 'show-stoppers' zoals dataconversie, datatype conversie en protocolconversie worden door middel van Microsoft HIS 2004 opgelost.



Afbeelding 8. HIS Project templates

Name	Type	Direction	Host/URL	App. Service	Port
ACCUA_M	Message	toOut	HL.DMS.LIB.LU	HL.DMS	8080
ACCUA_M	Message	toOut	HL.DMS.LIB.LU	HL.DMS	8080

Afbeelding 10. TI Interfacedefinitie



Afbeelding 11. TI Manager

Afhankelijk van de situatie (soort applicatie, platform en eisen) kun je voor een specifiek scenario kiezen. In de praktijk kun je de volgende vuistregels hanteren:

- Wil je op basis van applicatieloga (of transacties) applicatie's integreren, dan is toepassing van de Transaction Integrator het meest voor de hand liggend.
- Wil je gebruik maken van gegevens op een mainframe\midrange-systeem, dan is toepassing van de Data Services de logische keuze.
- Wil je integreren op messaging-niveau, dan is toepassing van Messaging Services wellicht een geschikte oplossing.

Per situatie zul je bewust moeten afwegen op welk platform je welke functionaliteit gaat implementeren. Bovendien zijn zulke afwegingen vaak afhankelijk van de strategische doelen of beleidskeuzes van een specifieke organisatie.

Arno Hartevelde is consultant bij Microsoft Services Nederland. Zijn e-mail: arno@microsoft.com

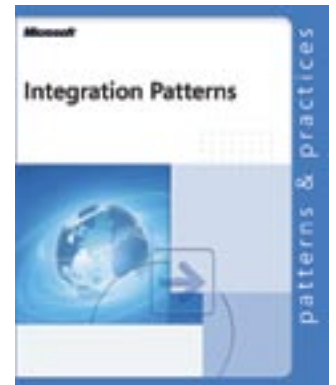
Nuttige internetadressen

- Host Integration Server: www.microsoft.com/hiserver/
- HIS Evaluatie Versie: www.microsoft.com/hiserver/evaluation/trial/default.asp
- Visual Studio Home Page: msdn.microsoft.com/vstudio/
- HIS Development center: msdn.microsoft.com/hiserver/
- HIS Reviewers guide: www.microsoft.com/hiserver/evaluation/productguide/default.asp

(advertentie Microsoft Press)



Enterprise Integration Solutions
 ISBN: 0-7356-2060-1
 Auteurs: Devin Spackman, Mark Speaker
 Pagina's: 368



Integration Patterns
 ISBN: 0-7356-1850-X
 Auteur: Microsoft Corporation
 Pagina's: 432



Inspired by
 Innovation

Specialist in gegarandeerde, professionele softwareontwikkeling op het Microsoft platform. www.quriusetx.nl