Gedistribueerd installeren

Automatisering processen met Omni Installer

Bij een Nederlandse grootbank in Amsterdam worden regelmatig releases van applicaties geïnstalleerd in een multitier omgeving waarbij verschillende Oracle databases, AIX servers en Wintel servers betrokken zijn, ieder met hun eigen specifieke accounts en omgevingen. Het correct opleveren van releases en het uitrollen daarvan is een grote uitdaging. Bij de afdeling Development & Maintenance is een installatietool ontwikkeld om dit proces zoveel mogelijk automatisch te laten verlopen om de kwaliteit en snelheid te verbeteren.

Binnen de bank is een professionele ontwikkelstraat ingericht waarbij software op een gecontroleerde manier van een ontwikkelomgeving via één of meerdere test- en acceptatie-omgevingen naar een productie-omgeving wordt gepromoveerd. Dit wordt de zogenaamde OTAP-straat genoemd.

Als versiebeheer tool wordt gebruik gemaakt van Oracle Designer 10g. Alle op te leveren software wordt opgeslagen in de repository van Designer. Hierin wordt de te releasen software vastgelegd in een configuratie. Zo'n configuratie wordt vervolgens handmatig naar de juiste OTAP-repository gepromoveerd waarna de software in de bijbehorende doelomgeving kan worden geïnstalleerd. Dit gebeurt door de installatie-instructies te volgen die in een begeleidend draaiboek zijn vastgelegd.
Deze manier van software promoveren kent een aantal zwakke punten:

- het handmatig uitvoeren van installatie-instructies aan de hand van een uitgebreid draaiboek is foutgevoelig en tijdrovend. De accuratesse van de uitvoerder is van grote invloed op de kwaliteit van de installatie;
- het uitvoeren van dit soort installatiewerkzaamheden is niet uitdagend voor de werknemer;
- door een grote diversiteit aan applicaties en omgevingen gaat het overzicht verloren met betrekking tot welke release in welke omgeving is geïnstalleerd.

Vanuit de organisatie werden deze gevaren onderkend en kwam men met een innovatief en uitdagend voorstel: ontwikkel een installatietool waarbij de applicatiebeheerder met één druk op de knop vanaf zijn eigen werkstation alle software zonder verder handmatig ingrijpen kan uitrollen naar alle verschillende omgevingen. Daarbij dient tevens een tracking en tracing module ingebouwd te worden waardoor op elk moment een overzicht kan worden verkregen wie welke software waar heeft geïnstalleerd en met welk resultaat.

"A new tool is born: The Omni[1] Installer"

De Omni Installer

Aan de Omni Installer werden de volgende eisen gesteld:

- de huidige releasemethodiek, waarbij vanuit Designer configuraties worden gepromoveerd, moet worden ondersteund. Dit betekent dat Designer het uitgangspunt is voor het uitrollen van software. Releases worden vastgelegd binnen configuraties en/of een workarea en worden OTAPs-gewijs gepromoveerd. Binnen Designer heeft elke OTAP-fase zijn eigen repository. Ten behoeve van de OTAP-straat kunnen meerdere doelomgevingen beschikbaar zijn. Er zijn dus per applicatie meerdere test- en acceptatie-omgevingen waarin de software naar keuze moet kunnen worden geïnstalleerd;
- het tool moet voldoen aan de strenge beveiligingsvoorwaarden zoals die door IRM (Information Risk Management) zijn vastgelegd;
- single logon. De gebruiker hoeft slechts één maal zijn credentials op te geven waarna op alle servers en omgevingen zonder opgaaf van gebruikersnamen en wachtwoorden de installatie kan worden uitgevoerd;
- alle gebruikte software moet kunnen worden uitgerold. Naast in-house ontwikkelde applicaties moet ook 'third party software' via de Omni Installer kunnen worden uitgerold;
- er dient een centrale logging te zijn. Vanwege het feit dat bij een installatie meerdere servers betrokken kunnen zijn (data-

1 Omni, van omnis (Lat.): al, geheel, alle(n)

base server, middletier, scheduling systeem etc) treedt het probleem op dat logfiles met informatie over de geïnstalleerde modules over verschillende servers worden verspreid. De Omni Installer dient te zorgen voor een gecentraliseerde logfile waarin alle loginformatie met betrekking tot de installatie wordt verzameld;
- middels een tracking en tracing module dient vastgelegd te worden welke software door wie, waar, wanneer en met welk resultaat is geïnstalleerd;
- een release mag pas naar een volgende omgeving binnen de OTAP-straat worden gepromoveerd wanneer de huidige omgeving is getest en goedgekeurd. Binnen Omni Installer wordt hiervoor een elektronische handtekening opgenomen waarmee de release op basis van systeem- en/of acceptatie testen wordt goedgekeurd;
- een installatie moet achteraf kunnen worden gecontroleerd. Dat betekent dat alle objecten in een release moeten kunnen worden gecontroleerd tegen de versie zoals die in de Designer repository is vastgelegd. Dat geldt dus niet alleen voor de geïnstalleerde bestanden maar ook voor Oracle Webforms-modules, menu's, libraries en voor alle database-objecten (behalve sequences);
- backup en restore van bestanden en database objecten moet ondersteund worden;
- installatie-procedures voor nieuwe applicaties moeten eenvoudig kunnen worden toegevoegd.

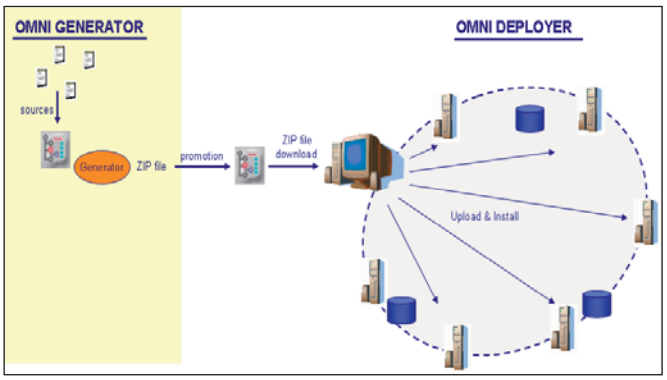
De hulpmiddelen

De Omni Installer moest worden geïmplementeerd, gebruik makend van de hulpmiddelen die op dat moment binnen de afdeling beschikbaar waren.

Hierdoor kon gebruik worden gemaakt van de volgende tools en utilities:
- Oracle Designer 10g
- Perl
- Ant
- Log4J
- Webforms GUI
- Oracle10g
- Java
- Oracle OCI

Omni Generator en Deployer

De Omni Installer kent twee onderdelen. De Omni Generator en de Omni Deployer. De Omni Generator wordt gebruikt door de ontwikkelaar. Op basis van een workarea of één of meerdere configuraties in Designer, waarin de de software van de nieuwe release is gebundeld, wordt met behulp van de Omni Generator vastgelegd hoe, waar en door wie deze software uiteindelijk wordt uitgerold.



Afbeelding 1. De Omni Installer

Met één druk op de knop genereert de Omni Generator een releasebestand met daarin alles wat nodig is om de release uit te rollen.
Naast alle sources bevat het releasebestand ook een XML bestand dat dient als elektronisch draaiboek. Dit XML bestand wordt met behulp van XSLT omgezet naar een Ant build file en een installatiehandleiding voor het geval dat de release, door bijvoorbeeld een technische storing, met de hand moet worden geïnstalleerd.
Het releasebestand wordt in Designer opgenomen. Vanaf dat moment is de release beschikbaar om uit te rollen.
Voor een installatie in een test-, acceptatie- of productie-omgeving zal het releasebestand vanuit de desbetreffende OTAP-repository moeten worden gedownload naar het werkstation waarna de Omni Installer kan worden gestart. Met behulp van het elektronische draaiboek in het releasebestand weet de Omni Installer op welke servers welke software dient te worden geïnstalleerd.

Het datamodel

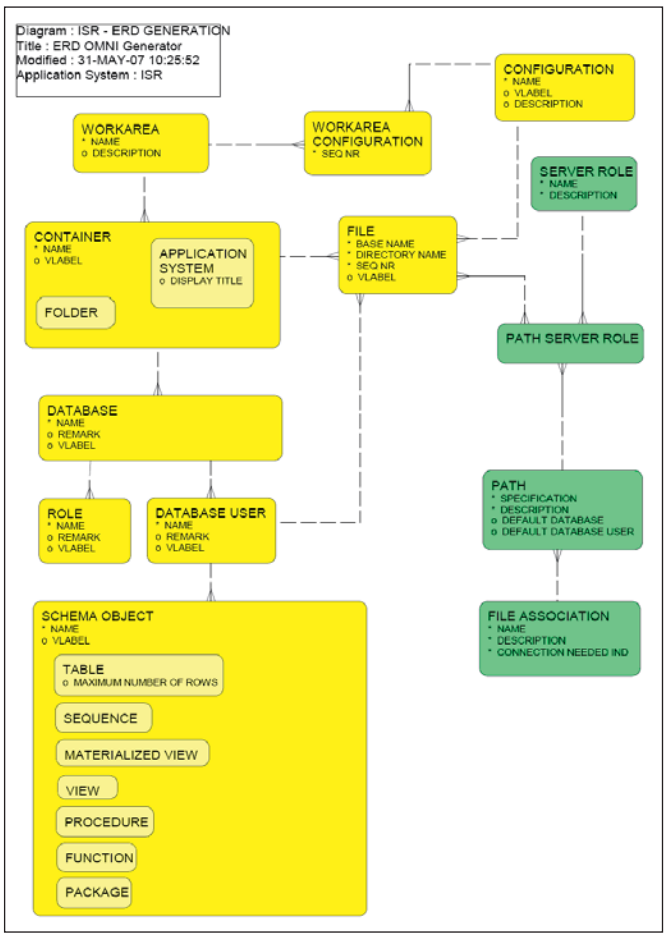
Het datamodel van de Omni Generator is afgeleid van het datamodel van de Designer repository.

De hoofdentiteit is WORKAREA, die overeenkomt met een Designer workarea. Een workarea is voor de Omni Generator te vergelijken met een release. Alle release-informatie wordt vastgelegd in dit datamodel, zoals:
- de configuraties (entiteiten WORKAREA CONFIGURATION en CONFIGURATION) waarop de workarea eventueel is gebaseerd;
- de containers (entiteit CONTAINER met subentiteiten APPLICATION SYSTEM en FOLDER) die applicatie-informatie bevatten;
- de databases, rollen en databasegebruikers (entiteiten DATABASE, ROLE en DATABASE USER) van een applicatie;
- de schemaobjecten (entiteit SCHEMA OBJECT) die alle mogelijke databaseobjecten bevatten zoals tabellen, views en packages;

- de bestanden (entiteit FILE) die horen bij (de folders van) een applicatiesysteem of die gegenereerd zijn door andere bestanden van het applicatiesysteem (zie kopje "Generator Batch Utility bestanden").

Bovenstaande entiteiten zijn dynamisch (de inhoud wordt elke release gewijzigd), dit in tegenstelling tot de volgende entiteiten:

- de rol van een server (entiteit SERVER ROLE) geeft aan wat voor activiteiten op een server kunnen plaatsvinden, bijvoorbeeld een databaseserver of een applicatieserver (per server zijn meerdere rollen mogelijk);
- een bestandsassociatie (entiteit FILE ASSOCIATION) die aangeeft wat voor programma (of actie) gekoppeld is aan een bestand, bijvoorbeeld het starten van SQL*Plus of het compileren van een Forms bestand met als input dat bestand;
- een pad (entiteit PATH) is een manier om bestandsspecificaties via wildcards te koppelen aan een associatie, bijvoorbeeld "*/ins/cdsddl.sql" wordt gekoppeld aan SQL*Plus en "*/src/.fmb" wordt gekoppeld aan de Forms compiler;
- een intersectie-entiteit PATH SERVER ROLE die de mogelijke paden en bestandsassociaties koppelt per rol van de server, omdat niet alle mogelijkheden zinvol zijn (een Forms bestand kan niet worden gecompileerd op een server als het geen applicatieserver is).



Afbeelding 2. Het Omni datamodel

Generator

Het is de taak van de Generator om een release samen te stellen en deze als bestand (in Zip-formaat) beschikbaar te stellen. Dit releasebestand kan dan door de Deployer geïnstalleerd worden op een of meerdere servers.

De Generator is een applicatie met de volgende onderdelen:

- een Ant build file;
- een Forms GUI die onder water de Ant build file aanroept via de Oracle Forms WebUtil library.

De input die de gebruiker moet leveren is:

- de naam van een Designer workarea;
- de namen van de Designer configuraties waaruit de workarea moet bestaan (als geen configuraties worden opgegeven dan moet de workarea al bestaan).

De acties die – via de Ant build file – worden uitgevoerd zijn:

1. het (opnieuw) aanmaken van een Designer workarea op basis van de opgegeven configuraties (als die opgegeven zijn);
2. het downloaden van bestanden van de Designer workarea naar een build directory;
3. het uitvoeren van Generator Batch Utility bestanden met het Designer programma dwzrun61 (zie sectie Generator Batch Utility bestanden);
4. het vullen van de volgende Omni tabellen op basis van de informatie van de Designer workarea:

- WORKAREA
- WORKAREA CONFIGURATION en CONFIGURATION
- CONTAINER
- DATABASE
- ROLE
- DATABASE USER
- SCHEMA OBJECT

5. het vullen van de tabel FILE op basis van de gedownloade en gegenereerde bestanden in de build directory.

De laatste stap is de lastigste. Ten eerste moet worden bepaald tot welk pad (PATH) een bestand hoort. Dit gebeurt op basis van de beste (langste) match; `"*.sql"` is een mindere match dan `"*/ins/*.sql"`. Een SQL bestand kan dus afhankelijk van zijn locatie wel of niet door SQL*Plus gestart worden. Dit is ook noodzakelijk omdat een SQL bestand soms alleen maar tijdens de installatie nodig is en soms alleen maar als de applicatie in bedrijf is (bijvoorbeeld batchbestanden). Ten tweede moet worden bepaald welke logische databasegebruiker gekoppeld moet worden aan een bestand. Het is namelijk mogelijk dat een applicatiesysteem meerdere databasegebruikers heeft met elk

een deel van de gehele applicatie. De attributen DEFAULT DATABASE en DEFAULT DATABASE USER in entiteit PATH zorgen ervoor dat een databasegebruiker kan worden gekoppeld aan een bestand. Merk op dat de database en databasegebruiker logische namen zijn zoals ze in Designer zijn gedefinieerd. Pas tijdens de installatie worden ze door de fysieke databasenaam en schemanaam vervangen.

Vervolgens wordt alle informatie aan de gebruiker getoond die dan nog de mogelijkheid heeft om de volgorde te veranderen, bestanden te verwijderen, andere databasegebruikers of bestandsassociaties te koppelen aan een bestand. Al deze gewijzigde instellingen worden bewaard voor een volgende release (met dezelfde workarea als naam). Als alles naar wens is kan de gebruiker het releasebestand aanmaken.

Generator Batch Utility bestanden

Een van de minst gebruikte onderdelen van Designer is de Generator Batch Utility (GBU). De GBU kan worden aangeroepen via de Batch Generate Wizard (Oracle Design Editor -> Tools -> Batch Generate).

De GBU stelt je in staat om bestanden te genereren van de volgende types:
- Client Applications (Forms en Reports)
- Help Systems
- Server Model (tabellen, sequences, views, packages, synoniemen, enzovoorts)
- Database Administration Objects (rollen en database)
- Module Component API
- Table API
- Reference Code Tables

Voordelen van de GBU:
- er kan een generatiebestand worden opgeslagen (via Save Generation Details in laatste dialoogscherm);
- je hoeft bestanden niet meer handmatig te genereren als je een generatiebestand gebruikt (gewoon starten via de command prompt);
- je kunt ook een schema incrementeel bijwerken (genereer Server Model type tegen een schema).

Nadelen van de GBU:
- het GBU bestand is binair en daardoor niet (goed) te lezen of aan te passen. Dus als je een nieuwe samenstelling van te genereren objecten wilt, dan moet je de GBU Wizard opnieuw starten;
- de Windows registry wordt niet opgeslagen in de GBU terwijl die wel relevant is en kan verschillen per applicatiesysteem;
- als je bestanden van het type Server Model genereert tegen een schema, dan wordt het gegenereerde bestand ook gelijk uitgevoerd, dus kun je de gegenereerde scripts niet nog een keer uitvoeren (tenzij je ervoor zorgt dat het doelbestand niet te genereren is door bijvoorbeeld eerst een directory met exact dezelfde naam aan te maken);
- het genereren via een GBU bestand kan lang duren, dus je start bij voorkeur GBU bestanden parallel op maar je mag niet gelijktijdig twee GBU bestanden draaien met dezelfde uitvoerbestemming omdat dat tot fouten leidt, dus je moet zelf zorg dragen voor een bepaalde vorm van locking.

Ondanks de nadelen is een GBU bestand toch noodzakelijk als je zoveel mogelijk bestanden automatisch wilt genereren. Een van onze toekomstige uitbreidingen aan de Omni Generator zal zijn om een toepassing te schrijven die GBU bestanden genereert op basis van de inhoud van een Designer workarea en die ook de Windows registry settings correct definieert voordat een GBU bestand wordt gestart. Pas dan kunnen we een start maken met Continuous Integration in een Oracle ontwikkelomgeving (zie sectie Referenties).

Deployer

Zoals eerder is beschreven, bevat de Omni Generator alle logische informatie over hoe, waar en door wie de software moet worden geïnstalleerd:
- hoe wordt bepaald door de bestandsassociatie zoals die in Omni Generator gekoppeld is aan het te installeren object;
- waar wordt bepaald door de logische server die aan het object is gekoppeld;
- wie de software installeert wordt bepaald door een Operating Systeem of database-account property. Over het gebruik van properties volgt hieronder meer.

De Omni Deployer moet de vertaalslag maken van de logische naar de fysieke omgeving. Deze fysieke omgeving is voor elke OTAP-fase verschillend. Er worden andere servers gebruikt, andere Unix- en database-accounts, andere folderstructuren, enzovoorts.

Deze fysieke gegevens worden voor elke omgeving vastgelegd in properties (die opgeslagen worden in een properties tabel). Er wordt onderscheid gemaakt tussen server level properties en application level properties. Zoals de namen al impliceren gelden de server level properties voor een specifieke server en gelden de application level properties voor een specifieke applicatie op een bepaalde server.

Server level properties

Op server level wordt vastgelegd op welke manier een connectie moet worden opgezet naar die server en waar op die server de Omni Deployer software staat geïnstalleerd.

Application level properties

Op applicatieniveau worden drie soorten properties vastgelegd:

- server rol properties. Per server wordt per applicatie vastgelegd welke rol deze server speelt voor die applicatie. Eén server kan voor een bepaalde applicatie bijvoorbeeld zowel de database- als application server zijn;
- database properties. Hierbij worden database namen en schema's vastgelegd waarin de database-objecten moeten worden gecreëerd;
- O/S properties. De O/S properties bepalen op welke directories en onder welk account de bestandsassociaties worden uitgevoerd. M.a.w. welke user installeert welke software op welke locatie.

Op dit moment worden de volgende soorten server rollen onderscheiden:

- database. Op deze server draait de database en staat tevens de software van de applicatie geïnstalleerd;
- application. Op deze server draait de middletier software. Het gaat daarbij om Webform-applicaties;
- autosys. Autosys is het scheduling systeem dat gebruikt wordt bij de bank. Deze software wordt op een specifieke server geïnstalleerd;
- workstation. Met het workstation wordt de computer aangegeven vanwaar de gebruiker de Omni Deployer heeft opgestart (localhost).

Elke applicatie heeft zijn eigen database(s) met bijbehorende database accounts. Op logisch niveau zijn in de Omni Generator de namen overgenomen uit Designer. Deze namen kunnen natuurlijk afwijken van de fysieke namen. De vertaling van logisch naar fysiek wordt via de database properties vastgelegd.

Wat nu nog resteert zijn de properties die de locaties en accounts op het O/S specificeren waaronder de software geïnstalleerd dient te worden. Van de locaties waar de bestanden moeten worden geïnstalleerd worden alleen de home directories vastgelegd. Het uiteindelijke pad waar een bestand komt te staan wordt afgeleid van de folder structuur zoals die in Designer is vastgelegd. De folders zijn de relatieve paden vanaf de home directory.
Deze properties dienen eenmalig te worden ingesteld waarna de Omni Deployer aan de slag kan.

De Omni Deployer pakt het door de Omni Generator gebouwde releasebestand op en bepaalt aan de hand van het elektronische draaiboek in combinatie met de properties op welke servers de software dient te worden uitgerold.

Op elke server wordt een tijdelijke installatiefolder aangemaakt waar het releasebestand naar toe wordt geupload en vervolgens wordt uitgepakt. De Omni Generator heeft middels XSLT een Ant build file gegenereerd die op elke server kan worden gestart. Via deze build file wordt op basis van de bestandsassociatie bepaald of de software vanuit de tijdelijke installatiefolder kan worden geïnstalleerd of dat de software eerst naar de doelomgeving moet worden gekopieerd alvorens het te installeren.

Wanneer één server meerdere rollen vervult waarbij verschillende O/S accounts betrokken zijn wordt op die server de build voor beide accounts parallel gestart. Omdat de volgorde van de installatie, zoals die in de Omni Generator is gespecificeerd, behouden moet blijven, communiceren de parallelle build sessies met elkaar om de vastgelegde volgorde te garanderen. Dit doet de Omni Installer volgens het check-process-wait principe:

- check: de check fase controleert of het huidige O/S-account het object op de server moet installeren, oftwel of process of wait of geen van beide moet worden uitgevoerd;
- process: het object wordt geïnstalleerd door het huidige O/S-account;
- wait: het object wordt wel op deze server geïnstalleerd, maar niet door het huidige O/S-account. Wacht tot het object is geïnstalleerd.

De feitelijke installatie middels een bestandsassociatie is geïmplementeerd in een Ant target. Elke bestandsassociatie kent een viertal Ant targets:

- backup. De backup target moet er voor zorgen dat er voldoende informatie van het te installeren object is vastgelegd om eventueel een uninstall te kunnen uitvoeren;
- install. De install target bevat de aanroep naar de feitelijke installatieprocedure. Dit kan bijvoorbeeld een SQL*Plus aanroep zijn voor het aanmaken van een tabel;
- uninstall. De uninstall target draait de installatie terug op basis van de gegevens die door de backup target zijn weggezet;
- check. De check target bevat instructies om het geïnstalleerde object achteraf te kunnen vergelijken met het te installeren object uit het releasebestand. Dit kan betekenen dat bestanden met elkaar worden vergeleken maar ook kunnen revisienummers, zoals die in de Designer repository zijn vastgelegd, worden vergeleken met de revisies in de runtime omgeving. De Designer revisies zijn opgenomen in het elektronische draaiboek.

In de build file zijn niet alleen de installatie-instructies opgenomen maar ook de aanroepen naar de tracking en tracing module. Elke keer als een installatie wordt uitgevoerd, worden de installatiegegevens (wie, wat, wanneer en waar) bijgehouden. Op basis van deze gegevens zijn rapportages gebouwd die over de hele OTAP-straat heen, snel en eenvoudig, inzicht geven in welke releases waar geïnstalleerd zijn en met welk resultaat.

Adv.

De logfiles die tijdens het installeren op de diverse servers worden aangemaakt, worden aan het eind van de installatieprocedure overgehaald naar het werkstation vanwaar de Omni Deployer is gestart. Daar wordt al een overkoepelende logfile bijgehouden via de Omni Deployer GUI. Door het instellen van een bepaald loglevel bevatten de logfiles meer of minder informatie.

Wanneer er tijdens een installatie een fatale fout optreedt, wordt meteen gestopt en kan de gebruiker eventueel het probleem oplossen. Wordt de fout veroorzaakt door fouten in de release dan dient de ontwikkelaar de fout te herstellen en een nieuwe releasebestand aan te leveren. Op het moment dat de problemen opgelost zijn kan de Omni Deployer opnieuw worden gestart. Hierbij kan ervoor worden gekozen om door te gaan op het punt waar tijdens de vorige run de fout optrad of om de installatie volledig opnieuw uit te voeren.

Tips en trucs

Aanmaken package met objecten die via een rol zijn toegekend

De Repository objecten zijn via een rol toegekend aan gebruikers. Dit betekent dat je niet zomaar een package kunt maken waarin die objecten gebruikt worden, omdat dan objecten direct aan de eigenaar moeten zijn toegekend. Het package is wel aan te maken als je gebruik maakt van authid current_user en dynamisch SQL. De clausule authid current_user wordt echter niet door Designer ondersteund. De truc is om free format te gebruiken en het eerste deel van de packagespecificatie als volgt te definïeren:

```
/* This script needs to be run by SQL*Plus */
end isr_generator;
.

CLEAR BUFFER

REMARK This package needs AUTHID CURRENT_USER because of the access to
REMARK Repository objects via role CKR_DES6i. Now these objects need
not
REMARK be granted directly when dynamic SQL is used.

create or replace package isr_generator authid current_user is
```

Time-out in een Forms sessie

De installatie wordt via WebUtil als een achtergrondproces gestart. Aangezien een installatie lang kan duren, geeft Forms uiteindelijk een time-out en is de sessie weg. Dat is gemakkelijk te verhelpen door continu een dialoogbox te starten zolang de installatie (het achtergrondproces) loopt.

Logging

Voor de logging tijdens de installatie is de Log4J database appender gebruikt (zie de referenties aan het eind van dit artikel). Ant kan namelijk zijn logging via een listener verwerken, in ons geval toevoegen van records aan een databaseview, zodat we de logging in de GUI kunnen tonen. De view is niet gebaseerd op een tabel, maar op een table function (pipelined package functie):

```
function show
( p_session_name in varchar2 default get_session_name
, p_timeout in integer default get_timeout
)
return log4j_tabtype
pipelined;

create view isr_v_log4j as
select  t.*
from    table(isr_log4j.show) t;
```

Op de view is een 'instead of trigger' gedefinieerd zodat een insert omgezet wordt in het schrijven naar een database pipe. In de show functie wordt deze database pipe gelezen. Op deze manier wordt voorkomen dat een logtabel volloopt, terwijl we alleen maar geïnteresseerd zijn in de logging van de huidige sessie.

Conclusies

De Omni Installer heeft ervoor gezorgd dat de tijd die is gemoeid met samenstellen van releases flink is verminderd. Zeker als GBU bestanden gebruikt worden is een release heel snel samen te stellen.
De grootste tijdwinst wordt echter gehaald tijdens een installatie omdat alles automatisch wordt uitgevoerd. Grootste voordelen zijn betrouwbaarheid, snelheid, herstartbaarheid, logging en traceerbaarheid. Sinds de invoering van de Omni Installer zijn er aantoonbaar minder problemen als gevolg van installaties.

De bouw van een hulpmiddel als de Omni Installer toont wederom de kracht van automatisering aan: saai, arbeidsintensief, foutgevoelig en tijdrovend werk wordt door de techniek in korte tijd foutloos en eenduidig uitgevoerd.

Referenties

[1] Continuous Integration, http://en.wikipedia.org/wiki/Continuous_Integration
[2] Log4J database appender, http://www.dankomannhaupt.de/projects/jdbcappender

Gert-Jan Paulissen is werkzaam bij Transfer Solutions