

# Technische documentatie Edubox 2.0 Tools

Citation for published version (APA):

Martens, H. (2001). *Technische documentatie Edubox 2.0 Tools*.

## Document status and date:

Published: 22/02/2001

## Document Version:

Peer reviewed version

## Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

## General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

## Take down policy

If you believe that this document breaches copyright please contact us at:

[pure-support@ou.nl](mailto:pure-support@ou.nl)

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 17 Jun. 2024

Open Universiteit  
[www.ou.nl](http://www.ou.nl)



**Onderwijstechnologisch expertisecentrum Otec  
Open Universiteit Nederland**

# **Technische Documentatie Edubox 2.0 Tools**

**MDAmanager, Roleplanner, Runplanner,  
Legacykoppelingen**



## **Otec werkdocumenten**

De Open Universiteit Nederland ontwikkelt en verzorgt open hoger afstandsonderwijs en is tevens een centrale partner in het consortium voor vernieuwing van het Hoger Onderwijs. Onderwijstechnologische vernieuwingen krijgen daarbij speciale aandacht. Binnen de Open Universiteit Nederland is de onderwijskundige en onderwijstechnologische expertise samengebracht in het Onderwijstechnologisch expertisecentrum (Otec). Dit centrum vervult taken in het kader van ontwikkeling, vernieuwing, onderzoek en evaluatie van het onderwijs van de Open Universiteit Nederland en haar consortiumpartners. Deze taken worden veelal uitgevoerd in nauwe samenwerking met directoraten en faculteiten van de OU en/of samenwerkingspartners in het consortium.

De werkzaamheden van het Otec leiden regelmatig tot producten, zoals voorlopige onderzoeksresultaten, strategische stellingnames en overwegingen op projektniveau, functionele specificaties van informatiesystemen, en dergelijke. Deze worden door het Otec vastgelegd en onder de aandacht gebracht in een reeks werkdocumenten waarvan het voorliggende deel uitmaakt.

Naast deze reeks werkdocumenten geeft het Otec een reeks rapporten uit met een meer geformaliseerd en/of afgerond karakter.

De Otec rapporten en sommige werkdocumenten kunnen worden besteld bij:

Open Universiteit Nederland  
secretariaat Otec  
Postbus 2960  
6401 DL Heerlen  
Tel. 045-5762406 of 5762942  
Fax. 045-5762802

Otec werkdocument XXX

of opgehaald via Internet: <http://www.ou.nl/otec>

Onderwijs<sup>s</sup>technologisch expertisecentrum (Otec)  
Open Universiteit Nederland

**Technische documentatie Edubox 2.0 Tools**

MDAmanager, Roleplanner, Runplanner, Legacykoppelingen

## Colofon

Titel:	Technische documentatie Edubox 2.0 Tools
Subtitel:	MDAmanager, Roleplanner, Runplanner, Legacykoppelingen
Auteurs:	Harrie Martens
Projectleiding:	Ghislain Rodenburg
Projectondersteuning:	Mieke Haemers
Uitgifte:	Otec
Datum druk:	22 februari 2001
Adviezen:	-

© 2004, Onderwijstechnologisch expertisecentrum,  
Open Universiteit Nederland, Heerlen.

Behoudens uitzonderingen door de wet  
gesteld mag zonder schriftelijke toestemming  
van de rechthebbende(n) op het auteursrecht  
niets uit deze uitgave worden verveelvoudigd  
en/of openbaar gemaakt door middel van  
druk, fotokopie, microfilm of anderszins,  
hetgeen ook van toepassing is op de gehele of  
gedeeltelijke bewerking.

# Inhoudsopgave

## **Inleiding 6**

### **MDA manager 7**

Functionaliteit 7

Applicatie architectuur 7

Interface 8

MDAManager.exe 10

*Het configuratiebestand 11*

*Ophalen serverdata 11*

*Het Userinterface 12*

*Bestanden 14*

ELOSRV20.DLL 16

*Interfaces 16*

*Bestanden 17*

### **Roleplanner 18**

Functionaliteit 18

Applicatie architectuur 18

Interface 19

*Getrunlist 20*

*Getrun 20*

*Store 24*

Client tools 24

*Het configuratiebestand 25*

*Ophalen serverdata 25*

*Het userinterface 26*

*Bestanden 27*

ELORPSRV20.DLL 28

*Interfaces 28*

*Bestanden 28*

### **Runplanner 29**

Functionaliteit 29

Applicatie architectuur 29

Interface 30

*Server.asp 31*

*Ebrelease 36*

*Ebpublication 36*

Client tool 36

*Het configuratiebestand 37*

*Ophalen serverdata 37*

*Het userinterface 37*

*Bestanden 41*

### **Legacy koppelingen 43**

Inleiding 43

ParsePropDefs 44

ImportLegacy 46

ImportInschrijvingen 48



## Inleiding

De door de Open Universiteit Nederland ontwikkelde elektronische leeromgeving Edubox bestaat uit twee hoofdonderdelen. Het eerste onderdeel is de module die de opslag van het onderwijsmateriaal regelt, de MDA. Het tweede onderdeel draagt zorg voor het afspelen van deze content op een medium naar keuze. In de huidige implementatie is dit de eoplayer voor het web. Beide omgevingen zijn ontwikkeld rondom een centrale dataopslag. Voor de MDA is deze opslag de MDA-database. De webplayer gebruikt hiervoor de dossier database.

In deze documentatie wordt het aanmaken van het in eml (Educational Modelling Language) gecodeerde onderwijsmateriaal niet tot Edubox gerekend.

Een deel van het beheer en gebruik van deze databases is een integraal onderdeel van de component die gebruik maakt van de betreffende database. Een voorbeeld hiervan is de manipulatie van de eml-properties in de dossierdatabase. Deze database interacties zijn gedefinieerd vanuit de eml en worden volledig door de player geïmplementeerd.

Andere functionaliteiten zijn niet door de player geïmplementeerd. Een voorbeeld hiervan is het aanmaken van een run of het toekennen van personen aan één of meerdere rollen. Verder zijn er in Edubox ook functies geïmplementeerd die geen user-interface kennen. Dit betreft overwegend in Omnimark gemaakte programma's die de aangeleverde eml-content verwerken. Denk hierbij aan import, validatie en publicatie van het eml-materiaal.

De MDAManager is een tool die ontwikkeld is om verschillende functionaliteiten rondom het contentbeheer aan te bieden.

De runplanner is een tool die ontwikkeld is om een runplanner in staat te stellen om runs te beheren. Dit houdt ondermeer in dat de runplanner in staat wordt gesteld om runs aan te maken, publicaties te maken en personen aan een run te koppelen.

De roleplanner is een tool die ontwikkeld is om een roleplanner in staat te stellen om deelnemers aan een specifieke run van een unit-of-study aan de verschillende subrollen toe te kennen. Ook kan de roleplanner met deze tool verschillende publicaties aan een rol, en daarmee aan een groep gebruikers toekennen.

In de komende hoofdstukken worden achtereenvolgens de MDA manager, Roleplanner en runPlanner tool technisch beschreven. Tijdens het proces van EML transformatie naar Web technologie zijn vaak externe gegevens nodig. Hiervoor is een standaard interface ontwikkeld. De Legacy koppelingen wordt als laatste hoofdstuk behandeld.

# MDA manager

## Functionaliteit

In deze versie van de MDAManager ondersteunt deze tool slechts een beperkt aantal functionaliteiten. Deze bestaan uit:

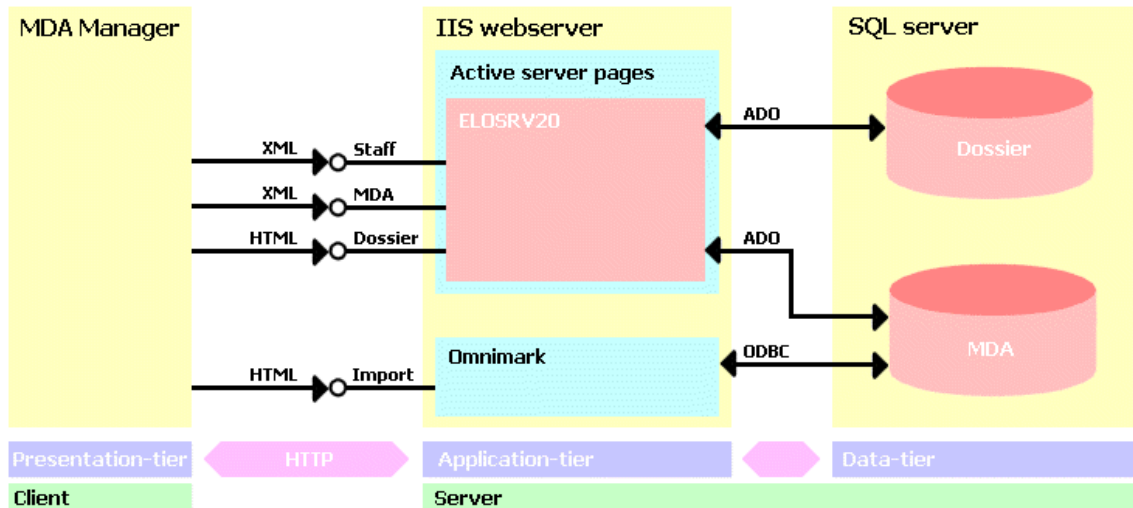
- Het geven van een overzicht van de in het systeem aanwezige Unit-of-studies, de geïnstalleerde / ondersteunde talen, media en vormgevingen.
- Het importeren van nieuw eml-gecodeerd materiaal in de MDA database.
- Het registreren van een Unit-of-study in de dossier database.
- Het toekennen van een in Edubox bekende persoon aan een geregistreerde Unit-of-study als runplanner.

## Applicatie architectuur

De functionaliteiten die de tool biedt worden in het typische geval niet door technische beheerfunctionarissen uitgevoerd, maar veeleer door personen met een zekere affiniteit met de inhoudelijke werking van Edubox. Denk hierbij aan een Edubox content manager. Deze functionarissen beschikken niet noodzakelijkwijs over een technische beheerwerkplek met rechtstreekse toegang tot de onderhavige databases. Om veiligheidsredenen is het zelfs af te raden om werkplekken met rechtsreeks toegang tot de databases beschikbaar te hebben voor personen die dit uit hoofde van hun functie niet nodig hebben.

De toegang tot de databases moet dus niet rechtstreeks vanuit de client applicatie plaatsvinden maar via een tussenlaag die op haar beurt wel rechtstreeks toegang moet hebben tot de databases. De koppeling tussen de client en de tussenlaag moet conform de Edubox uitgangspunten via het World Wide Web verlopen.

Deze eis heeft tot gevolg dat de applicatie architectuur een loskoppeling van de clientapplicatie en de databaseserver moet implementeren via een meerlagen architectuur waarbij de clientlaag en de applicatielaag gekoppeld zijn via web, meer precies, via het http protocol om de firewall en proxy problematiek zoveel mogelijk te beperken. In onderstaande figuur is de gekozen applicatie architectuur weergegeven.



Figuur 1 Applicatie architectuur

De applicatie valt ruwweg in vier onderdelen te onderscheiden. In de presentation-tier is er een Windows applicatie, MDAManager.exe, die met name het user-interface behelst. In de application-tier zijn er twee componenten die de data-tier aansturen. Dit zijn de Omnimark import module en de COM component ELOSRV20. Als laatste onderdeel is er nog de SQL server database met daarin de Dossier en de MDA database. De databases en de Omnimark import module zijn geen onderwerp van deze documentatie. De interfaces tussen de verschillende lagen zijn duidelijk omschreven en maken het mogelijk om enerzijds de componenten onafhankelijk van elkaar te ontwikkelen, maar maken het anderzijds ook mogelijk om de componenten onafhankelijk van elkaar te vervangen omdat er geen onderlinge afhankelijkheden bestaan anders dan de ontwikkelde interfaces.

## Interface

Conform de applicatie architectuur is de applicatie opgebouwd uit verschillende losse componenten die via een duidelijk interface met elkaar communiceren. Voor het interface tussen de presentation-tier en de application-tier geldt dat het over het http protocol moeten communiceren. Het interface beschrijft hier een RPC mechanisme (Remote Procedure Call) om vanaf de client code op de application-tier te kunnen aanroepen. Dit RPC mechanisme is geïmplementeerd volgens een eigen standaard.

Ten tijde van de ontwikkeling van de applicatie was de SOAP (Simple Object Access Protocol) standaard nog volop in ontwikkeling. Op dit moment is SOAP v1.1 vastgesteld en is het aan te raden om bij een her/door ontwikkeling van de applicatie aan te sluiten bij deze standaard.

Er zijn vier verschillende methoden die op de server aangeroepen worden. Twee methoden halen data op uit de databases op de server en de twee overige wijzigen data. De vier geïmplementeerde RPC's zijn:

### *Get-staff*

Via deze call haalt de clientapplicatie een overzicht van alle staff-medewerkers in edubox over.

De servercode wordt aangeroepen door een url op de server te benaderen, deze antwoord door een in de http response een xml bestand terug te sturen. De url is gedefiniëerd via de get-staff parameter in het configuratiebestand.

De xmlresponse heeft het volgende formaat:

```
<staff>
  <person id="staff1" name="Staff 1"/>
  <person id="staff2" name="Staff 2"/>
  <person id="staff3" name="Staff 3"/>
  <person id="staff4" name="Staff 4"/>
</staff>
```

### *get-mda*

Via deze call haalt de clientapplicatie een overzicht van alle unit-of-studies en roleplanners in edubox over.

De servercode wordt aangeroepen door een url op de server te benaderen, deze antwoord door een in de http response een xml bestand terug te sturen. De url is gedefiniëerd via de get-mda parameter in het configuratiebestand.

De xmlresponse heeft het volgende formaat:

```
<mda>
<styles>
  <presentationstyle id="2" name="Hoge Hotelschool Maastricht"/>
  <presentationstyle id="1" name="Open Universiteit Nederland"/>
</styles>
<media>
  <medium id="Neutral" name="Null medium needed for medium independent files"/>
  <medium id="Web" name="Publication for the world wide web"/>
</media>
<languages>
  <language id="[nl]" name="Dutch (Nederlands)"/>
  <language id="[en-uk]" name="English (United Kingdom)"/>
</languages>
<unitofstudies>
  <uos id="FDAC5D75-6E08-11D4-9D11-0004AC96F8CC" name="FAQ testen"/>
  <uos id="FFE2F784-9DC1-11D4-A22B-006094AE04C0" name="Organisatie en Personeel"/>
</unitofstudies>
<elementtypes>
  <elementtype id="Unit-of-study"/>
  <elementtype id="True-false-question"/>
  <elementtype id="Tool-object"/>
  <elementtype id="Test-group"/>
  <elementtype id="Subactivity"/>
  <elementtype id="Short-answer-question"/>
  <elementtype id="Sequence-question"/>
  <elementtype id="Section"/>
  <elementtype id="Role-information-object"/>
  <elementtype id="Questionnaire-object"/>
  <elementtype id="Question-answer"/>
  <elementtype id="Prerequisite"/>
  <elementtype id="Personal-object"/>
  <elementtype id="Multiple-response-question"/>
  <elementtype id="Multiple-choice-question"/>
  <elementtype id="Matching-question"/>
  <elementtype id="Learning-objective"/>
  <elementtype id="Knowledge-object"/>
  <elementtype id="Index-search-object"/>
  <elementtype id="Environment"/>
  <elementtype id="Communication-object"/>
  <elementtype id="Classify-question"/>
  <elementtype id="Classify-disjunct-question"/>
  <elementtype id="Bookmark"/>
  <elementtype id="Announcement-object"/>
  <elementtype id="Activity-structure"/>
```

```
<elementtype id="Activity"/>
</elementtypes>
<runplanners>
  <uos_dossier id="FDAC5D75-6E08-11D4-9D11-0004AC96F8CC" name="FAQ
testen" planner="ham"/>
  <uos_dossier id="05BB4CDA-D0CF-4CB8-8A86-E2B3FA45B845"
name="Organisatie en Personeel" planner="hvo"/>
</runplanners>
</mda>
```

### *post-dossier*

Via deze call kan de clientapplicatie een unit-of-study registreren in de dossier database of een runplanner toekennen/wijzigen aan een unit-of-study. De servercode wordt aangeroepen via een url op de server. De parameters van deze functie worden meegegeven op de url commandline. De action parameter geeft aan welke actie van de server verwacht wordt:

Action=add betekent dat de server de opgegeven unit-of-study in het dossier registreert en een runplanner toekent aan deze unit-of-study. Het formaat van deze commandline is:  
...asp?action=add&guid=<guid van de uos>&name=<uosnaam>&person=<runplanner-id>

Action=assign, de server wijzigt de runplanner van de opgegeven unit-of-study. Het formaat van deze commandline is:  
...asp?action=assign&guid=<de guid van de uos>&person=<runplanner-id>

### *import-element*

Deze call roept een Omnimark proces aan dat het opgegeven top level element importeert in de mda database. Ook deze procedure wordt aangeroepen via een url op de server. De parameters van deze call zijn:  
url?toplevel=<type toplevel element>&implurl=<een url naar het imlbestand dat het te importeren materiaal beschrijft>

## **MDAManager.exe**

Deze tool implementeert het userinterface van de applicatie. Dit tool is ontwikkeld in Delphi 5 en maakt gebruik van de standaard libraries WinInet en MSXML voor de communicatie met de application-tier. Alle functionaliteiten die de applicatie bevat worden via het userinterface van deze tool aan de eindgebruiker aangeboden.

Het tool is een standaard windows applicatie en de implementatie volgt de standaardmethodiek die gebruikelijk is voor een Delphi Windows applicatie. Deze methodiek wordt bekend veronderstelt. Verdere informatie over het ontwikkelen van een windows applicatie met Delphi staat in de Delphi documentatie of in één van de vele boeken die zijn verschenen over het ontwikkelen met deze programmeeromgeving. Een consequentie van deze manier van implementeren is dat de applicatie een event-driven architectuur volgt. Hierdoor is het niet mogelijk om de

flow van het programma in een flowchart te beschrijven, er ontbreekt tenslotte een vaste programma afloop van start tot einde.

Er is wel een metaflow te onderkennen in de applicatie.

1. Start de applicatie
2. Lees het configuratiebestand in
3. Haal de gegevens van de applicatie-tier op
4. Reageer op de door de gebruiker gemaakte keuzes in het interface. Roep eventueel een servercomponent aan om de acties van de gebruiker te persisteren in de databases.
5. Als de gebruiker in stap 4 daarvoor kiest, sluit de applicatie af.

## Het configuratiebestand

Dit bestand bevat de internetadressen van de vier servercomponenten van de application-tier waarmee het tool communiceert. Hierdoor is het mogelijk om met een en dezelfde tool meerdere Edubox-servers te beheren. De vier adressen vormen een combo van één server.

Een volgende versie van de applicatie zou moeten volstaan met één serveradres. Differentiatie naar andere componenten verloopt dan via dit adres. Ook hier kan de SOAP v1.1 een standaard oplossing bieden.

Het bestand wordt ingelezen met behulp van het `unCommon.Tparameters` object.

## Ophalen serverdata

Het tool haalt de gegevens op van de application-tier via communicatie over het http-protocol met het `ELOSRV20.Staff` en het `ELOSRV20.MDA` object. De communicatie wordt aangeroepen vanuit `TfmMDA.Refresh1Click`. De data wordt niet alleen opgehaald bij het starten van de applicatie, maar kan ook gebeuren op verzoek van de gebruiker door de refresh optie te kiezen in het hoofdmenu. Verder kan de applicatie deze methode zelf aanroepen om de gegevens te verversen na het verwerken van een opdracht van de gebruiker.

De volgende gegevens worden opgehaald van de server.

Een overzicht van alle staff medewerkers die bekend zijn binnen Edubox.

Aanroep: `load get-staff url via uncommon.GetInternetData`

Resultaat: de http response bevat de gevraagde lijst van staff medewerkers in een xml-formaat.

Dit resultaat wordt geparsed door de msxml parser. De resulterende DOM object boom wordt door de functie `FillTree` omgezet in een boomstructuur van `TXMLNodes` en opgeslagen in de `Fstaff` variabele.

Een overzicht van alle unit-of-studies in de mda.

Aanroep: `load get-mda url via unCommon.GetInternetData`

Resultaat: de http response bevat de lijst in een xml-formaat.

Ook deze lijst wordt door de msxml parser verwerkt en opgeslagen in globale variabelen. Zie hiervoor `Refresh1Click`.

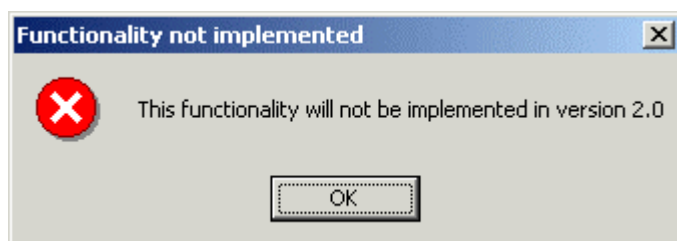
## Het Userinterface

### *Het applicatievenster*



Na het inlezen van de serverdata biedt de applicatie de gebruiker de mogelijkheid om deze data te bekijken. Dit overzicht wordt getoond in een tabblad in een tabcontrol. De treeview aan de linkerkant biedt toegang tot de verschillende tabbladen. Ook zijn er menuopties in het view menu die toegang verlenen. Ieder tabblad toont de relevante informatie in een listview.

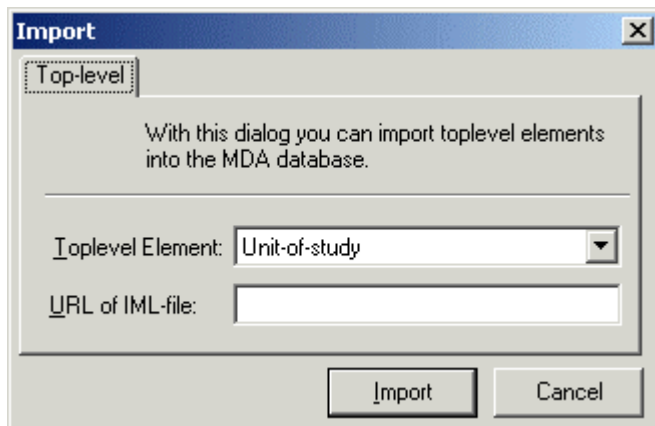
Niet alle buttons in het userinterface hebben hun functionaliteit geïmplementeerd.



### *Import eml materiaal*

Om een eml top-level-element te kunnen importeren is er een apart Tform gemaakt waar de gebruiker de parameters kan selecteren en invullen, waarna de applicatie de import-element functie aanroept op de server om het materiaal feitelijk in de mda

database te kunnen importeren. Deze aanroep gebeurt met de functie GetInternetData()

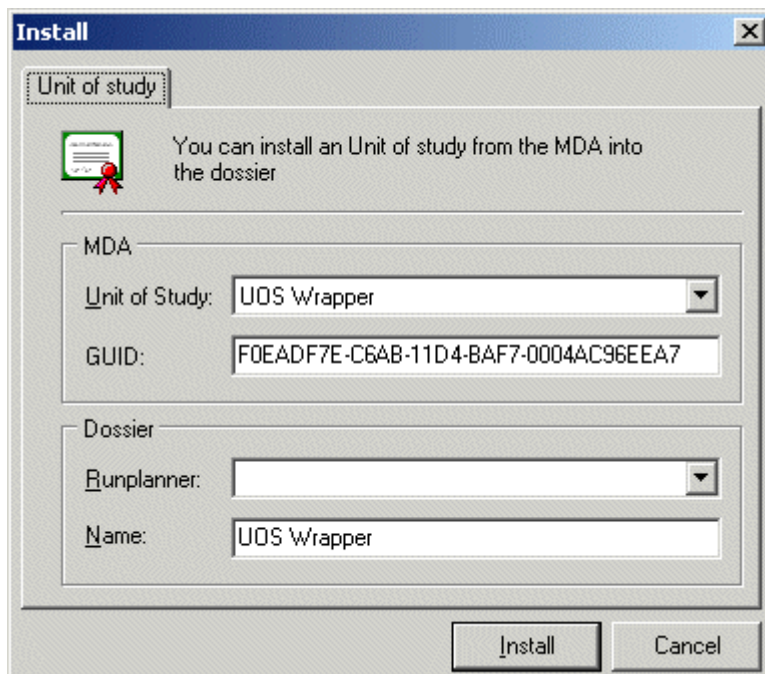


De lijst van geldige typen toplevel elementen wordt door de applicatie van de server gehaald.

De response van de server wordt getoond in een ie-browser object via de ShowHtrmIMessage.

### *Registratie UOS in dossier*

Ook voor het registreren van een is een dialoog gemaakt. Deze staat in unAddUos.pas en heet TfmAddUos.



Alleen unit-of-studies die nog niet in de dossier database staan geregistreerd kunnen worden geïnstalleerd. De applicatie bepaalt zelf de mogelijke uos door de MDA en de Dossier variabelen met elkaar te vergelijken.

Ook de runplanner combobox wordt gevuld door de applicatie. Hiervoor haalt de applicatie de data uit de meegeven Staff parameter.

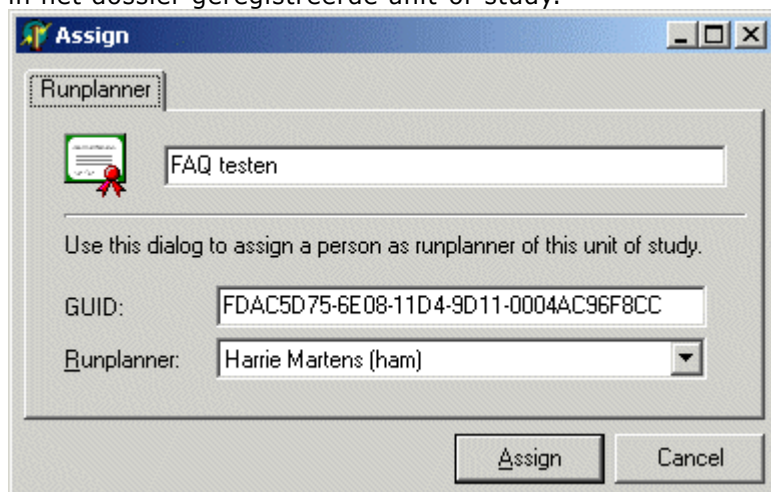


De registratie gebeurt door het aanroepen van de post-dossier-url met de action=assign parameter.

### Wijzigen runplanner

De functionaliteit wordt aangeboden via het TfmAssignRP form. Dit form staat in unAssignRP.pas.

Met deze dialoog kan de gebruiker een runplanner toekennen aan of wijzigen van een in het dossier geregistreerde unit-of-study.



Alle data in deze dialoog is read-only. De enige optie die een gebruiker heeft is het kiezen van een andere runplanner in de runplanner combobox.

De gemaakte keuze wordt met behulp van GetInternetData aan de post-dossier methode doorgegeven. In dit geval is de action parameter gedefiniëerd als action=assign.

### Bestanden

In onderstaande tabel zijn de verschillende bronbestanden opgesomd waaruit de applicatie bestaat.

<b>Naam</b>	<b>Locatie</b>	<b>Functie</b>
MDAManager.dpr	.	Main application source
MDAManager.cfg	.	Delphi configuratie bestand
MDAManager.dof	.	Delphi configuratie bestand
MDAManager.dsk	.	Delphi configuratie bestand
MDAManager.res	.	Application resource bestand
UnAddUos.dfm	.	Add Uos Form definitie
UnAddUos.pas	.	Add Uos Form unit source
UnAssignRP.dfm	.	Assign runplanner form definitie
UnAssignRP.pas	.	Assign runplanner form unit source
UnImport.dfm	.	Import eml form definitie
UnImport.pas	.	Import eml form unit source
UnMda.dfm	.	Main application form definitie
UnMda.pas	.	Main application form unit source

UnObjecten.pas	.	Unit met globale datastructuur definities
UnCommon.pas	..\common	Unit met general purpose routines
UnTools.pas	..\common	Unit met commandline parameters object
UnHtmlDlg.pas	..\common	Unit met de internet explorer dialoog.
Language.bmp	.\images	
Medium.bmp	.\images	
No-user.bmp	.\images	
Presentationstyle.bmp	.\images	
Uos.bmp	.\images	
User.bmp	.\images	
Users.bmp	.\images	

---

## ELOSRV20.DLL

Deze dll bevat de server componenten van de edubox content management applicatie. Deze servercomponenten zijn geïmplementeerd als delphi ASP server objecten, opgenomen in een met delphi gemaakte dll. Delphi ASP Server objecten zijn com objecten die een aantal extra com interfaces ondersteunen die een betere integratie met de asp webserver mogelijk maken. Specifiek stellen deze interfaces de asp server objecten ter beschikking aan de com objecten.

Een comobject wordt gedefiniëerd door zijn interfaces. In Delphi wordt voor ieder geïmplementeerd interface een delphi TASPObject aangemaakt, dat op zijn beurt de implementatie van het gepubliceerde interface bevat.

Waar de objecten toegang hebben tot een database gebeurt dit via ADO.

### Interfaces

De dll bevat de volgende vier interfaces; IDossier, IMDA, IStaff en IPortal.

#### *IDossier*

<b>naam</b>	<b>type</b>	<b>functie</b>
AddUos	Method	Registreert de opgegeven uos in het dossier, kent de runplanner toe
AssignRunplanne r	Method	Wijzigt de runplanner van de opgegeven uos
ConnectionString Provider	Property Property	Bevat de database connection string Bevat de ADO database provider

#### *IMDA*

<b>naam</b>	<b>type</b>	<b>functie</b>
GetData	Method	Haalt de gevraagde informatie uit de mda en de dossier database, maakt deze op in een xml formaat (zie interface beschrijving) en geeft deze via het asp response object terug aan de client applicatie.
MDAConnectionStrin g	Property	Bevat de MDA database connection string
ConnectionString Provider	Property Property	Bevat de dossier database connection string Bevat de ADO database provider

## *IStaff*

<b>naam</b>	<b>type</b>	<b>functie</b>
GetAll	Method	Haalt alle staff medewerkers via ado uit de dossier database, maakt deze op in een xml formaat (zie interface beschrijving) en geeft deze via het asp response object terug aan de client applicatie.
ConnectionString	Property	Bevat de dossier database connection string
Provider	Property	Bevat de ADO database provider

## *IPortal*

<b>naam</b>	<b>type</b>	<b>functie</b>
GetUosList	Method	Bepaalt tot welke unit-of-studies de opgegeven user toegang heeft door via ado de dossierdatabase te raadplegen. De gevraagde informatie wordt in een xml formaat via het asp response object teruggeven.
ConnectionString	Property	Bevat de dossier database connection string
Provider	Property	Bevat de ADO database provider

## Bestanden

In onderstaande tabel zijn de verschillende bronbestanden opgesomd waaruit elosrv20 bestaat.

<b>Naam</b>	<b>Locatie</b>	<b>Functie</b>
ELOSRV20.cfg	.	Delphi configuratie bestand
ELOSRV20.dof	.	Delphi configuratie bestand
ELOSRV20.dpr	.	Bevat de code van de dll
ELOSRV20.dsk	.	Delphi configuratie bestand
ELOSRV20_tlb.pas	.	Bevat de type library interface in pascal
ELOSRV20.tlb	.	Bevat de type library van de dll
UnDossier.pas		Bevat de implementatie van Idossier
UnMDA.pas	.	Bevat de implementatie van IMDA
UnPortal.aps	.	Bevat de implementatie van Iportal
UnStaff.pas	.	Bevat de implementatie van IStaff

## Roleplanner

### Functionaliteit

De roleplanner tool ondersteunt de volgende functionaliteiten.

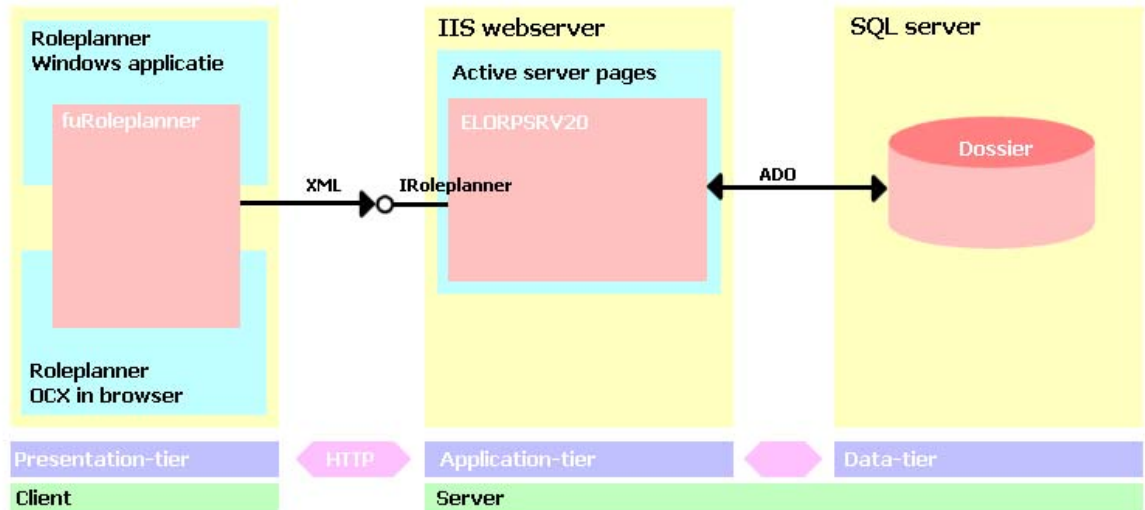
- Het geven van een overzicht van de gebruikerspopulatie van een run.
- Het toekennen van gebruikers aan een subrol.
- Het verwijderen van gebruikers uit een subrol.
- Het toewijzen van een publicatie aan een subrol.
- Het verwijderen van een publicatie bij een subrol.

### Applicatie architectuur

De functionaliteiten die de tool biedt worden in het typische geval niet door technische beheerfunctionarissen uitgevoerd, maar veeleer door personen met een zekere affiniteit met de inhoudelijke werking van Edubox. Denk hierbij aan een onderwijs manager. Deze functionarissen beschikken niet noodzakelijkwijs over een technische beheerwerkplek met rechtstreekse toegang tot de onderhavige databases. Om veiligheidsredenen is het zelfs af te raden om werkplekken met rechtsreeks toegang tot de databases beschikbaar te hebben voor personen die dit uit hoofde van hun functie niet nodig hebben.

De toegang tot de databases moet dus niet rechtstreeks vanuit de client applicatie plaatsvinden maar via een tussenlaag die op haar beurt wel rechtstreeks toegang moet hebben tot de databases. De koppeling tussen de client en de tussenlaag moet conform de Edubox uitgangspunten via het World Wide Web verlopen.

Deze eis heeft tot gevolg dat de applicatie architectuur een loskoppeling van de clientapplicatie en de databaseserver moet implementeren via een meerlagen architectuur waarbij de clientlaag en de applicatielaag gekoppeld zijn via web, meer precies, via het http protocol om de firewall en proxy problematiek zoveel mogelijk te beperken. In onderstaande figuur is de gekozen applicatie architectuur weergegeven.



Figuur 2 Applicatie architectuur

De applicatie valt ruwweg in vier onderdelen te onderscheiden. In de presentation-tier zijn er twee varianten van de roleplanner. Enerzijds is er een ActiveX variant die in een webbrowser opgestart wordt. Anderzijds is er een standalone windows applicatie. Deze beide varianten vormen het grafische front-end van de applicatie. In de application-tier is er één component die de data-tier aanstuurt. Dit is de COM component ELORPSRV20. Als laatste onderdeel is er nog de SQL server database met daarin de Dossier en de MDA database. De databases zijn geen onderwerp van deze documentatie. De interfaces tussen de verschillende lagen zijn duidelijk omschreven en maken het mogelijk om enerzijds de componenten onafhankelijk van elkaar te ontwikkelen, maar maken het anderzijds ook mogelijk om de componenten onafhankelijk van elkaar te vervangen omdat er geen onderlinge afhankelijkheden bestaan anders dan de ontwikkelde interfaces.

## Interface

Conform de applicatie architectuur is de applicatie opgebouwd uit verschillende losse componenten die via een duidelijk interface met elkaar communiceren. Voor het interface tussen de presentation-tier en de application-tier geldt dat het over het http protocol moeten communiceren. Het interface beschrijft hier een RPC mechanisme (Remote Procedure Call) om vanaf de client code op de application-tier te kunnen aanroepen. Dit RPC mechanisme is geïmplementeerd volgens een eigen standaard.

Ten tijde van de ontwikkeling van de applicatie was de SOAP (Simple Object Access Protocol) standaard nog volop in ontwikkeling. Op dit moment is SOAP v1.1 vastgesteld en is het aan te raden om bij een her/door ontwikkeling van de applicatie aan te sluiten bij deze standaard.

De roleplanner kent één com interface, namelijk IRolePlanner. Dit interface wordt op de server aangeroepen vanuit één centrale asp pagina. Deze pagina vormt dus de verbinding tussen enerzijds het http-protocol en anderzijds het com interface. De client applicaties op hun beurt roepen niet rechtstreeks het com object aan maar de asp pagina.

De com interfaces worden dus via een custom protocol aangeroepen. Dit protocol kent de volgende functies:

## Getrunlist

### Syntax

```
...asp?action=getrunlist
```

Deze functie geeft alle runs terug waarvan de gebruiker de roleplanner is. Het resultaat wordt via de http response teruggegeven. Het formaat van de response is een naar xml persisted ado recordset:

```
<xml xmlns:s='uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882'
  xmlns:dt='uuid:C2F41010-65B3-11d1-A29F-00AA00C14882'
  xmlns:rs='urn:schemas-microsoft-com:rowset'
  xmlns:z='#RowsetSchema'><s:Schema id='RowsetSchema'>
  <s:ElementType name='row' content='eltOnly' rs:CommandTimeout='30'>
    <s:AttributeType name='id' rs:number='1'>
      <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
    </s:AttributeType>
    <s:AttributeType name='name' rs:number='2' rs:nullable='true'
rs:writeunknown='true'>
      <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' />
    </s:AttributeType>
    <s:AttributeType name='runid' rs:number='3'>
      <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
    </s:AttributeType>
    <s:extends type='rs:rowbase' />
  </s:ElementType>
</s:Schema>
<rs:data>
  <z:row id='621' name='run ' runid='20' />
</rs:data>
</xml>
```

## Getrun

### Syntax:

```
...asp?action=getrun&rundeelnameid=<roleplanner's rundeelname id>
```

Deze functie haalt alle relevante informatie betreffende de in rundeelnameid gespecificeerde run op uit de dossier database. Deze informatie wordt in een xml formaat teruggegeven aan de clientapplicatie via de http response. Het formaat van de xml response is een door ado naar xml geconverteerde recordset:

```
<xml xmlns:s='uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882'
  xmlns:dt='uuid:C2F41010-65B3-11d1-A29F-00AA00C14882'
  xmlns:rs='urn:schemas-microsoft-com:rowset'
  xmlns:z='#RowsetSchema'>
<s:Schema id='RowsetSchema'>
  <s:ElementType name='row' content='eltOnly' rs:CommandTimeout='30'
rs:ReshapeName='run'>
    <s:AttributeType name='Id' rs:number='1'>
      <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
    </s:AttributeType>
    <s:AttributeType name='PersonId' rs:number='2'
rs:writeunknown='true'>
      <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='50' rs:maybenull='false' />
    </s:AttributeType>
    <s:AttributeType name='ImpersonationId' rs:number='3'
rs:nullable='true' rs:writeunknown='true'>
      <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='50' />
  </s:ElementType>
</s:Schema>
```

```

        </s:AttributeType>
        <s:AttributeType name='RunId' rs:number='4'
rs:writeunknown='true'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='RoleId' rs:number='5'
rs:writeunknown='true'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='Status' rs:number='6'
rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='2' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='Startdate' rs:number='7'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='dateTime' rs:dbtype='timestamp'
dt:maxLength='16' rs:scale='3' rs:precision='23' rs:fixedlength='true' />
        </s:AttributeType>
        <s:AttributeType name='EndDate' rs:number='8'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='dateTime' rs:dbtype='timestamp'
dt:maxLength='16' rs:scale='3' rs:precision='23' rs:fixedlength='true' />
        </s:AttributeType>
        <s:AttributeType name='LastPublicationRecepyId' rs:number='9'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' />
        </s:AttributeType>
        <s:AttributeType name='RunDefault' rs:number='10'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='1' />
        </s:AttributeType>
        <s:ElementType name='publication' content='eltOnly'
rs:CommandTimeout='30' rs:ReshapeName='publication' rs:relation='
040000000700000000000000'>
            <s:AttributeType name='id' rs:number='1'>
                <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
            </s:AttributeType>
            <s:AttributeType name='name' rs:number='2'
rs:nullable='true' rs:writeunknown='true'>
                <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' />
            </s:AttributeType>
            <s:AttributeType name='c2' rs:name='name' rs:number='3'
rs:writeunknown='true'>
                <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' rs:maybenull='false' />
            </s:AttributeType>
            <s:AttributeType name='c3' rs:name='id' rs:number='4'
rs:writeunknown='true'>
                <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='50' rs:maybenull='false' />
            </s:AttributeType>
            <s:AttributeType name='language' rs:number='5'
rs:writeunknown='true'>
                <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' rs:maybenull='false' />
            </s:AttributeType>
            <s:AttributeType name='presentation' rs:number='6'
rs:writeunknown='true'>
                <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' rs:maybenull='false' />
            </s:AttributeType>
            <s:AttributeType name='runid' rs:number='7'
rs:writeunknown='true'>
                <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
            </s:AttributeType>

```



```

        <s:extends type='rs:rowbase' />
    </s:ElementType>
    <s:ElementType name='role' content='eltOnly'
rs:CommandTimeout='30' rs:ReshapeName='role' rs:relation='0400
00000d0000000000000000'>
        <s:AttributeType name='roleid' rs:number='1'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='Id' rs:number='2'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='WuId' rs:number='3'
rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='50' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='UosReleaseId' rs:number='4'
rs:writeunknown='true'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:AttributeType name='Name' rs:number='5'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' />
        </s:AttributeType>
        <s:AttributeType name='Class' rs:number='6'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='1' />
        </s:AttributeType>
        <s:AttributeType name='Type' rs:number='7'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='50' />
        </s:AttributeType>
        <s:AttributeType name='MatchPersons' rs:number='8'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='1' />
        </s:AttributeType>
        <s:AttributeType name='MaxPersons' rs:number='9'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='ui1' dt:maxLength='1'
rs:precision='3' rs:fixedlength='true' />
        </s:AttributeType>
        <s:AttributeType name='MinPersons' rs:number='10'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='ui1' dt:maxLength='1'
rs:precision='3' rs:fixedlength='true' />
        </s:AttributeType>
        <s:AttributeType name='ParentRoleId' rs:number='11'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' />
        </s:AttributeType>
        <s:AttributeType name='InformationUrl' rs:number='12'
rs:nullable='true' rs:writeunknown='true'>
            <s:datatype dt:type='string' rs:dbtype='str'
dt:maxLength='250' />
        </s:AttributeType>
        <s:AttributeType name='runid' rs:number='13'>
            <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:ElementType name='person' content='eltOnly'
rs:CommandTimeout='30' rs:ReshapeName='person' rs:relation='
010000000200000000000000'>
            <s:AttributeType name='personid' rs:number='1'
rs:writeunknown='true'>
                <s:datatype dt:type='string'

```

```

rs:dbtype='str' dt:maxLength='50' rs:maybenull='false' />
  </s:AttributeType>
  <s:AttributeType name='roleid' rs:number='2'
rs:writeunknown='true'>
    <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
    </s:AttributeType>
    <s:extends type='rs:rowbase' />
  </s:ElementType>
  <s:ElementType name='recepy' content='eltOnly'
rs:CommandTimeout='30' rs:ReshapeName='recepy' rs:relation='
010000000200000000000000'>
    <s:AttributeType name='publicationrecepyid'
rs:number='1' rs:writeunknown='true'>
      <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
      </s:AttributeType>
      <s:AttributeType name='roleid' rs:number='2'
rs:writeunknown='true'>
        <s:datatype dt:type='int' dt:maxLength='4'
rs:precision='10' rs:fixedlength='true' rs:maybenull='false' />
        </s:AttributeType>
        <s:extends type='rs:rowbase' />
      </s:ElementType>
      <s:extends type='rs:rowbase' />
    </s:ElementType>
  </s:Schema>
<rs:data>
  <rs:data>
    <z:row Id='621' PersonId='ham' RunId='20' RoleId='144' Status='01'
Startdate='2001-02-08T15:19:24.660000000'>
      <publication id='20' name='2e pub' c2='Publication for world
wide web' c3='[nl]' language='Dutch (Nederlands)'
presentation='Open Universiteit Nederland'
runid='20' />
      <role roleid='140' Id='140' WuId='studenten'
UosReleaseId='1038' Class='L' InformationUrl='role_studenten.htm'
runid='20'>
        <recepy publicationrecepyid='20' roleid='140' />
      </role>
      <role roleid='141' Id='141' WuId='docenten'
UosReleaseId='1038' Class='S' InformationUrl='role_docenten.htm'
runid='20'>
        <person personid='fbr' roleid='141' />
        <person personid='ham' roleid='141' />
        <recepy publicationrecepyid='20' roleid='141' />
      </role>
      <role roleid='142' Id='142' WuId='docent' UosReleaseId='1038'
Class='S' ParentRoleId='141' InformationUrl='role_docent.htm'
runid='20'>
        <recepy publicationrecepyid='20' roleid='142' />
      </role>
      <role roleid='143' Id='143' WuId='begeleider'
UosReleaseId='1038' Class='S' ParentRoleId='141'
InformationUrl='role_begeleider.htm'
runid='20'>
        <recepy publicationrecepyid='20' roleid='143' />
      </role>
      <role roleid='144' Id='144' WuId='Role Planner'
UosReleaseId='1038' Class='S' ParentRoleId='141' runid='20'>
        <person personid='ham' roleid='144' />
        <recepy publicationrecepyid='20' roleid='144' />
      </role>
    </z:row>
  </rs:data>
</xml>

```

## Store

### Syntax:

```
...asp?action=store&rundeelnameid=<roleplanner's rundeelname id>
```

Met deze functie worden de wijzingen die door de roleplanner zijn aangebracht doorgegeven aan de application-tier, om via de servercomponent opgeslagen te worden in de database.

Er zijn vier verschillende mutaties die op de database uitgevoerd kunnen worden, namelijk het toevoegen van een persoon aan een rol, het verwijderen van een persoon, het toekennen van een publicatie aan een rol, of het verwijderen van een publicatie bij een rol. Alle mutaties worden afzonderlijk gecodeerd in een overkoepelend xml bestand. Het formaat van dit bestand is:

```
<?xml version="1.0"?>
<transactions>
<transaction subject="user" action="add" id="" roleid="" />
<transaction subject="user" action="delete" id="" roleid="" />
<transaction subject="publication" action="add" id="" roleid="" />
<transaction subject="publication" action="delete" id="" roleid="" />
</transactions>
```

De servercomponent schrijft via het ASP Response object het resultaat van de processing terug naar de client. Als de processing zonder fouten verloopt, dan geeft de component 0 – Operation completed successfully terug. Bij fouten geeft de component de code 500 – Errors occurred<CR><LF> gevolgd door een lijst met individuele foutmeldingen.

## Client tools

Er zijn twee varianten van de clienttools ontwikkeld. Er is een ocx variant gemaakt die oproepbaar is vanuit een webpagina in de webbrowser, en de andere variant is een standalone windows applicatie.

Beide varianten vormen een schil rondom een delphi frame object. Dit frame object, TfrmRoleplanner in fuRoleplanner, bevat de feitelijke user-interface en applicatie logica. De beide varianten voegen alleen een wrapper toe die dit frame toont in de relevante applicatie context.

Het enige functionele verschil tussen beide implementaties is dat de standalone applicatie de mogelijk biedt aan de gebruiker om zelf te kiezen welke run deze wil beheren. In de ocx variant wordt deze keuze op voorhand gemaakt en kan de gebruiker alleen de reeds ingevulde run beheren.

In technische zin zijn de verschillen dat de ocx variant een Delphi windows dll is met daarin opgenomen een ActiveForm (die op zijn beurt weer het delphi frame object bevat), terwijl de standalone applicatie een standaard delphi windows applicatie is. Beide producten zijn ontwikkeld met Delphi 5 enterprise.

De rest van deze documentatie gaat over het gemeenschappelijke deel van de twee varianten.

De standaardcomponent TfrmRoleplanner bevat het userinterface van de applicatie. Dit frame maakt gebruik van de standaard libraries van delphi, WinInet en MSXML voor de implementatie. Alle functionaliteiten die de applicatie bevat worden via het userinterface van deze tool aan de eindgebruiker aangeboden.

Het tool volgt de standaard windows gui en de implementatie volgt de standaardmethodiek die gebruikelijk is voor een Delphi Windows applicatie. Deze methodiek wordt bekend veronderstelt. Verdere informatie over het ontwikkelen van een windows applicatie met Delphi staat in de Delphi documentatie of in één van de vele boeken die zijn verschenen over het ontwikkelen met deze programmeeromgeving. Een consequentie van deze manier van implementeren is dat de applicatie een event-driven architectuur volgt. Hierdoor is het niet mogelijk om de flow van het programma in een flowchart te beschrijven, er ontbreekt tenslotte een vaste programma afloop van start tot einde.

Er is wel een metaflow te onderkennen in de applicatie.

6. Start de applicatie
7. Lees het configuratiebestand in
8. Haal de gegevens van de applicatie-tier op
9. Reageer op de door de gebruiker gemaakte keuzes in het interface. Roep de servercomponent aan om de acties van de gebruiker te persisteren in de databases na het aanklikken van de submit button.
10. Als de gebruiker in stap 4 daarvoor kiest, sluit de applicatie af.

## Het configuratiebestand

Dit bestand bevat het internetadres van de servercomponent van de application-tier waarmee het tool communiceert. Hierdoor is het mogelijk om met een en dezelfde tool meerdere Edubox-servers te beheren.

## Ophalen serverdata

Het tool haalt de gegevens op van de application-tier via communicatie over het http-protocol met het ELORPSRV20.Roleplanner object. De communicatie wordt aangeroepen vanuit TfrmRoleplanner.btResetClick. De data wordt niet alleen opgehaald bij het starten van de applicatie, maar kan ook gebeuren op verzoek van de gebruiker door de reset button aan te klikken. Verder kan de applicatie deze methode zelf aanroepen om de gegevens te verversen na het verwerken van een opdracht van de gebruiker.

De opgehaalde data (via het getrun-protocol) worden geparsed met behulp van de msxml parser en vervolgens lokaal verwerkt en opgeslagen in het FRoles object. Dit object is een tree structuur die logische eml rollen structuur volgt. Per rol worden in iedere TRoleNode de diverse kenmerken van de betreffende rol bijgehouden. De volgende informatie wordt per rol opgeslagen:

```
property Id: integer
property Name: string;
property ExclusiveMatch: boolean;
property MaxPersons: integer;
property MinPersons: integer;
property IsStaff: boolean;
property Users: TList;
property Pubs: TList;
```

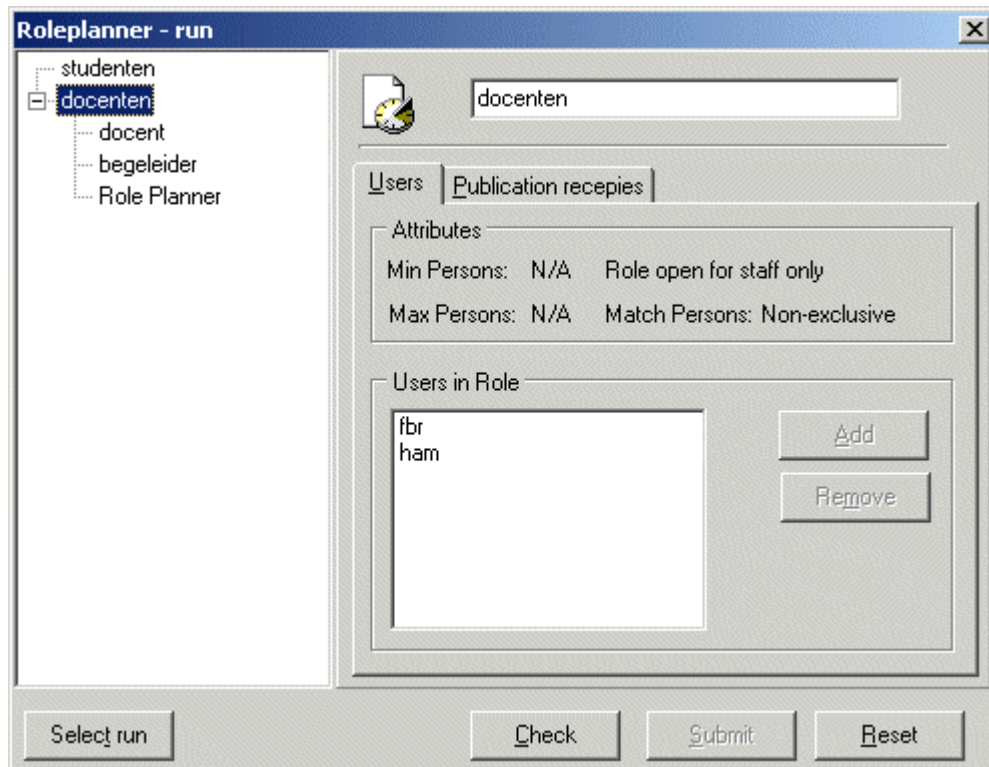
Bij het initialiseren van de node via de create constructor worden kopieën gemaakt van de Users en de Pubs properties. Deze worden opgeslagen in de volgende properties:

```
property OriginalUsers: TList;
property OriginalPubs: TList;
```

Aan de hand van deze originele kopie bepaalt de applicatie welke wijzingen door de eindgebruiker zijn aangebracht aan de rol instellingen. Conform het server protocol worden namelijk alleen de wijzigingen doorgegeven aan de application-tier.

## Het userinterface

### Het applicatievenster



Zoals uit bovenstaande afbeelding blijkt staan de rollen centraal in het userinterface van de roleplanner. Alle rollen van de run staan afgebeeld in de Ttreeview tvRoles aan de linkerkant. Door een rol aan te klikken wordt de informatie van de betreffende rol getoond in de TpageControl PageControl1. De informatie wordt getoond in twee tabsheets. TabSheet1 toont de userinformatie van de rol, terwijl TabSheet2 de toegewezen publicaties laat zien.

De rollen op het hoofd-nivo, Staff en Learner worden niet door de roleplanner beheerd, maar door de runplanner. Vandaar dat de Add en Remove buttons in het interface zijn gedisabled.

Zowel bij het toevoegen van users in een role, als bij het toekennen van een publicatie aan een role wordt de set van geldige keuzes bepaald door de populatie van de parentrole. Het is dus niet mogelijk om personen aan een sub-role toe te kennen, als deze personen niet reeds aan de role waren toegewezen. En als een user uit een parentrole wordt verwijderd, dan wordt hij tevens ook uit alle subrollen verwijderd.

Alle wijzigingen die de gebruiker met het tool aanbrengt zijn lokaal. Pas nadat de gebruiker op de submit button drukt valideert de applicatie of er geen eml constraints

geschonden zijn en stuurt de wijzigingen op naar de application-tier via het 'store-protocol'.

## Bestanden

In onderstaande tabel zijn de verschillende bronbestanden opgesomd waaruit de applicatie bestaat.

<b>Naam</b>	<b>Locatie</b>	<b>Functie</b>
afRoleplanner.dfm	.	ActiveForm definitie
afRoleplanner.pas	.	ActiveForm interface definitie
ELOROLE20.cfg	.	Delphi configuratie bestand
ELOROLE20.dpr	.	OCX applicatie sourcecode
ELOROLE20.dof	.	Delphi configuratie bestand
ELOROLE20.dsk	.	Delphi configuratie bestand
ELOROLE20.res	.	Application resource bestand
ELOROLE20.tlb	.	Typelibrary van de OCX
ELOROLE20_TLB.pas	.	Pascal implementatie van de typelibrary
fuRoleplanner.dfm	.	Frame definitie
fuRoleplanner.pas	.	Frame source
RolePlanner.cfg	.	Exe applicatie configuratie bestand
RolePlanner.dof	.	Exe applicatie configuratie bestand
RolePlanner.dpr	.	Exe applicatie source
RolePlanner.dsk	.	Exe applicatie configuratie bestand
RolePlanner.res	.	Exe applicatie resource bestand
unAdd.dfm	.	Add users form definitie
unAdd.pas	.	Add users sourcecode
unAddPub.dfm	.	Add publications form definitie
unAddPub.pas	.	Add publications sourcecode
unConstraints.dfm	.	Constraints form definitie
unConstraints.pas	.	Constraints form sourcecode
unMain.dfm	.	Exe applicatie main window definitie
unMain.pas	.	Exe applicatie main window sourcecode
unObjecten.pas	.	Globale datadefinities sourcecode
unPropsPub.dfm	.	Publication properties form definitie
unPropsPub.pas	.	Publication properties form sourcecode
unRunSelect.dfm	.	Exe app select run form definitie
unRunSelect.pas	.	Exe app select run form sourcecode
unSubmitErrors.dfm	.	Submit errors form definitie
unSubmitErrors.pas	.	Submit errors form sourcecode

## ELORPSRV20.DLL

Deze dll bevat de server component van de edubox roleplanner applicatie. Deze servercomponent is geïmplementeerd als een delphi ASP server object, opgenomen in een met delphi gemaakte dll. Delphi ASP Server objecten zijn com objecten die een aantal extra com interfaces ondersteunen die een betere integratie met de asp webserver mogelijk maken. Specifiek stellen deze interfaces de asp server objecten ter beschikking aan de com objecten.

Een comobject wordt gedefiniëerd door zijn interfaces. In Delphi wordt voor ieder geïmplementeerd interface een delphi TASPObject aangemaakt, dat op zijn beurt de implementatie van het gepubliceerde interface bevat.

Waar de objecten toegang hebben tot een database gebeurt dit via ADO.

### Interfaces

De dll bevat het volgende interfaceIRoleplanner.

#### *IRoleplanner*

<b>Naam</b>	<b>type</b>	<b>Functie</b>
GetRunData	Method	Haalt de gegevens van de aangegeven run op
ProcessTransactions	Method	Verwerk de database mutaties
ConnectionString	Property	Bevat de database connection string
GetRunList	Method	Haalt lijst van runs op waarvan UserID de roleplanner is

### Bestanden

In onderstaande tabel zijn de verschillende bronbestanden opgesomd waaruit elorpsrv20 bestaat.

<b>Naam</b>	<b>Locatie</b>	<b>Functie</b>
ELORPSRV20.cfg	.	Delphi configuratie bestand
ELORPSRV20.dof	.	Delphi configuratie bestand
ELORPSRV20.dpr	.	Bevat de code van de dll
ELORPSRV20.dsk	.	Delphi configuratie bestand
ELORPSRV20_tlb.pas	.	Bevat de type library interface in pascal
ELORPSRV20.tlb	.	Bevat de type library van de dll
UnRoleplanner.pas		Bevat de implementatie van IRoleplanner

# Runplanner

## Functionaliteit

De runplanner tool ondersteunt de volgende functionaliteiten.

- Het geven van een overzicht van alle unit-of-studies waar de gebruiker van de applicatie de runplanner is.
- Het beheren van de releases van de unit-of-studies (aanmaken, wijzigen en verwijderen)
- Het beheren van publicaties van een release
- Het beheren van de runs van een release
- Het inschrijven van gebruikers in een run (in de hoofdrollen staff en learner)
- Het toekennen van een publicatie aan een (rol in een) run.
- Het overzetten van de gebruikerspopulatie naar een nieuwe run.

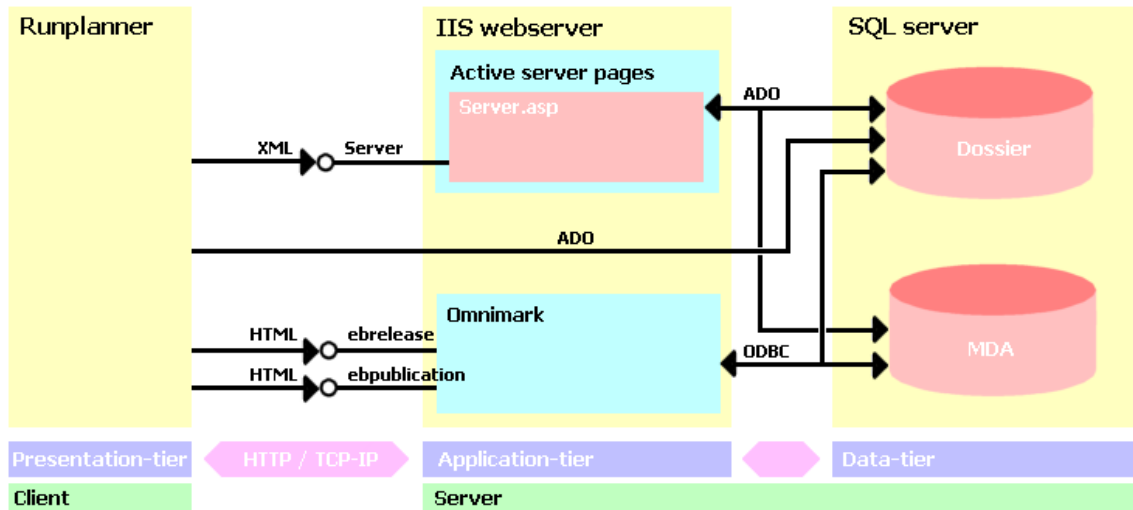
## Applicatie architectuur

De functionaliteiten die de tool biedt worden in het typische geval niet door technische beheerfunctionarissen uitgevoerd, maar veeleer door personen met een zekere affiniteit met de inhoudelijke werking van Edubox. Denk hierbij aan een onderwijs manager. Deze functionarissen beschikken niet noodzakelijkwijs over een technische beheerwerkplek met rechtstreekse toegang tot de onderhavige databases. Om veiligheidsredenen is het zelfs af te raden om werkplekken met rechtsreekse toegang tot de databases beschikbaar te hebben voor personen die dit uit hoofde van hun functie niet nodig hebben.

De toegang tot de databases moet dus niet rechtstreeks vanuit de client applicatie plaatsvinden maar via een tussenlaag die op haar beurt wel rechtstreeks toegang moet hebben tot de databases. De koppeling tussen de client en de tussenlaag moet conform de Edubox uitgangspunten via het World Wide Web verlopen. In deze versie van de tool is deze 'web-koppeling' maar deels geïmplementeerd.

Deze eis heeft tot gevolg dat de applicatie architectuur een loskoppeling van de clientapplicatie en de databaseserver moet implementeren via een meerlagen architectuur waarbij de clientlaag en de applicatielaag gekoppeld zijn via web, meer precies, via het http protocol om de firewall en proxy problematiek zoveel mogelijk te beperken. In onderstaande figuur is de gekozen applicatie architectuur weergegeven.





Figuur 3 Applicatie architectuur

De applicatie valt ruwweg in vier onderdelen te onderscheiden. In de presentation-tier is er de als windows applicatie geïmplementeerd user-interface van de applicatie. In de application-tier zijn er verschillende componenten die de applicatie logica implementeren. Twee componenten, ebrelease en ebpublication zijn omnimark processen bedoeld om een nieuwe unit-of-study release en om een nieuwe publicatie van een release te maken. Deze componenten worden op een andere plek gedocumenteerd. Verder is er in deze laag nog een server component, geïmplementeerd als een asp-pagina die het ophalen van de data uit de data-tier regelt. Als laatste onderdeel is er nog de SQL server database met daarin de Dossier en de MDA database. De databases zijn geen onderwerp van deze documentatie. De interfaces tussen de verschillende lagen zijn duidelijk omschreven en maken het mogelijk om enerzijds de componenten onafhankelijk van elkaar te ontwikkelen, maar maken het anderzijds ook mogelijk om de componenten onafhankelijk van elkaar te vervangen omdat er geen onderlinge afhankelijkheden bestaan anders dan de ontwikkelde interfaces. Zoals uit bovenstaande figuur blijkt is er, in tegenstelling tot de systeemeisen, ook nog een rechtstreekse koppeling tussen de client applicatie en de database-server. In een volgende versie moet de logica uit de client naar de application-tier getransporteerd worden.

## Interface

Conform de applicatie architectuur is de applicatie opgebouwd uit verschillende losse componenten die via een duidelijk interface met elkaar communiceren. Voor het interface tussen de presentation-tier en de application-tier geldt dat het over het http protocol moeten communiceren. Het interface beschrijft hier een RPC mechanisme (Remote Procedure Call) om vanaf de client code op de application-tier te kunnen aanroepen. Dit RPC mechanisme is geïmplementeerd volgens een eigen standaard.

Ten tijde van de ontwikkeling van de applicatie was de SOAP (Simple Object Access Protocol) standaard nog volop in ontwikkeling. Op dit moment is SOAP v1.1 vastgesteld en is het aan te raden om bij een her/door ontwikkeling van de applicatie aan te sluiten bij deze standaard.

De runplanner roept drie verschillende componenten aan in de applicatie-tier. In twee gevallen is dit de aanroep van een omnimark server en in het derde is dit de aanroep van een asp-pagina.

Ieder interface bestaat uit één of meerdere functieaanroepen die vanaf de client worden uitgevoerd.

## Server.asp

Dit interface beschrijft de functies waarmee de client applicatie de gegevens uit de database-tier kan ophalen. Het interface bestaat uit de volgende functies:

### Staff

#### Syntax

```
...asp?action=staff
```

De functie geeft een lijst van alle staff medewerkers in Edubox. Het resultaat wordt via de http respons teruggegeven. Het formaat van de respons is:

```
<staff>
<person id="ou_runplanner" name="OU Runplanner"/>
<person id="ou_staf1" name="ou_staf 1"/>
<person id="ou_staf2" name="ou_staf 2"/>
<person id="ou_staf3" name="ou_staf 3"/>
</staff>
```

### Root

#### Syntax

```
...asp?action=root
```

Deze functie geeft alle unit-of-studies met hun informatie terug waavan de gebruiker de runplanner is. Verder worden alle geïnstalleerde talen, media en huisstijlen getoond. Het overzicht wordt in de http respons teruggegeven aan de client. Het formaat van de xml response is

```
<root>
  <language id="[nl]" name="Dutch (Nederlands)"/>
  <style id="1" name="Open Universiteit Nederland"/>
  <medium id="Web" name="Publication for the world wide web"/>
  <uos id="553E08E1-357D-11D4-A242-000629E2559F" name="Beleidskunde 31-8-2000" defaultrun="8">
    <pct id="23" name="Beleidskunde release 31-8-2000" status="1" version="1.0.82" creationdate="36767.66368055555">
      <recepy id="7" name="Beleidskunde publicatie 31-8-2000 (update 23-10-2000)" mediumref="Web" styleref="1" language="[nl]" creationdate="36820.5137869213"/>
      <run id="8" name="Beleidskunde run 1-9-2000" startdate="36768" status="10" registrationsallowed="true">
        <media>
          <medium idref="7"/>
        </media>
      </run>
      <run id="11" name="Demo run voor verkeer en waterstaat" startdate="36774" status="10" registrationsallowed="false">
        <media>
```

```

        <medium idref="7"/>
      </media>
    </run>
    <run id="9" name="Beleidskunde demo run 1-9-2000"
startdate="36768" status="10" registrationsallowed="false">
      <media>
        <medium idref="7"/>
      </media>
    </run>
  </pct>
  <pct id="24" name="Beleidskunde release 17-1-2001" status="1"
version="1.0.101" creationdate="36906.54108796296">
    <recepy id="8" name="Beleidskunde publicatie 17-1-2001"
mediumref="Web" styleref="1" language="[nl]"
creationdate="36906.54940625"/>
  </pct>
</learners>
</learners>
</uos>
<uos id="FB1E0AA1-5641-11D4-802C-006097721F71" name="Cursus
Gemeentekunde 31-8-2000" defaultrun="7">
  <pct id="22" name="Gemeentekunde release 31-8-2000" status="1"
version="1.0.32" creationdate="36767.658784722225">
    <recepy id="6" name="Publicatie gemeentekunde 31-8-2000 (update
23-10-2000)" mediumref="Web" styleref="1" language="[nl]"
creationdate="36820.516294251545"/>
    <run id="7" name="Gemeentekunde 31-8-2000" startdate="36767"
status="10" registrationsallowed="true">
      <media>
        <medium idref="6"/>
      </media>
    </run>
    <run id="10" name="Gemeentekunde demo run 31-8-2000"
startdate="36768" status="10" registrationsallowed="false">
      <media>
        <medium idref="6"/>
      </media>
    </run>
    <run id="12" name="Demo run verkeer en waterstaat"
startdate="36774" status="10" registrationsallowed="false">
      <media>
        <medium idref="6"/>
      </media>
    </run>
  </pct>
  <pct id="25" name="Gemeentekunde release 17-1-2001" status="1"
version="1.0.101" creationdate="36906.54326388889">
    <recepy id="9" name="Publicatie gemeentekunde 17-1-2001"
mediumref="Web" styleref="1" language="[nl]"
creationdate="36906.55221342592"/>
  </pct>
</learners>
</learners>
</uos>
</root>

```

## Persons

### Syntax

```
...asp?action=persons&id=<run id>
```

De functie geeft de twee hoofdrollen staff en learner terug, evenals de rol "role planner". Tevens worden alle gebruikers in deze rollen teruggeven. Het formaat van de response is:

```

<roles>
  <role id="24" name="Student" class="L" parent="null">
    <person id="990731001"/>

```

```
<person id="990731002" />
<person id="990731003" />
<person id="990731004" />
<recepy id="7" />
</role>
<role id="25" name="Begeleider" class="S" parent="null">
  <recepy id="7" />
</role>
<role id="26" name="Role Planner" class="S" parent="25">
  <recepy id="7" />
</role>
</roles>
```

## *Styles*

### Syntax

```
...asp?action=styles
```

Geeft een lijst terug met alle geïnstalleerde (huis)stijlen in Edubox. Het formaat van de response is

```
<styles>
  <style id="1" name="Open Universiteit Nederland" />
</styles>
```

## *Languages*

### Syntax

```
...asp?action=languages
```

Geeft een lijst terug met alle ondersteunde talen in Edubox. Het formaat van de response is

```
<languages>
  <language id="[nl]" name="Dutch (Nederlands)" />
</languages>
```

## *Media*

### Syntax

```
...asp?action=media
```

Geeft een lijst terug met alle ondersteunde media in Edubox. Het formaat van de response is

```
<media>
  <medium id="Web" name="Publication for the world wide web" />
</media>
```

## *Learners*

### Syntax

```
...asp?action=learners&id=<unit-of-study guid>
```

Geeft een lijst terug met alle learners die nog moeten worden ingeschreven voor deze unit-of-study. Het formaat van de response is

```
<learners>
  <learner id="8001" />
</learners>
```

## Versions

### Syntax

```
...asp?action=versions&id=<unit-of-study guid>
```

Geeft een lijst alle versiesnummers die van de unit-of-study staan geregistreerd in de mda database. Het formaat van de response is

```
<version-list>
  <version name="1.0.101"/>
  <version name="1.0.32"/>
</version-list>
```

## New\_pr

### Syntax

```
...asp?action=new_pr&id=<unit-of-study release id>
```

Bepaalt alle toegestane parameters voor een nieuwe publicatie. Dit zijn de taal, huisstijl en mediumkeuze. Het formaat van de response is

```
<pr>
  <presentationstyle id="1" name="Open Universiteit Nederland"/>
  <medium id="Web" name="Publication for the world wide web"/>
  <language id="[nl]" name="Dutch (Nederlands)"/>
</pr>
```

## uos

### Syntax

```
...asp?action=uos&id=<unit-of-study guid>
```

Geeft alle gegevens van de opgegeven unit-of-study terug in de http response. Het formaat van de response is

```
<uos id="FB1E0AA1-5641-11D4-802C-006097721F71" name="Cursus
Gemeentekunde 31-8-2000" defaultrun="7">
  <pct id="22" name="Gemeentekunde release 31-8-2000" status="1"
version="1.0.32" creationdate="36767.658784722225">
  <recepy id="6" name="Publicatie gemeentekunde 31-8-2000 (update
23-10-2000)" mediumref="Web" styleref="1" language="[nl]"
creationdate="36820.516294251545"/>
  <run id="7" name="Gemeentekunde 31-8-2000" startdate="36767"
status="10" registrationsallowed="true">
    <media>
      <medium idref="6"/>
    </media>
  </run>
  <run id="10" name="Gemeentekunde demo run 31-8-2000"
startdate="36768" status="10" registrationsallowed="false">
```

```

        <media>
          <medium idref="6"/>
        </media>
      </run>
      <run id="12" name="Demo run verkeer en waterstaat"
startdate="36774" status="10" registrationsallowed="false">
        <media>
          <medium idref="6"/>
        </media>
      </run>
    </pct>
    <pct id="25" name="Gemeentekunde release 17-1-2001" status="1"
version="1.0.101" creationdate="36906.54326388889">
      <recept id="9" name="Publicatie gemeentekunde 17-1-2001"
mediumref="Web" styleref="1" language="[nl]"
creationdate="36906.55221342592"/>
    </pct>
    <learners>
  </learners>
</uos>

```

## *pct*

### Syntax

```
...asp?action=pct&id=<unit-of-study release id>
```

Geeft alle gegevens van de unit-of-study release terug. Het formaat van de response is

```

<pct id="23" name="Beleidskunde release 31-8-2000" status="1"
version="1.0.82" creationdate="36767.66368055555">
  <recept id="7" name="Beleidskunde publicatie 31-8-2000 (update 23-10-
2000)" mediumref="Web" styleref="1" language="[nl]"
creationdate="36820.5137869213"/>
  <run id="8" name="Beleidskunde run 1-9-2000" startdate="36768"
status="10" registrationsallowed="true">
    <media>
      <medium idref="7"/>
    </media>
  </run>
  <run id="11" name="Demo run voor verkeer en waterstaat"
startdate="36774" status="10" registrationsallowed="false">
    <media>
      <medium idref="7"/>
    </media>
  </run>
  <run id="9" name="Beleidskunde demo run 1-9-2000" startdate="36768"
status="10" registrationsallowed="false">
    <media>
      <medium idref="7"/>
    </media>
  </run>
</pct>

```

## *recepties*

### Syntax

```
...asp?action=recepties&id=<unit-of-study release id>
```

Bepaalt alle bestaande publicatierecepten voor een nieuwe publicatie. Het formaat van de response is

```
<recepies>
  <pct id="23" name="Beleidskunde release 31-8-2000" status="1"
  version="1.0.82" creationdate="36767.66368055555">
    <recepty id="7" name="Beleidskunde publicatie 31-8-2000 (update 23-
  10-2000)" mediumref="Web" styleref="1" language="[nl]"
  creationdate="36820.5137869213"/>
    <run id="8" name="Beleidskunde run 1-9-2000" startdate="36768"
  status="10" registrationsallowed="true">
      <media>
        <medium idref="7"/>
      </media>
    </run>
    <run id="11" name="Demo run voor verkeer en waterstaat"
  startdate="36774" status="10" registrationsallowed="false">
      <media>
        <medium idref="7"/>
      </media>
    </run>
    <run id="9" name="Beleidskunde demo run 1-9-2000"
  startdate="36768" status="10" registrationsallowed="false">
      <media>
        <medium idref="7"/>
      </media>
    </run>
  </pct>
</recepies>
```

## Ebrelease

Dit interface beschrijft de functie waarmee de clientapplicatie een nieuwe unit-of-study release kan aanmaken. Dit is geïmplementeerd als een http request naar een omnimark server proces

### Syntax

```
/ebrelease.xvc?UOSGUID=<uos guid>&UOSVERSION=<versie van de uos die
gebruikt wordt>&UOSNAME=<logische naam voor de unit-of-study release>
```

## Ebpublication

Dit interface beschrijft de functie waarmee de clientapplicatie een nieuwe unit-of-study release kan aanmaken. Dit is geïmplementeerd als een http request naar een omnimark server proces

### Syntax

```
ebpublication.xvc?RELEASEID=<unit-of-study release id>&MEDIUM=<medium
id>&STYLE=<huisstijl id>&LANGUAGE=<taal id>&DESCRIPTION=<logische naam
van publicatie>
```

## Client tool

De delphi 5 windows applicatie runplanner.exe bevat het userinterface van de applicatie. Deze applicatie maakt gebruik van de standaard libraries van delphi, WinInet en MSXML voor de implementatie. Alle functionaliteiten die de applicatie bevat worden via het userinterface van deze tool aan de eindgebruiker aangeboden.

Het tool volgt de standaard windows gui en de implementatie volgt de standaardmethodiek die gebruikelijk is voor een Delphi Windows applicatie. Deze methodiek wordt bekend veronderstelt. Verdere informatie over het ontwikkelen van een windows applicatie met Delphi staat in de Delphi documentatie of in één van de vele boeken die zijn verschenen over het ontwikkelen met deze programmeeromgeving. Een consequentie van deze manier van implementeren is dat de applicatie een event-driven architectuur volgt. Hierdoor is het niet mogelijk om de flow van het programma in een flowchart te beschrijven, er ontbreekt tenslotte een vaste programma afloop van start tot einde.

Er is wel een metaflow te onderkennen in de applicatie.

11. Start de applicatie
12. Lees het configuratiebestand in
13. Haal de gegevens van de applicatie-tier op
14. Reageer op de door de gebruiker gemaakte keuzes in het interface.
15. Als de gebruiker in stap 4 daarvoor kiest, sluit de applicatie af.

## Het configuratiebestand

Dit bestand bevat de internetadressen van de servercomponenten van de application-tier waarmee het tool communiceert. Door deze parameters via een configuratie aan te bieden is het mogelijk om met een en dezelfde tool meerdere Edubox-servers te beheren.

## Ophalen serverdata

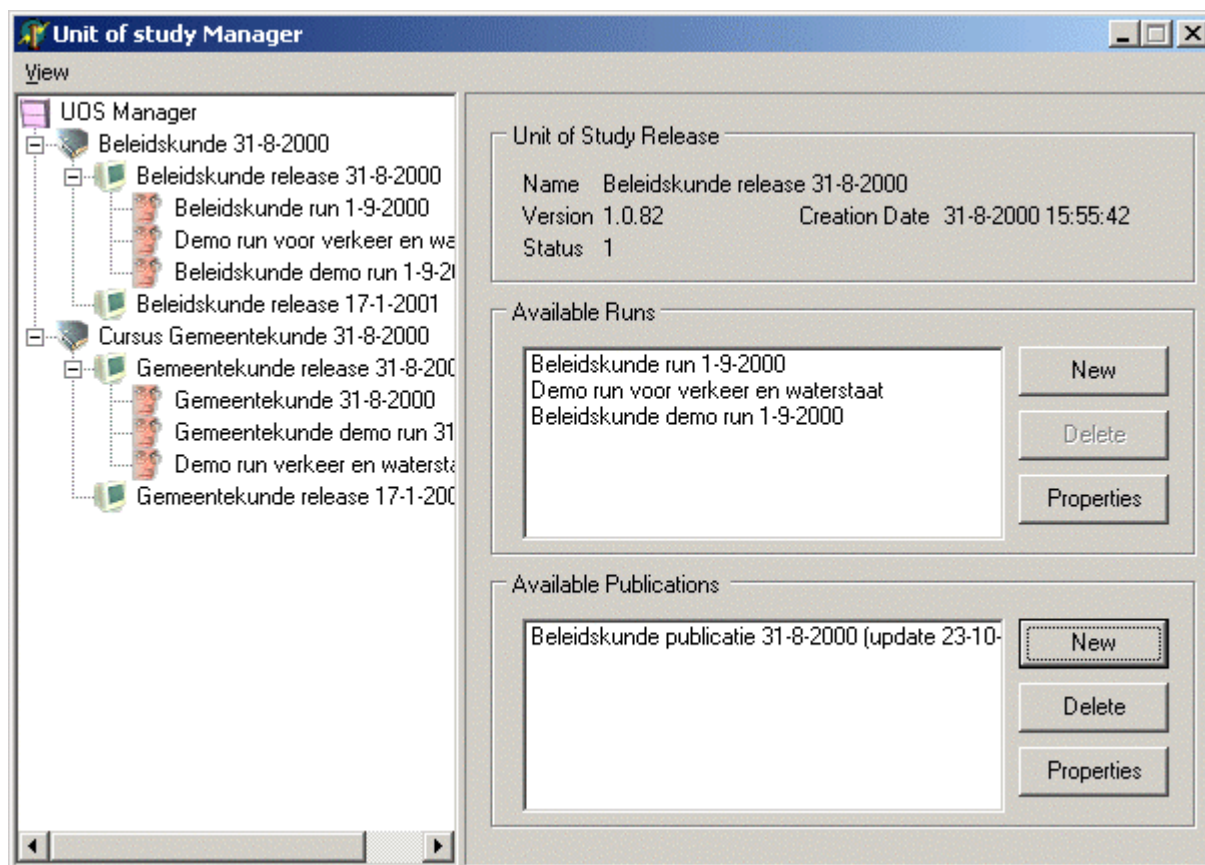
Het tool haalt de gegevens op van de application-tier via communicatie over het http-protocol met het server.asp object. De data wordt niet alleen opgehaald bij het starten van de applicatie, maar kan ook gebeuren op verzoek van de gebruiker door de refresh optie aan te klikken. Verder kan de applicatie deze methode zelf aanroepen om de gegevens te verversen na het verwerken van een opdracht van de gebruiker.

De opgehaalde data worden geparsed met behulp van de msxml parser en vervolgens lokaal verwerkt en opgeslagen in een tree structuur van polymorfe TXMLNode objecten. Dit root object implementeert het parsen van een xml-dom object. De afgeleide objecten als TUOS en TPCT breiden het TXMLNode object uit met de mogelijkheid om extra, object specifieke informatie op te slaan.

## Het userinterface

### *Het applicatievenster*





Zoals uit bovenstaande afbeelding blijkt staan de unit-of-studies centraal in het userinterface van de roleplanner. Ze staan allemaal afgebeeld in de Ttreeview tvUos aan de linkerkant. Door een item aan te klikken wordt de informatie van het betreffende item getoond in een tabblad van de TpageControl pcPages. De informatie die wordt getoond is bereikbaar via de data property van het geselecteerde treeview item.

<b>Level</b>	<b>Tabblad</b>	<b>Type informatie object</b>
<b>Root</b>	TsRoot	-
<b>Uos</b>	TsUos	TUOS
<b>Pct</b>	TsPct	TPCT
<b>Run</b>	TsRun	Trun

Op ieder tabblad kan de gebruiker relevante functionaliteiten aanroepen die betrekking hebben op het getoonde object.

### *Aanmaken nieuwe unit-of-study release*

The screenshot shows a 'New' dialog box with a tab labeled 'Unit of study release'. The dialog contains the following text: 'You can create a new release of "Beleidskunde 31-8-2000" based on the specified version.' Below this, there is a GUID field with the value '553E08E1-357D-11D4-A242-000629E2559F'. The 'Version' field is a dropdown menu currently showing '1.0.101'. The 'Name' field is an empty text box. At the bottom, there are 'OK' and 'Cancel' buttons.

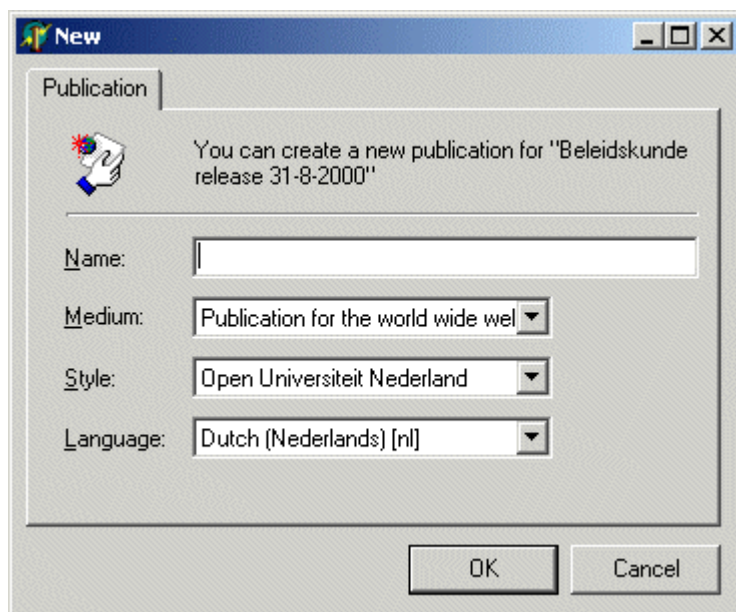
De gebruiker kan met bovenstaand formulier een nieuwe unit of study release aanmaken. De applicatie bepaalt via het `server.asp?action=versions` protocol welke versie van deze unit-of-study kunnen dienen als uitgangspunt voor de aanmaak van de release. De gemaakte keuze wordt via het `ebrelease` protocol uitgevoerd door de application-tier. Een eventuele foutmelding wordt in een webbrowser dialog getoond via de functie `ShowHtmlMessage`.

#### *Aanmaken nieuwe run*

The screenshot shows a 'New' dialog box with a tab labeled 'Run'. The dialog contains the following text: 'You can start a new run for "Beleidskunde release 31-8-2000".' Below this, there is a 'Name' field, a 'Status' dropdown menu showing 'Run created', and a 'Start date' dropdown menu showing '14-2-2001'. There is also a checkbox labeled 'Registrations Allowed' which is currently unchecked. At the bottom, there are 'OK' and 'Cancel' buttons.

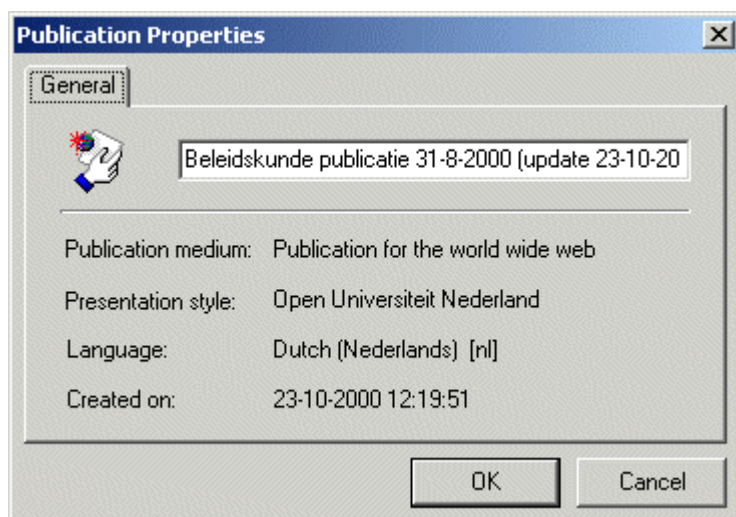
Met het formulier kan de gebruiker een nieuwe run aanmaken van de betreffende unit of study release. De startdatum kan niet voor de huidige datum liggen. De run wordt aangemaakt door het aanroepen van de functie `unServer.AddRun`.

#### *Aanmaken nieuwe publicatie*



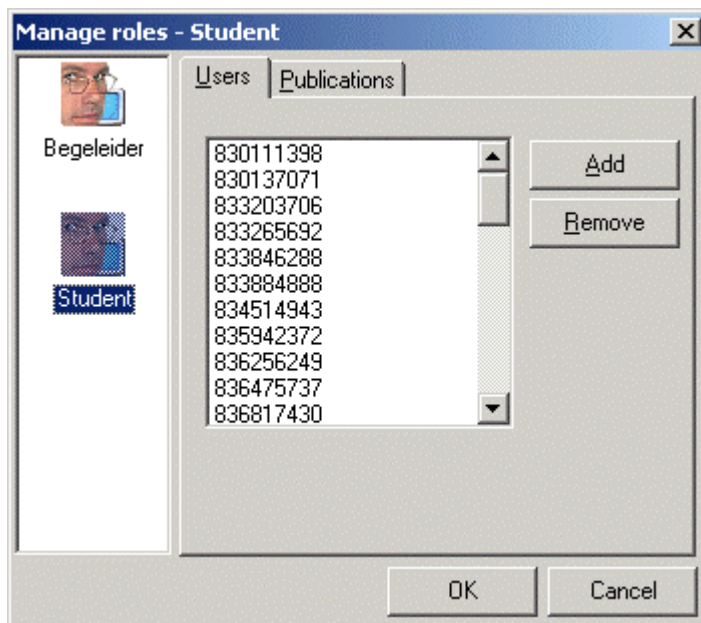
Het programma bepaalt via `server.asp?action=new_pr` welke media, stijlen en talen zijn toegestaan voor het aanmaken van een nieuwe publicatie. Niet alle door Edubox ondersteunde media zijn noodzakelijkerwijs geldige media voor de publicatie van de onderhavige unit of study. Tijdens de import van een unit of study wordt bepaald naar welke media hij gepubliceerd kan worden. Het aanmaken van de publicatie gebeurt door een server component van de applicatie tier. De component wordt aangeroepen via het `ebpublication` interface.

### *Propertes van een publicatie*



Toont de eigenschappen van de geselecteerde publicatie.

## Het managen van een run



Met dit formulier kan de gebruiker personen toekennen aan de hoofdrollen binnen de run. Verder kan de gebruiker publicaties aan deze rollen toekennen. De informatie over de betreffende rol wordt via `server.asp?action=persons` opgehaald via de application tier uit de database. Alleen de wijzigingen worden verwerkt door `unServer.ProcessTransactions`.

## Bestanden

In onderstaande tabel zijn de verschillende bronbestanden opgesomd waaruit de applicatie bestaat.

<b>Naam</b>	<b>Locatie</b>	<b>Functie</b>
runplanner.cfg	.	Delphi configuratie bestand
runplanner.dof	.	Delphi configuratie bestand
runplanner.dpr	.	Source hoofdprogramma
runplanner.dsk	.	Delphi configuratie bestand
runplanner.res	.	Application resource bestand
UnAddPub.dfm	.	Add publication form
UnAddPub.pas	.	Add publication form sources
UnAddUsers.dfm	.	Add users to role form
UnAddUsers.dfm	.	Add users to role form sources
UnChangeRun.dfm	.	Change run properties form
UnChangeRun.ps	.	Change run properties form sources
UnLogging.dfm	.	Logging form
UnLogging.pas	.	Logging form source
UnNewPct.dfm	.	New unit-of-study release form
UnNewPct.pas	.	New unit-of-study release form source
UnNewPub.dfm	.	Create new publication form
UnNewPub.pas	.	Create new publication form sources
UnNewRun.dfm	.	Create new run form

UnNewRun.pas	.	Create new run form sources
UnObjecten.pas	.	Global datastructures sources
UnPersons.dfm	.	Manage user and publication form
UnPersons.pas	.	Manage user and publication form sources
UnPlanner.dfm	.	Main application form
UnPlanner.pas	.	Main application form sources
UnPubProps.dfm	.	Show publication properties form
UnPubProps.pas	.	Show publication properties form source
UnServer.pas	.	Contains database access routines
UnTransfer.dfm	.	Transfer users from run to run form
UnTransfer.pas	.	Transfer users from run to run form sources
UnCommon.pas	..\common\	Standaard objecten
UnHtmlDlg.pas	..\common\	Webbrowser dialog sources
UnHtmlDlg.dfm	..\common\	Webbrowser dialog
UnTools.pas	..\common\	Standaard routines
0language.bmp	images/	
0medium.bmp	images/	
0style.bmp	images/	
1manager.bmp	images/	
1manager16.bmp	images/	
2us.bmp	images/	
2us-16.bmp	images/	
3release.bmp	images/	
3release-16.bmp	images/	
4run.bmp	images/	
4run-16.bmp	images/	
Active.bmp	images/	
Deleted.bmp	images/	
Keys.bmp	images/	
Language.bmp	images/	
Medium.bmp	images/	
Presentationstyle.bmp	images/	
Publication.bmp	images/	
Release 16.bmp	images/	
Release.bmp	images/	
Run 16.bmp	images/	
Run.bmp	images/	
Staff.bmp	images/	
Unit of study 16.bmp	images/	
Unit of study.bmp	images/	
Uos.bmp	images/	
Us-manager 16.bmp	images/	
Us-manager.bmp	images/	

# Legacy koppelingen

## Inleiding

In de dossierdatabase worden ondermeer de properties bijgehouden. De meeste van deze properties worden expliciet of impliciet door de eml gedefiniëerd. Deze worden aangemaakt tijdens de publicatie van het eml-materiaal. Al deze properties hebben dus een relatie met het onderwijsmateriaal. Maar daarnaast zijn er ook nog properties die niet aan het onderwijsmateriaal zijn gekoppeld maar aan de individuele gebruiker. Dit zijn dus persoonsgebonden eigenschappen. Voorbeelden hiervan zijn de naam van de gebruiker, zijn of haar email adres, naw-gegevens etcetera.

Een deel van deze properties wordt indirect door de eml geïmpliceerd; in eml is het mogelijk om te verwijzen naar bepaalde kenmerken van een persoon. Dit is een vaste set aan gegevens te weten:

Naam	Type	beschrijving
_staff	Boolean	Gebruiker is wel/geen staff
_name	String	Naam van de gebruiker
_street	String	Straat van gebruiker
_zip	String	Zip/postcode van gebruiker
_city	String	Woonplaats van de gebruiker
_country	String	Land van de gebruiker
_email	String	Emailadres van de gebruiker
_telephone	String	Telefoonnummer van de gebruiker

Een andere eigenschap van deze properties is dat ze in tegenstelling tot de overige properties niet door een eml-gebruiker of door edubox worden onderhouden, maar dat ze uit een extern systeem afkomstig zijn. Vandaar dat deze properties in de dossierdatabase gegroepeerd zijn onder de noemer legacy properties. Het externe systeem dat deze variabelen aanlevert hoeft overigens geen geautomatiseerd systeem te zijn.

Naast deze door eml verwachte legacy gegevens kunnen er ook nog andere properties gedefiniëerd worden. Dit kan bijvoorbeeld klantspecifieke gegevens betreffen.

In de database zijn de definitie en de inhoud van de properties gescheiden. Er kunnen alleen properties gezet worden waarvan een definitie in het systeem is opgeslagen.

Om de dossierdatabase te vullen zijn drie programma's ontwikkeld:

- ParsePropDefs om de properties te definiëren in de database
- ImportLegacy om de legacygegevens vanuit een xml bestand in de database in te lezen
- ImportInschrijvingen om de inschrijvingen voor een unit-of-study in de database op te slaan.

Alle programma's zijn in Delphi 5 service pack 1 ontwikkelde console applicaties. Ze volgen alle drie hetzelfde stramien:

- Lees inifile in met de configuratie gegevens
- Lees het xml bestand met de data
- Parse dit bestand met de msxml parser
- Doorloop de resulterende DOM boom.
- Sla de data op.

## ParsePropDefs

Het programma ParsePropDefs is een console applicatie die legacy property definities aanmaakt in de dossierdatabase.

Naam	ParsePropDefs.dpr
Versie	2.0.0.0001
Auteur	Harrie Martens
Ontwikkelomgeving	Delphi 5

De definities van de legacy properties worden beschreven in een xml-bestand. Dit bestand moet valide zijn ten opzichte van de propdef.dtd. Deze dtd ziet er als volgt uit:

```
<!ELEMENT PropertyDefs (PropertyDef*)>
<!ELEMENT PropertyDef (InitialValue?)>
<!ATTLIST PropertyDef action (define | update | archive | delete)
"define">
<!ATTLIST PropertyDef name CDATA #REQUIRED>
<!ATTLIST PropertyDef WUID CDATA #REQUIRED>
<!ATTLIST PropertyDef type (LEGACY) "LEGACY">
<!ATTLIST PropertyDef datatype (BOOLEAN | INTEGER | REAL | STRING | DATE |
URL | TEXT | FILE) "STRING">
<!ELEMENT InitialValue (#PCDATA)>
```

Een voorbeeld van een dergelijk xml-bestand is:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE PropertyDefs SYSTEM "propdef.dtd">

<PropertyDefs>
  <PropertyDef type="LEGACY" datatype="BOOLEAN" name="staff"
WUID="_staff">
    <InitialValue>>false</InitialValue>
  </PropertyDef>
  <PropertyDef type="LEGACY" datatype="STRING" name="name"
WUID="_name" />
  <PropertyDef type="LEGACY" datatype="STRING" name="street"
WUID="_street" />
  <PropertyDef type="LEGACY" datatype="STRING" name="zip"
WUID="_zip" />
  <PropertyDef type="LEGACY" datatype="STRING" name="city"
WUID="_city" />
  <PropertyDef type="LEGACY" datatype="STRING" name="country"
WUID="_country" />
  <PropertyDef type="LEGACY" datatype="STRING" name="email"
WUID="_email" />
  <PropertyDef type="LEGACY" datatype="STRING" name="telephone"
WUID="_telephone" />
</PropertyDefs>
```

Het programma ParsePropDefs verwerkt dus een dergelijk xml-bestand en slaat de resultaten op in de dossierdatabase.

De globale werking van het programma is als volgt:

- Initialiseer programma
- Open xml bestand met daarin beschreven de legacy definities
- Parse dit document en sla de individuele definities op in een lijst
- Doorloop de lijst en sla de definities op in de database.

Het programma maakt gebruik van de Microsoft XML parser om de xml file te verwerken. De databasetoegang verloopt via ADO.

Het programma is geparametriseerd met behulp van een ini-file. Dit is een tekstbestand met per regel een parameter. Het algemene formaat van een parameter is

<naam>=<waarde>

Parameter	Functie	Voorbeeld
xml-path	Het te verwerken xml-bestand.	D:\ou\admin\componenten\inschrijvingen\inschrijvingen.xml
db-provider	De door ADO te gebruiken oledb-provider. Deze ligt vast.	SQLOLEDB
waitoncompleted	Bepaalt of de applicatie wacht op <return> na beëindiging of automatisch doorloopt.	Mogelijke waarden zijn 0 en 1: 0 - applicatie wacht niet. 1 - applicatie wacht op <return>
db-connectionstring	Beschrijft de databasekoppeling	Zie onderstaande tabel

**Data Source=e-chi-bdc-3;Initial Catalog=ounl\_dos20;User Id=edumast;Password=eb20intmast**

Parameter	Functie	Voorbeeld
Data Source	De naam van de database server	e-chi-bdc-3
Initial Catalog	De naam van de dossierdatabase van de klant	ounl_dos20
User Id	Gebruiker van deze database	edumast <sup>1</sup>
Password	Wachtwoord van de gebruiker	*****

<sup>1</sup> In het voorbeeld wordt edumast gebruikt als user, deze heeft echter teveel rechten. Gebruik hier in een productieomgeving een database user voor met 'normale' rechten.

---

<sup>1</sup> In het voorbeeld wordt edumast gebruikt als user, deze heeft echter teveel rechten. Gebruik hier in een productieomgeving een database user voor met 'normale' rechten.



## ImportLegacy

Het programma ImportLegacy is een console applicatie die legacy property importeert vanuit een xml bestand in de dossierdatabase.

Naam	ImportLegacy.dpr
Versie	2.0.0.0001
Auteur	Harrie Martens
Ontwikkelomgeving	Delphi 5

De formaat van de legacy properties worden beschreven in een xml-bestand. Dit bestand moet valide zijn ten opzichte van de legacy.dtd. Deze dtd ziet er als volgt uit:

```
<!ELEMENT persons (person*)>
<!ELEMENT person (staff, name, street?, zip?, city?, country?, email,
telephone?)>
<!ATTLIST person userid CDATA #REQUIRED>
<!ELEMENT staff (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT telephone (#PCDATA)>
```

Een voorbeeld van een dergelijk xml-bestand is:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE persons SYSTEM "legacy.dtd">
<persons>
  <person userid="ham">
    <staff>true</staff>
    <name>Harrie Martens</name>
    <email>harrie.martens@ou.nl</email>
  </person>
  <person userid="hvo">
    <staff>true</staff>
    <name>Hubert Vogten</name>
    <email>hubert.vogten@ou.nl</email>
  </person>
  <person userid="veg">
    <staff>true</staff>
    <name>Wim van der Vegt</name>
    <email>wim.vandervegt@ou.nl</email>
  </person>
  <person userid="mve">
    <staff>true</staff>
    <name>Marc Verhooren</name>
    <email>marc.verhooren@ou.nl</email>
  </person>
  <person userid="rkp">
    <staff>true</staff>
    <name>Rob Koper</name>
    <email>rob.koper@ou.nl</email>
  </person>
  <person userid="jzb">
    <staff>true</staff>
    <name>Jeroen Berkhout</name>
    <email>jeroen.berkhout@ou.nl</email>
  </person>
  <person userid="fbr">
```

```

    <staff>true</staff>
    <name>Francis Brouns</name>
    <email>francis.brouns@ou.nl</email>
  </person>
  <person userid="wis">
    <staff>true</staff>
    <name>Wim Slot</name>
    <email>wim.slot@ou.nl</email>
  </person>
</persons>

```

Het programma ImportLegacy verwerkt dus een dergelijk xml-bestand en slaat de resultaten op in de dossierdatabase.

Het programma maakt gebruik van de Microsoft XML parser om de xml file te verwerken. De databasetoegang verloopt via ADO.

Het programma is geparametriseerd met behulp van een ini-file. Dit is een tekstbestand met per regel een parameter. Het algemene formaat van een parameter is

<naam>=<waarde>

Parameter	Functie	Voorbeeld
xml-path	Het te verwerken xml-bestand.	D:\ou\admin\componenten\inschrijvingen\inschrijvingen.xml
db-provider	De door ADO te gebruiken oledb-provider. Deze ligt vast.	SQLOLEDB
waitoncompleted	Bepaalt of de applicatie wacht op <return> na beëindiging of automatisch doorloopt.	Mogelijke waarden zijn 0 en 1: 0 - applicatie wacht niet. 1 - applicatie wacht op <return>
db-connectionstring	Beschrijft de databasekoppeling	Zie onderstaande tabel

**Data Source=e-chi-bdc-3;Initial Catalog=ounl\_dos20;User Id=edumast;Password=eb20intmast**

Parameter	Functie	Voorbeeld
Data Source	De naam van de database server	e-chi-bdc-3
Initial Catalog	De naam van de dossierdatabase van de klant	ounl_dos20
User Id	Gebruiker van deze database	edumast <sup>2</sup>
Password	Wachtwoord van de gebruiker	*****

<sup>2</sup> In het voorbeeld wordt edumast gebruikt als user, deze heeft echter teveel rechten. Gebruik hier in een productieomgeving een database user voor met 'normale' rechten.

## ImportInschrijvingen

Het programma ImportInschrijvingen is een console waarmee studenten aan een of meerdere unit-of-studies gekoppeld worden in de dossierdatabase.

Naam	ImportInschrijvingen.dpr
Versie	2.0.0.0001
Auteur	Harrie Martens
Ontwikkelomgeving	Delphi 5

De inschrijvingen worden beschreven in een xml-bestand. Dit bestand moet valide zijn ten opzichte van de inschrijvingen.dtd. Deze dtd ziet er als volgt uit:

```
<!ELEMENT inschrijvingen (person*)>
<!ELEMENT person (uos*)>
<!ATTLIST person userid CDATA #REQUIRED>
<!ELEMENT uos EMPTY>
<!ATTLIST uos guid CDATA #REQUIRED>
```

Een voorbeeld van een dergelijk xml-bestand is:

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE inschrijvingen SYSTEM "inschrijvingen.dtd">
<inschrijvingen>
  <person userid="8001">
    <uos guid="6F2D0EE4-0B9D-11D4-AF57-0004ACA24CC4"/>
  </person>
  <person userid="8003">
    <uos guid="6F2D0EE4-0B9D-11D4-AF57-0004ACA24CC4"/>
  </person>
  <person userid="8005">
    <uos guid="6F2D0EE4-0B9D-11D4-AF57-0004ACA24CC4"/>
  </person>
  <person userid="8006">
    <uos guid="6F2D0EE4-0B9D-11D4-AF57-0004ACA24CC4"/>
  </person>
  <person userid="8007">
    <uos guid="6F2D0EE4-0B9D-11D4-AF57-0004ACA24CC4"/>
  </person>
</inschrijvingen>
```

Het programma ImportInschrijvingen verwerkt dus een dergelijk xml-bestand en slaat de resultaten op in de dossierdatabase. De gegevens worden opgeslagen in de UosRegistration tabel.

Het programma maakt gebruik van de Microsoft XML parser om de xml file te verwerken. De databasetoegang verloopt via ADO.

Het programma is geparametriseerd met behulp van een ini-file. Dit is een tekstbestand met per regel een parameter. Het algemene formaat van een parameter is

```
<naam>=<waarde>
```

Parameter	Functie	Voorbeeld
xml-path	Het te verwerken xml-bestand.	D:\ou\admin\componenten

		\ inschrijvingen\ inschrijvingen.xml
db-provider	De door ADO te gebruiken oledb-provider. Deze ligt vast.	SQLOLEDB
waitoncompleted	Bepaalt of de applicatie wacht op <return> na beëindiging of automatisch doorloopt.	Mogelijke waarden zijn 0 en 1: 0 – applicatie wacht niet. 1 – applicatie wacht op <return>
db-connectionstring	Beschrijft de databasekoppeling	Zie onderstaande tabel

**Data Source**=*e-chi-bdc-3*;**Initial Catalog**=*ounl\_dos20*;**User Id**=*edumast*;**Password**=*eb20intmast*

Parameter	Functie	Voorbeeld
Data Source	De naam van de database server	e-chi-bdc-3
Initial Catalog	De naam van de dossierdatabase van de klant	ounl_dos20
User Id	Gebruiker van deze database	edumast <sup>3</sup>
Password	Wachtwoord van de gebruiker	*****

**Otec 2001/14**

---

<sup>3</sup> In het voorbeeld wordt edumast gebruikt als user, deze heeft echter teveel rechten. Gebruik hier in een productieomgeving een database user voor met 'normale' rechten.