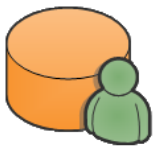


WisaReg

Maxim Van de Wynckel

Download als PDF





WisaReg Documentatie
Don Bosco Hoboken 2012-2013
© Maxim Van de Wynckel

1. Inleiding WisaReg	5
1.1 Structuur.....	5
1.1.1 Systeem	5
1.1.2 Programma.....	5
2. Verbindingen met Databases	6
2.1 ODBC/OLEDB Drivers.....	6
2.1.1 Beschikbare drivers	6
2.1.2 Toevoegen van drivers	7
2.2 Kopie Wisa naar Lokale database.....	8
2.2.1 Ophalen van gegevens uit Wisa	8
2.2.2 Gegevens worden gesaved	11
2.3 Update Wisa naar Lokale database.....	11
2.3.1 Duurtijden synchronisaties	11
2.4 Toevoegen van gebruikers in <Dokeos>	12
2.5 Verwijderen van gebruikers in <Dokeos>	12
2.6 Synchroniseren van wachtwoorden.....	12
2.6.1 Ophalen uit LDAP	12
2.6.2 Wachtwoordwijzigingen in LDAP onderscheppen	13
2.6.2 Invoegen in een database	16
2.7 Module dbDriver	16
2.7.1 Documentatie API	16
2.8 Module dbSync.....	20
2.8.1 Documentatie API	20
2.8.2 Code Kopie van Wisa naar Lokaal	22
2.8.3 Schematische voorstelling koppie Wisa naar Lokaal	24
2.9 Module config_ini.....	25
2.9.1 Documentatie API	25
2.10 Module reg_edit.....	26
2.10.1 Documentatie API	26
3. Gebruikersinterface.....	27
3.1 Configuratie	27
3.1.1 Wisa database configuratie.....	28
3.1.2 Lokale database configuratie	29
3.1.3 Sync. database configuratie	31
3.1.3 LDAP configuratie	31
3.1.4 WisaReg Settings paneel	32
4. Hoofdprogramma.....	33
4.1 Beveiliging	33
4.2 Programmaverloop	34
4.2.1 Opstart verloop	34
4.2.2 Polling verloopt	34
4.3 Commando's.....	35
4.4 Root directory structuur.....	36
4.4.1 WisaReg benodigde files	36

5. Commercialisering programma.....	37
5.1 Naam en logo	37
5.2 Commercieel bruikbare afbeeldingen.....	38
5.3 Licentie voor WisaReg	38
5.4 Compatibiliteit.....	38
5.5 Application Interface	39
5.6 Smartschool integreren.....	40
5.6.1 Lezen van de API documentatie van smartschool	40
5.6.2 Maken van een interface	40
5.6.3 De interface laten starten	42
6. Failsafes	43
6.1 Beschermen tegen SQL Errors.....	43
6.2 Beschermen tegen stroompannes	43
6.2.1 Tussen twee synchronisaties.....	43
6.2.2 Tijdens een synchronisatie	44
6.3 Beschermen tegen resource tekorten	44
7. Praktische test.....	45
7.1 Installatie	45
7.2 Configuratie	45
7.3 Starten van WisaReg	46
7.4 Syncen naar platform	47
7.5 Syncen van wachtwoorden	48
8. Gekende bugs.....	49
9. Niet geïmplementeerde functies	50
9.1 Module config_langldr	50
9.2 Updater.....	53
9.3 Windows share per gebruiker	53
10. Besluit.....	54
11. Bibliografie	55

Voorwoord

Mijn naam is Maxim Van de Wynckel. Ik ben 18 jaar en programmeer al sinds mijn 9^{de}. Ik zit in mijn laatste jaar Industriële Informatie en Communicatie Technologieën.

Ik heb dit project gekozen omdat ik iets wou doen wat ik graag doe, en wat ik ook goed doe. Namelijk programmeren. Het project lag me omdat het vertrekt uit een simpel idee dat naarmate je verder gaat zoeken, steeds moeilijker wordt.

Het project bevat kennis van de vakken: Netwerken en Programmeren. Maar ik heb wel een andere programmeertaal gebruik dan dat we hebben geleerd.

In deze documentatie zal ik uitleggen hoe dit precies in zijn werk gaat. Graag zou ik volgende personen willen bedanken:

Dhr. Mertens – *Voor het toestaan van het project, toegang geven tot benodigde data*

Dhr. Wielandt – *Tijd vrijmaken voor de GIP*

Dhr. Thijs – *Het nakijken van mijn documentatie*

Dhr. Van Reck – *Tijd vrijmaken voor de GIP*

1. Inleiding WisaReg

WisaReg is een applicatie die nieuw ingeschreven leerlingen uit de Wisa databank accounts geeft voor verschillende services. Het systeem maakt accounts voor de Active Directory van Windows, het Virtueel Leerplatform, het leerlingen volgsysteem en ook voor een Gmail e-mail adres.

Wisa is een administratiepakket dat de meeste scholen gebruiken om leerlingen-en personeel administratie te regelen. De gegevens die hiermee worden aangemaakt worden gesaved in een Firebird SQL database.

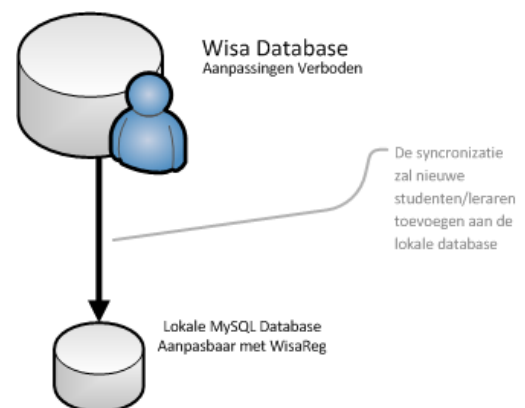
Zodra ergens een wachtwoord wordt gewijzigd zal dit over alle services ook wijzigen, en wanneer er een student uitgeschreven wordt, zullen ook zijn accounts vermits enkele beperkingen ook verwijderd worden.

1.1 Structuur

1.1.1 Systeem

WisaReg is een programma gemaakt in **C#**. Het is gemaakt zodat het zonder probleem op een server kan draaien, en dat eventuele uitbreidingen in “.NET” mogelijk zijn.

Op de server waar deze applicatie draait moet een MySQL server geïnstalleerd zijn. Deze wordt gebruikt om de data die benodigd is om voor alle leerlingen accounts aan te maken op te slaan, hier kunnen dan ook extra parameters worden gegeven zodat men kan kiezen welke accounts moeten gesynchroniseerd worden.



1.1.2 Programma

Het programma is opgebouwd uit een Framework dat voornamelijk zorgt voor de **UI** naar de gebruiker toe. Er worden enkele zelfgemaakte componenten (DLL's) geladen die voor de voornamelijk verbindingen zorgen met de **databases**.

In de map 'config' in de root van het programma worden de instellingen gesaved. Deze bestaan uit de basisinstellingen (IP Wisa, Username, Password, ...) en uit de te synchroniseren databanken. De instellingen worden als gesaved als INI-files.

Ik koos voor deze indeling omdat INI-files gemakkelijk aanpasbaar zijn.

2. Verbindingen met Databases

Verbindingen met databases worden gemaakt met:

- ODBC
- OLEDB

Deze worden basis gebruikt voor **MySQL, Firebird, Access en Active Directory**

De drivers dienen te worden toegevoegd aan de drivers in de configuratie, en eventueel bijhorende connectoren dienen te worden geïnstalleerd.

De basis voor het maken van deze verbindingen bevindt zich in de component "**db_driver.dll**". Deze file wordt gebruikt om:

- 1) verbindingen te maken met databases
- 2) commando's uit te voeren op een database (SQL)
- 3) instellingen betreffend met de database te laden

























Deze component wordt echter niet rechtstreeks aangesproken als het gaat om verbindingen met databases te maken. Hiervoor wordt "**db_sync.dll**" opgeroepen, om meerdere databases met elkaar te synchroniseren.

































2.1 ODBC/OLEDB Drivers

2.1.1 Beschikbare drivers

ODBC of OLEDB-drivers zijn één van de meest gebruikte drivers voor het verbinden met databases, het spreek dus voor zich dat er talloze drivers beschikbaar zijn.

Op de site <http://www.connectionstrings.com> vindt u een lijst met beschikbare connection strings voor de drivers.

	SQL Server 2012		ComputerEase
	SQL Server 2008		Sybase Adaptive
	SQL Server 2005		Informix
	SQL Server 2000, 7.0		Postgre SQL
	SQL Server Compact		UniVerse
	Oracle		UniData
	IBM DB2		AS/400 (IBM iSeries)
	MySQL		Firebird
	Sybase Advantage		Interbase
	Pervasive		SQLBase
	Netezza DBMS		SQLite
	Paradox		Mimer SQL

 Teradata	 VistaDB
 Valentina	 Progress
 Ingres	 Lightbase
 EffiProz	Excel 2007
 Visual FoxPro / FoxPro 2.x	 DBF / FoxPro
 ZIM	 Excel
 DSN	 Access
 Filemaker	 OpenOffice SpreadSheet
 Caché	 DBMaker
 Textfile	Access 2007
 SQL Azure	 Integration Services
 Windows Azure Storage	 SharePoint
 OLAP, Analysis Services	 MS Project
 Exchange	 Intuit QuickBase
 UDL	 Active Directory
 Composite Information Server	 Lotus Notes
 Index Server	 HTML Table

In bovenstaande tabel vindt U enkele vetgedrukte databases. Dit zijn de databases die hoogstwaarschijnlijk zullen worden gebruikt in het programma.

2.1.2 Toevoegen van drivers

Het toevoegen van drivers kan gebeuren in de instellingen van WisaReg.

Men moet volgende gegevens invoeren:

- **Connection String** die men kan verkrijgen op www.connectionstrings.com
- **Naam** de naam van de driver
- **Verkorte naam** de naam die gebruikt wordt om te sorteren
- **Installer** eventueel benodigd bestand voor de driver

1. Ga naar de map 'config\database\' in de installatie map van het programma.
2. Open de file 'drivers.ini' met wordpad of notepad++
3. Vind volgende regel:

```

22 # Add new drivers under this Drivers.list (Seperate with ;)
23 [Drivers]
24 list=mysql;firebird;access;text;activedirectory

```

4. Voeg een ';' toe, gevolgd door de **Verkorte naam**
5. Voeg helemaal onderaan de file volgende gegevens in:

```

[mijndriver]
name=Mijn Driver
constr=DRIVER={Mijn Driver};Dbq=%DATABASE%;Uid=%USERNAME%;Pwd=%PASSWORD%;

```

Waarbij tussen de [] uw verkorte naam komt.
Achter **Name** de volledige naam komt.
Achter **constr** de connection string komt.

Meer informatie over de variabelen in de connection string kunnen gevonden worden in de File zelf.

2.2 Kopie Wisa naar Lokale database

De kopie van Wisa met de Lokale database gebeurt in twee stappen. Eerst worden de gegevens uit Wisa opgehaald en tijdelijk bewaard. Vervolgens zullen deze gegevens in een lokale database worden opgeslagen, zodat er niet steeds een connectie met een andere server moet gebeuren als er moet worden gesynchroniseerd.

2.2.1 Ophalen van gegevens uit Wisa

Eerst wordt er een 'SELECT' uitgevoerd op de Wisa database. Deze haalt alle gegevens op van de leerlingen die dat jaar les krijgen of van de leerkrachten die dat jaar les geven.

De gebruiker dient de query zelf op te stellen, aangezien het zeer moeilijk is om een vaste logica te gebruiken in de tabellen die Wisa genereert. Deze tabellen zijn voor elke school anders. Daarom heb ik ervoor gekozen heb dat de query die leerlingen en leerkrachten uit de firebird database van Wisa haalt door de gebruiker wordt opgemaakt.

Men hoeft geen aparte query te maken om leerlingen van een verschillende graad op te halen. Dit kan gewoon gewijzigd worden door een specifieke waarde toe te voegen bij de filter.

Omdat de query dynamisch moet zijn voor elk jaar. Zijn er twee argumenten bijgevoegd in het programma die het begin en einde van het schooljaar aanduiden.

Voorbeeld:

Men kan leerlingen van de 1^{ste} graad onderscheiden door volgende filter toe te voegen:

```
AND ((Klas.KL_CODE LIKE '%1%') OR (Klas.KL_CODE LIKE '%2%'))
```

Hiermee zal de query enkel de leerlingen ophalen waar in de klas een `1` of een `2` voorkomt.

Ophalen Leerlingen (SELECT QUERY)

SELECT
DISTINCT

```

Leerling.LL_ID,
Leerling.LL_EMAIL,
InUit.IU_STAMBOEKNUMMER AS Stamboeknummer,
Leerling.LL_NAAM AS Naam,
Leerling.LL_VOORNAAM AS Voornaam,
Leerling.LL_ROEPNAAM AS Roepnaam,
Leerling.LL_NAAMSORT AS NaamS,
Leerling.LL_VOORNAAMSORT AS VoornaamS,
Leerling.LL_GESLACHT AS Geslacht,
Leerling.LL_GEBOORTEDATUM AS Geboortedatum,
Gemeente.GM_FusieGemeente AS GeboortePlaats,
Leerling.LL_RIJKSREGISTERNR AS Rijksregisternummer,
TESTEN1.NA_OMSCHRIJVING AS Nationaliteit,
TEST.P_OMSCHRIJVING AS Herkomst,
LeerlingAdres.LA_STRAAT AS Straat,
LeerlingAdres.LA_STRAATNR AS Huisnummer,
LeerlingAdres.LA_STRAATBUS AS Postbus,
Gemeente.GM_DEELGEMEENTE AS Gemeente,
Gemeente.GM_POSTCODE AS Postnummer,
LeerlingAdres.LA_TELEFOON AS Telefoonnummer,
Leerling.LL_NOODTELEFOON AS Noodtel,
Leerling.LL_VADERNAAM AS Naam_vader,
Leerling.LL_VADERVOORNAAM AS Voornaam_vader,
TESTEN2.NA_OMSCHRIJVING AS Nationaliteit_vader,
TEST2.P_OMSCHRIJVING AS Beroep_Vader,
Leerling.LL_MOEDERNAAM AS Naam_moeder,
Leerling.LL_MOEDERVOORNAAM AS Voornaam_moeder,
TESTEN3.NA_OMSCHRIJVING AS Nationaliteit_moeder,
TEST3.P_OMSCHRIJVING AS Beroep_Moeder,
Klas.KL_CODE AS Ingeschreven_in_klas,
Loopbaan.LB_KLASNUMMER AS KLASnr,
Instellingparmtab.IP_CODE AS Middag

```

GEGEVENS KRIJGEN EEN GEMAKKELIJKERE NAAM
LL_ID,LL_EMAIL, Stamboeknummer, Naam ,..

```

FROM (((((((((((Klas INNER JOIN Klasgroep ON Klas.Kl_Id = Klasgroep.KG_Klas_FK)
INNER JOIN Schooljaar ON Klas.KL_Schooljaar_FK = Schooljaar.SJ_ID)
INNER JOIN Loopbaan ON Klasgroep.KG_ID = Loopbaan.LB_Klasgroep_FK)
INNER JOIN InUit ON Loopbaan.LB_InUit_FK = InUit.IU_ID)
INNER JOIN Leerling ON InUit.IU_Leerling_FK = Leerling.LL_ID)
LEFT JOIN Instellingparmtab ON Loopbaan.LB_MIDDAGMAAL_FKP = Instellingparmtab.IP_ID)
LEFT JOIN PARMTAB TEST ON Leerling.LL_HERKOMST_FKP = TEST.P_ID)
LEFT JOIN PARMTAB TEST2 ON Leerling.LL_VADERBEROEP_FKP = TEST2.P_ID)
LEFT JOIN PARMTAB TEST3 ON Leerling.LL_MOEDERBEROEP_FKP = TEST3.P_ID)
LEFT JOIN Nationaliteit TESTEN1 ON TESTEN1.NA_ID = LEERLING.LL_NATIONALITEIT_FK)
LEFT JOIN Nationaliteit TESTEN2 ON TESTEN2.NA_ID = Leerling.LL_VADERNATIONALITEIT_FK)
LEFT JOIN Nationaliteit TESTEN3 ON TESTEN3.NA_ID = Leerling.LL_MOEDERNATIONALITEIT_FK)
LEFT JOIN Stamboekregister ON Leerling.LL_ID = Stamboekregister.SR_Leerling_FK)
LEFT JOIN LeerlingAdres ON Leerling.LL_ID = LeerlingAdres.LA_Leerling_FK)
LEFT JOIN Gemeente ON LeerlingAdres.LA_Gemeente_FK = Gemeente.GM_ID)

```

WHERE

```

(Schooljaar.SJ_Code = '@@SCHOOL_A')
AND (Klas.KL_VestigingsPlaats_FK = '1')
AND (Loopbaan.LB_VAN >= '09/01/@@SCHOOL_A')
AND (Loopbaan.LB_TOT = '06/30/@@SCHOOL_B')
AND (Leerlingadres.LA_TYPEADRES_FKP = '1018')

```

VOORWAARDEN OM LEERLINGEN TE SELECTEREN

Loopbaan van 1 sept (@@SCHOOL_A is jaarbegin)
Loopbaan tot 30 juni (@@SCHOOL_B is jaareinde)

ORDER BY Klas.KL_CODE,Loopbaan.LB_KLASNUMMER;

Ophalen Leerkrachten (SELECT QUERY)

SELECT DISTINCT

```
PERSENEEL.PS_ID,  
PERSENEEL.PS_CODE AS AFK,  
PERSENEEL.PS_NAAM AS NAAM,  
PERSENEEL.PS_NAAMSORT,  
PERSENEEL.PS_VOORNAAM,  
PERSENEEL.PS_VOORNAAMSORT,  
PERSENEELADRES.PA_GEMEENTE_FK,  
PERSENEELADRES.PA_STRAAT,  
PERSENEELADRES.PA_STRAATBUS,  
PERSENEELADRES.PA_STRAATNR,  
PERSENEELADRES.PA_TELEFOON,  
PERSENEELADRES.PA_GSM,  
PERSENEELADRES.PA_EMAIL,  
PERSENEELADRES.PA_TYPEADRES_FKP,  
GEMEENTE.GM_POSTNUMMER,  
GEMEENTE.GM_DEELGEMEENTE
```

GEGEVENS KRIJGEN EEN GEMAKKELIJKERE NAAM

FROM

```
PERSENEEL , PERSENEELADRES, GEMEENTE, DIENSTVERBAND, INSTELLING
```

WHERE

```
PERSENEEL.PS_ID = PERSENEELADRES.PA_PERSONEEL_FK  
AND GEMEENTE.GM_ID = PERSENEELADRES.PA_GEMEENTE_FK  
AND PERSENEELADRES.PA_TYPEADRES_FKP  
  IN (select P_ID From Parmtab Where P_Code = 'O' And P_Type = 'TPAD')  
AND PERSENEEL.PS_ID = DV_Personeel_fk  
AND ('05/26/2012' between DV_VAn and DV_tot)  
AND dv_school_fk = is_school_fk  
AND is_id = 2
```

VOORWAARDEN VOOR HET KIEZEN

ORDER BY PS_NAAMSORT , PS_VOORNAAMSORT

2.2.2 Gegevens worden gesaved

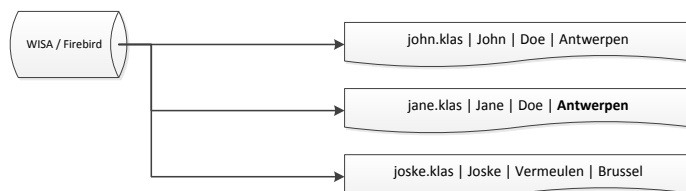
Deze gegevens worden vervolgens met een 'INSERT' op de lokale database gesaved.

De werking van dit proces duurt ongeveer 1 minuut

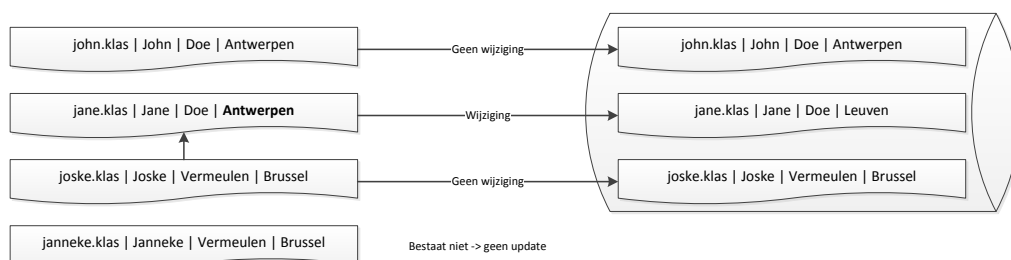
2.3 Update Wisa naar Lokale database

Omdat het te veel tijd kost om telkens de hele database te gaan weghalen, en het ook te veel tijd kost om te gaan kijken wat er bijgekomen is. Heb ik gekozen voor het update van de Lokale database op twee afzonderlijke manieren te doen.

- 1) Alles wordt terug uit Wisa gehaald. En met het 'UPDATE' statement worden bestaande leerlingen/leerkrachten geüpdatet.



Tijdelijk saven naar lijst



```
UPDATE TABLE 'lln' SET 'user'='joske.klas', 'firstname'='Joske', 'surname'='Vermeulen', 'location'='Brussel' WHERE 'user'='joske.klas'
```

- 2) Na elke (dag) wordt heel de database leeg gehaald, en dan terug een kopie gemaakt van Wisa.

Deze methode is het meest betrouwbaar. De kans dat een leerling een computer nodig heeft de eerste dag dat hij (in het midden van het jaar) op school wordt ingeschreven is zeer klein.

2.3.1 Duurtijden synchronisaties

Opdracht	Duurtijd voor 500 personen
Wisa naar lokaal <u>Koppieren</u>	1 minuut
LDAP accounts <u>Aanmaken</u>	2 minuten
Lokaal naar platform <u>Koppieren</u>	30 seconden
Lokaal <u>Updaten</u>	90 seconden
Platform accounts <u>Updaten</u>	50 seconden
LDAP accounts <u>Deleten</u>	1 minuut
Platform accounts <u>Deleten</u>	50 seconden

2.4 Toevoegen van gebruikers in <Dokeos>

Het toevoegen van gebruikers in een service zoals Dokeos of het Volgsysteem, gebeurt via een gemakkelijke configuratie. De gebruiker moet enkel de tabel weten van de gebruikers, en de kolommen definiëren aan een variabele uit de lokale database (vb. "LL_Voornaam"). Indien de tabel al bestaat kunnen de kolommen opgehaald worden.

Met deze informatie wordt dan een insert gedaan in de database waar de service zich bevindt. De Update en Insert statement worden door het programma gegenereerd.

2.5 Verwijderen van gebruikers in <Dokeos>

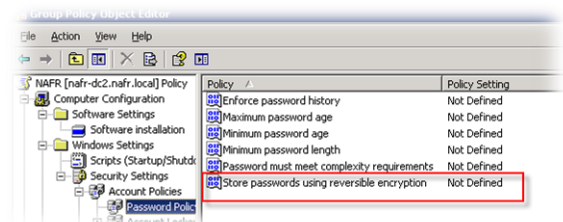
Het verwijderen van gebruikers in een platform zoals Dokeos is moeilijk. Aangezien nagegaan moet worden of de gebruiker 'nog' bestaat. Als WisaReg telkens zou moeten controleren of er een leerling is uitgeschreven. Dan zou elke rij moeten worden gecontroleerd in de database waar de gebruikers staan. Dit is zeer tijdrovend, en wordt daarom dus niet toegepast.

2.6 Synchroniseren van wachtwoorden

Het synchroniseren van wachtwoorden gebeurt vooraleer de lokale database wordt gedropt. WisaReg zal de wachtwoorden uit **LDAP** halen zodra deze wijzigen en dan permanent opslagen in een persistente tabel van de lokale database.

2.6.1 Ophalen uit LDAP

Mijn eerste voorkeur om wachtwoorden uit LDAP te halen was met het gebruik van reversible encryption. Dit staat toe om een geencrypteerd wachtwoord uit LDAP te halen, en via een serie van algoritmes het wachtwoord in leesbare tekst te bekomen.



Voordelen:

- 1) Wachtwoorden kunnen op eender welk tijdstip worden opgevraagd

Nadelen:

- 1) Enkel programmeerbaar in C++ of een andere Low level taal
- 2) Vereist aanpassing bij elke Windows Server versie
- 3) Niet officieel toegestaan door applicaties door Microsoft
- 4) Grote veiligheidslek

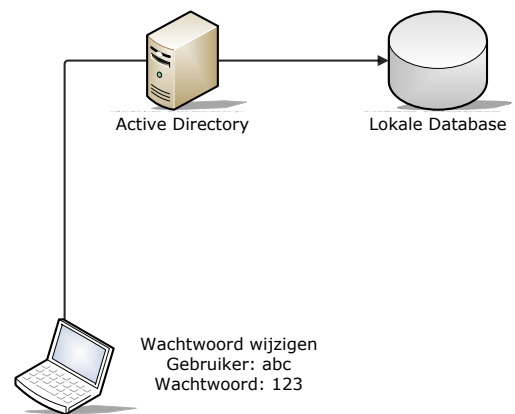
Hoewel er veel nadelen waren, was het wel handig om het wachtwoord op elk moment (moment van een sync.) te kunnen opvragen. Maar omdat het bij elke Windows versie opnieuw moet gecompileerd worden, kon ik het niet gebruiken.

2.6.2 Wachtwoordwijzigingen in LDAP onderscheppen

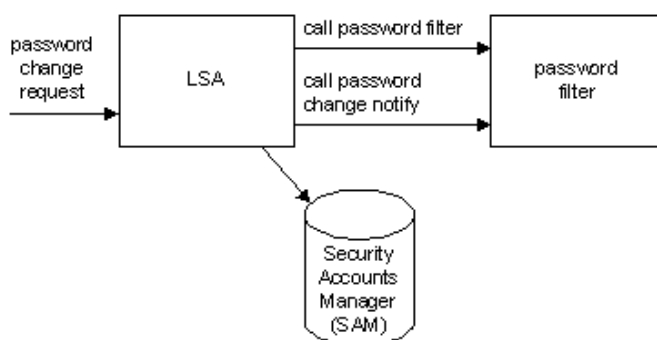
Om de wachtwoordwijzigingen te onderscheppen, maak ik gebruik van wachtwoordfilters. Deze filters zijn geregistreerde dll's die het wachtwoord en gebruikersnaam in leesbare tekst krijgen alvorens dit wordt gesaved of goedgekeurd in het **SAM** (Microsoft database voor wachtwoorden).

Het idee achter deze filters is om gebruikers in staat te stellen om hun eigen manier van encryptie toe te voegen, maar ik gebruik het enkel om de wachtwoorden te onderscheppen.

- 1) Bij de start van het schooljaar krijgt elke gebruiker een standaard wachtwoord, dit wachtwoord wordt gesaved in de lokale database op een persistente plaats.
- 2) Zodra er een wachtwoordwijziging in de Active Directory is, wordt het wachtwoord in de lokale database vernieuwd, en bij de eerst volgende synchronisatie verspreid over alle databases/services.



Om een wachtwoordfilter te maken en registreren moet je C++ of een andere Low level programmeertaal gebruiken. Aangezien Windows niet toestaat dat corebestanden worden geschreven onder een framework.

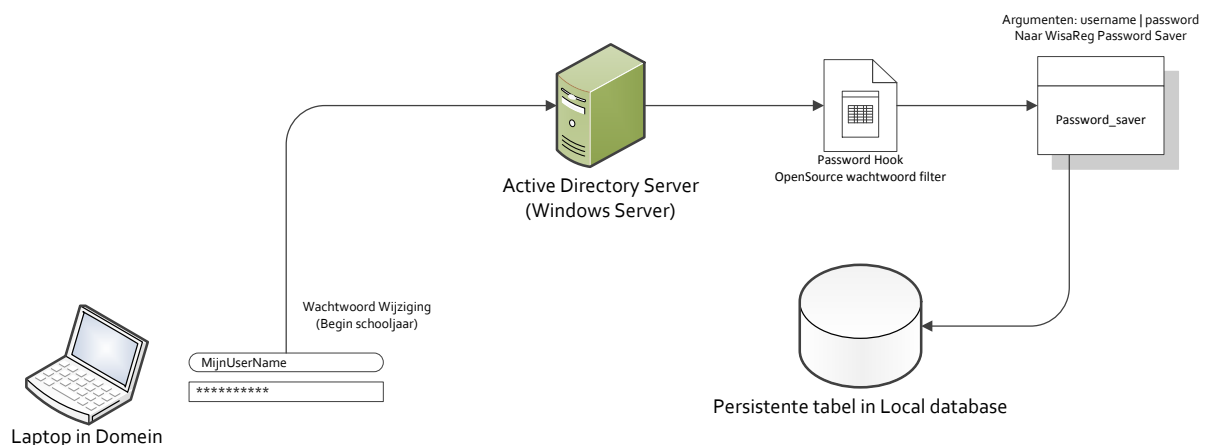


We hebben met de wachtwoordfilter toegang tot twee events. Maar we willen enkel weten wanneer het wachtwoord wijzigt.

De eigen DLL zelf moet in de installatie map van Windows staan (C:\WINDOWS\system32\) en moet geregistreerd worden in de registry key **Notification Packages**

Pad naar Registry key
HKEY_LOCAL_MACHINE
SYSTEM
CurrentControlSet
Control
Lsa

Hoewel het me gelukt is om een wachtwoordfilter te maken, maak ik gebruik van **Password Hook**. Dit is een wachtwoordfilter die maandelijks wordt gecontroleerd. Het is steeds up-to-date, en omdat Microsoft de interface van de wachtwoordfilters vaak wijzigt, heb ik geopteerd om dit programma te gebruiken om de wachtwoord wijzigingen naar WisaReg te sturen.



Figuur: Gedetailleerd overzicht van ophalen van wachtwoorden

Password Hook Informatie

Site: <http://sourceforge.net/projects/passwdhk/>

Bestanden: 32Bit en 64Bit DLL's

Configuratie: Moet enkele Register waarden saveen

Output: Username en wachtwoord in leerbare tekst kunnen naar EXE / Bat of andere programma met argumenten worden opgeroepen.



Om het geheel wat professioneel te laten overkomen, heb ik de installatie van Password Hook geïntegreerd in mijn programma.

Grafische voorstelling van Installatie Password Hook

Na de installatie wordt er gekeken of het systeem een domein is. Zo ja, dan wordt dit scherm getoond. Het zal alle registry instellingen aanmaken met de configuratie, en zal de passwordhook DLL installeren (32/64 bit).

Deze stap kan men overslaan indien men de LDAP functies niet wenst te gebruiken.

Op de volgende pagina vindt U een eenvoudige implementatie in C++ voor een wachtwoord filter.

Voorbeeld van C++ Password Filter (BASIC)

```
#include "stdafx.h"
#include <stdio.h>
#include <string>
#include <atlbase.h>

using namespace std;

// Implementatie met Password Filter
BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}

// Zodra LSA opstart (dus windows server opstart)
// word dit opgeroepen, return true om te laten weten
// dat alles is geïnitializeert
BOOLEAN __stdcall InitializeChangeNotify(void)
{
    return TRUE;
}

// Deze functie word opgeroepen als het wachtwoord
// is gewijzigd. Dit is dus NADAT andere filters er iets
// mee hebben gedaan. (Niet gebruiken)
NTSTATUS __stdcall PasswordChangeNotify(
    PUNICODE_STRING UserName,
    ULONG RelativeId,
    PUNICODE_STRING NewPassword
)
{
    return 0;
}

// Deze functie word opgeroepen Alvorens een gebruiker
// een wachtwoord wijzigd. Hiervan kunnen we alle data halen
BOOLEAN __stdcall PasswordFilter(
    PUNICODE_STRING AccountName,
    PUNICODE_STRING FullName,
    PUNICODE_STRING Password,
    BOOLEAN SetOperation
)
{
    return TRUE;
}
```


2.6.2 Invoegen in een database

Het invoegen van een wachtwoord in een database is redelijk simpel. Het wachtwoord wordt bij een eerstvolgende synchronisatie, uit de lokale database gehaald.

Vervolgens wordt met de "UPDATE" statement ingegeven in de juiste kolom en wordt de gebruikersnaam gebruikt als vaste referentie.

2.7 Module dbDriver

dbDriver is een module gemaakt voor WisaReg. Het bevat allerlei functies die toegang hebben tot databases, drivers en de volledige configuratie van WisaReg.

Omdat dbDriver geschreven is in '.NET' is het mogelijk om deze bibliotheek op te roepen vanaf eender welke '.NET' taal, en kan het dus als API gebruikt worden.












2.7.1 Documentatie API

De volledige documentatie van alle Namespaces en Methods kan gevonden worden bij het programma.

Constructors

	Name	Description
	dbDriver	Get data and initialize connection

Methods dbDriver


	Name	Description
	addCredentials	Add credentials settings to Wisa source
	addDatabase	Add a new database (source)
	addDriver	Add a driver in the config file
	addScheduled_COPY	Save Scheduler information
	addScheduled_UPDATE	Save Scheduler information
	addTable	Add a new table to the database configuration
	addTableColumns	Add columns to table setting (in local database)
	addTableWisa	Add a new table to the database configuration
	command	Perform a Query Language command
	deleteDriver	Delete a driver from the list
	deleteDriverName	Delete a driver from the list

	disconnect	Disconnect from the session
	editDatabase	Edit an existing database
	generateAlias	Generate the alias using the name
	getAD	Get the Active Directory name
	getAvailableSettings	Get the available settings as bool value
	getCreateSQL	Get the Create SQL command
	getCredentialsPassword	Gets the default password of the credentials
	getCredentialsToLower	Returns if the username has to be toLower
	getCredentialsUsername	Get the username of the credentials
	getDatabaseAlias	Get the database alias using file path
	getDatabaseAliases(String)	Get a list of all database aliases using path
	getDatabaseAliases(String[])	Get a list of all database aliases using file string
	getDatabaseAliasToName	Convert database alias to name
	getDatabaseFile	Get database ini file using path
	getDatabaseFiles	Get a list of all database files inside path
	getDatabaseNames(String)	Get a list of all database names using path
	getDatabaseNames(String[])	Get a list of all database names using file string
	getDatabaseNameToAlias	Convert database name to alias
	getDatabaseSettings	Get all settings to make a connection with a database
	getDriverAlias	Get the driver alias using driver name
	getDriverConStr	Get the connection string of the driver
	getDriverInstaller	Get The most compatible installer
	getDriverInstaller_32b	Get the Driver installation path for a 32bit system
	getDriverInstaller_64b	Get the Driver installation path for a 64bit system
	getDriverInstallPath	Get the installer file for the driver
	getDriverList	Get a list of all drivers
	getDriverName	Get driver name from driver alias




 getDriverNames	Get an array with all driver names
 getDropSQL	Get the Drop SQL command
 getEndYear	Get the start year
 getInsertSQL	Get the Insert SQL command
 getLDAP_domain	Get the LDAP server domain
 getLDAP_host	Get the LDAP server hostname/ip
 getLdapData	Get Ldap data
 getLdapFile	Get ldap ini file using path
 getLdapFiles	Get a list of all ldap config files inside path
 getNextScheduler_COPY	Get the next Scheduler date
 getNextScheduler_UPDATE	Get the next Scheduler date
 getSchedulerEnd_COPY	Get the scheduled date for end
 getSchedulerEnd_UPDATE	Get the scheduled date for end
 getSchedulerInterval_COPY	Get the interval of the scheduler
 getSchedulerInterval_UPDATE	Get the interval of the scheduler
 getSchedulerLastRun_COPY	Get the Timespan of the last Scheduled run
 getSchedulerLastRun_UPDATE	Get the Timespan of the last Scheduled run
 getSchedulerStart_COPY	Get the scheduled date for start
 getSchedulerStart_UPDATE	Get the scheduled date for start
 getSelectSQL	Get the Select SQL command
 getStartYear	Get the start year
 getSyncDbSource	Get the source of the Sync db (source from localdb)
 getTableCmd	Get an unknown command of the given table
 getTableColumns	Get an array with all the columns of localdb
 getTableCreateCmd	Get the Create command of the given table
 getTableData	Get an array with all data of the localdb table
 getTableDeleteCmd	Get the Delete command of the given table

	getTableDropCmd	Get the Drop command of the given table
	getTableInsertCmd	Get the Insert command of the given table
	getTableList(String)	Get a list of all tables
	getTableList(String, String)	Get a list of all tables
	getTableNames	Get the table names of the locadb
	getTableSelectCmd	Get the Select command of the given table
	getTableSource	Get the source of the variables used in the table
	getTableTruncateCmd	Get the Truncate command of the given table
	getTableUpdateCmd	Get the Update command of the given table
	getTableVariables	Get a list of all variables used in a table
	getTruncateSQL	Get the Truncate SQL command
	getUpdateSQL	Get the UPDATE SQL command
	setLdapData	Set Ldap data
	setSchedulerRun_COPY	Set the last run to the current moment
	setSchedulerRun_UPDATE	Set the last run to the current moment

Constructors

	Name	Description
	dbDriver.LdapDriver	Initializes a new instance of the dbDriver.LdapDriver class

Methods *dbDriver.LdapDriver*

	Name	Description
	createAccount	Creat an AD User account
	objExists	Check if an object exists
	userExists	Check if a user exists

2.8 Module dbSync

dbSync is een module gemaakt voor WisaReg die alle synchronisaties en updates uitvoert van databases. Net zoals dbDriver is deze bibliotheek geschreven in '.NET 4.0'.














Oorspronkelijk was het de bedoeling dat de database connecties 'gecacht' werden vooraleer er grote iteraties gebeurde. Deze manier werkte goed, en was zeer snel. Maar omdat het niet flexibel genoeg was (alle groepen op hetzelfde moment) heb ik het cachen ertussenuit gelaten.














2.8.1 Documentatie API

Constructors




	Name	Description
	dbSync	Initializes a new instance of the dbSync class

Methods

	Name	Description
	addLDAPAccount	
	copyDatabases(String, String, String, String)	Copy databases from local to syncdatabase
	copyDatabases(String, String, String, String, String)	Copy databases from local to syncdatabase
	copyWisa	Copy all data from Wisa Select to the local database
	copyWisa_ALL	Copy all data from Wisa Select to the local database
	createLocal	Create all tables in the local database
	createLocal_ALL	Create all tables in the local database for all sources
	deleteLocal	Delete data in the local database
	deleteLocal_ALL	Delete data in the local database
	dropLocal	Drop data in the local database
	dropLocal_ALL	Drop data in the local database
	dropP_Local(String, String)	Drop persistent data in the local database
	dropP_Local(String, String, String)	Drop persistent data in the local database
	getCount	Get student count

	getUserPassword	Get the password of the given user
	replaceVariables	Replace variables from a SQL string
	setUserPassword	Set the MD5 password of the user account
	syncPasswords	Synchronize passwords
	truncateLocal	Empty the local database
	truncateLocal_ALL	Empty the local database
	updateDatabase	Update all data from Wisa Select to the local database
	updateDatabase_ALL	Update all data from Wisa Select to the local database
	updateLocal	Update all data from Wisa Select to the local database
	updateLocal_ALL	Update all data from Wisa Select to the local database
	usernameFilter	Filter all variables from username string
	usernameFilter_Naam	Filter all usernames with PS_VOORNAAM and PS_NAAMSORT
	usernameFilter_Voornaam	Filter all usernames with LL_VOORNAAM and INGESCHREVEN_IN_KLAS

Events

	Name	Description
	copyWisa_increment	
	error_dbsync	
	syncDb_increment	

2.8.2 Code Kopie van Wisa naar Lokaal

De kopie van Wisa naar de Lokale database was een grote stap, aangezien er snel en foutloos 1300 leerlingen Local database moesten opgevraagd en ingevoegd worden.

Dankzij het goed verdelen van de functies tussen dbDriver en dbSync, is het overzichtelijk gemakkelijk aanpasbaar.

```
public static void copyWisa(String localdbPath, String wisadbPath, String driverPath, String item_to)
{
    // Check if the paths are valid
    if (Directory.Exists(localdbPath) && Directory.Exists(wisadbPath))
    {
        // The directory local and wisa exist
        // Copy Wisa(table.xxx) to Local(table.xxx)
        String[] data = dbDriver.getDatabaseAliases(wisadbPath);
        foreach (String item in data)
        {
            String[] Settings = dbDriver.getDatabaseSettings(item, wisadbPath);
            // Get the command to select data from wisa
            String[] variables = dbDriver.getTableVariables(item, wisadbPath, "select"); // Default "Select" is select sql
            String cmd = dbDriver.getTableSelectCmd(item, wisadbPath, "select"); // Default "Select" is select sql
            String constr = dbDriver.getDriverConStr(Settings[4], driverPath, Settings[0], Settings[1], Settings[2], Settings[3]);
            String sourceID_wisa = item; // Source must match localdb Source
            dbDriver db = new dbDriver(constr);
            // Replace Begin and end year
            cmd = cmd.Replace("@@SCHOOL_A", dbDriver.getStartYear().ToString());
            cmd = cmd.Replace("@@SCHOOL_B", dbDriver.getEndYear().ToString());
            // Execute create command
            DataSet data_wisa = db.command(cmd); // Remove newline characters
            // Now get all variables and save them into an array

            // Get the data from each item
            String[] Settings_to = dbDriver.getDatabaseSettings(item_to, localdbPath);
            // Get the connection string, and create a connect
            String constr_to = dbDriver.getDriverConStr(Settings_to[4], driverPath, Settings_to[0], Settings_to[1], Settings_to[2],
            Settings_to[3]);

            // Initialize all database connection before they get in the loop
            dbDriver dbconnection_local = new dbDriver(constr_to);

            // Each table (name: name1 ,age:... name: name2,age:...)
            // NOTE: This loop will be run x times the students count
            // Place time intensive actions above this line
            foreach (DataTable tb in data_wisa.Tables)
            {
                String[,] data_wisa_vars = new String[variables.Length, tb.Rows.Count];
                DataColumn dc = tb.Columns[0];
                // Each row (name1, 21,...)
                // Each variable into the list
                for (int i = 0; i < variables.Length - 1; i++)
                {
                    for (int j = 0; j < tb.Rows.Count; j++)
                    {
                        DataRow row = tb.Rows[j];
                        data_wisa_vars[i, j] = row[variables[i].Replace("%", "")].ToString(); // Add every data var
                    }
                }
                // First check if item_to matches sourceID
                if (item_to == sourceID_wisa)
                {
                    // Define Variables before we get into a loop of +- 1300
                    String cmd_to = "";
                    String cmd_insert = "";
                    String username = "";
                    String password = "";
                    String cmd_p = "";
                    // Get the table name
                    String tableName = dbDriver.getTableName(item_to, localdbPath);

                    // Get a List of all usernames + passwords
                    List<String> usernames = new List<String>();
                    List<String> passwords = new List<String>();
                    try
                    {
                        // Get all users in the persistent database (if any)
                        cmd_p = "SELECT * FROM `p_" + tableName + "`";
                        DataSet output = dbconnection_local.command(cmd_p);
                        for (int r = 0; r < output.Tables[0].Rows.Count; r++)
                        {
                            // Save Information
                            usernames.Add(output.Tables[0].Rows[r].ItemArray.GetValue(0).ToString());
                            passwords.Add(output.Tables[0].Rows[r].ItemArray.GetValue(1).ToString());
                        }
                    }
                    catch (Exception)
                }
            }
        }
    }
}
```

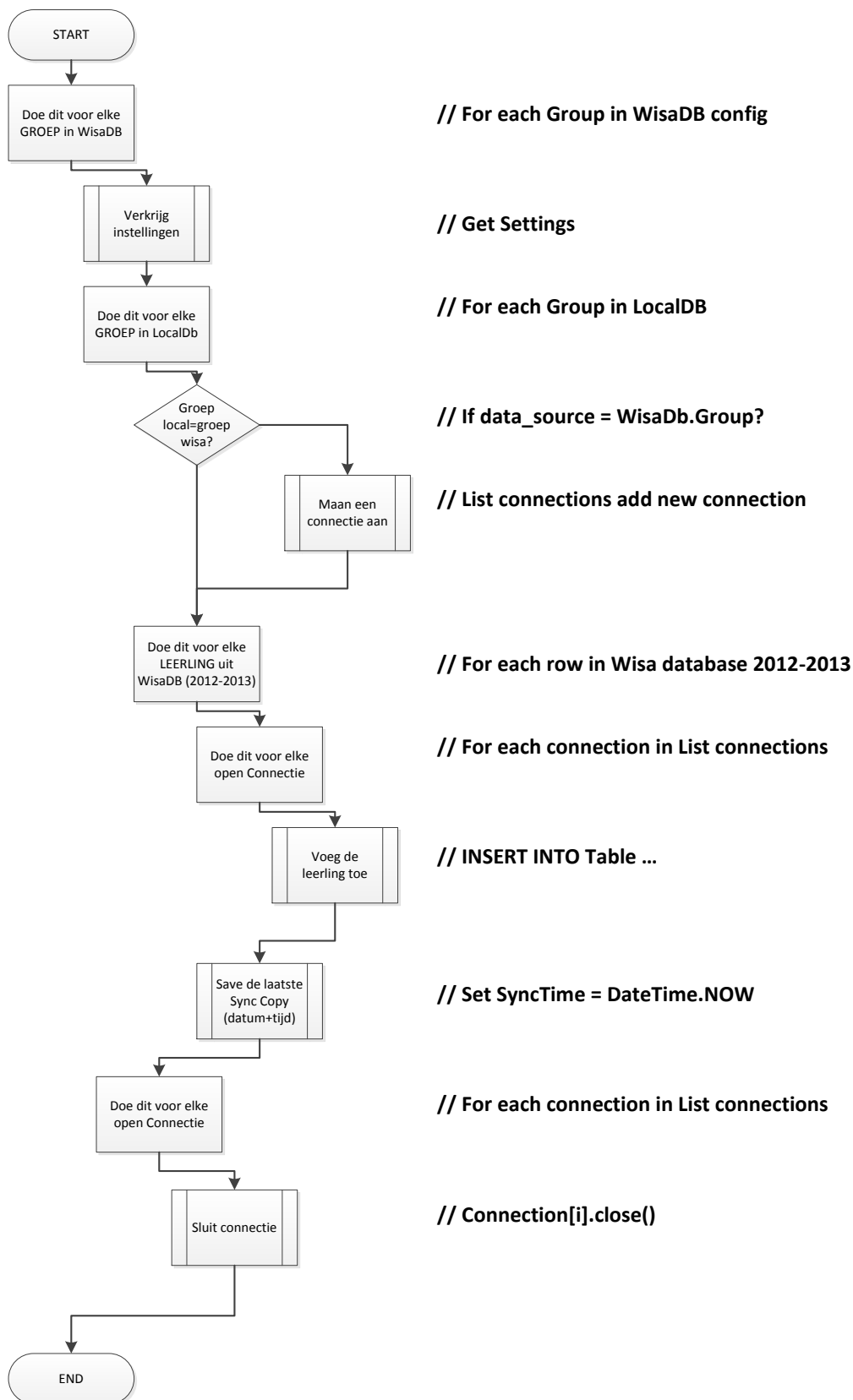
```
{
    // Some information could not be saved
}

// Use a special filter
List<String> usernameFiltered = new List<String>();
if (dbDriver.getCredentialsUsername(wisadbPath, item).Contains("%VOORNAAM.KLAS%"))
{
    usernameFiltered = usernameFilter_Voornaam(data_wisa_vars,
        dbDriver.getCredentialsUsername(wisadbPath, item), variables);
}
else if (dbDriver.getCredentialsUsername(wisadbPath, item).Contains("%NAAM.ICT%"))
{
    usernameFiltered = usernameFilter_Naam(data_wisa_vars,
        dbDriver.getCredentialsUsername(wisadbPath, item), variables);
}

// Each user
for (int j = 0; j < data_wisa_vars.GetUpperBound(1) + 1; j++)
{
    try
    {
        // Trigger event
        copyWisa_increment(null, null);
        cmd_to = dbDriver.getTableInsertCmd(item_to, localdbPath, "insert"); // Default "Select" is select sql
        cmd_insert = replaceVariables(data_wisa_vars, cmd_to, variables, j);
        // Replace Begin and end year
        cmd = cmd.Replace("@@SCHOOL_A", dbDriver.getStartYear().ToString());
        cmd = cmd.Replace("@@SCHOOL_B", dbDriver.getEndYear().ToString());
        // Execute insert command
        dbconnection_local.command(cmd_insert.Replace("\n", "").Replace("\r", "")); // Remove newline characters
        // Create password and username
        // Check for a special NAAM/VOORNAAM/KLAS filter
        if (dbDriver.getCredentialsUsername(wisadbPath, item).Contains("%VOORNAAM.KLAS%"))
        {
            // Special filter made for Don Bosco Hoboken
            username = usernameFiltered[j];
        }
        else if (dbDriver.getCredentialsUsername(wisadbPath, item).Contains("%NAAM.ICT%"))
        {
            // Special filter made for Don Bosco Hoboken
            username = usernameFiltered[j];
        }
        else
        {
            // Normal filtering
            username = usernameFilter(data_wisa_vars,
                dbDriver.getCredentialsUsername(wisadbPath, item), variables, j);
        }
        if (dbDriver.getCredentialsToLower(wisadbPath, item) == true)
        {
            username = username.ToLower();
        }
        MD5 md5Hash = MD5.Create();
        password = GetMd5Hash(md5Hash, dbDriver.getCredentialsPassword(wisadbPath, item));

        // Get the username, if found
        if (!usernames.Contains(username))
        {
            // The user has not been added yet.
            cmd_p = "INSERT INTO `p_" + tableName + "` SET `wisareg_username`='" + username + "',
`wisareg_password`='" + password + "'";
            // Try to save the password
            // If this fails, the passwords won't sync
            try
            {
                // Run The command for the persistent passwords
                dbconnection_local.command(cmd_p);
            }
            catch (Exception)
            {
            }
        }
    }
    catch (Exception ex)
    {
        // Error
        error_dbsync((object)ex.Message, null);
    }
}
// Set Last run
dbDriver.setSchedulerRun_COPY(localdbPath, item_to);
}
// Close session(s)
try
{
    dbconnection_local.disconnect();
}
catch { }
}
}
```


2.8.3 Schematische voorstelling koppie Wisa naar Lokaal



2.9 Module config_ini

Config_Ini is een module die gebruikt wordt in zowat elke module, en elk programma van WisaReg. Het is een handige verzameling van functies om keys en values te wijzigen in een INI-file.

Ini-files worden gebruikt om de instellingen te save. En heeft de volgende opmaak:


```
[Section]  
Key=value
```

```
[Secution2]  
Key=value  
Key2=value
```





Men kan het opdelen in verschillende stukken, die elke hun eigen data bevatten.

2.9.1 Documentatie API

Constructors

	Name	Description
	iniConfig	Get filename and open file

Methods


	Name	Description
	DeleteKey	Delete the key from the ini file
	DeleteSection	Delete the section from the ini file
	ReadValue	Read data from the ini file
	WriteValue	Write data to the ini file

2.10 Module reg_edit







De registry editor module is een dll die benodigd is bij het installeren van password hook. Het slaagt alle instellingen die de normale installer van Password Hook zou doen op in het register.

2.10.1 Documentatie API



Constructors

	Name	Description
	regEdit	Initializes a new instance of the regEdit class

Methods

	Name	Description
	DeleteKey	Delete a key
	DeleteSubKeyTree	Delete a whole subkey tree
	Read	Read a key from the registry
	SubKeyCount	Count all keys
	ValueCount	Count all values
	Write	Write a value to a key

Fields

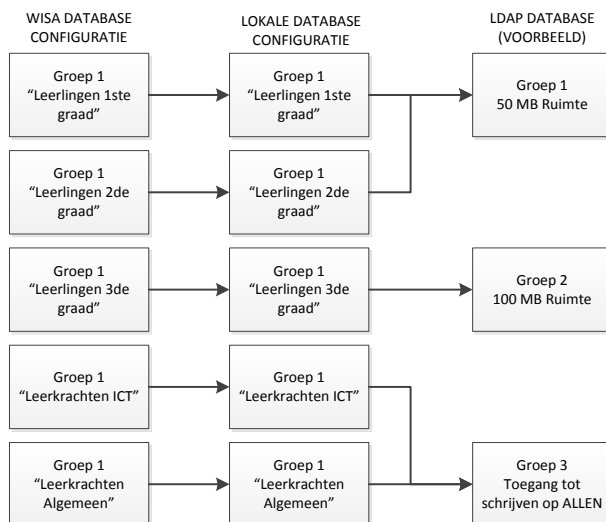
	Name	Description
	baseRegistryKey	
	subKey	

3. Gebruikersinterface

Een programma dat doet wat het moet doen, is niets waard als de gebruiker het niet kan bedienen. Zelf vond ik het cruciaal dat de gebruiker een duidelijke, gemakkelijke en snelle gebruikersinterface had.

Dit gaat niet enkel over het bedienen van het programma, maar ook op de manier hoe men instellingen moet maken. Meestal doe je dit maar één enkele keer, maar dan moet het ook juist zijn.

3.1 Configuratie



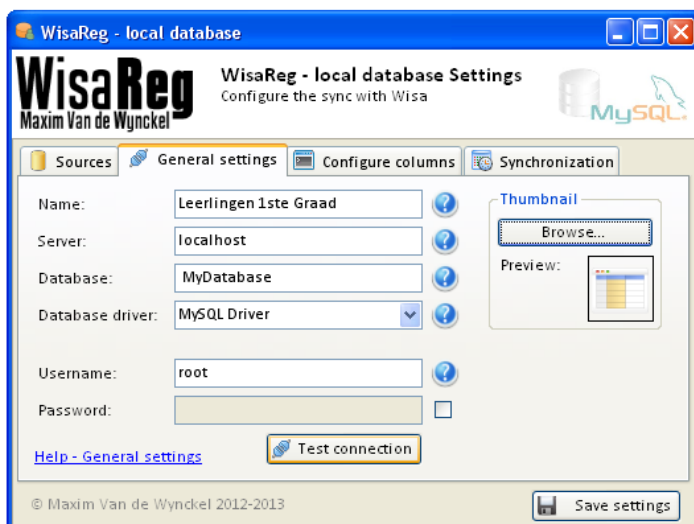
De Configuratie is zo gemaakt, dat bij elk onderdeel het mogelijk is om een groep te maken.

Dit laat de gebruiker toe, om zelf toestemmingen te verzinnen zonder moeilijke selecties te maken.

Men start met alle groepen die ERGENS in de instellingen een aparte toestemming moeten hebben.

Het nadeel hiervan is dat per groep steeds een aparte groep bijkomt.

Zo kan je bijvoorbeeld alle leerkrachten (Zowel ICT, als algemeen) naar de database volgsysteem laten gaan en de leerlingen niet toevoegen als groep. Of alle Leerlingen (1,2 en 3^{de} graad) in het Virtueel Leerplatform zetten met dezelfde toestemmingen, die toch individueel gemakkelijk kunnen worden aangepast.



Basisinstellingen

Elke configuratie met een database zal dit tabblad hebben.

Het zijn de basisinstellingen voor de verbinding met de server/database.

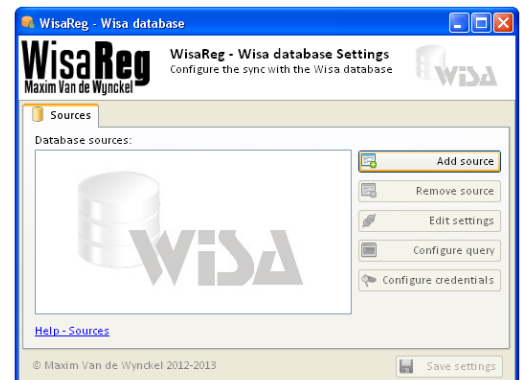
Beweeg je muis over het vraagteken voor meer informatie.

3.1.1 Wisa database configuratie

De Wisa Database configuratie GUI maakt zoals alle configuratieschermen gebruik van het tabblad interface.

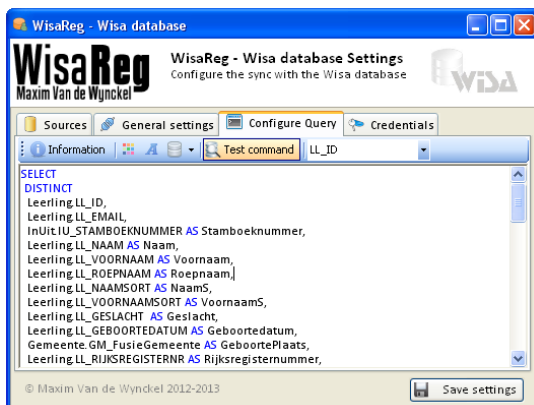
Deze laat de gebruiker toe om gemakkelijk en overzichtelijk een groep toe te voegen.

De groepen die in deze configuratie worden gedefinieerd, zijn bepalend voor de toestemmingen en rechten op alle databases.



Voor het geval van 'Don Bosco Hoboken' moeten er waarschijnlijk volgende groepen bestaan:

- Leerlingen 1^{ste} graad
- Leerlingen 2^{de} graad
- Leerlingen 3^{de} graad
- Leerkrachten – Algemene vakken
- Leerkrachten ICT



Deze moeten opgehaald worden uit Wisa, d.m.v. SQL code die in het tabblad **Configure Query** moet ingegeven worden.

Figuur: SELECT query voor leerlingen eerste graad
De query haalt info over alle leerlingen, maar heeft als extra vereiste, dat de leerling in graad 1 zit.

Deze is moeilijk samen te stellen, maar moet normaal 1 keer gebeuren, en wordt door Wisa zelf gemaakt.

Overige instellingen voor de database zijn:

- Server instellingen voor connectie
- De gebruikersnaam en default wachtwoord dat elke persoon krijgt

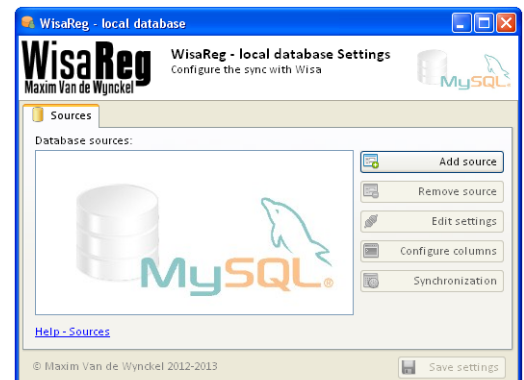
3.1.2 Lokale database configuratie

De Lokale Database configuratie GUI maakt zoals alle configuratie schermen gebruik van het tabblad interface.

Deze laat de gebruiker toe om gemakkelijk en overzichtelijk een groep toe te voegen.

De groepen die in deze configuratie worden gedefinieerd, moeten best gelijk zijn aan die van uit de Wisa database.

Het toevoegen van tabellen en kolommen in de lokale database gaat makkelijker dan in de Wisa database, de variabelen en groepen uit de Wisa configuratie worden gebruikt om automatisch SQL commando's te maken op basis van instellingen.



De tabel die in de lokale database komt

Voorbeeld van al toegevoegde kolommen

Column	Data

De naam van de kolom voor de lokale database

De groep van de Wisa database (voor variabelen)

De variabele die in de kolom zal worden ingegeven per (leerling)

Ook moet de synchronisatie ingesteld worden tussen de Wisa database en de lokale database.

Men kan er vanuit gaan dat dit niet zo veel veranderingen gaan zijn. Enkel informatie kan veranderen, of er kunnen gebruikers bijkomen of weg gaan.

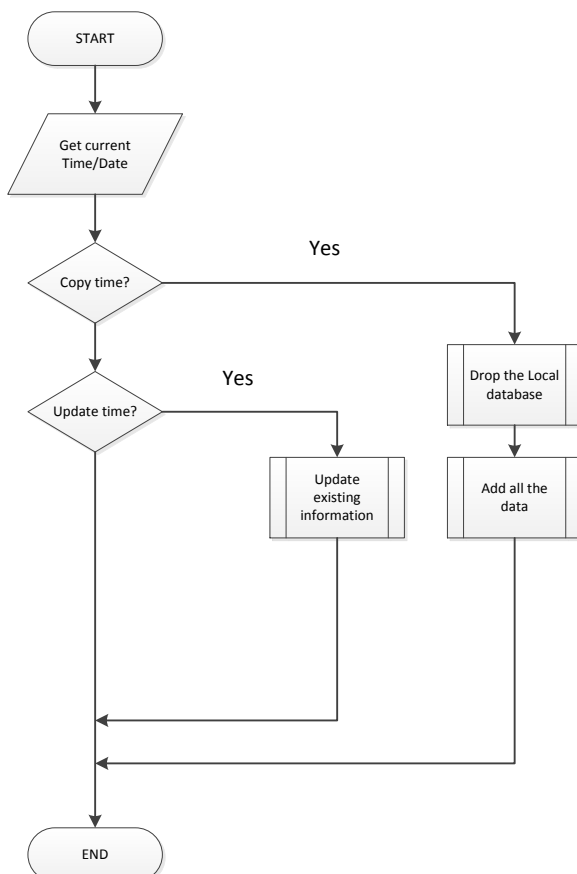


Op het tabblad, kan men een onderscheid zien tussen Update en Copy dit zijn twee

- Update: bestaande gegevens worden vernieuwd (informatie van leerlingen/leerkrachten)
- Copy: nieuwe leerlingen of verwijderde leerlingen worden toegepast.

Een volledig kopie van Wisa naar Lokaal (enkel leerlingen van dat jaar) duurt ongeveer 1 minuut. Maar dit moet slechts om de dag of week gedaan worden.

Aangezien de kans dat een nieuwe leerling de eerste dat al direct op de computer moet, klein is.



// Start het programma

// Verkrijg de huidige datum/tijd

// Kijk of het tijd is om te kopiëren?, Indien niet kijk dan of het tijd is om te updaten

Dit proces wordt uitgevoerd door het main programma 'WisaReg'. Deze schematische voorstelling geeft aan hoe het update en kopieer proces loopt voor de lokale database.

3.1.3 Sync. database configuratie

De Synchronisatie databaseconfiguratie is niet veel verschillend dan de lokale databaseconfiguratie. Het enige verschil is, dat er de mogelijkheid is om de kolommen uit de reeds bestaande database te halen. Om het zo gemakkelijker te maken voor de gebruiker.

Ook is de manier dat data in kolommen worden gesaved anders. Nu kan men statische gegevens toevoegen, of statisch en een variabel in elkaar:

Vb.

Kolom	Data
firstName	%VOORNAAM%
picture_url	http://volgsys.dbh-admi.be/foto/%RIJKSREGISTERNUMMER%.jpg
isAdmin	0

De reden waarom dit niet wordt toegepast in de lokale configuratie, is omdat ik eigenlijk wil dat de lokale database een stuk is van de Firebird database van Wisa.

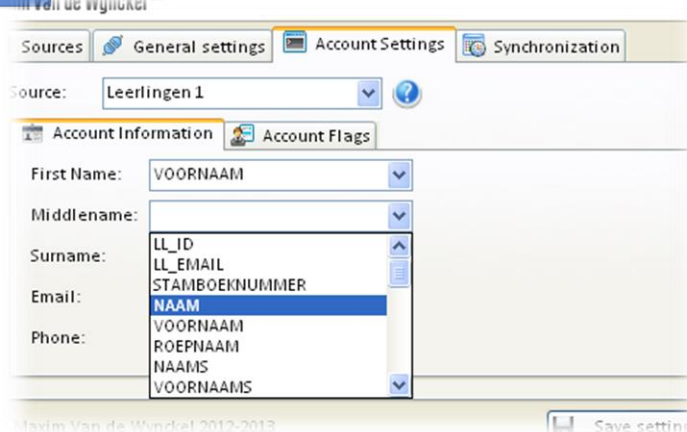
3.1.3 LDAP configuratie



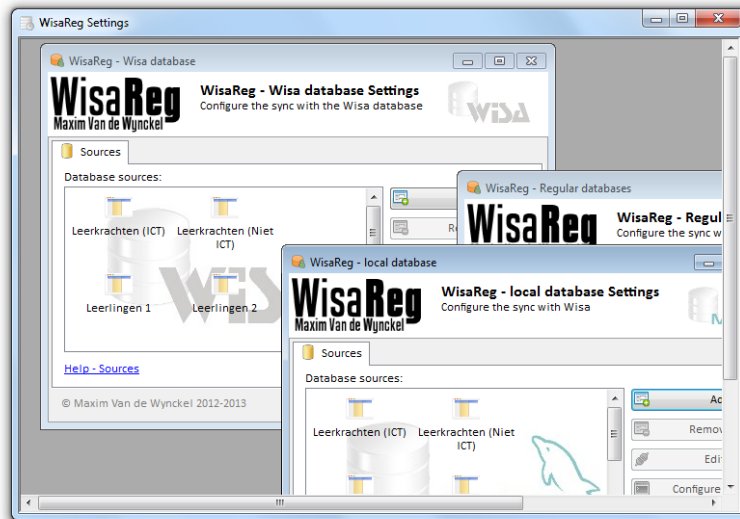
De configuratie van LDAP verloopt met enkele klikken, het enigste wat moet ingegeven worden is de domein naam, groep en administrator gebruikersnaam en wachtwoord.

De accountinformatie gaat door de kolom van de lokale database te selecteren in een lijst.

Eerst zal men de 'Source' selecteren (De gegevens uit de lokale database)
Vervolgens kan men de juiste kolom selecteren bij elk gespecificeerd gegeven.



3.1.4 WisaReg Settings paneel



In het settings paneel van het hoofdapplicatie, worden alle schermjes van de settings in één venster getoond.

Op deze manier kan de gebruiker alle instellingen met elkaar vergelijken, om ze de groepen te optimaliseren.

De methode die ik hiervoor gebruik is het laden van de forms uit de executable, en vervolgens getoond in één venster.

Eigenlijk was het niet de bedoeling om het op deze manier te doen, maar om wat tijd te besparen, heb ik voor deze methode om alle configuraties samen te voegen, geopteerd.

Code voor toevoegen van externe form als MDI

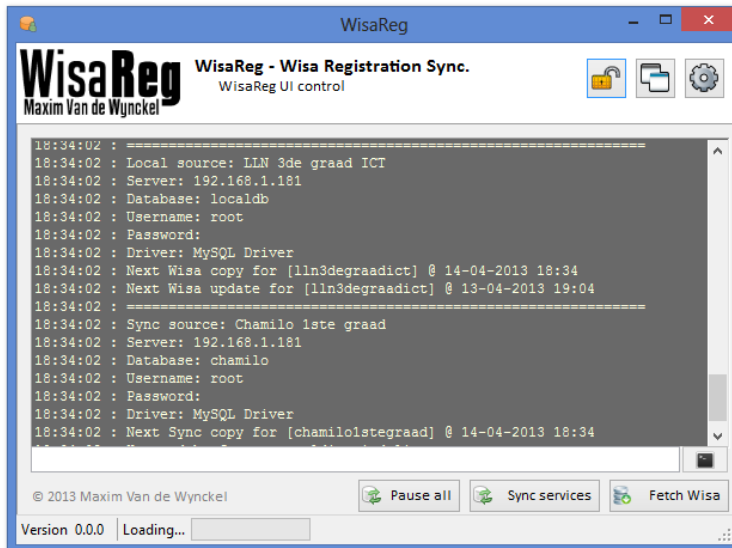
```
using System.Reflection; // Voor 'Assembly' te laden

// Load WisaDB settings
try
{
    Assembly asm = Assembly.LoadFrom(Application.StartupPath +
@"\WisaReg_settings_wisadb.exe");
    object objClass = asm.CreateInstance("WisaReg_settings.frm_main");
    Form frm = (Form)objClass;
    frm.MdiParent = this;
    frm.Show();
}
catch (Exception)
{
}
```

```
From(Application.StartupPath +
xe");
Instance("WisaReg_settings.frm_main");
```

WisaReg_settings : Namespace
Frm_main : Begin form die gestart word

4. Hoofdprogramma



Het hoofdprogramma van WisaReg bevat enkele thread's die wachten op wachtende synchronisaties of eventuele gebruikersinvoer. Men kan manueel een synchronisatie uitvoeren via een knop (alle groepen) of via een bepaald commando (specifiek).

Merk op dat het scherm vergrendeld kan worden zodat bij een inbreuk op de server, deze niets kan aanpassen.



De drie knoppen bovenaan dienen respectievelijk voor:

- Het systeem te ver(ont)grendelen
- Minimaliseren naar Tray menu
- Instellingen openen

4.1 Beveiliging

De beveiliging van WisaReg gebeurt op Windows niveau. Ik ga er van uit dat WisaReg draait op een domain controller, aangezien dit noodzakelijk is voor LDAP synchronisaties. Ik ga er ook van uit dat iemand die als administrator kan inloggen op de server, toegang mag hebben tot de files en applicaties van WisaReg.

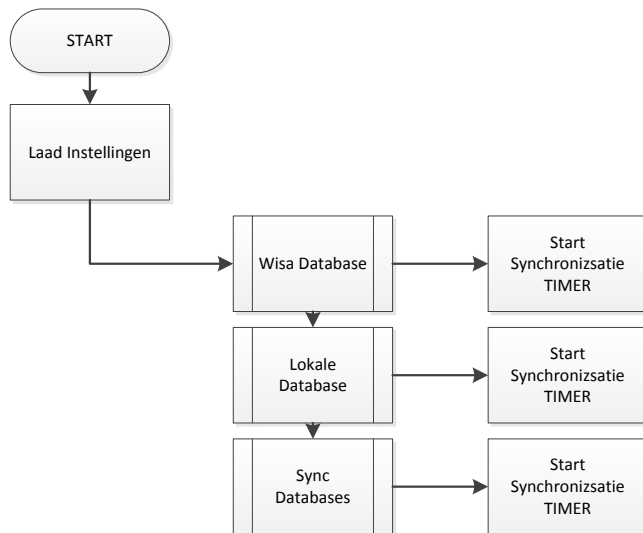
Daarom gebruikt WisaReg de logingegevens van eender welke gebruiker die zich lokaal kan aanmelden op de server.

Zolang de gebruiker zich niet heeft aangemeld, kan men nog wel het LOG lezen, maar geen acties uitvoeren.



4.2 Programmaverloop

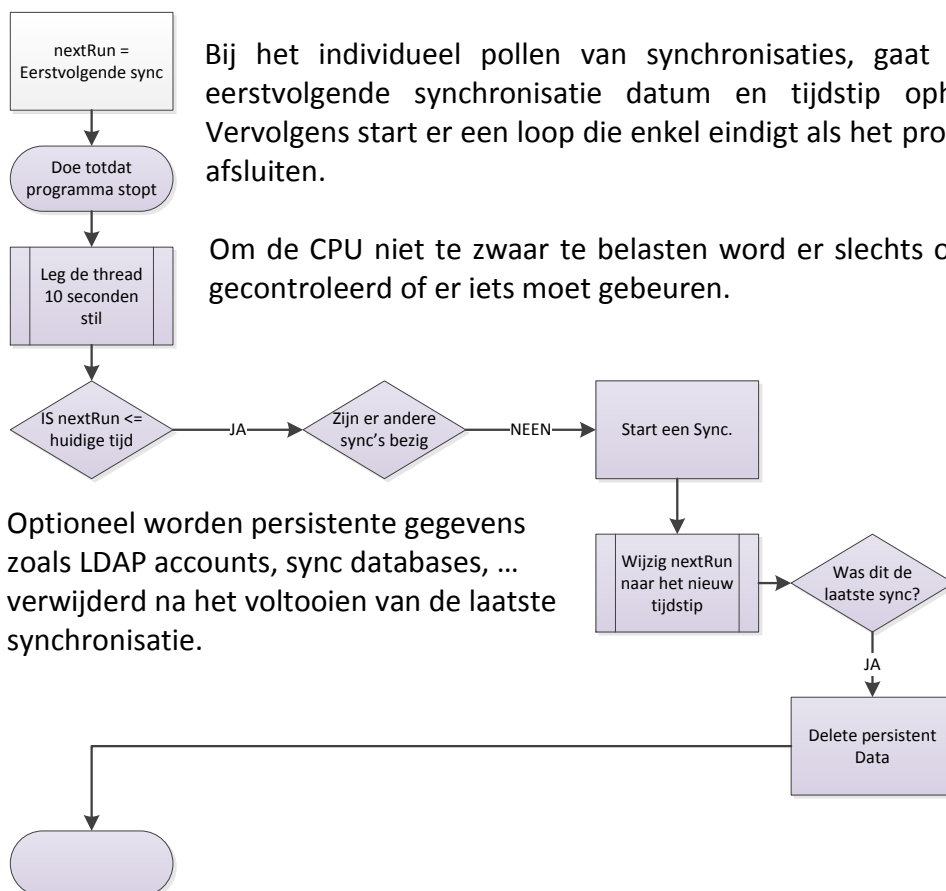
4.2.1 Opstart verloop



Bij het opstarten van het programma zullen eerst alle instellingen worden geladen van elke database, en groep. Vervolgens wordt er voor elke groep per database, een thread gestart die wacht totdat het tijd is om een synchronisatie uit te voeren.

Op deze manier ben je er steeds zeker van dat synchronisaties elkaar niet overlappen, omdat het alternaties (geplande taken) redelijk moeilijk is bij meerdere taken op het zelfde moment.

4.2.2 Polling verloopt



Bij het individueel pollen van synchronisaties, gaat het programma de eerstvolgende synchronisatie datum en tijdstip ophalen via dbDriver. Vervolgens start er een loop die enkel eindigt als het programma bezig is met afsluiten.

Om de CPU niet te zwaar te belasten word er slechts om de tien seconden gecontroleerd of er iets moet gebeuren.

Optioneel worden persistente gegevens zoals LDAP accounts, sync databases, ... verwijderd na het voltooien van de laatste synchronisatie.

4.3 Commando's

Er bestaat de mogelijkheid om manueel synchronisaties uit te voeren. Daarom kan men via het veld onder het Log tekst invoeren.

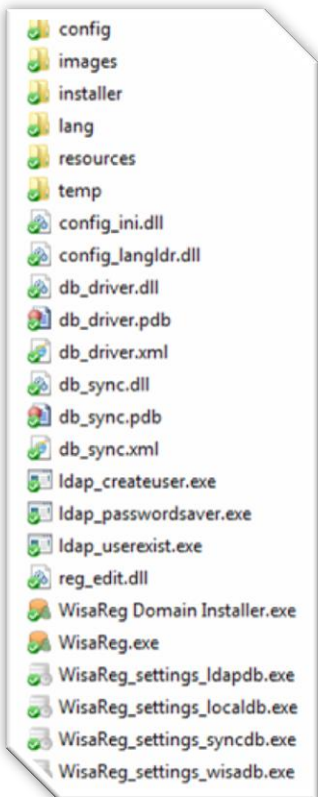
Hieronder vind U een lijst met alle mogelijke commando's en hun argumenten.

Commando	Argument	Uitleg
getsynccopy	Source	<i>Geeft als output de datum/tijd van de volgende koppie voor de Service databases</i>
getsyncupdate	Source	<i>Geeft als output de datum/tijd van de volgende update voor de Service databases</i>
getwisacopy	Source	<i>Geeft als output de datum/tijd van de volgende koppie voor de Lokale database</i>
getwisaupdate	Source	<i>Geeft als output de datum/tijd van de volgende update voor de Lokale database</i>
synccopy	Source	<i>Manuele synchronisatie koppie voor de Service databases</i>
syncupdate	Source	<i>Manuele synchronisatie update voor de Service databases</i>
wisacopy	Source	<i>Manuele synchronisatie koppie voor de Lokale database</i>
wisaupdate	Source	<i>Manuele synchronisatie update voor de Lokale database</i>
resetpass	Source	<i>Verwijder de persistente gegevens van de gebruikers in een groep</i>
resetldap	Source	<i>Verwijder de account uit de Active Directory</i>
resetsync	Source	<i>Verwijder de accounts uit een Service database</i>

Ideeën die mogelijk zijn om bij te maken (Allemaal makkelijk toe te voegen via de API):

- Voeg individuele accounts toe naar een Service-of LDAP
- Delete individuele accounts uit een bepaalde database
- Synchroniseer individuele accounts

4.4 Root directory structuur



config
images
installer
lang
resources
temp
config_ini.dll
config_langldr.dll
db_driver.dll
db_driver.pdb
db_driver.xml
db_sync.dll
db_sync.pdb
db_sync.xml
ldap_createuser.exe
ldap_passwordsaver.exe
ldap_userexist.exe
reg_edit.dll
WisaReg Domain Installer.exe
WisaReg.exe
WisaReg_settings_ldapdb.exe
WisaReg_settings_localdb.exe
WisaReg_settings_syncdb.exe
WisaReg_settings_wisadb.exe

Map structuur voor Configuratie
De map config bevat de configuratie bestanden.
De images map bevat thumbnail 's van de configuratie schermen.
De resources map bevat bestanden die dienen te worden geïnstalleerd.

Modules en documentatie
Zowel PDB (debug files) als de XML (Code docs) is toegankelijk
Deze DLL's zijn cruciaal om eender welke applicatie te starten. En dienen niet te worden verplaatst of gedeletet.

LDAP Tools voor het maken van users en wachtwoorden

Hoofd applicaties (WisaReg en domain installer)
WisaReg.exe is het hoofdprogramma

Configuratie met gebruikersinterface
Deze assembly files worden geladen door het hoofdprogramma in één scherm.

4.4.1 WisaReg benodigde files

WisaReg_settings_ldapdb.exe – Configuratie

WisaReg_settings_localdb.exe – Configuratie

WisaReg_settings_syncdb.exe – Configuratie

WisaReg_settings_wisadb.exe – Configuratie

Db_driver.dll – Opstarten

Db_sync.dll – Opstarten

Config_ini.dll – Opstarten

Config_langldr.dll - Opstarten (Niet geïmplementeerd)

5. Commercialisering programma

Het commercialiseren van het programma is ook een opgave, waar je van dag één met moet rekening houden. Voor dit programma is het bijna niet realistisch dat het op grote schaal verkocht/gepubliceerd wordt. Maar het was wel een goede oefening.

5.1 Naam en logo

Het begon met het vinden van een naam, ik koos voor **WisaReg** omdat het allemaal draait om de database die **Wisa** maakt, en het programma zorgt dat alle **geregistreerde** leerlingen accounts krijgen.

Vervolgens moest ik een icon vinden voor WisaReg, als klein logo koos ik een simpel icoontje van een Database en een mannetje (dat een leerling/leerkracht voorstelt)



Het bevat niet al te veel details, zodat het als 16x16 pixel icoon kan worden getoond in een programma.

Als logo koos ik voor de tekst **WisaReg** met daaronder mijn naam ingewerkt. Het logo moest duidelijk zijn en simpel.

Origineel Logo (183x80)



Geoptimaliseerd Logo (718x311)



5.2 Commercieel bruikbare afbeeldingen

Aangezien ik veel icoontjes, en afbeeldingen gebruik in mijn programma bij knoppen, tab pagina's, of de documentatie. Heb ik de site <http://www.iconarchive.com> gebruikt.

Hier kan men zien of bepaalde iconen bruikbaar zijn voor commerciële doeleinden.

Het controleren of iconen of afbeeldingen bruikbaar zijn voor commerciële doeleinden is zeer belangrijk, aangezien de afbeeldingen auteursrechtelijk zijn aan de eigenaar.

Elke afbeelding of icoon die in WisaReg wordt gebruikt is bruikbaar voor de doeleinden waar het programma is gemaakt.

Artist: Fast Icon (Available for custom work)
Iconset: Database Icons (26 icons)
License: Linkware (Backlink to <http://www.fasticon.com> required)
Commercial usage: Allowed
License URL: <http://www.fasticon.com/commercial-license/>

[Download PNG](#)[Download ICO](#)[Download ICNS](#)

Download other sizes of this icon:



5.3 Licentie voor WisaReg

WisaReg valt onder het Creative Commons Naamsvermelding, Niet commercieel ,Geen Afgeleide werken.



Dit houdt in dat mijn programma niet mag worden aangepast zonder mijn toestemming, en dat het niet mag worden gebruikt voor commerciële doeleinden.

5.4 Compatibiliteit

WisaReg is getest op onderstaande besturingssystemen:

- *Windows XP : Geen Password Hook (LDAP)*
- *Windows 7 : Geen Password Hook (LDAP)*
- *Windows 8 : Geen Password Hook (LDAP) + Betere GUI integratie*
- *Windows Server 2003 : Gui niet optimaal, Calibri font onbreekt*
- *Windows Server 2008 : Werkt volledig*
- *Windows Server 2012 : Werkt normaal volledig (Password Hook nog niet officieel)*

De reden waarom het op zoveel systemen is kunnen worden getest, is omdat ik programmeer zowel thuis (XP en Win8), op school (XP en Win7) en omdat ik begon in WinS2003 en naar 2008 en 2012 ben overgestapt omdat ik vooruit wilde denken.

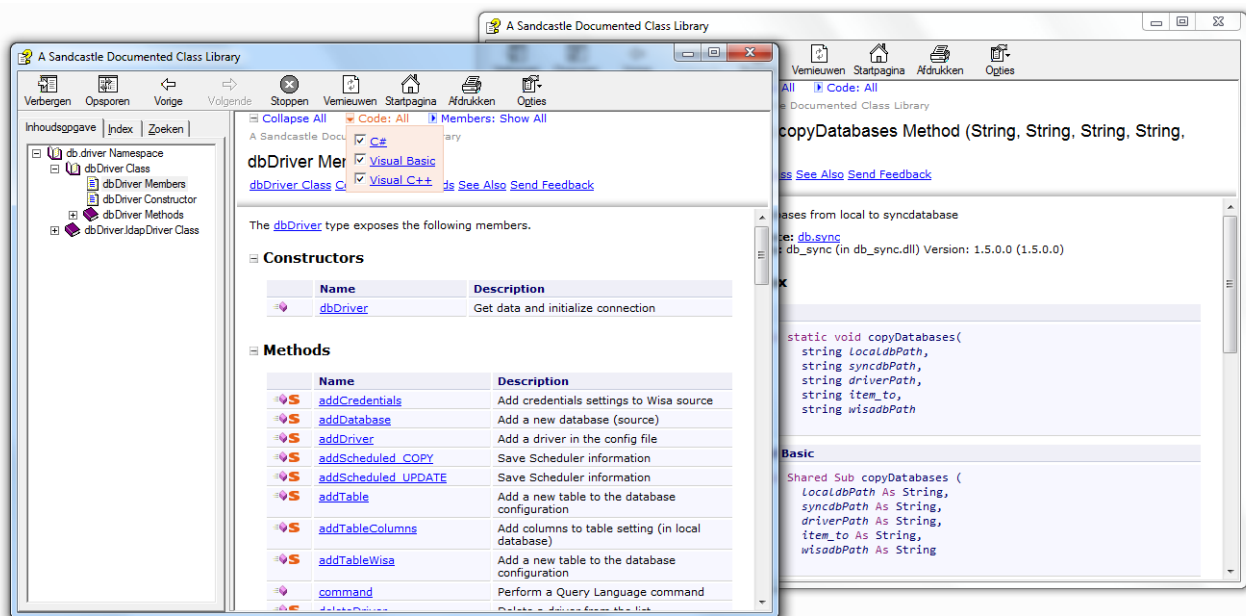
5.5 Application Interface

Zoals reeds vermeld in hoofdstuk 2, bestaat het programma uit modules.

Bij het maken van deze modules is er vanaf de eerste dag al rekening mee gehouden om alle informatie over elke routine duidelijk op te schrijven in de documentatie van de code.

Dankzij het duidelijk maken van de routines heb ik dus eigenlijk een API gemaakt waarmee WisaReg kan worden uitgebreid. Ook de LDAP tools (Zie mapstructuur pg. 34) zorgen ervoor dat met enkele argumenten naar een applicatie, verschillende functies kunnen worden gebruikt.

De documentatie van de API's zijn geconverteerd naar een 'Documentation Help file (.chm)' met het programma **Sandbox**. Hier kan men zien welke routine welke variabelen nodig heeft, wat het als uitvoer geeft, en nog veel meer informatie.



De API is toegankelijk in de .NET omgeving maar het kan via een makkelijk te maken interface naar JAVA of eventueel PHP geport worden.

De mogelijkheden met de API zijn redelijk groot, aangezien de volledige configuratie hiermee kan worden aangepast.

Voorbeeld:

De UPS van een server meldt dat de server zal moeten worden uitgezet indien de stroom niet tijdig terug aan is. Het programma dat dit merkt zal met het gebruik van de module db_driver.dll de routine `setSchedulerRun_COPY` uitvoeren zodat WisaReg denkt dat de sync. al gebeurd is, en niet direct zal starten met synchroniseren.

5.6 Smartschool integreren

Toen mijn GIP al bijna klaar was, besloot mijn school om over te stappen van Dokeos (en hun zelfgemaakt volgsysteem) naar Smartschool.



Smartschool is een platform dat draait op een externe server van het bedrijf zelf. De database, en de code van het platform zijn niet toegankelijk. Dit maakt het onmogelijk om met WisaReg een synchronisatie tussen 'Databases' te doen, omdat deze niet toegankelijk zijn.

Toch biedt smartschool een mogelijkheid om accounts te beheren via hun API. Aangezien ik slechts over een oude API beschikte van 5 jaar oud, heb ik hier niet mee kunnen experimenteren.

Mijn leerkracht Steven Mertens gaat de taak op zich nemen om WisaReg te integreren met Smartschool. Om hem hierbij te helpen laat ik in dit stuk documentatie na hoe hij het best te werk zou gaan.

5.6.1 Lezen van de API documentatie van smartschool

Het eerste dat hij zal moeten doen is het lezen van de API documentatie van smartschool. Hier moet hij vooral letten op welke gegevens een account heeft en hoe het wachtwoord word doorgestuurd:

- Als het MD5 encryptie is, dan kan hij de data in de persistente lokale database gebruiken
- Als het leesbare tekst is, dan zal hij in de module **ldap_passwordsaver** de MD5 hasher moeten uithalen en in de routine **copyWisa** van de module **dbSync**. Zodat de wachtwoorden in zuiver leesbare tekst wordt weergegeven (niet ethisch). Merk op dat hij dan ook moet voorzien dat wanneer hij toch zaken zoals Dokeos/volgsysteem gaat gebruiken, hij daar wel nog een MD5 hasher zal moeten tussensteken in **db_sync**
- Als het een andere encryptie is (SHA256) zal hij de hasher in **ldap_passwordsaver** en in de routine **copyWisa** van de module **dbSync** hieraan moeten aanpassen.

5.6.2 Maken van een interface

Het volgende dat hij zal moeten doen is het maken van een interface tussen de lokale database en smartschool. Deze methode laat hem toe om de lokale database te bereiken, zonder kennis te hebben van mijn programma. Hij kan in de module **db_sync** zien hoe hier de gegevens van de leerlingen/leerkrachten worden opgehaald.

Aangepaste code uit dbSync voor het halen van gegevens

```
// De items kunnen 1 voor 1 geladen worden via de routine
```

```
 getDatabaseAliases\(String\) Get a list of all database aliases using path
```

```
// Haal gegevens van een ITEM uit de directory van de lokale database
// Dit zal de configuratie laden van een item (l1n1stegraad)
String[] Settings = dbDriver.getDatabaseSettings(item, localdbPath);
// Haal de kolommen op (LL_ID, VOORNAAM, NAAM ,..)
String[] variables = dbDriver.getTableData(item, localdbPath);
// Verkrijg de SELECT query om de gegevens uit die kolommen te halen
String cmd = dbDriver.getTableSelectCmd(item, localdbPath, "select");
// Gebruik de settings om een connection string te maken
String constr = dbDriver.getDriverConStr(Settings[4], driverPath, Settings[0],
Settings[1], Settings[2], Settings[3]);

// Verbind met de lokale database met die connection string
dbDriver db = new dbDriver(constr);
// Doe een SELECT en save de data in de DataSet
DataSet data_local = db.command(cmd);
// Vervolgens moet je de dataset voor de gegevens (in volgorde)

// OM DE CREDENTIALS TE LADEN uit de persistente tabel, moet je nog een aparte query
// doen

// Haal de table naam op (l1n_1ste_graad)
String tableName = dbDriver.getTableName(item, localdbPath);

// In deze string komt onze nieuwe Query
String cmd_p;
try
{
    // Maak de query (Select alles van table p_l1n_1ste_graad)
    cmd_p = "SELECT * FROM `p_" + tableName + "`";

    // Save de output naar de dataset OUTPUT
    DataSet output = db.command(cmd_p);

    // Doorloop alle rijen
    for (int r = 0; r < output.Tables[0].Rows.Count; r++)
    {
        // Haal de credentials op (dit is 1 voor 1, maar het is
        // aangeraden dit tijdelijk op te slagen, vooralleer het naar
        // smartschool te sturen
        String username =
            (output.Tables[0].Rows[r].ItemArray.GetValue(0).ToString());
        String password =
            (output.Tables[0].Rows[r].ItemArray.GetValue(1).ToString()); // MD5 hash
        // Aangezien de tabel rijen evenveel is als de rijen in de
        // dataset van de gegevens, kunnen we het in de zelfde itteratie ophalen
        String ll_id =
            (data_local.Tables[0].Rows[r].ItemArray.GetValue(0).ToString());
    }
}
catch (Exception)
{
    // Some information could not be saved
}
}
```

Vervolgens moet hij die gegevens op zijn manier in smartschool krijgen. Het is aangeraden dat hij een manier heeft om te controleren of een account al bestaat, zodat er niet bij elke synchronisatie volledig nieuwe accounts bijkomen, of zelfs erger dat de vorige overschreven worden.

5.6.3 De interface laten starten

Hij zou zelf de synchronisatie kunnen laten starten op een geregeld tijdstip (via CRON of een geplande taak) of hij kan zijn interface laten starten na een Wisa->Local copy in het hoofdprogramma.

Het zou zelfs beter zijn om dit te doen, zo is hij zeker dat zijn synchronisatie niet begint op het moment dat Wisa->Local bezig is. En dan kan hij de source (**Item**) meesturen als argument, zodat enkel die groep wordt gesynchroniseerd.

Code WisaReg hoofdprogramma

```
public void wisaCopy_Thread(String alias)
{
    // Loop and wait for new run
    DateTime nextRun = DateTime.Now;
    try
    {
        nextRun = dbDriver.getNextScheduler_COPY(alias, config_path_localdb);
    }
    catch (Exception)
    {
        // Error, Performing force sync
        Log("Error while gathering Sync Date! Forcing now!");
    }

    while (thread_run == false)
    {
        Thread.Sleep(10000);
        DateTime currentTime = DateTime.Now;

        // Check if nextRun has paassed
        if ((nextRun.Ticks <= currentTime.Ticks) && (wisaUpdate_flag == false)
            && (wisaCopy_flag == false) && (syncCopy_flag == false) && pause == false)
        {
            wisaCopy(alias);
            nextRun = dbDriver.getNextScheduler_COPY(alias, config_path_localdb);
            if (nextRun.Ticks >= dbDriver.getSchedulerEnd_COPY(alias, config_path_localdb).Ticks)
            {
                Log("Performing reset for persistent data [" + alias + "] !");
                dbSync.dropP_Local(config_path_localdb, config_path_driverlist, alias);
            }
            Log("Next Wisa copy for [" + alias + "] @ " + nextRun.ToString("dd/MM/yyyy HH:mm"));
            // Start de interface met de source(alias) als argument
            // Zorg dat het ACHTER WisaCopy en ook ACHTER dropP
            Process.start("C:\MijnInterface.exe", alias);
        }
    }
}
```

6. Failsafes

Omdat WisaReg constant moet werken, heb ik veel failsafes moeten maken (dit is waarschijnlijk de grootste tijdconsumptie). Dit is nodig omdat er allerlei dingen kunnen mislopen:

- Stroompanne
- Netwerkfouten
- Fouten bij het toevoegen van gebruikers

6.1 Beschermen tegen SQL Errors

Namen en gegevens van leerlingen/leerkrachten worden alvorens ze in de SQL worden gezet, gecontroleerd op foutieve tekens. Soms kan het zijn dat deze bepaalde tekens bevatten die ook worden gebruikt in de syntax van SQL.

Mocht na het filteren toch nog een fout optreden, dan zal het toevoegen van die gebruiker mislukken. Maar alle andere gebruikers worden normaal zorgeloos toegevoegd.

6.2 Beschermen tegen stroompannes

Het kan gebeuren dat WisaReg ongewild wordt afgesloten. Ik heb alle mogelijke scenario's getest en ben uiteindelijk op een goede failsafe gekomen die al deze dingen beschermt.

6.2.1 Tussen twee synchronisaties

Wanneer WisaReg afsluit tussen een wachtperiode, dan zal bij het opstarten volgende berekening worden gemaakt:

```
PSEUDO:
DateTime now = DateTime.Now();
DateTime nextSync = dbDriver.getNextSync(..);
Int interval = dbDriver.getInterval(..);

If ((nextSync + interval) < now){
    dbSync.syncNow();
}else{
    waitForSync();
}
```

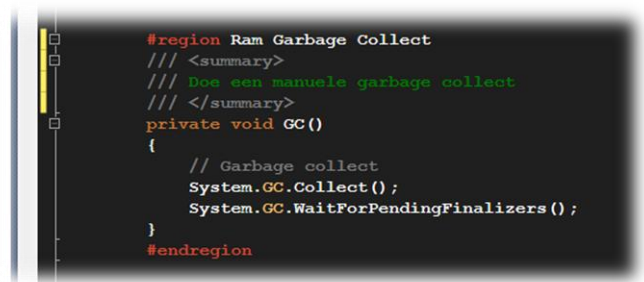
Het programma zal controleren of er niet te lang gewacht is tussen de laatste sync. en dat er geen synchronisaties zijn gemist.

6.2.2 Tijdens een synchronisatie

Wanneer het programma tijdens een synchronisatie stopt, dan is de database wellicht niet compleet. Daarom heb ik ervoor gezorgd dat bij het opstarten alle acties worden uitgevoerd, of deze nu op het moment zijn van de sync, of niet.

6.3 Beschermen tegen resource tekorten

Het hoofdprogramma van WisaReg, en alle configuratieschermen bevatten een thread die elke 10 seconden een manuele garbage collect doet. Dit zorgt ervoor dat het RAM verbruik van het programma minimaal blijft. Bij deze garbage collect gaat .NET nakijken welke variabelen er uit het geheugen mogen worden gehaald.

A screenshot of a code editor showing a C# region for garbage collection. The code is as follows:

```
#region Ram Garbage Collect
/// <summary>
/// Doe een manuele garbage collect
/// </summary>
private void GC()
{
    // Garbage collect
    System.GC.Collect();
    System.GC.WaitForPendingFinalizers();
}
#endregion
```

Alle loops hebben een time-out, zodat bij een error deze niet blijven lopen.

Ook is ervoor gezorgd dat in elke mogelijke situatie, de verbinding met databases veilig wordt gesloten.

7. Praktische test

Ik heb het programma meerdere keren getest, maar op het einde heb ik een volledige test gedaan van alle functies. Zo weet ik dat het eindproduct werkt als het wordt geïnstalleerd op een server.

De praktische test is opgedeeld in drie stukken:

- Installatie en configuratie
- Aanmaken van accounts
- Synchroniseren en updaten van accounts

7.1 Installatie



De eerste stap in het gebruik van WisaReg, is het installeren. Ik heb een makkelijke installatie voorzien die WisaReg en zijn benodigheden installeert.

Omdat deze versie van WisaReg nog niet 365 dagen is getest, en ik ook niet verantwoordelijk wil zijn dat alles 100% werkt, heb ik voor deze versie een wachtwoordbeveiliging toegepast. Zo kan ik min of meer hopen dat het niet verspreid geraakt.

Ik heb zeer veel van deze GIP genoten, en de kans bestaat dat ik het nog verder zet na het einde van het schooljaar (maar dan meer algemeen naar database synchronisaties). Daarom heb ik in de EULA gezet dat mijn programma niet mag worden aangepast zonder mijn toestemming.

De vereisten van de installatie zijn als volgt:

- .NET Framework 4.0
- Windows 2000 of hoger
- 40 Mb vrije ruimte



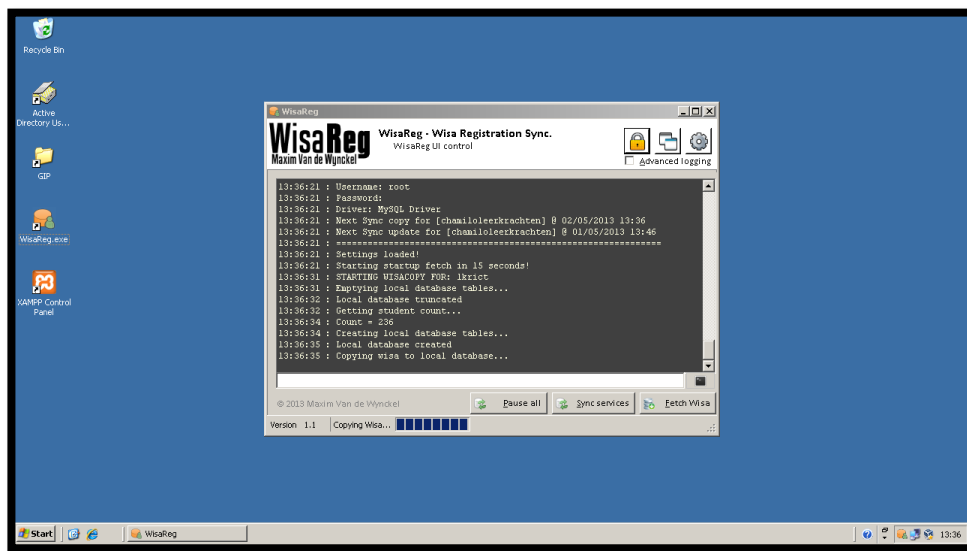
7.2 Configuratie

De volgende stap in het gebruik van WisaReg is het configureren. Voor de situatie van Don Bosco Hoboken, heb ik een aparte installatie voorzien die de configuratie mee installeert.

Voor meer informatie over het configureren, raadpleeg hoofdstuk 3

7.3 Starten van WisaReg

WisaReg wordt gestart door WisaReg.exe te starten. Het programma wordt standaard niet bij startup programma's gezet, zodat de volgorde van opstarten kan worden bepaald door de gebruiker zelf. Het spreekt vanzelf dat de database eerder opstart dan het programma.



Bij het opstarten zal eerst de configuratie worden getoond , en vervolgens word er een geforceerde sync van Wisa naar Lokaal gedaan. (Zie programmaverloop)

Als we nu eens gaan kijken in onze MySQL databank, zal U zien dat er verschillende tables zijn aangemaakt.

Tabel	Actie	Rijen	Type	Collatie	Grootte	Overhead
<input type="checkbox"/> lkr_ict	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	236	InnoDB	latin1_swedish_ci	64 K1B	-
<input type="checkbox"/> lln_1ste_graad	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	238	InnoDB	latin1_swedish_ci	112 K1B	-
<input type="checkbox"/> lln_2de_graad	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	459	InnoDB	latin1_swedish_ci	176 K1B	-
<input type="checkbox"/> lln_3de_graad	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	360	InnoDB	latin1_swedish_ci	160 K1B	-
<input type="checkbox"/> p_lkr_ict	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	236	InnoDB	latin1_swedish_ci	48 K1B	-
<input type="checkbox"/> p_lln_1ste_graad	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	238	InnoDB	latin1_swedish_ci	48 K1B	-
<input type="checkbox"/> p_lln_2de_graad	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	459	InnoDB	latin1_swedish_ci	64 K1B	-
<input type="checkbox"/> p_lln_3de_graad	Verkennen Structuur Zoeken Invoegen Legen Verwijderen	360	InnoDB	latin1_swedish_ci	64 K1B	-
8 tabellen	Som	2,586	InnoDB	latin1_swedish_ci	736 K1B	0 B

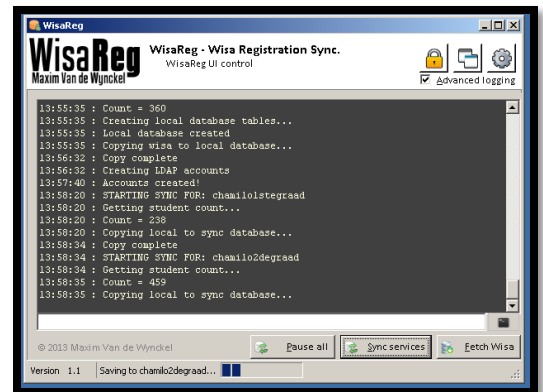
Voor elke groep is een aparte tabel aangemaakt. De persistente data (Tabellen met een 'p_' voor) zijn de tabellen die de login gegevens bewaren. Deze zijn niet onderhevig aan volledige kopies, zodat de wachtwoorden niet verloren gaan.

7.4 Syncen naar platform

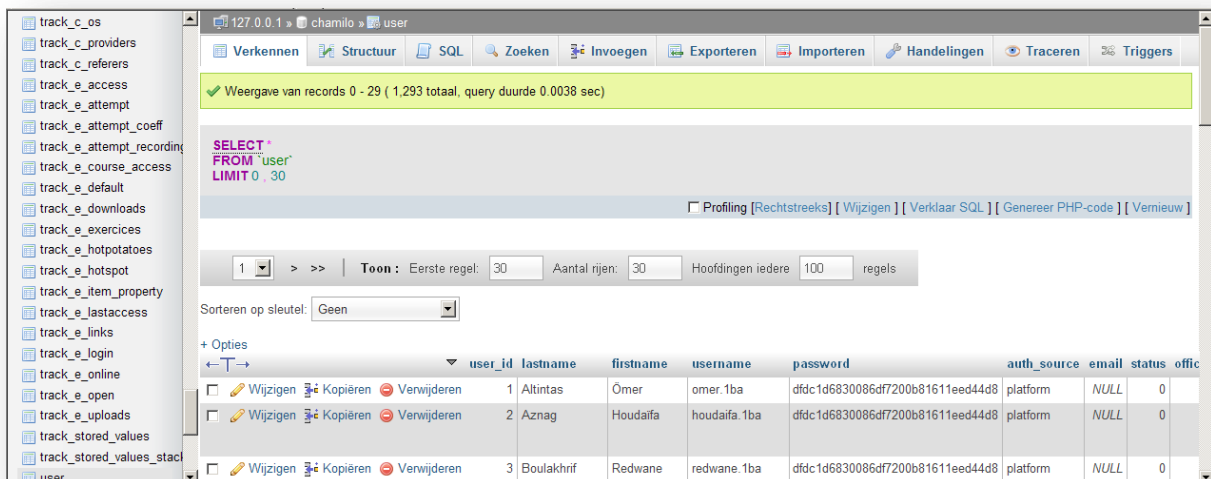
Als het de eerste keer is dat U WisaReg start na het configureren, dan zal (indien het schooljaar al begonnen is) de sync starten die de gegevens van de leerlingen gaat gebruiken om accounts aan te maken.

Deze accounts zullen vervolgens gekopieerd worden naar het ingestelde platform.

Indien het platform is toegevoegd nadat er reeds accounts zijn aangemaakt in LDAP. Dan zullen de laatste wachtwoorden worden gebruikt om de accounts toe te voegen.



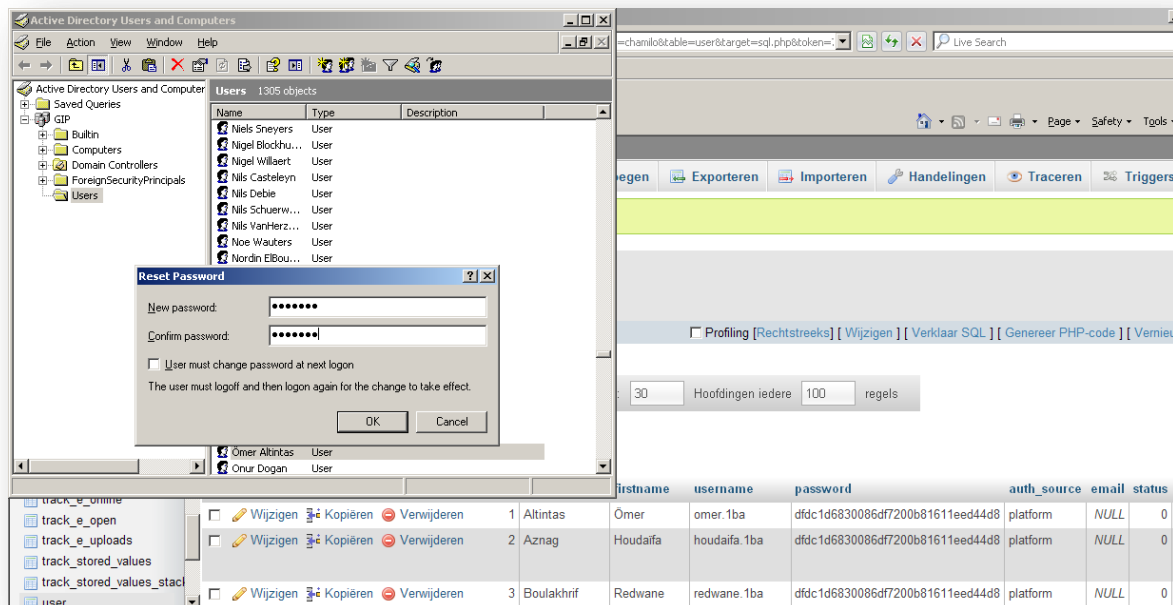
Als we in de database van Chamilo gaan zien, dan zullen we merken dat alle leerlingen en leerkracht nu in de tabel **user** staan.



We kunnen nu inloggen met het meest recente wachtwoord op ons platform! merk op dat voor het goede verloop van wachtwoord synchronisatie, wachtwoord wijzigingen in het platform moeten worden uitgezet.

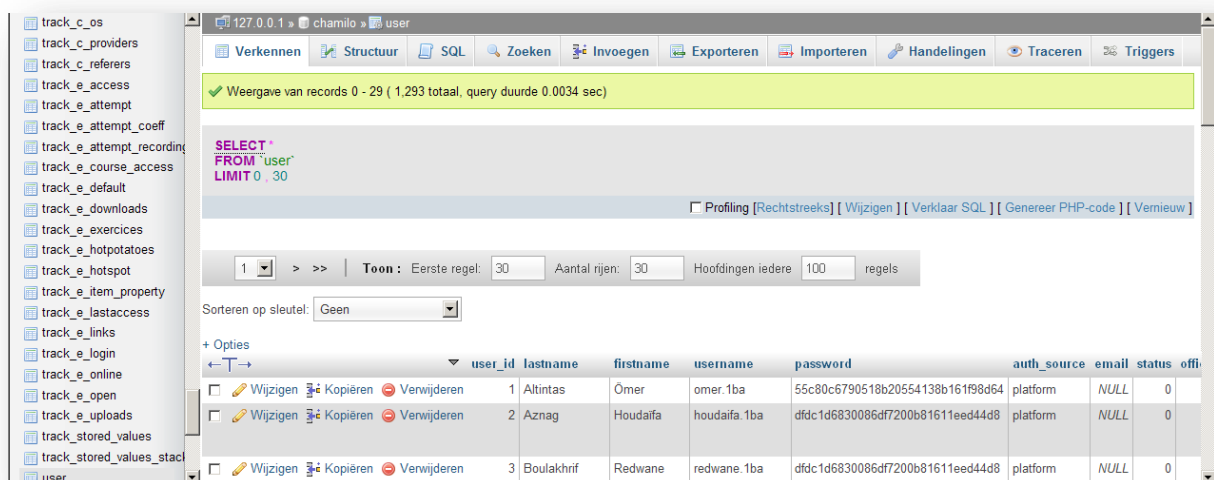
7.5 Syncen van wachtwoorden

Voor deze test gaan we het wachtwoord van een LDAP gebruiker resetten. Dit zal de wachtwoord filter een bestand laten lopen die de gegevens in de lokale database aanpast.



Op de screenshot ziet U op voorgrond het resetten van een wachtwoord van een willekeurige persoon. Op de achtergrond zie U de webinterface van phpMyAdmin die de tabel 'user' toont met het ongewijzigde wachtwoord.

Zodra U een dialoog krijgt dat het wachtwoord gewijzigd is, is dit ook gebeurt voor de lokale database. Het wachtwoord op het platform wijzigt niet totdat er een update is gebeurt (5 minuten).



8. Gekende bugs

Deze lijst met Gekende bugs werd opgemaakt op 26 April 2013 als een Todo lijst voor de laatste afwerking van mijn GIP. In de tussentijd kunnen sommige van deze bugs zijn hersteld, maar deze zullen dan doorstreept zijn. In onderstaande lijst vindt U telkens de BUG ID, met meer uitleg. En de benodigde tijd om dit te herstellen.

BUG ID	UITLEG	Benodigde Tijd
WS260413_01	Bij het wijzigen van een source in ldap settings, zullen de comboboxen waarmee je de voornaam selecteert, niet opnieuw laden	2 minuten
WS260413_02	Bij het wijzigen van een source in alle settings met een synchronisatie tab, zullen de comboboxen met minuten, uur, .. niet opnieuw laden	2 minuten
WS260413_03	Bij het uitvoeren van een command die een Sync start, zal de progressbar niet gecleared worden, en word er geen status gezet	40 minuten
WS260413_03	De Settings knop word niet gelocked bij het opstarten van WisaReg	3 minuten
W260413_04	WisaReg kan geunlocked worden, maar nog niet gelocked	10-15 minuten
W260413_05	History toont HTML sites in textbox van de commando's	1 minuut
WD260413_06	Domain installer moet de passwordhook in POST change zetten i.p.v. PRE change	15 minuten
WM260413_07	LDAP Accounts worden nog 'redelijk' traag aangemaakt, dit moet nageken worden	30min – 1 uur
WM260413_08	Update voor Syncdb en localdb werkt niet (path not valid)	1 – 2 uur
W260413_09	Tray menu moet nageken worden	5 minuten
W260413_10	De laatste sync moet gecontroleerd worden dat deze persistent data delete	2 uur
WS260413_11	Er moet een checkbox die aanduid dat LDAP account mogen worden gedeleted op het einde van het jaar	20 minuten
WM260413_12	De Interval is meestal groter dan de ingestelde waarde, omdat er gewacht word vooraleer alle voorgaande sync's gedaan zijn.	Niet oplosbaar zonder herberekening
W260413_13	Na het maken van LDAP accounts in CopyWisa, word de progressbar niet gereset	40 minuten
W260413_13	Bij sommige synchronisaties komt er 'Path is not of legal form' op	2 uur
WS270413_1	Soms geeft hij in de settings bij het opslaan, dat er niets is gewijzigd, terwijl dit wel zo is	30 minuten
W280413_1	Er gebeurt een Truncate op het einde van het jaar i.p.v. individueel deleten	1 uur
WS290413_1	Settings thumbnails naar 1 map zetten voor snelheid	10 minuten
W290413_2	Memory leak in WisaReg hoofdapplicatie	10 minuten
W290413_3	Password hook wacht nu default 0 ms , waardoor er veel ldap_passwordsaver.exe's openen.	Gemiddelde zoeken (niet oplosbaar)
W010513_1	SyncReset delete geen gegevens	1 uur
WS240513_1	Sync database saved een verkeerde DELETE SQL code	20 minuten

Laatst aangepast op 24/05/2013

9. Niet geïmplementeerde functies

Wegens tijdsgebrek zijn sommige “extra” functies niet geïmplementeerd. Bij veel van deze functies is er wel al een begin gelegd, en zijn zeker de moeite waard om verder mee te gaan.

9.1 Module config_langldr

De module config_langldr is een module die elk bestuuringselement in het programma gaat scannen op tekst. Vervolgens gaat hij deze tekst opslagen in een ‘language file’. Op deze manier kan ik gemakkelijk mijn programma maken in een bepaalde taal, en zal bij elke build alle tekst in een file worden gezet. Vervolgens hoef ik enkel in deze file de tekst te vertalen.

De reden waarom deze module niet in het programma is geraakt, komt omdat het te veel tijd kost om alles te vertalen. Maar eigenlijk werkt het volledig.

Code van deze kleine maar krachtige module

```
/* ----- *
 * File: config_langldr.dll *
 * Author: Maxim Van de Wynckel *
 * Version: 1.0.0 *
 * Date: 12/2012 *
 * ----- */

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using config.ini;

namespace config.langldr
{
    public class loadLanguage
    {
        // Global Variables
        #region Global Variables
        public Object frm = null; // Form that contains controls
        public String saveName = null; // Project name
        #endregion

        // Initialize the class
        #region Initialize
        /// <summary>
        /// Initialize the language loader/saver
        /// </summary>
        /// <param name="sender">The form you want to load/save language for</param>
        /// <param name="project">The project name</param>
        public loadLanguage(Object sender, String project)
        {
            frm = sender; // Save global variables
            saveName = project;
        }
        #endregion

        // Save the control text/titles to a file
        #region Save Controls
        /// <summary>
        /// Save the text and items of all controls to language file
        /// </summary>
        /// <param name="langFile">The location of the language file</param>
        public void saveTextLanguage(String langFile)
        {
            // Perform the action for Every control
            // Get all controls on form
            var controls = GetAll((Form)frm);
            // Open configuration
            iniConfig config = new iniConfig(langFile);
        }
    }
}
```

```
// Write text from controls into language file
foreach (Control control in controls)
{
    if (control.Text != "")
    {
        // If the control has a 'Text' then save it
        config.WriteValue(this.saveName, control.Name, control.Text.Replace("\n", @"\n").Replace("\r",
@"\r"));
    }
    if (control.GetType() == typeof(ToolStrip))
    {
        // Get all the items in that ToolStrip
        foreach (ToolStripItem item in ((ToolStrip)control).Items)
        {
            // Save the items
            if (item.Text != null)
            {
                config.WriteValue(this.saveName, item.Name, item.Text.Replace("\n", @"\n").Replace("\r",
@"\r"));
            }
            if (item.ToolTipText != null)
            {
                config.WriteValue(this.saveName, item.Name + "_tt", item.ToolTipText.Replace("\n",
@"\n").Replace("\r", @"\r"));
            }
        }
    }
}

/// <summary>
/// Save the tooltip text of all items into the language file
/// </summary>
/// <param name="langFile">The location of the language file</param>
/// <param name="tooltip">The tooltip object to get txt from</param>
/// <param name="tooltipctrl">The name of the Tooltip control</param>
public void saveToolTipLanguage(String langFile, ToolStrip tooltip, String tooltipctrl)
{
    // Perform the action for Every control
    // Get all controls on form
    var controls = GetAll((Form)frm);
    // Open configuration
    iniConfig config = new iniConfig(langFile);
    // Write text from controls into language file
    foreach (Control control in controls)
    {
        if (tooltip.GetToolTip(control) != "")
        {
            // If the control has a 'ToolTip' then save it
            config.WriteValue(this.saveName, control.Name + "_" + tooltipctrl.Replace("\n",
@"\n").Replace("\r", @"\r"), tooltip.GetToolTip(control).Replace("\n", @"\n").Replace("\r", @"\r"));
        }
        if (tooltip.ToolTipTitle != "")
        {
            // Save tooltip title
            config.WriteValue(this.saveName, tooltipctrl.Replace("\n", @"\n").Replace("\r", @"\r"),
tooltip.ToolTipTitle.Replace("\n", @"\n").Replace("\r", @"\r"));
        }
    }
}
#endregion

#region Load Controls
/// <summary>
/// Load the tooltip text of all items into the language file
/// </summary>
/// <param name="langFile">The location of the language file</param>
/// <param name="tooltip">The tooltip object to get txt from</param>
/// <param name="tooltipctrl">The name of the Tooltip control</param>
public void loadToolTipLanguage(String langFile, ToolStrip tooltip, String tooltipctrl)
{
    // Perform the action for Every control
    // Get all controls on form
    var controls = GetAll((Form)frm);
    // Open configuration
    iniConfig config = new iniConfig(langFile);
    // Write text from controls into language file
    foreach (Control control in controls)
    {
        if (tooltip.GetToolTip(control) != "")
```

```
        {
            // If the control has a 'ToolTip' then save it
            if (config.ReadValue(this.saveName, control.Name + "_" + tooltipctrl.Replace("\n",
@"\n").Replace("\r", @"\r")) != null)
            {
                tooltip.SetToolTip(control, config.ReadValue(this.saveName, control.Name + "_" +
tooltipctrl.Replace("\n", @"\n").Replace("\r", @"\r")).Replace(@"\r", "\r").Replace(@"\n", "\n"));
            }
        }
        if (tooltip.ToolTipTitle != "")
        {
            // Save tooltip title
            tooltip.ToolTipTitle = config.ReadValue(this.saveName, tooltipctrl.Replace("\n",
@"\n").Replace("\r", @"\r")).Replace(@"\r", "\r").Replace(@"\n", "\n");
        }
    }
}

/// <summary>
/// Load the language file of the text
/// </summary>
/// <param name="langFile">The language file of the translation</param>
public void loadTextLanguage(String langFile)
{
    // Perform the action for Every control
    // Get all controls on form
    var controls = GetAll((Form)frm);
    // Open configuration
    iniConfig config = new iniConfig(langFile);
    // Read text from controls into language file
    foreach (Control control in controls)
    {
        if (control.Text != "")
        {
            // If the control has a 'Text' then load it
            control.Text = config.ReadValue(this.saveName, control.Name).Replace(@"\r", "\r").Replace(@"\n",
"\n");
        }
        if (control.GetType() == typeof(ToolStrip))
        {
            // Get all the items in that ToolStrip
            foreach (ToolStripItem item in ((ToolStrip)control).Items)
            {
                // Load the items
                item.Text = config.ReadValue(this.saveName, item.Name).Replace(@"\r", "\r").Replace(@"\n",
"\n");
                item.ToolTipText = config.ReadValue(this.saveName, item.Name + "_tt").Replace(@"\r",
"\r").Replace(@"\n", "\n");
            }
        }
    }
}
#endregion

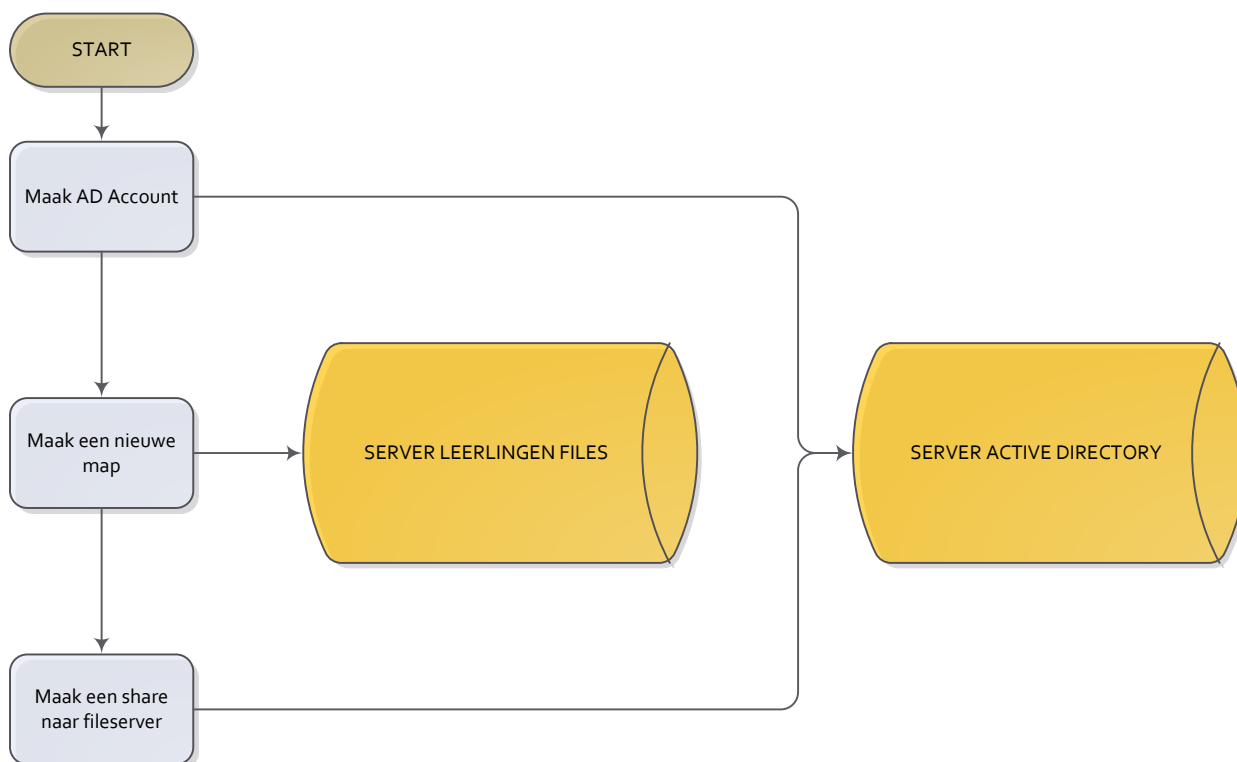
/// <summary>
/// Get all controls from an object
/// </summary>
/// <param name="control">The object (form)</param>
/// <returns>A list with controls</returns>
public IEnumerable<Control> GetAll(Control control)
{
    var controls = control.Controls.Cast<Control>();
    return controls.SelectMany(ctrl => GetAll(ctrl))
        .Concat(controls);
}
}
```

9.2 Updater

Een updater voor WisaReg was één van de extra's die voor mij veel prioriteit had. Ik hoopte dat wanneer ik een updater had, ik een update kon voorzien nadat ik (hopelijk) was afgestudeerd. De updater maakte gebruik van mijn persoonlijke engine, die ik in andere projecten gebruikte. Maar het aanpassen voor WisaReg zou iets te lang duren. Daarom heb ik wijs besloten om deze extra niet te maken.

9.3 Windows share per gebruiker

In veel bedrijven en scholen krijgt elke gebruiker op zijn computer een schijf, waar hij/zij documenten kan zetten die zichtbaar zijn op elke computer in dat netwerk. Deze functie heb ik jammer genoeg niet kunnen toepassen op de manier zoals het school het nu gebruikt.



Eerst werd er een Active directory account gemaakt. Vervolgens werd er via het UNIX command **mkdir** een map gemaakt in de leerlingen file server. Tot slot ging het script de AD server aanpassen zodat de user share op de file server was.

Het linken van een folder naar het AD ging (en is intern ook geïmplementeerd) maar het maken van een folder op een remote fileserver ging niet.

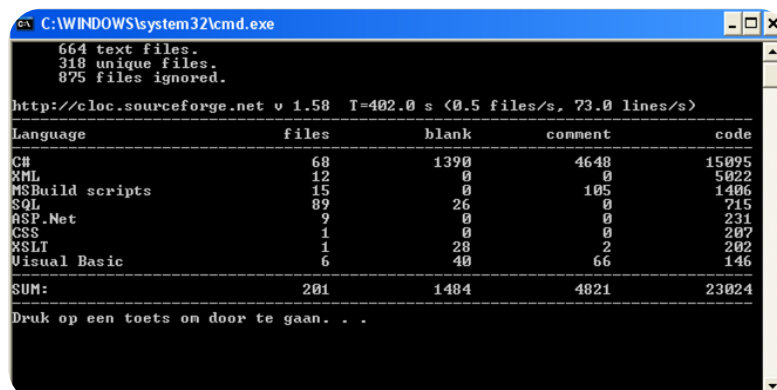
Dit alles is niet echt zo een groot probleem, omdat ik op de opendeurdag tot de ontdekking kwam van iemand die al een geruime tijd met AD's en Shares bezig is, dat dit mogelijk is om automatisch per groep en per account een user share aan te maken.

10. Besluit

Ik heb zeer veel bijgeleerd met dit project. Vooral over LDAP en Active Directory. Toen ik begon met dit project was mijn grootste angst dat het niet mogelijk zou zijn om hier wachtwoorden uit te halen. Dit gevoel werd nog erger toen ik me er begon in te verdiepen. Als er iets is dat ik over LDAP heb geleerd is dat het alle deuren drie keer sluit, maar het raam openlaat, ik ben blij dat ik via de wachtwoord filters een goede oplossing heb gevonden.

Ook mijn manier van gebruikersinterface is volgens mij verbeterd. Telkens als ik een methode had om deze te verbeteren, kreeg ik steeds weer een beter idee. Ik hoop dat het eindresultaat begrijp baar is voor anderen, en dat ze het net zo handig vinden als ik.

Het enige dat ik niet volledig heb afgekregen aan mijn GIP was het maken van shared folders. Dit omdat ik met mijn programma geen externe verbindingen kon maken naar een file server. Toch is dit probleem geen ramp, aangezien er veel third party software bestaat die dit kan toepassen bij LDAP groepen.



```
C:\WINDOWS\system32\cmd.exe
664 text files.
318 unique files.
875 files ignored.

http://cloc.sourceforge.net v 1.58 T=402.0 s (0.5 files/s, 73.0 lines/s)
-----
Language          files      blank      comment      code
-----
C#                  68          1390         4648        15895
XML                 12           0           0           5822
MSBuild scripts    15           0           105          1406
SQL                 89          26           0           715
ASP.Net            9            0           0            231
CSS                 1            0           0            207
XSLT                1           28           2            202
Visual Basic       6            40          66            146
-----
SUM:                201         1404         4821        23024
Druk op een toets om door te gaan. . .
```

Handtekening:

11. Bibliografie

- EWoodruff. (sd). *Sandcastle Help File builder*. Opgehaald van XML to html help files:
<http://shfb.codeplex.com/>
- Goel, N. (2009, 04 20). *Sync LDAP users with Google apps*. Opgeroepen op 09 20, 2012, van Google Enterprise: <http://googleenterprise.blogspot.be/2009/04/sync-google-apps-user-accounts-with.html>
- Microsoft. (2005, Januari 21). *LDAP - Reversible encryption*. Opgeroepen op Januari 5, 2013, van Microsoft Technet: <http://technet.microsoft.com/en-us/library/cc784581%28WS.10%29.aspx>
- Microsoft. (sd). *MSDN Microsoft*. Opgeroepen op November 5, 2012, van MSDN: [http://msdn.microsoft.com/en-us/library/ms257181\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/ms257181(v=vs.80).aspx)
- Password Hook. (2011, Maart 4). Password Hook DLL.
- Stackoverflow. (sd). *Update or Delete LDAP users*. Opgeroepen op 04 2013, van <http://stackoverflow.com/questions/8364944/how-to-programmatically-update-and-delete-ldap-users-from-sql>
- Teusink, N. (2009, Augustus 25). *LDAP - Reversible encryption*. Opgeroepen op September 20, 2012, van teusink: <http://blog.teusink.net/2009/08/passwords-stored-using-reversible.html>
- Wisa bvba. (sd). *Wisa Schoolsoftware*. Opgeroepen op September 02, 2012, van Wisa Schoolsoftware: <http://www.wisa.be/>
- Zuijderwijk, J. (2011, Juni 9). *Helpmij.nl Configuratiebestanden*. Opgeroepen op Juni 23, 2011, van Helpmij.nl: <http://www.helpmij.nl/forum/showthread.php/633161-Configuratiebestanden-VB-NET>