



Datavetenskap

Daniel Rådesjö

**Automatisk uppdatering av information
angående datorer i ett nätverk**

Examensarbete

2004:29

Automatisk uppdatering av information
angående datorer i ett nätverk

Daniel Rådesjö

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Daniel Rådesjö

Godkänd, 2004-06-04

Handledare: Johan Eklund

Examinator: Martin Blom

Sammanfattning

Rapporten beskriver ett projekt som utfördes för TietoEnator Processing & Manufacturing (PM) Sweden.

En av PMs kunder använder idag en applikation som tillhandahåller information om datorer och programvaror som finns i datorer i ett nätverk, samt information om licenser för programvarorna. Informationen används för att bland annat ha kontroll över vilka licenser som finns och när dessa löper ut. Under applikationen finns informationen sparad i en databas. Via ett webbgränssnitt visas information för användaren och via detta gränssnitt matas informationen in manuellt. Nu vill TietoEnator slippa den manuella inmatningen, både för att slippa det tidskrävande arbetet och risken för felinmatning.

Uppgiften som denna rapport beskriver innebär att komplettera applikationen så att den automatiskt uppdaterar databasen utan att därför begränsa möjligheten till manuell inmatning eller förändra webbgränssnittet.

Det mesta av informationen som automatiskt skall importeras till applikationen finns att hämta från en annan programvara som finns installerad hos kund.

Examensarbetet resulterade i en programvara som hämtar informationen om datorerna och sedan importerar informationen till den befintliga applikationen.

Automatic update of information about computers in a network

Abstract

This document describes a project which was performed for TietoEnator Processing & Manufacturing (PM) Sweden.

One of PM's customer is today using an application which supplies information about computers and software from computers connected to a network, and information about the software licenses. The information is used for controlling which licences are used and when they expire. Under the application the information is stored in a databas.

By a webinterface the information is shown to the user and manually entered to this application. Now the customer want to skip this because it takes time and there is a risk of entering the wrong information.

The task this report describes involves a complement to the application which automaticly updates the database without limit the possibility to manual entering of information or change the webinterface.

The most of the information which will automaticly be imported to the application can be collected from another software which is installed on a computer in the customers network. The result of the work were a software which collects the information about the computers and then imports it to the existing application.

Tack

Ett stort tack till Johan Eklund på Karlstad Universitet för hjälp med rapporten.

Även ett stort tack till Ann Fahlin och Kenneth Åhman på TietoEnator PM i Karlskoga för bra ledning genom examensarbetet och för att de alltid fanns till hands när frågor uppstod.

Tack även till Magnus Hjalmarsson på TietoEnator för hjälp och idéer med utvecklingen.

Tack till Per Hurtig för hjälp med LaTeX.

Innehåll

1	Introduktion	1
1.1	Definition av problemet	1
1.2	Uppgiften	1
1.3	Förutsättningar	2
1.4	Övrigt	3
2	Bakgrund	4
2.1	Beskrivning av TietoEnator	4
2.2	Beskrivning av applikationen som används idag	5
2.3	Överblick av uppgiften	6
2.4	Specifikation av uppgiften	7
2.5	Manuell inmatning av information	7
2.6	Microsoft System Management Server (SMS)	8
2.7	Microsoft Visual Basic .Net	8
3	Design	9
3.1	Introduktion	9
3.2	Detaljerad beskrivning	10
3.3	Tillägg SMS 2003	11
3.4	Läsa XML och Import till applikationens databas	14
3.5	Tjänst	15
4	Implementation	16
4.1	Introduktion	16
4.2	Utvecklingsmiljö	16
4.3	Uppdelning av funktioner	16
4.4	Systemkrav	17

4.5	Kommentar till implementationen	17
5	Resultat	18
5.1	Introduktion	18
5.2	Beskrivning	18
5.3	Problem	19
6	Slutsats	21
6.1	Introduktion	21
6.2	Beskrivning	21
6.3	Ytterligare arbete	22
6.4	Framtida utveckling	22
6.5	Sammanfattning av arbetet	23
	Referenser	24
A	Bilagor	25
A.1	Systemdokumentation	25
A.2	Användarmanual	30
A.3	Källkod	39

Figurer

2.1	TietoEnator	5
3.1	Översiktsbild av programmet	9
3.2	Steg för importera information	11
3.3	Detaljerad bild över “Tillägg SMS 2003”	12
3.4	Detaljerad bild över “Läsa XML och Import till applikationens databas”	14

1 Introduktion

Inom företag finns ofta många datorer som är sammankopplade i ett nätverk. I datorerna finns både hård- och mjukvara som företaget skall kontrollera, underhålla och uppdatera. Information om licenser, hård- och mjukvara hanteras ofta i en databas. Anledningen till att företag samlar informationen är för att underlätta vid inköp och förnyelse av produkter och licenser. Vid manuell hantering av databasen finns flera problem, dels kräver det mycket administration, vilket kostar pengar, dels finns risker att fel information hamnar i databasen.

1.1 Definition av problemet

Examensarbetet kommer att utföras på TietoEnator Process & Manufacturing Sweden (PM), men kommer att utföras åt en av deras kunder.

Kundens namn är i detta examensarbete irrelevant och kommer därför inte att nämnas.

Hos kunden finns idag en applikation på en server för att hålla ordning på vilka licenser som finns för olika program, samt vilka program som finns installerade på respektive dator i nätverket. Applikationens databas är en Microsoft SQL Server 2000 [1] databas som finns på en server i kundens nätverk. Denna information är samlad för att undvika att licenser löper ut och för att ha kontroll på hur många licenser som används. Idag skrivs all information om licenserna in till applikationen manuellt via ett webbgränssnitt. För att slippa kostnaden, det tidskrävande arbetet, och för att minimera risken för fel vill uppdragsgivaren nu att informationslagringen skall skötas så mycket som möjligt med automatik.

1.2 Uppgiften

Examensarbetet innebär att utveckla ett program som automatiskt samlar information om datorerna och installerade program i ett nätverk och importerar denna information till den befintliga applikationens databas.

Informationen om datorerna skall hämtas automatiskt för att uppdatera databasen, men möjlighet till manuell inmatningen skall finnas kvar. Exempelvis kan automatisk import ske varje natt, då det inte finns någon "normaltrafik" på nätverket. Programmet skall även vara dynamiskt konstruerat för att det i ett senare skede skall kunna kompletteras för att enkelt kunna hämta information från olika typer av hårdvaror och datakällor.

1.3 Förutsättningar

Hos kunden finns idag en blandad datormiljö med både persondatorer, arbetsstationer och servrar. De flesta anställda använder en Intel-baserad dator med Microsoft Windows [2] som operativsystem. Därför har TietoEnator PM valt att detta examensarbete skall koncentrera sig på licenser för den typen av datorer.

Kunden använder idag ett program som heter System Management Server (SMS) [3]. Detta program används till bland annat fjärrhjälp och fjärrinstallation av program, men innehåller också information i en databas om varje dator i nätverket. Informationen kan visas i rapportform via ett administrationsverktyg. Idag används en version som heter 2.0 av SMS, men det är tänkt att kunden skall uppgradera till den nyare version som heter SMS 2003. Därför har TietoEnator framfört att i detta examensarbete bör import av data ske från SMS 2003. Informationen som finns i SMS är sparad i en Microsoft SQL Server 2000 [1] databas.

Programmen som skall sköta den automatiska importen av data kommer att exekveras på kundens nät där tillgång finns till både SMS och applikationens databas.

Anledningen till att den befintliga applikationens databas inte kopplas direkt mot SMS databas är att applikationen skall handha mer information om datorerna och licenserna än vad som finns i SMS databas, och för att stöd för andra hårdvaror och system, exempelvis Unix, skall finnas med. Denna information skall även fortsättningsvis matas in manuellt via webbgränssnittet. Exempel på information som skall matas in manuellt är pris, inköpsdatum och antal för varje licens.

Det programspråk som skall användas vid utvecklingen är Visual Basic .NET där mjukvara kan ansluta till databaser på ett förhållandevis enkelt sätt. En annan anledning är att kompetens inom Visual Basic finns hos TietoEnator. Utvecklingsverktyget som används heter Microsoft Visual Studio .NET 2003 [4] där Visual Basic .NET [5] är en del av det verktyget.

1.4 Övrigt

Namnet TietoEnator används i rapporten synonymt med uppdragsgivaren.

Företaget där examensarbetet skall driftsättas används synonymt med kunden.

2 Bakgrund

Kapitlet beskriver bakgrunden till examensarbetet. Det finns en beskrivning av uppdragsgivaren TietoEnator samt en överblick över uppgiften i examensarbetet.

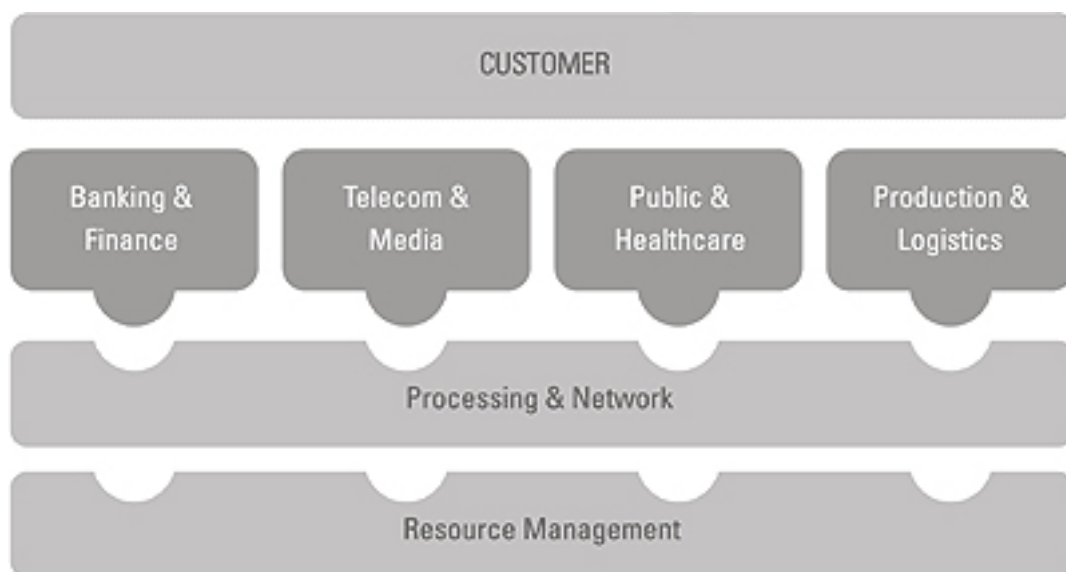
System Management Server (SMS) och Visual Basic .Net kommer även att beskrivas eftersom det är dessa applikationer som skall användas.

2.1 Beskrivning av TietoEnator

Examensarbetet utfördes hos TietoEnator Process & Manufacturing (PM) Sweden i Karlskoga. Företaget är en del av TietoEnator som är en sammanslagning av finska Tieto och svenska Enator. Sammanslagningen skedde den 7 juli 1999 [6]. TietoEnator har i dagsläget ungefär 14000 anställda i över 20 länder. Företaget har specialiserat sig på konsulttjänster, utveckling, drift och support inom områdena:

- Bank och finans
- Telekommunikation
- Kommun och landsting
- Energi och skogsindustri

TietoEnator PM ingår i Processing & Network, vilka i huvudsak arbetar med drift av servrar och support inom informationssystem och nätverk. Genom att specialisera sig inom detta område kan de bland annat tillhandahålla ökad effektivitet, kvalitet och datasäkerhet till sina kunder. Processing & Network finns idag i 8 europeiska länder och har drygt 2400 anställda. [7][9]



Figur 2.1: TietoEnator
[8]

I Karlskoga, där examensarbetet utfördes, finns ca 100 anställda. Det var efter uppdelningen av Bofors AB i Karlskoga som Enator köpte datasupportavdelningen och tog över all drift och support av datorutrustningen som fanns.

Processing & Manufacturing tillhandahåller tjänster inom bland annat drift av servrar, kundsupport och projektledning. De tillhandahåller även en egen ServiceDesk, dit kunder kan ringa in om de exempelvis har problem med en arbetsstation.

2.2 Beskrivning av applikationen som används idag

Idag används en egenutvecklad applikation för att få en bra överblick över vilka licenser som finns och vilka programvaror som används av de anställda. Applikationen är utvecklad av TietoEnator, men har fram till i början av 2004 underhållits av kunden själv, men skall nu underhållas av TietoEnator. Anledningen till att detta system används är för att förhindra att exempelvis licensavtal löper ut.

TietoEnator har valt att utveckla denna applikationen för hantering av licenser istället för att använda System Management Servers (SMS) inbyggda rapporter. Detta för att dels ha stöd för andra system än dem som SMS stödjer, dels för att kunna söka i informationen på ett enkelt sätt och dessutom kunna lägga till ytterligare egenskaper för exempelvis varje licens som matas in manuellt. Både kund och utvecklare kan även då komma med egna idéer på förbättringar och förändringar, vilket gör att applikationen blir anpassad efter kundens datormiljö.

2.3 Överblick av uppgiften

Idag skrivs all information angående licenser in manuellt vilket gör att det är både tidskrävande och dyrt att underhålla applikationen. En annan aspekt är att den mänskliga faktorn gör att det lätt blir fel information som skrivs in, vilket gör att applikationens databas inte alltid innehåller rätt information vilket i sin tur medför att på lång sikt kommer applikationen inte att användas. För att underlätta och förhindra att felinmatning sker, skall så mycket som möjligt av informationen hämtas automatiskt.

All information kommer inte att kunna hämtas från SMS databas på grund av att all information som skall finnas i applikationens databas inte finns sparad i SMS databas. Den grundläggande informationen rörande exempelvis vilka programvaror som finns installerade på en dator och vilka anställda som använder en dator ska dock kunna samlas in från SMS databas.

Eftersom större delen av datorerna hos kunden är Intel-baserade med Microsoft Windows som operativsystem har uppdragsgivaren valt att specialisera sig på information om dessa applikationer och datorer. Stöd för andra befintliga och kommande system finns också med, då krävs dock viss anpassning av programmet. Anpassningen till andra system skall kunna gå snabbt, varför programmet måste vara anpassad för utbyggnad.

2.4 Specifikation av uppgiften

I uppgiften finns ett antal delar som skall utföras:

1. Hämta information från en datakälla som innehåller uppdaterad information om datorerna i ett nätverk
2. Importera information om datorerna till applikationens databas utan att begränsa möjligheten till manuell inmatning som sker idag
3. Skapa ett program som möjliggör automatisk import enligt ovanstående punkter

Anledningen till uppdelning i flera steg är att stöd för andra datakällor än SMS 2003 skall finnas med och då lätt kunna anpassas till dessa datakällor.

Den databasserver som både applikationen och SMS 2003 använder är Microsoft SQL Server 2000. Med tanke på detta implementeras uppgiften i Microsoft Visual Basic .Net eftersom det där finns färdiga gränssnitt mot databasservern, vilket gör det enklare att ställa frågor och hämta information från databaserna.

2.5 Manuell inmatning av information

Via ett webbgränssnitt skrivs idag informationen (T.ex. programnamn, datornamn osv.) om varje licens och dator in för att sedan sparas till applikationens databas. Gränssnittet är utvecklat i Microsoft Active Server Pages (ASP) [11] av TietoEnator för deras kund.

Med hjälp av HTML-sidor presenteras ett antal olika formulär där exempelvis information om programvaror och dess licenser matas in . Det finns också sidor där information från databasen presenteras för användaren.

2.6 Microsoft System Management Server (SMS)

Denna serverprogramvara är en klient-server programvara för nätverk med befintliga servrar och klienter. SMS tillhandahåller funktionalitet för att exempelvis fjärrinstallera programvaror, fjärradministrera datorer och även undersöka vilka datorer som finns i nätverket.

Den förutsättning som krävs är att varje dator i nätverket måste ha en SMS-klient installerad, detta medför att endast Microsoft Windows kan användas som operativsystem tillsammans med SMS. Informationen om varje dator är sparad i en databas under servern och därifrån kan informationen enkelt hämtas.

Det finns olika versioner av SMS, de två vanligaste är SMS 2.0 och SMS 2003.

De största nyheterna i SMS 2003 är bland annat stöd för operativsystemen Windows CE, Pocket PC och Windows XP Embedded vilket täcker in de flesta handdatorer, kassaapparater och inbyggda system [10].

All information som finns i SMS sparas i en Microsoft SQL Server 2000 databas. Ifrån den kan information om datorerna i nätverket hämtas till antingen Microsofts egen applikation för att visa det som en rapport, alternativt kan utveckling ske av en egen applikation.

2.7 Microsoft Visual Basic .Net

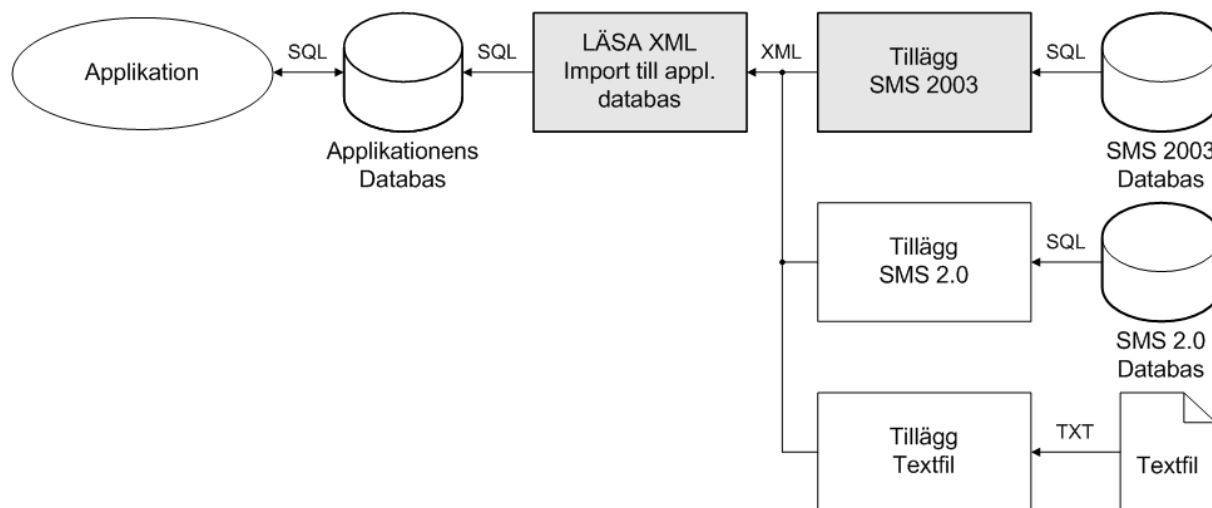
Microsoft Visual Basic .Net ingår i Microsoft Visual Studio .Net paketet. I det paketet ingår även Visual C#, Visual J# och andra utvecklingsverktyg för olika programspråk [4]. Visual Basic .Net är inte det effektivaste programspråket om hänsyn tas till tiden för program-exekvering. Eftersom funktionen som examensarbetet behandlade, inte är beroende av att snabbt överföra informationen om datorerna, togs ingen hänsyn till detta.

3 Design

Det här kapitlet beskriver designen av programmen som var utgångspunkten för examensarbetet. Dessutom beskrivs funktionen för att importera data från SMS 2003 till applikationen detaljerat.

3.1 Introduktion

Examensarbetet skall resultera i ett gränssnitt mellan SMS 2003 databasen och applikationens databas. Lösningen som har valts, och som även uppdragsgivaren ansåg vara lämpligast kan ses i figur 3.1.



Figur 3.1: Översiktsbild av programmet

Figur 3.1 visar en skiss över hur applikationen skall samverka med resten av systemet. Applikationen är ett webbgränssnitt som används hos kunden för att visa informationen som finns i databasen och för att skriva in information som inte kan hämtas från SMS databas. Uppgiften är att utveckla de delar som är gråmarkerade i figur 3.1, alltså att utveckla ett gränssnitt mellan SMS 2003 databasen och applikationens databas.

Möjlighet skall också ges för att senare lägga till andra delar för att på så sätt kunna importera data från andra datorsystem som exempelvis SUN. I figur 3.1 finns två andra exempel på datakällor, SMS 2.0 och textfil, som innehåller information om datorerna i en annan datormiljö. Detta kommer inte att ingå i detta examensarbete.

Mellan de två delarna valdes XML som datakälla eftersom Visual Basic .Net som examensarbetet utvecklades i har bra stöd för detta. En annan anledningen är för att underlätta senare implementation av andra delar för import av information från datorer som har exempelvis Unix som operativsystem. XML är också bra om informationen som genererats skall användas senare i andra program, eftersom stöd ofta finns för XML i andra programvaror.

3.2 Detaljerad beskrivning

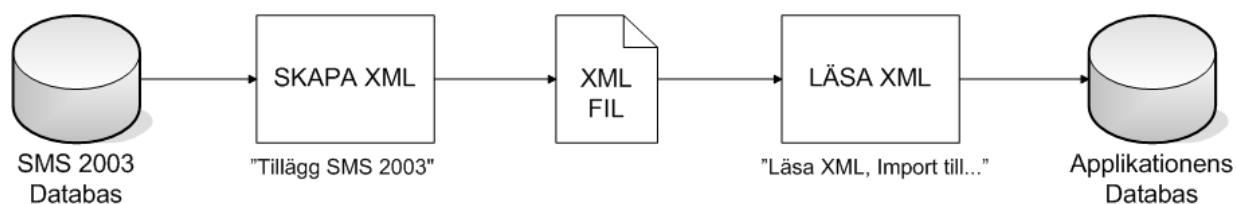
De steg som finns i utvecklingsfasen är:

- Funktion för att hämta information från SMS 2003 databasen och generera en XML-fil av informationen
- Läs från XML-filen och importera informationen till applikationens databas
- Utveckla funktionalitet som automatiskt utför import av informationen om datorerna

För att utföra import av information används en XML-fil som innehåller strukturerad information om varje dator.

För att importera informationen till applikationens databas finns alltså två steg som visas i figur 3.2:

1. Hämta information från SMS 2003 databasen och spara denna i en XML-fil.
2. Läs från XML-filen och importera denna till applikationens databas.



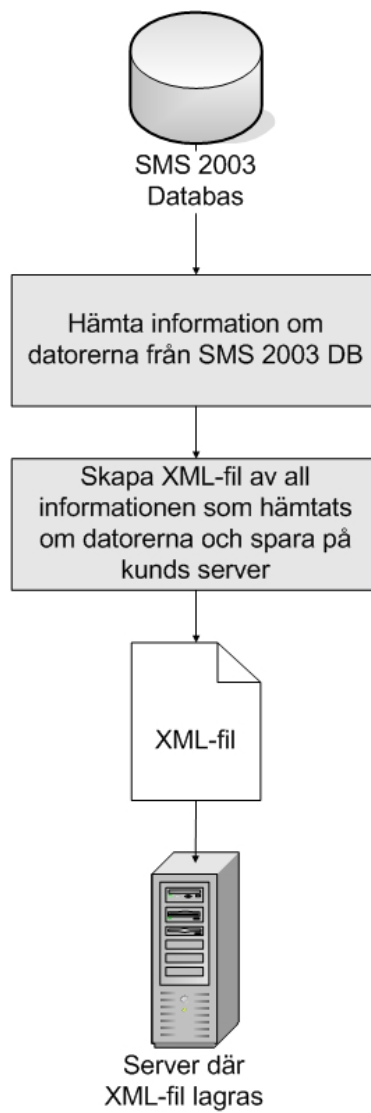
Figur 3.2: Steg för importera information

3.3 Tillägg SMS 2003

Denna del av programmet används för att hämta information från SMS databasen där information om datorerna finns. Det som sker är att via en databaskoppling hämtas all information som är av intresse och sedan skapas en XML-fil av det. XML-filen sparas på en server i samma nät som programmet exekveras. Filen skall sedan läsas in av den andra delen av programmet som importerar informationen till applikationens databas.

Figur 3.3 beskriver hur information hämtas från SMS 2003 databasen och sedan exporteras till en XML-fil. Informationen som samlas in från SMS databas är:

- “Applikationer” - Information rörande applikationens namn, leverantör, version, uppdatering, installationsdatum och språk
- “Datorinformation” - Information rörande bland annat datorns namn, modell, fabrikat och installationsdatum
- “Personer” - Information om signaturen/användarnamnet hos den anställda som senast använde datorn



Figur 3.3: Detaljerad bild över "Tillägg SMS 2003"

Anledningen till att enbart signaturen hos personerna hämtas är att SMS databasen endast innehåller information om vem som var inloggad senast. Detta är oftast tillfreds-tällande eftersom de flesta anställda hos kunden har egna datorer som bara de använder. Detaljerad information om den anställda importeras automatiskt från kundens intranät och finns därför redan i applikationen. Denna import ingår inte i examensarbetet. Möjlighet finns dock att utöka XML-filen med fler personer från andra system, exempelvis Microsoft Active Directory [12].

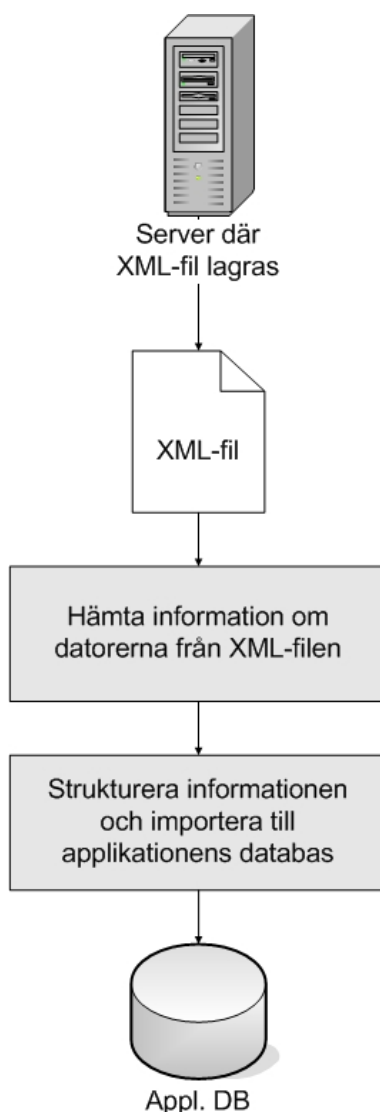
All information som hämtats från SMS databas sparas i XML-filen. Exempelvis inte bara de applikationer som kunden har licenser på redan, utan alla applikationer som finns installerade på varje dator i nätverket. TietoEnator har framfört att det i ett senare skede kan bli aktuellt att importera all information om vilka programvaror som finns hos datorerna i nätverket, därför valde uppdragsgivaren att all information skall sparas i XML-filen.

I detta skede behövs åtkomst till SMS databas, men inte till applikationens databas.

3.4 Läs XML och Import till applikationens databas

När informationen om alla datorerna i nätverket har sparats till XML-filen ska informationen importeras till applikationens databas.

Informationen struktureras efter strukturen i XML-filen för att sedan kunna importeras till applikationens databas på ett korrekt och effektivt sätt.



Figur 3.4: Detaljerad bild över "Läs XML och Import till applikationens databas"

Servern som XML-filen först sparas på behöver inte specifikt vara den server som importerar ifrån, programmet klarar av att läsa in XML-filer från vilken server i nätverket som helst, givet att behörighet finns. Bäst prestanda fås dock om filen finns lokalt på servern. I databasen skapas kopplingar mellan information gällande:

- Maskin/Dator och Person - Personen som var inloggad senast på datorn
- Maskin/Dator och Applikation - Applikationerna som det finns licens för och som finns på datorn där informationen uppdateras

I detta skede behövs åtkomst till applikationens databas, men inte till SMS databas.

3.5 Tjänst

Tanken med examensarbetet är att med automatik importera informationen från SMS databas till applikationens databas, då behövs ett tillvägagångssätt för att slippa exekvera programmen manuellt. Lösningen på detta är att utveckla en *tjänst*¹ som startas på datorn där programmen exekveras. Då kan programmen exempelvis implementeras för exekvering två gånger per dag vid givna tidpunkter. Uppdragsgivaren hade som önskemål att import skulle kunna ske två gånger varje dygn, en gång vid lunchtid och en gång på natten. Detta för att den ordinära trafiken i nätverket är som minst då, och för att slippa vänta 24 timmar innan uppdateringarna visas i applikationen.

Tjänsten startas automatiskt varje gång datorn startas och *tjänsten* skall sedan vara aktiv så länge den automatiska importen skall ske.

För att ha kontroll över om import har skett sker både en skrivning till "Loggboken" och till en textfil på datorn som har hand om importen.

Mer information om *tjänsten* finns i bilaga A.2.

¹Program som alltid exekveras, även om ingen användare är inloggad på datorn

4 Implementation

Kapitlet beskriver tillvägagångssättet för att utveckla programmen.

I bilaga A.3 finns fullständig källkod bifogad.

4.1 Introduktion

Efter att designen av funktionen för att importera information om datorerna till applikationen var klar, påbörjades implementationen. Implementation innefattade utveckling av programmen och databasens sparade procedurer som används vid import.

4.2 Utvecklingsmiljö

Programmet skrivs i Microsoft Visual Studio .Net 2003. Programspråket är Visual Basic .Net, där det finns inbyggt stöd för att koppla upp sig mot Microsoft SQL Server 2000 och stöd för hantering av XML-dokument. Detta medför att implementation för att hämta och hantera information från SMS databas och XML-filen blir enkel. Microsoft SQL Server 2000 användas både i applikationen och i SMS 2003. För implementationen av de sparade procedurerna i databasen används det medföljande verktyget "Enterprise Manager", som är en klient till databasen utvecklad av Microsoft.

4.3 Uppdelning av funktioner

Ett projekt skapades i utvecklingsmiljön där programmen utvecklades.

Projektet delades upp i fyra funktioner:

- MainClasses - Huvudklasser som används i båda programmen. Exempelvis strukturen hos varje dator och klasser som innehåller funktioner för att hämta information från en databas

- SQLToXML - Skapa XML-fil från informationen som hämtas från SMS 2003 databasen, även kallad "Tillägg SMS 2003"
- XMLToSQL - Importera informationen från XML-filen till applikationens databas, även kallad "Läsa XML och Importera till appl. databas"
- Service - *Tjänsten* som används för automatisk import

4.4 Systemkrav

För att programmen skall fungera krävs följande av datorn som programmen exekveras på:

- Microsoft Windows Server 2000, Windows Server 2003 eller annat operativsystem som har stöd för Microsoft Framework 1.1
- Microsoft System Management Server 2003
- Microsoft SQL Server 2000 - Hos den befintliga applikationen
- Microsoft Framework 1.1 - För exekvering av programmen
- Minst 512MB i internminne, mer är att rekommendera då exekveringstiden minskas desto större mängd internminne som finns

4.5 Kommentar till implementationen

Arbetet med uppgiften fortlöpte bra och två prototyper visades för uppdragsgivaren med lyckat resultat. Det fanns klara fördelar att visa prototyper och att samtidigt ha en nära dialog med uppdragsgivaren, detta för att snabbt få respons på funderingar och frågor gällande hur programmen skall fungera.

5 Resultat

Kapitlet beskriver vad examensarbetet resulterade i, vilka problem som uppstod och hur dessa löstes.

5.1 Introduktion

Målet med examensarbetet var att automatisera största delen av inmatningen av information om datorer till en applikation hos ett företag. Mycket av informationen som fanns i applikationen kunde hämtas från SMS databas, vilket gjorde att SMS användes som datakälla.

Stöd för andra datakällor skulle också finnas, dock med viss modifikation eller tillägg. Detta för att underlätta vid import av andra systems datorer och licenser.

5.2 Beskrivning

I samråd med uppdragsgivaren valdes en lösning som bestod av tre delar:

- Hämta information om datorerna från SMS databas och spara ner den till en XML-fil för vidare bearbetning
- Hämta information från XML-fil och importera den till applikationens databas
- Skapa en tjänst för automatisk exekvering

Detta resulterade i att det är enkelt att lägga till andra datakällor då utveckling endast behöver ske av ett program som hämtar information om datorerna och sparar det i det givna XML-formatet.

Utvecklingen för att få fram en prototyp gick snabbt och flera funktioner implementerades till varje program. Exempelvis var det inte tänkt att attribut skulle kunna användas till programmen för att ändra exekveringen. Nu kan vissa egenskaper ändras vid exekveringen,

se mer information i användarmanualen i bilaga A.2.

Målet med att utveckla en produkt som automatiskt hämtar information om datorerna i ett nätverk och sedan importerar den till den befintliga applikationens databas, har uppnåtts med lyckat resultat. Provkörning har utförts i kundens nätverk.

Dock skedde provkörningen mot SMS 2.0, eftersom kund inte använde SMS 2003 vid provkörningstillfället. Detta medförde endast vissa förändringar i frågorna som ställdes mot SMS databasen. Inget program behövde kompileras om för att få programmen att fungera med SMS 2.0, vilket får anses vara en bra egenskap hos programmen.

Uppgiften att skapa en *tjänst* var inte med från början, men eftersom import skall utföras automatiskt implementerades denna. Det visade sig vara en bra lösning på problemet eftersom det även medförde mindre arbete vid test och exekvering i kundens miljö.

Systemdokumentation och användarmanual, se bilaga A.1 och A.2, skrevs med tanke på att underlätta för uppdragsgivaren att exekvera, vidareutveckla och vid eventuell felsökning av programmen.

5.3 Problem

Under examensarbetets gång uppstod en del problem.

Försök till att hålla nere antalet frågor mot SMS gjordes av prestandaskäl, då databasen vid många anrop och uppkopplingar både får dålig prestanda i programmen och i databasservern. Lösningen på det var att enbart exekvera databasfrågorna en gång och då spara strukturen i minnet medan programmet exekveras. Databasen är endast ansluten då frågor exekveras i databasen.

Problemet som då uppstod var att vid exekvering på kundens nätverk krävdes en hel del internminne på datorn där programmen exekverades. Detta beror delvis på att mängden information som hämtas upp från SMS databas växer mycket med varje ny dator som finns i nätverket. Exekveringen som utfördes hämtade i storleksordningen information om 2000 datorer och 600000 installerade program, det vill säga att varje dator har i snitt 300 pro-

gram installerade. Detta medförde att XML-filen som sparades blev ungefär 150MB stor, som sedan läsas in för att importeras till applikationen. Exekveringen tog totalt tre timmar, då den utfördes på en dator med 512MB i internminne. Utökas interminnet minskas exekveringstid.

De sparade procedurerna (SP) som implementerades för att hantera importen till applikationens databas ändrades i slutfasen av implementationen. Detta eftersom SP först inte kontrollerade att licens fanns för programmen som importerades. Nu kontrolleras först att licens finns till programmen i applikationens databas innan beroenden mellan dator och programvaror skapas i applikationens databas. SP utvecklades från befintliga sparade procedurer som redan fanns för den manuella inmatningen, men anpassades så att de fungerade med de utvecklade programmen.

6 Slutsats

Kapitlet drar slutsatser från examensarbetet, beskriver framtida och andra användningsområden för programmen och applikationen.

6.1 Introduktion

Programmen ska på sikt minska tidsåtgången för att skriva in information till applikationen om vilka programvaror och datorer som finns i ett nätverk hos en kund. Detta medför att kostnaden blir mindre och informationen blir mindre känslig för felinmatning. Kontroll fås även över vilka programvaror som används av flest användare och kan då prioritera och effektivisera installation och underhåll av de programvarorna. Hantering av licenser är ett stort, betydelsefullt arbete då licenser skall köpas in och programvaror och operativsystem skall uppdateras.

6.2 Beskrivning

I början av examensarbetet upprättades en tidsplan som skulle förhindra misslyckande med arbetet på grund av att tiden inte räckte till. Denna tidsuppskattning följdes bra under hela examensarbetet vilket gjorde att mängden utfört arbete var konstant under hela arbetet. Examensarbetaren var redan innan examensarbetet påbörjades införstådd i vad uppgiften behandlade, vilket medförde att design och implementation av produkten påbörjades tidigt. Det som skulle kunna utförts annorlunda är en tidig test med större datamängd. De första testerna gjordes med ett 100-tal datorer. När sedan ett test med 2000 datorer utfördes uppstod problem med att programmen krävde mer av datorn som exekverade programmen än beräknat. Inga krav fanns dock på dimensionering från uppdragsgivaren.

En tanke som har funnits är att SQL Servern skulle ha använts ytterligare. Exempel på detta hade varit att skapa XML-filen direkt i databasservern och att importera den direkt, vilket hade förbättrat prestandan. Dock är stödet för XML i Microsoft SQL Server 2000

något begränsat och användningsområdena begränsas ytterligare vilket gjorde att lösningen som examensarbetet har valdes.

De begränsningar som produkten har är främst att den måste exekveras på en dator som har Windows som operativsystem och Framework 1.1 installerat. Denna begränsning togs ingen hänsyn till eftersom det är tänkt att produkten skall exekveras i ett sådant system. Produkten skulle kunna utökas med flera tillägg för andra datakällor, vilket skulle innebära att kunden kan använda produkten även för andra operativsystem än Windows. Avgränsningen till en datakälla medförde att produkten är beroende av SMS, men eftersom implementationen av ett nytt tillägg är förhållandevis enkel, kan produkten utökas efter kundens operativsystem, programvaror och nätverk.

6.3 Ytterligare arbete

För att produkten ska bli mer generell bör gränssnitt mot fler datakällor implementeras. Detta skulle kunna vara ett lämpligt framtida projekt då produkten skulle kunna hantera andra typer av hårdvara än vad som omfattades i detta examensarbete. Andra kunder skulle då kunna införa den befintliga applikationen för licenshanteringen även om de inte använder SMS. Exempel på datakälla är stöd för system som använder SUN Solaris som operativsystem, vilket används av många kunder till uppdragsgivaren.

6.4 Framtida utveckling

Programmen skulle kunna användas till andra applikationer och system då val av vilken information som skall vara med kan ske. Eftersom en XML-fil används skulle denna fil kunna exporteras till andra applikationer som skulle kunna hantera informationen om datorerna på andra sätt än det som den befintliga applikationen gör idag. Exempel på information skulle kunna vara att lista alla applikationer som finns installerade hos en kund. I framtiden skulle applikationen även kunna larma då en licens håller på att gå ut. Dessutom skulle applikationen kunna visa vilka program som kan köpas in centralt eftersom

många användare använder dem och har köpt in dem lokalt på avdelningarna. Detta blir på sikt lönsamt då priset på licensen centralt är ett bättre pris och då programvarorna kan fjärrinstalleras direkt på en användares dator, givet att licens finns till programmet. Att se vilka datorer som inte uppfyller företagets policy skulle kunna exponeras, dock är detta ett känsligt ämne då misstycke kan uppstå hos de anställda om sådana kontroller sker.

6.5 Sammanfattning av arbetet

Examensarbetet resulterade i en produkt som är tänkt att minska inmatningen och minska hanteringen av information genom att automatiskt importera informationen från en datakälla.

Programmen är ännu inte driftsatta, men detta skall ske inom ett par månader. Risken för att mata in felaktig information om datorerna kommer att minska betydligt och teknikerna som annars skulle uppdaterat applikationen manuellt, slipper då detta och sparar därför mycket tid.

Referenser

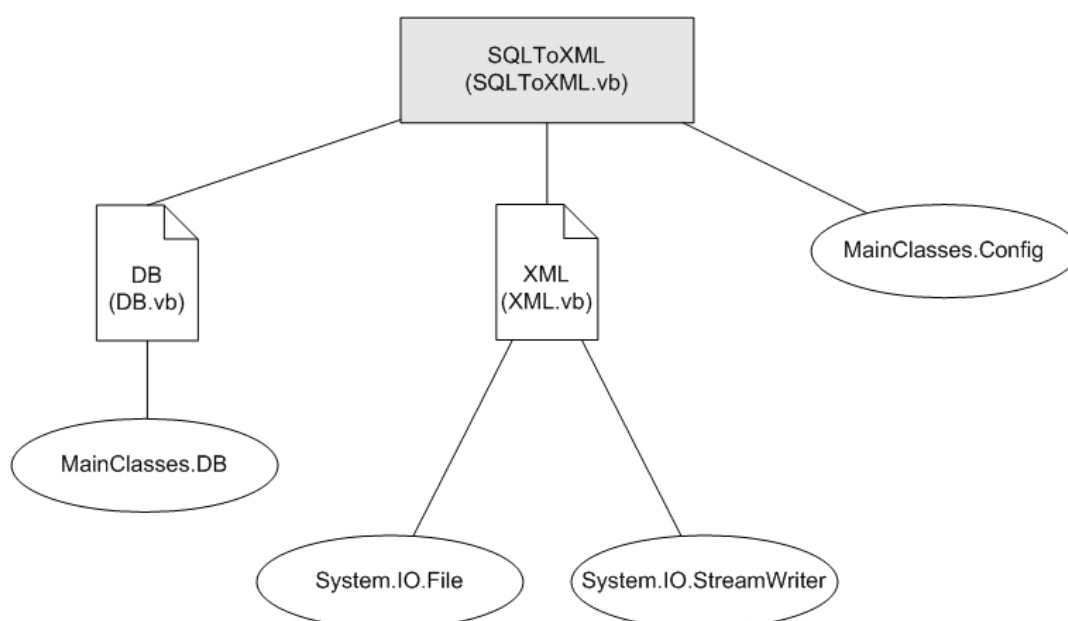
- [1] <http://www.microsoft.com/sql/evaluation/overview/default.asp> (2004-02-17)
- [2] <http://www.microsoft.com/windows/default.mspx> (2004-05-19)
- [3] <http://www.microsoft.com/smsserver/evaluation/overview/default.asp> (2004-02-17)
- [4] <http://msdn.microsoft.com/vstudio/productinfo/> (2004-02-17)
- [5] <http://msdn.microsoft.com/vbasic/> (2004-05-19)
- [6] <http://www.tietoenator.com/default.asp?Path=1,92,1640> (2004-02-25)
- [7] <http://www.tietoenator.com/default.asp?Path=1,92,217> (2004-02-25)
- [8] <http://www.tietoenator.com/binary.asp?guid=928CE919-7408-434D-BB2A-83279605E778&field=image> (2004-02-25)
- [9] <http://www.tietoenator.com/default.asp?Path=1,92,1637,1674> (2004-02-25)
- [10] <http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032247071>
(2004-05-13)
- [11] <http://msdn.microsoft.com/asp/> (2004-02-25)
- [12] <http://www.microsoft.com/windows2000/technologies/directory/AD/default.asp>
(2004-04-23)

A Bilagor

A.1 Systemdokumentation

Implementation av “Tillägg SMS 2003”

För att hämta informationen om datorerna från SMS databas skapades program som kallades SQLToXML (Läses SQL-To-XML).



Figur A.1: Fildiagram för implementationen av “Tillägg SMS 2003”

Programmet har hand om att skapa objekten för att hantera inläsning av information om datorerna från databasen och skapandet av XML-filen. Observera att XML-filen skrivs över varje gång programmet exekveras.

Klasserna som användes är:

- DB - Används för databaskopplingen och hämtning av information från SMS-databasen
- XML - Skriver informationen om datorerna som hämtats till en XML-fil

Klasserna använder i sin tur andra underliggande klasser för hantering av databaskoppling och skrivning av XML-fil. Detta illustreras i figur A.1.

Konfiguration

Konfiguration hämtas genom MainClasses.Config-klassen, som i sin tur hämtar alla inställningar från SQLToXML.exe.config. Filen är uppbyggd i XML och innehåller ett antal element och värden som kan förändras för att ändra hämtning av information från SMS 2003 databasen och skrivning av XML-filen.

Lista med element och konfigurationsalternativ finns i tabell A.3.

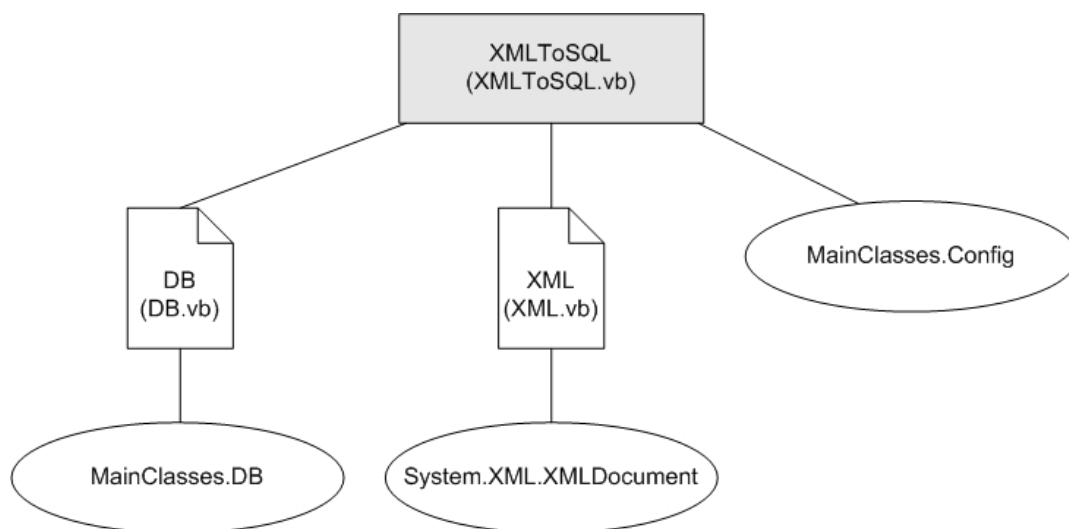
Attribut till programmet

Vid exekvering av programmet kan olika attribut användas för att exempelvis inte skriva applikationerna till XML-filerna. Programmet har fördefinierade attribut som finns och kan ändras enligt tabell A.2, som innehåller de namn- och värdepar som används. Dessa ligger till grund för en normal exekvering. Vid enskilda exekveringar med speciella attribut kan attributen i tabell A.2 användas.

Implementation av “Läsa XML och Importera till appl. databas”

När XML-filen har genererats ska den importeras till applikationens databas. Detta utförs genom att via “XMLToSQL”-programmet (Läses XML-To-SQL) importeras alla datorerna och dess information om applikationer och personer. Programmet startas via XMLToSQL.vb och innehåller ytterligare två egenutvecklade klasser:

- DB - Används för databaskopplingen och skrivning av information till applikationens databas
- XML - Hämtar informationen om datorerna som finns i XML-filen



Figur A.2: Fildiagram för impl. av “Läs XML och Importera till appl. databas”

Programmet startas från XMLToSQL, som har hand om inläsning av de alternativ som finns. Det har även hand om att skapa objekten för att hantera inläsning av information om datorerna från XML-filen och import till applikationens databas. Klasserna som användes är:

- DB - Används för databaskopplingen och import av information till applikationens databas
- XML - Läser informationen om datorerna från XML-filen

Klasserna använder i sin tur underliggande klasser som beskrivs i figur A.2.

Konfiguration

Konfiguration hämtas genom MainClasses.Config-klassen, som i sin tur hämtar alla inställningar från XMLToSQL.exe.config. Filen är uppbyggt i XML och innehåller ett antal element och värden som kan förändras för att ändra hämtning av information från XML-filen och skrivning av applikationens databas.

Lista med element och konfigurationsalternativ finns i tabell A.5.

Attribut till programmet

Vid exekvering av programmet kan olika attribut för att exempelvis inte importera applikationerna till applikationens databas användas. Programmet har fördefinierade attribut som finns och kan ändras enligt tabell A.3, som innehåller de namn- och värdepar som används. Dessa ligger till grund för en normal exekvering. Vid enskilda exekveringar med speciella attribut kan attributen i tabell A.4 användas.

XML-filen

XML-filen har ett speciellt format som skall följas för att import av informationen om datorerna skall fungera på ett korrekt sätt.

Sker ändringar i XML-filen så bör även programmen anpassas för det nya utseendet.

Filen har detta format:

```
<?xml version="1.0" encoding="UTF-8"?>
<maskiner>
  <maskin>
    <namn>Datornamn</namn>
    <fabrikat>Datorns fabrikat</fabrikat>
    <maskinID>Maskinens ID-nummer i SMS databas</maskinID>
    <modell>Datorns modell</modell>
    <installationsdatum>Datumet då datorn installerades</installationsdatum>
    <serienummer>Datorns serienummer</serienummer>
    <applikationer>
      <applikation>
        <version>Applikationens version</version>
        <beskrivning>Applikationens namn/beskrivning</beskrivning>
        <servicepack>Uppdatering till applikationen som är installerad</servicepack>
        <installationsdatum>Datumet då applikationen installerades</installationsdatum>
        <språk>Applikationens språk</språk>
        <leverantör>Applikationens tillverkare/leverantör</leverantör>
      </applikation>
      <applikation>
        ...
      </applikation>
    </applikationer>
```



```

<personer>
  <person>
    <sign>Användarens signatur som använder datorn</sign>
  </person>
  <person>
    ...
  </person>
</personer>
</maskin>
<maskin>
  ...
</maskin>
</maskiner>

```

Varje element har ett värde som hämtats från datakällan. Värdena i varje element i beskrivningen ovan anger en mer detaljerad beskrivning om vad varje element innehåller för information. Om fler fält skall läggas till, utförs detta genom att ändra frågan som ställs mot databasen för att hämta information om datorerna. Vid ändring av ett elements namn ändras namnet på attributet i samma fråga. Kom då ihåg att kompilering av programmet som importerar XML-filen till applikationens databas krävs, eftersom de är fasta där. Normalt skall inte namnen ändras på elementen, men möjligheten finns ändå.

Det som inte går att ändra är uppbyggnaden av varje dator, dvs. “<maskiner>”, “<maskin>”, “<applikationer>”, “<applikation>”, “<personer>” och “<person>” (och deras sluttaggar). Dessa är fasta för att koden till inläsningen av XML-filen skall bli lättförståelig.

Detta påverkar inte prestandan eller examensarbetets syfte.

A.2 Användarmanual

Introduktion

Produkten används för att automatiskt importera information om datorerna i ett nätverk till den befintliga applikationens databas. Vanligtvis sker en import 2 gånger varje dygn, en gång klockan 12:15 och en gång klockan 00:15. En tjänst är då installerad på en server för att exekvera programmen. Servern har tillgång till SMS och applikationens databas.

Tjänst

För att automatisk import av information om datorerna skall ske måste en tjänst installeras på servern som hanterar importeringen. Tjänsten heter "ImportSMS.exe" och filerna finns i Service-mappen.

Installation av tjänst

För att installera detta program som en tjänst skriver administratören i kommandoprompten: `installutil Service.exe` givet att administratören står i den mappen där `Service.exe` finns. Då installeras tjänsten och den startas sedan med automatik även efter omstart och ingen behöver vara inloggad på servern.

Avinstallation av tjänst

För att avinstallera tjänsten så skriver administratören i kommandoprompten: `installutil /U Service.exe` givet att administratören står i den mapp där `Service.exe` finns. Då stoppas och avinstalleras tjänsten från systemet. För att exekvera tjänsten igen måste den ominstalleras.

Starta tjänst

För att starta tjänsten, som av någon anledning stannat eller stoppats, skriver användaren i kommandoprompten: `net start service` givet att tjänsten redan är installerad och inte exekveras.

Stoppa tjänst

För att stanna tjänsten då ingen import skall ske skriver användaren i kommandoprompten: `net stop service`, givet att tjänsten redan är installerad och exekveras.

Loggning av tjänst

För att ha kontroll över när en import har skett skrivs information om detta både till datorns loggbok (under "Programlogg") och till `C:\SMS2003Import.log`.

Konfiguration

Konfiguration hämtar alla inställningar från `Service.exe.config`. Filen är uppbyggt i XML och innehåller ett antal element och värden som kan förändras för att ändra den automatiska exekveringen av programmen.

Filen ser ut på följande sätt:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="namn" value="värde" />
  </appSettings>
</configuration>
```

Lista med alla namn- och värdepar som finns med enligt ovanstående utseende:

Namn	Värde	Förklaring
sqltoxml	SQLToXML.exe	Sökväg till SQLToXML ("Tillägg SMS 2003")
xmltoxml	XMLToSQL.exe	Sökväg till XMLToSQL ("Läsa XML och Import...")
xmlfile	SMS2003Import.xml	Sökväg till XML-filen där all information om datorerna sparas och hämtas vid import
logfile	SMS2003Import.log	Sökväg till loggfilen som sparar information om import som utförts
eventlog	SMS2003Import	Namnet på källan i loggboken

Tabell A.1: Lista med standardkonfigurationsalternativ för tjänst

Tillägg SMS 2003

Programmet sparar ner information om datorerna i ett nätverk. Informationen hämtas från SMS 2003 databasen och sparar sedan ner den till en XML-fil som används vid importeringen till applikationens databas. Hur lång tid exekveringen tar är beroende av hur mycket information det finns i SMS databas och hur mycket internminne datorn har, ju mer internminne som finns, desto snabbare blir exekveringen.

Tiden för skrivning till XML-filen är försumbar.

Attribut

Vid exekvering av programmet kan olika attribut användas för att exempelvis inte skriva applikationerna till XML-filerna. Programmet har fördefinierade attribut som finns och kan ändras enligt tabell A.2. Dessa ligger till grund för en normal exekvering. Vid enskild exekvering med speciella attribut kan denna lista med attribut användas:

-l, -d, -v	mboxSkriv ut logg på skärmen, kan användas för att ge information om exempelvis hur många maskiner som har lästs in eller om fel uppstod vid inläsningen
-a	Inga applikationer som datorerna har installerade skrivs till XML-filen, kan användas då nya maskiner och vilka personer som använder den skall läggas till
-p	Inga personer som använder datorn skrivs till XML-filen
-os	Inga operativsystem som varje maskin använder sig av skrivs till XML-filen
-h, -help, /?	Visa hjälp om programmet med giltiga attribut
output.xml	Spara informationen om datorerna till en XML-fil annan än den som anges i konfigurations filen. (Här heter den output.xml)

Tabell A.2: Lista med attribut för "Tillägg SMS 2003"

Exekvering

Programmet kan exekveras via kommandoprompten eller så sker exekveringen automatiskt med medföljande tjänst. Några exempel på exekveringar av programmet via kommandoprompten är:

- “SQLToXML.exe” - Normal exekvering, alla attribut som finns i konfigurationen används och XML-fil skrivs
- “SQLToXML.exe -h” - Hjälpen visas till programmet
- “SQLToXML.exe -l >> SQLToXML.log” - Normal exekvering, men logginformation läggs till i SQLToXML.log
- “SQLToXML.exe -a -o -p” - Skriver bara datorerna och dess information (datornamn, fabrikat osv.) till XML-filen

Konfiguration

Konfiguration hämtar alla inställningar från SQLToXML.exe.config. Filen är uppbyggt i XML och innehåller ett antal element och värden som kan förändras för att ändra hämtning av information från SMS 2003 databasen och skrivning av XML-filen.

Filen ser ut på följande sätt:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="namn" value="värde" />
  </appSettings>
</configuration>
```

Lista med alla namn- och värdepar som finns med enligt ovanstående utseende:

Namn	Värde	Förklaring
log	false	Logg skrivs inte ut
apps	true	Information om applikationer som finns installerade hos varje dator skrivs till XML-filen
os	true	Information om operativsystem som varje dator använder skrivs till XML-filen
pers	true	Information om personer som använder datorn skrivs till XML-filen
Connection	...	Databaskopplingen till SMS 2003 databasen
xmlfile	text.xml	Filnamnet på XML-filen som innehåller all information om varje dator
SQLMachines	EXEC xMaskin_Sel	Fråga mot databasen för att hämta alla datorer och deras specifika information
SQLApps	EXEC xAppl_Sel	Fråga mot databasen för att hämta alla applikationer som varje dator har installerade och information om exempelvis vilken version de har
SQLOS	EXEC xOS_Sel	Fråga mot databasen för att hämta vilket operativsystem datorerna använder
SQLPers	EXEC xPers_Sel	Fråga mot databasen för att hämta alla personer och vilka dator de använder
help	...	Hjälp texten som skrivs ut när användaren vill se hjälp om programmet

Tabell A.3: Lista med standardkonfigurationsalternativ för "SQLToXML"

För anslutning mot SMS 2003 databasen krävs en giltig användare som har läsrättigheter i tabellerna. Enklast och säkrast är att skapa en ny användare i databasen som endast har hand om importeringen och tilldela den användaren databasrollen "db_datareader".

Import till applikationens databas

Programmet importerar informationen om datorerna från en XML-fil till applikationens databas. Tiden det tar att importera informationen är beroende av hur mycket information det finns i XML-filen och hur mycket internminne som finns på datorn. Är det mycket information som skall importeras behövs även mycket internminne. När importeringen är klar, frigörs även internminnet som använts, det är alltså bara då exekvering sker som mycket ledigt interminne krävs.

Attribut

Vid exekvering av programmet kan olika attribut användas för att exempelvis inte importera applikationerna till applikationens databas. Programmet har fördefinierade attribut som finns och kan ändras enligt namn- och värdeparen i tabell A.4. Dessa ligger till grund för en normal exekvering. Vid enskild exekvering med speciella attribut kan denna lista med attribut användas:

-l, -d, -v	Skriv ut logg på skärmen, kan användas för att ge information om exempelvis hur många maskiner som har lästs in eller om fel uppstod vid inläsningen
-a	Inga applikationer som datorerna har installerade importeras till applikationens databas, kan användas då import ska ske för nya maskiner och vilka personer som använder den
-p	Inga personer som använder datorn importeras till applikationens databas
-h, --help, /?	Visa hjälp om programmet med giltiga attribut
input.xml	Hämta informationen om datorerna från en XML-fil annan än den som anges i konfigurations filen. (Här heter den input.xml)

Tabell A.4: Lista med attribut för "Läsa XML och Importera till appl. databas"

Exekvering av programmet

Programmet kan exekveras via kommandopromten eller så sker exekveringen automatiskt

med medföljande tjänst.

Några exempel på exekveringar av programmet via kommandoprompten är:

- “XMLToSQL.exe” - Normal exekvering, alla attribut som finns i konfigurationen används och import till applikationens databas sker
- “XMLToSQL.exe -h” - Hjälpen visas till programmet
- “XMLToSQL.exe -l >> XMLToSQL.log” - Normal exekvering, men loginformation sparas i XMLToSQL.log
- “XMLToSQL.exe -a -p” - Importerar bara datorerna och dess information (datornamn, fabrikat osv.) till applikationens databas

Konfiguration

Standardkonfiguration som programmet använder hämtas via MainClasses.Config-klassen. Den hämtar i sin tur informationen från XMLToSQL.exe.config filen som är uppbyggt i XML-format. Filen innehåller ett antal element och attribut med värden.

Filen är uppbyggd enligt följande struktur:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="namn" value="värde" />
  </appSettings>
</configuration>
```

Följande namn- och värdepar används:

Namn	Värde	Förklaring
log	false	Logg skrivs inte ut
apps	true	Information om applikationer och operativsystem som finns i XML-filen importerar till applikationens databas
pers	true	Information om personer som använder datorn som finns i XML-filen importerar till applikationens databas
Connection	...	Databaskopplingen för anslutning till applikationens databas
xmlfile	text.xml	Filnamnet på XML-filen som innehåller all information om varje dator
SQLMachine	Admin_Maskin_Save	Sparad procedur för att spara datorns information till applikationens databas
SQLApp	xAdmin_ProgramAnv_Save	Sparad procedur för att spara information om datorernas applikationer och operativsystem till applikationens databas
SQLUser	xAdmin_Koppla_Save	Sparad procedur för att spara information om vilka användare som använder varje dator som finns i XML-filen
help	...	Hjälp texten som skrivs ut när användaren vill se hjälp om programmet

Tabell A.5: Lista med standardkonfigurationsalternativ för "XMLToSQL"

Det är viktigt att skriv- och exekveringsrättigheter finns för databasens användare och sparade procedurer som finns specificerat i konfigurationen.

A.3 Källkod

Service - Tjänst

Service.vb - Tjänst för automatisk import

MainClasses - Huvudklasser

Config.vb - Läsa från konfigurationsfil

DB.vb - Databaskoppling

DBParameter.vb - Databasparameter

Maskin.vb - Datorinformation

SQLToXML - "Tillägg SMS 2003"

SQLToXML.vb - Huvudprogram

DB.vb - Läsa från SMS databas

XML.vb - Skriva till XML-fil

XMLToSQL - "Läsa XML och Import..."

XMLToSQL.vb - Huvudprogram

DB.vb - Import till applikationens databas

XML.vb - Läsa från XML-fil

Stored Procedures - Sparade procedurer i applikationens databas

xAdmin_Maskin_Save - Spara information om dator

xAdmin_Koppla_Save - Skapa beroende mellan dator-person

xAdmin_ProgramAnv_Save - Skapa beroende mellan dator-applikation