

Automatiserat installationskript för Sopimus Pro

Mats Åkerblom

Examensarbete för Tradenom-examen
Utbildningsprogrammet i informationsbehandling
Raseborg 2013



EXAMENSARBETE

Författare: Mats Åkerblom

Utbildningsprogram och ort: Informationsbehandling, Raseborg

Handledare: Klaus Hansen

Titel: Automatiserat installationsskript för Sopimus Pro

Datum: 17.11.2013

Sidantal: 45

Bilagor: 1

Abstrakt

Uppgiften för arbetet var att utveckla ett automatiserat installationsskript som automatiskt och utan användarens inblandning installerar Sopimus Pro-systemet på en dator med Windows som operativsystem. Sopimus Pro är flaggskeppsprodukten för Rediteq Ab, som utvecklar utbildningsmjukvara för utbildare i Finland. Det färdiga skriptet installerar alla nödvändiga komponenter i tur och ordning samt gör alla nödvändiga konfigurationer som krävs för att systemet ska fungera.

Detta arbete behandlar hela installationsprocessen från början till slut, samt grunderna för hur en automatisk installation går till med hjälp av användning av s.k. svarsfiler. Skriptet använder sig av batchfiler som körs med CMD eller komandoprompten. Inom installationen används även en automatiseringsapplikation vid namnet Autolt.

Som anställd hos Rediteq Ab har jag redan haft användning av skriptet inom mina egna arbetsuppgifter. Slutresultatet är ett automatiserat skript som klarar av att installera Sopimus Pro-systemet snabbare än en installation som görs manuellt.

Språk: Svenska

Nyckelord: Automatiserad, installation, Sopimus Pro, CMD, batchfiler

BACHELOR'S THESIS

Author: Mats Åkerblom

Degree Programme: Business Information Technology, Raseborg

Supervisor: Klaus Hansen

Title: An automated Install Script for Sopimus Pro / Automatiserat
installationsskript för Sopimus Pro

Date: 17 November 2013

Pages: 45

Appendices: 1

Summary

The task for this thesis was to develop an automated and unattended install script for installation of the Sopimus Pro-system on a computer with Windows as its operating system. Sopimus Pro is the flagship product for Rediteq Ltd, a software development company focusing on solutions for education providers in Finland. The complete script installs all the components required for Sopimus Pro to run.

This thesis deals with the whole installation process from start to finish. It also covers the basics of an automated install and the usage of answer files. The script uses batch files which are run with CMD or the Command Prompt. An automation application named Autolt was used to solve various tasks during the installation.

As an employee at Rediteq Ltd I have already had the chance to use my script in my work tasks. The end result is an automated script which is capable of installing the Sopimus Pro-system by itself faster than a user would do manually.

Language: Swedish

Keywords: automated, install, Sopimus Pro, CMD, batch file

Innehållsförteckning

1	Introduktion.....	1
1.1	Uppdragsgivare	1
1.2	Om företaget.....	1
1.3	Bakgrund till uppdraget	2
1.4	Syfte och målsättning	2
1.5	Målgrupp.....	3
1.6	Avgränsning	3
2	Definitioner.....	4
2.1	CMD	4
2.2	Batchfil	4
3	Val av verktyg	5
3.1	CMD Command Shell	5
3.2	Notepad++	6
3.3	Replace Text.....	7
3.4	Autolt	7
4	Installationsprocessen	8
4.1	Firebird och FlameRobin	8
4.2	Applikationsservern och klienterna	9
4.3	Säkerhetskopieraren.....	11
5	Utförandet av en tyst och obevakad installation	11
5.1	Installationstyper	12
5.2	Skapandet av en svarsfil.....	13
5.3	Användning av svarsfilen	14
6	Den automatiserade installationen	15
6.1	Install.bat	15
6.1.1	Skapandet av en ny mapp	15
6.2	Firebird.....	16
6.2.1	Svarsfilen för Firebird	16
6.2.2	Wait-kommandot	17
6.3	FlameRobin	18
6.3.1	Konfigurering av FlameRobin med Autolt.....	19
6.3.2	GSEC.....	23
6.4	Kopiering med CMD.....	24
6.5	Sopimus Pro applikationsserver	25
6.5.1	Problem med svarsfilen	25

6.5.2	Pausning av skriptet.....	26
6.5.3	serverInstallTaskKill	27
6.5.4	Uppdatering av applikationsservern	28
6.6	Klienterna.....	31
6.7	Time To Backup.....	33
6.7.1	Konfigurering av Time To Backup.....	33
7	Användning av skriptet.....	35
7.1	Igång läggning av skriptet	36
7.2	Replace Text.....	37
8	Testning	38
9	Slutdiskussion	40
10	Avslutning	41
	Källförteckning.....	42
	Figurförteckning	44
	Kodförteckning	45
	Bilagor.....	46

1 Introduktion

Det här arbetet är skrivet som en del av ett projekt gjort för Rediteq Ab, ett finländskt mjukvaruföretag. Jag började arbeta hos Rediteq på hösten 2012 anställd som praktikant. Efter den fem månader långa praktikperioden anställdes jag som deltidsanställd. Genast i början blev jag introducerad till Rediteqs huvudvara, Sopimus Pro, som består av olika komponenter, och är en väsentlig del av det här arbetet. Jag fick först bekanta mig med användargränssnittet samt installationen av användargränssnitten för Sopimus Pro. Småningom fick jag bekanta mig med de olika komponenterna samt deras installation, så att jag slutligen kunde installera hela Sopimus Pro på en dator.

1.1 Uppdragsgivare

Uppdragsgivare för projektet är Petri Huttunen, verkställande direktör för Rediteq Ab. Efter praktikperioden på Rediteq hade Huttunen en bra bild av var mina kunskaper låg och vad som skulle passa som ett lämpligt examensarbete för mig så att jag kunde lära mig något nytt under projektets gång utan att projektet skulle vara för svårt eller en för stor bit för mig att klara av på egen hand.

1.2 Om företaget

Tietotoimi H&H grundades år 1991 i Åbo men existerar nuförtiden under namnet Rediteq Ab, namnbytet skedde år 2008. Rediteq Ab är ett mjukvaruföretag specialiserat på utveckling av produkter för utbildningssyften i Finland. Den äldsta och mest utvecklade produkten Sopimus Pro (Sopro) används för administrering av läroavtal. Idag använder över 90 % av Finlands läroavtalskontor Sopro för att skapa och hantera sina läroavtal tillsammans med studerande och olika utbildningsarrangörer. Rediteq har under många år samarbetat med dessa läroavtalskontor och många av dem har beställt skräddarsydda lösningar för att kunna sköta dagliga arbetsuppgifter på ett sätt som passar dem bäst. (Rediteq)

Under de senaste åren har Rediteq även börjat satsa på andra former av utbildning. Den nyaste produkten Study@Campus^{Pro} är en inlärningsplattform (kan jämföras med t.ex. Moodle) som kan användas av så gott som allt mellan lågstadielklasser ända fram till

akademier och yrkeshögskolor. Study@Campus^{Pro} är i skrivande stund under pilotanvändning vid yrkesinstitutet Practicum.

Vid skrivande stund sysselsätter Rediteq 12 personer med arbetsuppgifter varierande från programmering, utveckling, kundsupport, marknadsföring och försäljning.

1.3 Bakgrund till uppdraget

Programmet Sopimus Pro består av en hel del olika komponenter och konfigurationer som bör göras när programmet installeras på en dator, antingen hos en kund eller på någon av bolagets egna datorer. Huttunen kom med ett förslag där han berättade att det skulle vara bra ifall hela installationsprocessen kunde automatiseras så att programmet skulle installeras genom att trycka på så lite tangenter som möjligt och utan användarens inblandning.

1.4 Syfte och målsättning

Då och då behöver Sopro installeras på en dator. Om installationen görs på en dator som ägs av en kund, finns det varierande faktorer som man bör ta i beaktande, så som konfigureringar med nätverk. Installationen av själva Sopro däremot är alltid den samma, och därför är en automatiserad installation lämplig. Genom att automatisera installationen blir processen samtidigt snabbare, och på så sätt skapar tid för annat under tiden installationen körs. Det färdiga skriptet ska göra installationen av Sopro snabb och smidig. Samtidigt kommer jag att lära mig utan och innan hur hela installationen går till samt i vilken ordningsföljd.

Målet med projektet är att skapa ett installationsskript som "tyst" och automatiskt sköter installationen av Sopro med så lite inblandning från användaren som möjligt. Med en tyst installation menar man att användaren inte ser några installationsfönster och användaren frågas inte heller detaljer om installationen så som destinationsmapp eller om man vill ha en genväg till programmet på skrivbordet.

1.5 Målgrupp

Målgruppen för det skriftliga arbetet är personer intresserade av automatiserade skript. Möjligheterna för vad som kan göras med ett skript är otroligt många. Skript kan användas för att sköta dagliga rutinmässiga saker snabbt och enkelt. Texten är skriven för personer intresserade av datorer och IT och jag har förutsatt att läsaren har baskunskaper om datorer och vanliga datortermer. I skrivande stund kommer de anställda hos Rediteq tyvärr inte att ha värst mycket nytta av denna skriftliga del av arbetet, eftersom majoriteten av de anställda inte talar svenska. För nya svenskatalande anställda kan den skriftliga delen vara ett bra sätt att lära sig grunderna om hur Sopro installationen går till.

Målgruppen för själva installationsskriptet och den tekniska delen av arbetet är de anställda på Rediteq som sköter om installationer, inklusive jag själv.

1.6 Avgränsning

I det här arbetet kommer jag enbart att gå närmare in på hur själva installationen av Sopro går till på en dator med Windows som operativsystem. Eftersom en servermiljö kan variera mycket mellan kunder, kommer jag inte att gå in på olika konfigurationer som vanligtvis behöver göras när Sopro installeras på en ny server. Dessa konfigureringar kan innehålla t.ex. skapandet av nya regler och öppnandet av portar i brandväggen. Jag kommer inte heller att närmare gå in på Sopros olika webbmoduler och hur dessa installeras.

2 Definitioner

2.1 CMD

CMD står för Command Prompt, eller komandoprompt på svenska. Med hjälp av CMD kan man köra många olika textkommandon utan att behöva använda ett grafiskt användargränssnitt. Komandoprompten kallas också ibland för komandotolken, man brukar tala om att CMD tolkar de inskriva kommandona åt Windows och så förstår datorn vad användaren beordrar datorn att göra. Namnet komandoprompt hänvisar också till den högra vinkelparentesen (>), även känt som större än-tecknet. Detta tecken visar användaren i CMD att nya kommandon kan skrivas in. (Windows, 2013)

Kommandona som används inom CMD kan delas i två typer: interna och externa. De interna kommandona är inbyggda i CMD själv. Till dessa hör kommandon som t.ex. skapandet av en ny mapp, kopiering och pausandet av ett skript. Externa kommandon är kommandon som använder sig av separata filer för att köra angivna kommandon. Fast filerna är externa, körs de ändå inom komandoprompten. Exempel på externa kommandon är Ping, xcopy och taskkill. Jag kommer senare att förklara närmare vad dessa kommandon gör. (Bhardwaj, 2006, s 13).

Komandoprompten har så gott som alltid funnits med i Microsofts operativsystem, ända från de första systemen från 1980-talet till den nyaste versionen Windows 8, som släpptes ut hösten 2012.

2.2 Batchfil

Om man vill mata in korta enstaka kommandon till komandoprompten, går det bra att rakt skriva in dem. Men ifall man vill ge många order efter varandra, kan man spara dessa i en textfil med filformatet batch. En batchfils filändelse är bat. Istället för att mata in alla kommandona en efter en i CMD, kan man istället mata in batchfilens namn samt lokalisering på datorn, och CMD går igenom filens textsträngar i ordningsföljd. Det går i viss mån att använda villkorssatser i batchfiler, men för det mesta används kommandon som anropar olika program och funktioner. I tidiga versioner av Windows anropades en batchfil, autoexec.bat, vid dators upstart för att lägga igång väsentliga program.

(Computer Hope, 2013). I figur 1 ser man ett exempel på hur autoexec.bat såg ut i MS-DOS version 6 från år 1993.

```
@ECHO OFF
PROMPT $P$G
PATH C:\DOS;C:\WINDOWS
SET TEMP=C:\TEMP
SET BLASTER=A220 I7 D1 T2
GOTO %CONFIG%
:WIN
LH SMARTDRV.EXE
LH MOUSE.COM /Y
WIN
GOTO END
:XMS
LH SMARTDRV.EXE
LH DOSKEY
GOTO END
:END
```

Figur 1: Autoexec.bat i MS-DOS version 6

3 Val av verktyg

Eftersom jag inte hade tidigare erfarenhet av automatisering och användning av automationsskript, kunde jag inte från början säga vilka program och lösningar jag borde använda mig av. Jag sökte information på nätet om de mest kända program som används för att skapa skript, och stötte snabbt på Windows Powershell. Windows Powershell är ett kommandoradsverktyg som baserar sig på Microsofts .NET Framework (TechNet, 2013). Windows Powershell kommer nuförtiden färdigt med i de nyaste Windows versionerna, men går även att manuellt installera på maskiner med en äldre Windows version.

3.1 CMD Command Shell

Jag började läsa igenom och pröva olika guider för att komma igång med Windows Powershell och prövade ganska enkla saker som skapning av mappar och kopiering av filer. Jag märkte genast att användningen av Windows Powershell är mer invecklat än jag hade förväntat mig, sättet att använda olika kommandon skiljde sig från komandoprompten som funnits med i Windows olika versioner i väldigt många år, inklusive den nyaste versionen Windows 8.

Jag kom fram till att inläringen av Windows Powershell skulle ta för lång tid att lära sig och jag bestämde istället att använda mig av traditionella CMD eller rättare sagt CMD Command Shell. Med shell talar man om ett s.k. skal som omringar ett system eller en applikation. Skalet fungerar som ett användargränssnitt och är vanligtvis alltid i textform. När man startar CMD.exe på en dator är det detta skal som användaren ser. (TechTerms, 2013)

CMD verkade i planeringsskedet att kunna utföra nästan allt som behövdes för installationen. Det som inte kunde gå att utföra med CMD skulle jag försöka lösa på något annat sätt. Eftersom CMD är mycket äldre och mycket mer använt i världen än Windows Powershell, skulle också information om olika lösningar vara mycket lättare att hitta.

3.2 Notepad++

För att skriva kod och skripten går det att använda Windows eget textredigeringsprogram Notepad. Notepad kommer med färdigt i Windows. Dock för att underlätta arbetet ville jag definitivt använda mig av en förbättrad version av detta program, dvs. Notepad++ som vi även har använt oss av i skolan. En väldigt bra egenskap med Notepad++ är att man kan ändra färgerna på text och bakgrund, vilket underlättar läsning och genomgång av kod. I programmet kan man välja vilken slags fil man arbetar med, och på så sätt har textsträngar och kommandon olika färger. Programmet stöder färdigt många olika programmeringsspråk som t.ex. HTML (Hypertext Markup Language), PHP (Hypertext Preprocessor) och viktigaste av allt, batchfiler, som jag har använt mig av för att skapa automationsskriptfilerna.

Personligen brukar jag ha en svart bakgrund när jag använder Notepad++, men i skärmdumparna i detta projekt har jag använt mig av vit bakgrundsfärg eftersom det är programmets standardinställning, dessutom sparar det printerfärg.

Notepad++ är gratis att ladda ner från utvecklarens egen hemsida <http://notepad-plus-plus.org/>.

3.3 Replace Text

Vanligtvis väljer man i samband med installationer av Windows program en destinationsmapp för den färdiga installationen, och komponenterna som installeras tillsammans med Sopimus Pro är inga undantag. Beroende på programmet brukar de flesta Windows program föreslå färdigt en destinationsmapp i stil med t.ex. C:\Program Files\Bolog\Produkt\. Men för att bättre hålla koll på alla filer efter själva installationen, installeras majoriteten av alla filer i samma huvudmapp, som sedan placeras dit man kommit överens om med kunden.

Valet av denna destinationsmapp måste göras före själva installationsskriptet körs. Destinationsmappen är heller inte varje gång detsamma, det kan hända att installationen ska göras på en annan hårddisk än C-stationen, som i de flesta datorer är standardalternativet. I detta fall måste denna destinationsmapp ändras på rätt ställe för alla respektive filer som behöver veta denna destination. Det skulle vara opraktiskt att ändra denna destination manuellt, eftersom poängen med hela projektet är att spara tid. Därför har jag använt mig av ett färdigt program för att ändra på denna destination på alla rätta ställen i koden.

Programmet heter Replace Text och är utvecklat av Ecobyte. Replace Text är ett textsöknings- och ersättningsprogram, med hjälp av Replace Text kan man byta destinationen på endast ett ställe och köra programmet, så ändras destinationen automatiskt på alla rätta ställen, och de destinationsrader i koden som är ägnade åt annat, lämnas ifred. Replace Replace Text är också gratis att ladda ner från deras hemsida <http://www.ecobyte.com/replacetext/>.

(Ecobyte, u.å.)

3.4 Autolt

De flesta av alla komponenter som hör till Sopimus Pro installationen behöver inte genast efter sin installation konfigureras desto mera, utan det kan göras senare ifall det behövs. Till installationen hör dock en fas, där databasadministrators lösenord ska bytas. Detta brukar vanligtvis göras så fort som möjligt så att det inte glöms bort. För att komma åt och ändra detta lösenord, använder vi oss av ett program som heter FlameRobin, som är

ett databasadministrationsverktyg. Det visade sig senare under projektet att denna fas gick lätt att lösa med hjälp av ett annat program, men detta visste jag inte under planeringen.

Istället använde jag mig av Autolt. Autolt är ett program i vilket man kan med AutoIts eget programeringspråk färdigt ge exakta kommandon som sedan i ordningsföljd härmar knapptryckningar på musen och tangentbordet. Med hjälp av detta program kunde jag direkt efter dess installation med hjälp av ett skript öppna FlameRobin, mata in standardlösenordet, byta standardlösenordet till ett annat, stänga programmet, och fortsätta med resten av installationsprocessen. Autolt använde jag också för att konfigurera applikationsservern samt programmet som sköter om den automatiserade säkerhetskopieringen av databasen varje dag. Autolt är gratis att ladda ner från adressen <http://www.autoitscript.com/site/autoit/downloads/>.

(Autolt, 1999-2012)

4 Installationsprocessen

Till näst går jag kort igenom installationsprocessen för Sopimus Pro (Sopro) installeras manuellt för hand. Jag kommer senare att berätta mera i detalj om de olika komponenterna. När Sopro installeras, består Installationsprocessen i grund och botten av följande faser:

1. Firebird installeras
2. FlameRobin installeras och används för konfigurering av Firebird
3. Databasen kopieras till rätt mapp
4. Sopimus Pro servern installeras, uppdateras och konfigureras
5. Klienterna Sentraali och Kassööri installeras
6. Säkerhetskopieraren av databasen installeras och konfigureras

4.1 Firebird och FlameRobin

Installationsprocessen börjar med att Firebird installeras. Firebird är en gratis relationsdatabashanterare vars utveckling baserar sig på källkod som släpptes ut år 2000 av Inprise, nuvarande Borland Software. (Firebird, 2000-2013).

Efter att Firebird har installerats bör databashanterarens användares standardlösenord bytas. För att kunna byta detta lösenord, installeras ett databasadministrationsverktyg, FlameRobin, som samtidigt fungerar som ett grafiskt gränssnitt för Firebird. Detta lösenord går att byta med hjälp av ett skript utan att det skulle behöva installeras något extra program, men ett databasadministrationsverktyg är bra att inkludera i installationen i alla fall. Verktöget kan i framtiden användas för att t.ex. kolla upp bortglömda lösenord ifall användare inte kan logga in, eller för att kolla upp andra uppgifter och data.

I steg tre flyttas själva databasen till rätt mapp. I det här fallet är det inte så noga att någon specifik databas används, denna databas är bara tillfällig och används för att försäkra användaren om att kommunikationen mellan de olika komponenterna fungerar. Efteråt kan databasen bytas ut till kundens egen ursprungliga databas ifall kunden i fråga inte är ny. I dessa fall är det vanligtvis fråga om att kunden t.ex. har införskaffat en ny server. Ifall kunden i fråga är en ny kund som inte har en egen databas från tidigare, byts den tillfälliga databasen ut till en databas som inte innehåller annat data än demonstrationsavtal. Sopimus Pro databasens storlek kan variera mellan 30 MB (liten databas) till över 1 GB (väldigt stor databas).

4.2 Applikationsservern och klienterna

Till näst i installationsprocessen kommer den första pusselbiten som är i hopsatt och skapad av Rediteq själv, det vill säga Sopimus Pro Sovelluspalvelin, (Sopimus Pro applikationsserver). Applikationsservern används för att kommunicera och sköta nätverkstrafiken mellan databasen och användargränssnittet, och även webbsidor som kan vara kopplade till databasen. Efter att denna komponent är installerad, bör den uppdateras till rätt version. Uppdateringsversionen beror däremot på databasen som ska användas. Orsaken är följande: till databasen körs då och då nya så kallade utvecklingspaket. Dessa utvecklingspaket kan innehålla buggfixar, ändringar till nuvarande egenskaper, eller alldeles nya sådana. Dessa utvecklingspaket går hand i hand med applikationsserverns version. Ifall versionen är för gammal eller för ny i jämförelse med paketen som har (eller inte har) körts till databasen, kan det hända att man inte kan logga in till användargränssnittet överhuvudtaget.

Efter att applikationsservern är uppdaterad och konfigurerad, kan man installera de så kallade klienterna, eller användargränssnitten för Sopro. Sopro består av 2 olika användargränssnitt: Sentraali (Centralen) och Kassööri (Kassören). Centralen används bland annat till för att mata in nya studieavtal och nya företag, administrera studerande under deras studietid och ändra på inställningar som påverkar hela Sopro programmet.

Kassören används sen igen för att hantera räkningar, utbildningsersättningar och ärenden som för det mesta har med pengar att göra. Efter att användargränssnitten är installerade, bör även de uppdateras. Ifall datorn i fråga har tillgång till internet samt kontakt till Rediteqs uppdateringssystem via webben, är uppdateringen relativt enkel att utföra. Det räcker att starta någondera av klienterna och logga in, och programmet uppdateras automatiskt så länge som tidigare nämnda kriterier är uppfyllda. Versionerna kan också uppdateras genom att bara ersätta de gamla exe filerna med de nya, nackdelen med detta är att då får man inte samtidigt prövat ifall den automatiska uppdateringen via webben fungerar. I figur 2 kan man se hur Centralens inloggningsruta ser ut. Nere till vänster kan man se den nuvarande versionen.



Figur 2: Sopro Centralen inloggningsruta.

4.3 Säkerhetskopieraren

Ifall databasen är den som skall användas, är programmet nu färdigt att användas. Till installationsprocessen hör till installation och konfigurering av en automatisk säkerhetskopierare av databasen. Det är nödvändigt och mycket viktigt att ta säkerhetskopior av databasen eftersom databasen kan innehålla många år av mycket viktig information relaterat till studerandens studier m.m. Säkerhetskopieraren som används är Time To Backup. Time To Backup är utvecklat av Sqllly Development, och är skapad för att ta säkerhetskopior av Interbase eller Firebird databaser. Time to Backup består av två delar, servicen samt användargränssnittet (Sqllly, 1999-2011). Efter att säkerhetskopieraren är installerad behöver den ännu konfigureras.

5 Utförandet av en tyst och obevakad installation

Med en obevakad installation menar man en installation som inte kräver åtgärder från användaren under själva installationen. Med en tyst installation menar man en installation som utförs utan att notifikationer eller meddelanden visas under installationen åt användaren (Sourceworld, u.å.).

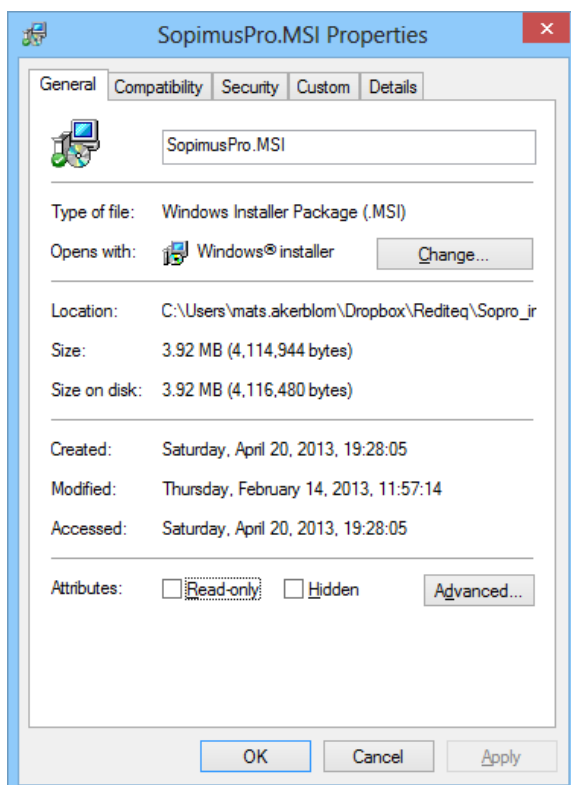
För att kunna utföra något av dessa installationssätt (eller båda), behöver installeraren veta i förväg väsentlig data för att inte under installationen behöva fråga användaren om uppgifterna som behövs. För att kunna överföra dessa data åt installeraren, behöver installeraren tillgång till en så kallad svarsfil, man kan tala om att installeraren får de rätta svaren från svarsfilen. Ibland kan installerarens leverantör inkludera denna svarsfil tillsammans med resten av installationsfilerna, men vanligtvis är man tvungen att skapa denna svarsfil själv.

För att skapa denna svarsfil, måste installationen av applikationen genomföras en gång för hand, medan alternativen man väljer samtidigt sparas i svarsfilen. För att kunna spara dessa installationsdata i svarsfilen måste installeraren startas via komandoprompten med en så kallad switch (växel) inkluderad i textsträngen, som behövs för att skapa svarsfilen. Alternativt kan man skriva textsträngen i en batchfil som sedan kan köras flera gånger utan att textsträngen behöver skrivas på nytt.

5.1 Installationstyper

Den nödvändiga switchen som behöver användas för att få svarsfilen skapad är inte den samma för varje installerare, switchen varierar beroende på vilket system installeraren är skapad med. Exempel på de vanligaste systemen som leverantörer använder för att skapa installerare med är Microsofts egen Windows Installer Service (MSI), InstallShield och Inno Setup. För att alltså veta vad för variant av switchen man behöver använda, behöver man veta typen av system som installeraren är skapad med.

Hur lätt man får reda på denna information varierar, typen av system kan man ibland få fram från själva installationsfilen. T.ex. Microsofts egen installerare Windows Installer Service har filändelsen MSI. Om man öppnar vyn för filens egenskaper, ser man genast på den första fliken att det är frågan om en MSI fil. Ett konkret exempel kan ses i figur 3 nedan.



Figur 3: Egenskaper-vyn för en MSI fil.

För de andra installationstyperna är det inte lika enkelt som med MSI att få reda på vad för typ av installerare man använder. Om installeraren har ett simpelt namn som setup.exe, är det möjligt att det är en installerare av typen InstallShield. Detta är inte skrivet i sten, det säkraste sättet att verifiera denna information är att pröva installera

applikationen med installeraren och samtidigt använda switchar som InstallShield identifierar.

Inno Setup filer har inte några synliga ledtrådar i själva filtypen, då rekommenderas det också att pröva Inno Setups egna switchar för att försäkra sig om att det faktiskt är en Inno Setup fil. Jag råkade av en slump hitta ett program vid namnet Exe to MSI Converter Pro, skapad av Aprel. Jag använde inte detta program för att konvertera installationsfilerna från en typ till en annan. Istället använde jag programmet för att få reda på typen av installationsfil. Med hjälp av programmet får man denna information tillgänglig genast efter att man matat in installationsfilens lokalisering på datorn.

5.2 Skapandet av en svarsfil

När man vet vilken typ av installerare man handskas med, kan man skapa svarsfilen. Som jag tidigare nämnde, skapas svarsfilen genom att köra installationen av applikationen en gång för hand, medan man samtidigt använder rätt sorts switch för att spara informationen i en fil. I kodexempel 1 nedan kan man se hur denna sträng kan se ut. I exemplet används Firebird, som har en installationsfil av typen Inno Setup. Med Inno Setup installationsfiler använder man switchen `/SAVEINF="filnamn"`. Om filnamnet inte innehåller mellanrum lämnas citattecknen bort. Om Firebird skulle råka vara av filtypen InstallShield, skulle man istället för `/SAVEINF` använda switchen `/r`. Bokstaven `r` står i det här fallet för record, eftersom man kan tala om att installationen "bandas in". I exemplet befinner sig även både installationsfilen samt batch filen i samma mapp, detta innebär att man inte behöver mata in installationsfilens lokalisering i textsträngen. Före switchen skriver man in installationsfilens fullständiga namn, inklusive filändelsen.

Kodexempel 1: Skapandet av en svarsfil

```
Firebird-2.1.5.18497_0_Win32.exe /SAVEINF=firebirdinf
```

Eftersom inga andra switchar används, ser installationen ut som vanligt för användaren. Installationen sker inte tyst, utan alla uppgifter som behövs frågas av användaren. Beroende på användarens UAC (User Account Control) inställningar, kan det också komma Windows säkerhetsvarningar i samband med installationen. Efter att installationen är utförd, sparas svarsfilen. Med Inno Setups `/SAVEINF` sparas svarsfilen med det namn man angett den med switchen. Filen sparas i samma mapp som

installationsfilen befinner sig i. Ifall det redan finns färdigt en svarsfil med samma namn i samma mapp, ersätts den gamla med den nya. Om Firebirds installationsfil skulle vara av typen InstallShield, skulle svarsfilen heta setup.iss, denna fil sparas alltid i samma mapp, vilken mapp det är beror på vilken version av Windows som man använder. Jag kommer senare att gå närmare in på hur svarsfilen ser ut och vad den innehåller.

5.3 Användning av svarsfilen

Efter att svarsfilen är skapad, kan den användas i framtida installationer för att läsas av och utan användarens inblandning. För att använda svarsfilen i samband med en installation, ersätts /SAVEINF med /LOADINF="filnamn", se Kodexempel 2.

Kodexempel 2: Användandet av svarsfilen

```
Firebird-2.1.5.18497_0_win32.exe /LOADINF=firebirdinf /SILENT
```

När man använder LOADINF, behöver man inte definiera svarsfilens lokalisering på datorn, så länge som svarsfilen och installationsfilen befinner sig i samma mapp på datorn.

När man använder sig av en svarsfil, räcker det inte med att specificera i textsträngen vad svarsfilen heter. Man behöver även beordra CMD att läsa svaren för installationsalternativen från filen och inte fråga användaren om dessa uppgifter. Detta förverkligar man med switchen /SILENT (tyst). /SILENT gömmer inte alla installationsfönster, man kan fortfarande se framstegsbalken i ett fönster på skärmen. Vill man att användaren inte ser några installationsfönster alls, kan /SILENT ersättas med /VERYSILENT ("väldigt tyst"). Ifall det under installationen uppstår något fel som skapar ett felmeddelande, syns denna meddelande ännu även om switchen /VERYSILENT används. Om man vill bli av med alla felmeddelanden kan man lägga till switchen /SUPRESSMSGBOXES. (Dämpa notifikationsfönstren).

(Sourceworld, u.å.)

6 Den automatiserade installationen

Till näst kommer jag att gå igenom själva installationsskriptet samt den automatiserade installationen av Sopimus Pro (Sopro).

6.1 Install.bat

Före någon kod hade skrivits ner hade jag i förväg tänkt att jag inom installationen kommer att använda en batchfil som huvudbatchfil, som sedan anropar i tur och ordning andra batchfiler som innehåller ett installationsskript för respektive komponent. Jag hade föreställt mig att det skulle på det här sättet vara lättare att hålla koll på de olika filerna.

När jag dock började lägga ihop installationsskriptet och huvudbatchfilen, märkte jag att det inte fungerar helt så som jag hade tänkt mig. Istället för att batchfilen anropades och den anropade filens innehåll kom i CMD fönstret för huvudfilen, öppnades den anropade filen i ett nytt CMD fönster, som sedan stängdes efter att det externa skriptet körts. Det här sättet tyckte jag inte alls om, eftersom användaren lätt kunde gå miste om den externa filens innehåll. Detta ledde till att jag skrapade iden med de externa batchfiler, och istället skulle huvudinstallationsfilen innehålla innehållet på nästan alla externa batchfiler.

6.1.1 Skapandet av en ny mapp

Installationsskriptet börjar med att en ny mapp skapas och ges namnet SopimusPro utan mellanrum i namnet. SopimusPro-mappens lokalisering på datorn bestäms tillsammans med kunden före installationen börjar. Till denna mapp kommer alla installationskomponenter att installeras ifall de inte har en standardplats på datorn vart filerna bör läggas, t.ex. i datorns Windows-mapp. Med CMD kan man skapa en ny mapp men två olika kommandon, MD eller MKDIR. Båda kommandona kommer från det engelska begreppet Make Directory (Skapa en ny mapp). Efter kommandot definieras ännu den nya mappens lokalisering på datorn samt mappens namn, se kod 1. (SS64, 1999-2013)

Kod 1. Skapandet av en ny mapp

```
mkdir C:\SopimusPro
```

6.2 Firebird

Den första komponenten i Sopro-installationen som installeras är Firebird, som fungerar som databashanterare och databasmotor för Sopro databasen. Med databasmotor menar man den centrala delen i en databashanterare, som lägger till, ändrar och tar bort uppgifter från databasen.

Firebird installeras på datorn som en service (tjänst). En service skiljer sig även från vanliga applikationer med att en service körs i bakgrunden, utan användarens inblandning eller medvetande. Detta betyder i praktiken att Firebird läggs igång automatiskt varje gång datorn startas, och hålls igång tills datorn stängs av.

Installationen av Firebird sker till som de flesta installationer med vissa tillägg. Under installationen frågas väsentliga uppgifter så som destinationsmapp, om användaren vill ha en genväg till applikationen via Start-menyn o.s.v. Förutom dessa alternativ så väljer vi ännu att Firebird installeras som en service och inte som en applikation, samt att Firebird läggs igång när datorn startas. Dessutom frågas vilka delar av Firebird man vill inkludera i installationen. Utvecklingsverktyg brukar lämnas bort eftersom de inte används och endast tar onödigt med utrymme.

6.2.1 Svarsfilen för Firebird

Ifall Firebird installerades genom att samtidigt använda switchen /SAVEINF, kan den nyskapade svarsfilen nu användas. I Kod 2 ser man hur svarsfilen för Firebird ser ut.

Kod 2. Firebirdinf – svarsfilen för Firebird

```
[Setup]
Lang=en
Dir=C:\SopimusPro\Firebird_2_1
Group=Firebird 2.1 (Win32)
NoIcons=1
SetupType=custominstall
Components=servercomponent,servercomponent\superservercomponent,clientcomponent
Tasks=useguardiantask,useservicetask,autostarttask,copyfbclienttosystask,copyfbclientasgds32task
```

Den första raden efter [Setup] definierar det valda språket, vilket i det här fallet är engelska. Rad nummer två definierar destinationsfoldern för installationen, vilket är inuti SopimusPro-mappen som vi skapade tidigare. De två följande raderna berättar i vilken startgrupp genvägen till applikationen placeras i, och om genvägen används överhuvudtaget. SetupType=custominstall berättar att vi har valt en anpassad

installation, dvs. att vi valde att lämna bort utvecklingsverktygen. Delarna som vi lämnade kvar syns på nästa rad. Servercomponent syns listat två gånger, detta beror på att ifall man väljer att Server komponenten hör till installationen, ska man ännu efteråt välja vilken s.k. arkitektur man vill använda sig av, klassisk eller superserver. Båda typerna kan användas för olika ändamål, men i överlag rekommenderas det att man väljer superserver om man använder Windows. (Firebird, 2000-2013)

Den sista raden i svarsfilen innehåller all övrig information. Useguardiantask definierar att ytterligare en annan service, Guardian (väktaren) används för att kontrollera Firebird. Guardian kan vid behov användas för att t.ex. stanna Firebird och hindra användare från att använda databasen. Detta kan vara nödvändigt t.ex. vid underhållsarbete. Guardian övervakar även själva Firebird servicen, och lägger igång den igen ifall den av någon orsak skulle avslutas. Useservicetask berättar att Firebird installeras som en service och inte som en applikation, autostarttask betyder att Firebird läggs automatiskt igång varje gång datorn startas.

De två sista uppgifterna i svarsfilen definierar att under installationen kopieras nödvändiga filer till datorns system-mapp, samt att de kopierade filernas filnamn ändras för att stöda äldre versioner av applikationer som använder sig av dessa filer.

Firebird kan nu installeras tyst med hjälp av svarsfilen. Installationssträngen som jag har använt mig av i själva skriptet kan man se i kod 3.

Kod 3. Installering av Firebird

```
start "" /WAIT Firebird-2.1.5.18497_0_Win32.exe /LOADINF=firebirdinf /SILENT /NORESTART
```

6.2.2 Wait-kommandot

Som vi vet från tidigare exempel, kan Installationsfilen läggas igång utan startkommandot. Varför det läggs till här är för att kunna använda switchen /WAIT. Om vi inte skulle lägga till denna switch så skulle installationsskriptet fortsätta till nästa punkt medan Firebird ännu håller på att installeras. Eftersom resten av komponenterna är beroende av att Firebird finns färdigt installerat, behöver vi vänta på att Firebird installationen har körts färdigt.

De två citattecknen mellan start och /WAIT definierar titeln på ett nytt fönster som öppnas i samband med start. Beroende på vad som startas, kan detta fönster bara synas på skärmen endast en bråkdelssekund. Därför är det inte viktigt vad för titel som anges och det kan lämnas tomt så som i exemplet. Fast man lämnar det tomt så rekommenderas det ändå inte att lämna bort citattecknen helt och hållet, det sägs att det kan leda till oväntade problem, dock stötte jag inte på dem själv under projektet. (SS64, 1999-2013)

/SILENT används för att installeraren ska veta att läsa svaren från svarsfilen och inte fråga användaren. Jag har använt switchen /SILENT istället för /VERYSILENT, för att ändå kunna se framstegsbalken på skärmen. Den sista switchen /NORESTART (ingen omstart) i textsträngen beordrar installeraren att inte starta om datorn efter att installationen är färdig, även om det skulle vara nödvändigt. Till största delen frågar Firebird inte efter en omstart efter en installation. Firebird har tendens att be om en omstart om Firebird har funnits tidigare på datorn, därför lade jag till switchen som en säkerhetsåtgärd. Vanligtvis får vi inte starta om en dator som ägs av en kund, utan deras lov.

6.3 FlameRobin

Eftersom Firebird installerades som en service, saknar det ett grafiskt användargränssnitt. Här kommer FlameRobin med i bilden. FlameRobin är ett gratis program med öppen källkod och är skapat för ändamålet att fungera som ett administrativt användargränssnitt för Firebird.

Installationen av FlameRobin går till som de flesta vanliga installationer utan något desto märkvärdigare, i samband med installationen definieras användarspråk, installationslokalisering och genvägsikoner. FlameRobin-installeraren är av typen Inno Setup, vilket innebär att switchen /SAVEINF kan användas för att skapa svarsfilen.

Kod 4. flamerobininf – svarsfilen för FlameRobin

```
[Setup]
Lang=default
Dir=C:\SopimusPro\FlameRobin
Group=FlameRobin
NoIcons=1
Tasks=
```

FlameRobin installeras i samband med Sopro så att man senare vid behov kan komma åt data i databasen. När Sopro installeras vanligtvis manuellt, behövs FlameRobin genast efter att Firebird har installerats. När Firebird installeras, skapas samtidigt ett administrativt användarkonto med användarnamnet Sysdba, som inte kan raderas. Detta användarkonto har ett färdigt inställt standardlösenord som bör bytas så snabbt som möjligt, och till detta används FlameRobin. När jag behövde få bytet av lösenordet automatiserat, visste jag inte om att det går att göra med ett annat program samt ett färdigt kommando i CMD. Jag stötte inte heller på kommandot medan jag sökte information på nätet, så jag blev tvungen att hitta på något annat.

6.3.1 Konfigurering av FlameRobin med Autolt

Medan jag sökte en lösning stötte jag på Autolt. Autolt är ett automatiseringsprogram med ett eget programmeringsspråk. Till skillnad från skripten som körs i CMD med batchfiler, består ett Autolt-skript av kommandon som simulerar tangent- och musknappstryckningar, som sedan körs i rätt ordning för att härma en användare. Även om Autolt använder sig av sitt eget programmeringsspråk, är kommandona väl dokumenterade och det var lätt att komma igång. (Autolt, 1999-2012)

Bytandet av lösenordet i FlameRobin består av följande steg:

1. FlameRobin startas
2. Inloggning med Sysdba
3. Ändring av lösenordet
4. Bekräftning med det nya lösenordet
5. FlameRobin avslutas

Steg ett och två av skriptet kan ses i kod 5 nedan. Autolt-skriptet börjar med att lokaliseringen av FlameRobin definieras och startas med kommandot run (kör). Ett Autolt skript fungerar liknande som ett batchskript, kommandona följs i tur och ordning. Om datorn inte är tillräckligt snabb kan det hända att föregående punkt inte hinner köras färdigt förrän nästa punkt börjar, vilket kan leda till problem. Därför lägger vi till kommandot WinWaitActive, som i det här fallet på svenska betyder att Autolt ska vänta tills ett fönster med namnet Flamerobin är aktivt, och därefter fortsätta med skriptet.

Kod 5. Startandet av FlameRobin och inloggning med Autolt-skriptet

```
Run ("C:\SopimusPro\FlameRobin\flamerobin.exe")

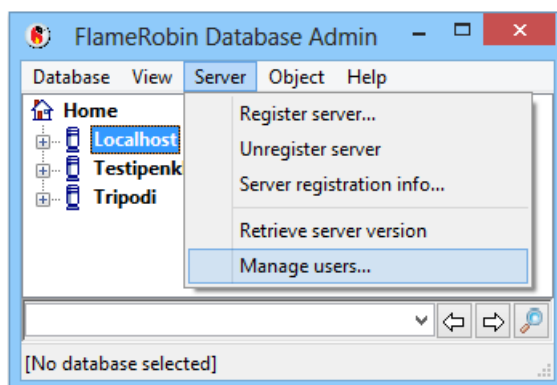
WinWaitActive("FlameRobin")

Send("!s")
Sleep ( 1000 )
Send("m")

WinWaitActive("Database Credentials")

send("masterkey")
Sleep ( 1000 )
Send("{ENTER}")
Sleep ( 1000 )
```

När FlameRobin väl är igång med fönstret aktivt, kan vi börja imitera tangentbordet och musen för att få lösenordet bytt. Helt som i de flesta Windows programmen, har FlameRobin högst upp en menybalk med meny alternativ samt deras undermenyer. Utseendet på FlameRobins grafiska användargränssnitt kan ses i figur 4.



Figur 4: FlameRobin

Huvudmenyn i FlameRobin består av 5 alternativ, där man vill välja den tredje, Server, och under detta alternativ ännu välja alternativet Manage users... (hantera användare). Här kan man välja ifall man vill använda tangentbordet, musen eller båda. Eftersom jag inte visste hur bra eller dåligt val med musen fungerar med Autolt, ville jag hålla mig till användning av tangentbordet samt snabbkommandon, eftersom dessa är samma oberoende av skärmupplösning och skärmens storlek.

För att simulera ett tryck på tangentbordet med Autolt. används kommandot send (skicka) för att imitera knapptryckningar. Som man kan se i kod 5, representeras de första knapptryckningarna efter att FlameRobin-fönstret är aktivt, av "!s". Utropstecknet i det

här fallet motsvarar i AutoIt-språk att Alt-tangenten trycks ner samtidigt som "s" trycks ner. Genom att trycka på alt, blir snabbkomandona för de olika menyalternativen synliga med ett understräck under namnet, och på så sätt kan vi välja rätt menyalternativ med respektive snabbkommando, i det här fallet med s från ordet server.

Före vi fortsätter med skriptet, ber vi AutoIt att vänta en stund. Som sagt läses skriptet i ordningsföljd från början till slut, och ifall det inte definieras i skriptet, väntar AutoIt inte på att ett föregående kommando har körts färdigt. En modern dator behöver inte någon märkbar tid för att tänka och bearbeta kommandona som kommer från AutoIt. Moderna datorer är kapabla av att köra alla dessa kommandon under en sekund. Dock i verkligheten händer inte resultatet av kommandot i samma exakta stund och kan leda till att knapptryckningen inte sker i rätt tillfälle därför att skriptet läses igenom för snabbt.

Därför vill vi att AutoIt väntar ett kort ögonblick mellan kommandona. Här kan vi inte använda oss av WinWaitActive, eftersom fönstret redan är färdigt aktivt. Istället använder vi kommandot sleep (sov) för att vänta. Längden på pausen definieras i parentes med millisekunder, 1000ms = en sekund.

Efter den sekund långa pausen är undermenyn fortfarande synlig och vald, och vi kan fortsätta till nästa fönster genom att välja Manage users från undermenyn. Snappkommandot är "m", och med snabbkomandona fortfarande aktiva behöver vi inte definiera på nytt att alt-tangenten ska tryckas ner.

Efter menyvalet dyker ett inloggningsfönster upp med namnet Database Credentials (inloggningsuppgifter). Sysdba är färdigt vald som användare och går inte att byta, så det räcker med att skriva in det nuvarande lösenordet. Standardlösenordet för användarkontot i FlameRobin är alltid masterkey. Det räcker med att skriva hela lösenordet inom ett ända send-kommando. Lösenordet bekräftas med Enter-tangenten och skrivs in mellan klamrar för att berätta åt AutoIt om att det handlar om en ända tangent som trycks ner. Om vi skulle lämna bort klamrarna, skulle AutoIt tolka det som ett ord och skriva det efter masterkey på samma rad.

Efter att vi är inloggade med administratörskontot, dyker ett nytt fönster med namnet Localhost upp. I detta fönster kan vi se alla nuvarande användare och lägga till nya. För att ändra på en nuvarande användares uppgifter, klickar man på en liten ikon bredvid

namnet. Denna ikon kommer man inte åt med TAB-tangenten, vilken man vanligtvis kan använda för att bläddra framåt mellan olika fält i ett aktivt fönster. Eftersom vi inte kommer åt ikonerna med tangentbordet, behöver vi nu beordra Autolt om att använda musen.

För att Autolt ska veta vilken punkt på skärmen ska klickas med musen, anger vi koordinater för detta. Koordinaterna får vi reda på genom att använda Autolt Window Info, en skild komponent som kommer med i installationen av Autolt. Localhost-fönstret öppnas inte nödvändigtvis varje gång i samma position på skärmen, vilket betyder att också koordinaterna kan variera. För att försäkra oss om att positionen alltid är det samma, kan vi förstora fönstret så att det fyller hela skärmen. Hur detta går till kan vi se i kod 6.

Kod 6. Användning av musen i Autolt

```
WinWaitActive("localhost")
Sleep ( 1000 )

Send("{ALT down}{SPACE down}")
Sleep ( 1000 )
send("x")
Sleep ( 1000 )

send("{ALT up}{SPACE up}")
Sleep ( 1000 )

MouseClicked ( "left", 20,126 )
```

För att förstora ett aktivt fönster i Windows, kan vi använda snabbkommandot för detta ändamål. Förstoringen lyckas genom att hålla ner samtidigt både alt- och mellanrumstangenten, och därefter trycka på "x". Till skillnad från det tidigare kommandot "!s" där allt trycktes ner en gång, behöver både alt och mellanslagstangenten hållas nere medan man trycker på x. Därför definieras detta även i koden med ordet "down" efter båda tangentnamnen. Efter att x blir nertryckt, förstoras fönstret. Alt- och mellanslagstangenten hålls fortfarande nertryckt, och därför behöver vi beordra Autolt att släppa tangenterna, detta lyckas med att skriva "up" efter tangentnamnen.

Till näst kommer själva mustryckningen med kommandot MouseClick. Inom parentes definieras vilken musknapp som trycks ner samt koordinaterna vi fick fram med Autolt Window Info.

När man klickar på ikonen bredvid namnet, dyker ett nytt fönster upp med alla användaruppgifter. Autolt fyller i fälten i tur och ordning och navigerar till nästa fält med TAB-tangenten. Samtidigt skrivs det nya lösenordet i två av fälten, där det andra fältet fungerar som kontroll för stavfel. Till sist navigerar man med TAB till en sparningsknapp, som samtidigt stänger fönstret. I Autolt-skriptet tillägger vi här kommandot WinWait, vilket betyder att Autolt stannar skriptet tills det föregående fönstret har stängts.

Efter att fönstret stängs dyker ett annat fönster upp med ett ända fält, hit behöver vi mata in det nya lösenordet. Efter att rätt lösenord har matats in, stängs fönstret automatiskt, och användaren hamnar till FlameRobins huvudmeny där man började. FlameRobin avslutas med kommandot WinClose("FlameRobin").

När jag ursprungligen skapade Autolt-skriptet, använde jag en dator med Windows Xp. Det färdiga skriptet fungerade så som jag ville utan problem. Det var givetvis nödvändigt att testa skriptet på en annan dator för att försäkra sig om att det fortfarande fungerar. Det som jag var mest orolig för var att kommandot med muspekaren inte skulle hamna på exakt samma punkt med de angivna muskoordinaterna på en annan dator. Och mycket riktigt, när jag prövade skriptet på en annan dator med Windows 8, hamnade muspekaren några millimeter åt sidan från ikonen, och skriptet stannade, även om jag använde samma fysiska skärm, troligtvis på grund av att skärmens skärmupplösning inte var det samma på de två datorerna.

6.3.2 GSEC

Detta ledde till att jag inte kunde använda Autolt för att ändra lösenordet, och var tvungen att hitta på något annat. Med hjälp av en kollega hittade vi en lösning som erbjöds av Firebird själv. I samband med att Firebird installeras, kommer det med Firebirds egen komandotolk, GSEC. Med hjälp av GSEC kommer man åt en krypterad fil i Firebird-mappen, där alla användare och deras uppgifter sparas. På så sätt kan man byta lösenordet med en ända textsträng utan att ens behöva installera FlameRobin över huvudtaget. (Heymann, S (2009, 20 november))

Kod 7. Bytandet av Firebird lösenordet med GSEC

```
start "" /wait C:\SopimusPro\Firebird_2_1\bin\gsec  
-user SYSDBA -password masterkey -modify sysdba -pw newPassword
```

Hur detta går till kan ses i kod 7. GSEC anropas på samma sätt som tidigare installationsfiler. På den andra raden definieras först användaren som man vill använda, dvs. Sysdba. För att kunna göra några ändringar, måste vi också ge det nuvarande lösenordet. Sedan definierar vi med kommandot `-modify` (modifiera) att vi vill ändra Sysdbas lösenord, samt vad det nya lösenordet är.

Efter lösenordsbytet stängs GSEC automatiskt och installationsskriptet kan fortsätta. Fast vi inte sist och slutligen behöver använda FlameRobin nu, inkluderas den ändå i installationen eftersom den kommer ändå att behövas förr eller senare.

6.4 Kopiering med CMD

Nästa steg inom installationen är att kopiera den tillfälliga databasen till sin egen mapp inne i SopimusPro-mappen. Denna mapp skapas på samma sätt som själva SopimusPro-mappen och namnges till Tietokanta (databas). Det finns åtminstone tre olika kommandon för kopiering i CMD: `copy`, `xcopy` och `robocopy`. Skillnaden mellan `copy` och `xcopy` är att `xcopy` har mera switchar som kan användas, men själva kommandot föråldrades när Windows Vista släpptes ut. Detta betyder inte att `xcopy` inte längre fungerar, men det är mycket möjligt att det händer med framtida operativsystemsversioner, och rekommenderas därför inte längre att användas. `Robocopy` är ett mångsidigare kommando än `copy` med många olika switchar, men till vårt ändamål räcker `copy` riktigt bra. (SS64, 1999-2013)

Kod 8. Copy.bat

```
copy SopimusPro.gdb C:\SopimusPro\Tietokanta
exit
```

I kod 8 kan man se strängen för kopiering. Denna sträng skrivs inte i själva huvudbatchfilen, utan i en egen batchfil med namnet `Copy.bat`. Orsaken till detta är att nu kan vi från huvudbatchfilen anropa `copy.bat` med kommandot `start` och switchen `/WAIT`, så att skriptet inte fortsätter medan kopieringen körs.

6.5 Sopimus Pro applikationsserver

Sopimus Pro Sovelluspalvelin (Sopimus Pro applikationsserver) är den första komponenten i Sopro installationen som är skapad och ihopsatt att Rediteq själv. Applikationsservern är en COM+ applikation. COM står för Component Object Model, och kombinerat med Microsoft Transaction Server (MTS) får vi COM+. En COM+ applikation installeras på servern för att hantera många olika uppgifter, så som resurshantering, säkerhet, samt nätverkstrafiken mellan databasen och klienterna. (Microsoft Developer Network, 2013, 12 oktober)

6.5.1 Problem med svarsfilen

Även applikationsserverns installationsfil är av typen Inno Setup. Svarsfilen skapas på samma sätt som tidigare. När jag dock skulle testa att svarsfilen för applikationsservern fungerar, stötte jag på ett problem. I samband med installationen frågas användaren om IP-adressen (Internet Protocol) till datorn där databasen befinner sig. Denna information sparades dock inte i svarsfilen, och när själva installationen kördes efteråt tillsammans med svarsfilen, dök fönstret för inmatning av databasdatorns IP-adress upp, även med switchen /SUPRESSMSGBOXES.

Vid en manuell installation sparas informationen om databasdatorns IP-adress i en skild initieringsfil i datorns Windows-mapp, med filnamnet SopimusPro-palvelin.ini. Jag märkte att om man tvingar installationsprocessen att avsluta när IP-adressfönstret dyker upp, har redan det installerats som behöver installeras på datorn för att applikationsservern ska fungera. Om installationsprocessen avslutas på det här sättet, skapas initieringsfilen inte i samband med installationen, men detta är inte något problem eftersom initieringsfilen kan kopieras skilt till rätt mapp och modifieras när som helst.

För att lösa detta problem så hamnade jag att improvisera med CMD, installationen av applikationsservern kan ses i kod 9 nedan. Applikationsservern installeras med samma sträng som Firebird och FlameRobin, med undantaget att switchen /WAIT lämnas bort. Orsaken till detta är att vi inte kan beordra CMD att vänta tills installationen är färdig, eftersom installationen stannar vid IP-adressfönstret. När vi lämnar bort switchen /WAIT, fortsätter skriptet till nästa steg medan applikationsservens installeras. I nästa steg

beordrar vi CMD att vänta en stund tills applikationsservern är installerad, och sedan avsluta installationen.

6.5.2 Pausning av skriptet

CMD har ett antal sätt att vänta, så som timeout, sleep (sov), pause (paus) eller ping. Både Timeout och sleep verkade lovande, enligt instruktionerna räcker det att efter kommandot definiera i sekunder hur länge skriptet ska vänta. Men av en okänd orsak fick jag inte dessa två kommandon att fungera, även fast jag följde skriftliga instruktioner. I och för sig har detta ingen betydelse, eftersom jag läste att båda dessa kommandon kräver mera processorresurser av datorn än kommandona ping och pause. (SS64,1999-2013)

Kod 9. Installering av applikationsservern

```
start "" setup.exe /LOADINF=propalvelininf /silent
PING 1.1.1.1 -n 1 -w 30000 >NUL
start "" /wait serverInstallTaskKill.bat
copy SopimusPro-palvelin.ini C:\WINDOWS
```

Kommandot pause gör så som namnet indikerar, att skriptet stannar med meddelandet "Press any key to continue. . .", dvs. Tryck på någon tangent för att fortsätta. Eftersom skriptet ska kunna genomföras från början till slut utan användarens inblandning, vill vi inte använda kommandot här. Istället använder vi kommandot ping. Ping-kommandots ursprungliga mening är inte att erbjuda väntetid, utan att pröva ifall man har en nätverksanslutning till en specifik IP-adress. Om inga switchar används, fungerar kommandot ping så att måladressen pingas fyra gånger med ca en sekunds mellanrum. Tiden kan variera på vissa datorer beroende på vilken adress som pingas. För att pingningen alltid ska ta lika lång tid, pingas en IP-adress som inte finns, t.ex. 1.1.1.1. Detta leder till att efter 4 sekunder får vi felmeddelandet timeout, vilket innebär att adressen inte kan nås.

Med switchar kan vi närmare specificera den exakta tiden som vi vill stanna skriptet. I koden uppnår vi detta med att inkludera `-n 1` och `-w 30000` efter ping-kommandot, där `n` är antalet pingningsförsök och `w` antalet tid i millisekunder förrän timeout-felmeddelandet visas (Van der Woude, R, 2012, 21 april).

Här måste vi bedöma själv hur lång tid vi vill att skriptet ska vänta så att installationen inte avslutas för tidigt, dvs. förrän fönstret var IP-adressen frågas dyker upp. På en modern dator tar det här mellan 5-10 sekunder, och för att vara på den säkra sidan ber vi CMD att vänta 30 sekunder. Den sista punkten i strängen är <NUL. <-tecknet definierar vart resultatet av kommandot skrivs ut. När vi använder NUL omdirigeras resultatet till en Windows fil med namnet NUL, som sedan skrotar resultatet, vilket betyder att när skriptet körs, ser användaren inte pingningsförsöket samt timeout-felmeddelandet. (Van der Woude, R, 2012, 24 juli)

6.5.3 serverInstallTaskKill

Efter att den angivna tiden går ut, fortsätter skriptet med att anropa en skild batchfil kallad serverInstallTaskKill.bat.

Kod 10. serverInstallTaskKill.bat

```
taskkill /f /im setup.tmp  
exit
```

Kommandot taskkill beordrar att en process ska avslutas. Switchen /im står för imagename eller namnet för processen. Samtidigt använder vi switchen /f för att tvinga processen att avsluta. /F bokstaven kommer från det engelska ordet forceful (kraftful). (SS64, 1999-2013) Setup.tmp är namnet för processen.

Efter att processen är avslutad, fortsätter skriptet med att kopiera initieringsfilen till Windows-mappen. Här behövs ingen definiering av någon väntetid, eftersom initieringsfilen är väldigt liten i storlek. I kodexempel 3 kan man se ett exempel på innehållet i initieringsfilen.

Kodexempel 3. SopimusPro-palvelin.ini

```
[SopimusPro]
DriverName=Firebird
BlobSize=-1
CommitRetain=False
Database=127.0.0.1:C:\SopimusPro\Tietokanta\SopimusPro.gdb
ErrorResourceFile=
LocaleCode=0000
Password=masterkey
RoleName=RoleName
ServerCharSet=iso8859_1
SQLDialect=3
Interbase TransIsolation=ReadCommitted
User_Name=sysdba
WaitOnLocks=True
```

6.5.4 Uppdatering av applikationsservern

Till näst uppdateras applikationsservern till en nyare version. Den installerade applikationsservern är av version 3.1. I skrivande stund uppdateras applikationsservern från 3.1 till version 3.3. Uppdateringen sker till genom att köra uppdateringsfilen SopimusPro.msi. Inga lokaliseringar behöver definieras, det ända vi behöver definiera är att skriptet ska vänta medan uppdateringen körs.

Efter att uppdateringen är körd, behöver applikationsservern konfigureras till att köras som en nätverkstjänst under ett nätverkskonto, istället för lokalt. Detta behöver göras så att användare kan logga in med klienterna från sin egen dator via samma nätverk, där servern med Sopro databasen befinner sig. Detta görs via Windows Component Services-menyn, och för att automatiskt göra detta används igen AutoIt. Till skillnad från det tidigare AutoIt-skriptet behöver musen inte användas alls för att konfigurera applikationsservern.

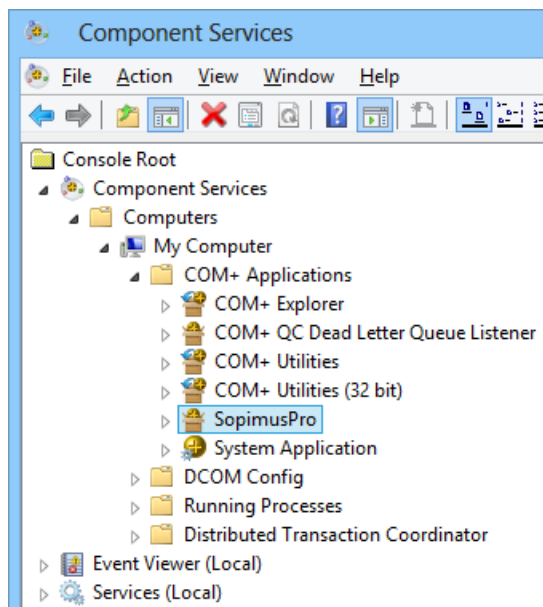
Kod 11. Start av Component Services och navigering till rätt flik

```
#requireadmin
Run ("C:\Windows\System32\dcomcnfg.exe")
WinWaitActive("Component Services")
Send("Component")
WinWaitActive("Component Services")
Send("{RIGHT}")
WinWaitActive("Component Services")
Send("computers")
WinWaitActive("Component Services")
Send("{RIGHT}")
WinWaitActive("Component Services")
Send("my")
WinWaitActive("Component Services")
Send("{RIGHT}")
WinWaitActive("Component Services")
Send("com")
WinWaitActive("Component Services")
Send("{RIGHT}")
WinWaitActive("Component Services")
Send("sopimus")
WinWaitActive("Component Services")
Sleep ( 1000 )
Send("{ALT down}")
Sleep ( 1000 )
Send("{ENTER}")
```

Till skillnad från föregående Autolt-skript, behöver vi definiera skilt att vi har administratörrättigheter när vi startar Component Services. Utan denna definition stannar skriptet utan felmeddelanden i Windows 7 och Windows 8 om Windows User Account Control (UAC) är aktiverat. Felet uppstår fast man skulle starta skriptet med administratörrättigheter. Efter att Component Services är igång och dess menyfönster är aktivt, navigerar vi till egenskaper-fönstret av rätt komponent och inom den ännu till rätt flik. Till skillnad från föregående Autolt-skript undviker vi att navigera med neråt-tangenten, för att antalet objekt i menyn inte nödvändigtvis alltid är det samma, beroende på vad som tidigare har installerats på datorn i fråga.

Vi navigerar till rätt meny med hjälp av att mata in det första ordet av objektet som vi vill välja, efter att objektet är valt utvidgas menyn genom att trycka på högerpil-tangenten. Här märkte jag att WinWaitActive fungerar, även fast Component Services-fönstret färdigt är valt. Det kan bero på att Component Services laddar nästa vy först när vyn öppnas. Efter detta blir objektets undermenyer synliga, och vals med att mata in

objektets första ord. Detta upprepas tre gånger, tills vi kommer till applikationsservern Sopimus Pro under Com+ Applications. Utseendet på denna meny kan ses i figur 5.



Figur 5. Component Services-huvudmenyn

Till näst vill vi komma åt applikationsserverns egenskaper, detta lyckas med snabbkommandot ALT + Enter. Efter att egenskaper-fönstret har öppnats, beordrar vi Autolt att släppa ALT-tangenten och navigera till rätt flik, Identity (identitet). Under denna flik kan man välja användarkontot som kör applikationen. Som standard är alltid den användaren vald som är inloggad för tillfället. Denna användare måste bytas eftersom annars kan ingen logga in med klienterna, om inte användaren är inloggad på datorn, vilket är opraktiskt. Istället väljer vi att applikationen ska köras under ett servicekonto med tillgång till nätverket. På så sätt kan användare logga in med klienterna så länge datorn med applikationsservern är på och är kopplad till nätverket.

Hur detta sker till med Autolt kan man se i kod 12 nedan. Alternativen syns som radio buttons (alternativsknappar), vilket betyder att vi kan komma åt rätt alternativ med TAB-tangenten och därefter skicka snabbkommandot "a" till fönstret för att aktivera Apply-knappen. Genom att trycka på apply bekräftar vi valet, samtidigt stängs egenskaper-fönstret. Efter detta kan vi avsluta Autolt-skriptet genom att avsluta Component Services.

Kod 12. Konfigurering av applikationsservern

```

WinWaitActive("SopimusPro Properties")
Send("{ALT up}")
Sleep ( 1000 )
Send("{CTRLDOWN}")
Sleep ( 1000 )
Send("{TAB}")
Sleep ( 1000 )
Send("{TAB}")
Sleep ( 1000 )
Send("{CTRLUP}")
Sleep ( 1000 )
Send("{TAB}")
Sleep ( 1000 )
Send("{DOWN}")
Sleep ( 1000 )
Send("{DOWN}")
Sleep ( 1000 )
Send("a")
Sleep ( 1000 )
Send("{ENTER}")
WinWaitActive("SopimusPro Properties")
WinClose("Component Services")

```

Efter detta lägger vi igång applikationsservern. Detta sker vanligtvis i samband med installationen av applikationsservern, men eftersom vi avslutar installationen vid IP-adressfönstret, behöver vi lägga igång applikationen själv. Detta lyckas med kommandot `net start "Borland Socket Server"`. Net kommer från det engelska ordet network (nätverk) och används för att hantera nätverksresurser. Eftersom applikationsservern fungerar som en Windows service, räcker det inte att använda kommandot `start`. (SS64,1999-2013)

6.6 Klienterna

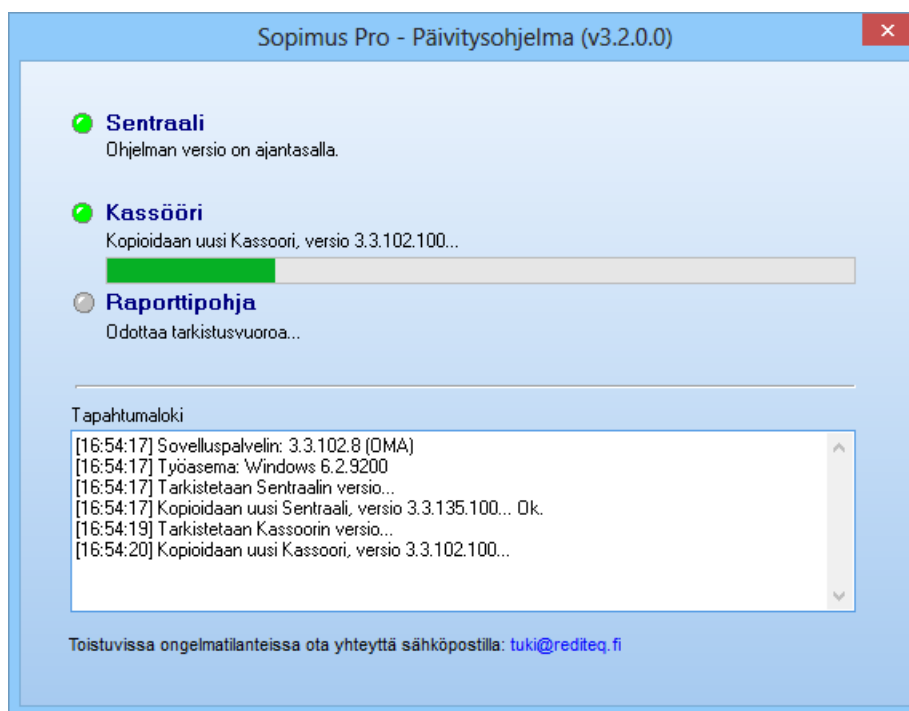
Efter att applikationsservern är konfigurerad, fortsätter skriptet med att installera användargränssnitten eller klienterna för Sopro. Med installationspaketet kommer två olika gränssnitt som används för olika ändamål, Sentraali (Centralen) samt Kassööri (Kassören).

Med installationspaketet följer även med uppdateringsapplikationen för klienterna, som anropas varje gång man loggar in med Sentraali eller Kassööri. Applikationen granskar de nuvarande versionerna samt botten för rapporter som används i Sentraali. Ifall en nyare version hittas av de tre alternativen, uppdateras de med hjälp av applikationen av sig

själv, så länge datorn har nätverkskontakt med Rediteqs uppdateringstjänst samt att användaren har skrivrättigheter till Sopimus Pro-mappen för att ersätta de gamla versionerna.

Än en gång är installationsfilen för klienterna av typen Inno Setup, och samma kommandon och switchar används som tidigare. Svarsfilen innehåller inte heller något nytt. Klienterna installeras i en egen mapp med namnet Client i SopimusPro-mappen. För att man ska kunna logga in med klienterna, behöver de veta var databasen befinner sig. Denna information fås från en initieringsfil som placeras i samma mapp som klienterna finns i. Filen har namnet Yhteys.ini (kontakt.ini) och kopieras till rätt mapp med copy-kommandot. Precis som med SopimusPro-Palvelin.ini-filen som kopierades tidigare, är Yhteys.ini också en fil med så liten filstorlek att switchen /WAIT inte behöver användas i samband med kommandot.

Uppdateringen anropas när man loggar in, men detta är inget krav för uppdateringen, uppdateringen lyckas också med att uppdateringsapplikationen Paivitys.exe körs. Tyvärr så stängs inte uppdateraren efter att uppdateringen är färdig. Därför startas den inte med switchen /WAIT och istället används samma kommandon som användes vid installation av applikationsservern, dvs. kommandot ping och taskkill. Utseendet på uppdateraren kan ses i figur 6.



Figur 6. Uppdateringsvyn

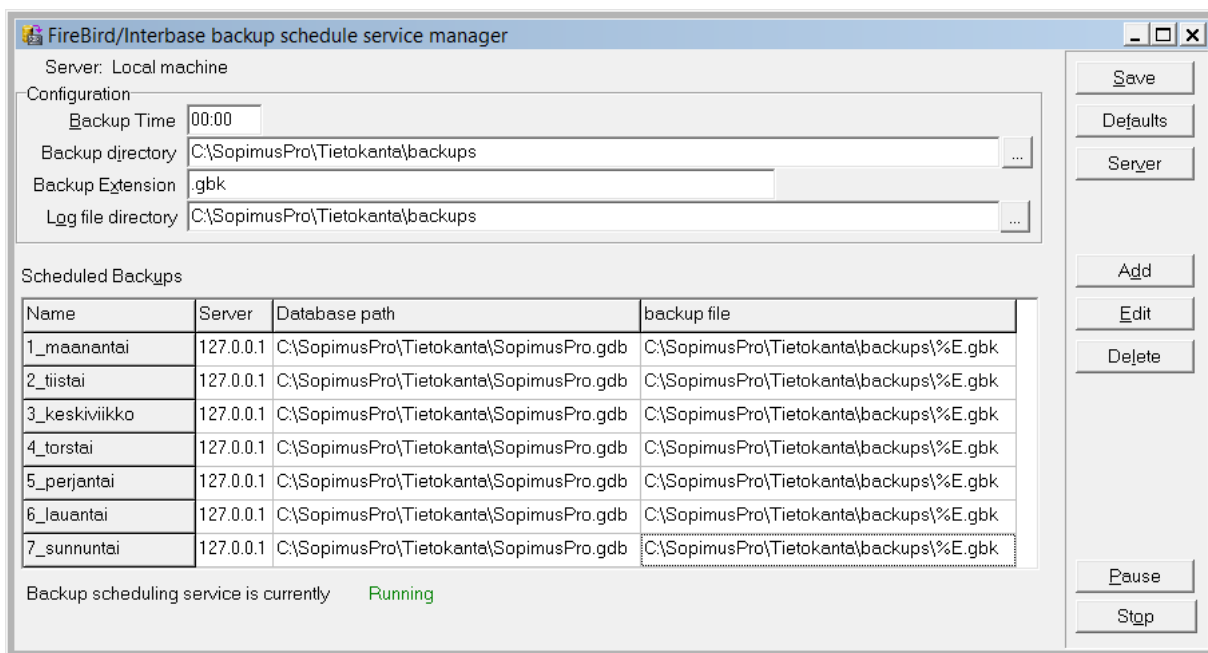
6.7 Time To Backup

Det sista som återstår att installeras är säkerhetskopieraren av databasen. Applikationen som används för detta är Time To Backup (Dags att säkerhetskopiera). Rätt konfigurerad beordrar Time To Backup Firebird varje dag vid en vald tidpunkt att ta en säkerhetskopia av databasen och packa den i en fil med filändelsen gbk för att spara utrymme och samtidigt kryptera innehållet. För att kunna packa upp och återställa databasen används en komponent som kommer med i Firebirdinstallationen, GBAK. Time To Backup installeras som en service, med följer också användargränssnittet för att konfigurera applikationen.

Time To Backup-installationsfilen är också av typen Inno Setup, men installationen skiljer sig en aning från de tidigare installationerna. Av en okänd orsak, kunde installationsdestinationen inte läsas från svarsfilen. Installationen skapade också en genväg till applikationen, fast det definieras i svarsfilen att genvägen ska lämnas bort. Det hjälpte inte heller att inkludera switchen /NOICONS (inga ikoner) i installationskommandot. Däremot lyckas definieringen av installationsdestinationen med switchen /DIR="C:\SopimusPro\TimetoBackup".

6.7.1 Konfigurering av Time To Backup

Time To Backup konfigureras med hjälp av AutoIt. Detta fungerar bra eftersom musen inte behövs alls och all navigering lyckas med hjälp av TAB och snabbkommandon. Utseendet på användargränssnittet kan ses i figur 7.



Figur 7. Användargränssnittet till Time To Backup

I huvudmenyn definieras först tidspunkten i klockslag för säkerhetskopieringen. Detta brukar vanligtvis ske under en tidpunkt som programmet inte används, t.ex. på natten. Till näst definieras destinationen för säkerhetskopior, vilket är en mapp inuti databas-mappen. Filändelsen gbk behöver också definieras på nästa rad. Säkerhetskopieringen loggas, dessa loggfiler sparas också i samma mapp som själva säkerhetskopior.

Efter detta behöver man ännu konfigurera varje veckodag skilt. För varje dag behöver man skilt definiera följande punkter:

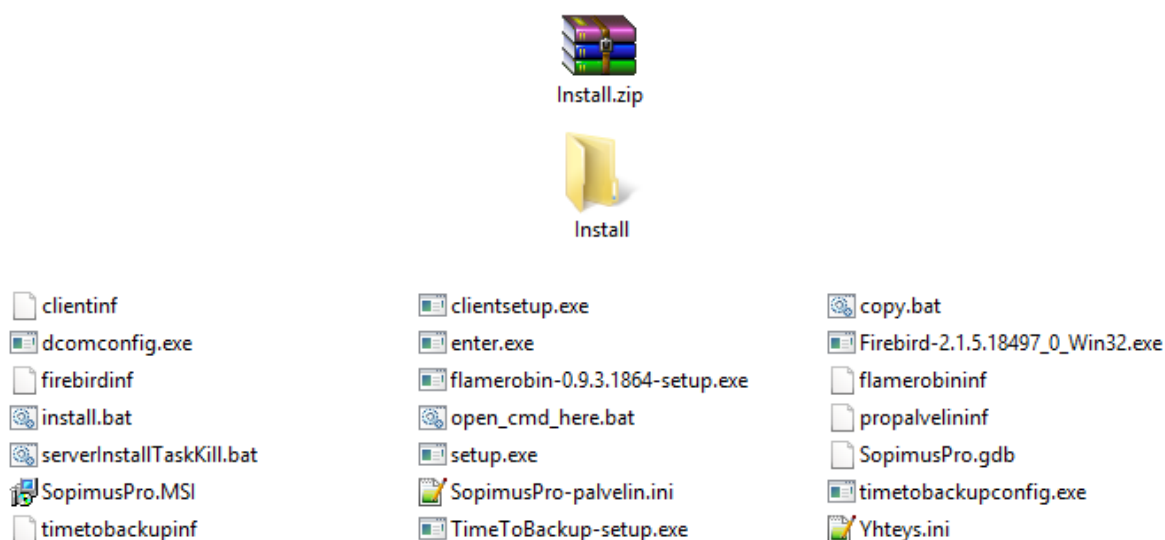
- namn för identifiering
- server
- databasens lokalisering på datorn
- Filnamn för säkerhetskopian samt destination
- Användarnamnet Sysdba
- Sysdbas lösenord
- Namn för loggfilen
- Veckodagen när säkerhetskopieringen körs

Detta behöver alltså göras skilt sju gånger för varje veckodag och för hand tar det en god stund. I hela automationsskriptet är det just här var automatiseringen av installationen och konfigurationen underlättar och sparar mest tid.

Efter att säkerhetskopieraren har under en veckas tid gjort sju säkerhetskopior, ersätts de gamla kopiorna under följande vecka av nya säkerhetskopior, detta innebär att mappen innehåller som mest sju säkerhetskopior och loggfiler.

7 Användning av skriptet

För att skriptet ska kunna köras från början till slut, behöver alla nödvändiga installation- och batchfiler finnas färdigt på datorn dit Sopro installeras. Dessa filer kopieras nerpackade i en zipfil till datorn. När kopieringen är gjord extraheras innehållet av zipfilen på datorn i en valfri mapp eller på skrivbordet. Fram dyker upp en ny mapp med namnet Install, som innehåller alla nödvändiga filer. Storleken på mappen är beroende av databasens storlek, resten av filerna kräver inte mycket utrymme. Innehållet av zipfilen Install.zip kan ses i figur 8.



Figur 8. Innehållet av Install.zip

Efter att innehållet är extraherat startar man CMD med administratörrättigheter för att försäkra sig om att man faktiskt har rättigheter för att utföra alla kommandon som finns i skriptet. Med CMD öppet navigerar man till Install-mappen där filerna finns. Jag har inkluderat i paketet en batchfil som automatiskt omdirigerar användaren till den mapp var batchfilen körs, dvs. Install-mappen. Om man väljer detta sätt, rekommenderas att denna fil också körs med administratörrättigheter, på så sätt ärver även själva skriptet dessa rättigheter.

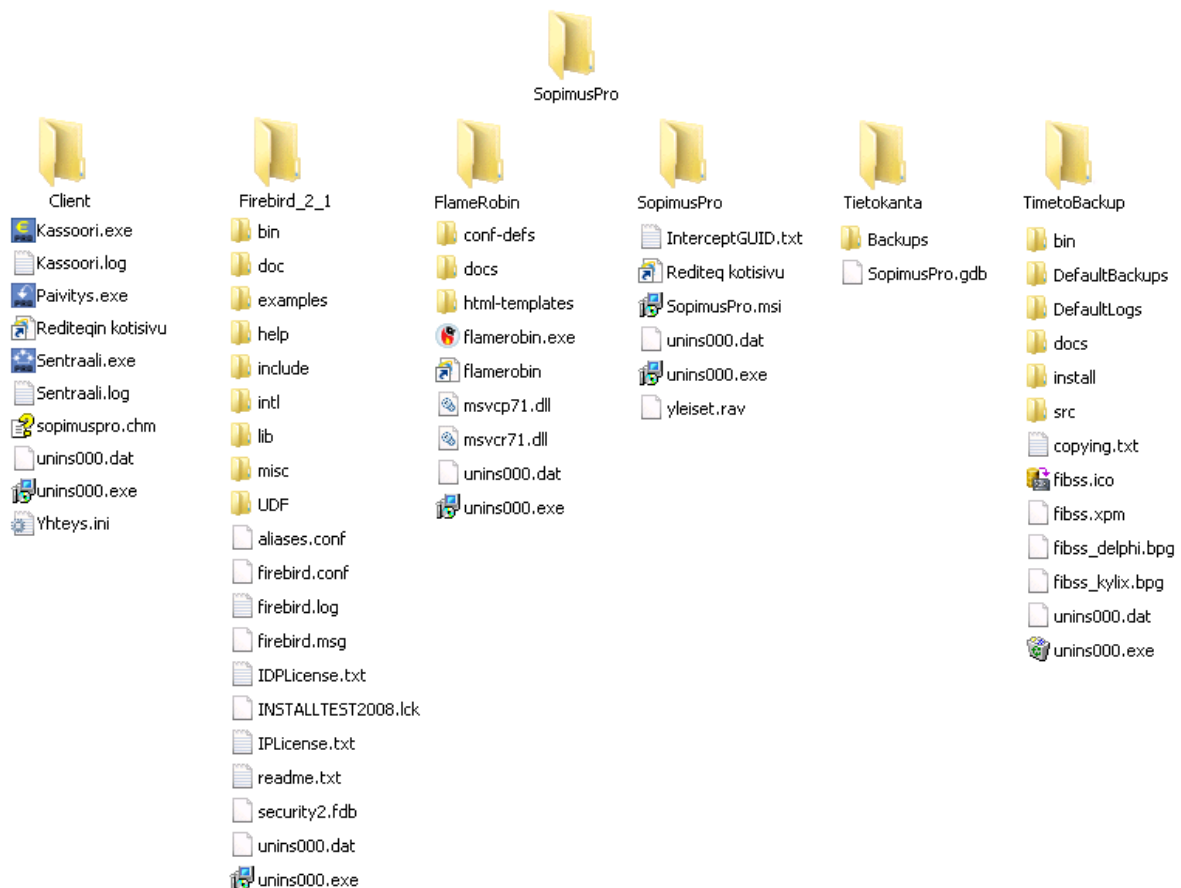
7.1 Igång läggning av skriptet

Efter att man har navigerat till rätt mapp, kan själva skriptet läggas igång. Detta sker genom att skriva i CMD huvudbatchfilens namn dvs. `install.bat`, och skriptet börjar. När skriptet läggs igång, börjar Firebird inte installeras automatiskt. Istället pausas skriptet genast med kommandot `pause`. Samtidigt syns ett meddelande i CMD-fönstret där användaren frågas ifall man vill installera Sopro. Detta har jag inkluderat ifall skriptet läggs igång av misstag. Skriptet kan avbrytas när som helst genom att stänga CMD-fönstret.

Genom att skriva `Echo (Eko)` framför en mening visas meningen som man har skrivit i CMD fönstret. På så sätt kan man inkludera meddelanden som t.ex. `"Steg 1 klart"` i skriptet så att användare har lättare att följa med i vilket skede skriptet är. Om man dock lämnar meddelandet så här, syns ordet `echo` med i fönstret och meddelandet två gånger, var det andra meddelandet kan beskrivas som ett eko. För att bara se själva meddelandet en gång, kan man inkludera `"@"`-tecknet framför kommandot `echo`. Detta gäller dock bara raden i fråga. Om man vill att alla meddelanden i skriptet bara visas en gång i fönstret, kan man inkludera `"Echo off"` i början av skriptet. (SS64, 1999-2013).

När kommandot `pause` används, syns samtidigt meddelandet `"Press any key to continue . .."`. Installationen börjar när användaren trycker på en valfri tangent.

Efter att skriptet har körts och Sopro har blivit installerad på datorn, behövs installationsfilerna inte längre och de kan raderas från datorn. Ifall filerna lämnas kvar av någon orsak på datorn, behöver `install.bat` filen modifieras så att lösenordet för Firebird inte är läsbart någonstans. Mappstrukturen som har blivit skapad i samband med installationen av Sopro kan ses i figur 9. Förutom dessa filer installeras även vissa filer i datorn Windows-mapp, såsom applikationsserverns komponent `scktsrvr.exe` samt initieringsfilen `SopimusPro-palvelin.ini` som berättar för applikationsservern var databasen befinner sig.



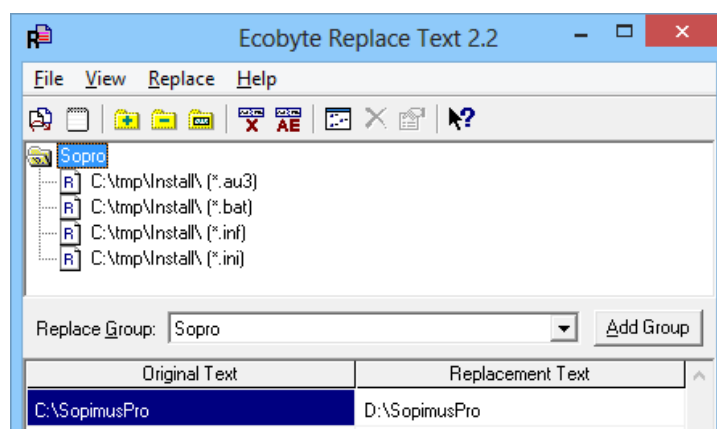
Figur 9. Mappstruktur

7.2 Replace Text

Om inga ändringar görs till skriptet, installeras Sopro i SopimusPro-mappen i datorns C-station. Det kan dock hända att man vill installera Sopro någon annanstans på datorn, kanske på en helt annan hårddiska. Detta betyder att alla nödvändiga destinationsdefinitioner behöver ändras så att de installeras till rätt destination. Detta kan man göra för hand, men för att spara tid har jag använt mig av ett program vid namnet Replace Text.

Med Replace Text kan man definiera den exakta textsträngen eller textsträngarna som man vill ändra. Samtidigt måste man definiera var filerna finns och vilka filformat Replace Text ska gå igenom i den angivna mappen. De nödvändiga filformaten som behöver definieras är bat, inf, ini och au3. Au3 är filändelsen för AutoIt-skript, ifall ändringar görs till dessa filer behöver de ännu kompileras på nytt, detta kräver att AutoIt är installerat på datorn.

Textsträngen som man vill ändra på bör inkludera SopimusPro-mappens namn. T.ex om man vill att Sopro istället installeras på datorns D-station, ändrar man texten C:\SopimusPro till D:\SopimusPro. Om man vill ha Sopro inne i en annan mapp under C-stationen, kan texten ändras t.ex. till "C:\Program Files\SopimusPro". På det här sättet ändras endast texten på de ställen där det behövs, och filer som alltid ska ha samma destination lämnas ifred, t.ex. SopimusPro-palvelin.ini som alltid placeras i datorns Windows-mapp. Eftersom denna destination definieras i koden som C:\Windows, ändrar inte Replace text denna destination.



Figur 10. Användargränssnittet för Replace Text

8 Testning

Den första versionen av skriptet skrevs på en dator med Windows XP 32-bit. Efter att varje fas var färdig, testade jag svarsfilen för respektive installationsfil och gick vidare först efter att jag hade konstaterat att det fungerar. Autolt-skripten prövade jag också med det samma skilt på en dator med Windows 8, var jag märkte problemet med muspekaren som behandlades redan tidigare i arbetet. Detta repeterades i varje fas ända tills hela installationsprocessen var inkluderad i installationen.

Efter att hela installationen kunde köras från början till slut på datorn med Windows XP, prövade jag att köra skriptet på en annan dator med Windows 7 32-bit. Dessa versioner har en hel del år mellan varandra vilket betyder att de inte fungerar på exakt samma sätt, vilket ledde till att vissa ändringar var nödvändiga. T.ex. då användaren frågas om databasens IP-adress vid installationen av applikationsservern, dyker det upp två fönster. I Windows XP stängdes det andra fönstret automatiskt, medan det inte gjorde det i Windows 7. En liknande situation uppstod vid konfigureringen av applikationsservern. När

man trycker på Apply efter att ändringarna är gjorda, stängs fönstret automatiskt, medan det inte i nyare versioner inte längre gör det. Båda dessa problem åtgärdades med Autolt.

När Sopro installeras hos en kund, har kunden vanligtvis en serverversion av Windows installerad på datorn i fråga. Därför var det också viktigt att testa installationsskriptet även på Windows serverversioner. Eftersom jag inte hade tillgång till dessa versioner använde jag mig i testandet av Windows Azure. Windows Azure är Microsofts molntjänst ägnat till webbutvecklare för att bygga, distribuera och hantera sina applikationer effektivt över nätet. Jag ville använda mig av Windows Azure pga. en av dess egenskaper, nämligen att man kan lätt skapa en virtuell maskin. I samband med skapandet av den virtuella maskinen definieras uppgifter så som datorns namn samt vad för operativsystem maskinen ska ha. Efter detta skapas den virtuella maskinen samt läggs igång, sedan kan man med hjälp av fjärranslutning ansluta sig till den virtuella datorn. På detta sätt skapade jag både en virtuell version av Windows Server 2008 R2 och Windows Server 2012 och flyttade över installationsfilerna till dem, och sedan körde skriptet.

I skrivande stund har skriptet testats och körts på följande Windows versioner:

- Windows XP 32-bit
- Windows 7 32-bit
- Windows 7 64-bit
- Windows 8 64-bit
- Windows Server 2008 R2 64-bit
- Windows Server 2012 64-bit

9 Slutdiskussion

Detta projekt har varit givande och utvecklandet av skriptet har varit roligt. När jag började planera projektet i början av februari 2013, hade jag och min uppdragsgivare inte bestämt någon tidsgräns för arbetet. Detta resulterade i att jag tog god tid på mig och hade även emellanåt många veckor långa pauser i utvecklingen av projektet. Delvis pga. varierande antal jobbtimmar i veckan. Sommaren påverkade även en hel del på motivationen av den skriftliga delen av arbetet. Jag upplevde även stunder där jag tvekade om jag någonsin skulle få arbetet färdigt.

Men nu, nio månader efter att projektet startades är det äntligen klart, och det känns väldigt bra att ha fått skapa något själv från början till slut. Jag har lärt mig en hel del nytt, vilket var ett av projektets mål. Samtidigt känner jag nu till installationsprocessen av Sopro mycket bättre än vad jag gjorde när projektet började, vilket också var ett mål för projektet. Både jag själv och uppdragsgivaren Petri Huttunen är nöjd med resultatet och skriptet har jag redan haft användning av inom mina egna arbetsuppgifter hos Rediteq Ab. Det finns även möjligheter för vidareutveckling av skriptet. Sopro har en hel del webbmoduler vars installation möjligtvis också kunde skriptas, databaskonfigurationer för nya kunder är också något som kunde inkluderas.

Något jag kunde ha behandlat bättre var episoden med användning av Autolt för första gången vid konfigureringen av FlameRobin. Jag borde ha tagit bättre reda på ifall bytandet av lösenordet kunde skötas effektivare. När problemet med muskoordinaterna dök upp, kan man argumentera om ifall utvecklandet av Autolt-skriptet i fråga var slöseri med tid. Själv tycker jag att det sist och slutligen inte var det, eftersom jag samtidigt lärde mig hur Autolt fungerar och jag hade nytta av denna kunskap vid de andra tillfällena där Autolt användes.

Den här sorten av automatiserad skriptning hade jag inte tidigare erfarenhet av, och liknande har inte heller behandlats under kurserna i skolan. Jag känner att jag har under projektets tid lärt mig en hel del nytt, och jag har även börjat tillämpat automatiska skript för att sköta dagliga datoruppgifter, såväl som på jobbdatorn som på hemdatorn.

10 Avslutning

Den första versionen av skriptet var klart före jag påbörjade arbetet av den skriftliga delen. Samtidigt under skrivandet gjorde jag förbättringar till skriptet och testade annorlunda lösningar.

Slutresultatet av arbetet är ett fungerande automatiserat skript som installerar alla nödvändiga komponenter av Sopimus Pro från installationens början till dess slut utan användarens inblandning. Det känns fint att jag nådde alla de utsatta målen och att skriptet faktiskt fungerar. Den färdiga versionen av install.bat kan ses i bilaga 1.

Källförteckning

Autolt (1999-2012). *Autolt: Automation and scripting language*

<http://www.autoitscript.com/site/autoit/> (hämtad 26.08.2013)

Autolt (1999-2012). *Autolt: Functions*

<http://www.autoitscript.com/autoit3/docs/functions/> (hämtad 04.03.2013)

Bhardwaj, P.K. (2006). *How to Cheat at Windows System Administration Using Command Line Scripts*. Massachusetts: Syngress Publishing, Inc.

Computer Hope (2013). *Information on batch files*

<http://www.computerhope.com/batch.htm> (Hämtad 3.10.2013)

Ecobyte (u.å.). *Replace Text*

<http://www.ecobyte.com/replacetext/> (hämtad 15.05.2013)

Firebird (2000-2013). *About Firebird*

<http://www.firebirdsql.org/en/about-firebird/> (hämtad 30.08.2013)

Firebird (2000-2013). *Classic or Superserver?*

<http://www.firebirdsql.org/manual/gsg15-classic-or-super.html> (hämtad 23.10.2013)

Heymann, S (2009, 20 november). *GSEC — Firebird User Administration*

<http://www.destructor.de/firebird/gsec.htm> (hämtad 5.6.2013)

Microsoft Developer Network (2013, 12 oktober). *COM+ (Component Services)*

[http://msdn.microsoft.com/en-us/library/windows/desktop/ms685978\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms685978(v=vs.85).aspx)

(hämtad 1.11.2013)

Rediteq (2013). *Om företaget*

<http://rediteq.fi/index.php/se/foretaget/om-foretaget> (hämtad 29.07.2013)

Sourceworld (u.å.). *Unattended, A Windows deployment system: Unattended/Silent Installation Switches for Windows Apps*

<http://unattended.sourceforge.net/installers.php> (Hämtad 8.02.2013)

Sourceworld (u.å.). *Unattended: Inno Setup Command Line Parameters*

http://unattended.sourceforge.net/InnoSetup_Switches_ExitCodes.html (Hämtad 11.02.2013)

Sqllly (1999-2011). *Time To Backup - Firebird/InterBase Backup Scheduler.*

<http://www.sqlly.com/timetobackup.html> (Hämtad 1.10.2013)

SS64 (1999-2013). *An A-Z Index of the Windows CMD command line*

<http://ss64.com/nt/> (hämtad 6.5.2013)

TechNet (2013). *Scripting with Windows PowerShell*

<http://technet.microsoft.com/en-us/library/bb978526.aspx> (hämtad 01.08.2013)

TechTerms (2013). *Shell*

<http://www.techterms.com/definition/shell> (hämtad 14.11.2013)

Van der Woude, R (2012, 21 april). *Rob van der Woude's Scripting Pages: Batch file examples – wait*

<http://www.robvanderwoude.com/wait.php> (hämtad 20.4.2013)

Van der Woude, R (2012, 24 juli). *Rob van der Woude's Scripting Pages: Batch files - How To ... Display and Redirect Output*

http://www.robvanderwoude.com/battech_redirection.php (hämtad 23.4.2013)

Windows (2013). *Command Prompt: frequently asked questions*

<http://windows.microsoft.com/is-IS/windows-vista/Command-Prompt-frequently-asked-questions> (Hämtad 3.10.2013)

Figurförteckning

Figur 1. Autoexec.bat i MS-DOS version 6.....	5
Figur 2. Sopro Centralen inloggningsruta.....	10
Figur 3. Egenskaper-vyn för en MSI fil.....	12
Figur 4. FlameRobin.....	20
Figur 5. Component Services-huvudmenyn.....	30
Figur 6. Uppdateringsvyn.....	32
Figur 7. Användargränssnittet till Time To Backup.....	34
Figur 8. Innehållet av Install.zip.....	35
Figur 9. Mappstruktur.....	37
Figur 10. Användargränssnittet för Replace Text.....	38

Kodförteckning

Kod 1. Skapandet av en ny mapp.....	15
Kod 2. Firebirdinf – svarsfilen för Firebird.....	16
Kod 3. Installering av Firebird.....	17
Kod 4. flamerobininf – svarsfilen för FlameRobin.....	18
Kod 5. Startandet av FlameRobin och inloggning med Autolt-skriptet.....	20
Kod 6. Användning av musen i Autolt.....	22
Kod 7. Bytandet av Firebird lösenordet med GSEC.....	23
Kod 8. Copy.bat.....	24
Kod 9. Installering av applikationsservern.....	26
Kod 10. serverInstallTaskKill.bat.....	27
Kod 11. Start av Component Services och navigering till rätt flik.....	29
Kod 12. Konfigurering av applikationsservern.....	31

Bilagor:

Bilaga 1:

Bilden visar den slutliga versionen av install.bat.

```

1 @echo off
2 echo [%date% %time%] Asenna Sopimus Pro?
3 pause
4 mkdir C:\SopimusPro
5 echo [%date% %time%] SopimusPro kansio luotu
6 echo [%date% %time%] Firebird asennetaan...
7 start "" /wait Firebird-2.1.5.18497_0_Win32.exe /LOADINF=firebirdinf /silent /NORESTART
8 echo [%date% %time%] Firebird asennettu.
9 echo [%date% %time%] Flamerobin asennetaan...
10 start "" /wait flamerobin-0.9.3.1864-setup.exe /LOADINF=flamerobininf /silent
11 echo [%date% %time%] Flamerobin asennettu.
12 echo [%date% %time%] Konfiguroidaan Firebird kayttaja...
13 start "" /wait C:\SopimusPro\Firebird_2_1\bin\gsec -user SYSDBA -password masterkey -modify sysdba -pw newPassword
14 echo [%date% %time%] Firebird kayttaja konfiguroitu.
15 mkdir C:\SopimusPro\Tietokanta
16 echo [%date% %time%] SopimusPro Tietokanta kansio luotu.
17 echo [%date% %time%] Kopioidaan tietokanta...
18 start "" /wait copy.bat
19 echo [%date% %time%] Tietokanta kopioitu.
20 echo [%date% %time%] Sopimus Pro palvelin asennetaan...
21 start "" enter.exe
22 start "" setup.exe /LOADINF=propalvelininf /silent
23 PING 1.1.1.1 -n 1 -w 15000 >NUL
24 start "" /wait /B serverInstallTaskKill.bat
25 echo [%date% %time%] Palvelin asennettu.
26 copy SopimusPro-palvelin.ini C:\WINDOWS
27 echo [%date% %time%] SopimusPro-palvelin.ini kopioitu.
28 echo [%date% %time%] Sovelluspalvelin paivitetaan...
29 start "" /wait SopimusPro.MSI
30 echo [%date% %time%] Sovelluspalvelin paivitetty.
31 echo [%date% %time%] Sovelluspalvelin konfiguroidaan...
32 start "" /wait dcomconfig.exe
33 echo [%date% %time%] Sovelluspalvelin konfiguroitu.
34 net start "Borland Socket Server"
35 echo [%date% %time%] Borland Socket Server kaynnistetty.
36 echo [%date% %time%] Clientit asennetaan...
37 start "" /wait clientsetup.exe /LOADINF=clientinf /silent
38 echo [%date% %time%] Clientit asennettu.
39 copy Yhteys.ini C:\SopimusPro\Client
40 echo [%date% %time%] Yhteys.ini kopioitu
41 start "" C:\SopimusPro\Client\Paivitys.exe
42 PING 1.1.1.1 -n 1 -w 30000 >NUL
43 taskkill /F /IM Paivitys.exe
44 echo [%date% %time%] TimetoBackup asennetaan...
45 start "" /wait TimeToBackup-setup.exe /LOADINF=timetobackupinf /silent /DIR="C:\SopimusPro\TimetoBackup"
46 echo [%date% %time%] TimetoBackup asennettu.
47 mkdir C:\SopimusPro\Tietokanta\Backups
48 echo [%date% %time%] SopimusPro Backups kansio luotu.
49 echo [%date% %time%] TimetoBackup konfiguroidaan...
50 start "" /wait timetobackupconfig.exe
51 echo [%date% %time%] TimetoBackup konfiguroitu.
52 echo [%date% %time%] VALMIS
53 pause
54 exit

```