

# MES i processindustri

-ett projekt med fokus på energi



---

**Patrik Kjellberg**  
**Viktor Stefansson**

Division of Industrial Electrical Engineering and Automation  
Faculty of Engineering, Lund University

# MES i processindustri

## -Ett projekt med fokus på energi



LUNDS  
UNIVERSITET

Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg  
Avd. för industriell elektroteknik och automation

Bachelor thesis:  
Patrik Kjellberg  
Viktor Stefansson

© Copyright Patrik Kjellberg, Viktor Stefansson

LTH Ingenjörshögskolan vid Campus Helsingborg  
Lunds universitet  
Box 882  
251 08 Helsingborg

LTH School of Engineering  
Lund University  
Box 882  
SE-251 08 Helsingborg  
Sweden

Tryckt i Sverige  
Avdelningen för industriell elektroteknik och automation  
Lunds universitet  
Lund 2014

## Sammanfattning

Implementering av ett MES (Management Execution System) gjordes på en server tillhörande en fabrik hos Oatly AB. Arbetet gjordes på uppdrag från Schneider Electric AB i Helsingborg. Största delen av tiden spenderades på ett kontor på Schneider Electric, där ny programvara inom energianalys studerades, testades och slutligen implementerades via fjärrstyrning mot Oatlys server.

Med hjälp av MES behandlades mätvärden och beräkningar från ett SCADA-projekt som sedan lagrades i en SQL-databas. Sista steget var att publicera databasens innehåll i form av översiktliga rapporter för att lätt kunna analysera energiförluster i fabriken process. Efter att uppgiften var löst, drogs slutsatsen att med hjälp av fler mätpunkter i fabriken skulle energianalysen kunna utvecklas betydligt mer då det hade funnits fler möjligheter att leta flaskhalsar i alla delprocesser istället för att endast kontrollera de uppenbara storförbrukarna. Investeringar i fler givare och komponenter kan alltså bidra till bättre analys och optimering av den totala energiförbrukningen.

Nyckelord: MES, SCADA, SQL, Rapporter, Energianalys

## **Abstract**

Implementation of a MES (Management Execution System) was made on a server that belongs to a factory owned by Oatly AB. The task was assigned by Schneider Electric AB in Helsingborg. Most of the time was spent in a Schneider Electric office, where new software in the field of energy analysis were studied, tested and finally implemented by remote controlling Oatly's server.

By using MES, measurements and calculations from a SCADA project was managed, then stored in a SQL database. The final step was to publish the database content as casual reports, as a mean to make further energy analysis easier. After the task was completed, the conclusion was made that a greater amount of tags would make further development of the energy analysis possible, since a greater amount of flaws could be spotted in all different parts of the entire process, instead of just checking the obvious energy consumers. Investments in more sensors and components could contribute to better analysis and optimization of the entire energy consumption.

Keywords: MES, SCADA, SQL, Reports, Energy analysis

## **Förord**

Oönskade energiförluster är ett stort problem i många industrier. Både när det gäller säkerhet, kostnad och varaktighet. Detta har skapat en marknad för energianalys och har givit oss ett uppdrag att genomföra som examensarbete.

Examensarbete utförs hos Schneider Electric Helsingborg och resultatet blir en implementering på Oatly AB i Landskrona.

Vi vill tacka vår handledare Peter Holmberg på Schneider Electric för mycket bra stöd, samt delaktiga på Oatly för hjälp och vägledning under projektets gång. Även stort tack till vår examinator Mats Lilja för hjälp med rapportskrivning.

## Terminologi

<b>Ampla</b>	Citects egna MES för analys av SCADA-system.
<b>CICODE</b>	Programspråk för styrning av Citects SCADA-system. Utgår ifrån Pascal.
<b>Citect</b>	Programvaruutvecklare (Ägs av Schneider Electric).
<b>Field</b>	En variabel i Ampla som innehåller ett mätvärde.
<b>IIS</b>	Internet Information Service. Serverprogramvara från Microsoft för Internetbaserade tjänster.
<b>KPI</b>	Key Performance Indicator, ett representerande och uträknat värde.
<b>MES</b>	Manufacturing Execution System. Används för analys av styrsystem.
<b>OEE + E</b>	Overall Equipment Efficiency + Energy. Ett uträknat värde som beskriver effektivitet för flera faktorer.
<b>OPC DA</b>	Skickar data mellan OPC servern och OPC klienten.
<b>PLC</b>	Ett programmerbart styrsystem.
<b>VPN</b>	Virtual Private Network. Används för att skapa en säker förbindelse mellan två punkter.
<b>SCADA</b>	Supervisory Control And Data Acquisition. System för övervakning och styrning av processer.
<b>SQL</b>	Programspråk för att hämta och modifiera data i en relationsdatabas.
<b>SSRS</b>	SQL Server Reporting Service. Program för att skapa och hantera rapporter.
<b>Tag</b>	En variabel som avläses från exempelvis givare eller ventil.
<b>REG_EDIT</b>	En del av Windows Registry, som är en hierarkisk databas för systemkonfigurationer.

# Innehållsförteckning

<b>1 Inledning</b> .....	<b>1</b>
<b>1.1 Bakgrund</b> .....	<b>1</b>
<b>1.2 Nulägesbeskrivning</b> .....	<b>1</b>
<b>1.3 Schneider Electric AB</b> .....	<b>1</b>
<b>1.4 Oatly AB</b> .....	<b>1</b>
<b>1.5 Syfte och Mål</b> .....	<b>2</b>
<b>1.6 Problemformuleringen</b> .....	<b>2</b>
<b>1.7 Avgränsningar</b> .....	<b>2</b>
<b>2 Metod</b> .....	<b>3</b>
<b>2.1 Källkritik</b> .....	<b>3</b>
<b>3 Teknisk bakgrund</b> .....	<b>3</b>
<b>3.1 Vijeo Citect (SCADA) v7.40</b> .....	<b>3</b>
<b>3.2 SQL 2012 Server</b> .....	<b>3</b>
<b>3.3 (MES) - Ampla Express v6.0</b> .....	<b>4</b>
3.3.1 Production.....	4
3.3.2 Downtime.....	4
3.3.3 Metrics.....	4
3.3.4 Energy.....	4
3.3.5 Ampla Server Manager.....	6
3.3.6 Ampla Studio.....	6
3.3.7 Ampla Production Analyst.....	7
<b>3.4 SQL Service Reporting Services</b> .....	<b>7</b>
<b>4 Genomförande</b> .....	<b>8</b>
<b>4.1 Sätta upp server</b> .....	<b>8</b>
<b>4.2 Inlärningsperiod</b> .....	<b>8</b>
<b>4.3 Implementering på Oatly</b> .....	<b>10</b>
<b>4.4 Rapporter i Ampla Express</b> .....	<b>11</b>
<b>5 Resultat</b> .....	<b>12</b>
<b>5.1 SCADA</b> .....	<b>12</b>
<b>5.2 Anslutning mellan SCADA och Ampla</b> .....	<b>15</b>
<b>5.3 Anslutning mellan SCADA och PLC</b> .....	<b>17</b>
<b>5.4 Ampla Express</b> .....	<b>18</b>
<b>5.5 Rapporter</b> .....	<b>21</b>
<b>6 Diskussion och Slutsatser</b> .....	<b>27</b>
<b>7 Utvecklingsmöjligheter</b> .....	<b>29</b>
<b>8 Referenser</b> .....	<b>30</b>
<b>9 Appendix</b> .....	<b>31</b>
<b>9.1 Appendix A</b> .....	<b>31</b>
<b>9.2 Appendix B</b> .....	<b>33</b>
<b>9.3 Appendix C</b> .....	<b>36</b>
<b>9.4 Appendix D</b> .....	<b>36</b>





# 1 Inledning

## 1.1 Bakgrund

En stor del av dagens industrier har ett stort informationsflöde som kan utnyttjas på bättre sätt med rätt verktyg. Ett sådant verktyg skulle idag kunna vara ett Manufacturing Execution System (MES) som behandlar mätvärden för att lättare hitta brister i processer. Inom industrin finns oftast endast tillgång till de allra senaste mätvärdena vilket gör att det är svårare att se förändringar när t.ex. ny maskinvara eller ändrat arbetssätt införs. MES är utvecklat med syftet att länka ihop affärssystemet med industrins kontrollsystem.

## 1.2 Nulägesbeskrivning

På Oatly har man fram tills nu endast haft tillgång till information om total energiåtgång i sin fabrik. Total energiförbrukning säger inte så mycket om effektiviteten i fabriken då analysering av enskilda processer och maskiner är nödvändigt för att finna potentiella flaskhalsar.

De tekniskt ansvariga på Oatly är i dagsläget säkra på att disksystemet och värmebehandlingarna är storförbrukarna i fabriken men för att bli helt övertygade krävs att man kan se problemet i fråga, svart på vitt. Personalen på Oatly vill därför i framtiden ha fortsatt kontroll över förbrukningen med ett användarvänligt gränssnitt med överskådliga rapporter som sammanställer mätvärden.

## 1.3 Schneider Electric AB

Schneider Electric är ett globalt företag som tillhandahåller tekniska lösningar i allt ifrån industriell automation till kraftdistribution. Även tillverkare av eluttag och diverse andra elinstallationsprodukter i många hem. Cirka 200.000 anställda inom koncernen.

## 1.4 Oatly AB

Grundidén för framtagning av havrebaserade mjölkprodukter utvecklades under 1980-talet av forskare vid Lunds universitet. 2001 lanserades varumärket Oatly och 2006 bytte företaget namn till Oatly AB med egen produktionsanläggning i Landskrona, där havrebaserade livsmedel tillverkas. Produkterna kan nyttjas både av laktos och glutenintoleranta. Har idag 50 anställda och har ökat sin omsättning med 20% årligen sedan start.

## 1.5 Syfte och Mål

Syftet med detta projekt är att Oatly vill få tydligare information om delar av sin industri för att i framtiden med underlaget framtagit av MES visa var förbättringar kan genomföras. Eftersom Oatly är intresserade av hur energianvändningen ser ut kommer fokus att ligga på de delprocesser där det finns möjlighet att implementera MES-verktyget och sammanställa den hämtade informationen till överskådliga rapporter.

Målsättningen för projektet kommer att vara som följer:

- Driftsätta en ny server hos Oatly, dvs. det ska finnas en fullt fungerande server med alla nödvändiga funktioner installerade för att vi ska kunna använda våra applikationer.
- Implementera MES-verktyget utifrån Oatlys önskemål. (Se Nulägesbeskrivning).
- Sammanställa informationen från verktyget i rapporter.

## 1.6 Problemformuleringen

Problemet i projektet är att med en server tillhandahållen av Oatly driftsätta ett fullt fungerande MES-verktyg utifrån de önskemål som finns hos Oatly. För att uppnå detta måste därför följande problem lösas:

- Hur man implementerar diverse programvaror och samtliga komponenter som krävs för detta på servern.
- Var och när MES verktyget skall hämta in data.
- Hur man går tillväga för att skapa rapporter som skall vara tillgängliga inifrån MES-applikationen.

## 1.7 Avgränsningar

Största fokus kommer att läggas på att undersöka endast en del av industrin då det i dagsläget saknas rätt mätvärden för att göra en fullskalig implementation av MES. Till en början skall det läggas fokus på Oatlys disksystem och värmebehandlingar för att sedan om tid finns, gå in och undersöka övriga delar i processen. Projektet anses vara klart när Oatly kan plocka ut rapporter ur systemet som ger en översikt över angivna delar.

## 2 Metod

Eftersom Ampla Express tidigare inte är implementerat inom Sveriges gränser ägnades mycket tid åt att studera installation- och användarmanualer (se referens [2,3]) för de programvaror användes under arbetets gång. Vid komplikationer söktes information i första hand på Citects KnowledgeBase (se referens [4]). Även Citects support användes flitigt i början för att konfigurera Windows-komponenter, *IIS* och dylikt för att ens få igång programvarorna. Servern fjärrstyrdes ifrån Helsingborg både med TeamViewer och fjärranslutning med *VPN*. Handedaren fanns även tillgänglig vid konfigurering. Det hölls även möten med IT- och tekniskt ansvariga på Oatly för att få tillgång till backuper av nuvarande system och för att få information om hur styrsystemet är uppbyggt. Kriterierna för arbetet är uppfylla då Oatly godkänt resultatet där de har möjlighet att hämta ut rapporter ur MES-verktyget uppbyggda efter deras önskemål.

### 2.1 Källkritik

Under examensarbetets gång användes manualer (installation och användning) som hör till de olika verktygen och dessa följer vanligtvis med när man hämtar hem programvarorna. Vissa uppgifter i källorna kan ibland vara lite väl kortfattade vilket medför att man får söka information i Citects Knowledgebase och utifrån fakta man kommit över pröva sig fram till önskat resultat. Dessa källor kan man lita på eftersom de använda manualerna tillhandahålls av produktutvecklarna som inte kan tjäna på att ge felaktig information.

## 3 Teknisk bakgrund

För att kunna få kunskaper om vad som är väsentligt när man vill optimera energiförbrukning i en fabrik, så användes olika verktyg för simulering, analys och rapportering.

### 3.1 Vijeo Citect (SCADA) v7.40

En programvara för övervakning och styrning av processer. Denna programvaran användes för att simulera grafiska fabriker där mätvärden också simulerades för att sedan plockas ut med MES-verktyget. Efter inlärningsfasen kördes MES-verktyget mot ett icke simulerat *SCADA*-projekt med reella mätvärden.

### 3.2 SQL 2012 Server

För att lagra all data ifrån *SCADA*-systemet används en *SQL*-server. All information skrivs/hämtas därefter från den servern.

### 3.3 (MES) - Ampla Express v6.0

Ampla Express är ett Manufacturing Execution System från Citect som underlättar analys och optimering av dagens SCADA-system. Kriterierna för ett MES-system är att de enligt MESA (Manufacturing Enterprise Solution Association) skall innehålla några eller alla av de 11 funktioner som finns definierade. Ampla Express är en billigare licens som använder sig av fyra funktioner men kan byggas ut med Plant Operation Ampla som är en full licens, där man använder alla 11 funktioner. I Ampla benämns dessa funktioner som moduler och Ampla Express använder sig av följande:

#### 3.3.1 Production

En modul som hanterar värden från produktion ex: kvantitet, kasserat, kWh, temperatur och CO<sub>2</sub>. Dessa så kallade *fields* kan man kombinera för att exempelvis jämföra energiförbrukning med antalet producerade produkter.

#### 3.3.2 Downtime

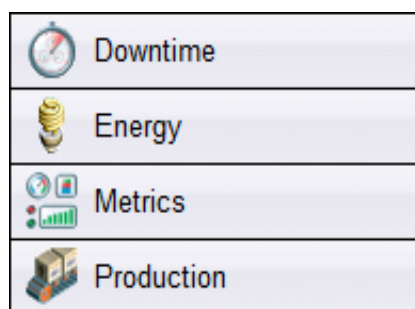
En modul som hanterar dödtider i processen. Exempelvis driftstopp, slö drift, service eller maskinfel. Dödtider är väsentliga för analys då dessa talar om att produktionen påverkats av tillfälligheter i systemet.

#### 3.3.3 Metrics

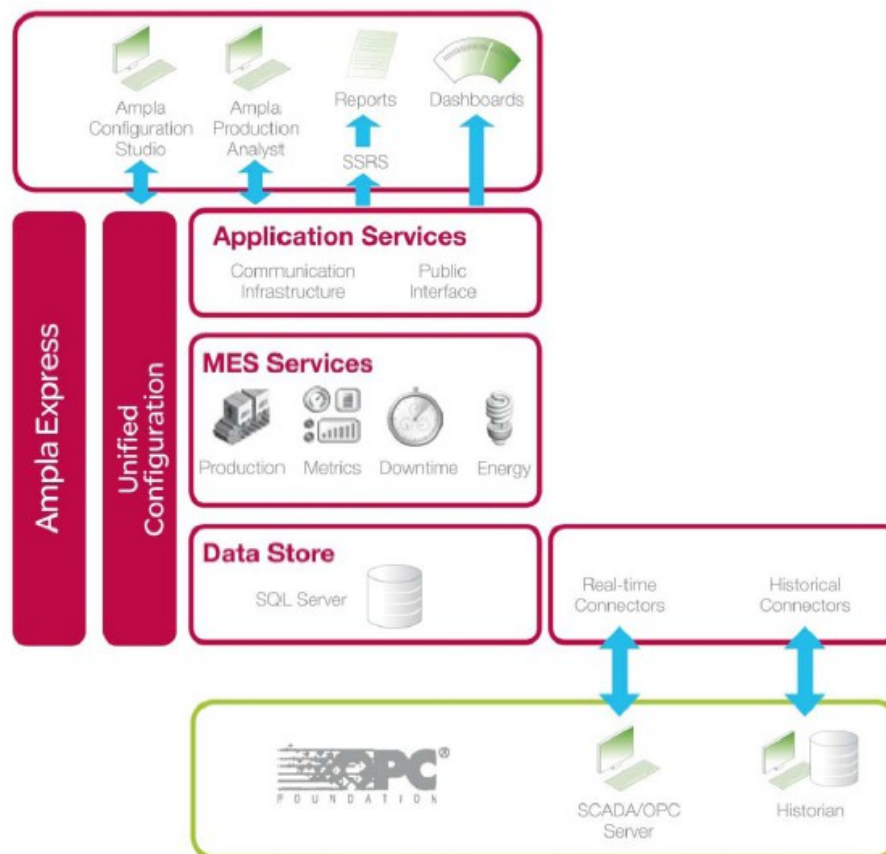
I Metrics använder man sig av OEE + E genom att definiera tillgänglighet, prestanda, kvalitet och energieffektivitet. OEE + E är ett värde som ger en snabb och effektiv översikt över hela processen. Det blir även lätt att överskåda effektiviteten under en önskad period.

#### 3.3.4 Energy

Energimodulen beräknar överkonsumtion där den jämför förväntad energiåtgång med riktig förbrukning. Om överkonsumtion finns, loggas detta och kopplas till orsak.



Figur 3.1 – De fyra modulerna



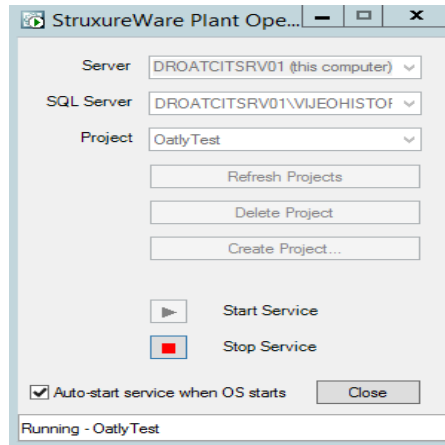
Figur 3.2 – Översikt över arkitekturen för Ampla Express

Arkitekturen i Ampla Express kan delas upp i tre olika lager

- Datalager – Detta lager består av en abstrakt datamodell för datalagring och subsystem för tag-baserade anslutningar
- Affärslager – Detta lager ligger på servern som implementerar MES-funktioner i form av moduler.
- Visualiserings- och integrationslager – Detta lager är menat för användaren att integrera med Ampla. Visualisering av data och analysverktyg i form av tabeller, diagram och rapporter.

### 3.3.5 Ampla Server Manager

Detta är vårt ”Runtime interface” för Ampla Express. Här anges vilken server som applikationen ska köra på och vilken SQL-server som skall hantera data. Ampla Express kan dock endast köra ett projekt åt gången på servern.

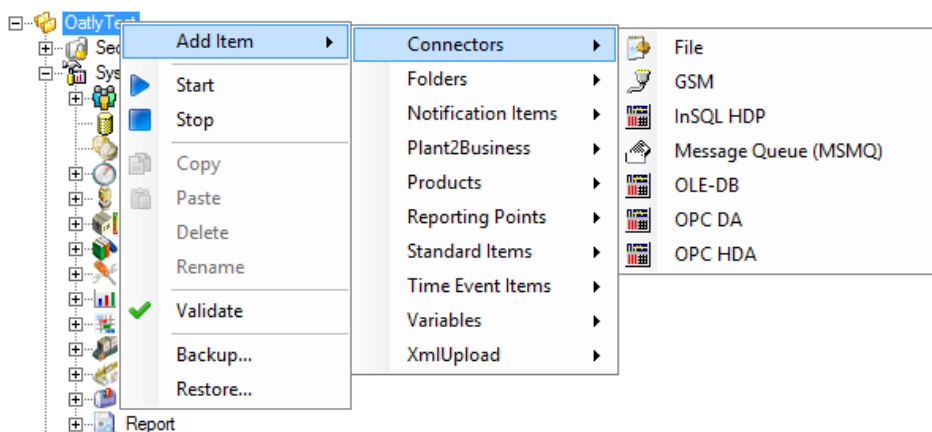


Figur 3.3 – Ampla Service Manager

### 3.3.6 Ampla Studio

Ampla Studio är klienten där ens skapade projekt hanteras. Här definieras var, när och hur data skall fångas för vidare behandling och analys. Här konfigureras anslutningen till de externa datakällorna. Ampla Express kan hantera tre olika källor:

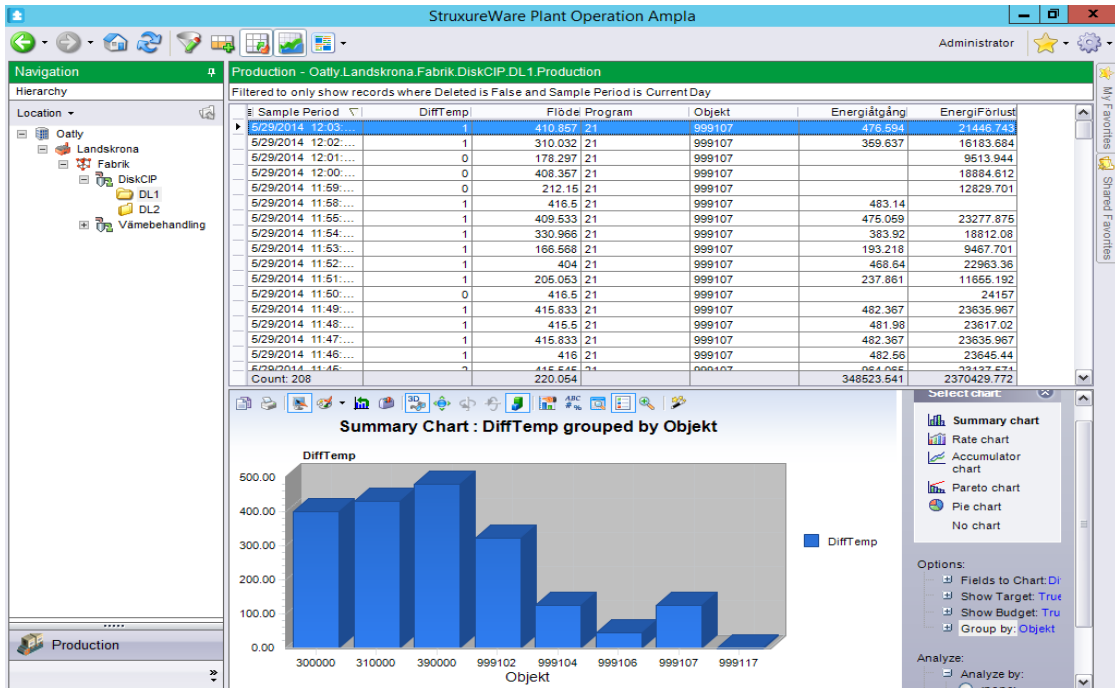
- OLE DB databaser där tidigare inhämtad data kan returneras.
- ”Runtime State” eller ”Instantaneous Value” som hämtas från citectSCADA eller *OPC DA2.0*
- Historical Data access (OPC HDA). Detta är en anslutning där historisk information om taggar hämtas.



Figur 3.4 – Ampla Connectors

### 3.3.7 Ampla Production Analyst

Ampla Production Analyst är verktyget man använder för att visualisera och analysera data. Användaren kan komma åt applikationen både som desktop- och webbaserad klient.



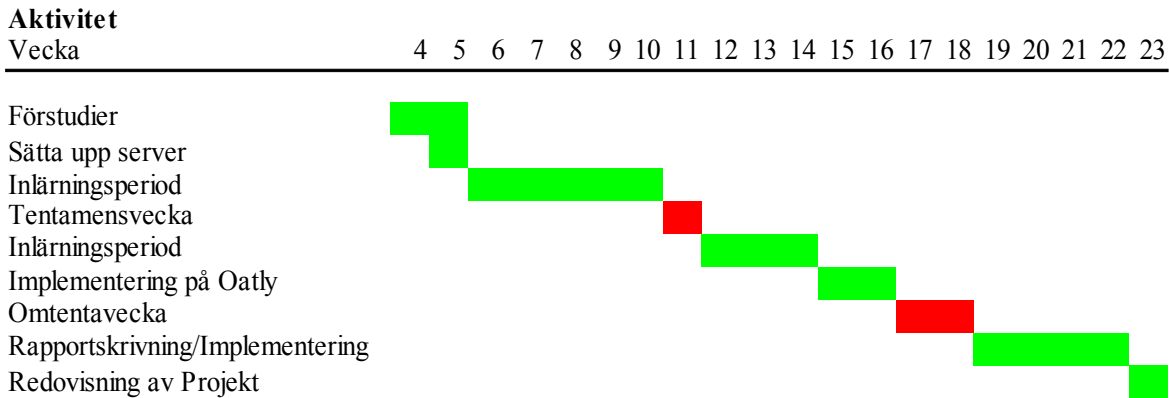
Figur 3.5 – Ampla Production Analyst

### 3.4 SQL Service Reporting Services

SSRS används för att skapa en rapportserver där man i sin tur skapar och hanterar rapporter. Rapporterna innehåller diagram och tabeller med mätvärden sammanställda ur MES med lättöverskådlig design utifrån kundens önskemål.



## 4 Genomförande

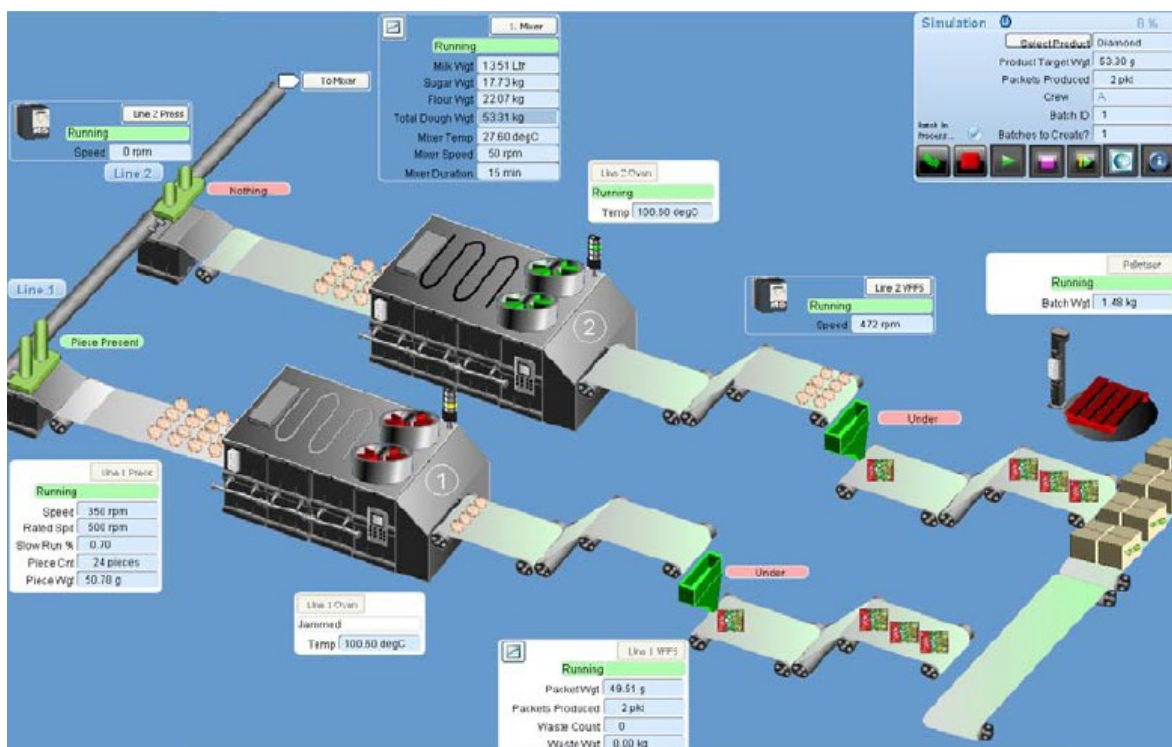


### 4.1 Sätta upp server

När examensarbetet väl drog igång så fanns en server på Schneider Electric's kontor som var ämnad för arbetet, med operativsystemet Windows Server 2008 R2. Tyvärr havererade den redan nästa morgon på grund av ett fel i moderkortet. Efter ett par dagar blev en ny server tillgänglig, dock ej installerad. Till denna server tillkom operativsystemet MS Small Business Server 2011 som i grunden är Windows Server 2008 med skillnaden att SQL server och MS Sharepoint ingår. Tyvärr skapade Sharepoint stora problem vid installation av de applikationer som var nödvändiga. En hel del andra inställningar fick göras manuellt då applikationerna, främst Ampla Express, ej är menat för denna version av operativsystemet. I detta steg togs det hjälp av Schneider Electric's supportavdelning för Citects programvaror. Problem i form av IIS, REG\_EDIT, behörighet till diverse applikationsmappar osv. Problemen löstes i form utav mejlutväxlingar och fjärrstyrning av servern från deras sida. Servern närmade sig felfri funktionalitet men det fanns fortfarande en del kompatibilitetsproblem. Lyckligt löstes detta med tillgång till den senaste versionen av Ampla Express. Därefter var applikationerna redo för användning.

### 4.2 Inlärningsperiod

Under inlärningsperioden användes ett SCADA-projekt som ingick i utgivet utbildningsmaterialet. Projektet simulerade en fabrik som producerade och paketerade kakor.



Figur 4.1 – Översikt över Fabrik i SCADA-projektet

Här lärde man sig alltså hur man väljer ut rätt mätvärden och lagrar dem som fields, för att sedan antingen publicera dem vid särskilda loggningstillstånd och/eller räkna ut exempelvis formler för energianalys.

Uppgiften i utbildningsmaterialet går ut på att bygga upp ett Ampla-projekt som hanterar och sammanställer mätvärden från det aktuella SCADA-projektet. Målet med utbildningsmaterialet var att få fram ett OEE+E-värde på ugnen i fabriken. OEE+E är ett nyckeltal för att mäta produktionseffektivitet. För att få fram detta använder man sig av *KPIs* i form av Availability, Performance, Quality och Energy Efficiency. Värdena som behövs för att få fram detta hämtas ifrån Production, Downtime och Energi modulerna.

Availability – Detta värde får man fram genom att ta kvoten av faktisk körtid och planerad körtid.

Performance – Kvoten av producerad mängd (vid optimal drift) och den riktiga mängden.

Quality – Kvoten av producerad (icke kasserad) mängd och total producerad mängd.

Energy Efficiency – Kvoten av önskad energianvändning och faktisk energianvändning.

Produkten av dessa värden blir vårt värde på OEE+E. Detta är ett nyckeltal för att mäta produktionseffektivitet.

Under den andra inlärningsperioden (se Gantt-schema) tillhandahöll Oatly en kopia av deras SCADA-projekt utan tillhörande databas. För att kunna hantera ett ”riktigt” projekt som inte är gjort efter ett utbildningsmaterial, så var det dags att påbörja simuleringarna.

I SCADA-projektet finns det scheman för varje enskild delprocess i fabriken. I projektet fanns ca 5000 olika *taggar* att välja mellan, här söktes det grundligt efter taggar som verkade vara intressanta och väsentliga (värden som passar energiformler, exempelvis temperatur) för att kunna göra en duglig energianalys. Till varje tag finns ett namn, en kortbeskrivning och ett unikt ID, vilket gör det möjligt att skumma igenom tagglistan utan att spendera flera timmar på en sådan sökning.

Före taggarna sattes i bruk, undersöktes de i grafiska scheman som finns i SCADA-projektet. Här kunde man oftast se tydlig logik och funktion mellan dem olika komponenter som taggarna representerade.

Eftersom det varken fanns en fylld databas eller rörliga mätvärden, så simulerades ändringar i SCADA-projektet genom att implementera kod i Cicode. Kodningen med Cicode lät oss exempelvis aktivera/avaktivera ventiler, tilldela givare värden efter sinuskurver och lägga in fördröjningar i processen.

Efter ett par veckor gav simuleringarna tillräckligt med kött på benen för att hålla ett möte med tekniskt ansvariga på Oatly. På detta möte presenterades idéer för var i processen man kunde hitta de stora flaskhalsarna och hur man går till tillväga för att logga detta. Bland annat var en idé att logga körningstider för olika motorer. De ansvariga på Oatly gav feedback på idéerna och därefter diskuterades vilka prioriteringar som skulle göras. ”Lågt hängande frukt” är ett begrepp som dök upp och blev i princip resultatet av mötet, dvs. eftersom det redan hade konstaterats att det saknades en hel del givare, så var det bättre att fokusera på dem som var tillgängliga. Här efter sållades de viktigaste taggarna ut och fokus lades på att simulera disksystemet.

### **4.3 Implementering på Oatly**

Efter simuleringsperioden hade tillräckligt med kompetens erhållits för att få tillgång att fjärrstyra en server som låg placerad på Oatly, så att det gick att kommunicera med styrsystemet i realtid. På denna server installerades åter igen samtliga nödvändiga applikationer (se Teknisk bakgrund). Nu togs det emot direkta signaler från PLC:n till SCADA-projektet, för att ta emot riktiga mätvärden till Ampla-projektet och på så vis avgöra om applikationen gjorde korrekta beräkningar. Här konstaterades att olika problem kan uppstå, exempelvis beroende på i vilken ordning sekvenser körs enligt styrsystemet. Även fördröjningar kom att påverka mätvärden, vilket uppdagade sig tydligt i Ampla Production Analyst då ett klassiskt symptom visade sig som ”negativ energiförbrukning”.

Till kommunikationen med styrsystemet användes en programvara som heter Siemens Industrial Ethernet. Programmet låter PLC:n skicka och ta emot information till och från ytterligare en server. Detta är ett väldigt smidigt sätt att kommunicera. Dock bör försiktighet tas i akt när man fjärrstyr eftersom man trots allt kan ändra i fabriken styrning utan att se konsekvenserna. Vid denna tidpunkt var det därför inte längre tillåtet att ändra i SCADA-projektet, utan endast i Ampla-applikationerna.

I och med att det riktiga styrsystemet inte var en simulering, så kördes inte varenda maskin och sekvens dygnet runt. Detta innebar att applikationerna för arbetet behövde vara igång dygnet runt för att inte värdefull data skulle gå förlorad.

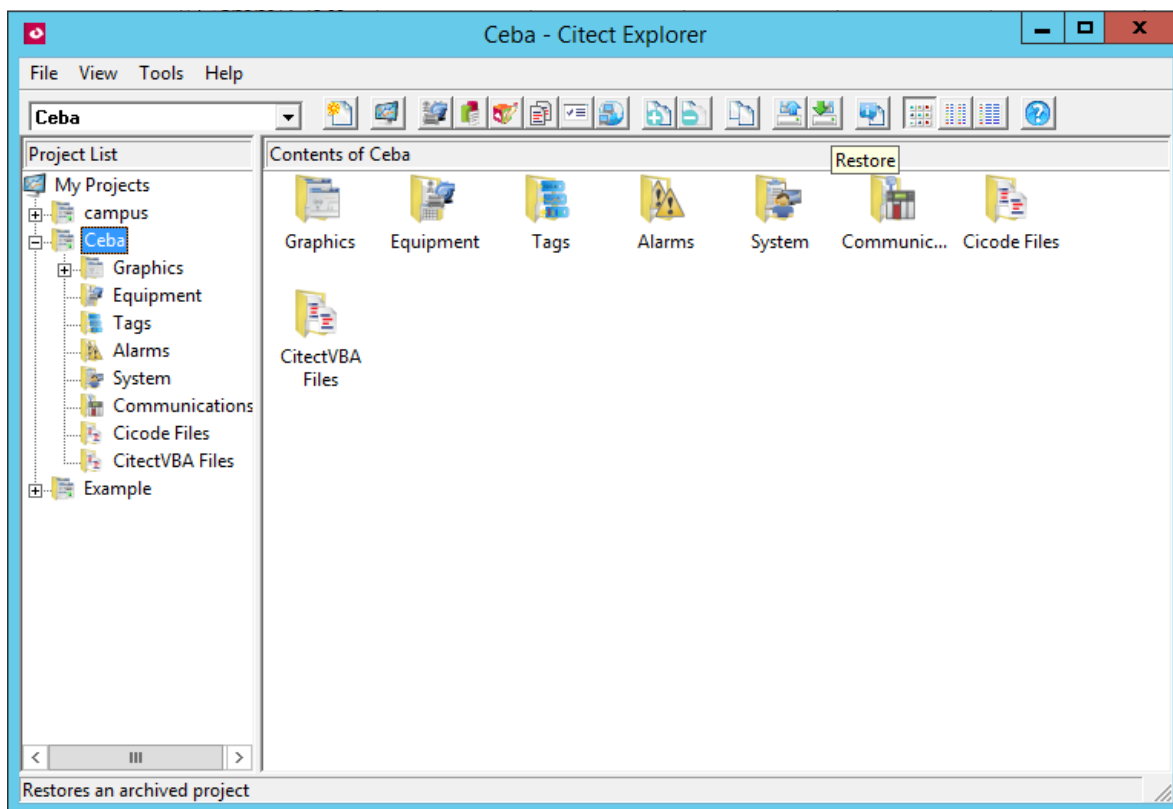
#### **4.4 Rapporter i Ampla Express**

I Ampla Express använder man sig av rapporter för att få en sammanställning av värdena. Färdiga rapporter finns inlagda i programmet men dessa var ej lämpliga då designen var anpassad efter utbildningsmaterialet och visade exempelvis grafer som inte var av intresse för examensarbetet. Vi byggde därför upp nya rapporter och använde endast dem färdiga rapporterna som skal. Rapporterna skapades med SSRS i SQL Server 2013 Report Builder v3.0 där vi följde en guide (se referens [1]) för att komma åt värdena i form av fält och parameter ifrån Ampla Production Analyst.

## 5 Resultat

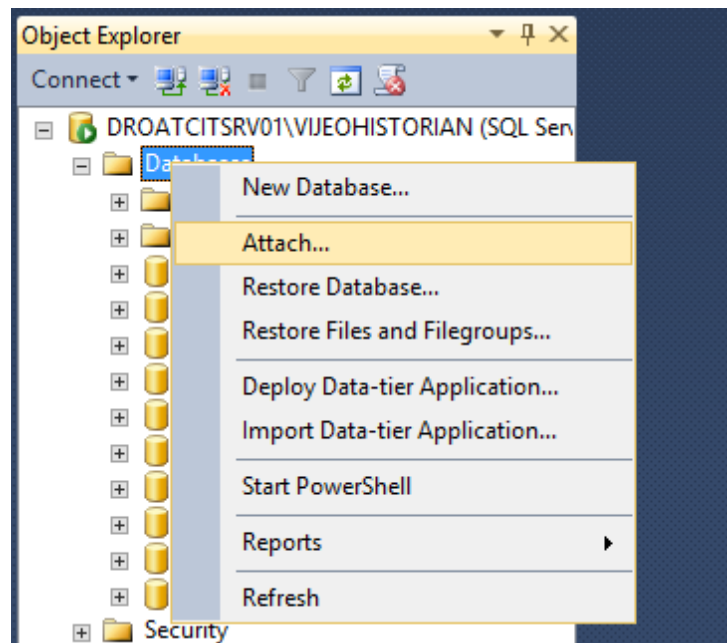
### 5.1 SCADA

Ett SCADA-projekt som är konstruerat för fabriken styrning, läses in (som .ctz-fil) med Citect Explorer. Med detta projekt undersöks samtliga delprocesser i Citects SCADA-fönster där scheman visar motorer, pumpar, ventiler, givare och andra komponenter som är angivna med ett tag-id. När det bestäms att ett mätvärde för en viss tag ska hämtas, så antecknas dess ID och därefter används kommandot Ctrl+F i Citect Project Editor. Här kontrolleras att taggen är definierad så som den anges i det grafiska schemat. För att kunna hantera ett mätvärde måste taggen vara en "Variable Tag" eftersom taglistan även innehåller bl. a "Alarm Tags" som inte är mätvärden. Det är viktigt att kontrollera vilken datatyp (Integer, String etc.) taggen är definierad som och vilken enhet mätvärdet uttrycks i, för att kunna hantera den rätt i Ampla Studio.



*Figur 5.1 – Citect Explorer*

En databas innehållande värden för ett ögonblick läggs till i SQL Server Management Studio. I vårt fall är databasen en .mdf-fil, vilket innebär att ”attach-funktionen” används istället för ”restore”. Nu kommer samtliga taggar (som inte är tomma i databasen) ha ett fixt värde som är taget från verkligheten, men som inte kommer uppdateras efterhand.



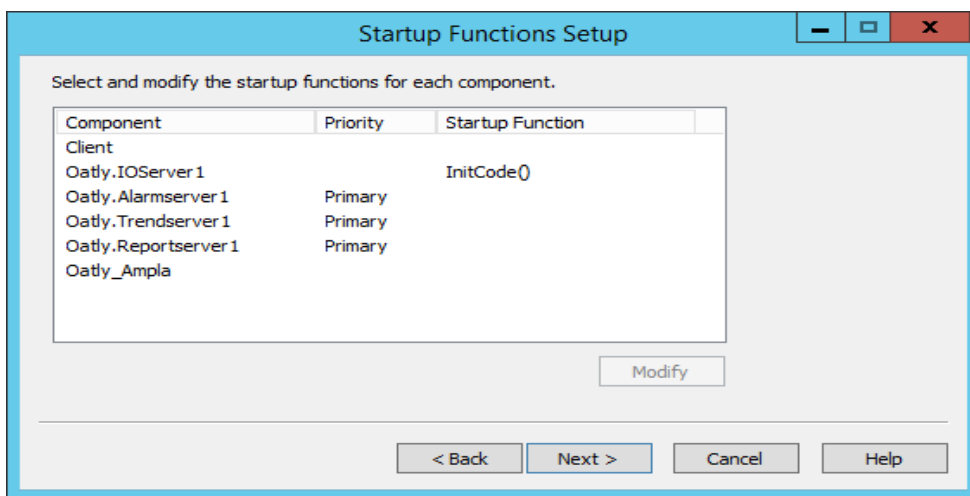
*Figur 5.2 – Sql Server Management Studio*

Under simulering används ”Memory Mode” till SCADA-projektet eftersom ingen hårdvara (PLC) är konfigurerad för systemet. Memory Mode låter SCADA skriva till ett cacheminne, istället för hårdvara. När SCADA körs mot en riktig PLC och använder en aktiv databas, så stängs Memory Mode av.

Under simuleringen användes Cicode för att programmera styrning av exempelvis disksystemet. Här implementeras kod i form av metoder som startar/stänger av diskprogram, byter diskprogram och öppnar ventiler ut mot avloppet.

För att kontinuerligt ändra olika mätvärden implementerades while-loopar där tillhörande variabler tilldelas värden på slumpvisa sinuskurver eller helt enkelt ökande värden. Sleep-funktionen låter oss lägga fördröjningar så att *MES*-verktyget hinner göra samplings under ett rimligt tidsintervall. Se Appendix A för exempel på kod.

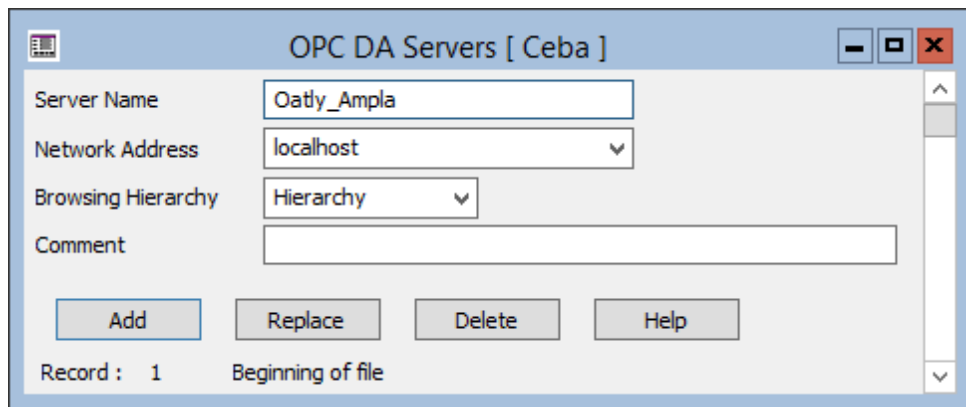
För att tala om för SCADA-projektet att den skapade cicode-funktionen skall användas vid uppstart körs "Computer Setup Wizard" ifrån Citect Explorer där det i Client anges att .ci-filen är en "StartUp Function".



*Figur 5.3 – StartUp Functions Setup*

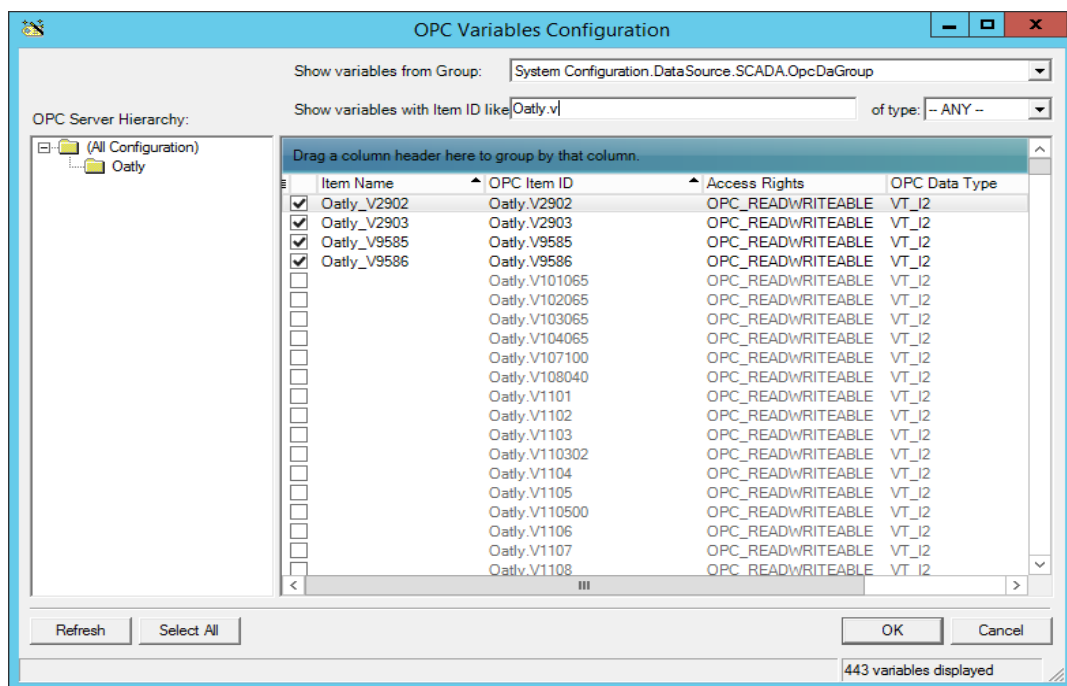
## 5.2 Anslutning mellan SCADA och Ampla

Ampla har tre olika anslutningsmöjligheter (se Figur 3.4). Eftersom en 64-bitars version av Ampla körs så skapar vi en OPC DA Server i Citect Explorer där kommunikationen mellan SCADA och Ampla hanteras.



Figur 5.4 – OPC DA Servers

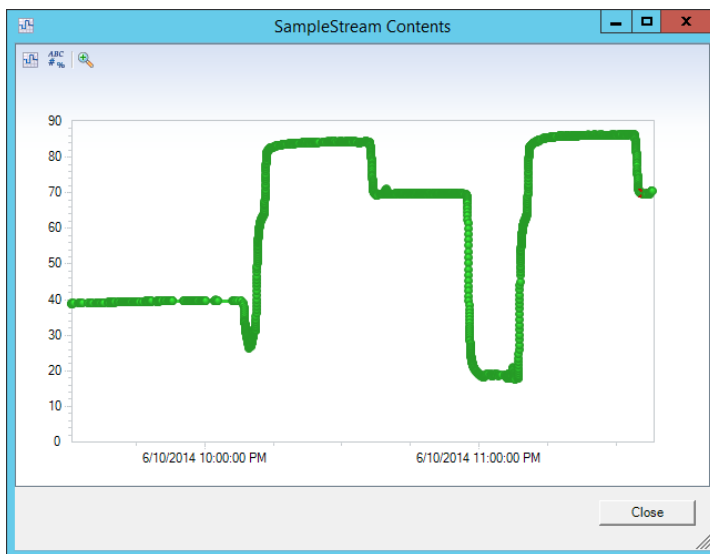
Nu skapas en OPC DA-anslutning i Ampla Studio där adressen som server befinner sig på anges. Med denna anslutning finns nu möjlighet att lägga in de taggar som man vill samla data ifrån.



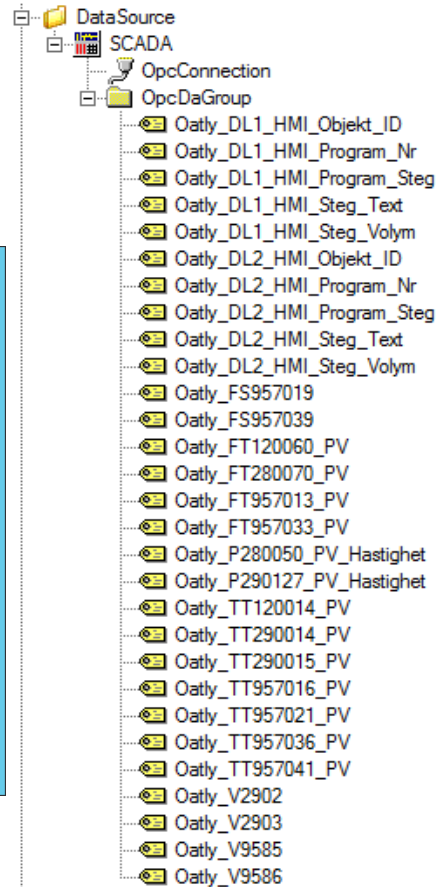
Figur 5.5 – OPC Variable Configuration



I mappen för OPC DA-anlutningen läggs sedan de valda taggarna, vilket gör dem åtkomliga för hantering i Ampla Studio. Ned visas en lista på valda taggar och ett exempel på samplingsvärden som visar de reella värdena för en särskild tag vid viss tidpunkt.



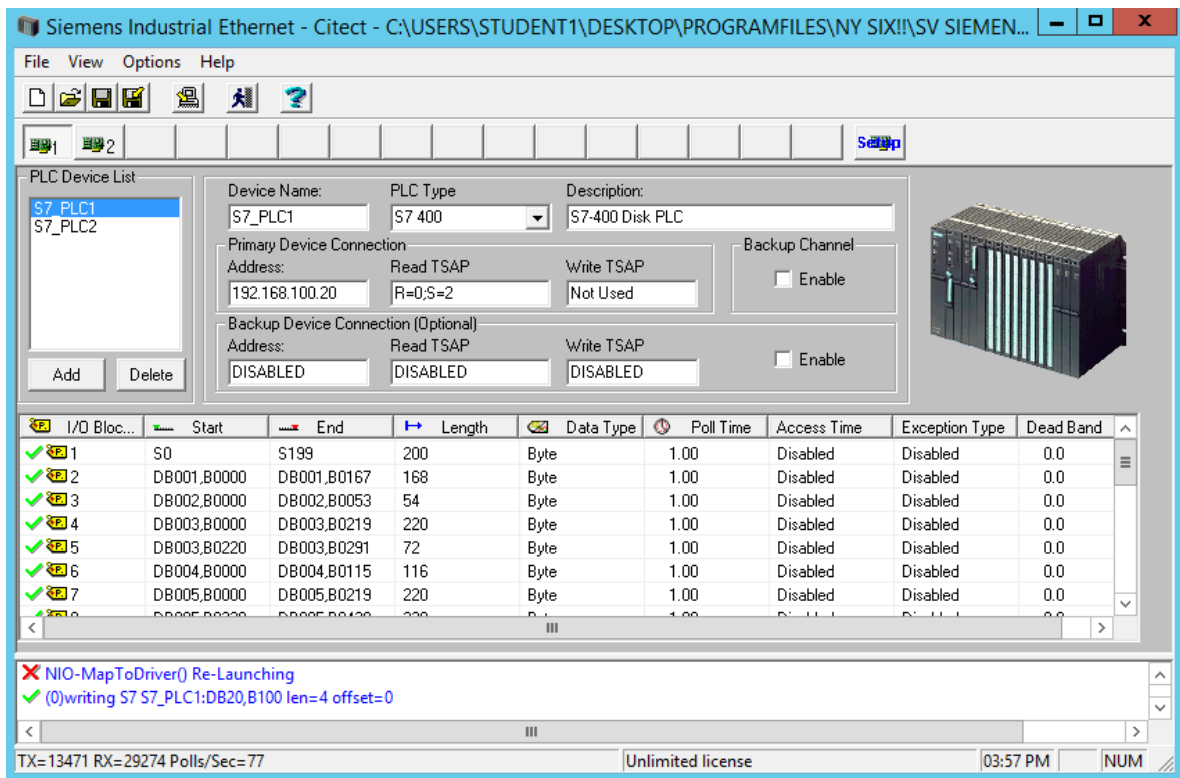
Figur 5.6 – Samplingsvärde



Figur 5.7 – Samtliga taggar

### 5.3 Anslutning mellan SCADA och PLC

Oatly använder sig av två Allen-Bradley PLC:er från Rockwell. För att få dessa att kommunicera med SCADA installerades drivrutiner för detta. Med dessa tillkom Siemens Industrial Ethernet Configuration som är ett verktyg för att konfigurera PLC:n mot SCADA. Med detta verktyg tas en backup av konfigurationen som körs på Oatlys program och lägger in den på vår servers motsvarande program. Eftersom I/O Device nu är konfigurerat så kan Memory Mode stängas av och därmed tar både SCADA och MES emot realtidsvärden.

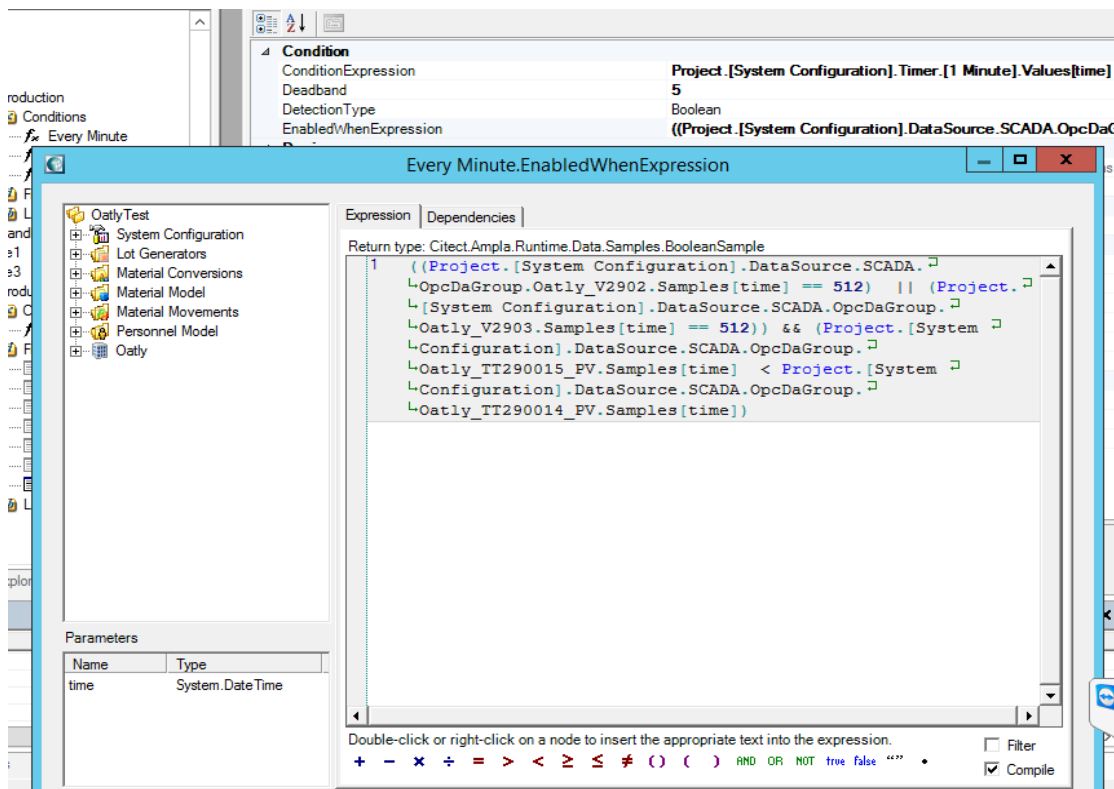


Figur 5.8 – Siemens Industrial Ethernet Configuration

## 5.4 Ampla Express

Disklinje 1 och 2 är delprocesser som fungerar på samma vis men behandlar olika objekt i fabriken. För att implementera dessa skapas två underkataloger DL1 och DL2 i Ampla Studio. Under varje katalog skapar vi "Capture Conditions" (villkor) som bestämmer när och hur ofta mätvärden skall hämtas från SCADA-projektet.

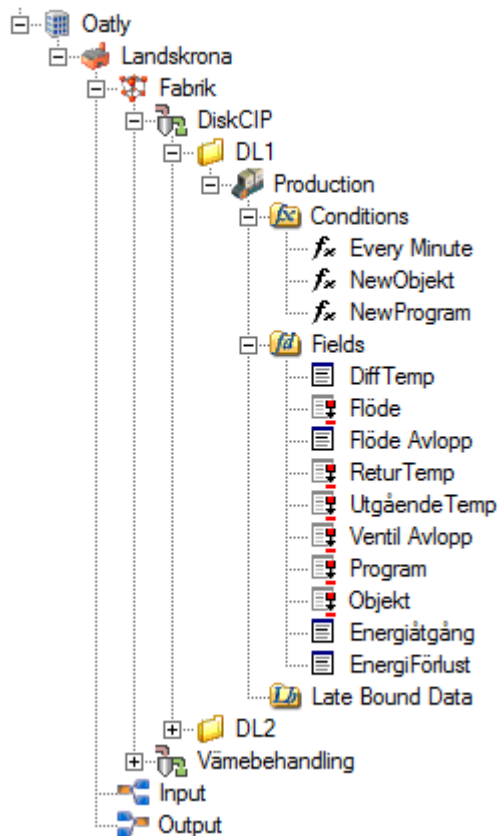
Villkoret för **när** sampling sker skrivs som ett logiskt uttryck där en viss tagg ska ha ett särskilt värde, exempelvis att en ventil (som läser på bit 512 och 1024) ska ha värdet 512, för att Ampla ska sampla från just den taggen. Nedan visas ett exempel från en värmebehandling som är en delprocess i Oatlys fabrik. Samplingen sker om någon av två ventiler har värdet 0 och att ingående temperatur till processen är mindre än utgående temperatur.



Figur 5.9 – EnableWhenExpression

Villkoret för **hur ofta** sampling sker, pekar på en Timer-funktion som skapas under en egen katalog.

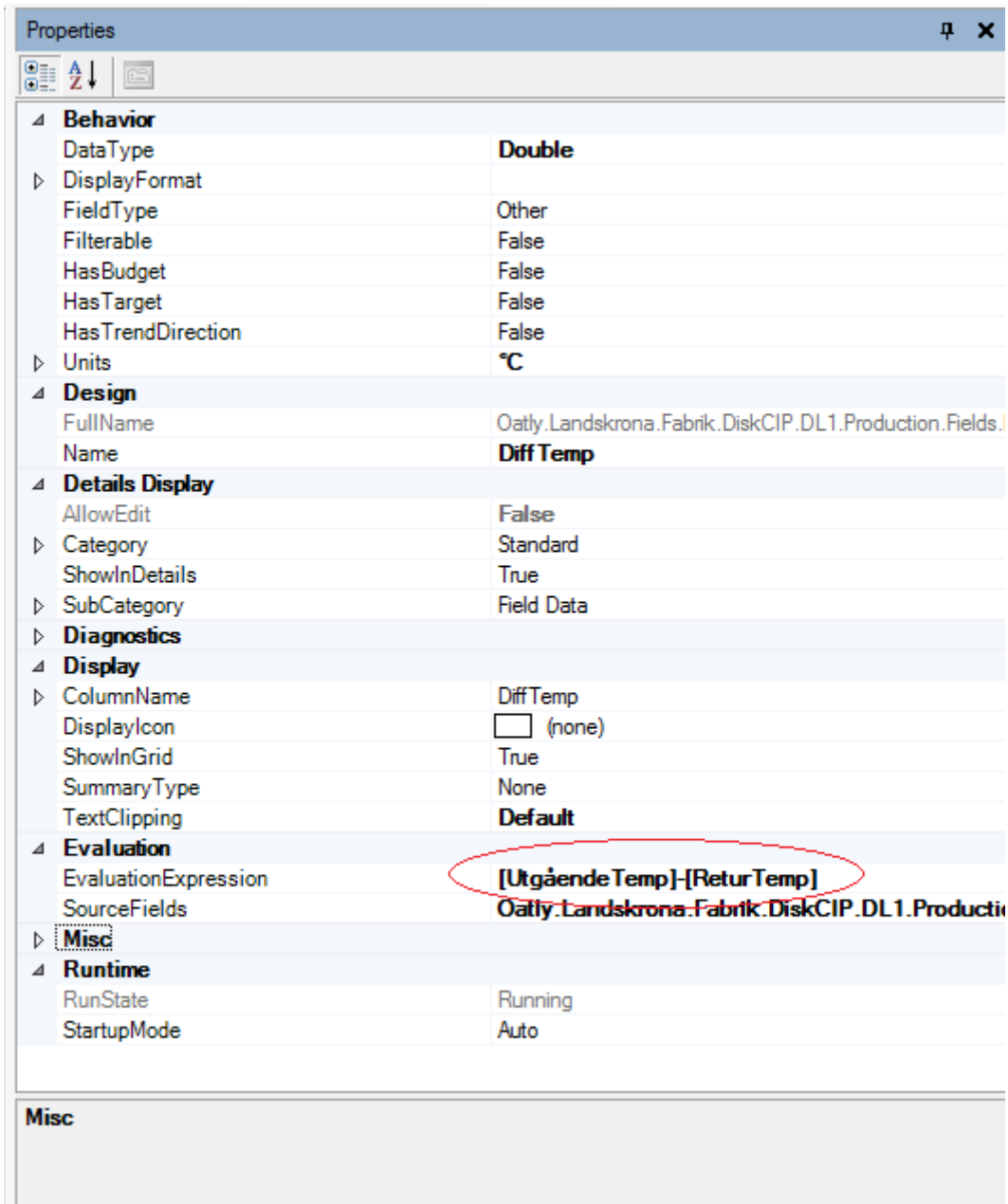
För varje mätvärde som skall plockas, skapas ett field som fungerar som en variabel där värdet sparas tills det skrivs över vid nästa sampling.



Figur 5.10 – fields i Ampla Studio

Till varje field bestäms egenskaper så att värdet behandlas korrekt. Detta kan röra sig om vilken datatyp (Integer,String,Boolean etc.) som taggen är beskriven enligt i SCADA-projektet eller vilken enhet som ska skrivas ut. Till exempel kan en ventil som öppnas och stängs beskrivas som en Boolean, medans en flödesmätare passar bättre som Double.

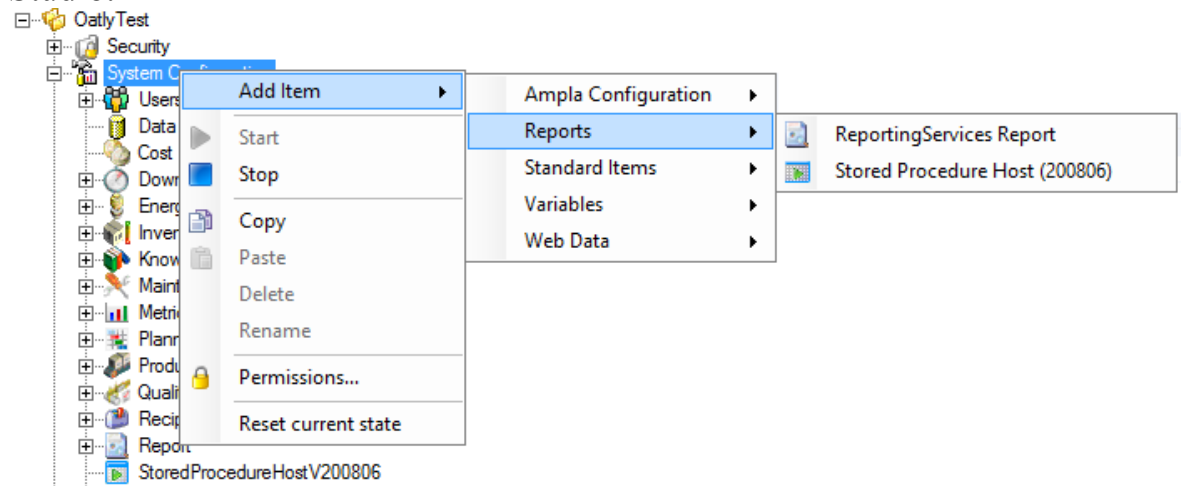
Ofta är det bra att publicera fields så som de är, men för en ordentlig energianalys krävs formler och beräkningar. För detta ändamål skapas "Evaluation Fields" där värden från flera fields kombineras efter önskad formel. Ett enkelt exempel är att räkna ut en differensstemperatur av en ingående och en utgående temperatur. Bild på detta visas nedan.



Figur 5.11 – Evaluation Field i Ampla Studio

## 5.5 Rapporter

Om man önskar att skapa egna rapporter med data ifrån Ampla Production Analyst får man först generera en Stored Procedure Host inifrån Ampla Studio.



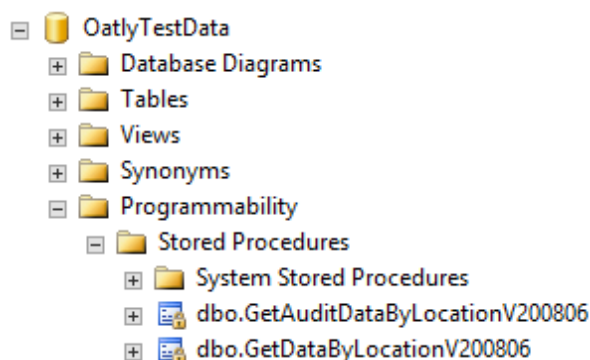
*Figur 5.12 – Stored Procedure Host*

Med StoredProcedureHostV200806 som nu skapades genererar man installationsfiler som lagras på C:\

Citect.Ampla.Public.MSSQL.Data.V200806.dll	4/10/2014 12:00 AM	Application extension
Citect.Ampla.Public.MSSQL.Data.V200806.XmlSerializers.dll	4/10/2014 12:00 AM	Application extension
InstallScriptV200806	5/7/2014 1:36 PM	Microsoft SQL Server Query File

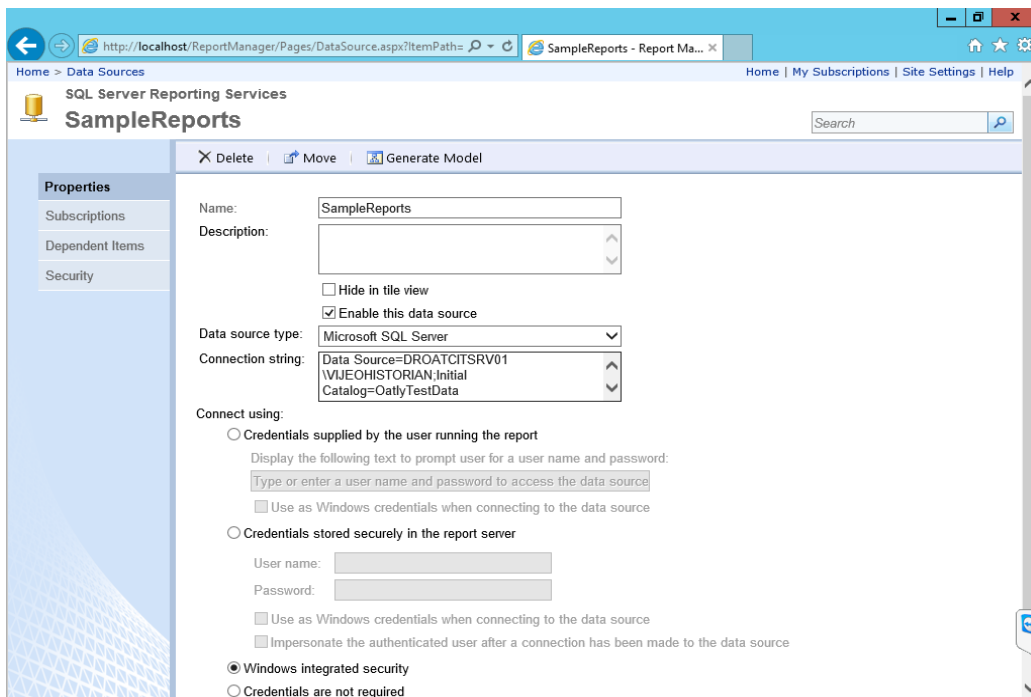
*Figur 5.13 – Genererade filer från Stored Procedure Host*

Med InstallScriptV200806 exekverar man i Sql Server Management Studio en sql-query (se *Appendix B*). I ALTER DATABASE anger man databasen som hanterar data ifrån ens Ampla projekt. I vårt fall OatlyTestData. Scriptet skapar dbo.GetAuditDataByLocationV200806 och dbo.getDataByLocationV200806 inuti databasen.



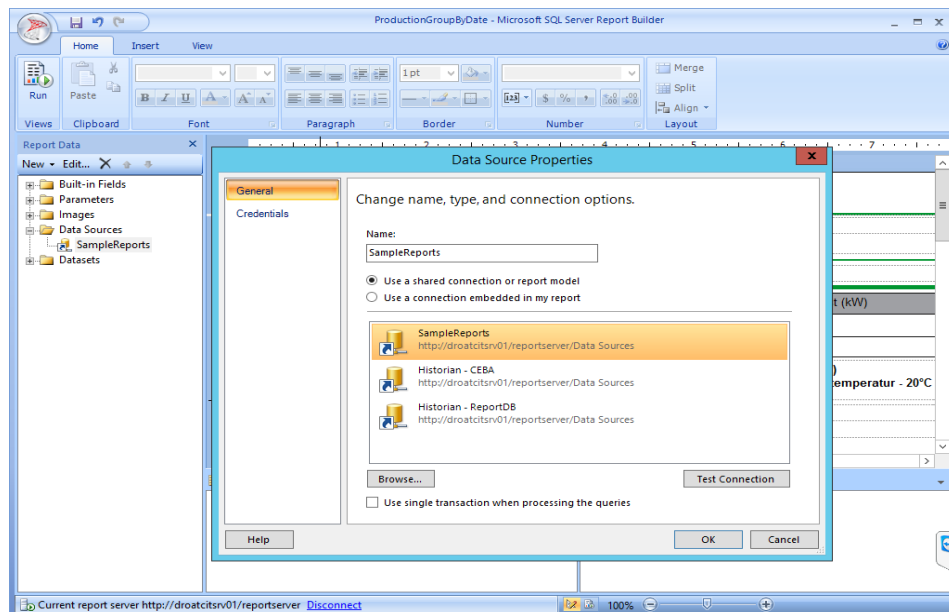
*Figur 5.14 – Sql Server Manager*

För att komma åt dessa metoder inifrån Report builder måste man skapa en data source där man anger vilken server och vilken katalog den skall hämta data ifrån. Detta anges i ”Connection string”.



*Figur 5.15 – Data Source*

Nu kan man börja använda verktyget för att skapa rapporterna. Microsoft Sql Server Report Builder i vårt fall. Först måste den datasourcen anges, som tidigare skapades.



Figur 5.16 – Ange datasource

Nu vill man hämta fälten som finns angivna i Ampla Production Analyst under respektive modul och location. Till detta används två versioner av sql-queryn `dbo.getDataByLocationV200806`. Denna queryn har inparametrar för hämta den önskade tabellen. I Appendix C hämtas alla fält från Ampla Production Analyst som finns på disklinje 1. Detta ses genom att parametern `location` är angivet som `N'Oatly.Landskrona.Fabrik.DiskCIP.DL1'`. I Appendix D anger vi tomma inparametrar. Detta eftersom när rapporten hämtas inifrån Ampla Production Analyst vill man inte ange om det är disklinje 1 eller disklinje 2 som rapporten skall hämta sina värden ifrån utan rapporten skall få sina inparametrars värde ifrån Ampla Production Analyst (se *Figur 5.20*).



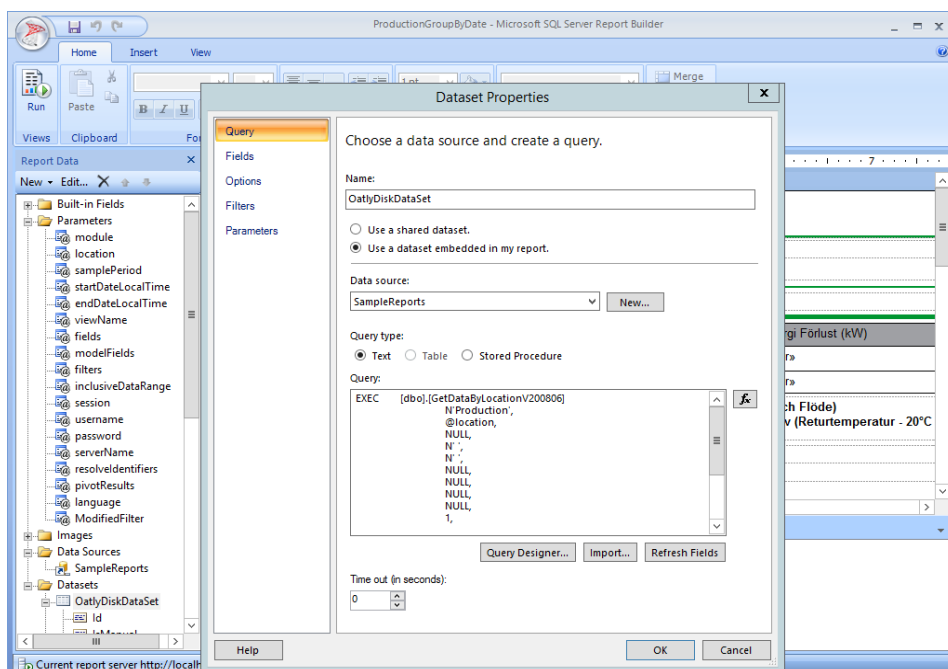
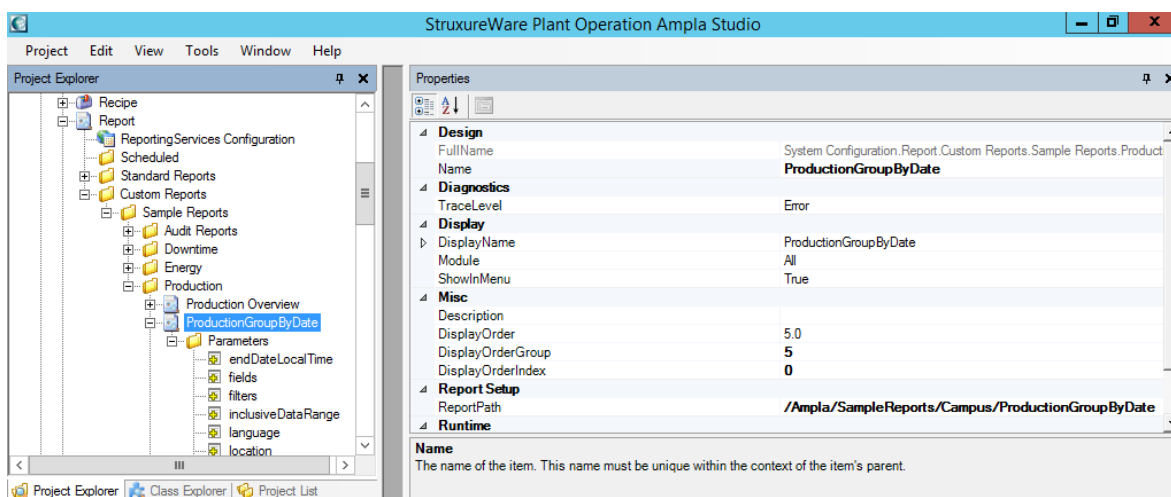


Fig 5.17 – Queryn, fält och parametrar

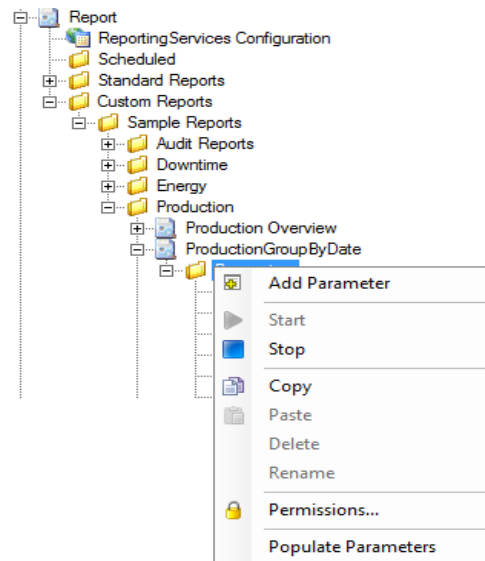
Vid själva skapandet av rapporterna, i vårt fall med tabeller och diagram, anger man vilka fält man vill visa, hur de skall grupperas och vilka värden som skall filtreras bort. De skapade rapporterna sparas på en ReportServer som ligger på adressen <http://localhost/Reportserver>.

I Ampla Studio skapas vi en ny rapport där ”ReportPath” blir rapportens plats på servern.



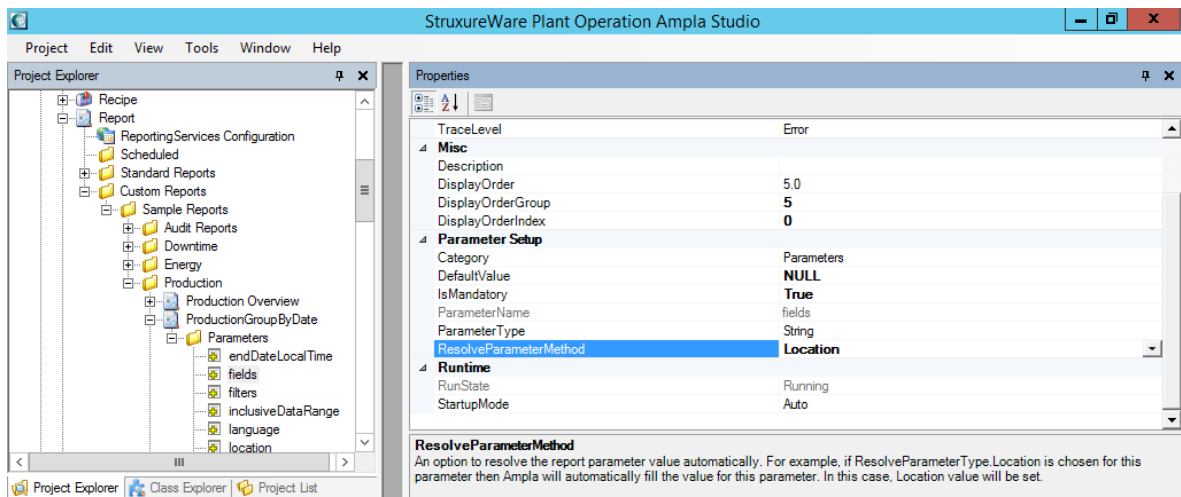
Figur 5.18 – Report Path

Nu kan vi hämta parametrarna (Populate Parameters) som finns angivna i rapporten.



Figur 5.19 – Populate Parameters

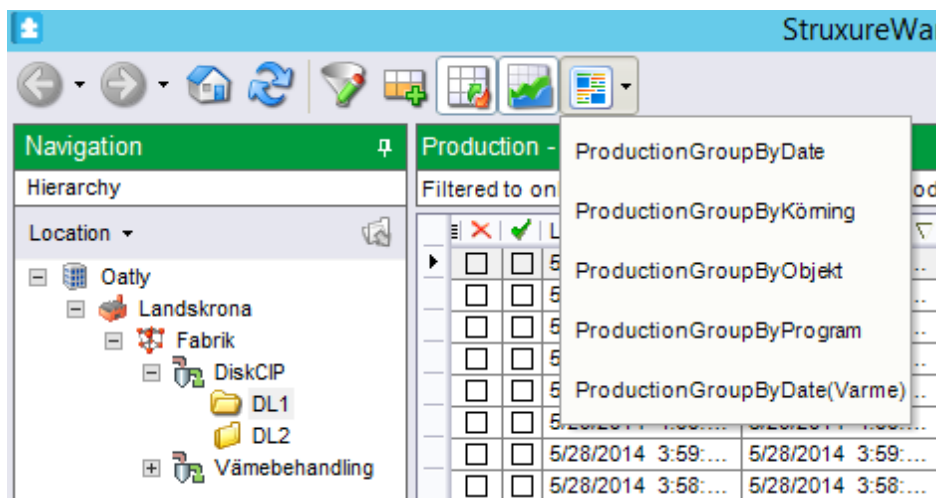
Därefter anges hur parametrarna skall hämta sina värden. I detta fallet skall parametrarna hämta sina värden beroende på vilken process man befinner sig i, samt vilket tidsintervall man angivit i filtreringen.



Figur 5.20 – ResolverParameterMethod

Till detta projekt skapades fyra rapporter för visualisering av data från disklinje 1 & 2 och en rapport för visualisering av data från

värmebehandlingen. Rapporterna till disklinjerna skiljer sig åt då de grupperar ihop data på olika sätt. Se exempel på datumgrupperade rapporter i Appendix E.



Figur 5.21 – Rapporter i Ampla Production Analyst

## 6 Diskussion och Slutsatser

Målet var att med hjälp av ett implementerat MES, kunna lagra väsentliga mätvärden och sedan konstruera översiktliga rapporter som beskriver energieffektiviteten i en process. Vi anser att den största anledningen till att vi lyckades med detta mål var att vi lade ner mycket tid på grundliga förstudier och simuleringar samt att vi ofta drog egna slutsatser efter de observationer vi gjorde under inlärningsperioden. Vi hade alltid för vana att prova olika scenarier när vi lärde oss en ny funktion i samtliga applikationer, för att finna eventuella komplikationer och brister. Om vi exempelvis lyckats göra en energiberäkning utav ett flertal variabler med tilldelade värden, så testade vi även vad som hände om en av variablerna hade värdet "null", alltså en tom variabel. Observation av just ett sådant scenario visade då att om bara en variabel har värdet null, så blir den totala uträkningen också null. Detta är inget fel utan snarare ett hinder som man får gå runt och genom att lägga in ett logiskt villkor som säger att värdet null ersätts med "0", tillåts kan man skriva ut korrekta beräkningar utan att ta större hänsyn till tomma variabler.

Lösningen till hur samtliga programvaror skulle implementeras blev att följa diverse manualer. Dock uppkom mycket problem när detta skulle göras eftersom de operativsystemen vi arbetade med inte var rekommenderade. Problem som kunde uppkomma vara att t.ex. de portar som Ampla behövde för kommunikation med SQL-servern redan användes av ett annat program. Problem av denna typen fick Citects support ägna ett antal timmar på att lösa med hjälp av fjärrstyrning. En slutsats är därför att endast använda rekommenderade operativsystem.

Var och när MES-verktyget skall hämta sin data, bestämdes helt enkelt av de ansvariga på Oatly eftersom de ansåg att disksystemet och värmebehandlingarna var väsentliga för uppgiften. Ett problem som uppkom här var att energiåtgång och energiförlust ibland fick negativa värden p.g.a. att utgående temperatur var lägre än retur temperaturen. Detta löstes genom att endast hämta värden när utgående är större än ingående. Detta gör att dem värdena vi får ut inte är exakta men de ger oss en uppfattning av åtgången och förlusterna i processen. För att få det mer exakt behövs mer mätpunkter.

Rapporterna var inte det enklaste att få till då den information som fanns tillgänglig oftast talade om hur man hämtade ut värden direkt ur databaser. Dock löstes detta med att pröva sig fram. När vi till slut fick rätt på hur en dataset skulle hämtas löste sig allt annat rätt snabbt. När rapporterna hämtades ifrån Ampla visade de ibland stora negativa värden vilket åtgärdades med hjälp av ett filter som endast visar de värden som är större eller lika med noll.

Vi tycker att det har varit viktigt för vårt arbete att vi haft möjlighet att sitta och arbeta på ett kontor hos Schneider Electric i Helsingborg, där vi har kunnat få bra rådgivning och ledning av vår handledare. Ett kontor är för oss en bättre arbetsplats än att jobba hemifrån där alla möjliga distraktioner kan uppstå.

Det som varit mindre bra med arbetet är de stunder då vi haft bristfällig utrustning och programvara, vilka skapade en hel del väntetider på grund av långa ominstallationer, uppdateringar och kompatibilitetsproblem. Vi har även varit sysslösa då och då när vi varit i behov av svar från ansvariga på Oatly och fått vänta länge innan vi kunnat gå vidare. Det är lätt hänt att motivationen för arbete sjunker och att frustration uppstår när man vill avancera men inte har möjlighet. Vi har dock förståelse för att oförutsedda problem kan uppstå och att kommunikation mellan företag kan svikta, då hela världen trots allt inte kretsar kring vårt examensarbete.

## 7 Utvecklingsmöjligheter

Vi vet att vi hade kunnat utveckla resultatet av vårt arbete rejält under andra omständigheter. Situationen var som sådan att Oatly ville att analysen skulle fokuseras på disksystemet och värmebehandlingarna eftersom dessa delar av processen uppenbarligen förbrukade stora mängder energi. Det var just av denna anledningen som avsaknaden på givare till resterande delprocesser var stor eftersom det inte fanns ett redan uppenbart behov av det från Oatlys sida. Med slutprodukten av vårt arbete kan nu Oatly se möjligheterna i energianalysen och därför överväga en investering i fler givare till fabriken. Med hjälp av mätvärden som representerar körningen av mindre delmängder i fabriken (så som motorer, pumpar etc.) skulle vi kunna utnyttja fler av de moduler i Ampla Express som vi lärde oss i utbildningsmaterialet. Att få logga så kallad ”Downtime” på exempelvis motorer skulle kunna avslöja onödigt långa drifttider mellan batchar. Det kan t. ex vara rätt att en motor står på med låga varv medans den väntar på nästa batch, för att inte behöva startas upp från noll varv/min, varje gång. Men låt säga att disksystemet körs i flera timmar innan nästa batch produceras, då hade man sparat mer energi genom att helt stänga av motorn i fråga.

Bäst hade varit om vi kunde utnyttja samtliga moduler i Ampla Express, så pass att vi kunde räkna ut ett  $OEE+E$ -värde för hela Oatlys fabrik. Då skulle man ha kontroll på den totala effektiviteten istället för att kontrollera varje enskild delprocess.

Med uträknade  $OEE+E$ -värden hade vi kunnat utveckla en så kallad Dashboard som låter arbetare ha konstant uppfattning om energieffektiviteten i fabriken. Detta visas då som ett diagram på en html-sida. Se nedanstående bild.



Figur 6.1 – Exempelbild på Dashboard, utbildningsmaterialet

## 8 Referenser

- [1] <https://schneider-electric.app.box.com/s/64fcrlyjh0mzvpik4e2m>  
(Maj 2014) Ampla SSRS Reports

Följande källor kräver ett konto på sidan. Det är gratis att skapa ett konto eftersom endast demo-applikationer är tillgängliga d.v.s. ingen licens krävs.

- [2] <http://www.citect.schneider-electric.com/mes/ampla/downloads-updates/product-downloads>  
(Maj 2014) Ampla 6.0 - Maintenance Release 2 (6.0.7010)

- [3] <http://www.citect.schneider-electric.com/SCADA/vijeo-citect/downloads-updates/product-downloads>  
(Maj 2014) Vijeo Citect V7.40 ISO

- [4] <http://www.citect.schneider-electric.com/mes/ampla/find-answers/knowledge-base>  
(Maj 2014) Citect Knowledgebase

## 9 Appendix

### 9.1 Appendix A

Ampla\_Simulering.ci

```
FUNCTION Startup()
```

```
    TaskNew("Simulering_Disk1_Dump","",8);  
    TaskNew("Simulering_StartDisk1","",8);  
    TaskNew("Simulering_StartProgramDisk1","",8);
```

```
END
```

```
!-----  
!Funktion som öppnar ventil till dumpvattnet för Disklinje 1  
!-----
```

```
FUNCTION Simulering_Disk1_Dump()
```

```
    WHILE 1 DO
```

```
        IF V9585 = 0 THEN
```

```
            TT957016_PV = TestSinWave(100,  
70,100);
```

```
            FT957013_PV = FS957019;  
            TT957021_PV = TestSinWave(80, 40,70);
```

```
                SleepMS(1000);
```

```
        ELSE
```

```
            FS957019 = 0;  
            TT957021_PV = 0;
```

```
        END
```

```
            SleepMS(1000);
```

```
    END
```

```
END
```



```

!-----
!Funktion som simulerar Disklinje1
!-----
FUNCTION Simulering_StartDisk1()

    WHILE 1 DO

        IF DL1_F = 1 THEN

            V9585 = 0;
            Simulering_Disk1_Dump();

        ELSE

            V9585 = 1;
            FT957013_PV = 0;
            TT957016_PV = 0;
            END

        END
        SleepMS(500);
    END

```

```

!-----
!Funktion som simulerar flöde och ändring av program i Disklinje 1
!-----
FUNCTION Simulering_StartProgramDisk1()

    DL1_Program_Nr = 0;

    WHILE 1 DO

        DL1_Program_Nr = DL1_Program_Nr + 1;
        FT957013_PV = TestRandomWave(60,3000,5000);
        FS957039 = TestRandomWave(60,3000,5000);
        Sleep(180);

    END
END

```

Appendix A visar Cicode som simulerar styrning av ett disksystem. Det som sker i koden är att man helt enkelt stänger av och på ventiler (variabler) och i samband med detta tilldelar värden till olika variabler, vilket simulerar att exempelvis en temperatursgivare mäter temperatur när ett vattenflöde tillkommer.

Vi använder ”While-satser” för att det skall ske ändringar kontinuerligt. I slutet på varje While-sats används ett ”Sleep-kommando” som gör att man får tid på sig att avläsa värdet innan det ändras. Siffran i kommandot anger hur lång fördröjningen är. Det finns korta beskrivningar av metoderna där raderna börjar med ett utropstecken.

Appendix B visar innehållet i InstallScriptV200806 som exekveras mot angiven databas för att skapa två StoredProcedure som används för att hämta ut den samplade informationen.

## 9.2 Appendix B

```
ALTER DATABASE [ReportDB] SET TRUSTWORTHY ON
EXEC sp_configure 'clr enabled', 1;
RECONFIGURE WITH OVERRIDE;
```

```
DECLARE @InstallPath nvarchar(1024);
Select @InstallPath = 'C:\Temp\CLR\';
```

```
if object_id('dbo.GetDataByLocationV200806', N'PC') is not null
    drop procedure dbo.GetDataByLocationV200806
if object_id('dbo.GetAuditDataByLocationV200806', N'PC') is not null
    drop procedure dbo.GetAuditDataByLocationV200806
if exists (select * from sys.assemblies where name =
N'Citect.Ampla.Public.MSSQL.Data.V200806.XmlSerializers')
    drop assembly
[Citect.Ampla.Public.MSSQL.Data.V200806.XmlSerializers]
if exists (select * from sys.assemblies where name =
N'Citect.Ampla.Public.MSSQL.Data.V200806')
    drop assembly [Citect.Ampla.Public.MSSQL.Data.V200806]
```

```
CREATE ASSEMBLY [Citect.Ampla.Public.MSSQL.Data.V200806] FROM
@InstallPath + 'Citect.Ampla.Public.MSSQL.Data.V200806.dll' WITH
PERMISSION_SET = UNSAFE;
CREATE ASSEMBLY
[Citect.Ampla.Public.MSSQL.Data.V200806.XmlSerializers] FROM
@InstallPath +
```

```
'Citect.Ampla.Public.MSSQL.Data.V200806.XmlSerializers.dll' WITH  
PERMISSION_SET = UNSAFE;  
GO
```

```
CREATE PROCEDURE [dbo].[GetDataByLocationV200806]  
    @module [nvarchar](128),  
    @location [nvarchar](4000),  
    @samplePeriod [nvarchar](4000) = null,  
    @startDateLocalTime [datetime] = N'',  
    @endDateLocalTime [datetime] = N'',  
    @viewName [nvarchar](4000) = null,  
    @fields [nvarchar](4000) = null,  
    @modelFields [nvarchar](4000) = null,  
    @filters [nvarchar](4000) = null,  
    @inclusiveDataRange [int] = 1,  
    @session[nvarchar](128) = null,  
    @username[nvarchar](512) = null,  
    @password[nvarchar](512) = null,  
    @serverName[nvarchar](4000) = N'http://localhost:8889/',  
    @resolveIdentifiers[bit] = 0,  
    @pivotResults[bit] = 1,  
    @language [nvarchar](128) = null  
WITH EXECUTE AS CALLER  
AS  
EXTERNAL NAME [Citect.Ampla.Public.MSSQL.Data.V200806].  
[Citect.Ampla.Public.MSSQL.Data.V200806.DataStoredProcedure].  
[GetDataByLocationV200806]  
GO  
EXEC sys.sp_addextendedproperty @name=N'AutoDeployed',  
@value=N'yes' , @level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'PROCEDURE',@level1name=N'GetDataByLocationV20080  
6'  
GO  
EXEC sys.sp_addextendedproperty @name=N'SqlAssemblyFile',  
@value=N'DataStoredProcedure.cs' ,  
@level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'PROCEDURE',@level1name=N'GetDataByLocationV20080  
6'  
GO  
EXEC sys.sp_addextendedproperty @name=N'SqlAssemblyFileLine',  
@value=27 , @level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'PROCEDURE',@level1name=N'GetDataByLocationV20080  
6'
```

GO

```
CREATE PROCEDURE [dbo].[GetAuditDataByLocationV200806]
```

```
    @module [nvarchar](128),  
    @location [nvarchar](4000),  
    @setId [nvarchar](512) = N'',  
    @editBy [nvarchar](4000) = N'',  
    @samplePeriod [nvarchar](4000) = null,  
    @startDateLocalTime [datetime] = N'',  
    @endDateLocalTime [datetime] = N'',  
    @session [nvarchar](128) = null,  
    @username [nvarchar](512) = null,  
    @password [nvarchar](512) = null,  
    @serverName [nvarchar](4000) = N'http://localhost:8889/',  
    @recordType [nvarchar](128) = N''
```

```
WITH EXECUTE AS CALLER
```

```
AS
```

```
EXTERNAL NAME [Citect.Ampla.Public.MSSQL.Data.V200806].
```

```
[Citect.Ampla.Public.MSSQL.Data.V200806.AuditDataStoredProcedure].
```

```
[GetAuditDataByLocationV200806]
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'AutoDeployed',  
    @value=N'yes' , @level0type=N'SCHEMA',@level0name=N'dbo',  
    @level1type=N'PROCEDURE',@level1name=N'GetAuditDataByLocationV2  
00806'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'SqlAssemblyFile',  
    @value=N'AuditDataStoredProcedure.cs' ,  
    @level0type=N'SCHEMA',@level0name=N'dbo',  
    @level1type=N'PROCEDURE',@level1name=N'GetAuditDataByLocationV2  
00806'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'SqlAssemblyFileLine',  
    @value=27 , @level0type=N'SCHEMA',@level0name=N'dbo',  
    @level1type=N'PROCEDURE',@level1name=N'GetAuditDataByLocationV2  
00806'
```

GO

### 9.3 Appendix C

```
EXEC      [dbo].[GetDataByLocationV200806]
          @module = N'Production',
          @location =
N'Oatly.Landskrona.Fabrik.DiskCIP.DL1',
          @samplePeriod = NULL,
          @startDateLocalTime = N' ',
          @endDateLocalTime = N' ',
          @viewName = NULL,
          @fields = NULL,
          @modelFields = NULL,
          @filters = NULL,
          @inclusiveDataRange = 1,
          @session = NULL,
          @username = N'*****',
          @password = N'*****',
          @serverName = N'http://localhost:8889/',
          @resolveIdentifiers = 1,
          @pivotResults = 1,
          @language = N'en-US
```

### 9.4 Appendix D

```
EXEC      [dbo].[GetDataByLocationV200806]
          N'Production',
          @location,
          NULL,
          N' ',
          N' ',
          NULL,
          NULL,
          NULL,
          NULL,
          1,
          NULL,
          N'Simple',
          N'simple',
          N'http://droatcitsrv01:8889/',
          1,
          1,
          N'en-US'
```

## 9.5 Appendix E

### DiskCIP: Overview

Ampla

Location "Oatly.Landskrona.Fabrik.DiskCIP.DL1.Production"  
Sample Period Last Week : 5/18/2014 12:00:00 AM - 5/25/2014 12:00:00 AM

Group by: Date

Sample Period	*Energiåtgång (kW)	*Energi Förlust (kW)
05/19/2014	47.82	390.58
05/20/2014	127.96	126.12
05/21/2014	1061.62	6725.08
05/22/2014	476	5245.59
05/23/2014	196.87	1459.26
Total	1910.27	13946.63

\*Energiåtgång - Den energin som användes (Beräknas med hjälp av Differens temperatur och Flöde)

\*Energi Förlust - Den energin som ej användes, utan gick till avlopp (Beräknas med hjälp av (Returtemperatur - 20°C \* Flöde)

Report Generated : 31/05/2014 02:08

Page 1 of 3

Schneider  
Electric

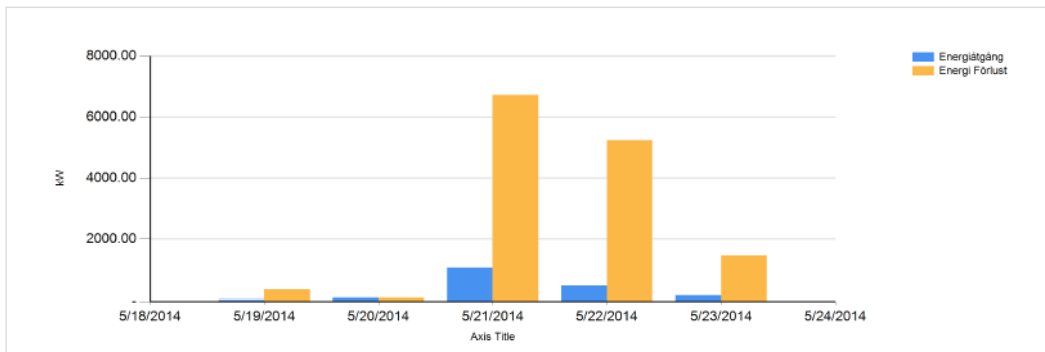
Rapporten visar jämförelse mellan energiåtgång och energiförlust i i disksystemet. Energiåtgång är den energi som det går åt för att köra en viss process och energiförlust är den mängd energi som man inte lyckats ta tillvara på. Ett exempel på förlorad energi kan vara värmeenergi i hett vatten som rinner ut i ett avlopp. Rapporten är grupperad efter datum.

## DiskCIP: Overview

Ampla

Location "Oatly Landskrona Fabrik DiskCIP DL1 Production"  
Sample Period Last Week : 5/18/2014 12:00:00 AM - 5/25/2014 12:00:00 AM

Group by: Date



Report Generated : 31/05/2014 02:08

Page 2 of 3

Schneider  
Electric

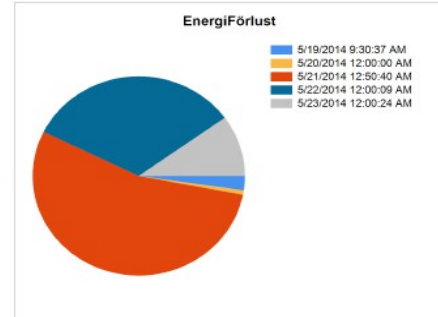
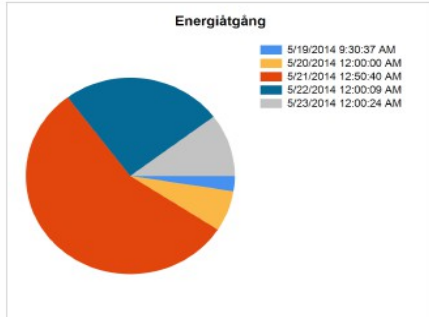
Samma som föregående rapport, men i form av stapeldiagram.

## DiskCIP: Overview

Ampla

Location "Oatly Landskrona Fabrik, DiskCIP, DL1, Production"  
Sample Period Last Week : 5/18/2014 12:00:00 AM - 5/25/2014 12:00:00 AM

Group by: Date



Report Generated : 31/05/2014 02:08

Page 3 of 3

Schneider  
Electric

Samma som de två föregående rapporterna, men i form av cirkeldiagram.