

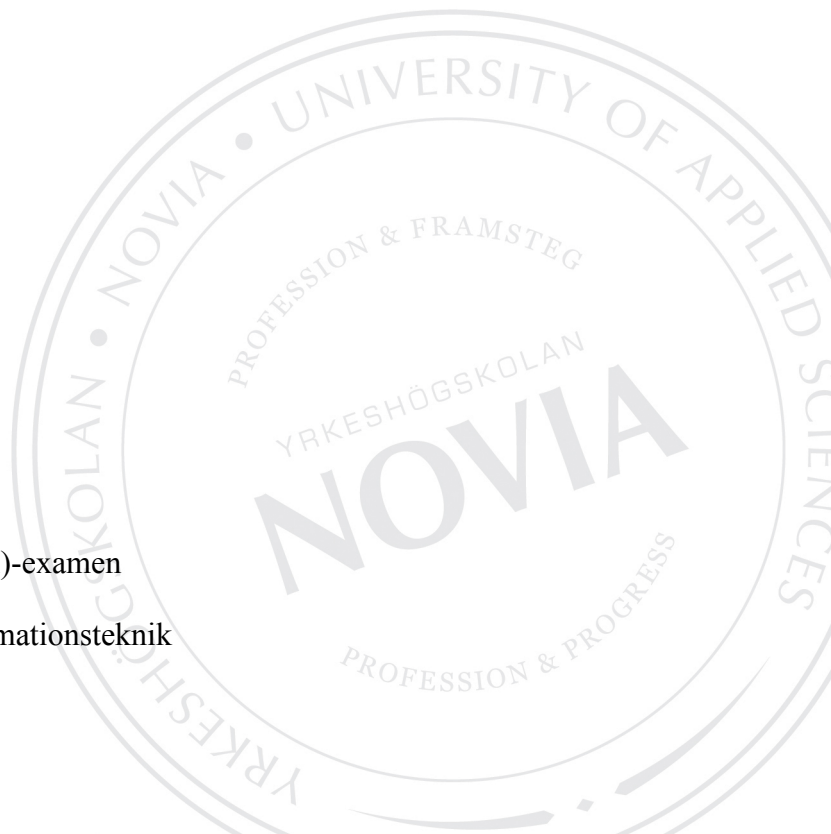
# **SAP-applikation för effektivisering av lastningsplanering**

Lars Holm

Examensarbete för Ingenjör (YH)-examen

Utbildningsprogrammet för Informationsteknik

Vasa 2014



## **EXAMENSARBETE**

Författare:

Lars Holm

Utbildningsprogram och ort:

Informationsteknik, Vasa

Handledare:

Kim Roos

Titel: SAP-applikation för effektivisering av lastningsplanering

---

Datum: 1 april 2014

Sidantal: 31

Bilagor: -

---

### **Abstrakt**

Detta examensarbete utreder ett sätt att skapa applikationer i SAP. Dessa skall användas för att förbättra beställaren Celsa Steel Service Oy:s lastningsplanering genom att skapa system som gör att information om planerade lastningar alltid hålls uppdaterad, och att alla personer som har behov av informationen kan alltid komma åt den. Dessutom skapas det ett system för att granska och uppfölja på företagets transportsäkerhet.

Applikationerna skapas i ERP-programpaketet SAP och skrivs i programmeringsspråket ABAP. Examensarbetet går också igenom några av de grundläggande funktionerna av både SAP och ABAP, samt exempel på dessa.

Resultatet av examensarbetet är en applikation som är skapad i en utvecklingsversion av SAP, som senare kan flyttas till Celsa Steel Service Oy:s egna SAP-installation.

---

Språk: Svenska

Nyckelord: SAP, ABAP, ERP, Affärssystem

---

## **BACHELOR'S THESIS**

Author: Lars Holm  
Degree programme: Information technology, Vasa  
Supervisor: Kim Roos

Title: SAP-application for increased efficiency of transport planning

---

Date: 1 April 2014                      Number of pages: 31                      Appendices: -

---

### **Summary**

This thesis explores a means for creating applications in SAP. These will be used to improve the client Celsa Steel Service Oy's transport planning by creating systems which makes information regarding transports always up-to-date and accessible by everyone who a need for it. In addition to this, a system is created for monitoring and following up on the company's delivery dependability.

The applications are created in the ERP software suite SAP and is written in the programming language ABAP. The thesis also brings up some of the basic features of both SAP and ABAP, as well as gives examples on these.

The result of the thesis is an application created in a development version of SAP, which can later be implemented in Celsa Steel Service Oy's own SAP installation.

---

Language: Swedish                      Key words: SAP, ABAP, ERP

---

# Innehållsförteckning

1 Inledning.....	1
1.1 Om Celsa .....	1
1.2 Nuläge och dess nackdelar .....	2
1.3 Syfte.....	2
1.4 Målgrupp .....	3
2 Om SAP.....	4
2.1 Beskrivning av SAP.....	4
2.2 Struktur .....	5
2.2.1 Databas.....	5
2.2.2 Applikationsserver .....	6
2.2.3 Klientapplikation.....	6
2.3 Programmeringsspråket ABAP.....	8
3 Verktyg och språk .....	9
3.1 SAP utvecklingsverktyg .....	9
3.1.1 Object Navigator.....	10
3.1.2 Screen Painter .....	10
3.1.3 Menu painter .....	11
3.1.4 ABAP Dictionary .....	12
3.1.5 Number Range Object.....	12
3.2 ABAP Programmering.....	14
3.2.1 Hello world .....	16
3.2.2 Databastabell.....	18
3.3.3 Spara data till databastabell .....	20
3.3.4 Hämta data från databastabell.....	20
3.3.5 Skapa transaktionskod .....	21
4 Utförande.....	22
4.1 Databas .....	23
4.2 Programmets huvudsida .....	24
4.3 Program för transportplanering .....	25
4.5 Program för leveranssäkerhet .....	28
4.6 Skriva ut rapporter i Excel-format.....	28
5 Resultat och diskussion .....	29
Källförteckning.....	31



# 1 Inledning

## 1.1 Om Celsa

Celsa Group är en Europeisk koncern med verksamhet i metallindustrin, med huvudkontor i Barcelona, Spanien, där det grundades år 1967. Celsa Group är specialiserat på produktion av armeringsprodukter som används i byggindustrin. Idag har de ca. 7000 anställda, och är en av världens största tillverkare av stål (Celsa 2013a) (World Steel Association 2012). Figur 1 nedan visar var i Europa Celsa har företagsverksamhet.



*Figur 1. Celsa Group i Europa (Celsa 2014).*

Största delen av Celsas produkter säljs till företag i Europa, men Celsa säljer också till företag i Afrika, Mellanöstern, Amerika, och Asien. År 2012 sålde de ca. 7,5 millioner ton armeringsprodukter för ca. 4,4 miljarder Euro (Celsa 2013b).

Celsa Steel Service är en del av Celsa Group som har fabriksverksamhet i Norge, Sverige, Danmark, och Finland, och blev en del av Celsa Group år 2006. Celsa Groups företag i Finland heter Celsa Steel Service Oy, och företaget har rötter som går tillbaks ända till slutet av 1800-talet. Celsa Steel Service Oy är det marknadsledande företaget i Finland som är specialiserat inom armeringsprodukter (Celsa 2013a) (Celsa 2013b).

## 1.2 Nuläge och dess nackdelar

Idag saknar Celsa Steel Service i Åminnefors ett bra system för logistik- och expeditionsavdelningarna för att få en översikt över beställningar som skall levereras, samt för att kunna planera lastningar, koordinera dessa lastningar med produktions-, planerings-, och försäljningsavdelningarna. Därtill saknas ett system för att koordinera lastningar med transportföretaget, och att få en översikt över tidigare levererade beställningar, t.ex. för rapporter över leveranssäkerhet.

En del av planeringen sköts med programmet SAP ERP, en annan del i ett Excel-dokument som skrivs ut med data som hämtas från SAP och läggs till i ett Excel-dokument i ett externt program. Resten sköts i ett annat Excel-dokument som måste skapas och uppdateras manuellt.

Detta är både för invecklat, för tidskrävande, och skapar för stor risk för felaktigheter. Då informationen är delad över flera olika platser varav vissa måste uppdateras manuellt, kan det leda till missförstånd mellan olika avdelningar, Det kan t.ex. hända att alla inte har tillgång till samma version av informationen, det kan vara svårt för vissa användare att veta vilken information som skall hämtas varifrån, och det finns en risk för att informationen är felaktig på grund av mänskliga misstag som kan uppstå i flera olika skeden. Eftersom systemet är beroende av att en person kan samla all information till rapporten, betyder det att om någon annan har behov av uppdaterad information måste den personen kontakta personen som hanterar informationen och be om uppdaterade rapporter.

## 1.3 Syfte

Målet med de nya systemen är att Celsa Steel Service Åminnefors skall ha bättre system för att kunna planera dess leveranser, vilket också kommer att leda till att leveranssäkerheten förbättras. Detta skall lösas genom ett SAP-program där information kan samlas, läsas, och ändras av de olika avdelningarna. Genom programmet kan man skicka information om leveranser och transporter till transportföretaget. Det skall finnas snabb och enkel tillgång till all information, alla skall alltid ha tillgång till den senaste informationen och det skall vara lättare att hantera informationen.

Logistik- och expeditionsavdelningarna skall kunna använda programmet för att planera och koordinera lastningar. För att uppfylla dessa ändamål skall programmet kunna användas till att kombinera olika leveranser med varandra till transporter, specificera transporternas lastningstid och -direktiv och skriva ut information om lastningarna till olika sorters Excel-dokument. Exempel på Excel-dokument som skall kunna skrivas ut är dokument som ger en översikt över alla lastningar och dokument som kan skickas till transportföretaget med både relevanta beställningar och transporter. Också dokument som andra avdelningar inom Celsa kan använda för att få en översikt över deras relevanta lastningar skall kunna skrivas ut.

Logistik- och finansavdelningarna skall kunna skriva ut rapporter över företagets leveranssäkerhet. Dessa skall både kunna ses direkt i SAP och skrivas ut till Excel-dokument. I fall en beställning har levererats för sent skall det gå att spara information om orsakerna till att beställningarna blev försenade.

Alla avdelningar inom Celsa Steel Service som har behov av information om beställningar, leveranser och lastningar skall snabbt och enkelt kunna komma åt den direkt via SAP, utan att behöva vara beroende av externa rapporter som kan innehålla föråldrad information.

## **1.4 Målgrupp**

Applikationen skall kunna användas av många olika sorters användare. Till dessa hör användare i expeditions-, logistik-, försäljnings-, produktions-, och planeringsavdelningarna. Eftersom programmets målgrupp är så varierande kommer också de olika användarnas syfte för att använda programmet att vara varierande.

Programmet måste därför vara snabbt och enkelt för användare som har olika erfarenheter både i datoranvändning överlag men också i SAP, men på samma gång måste programmet kunna användas för att utföra en mängd olika uppgifter.



## 2 Om SAP

### 2.1 Beskrivning av SAP

SAP AG är ett tyskt företag som grundades år 1972 av fem före detta IBM-anställda. De började med en kund och en handfull anställda, men har sedan dess växt till att bli det marknadsledande företaget i Enterprise Application Software. De har idag över 251000 kunder i 188 länder, och över 66000 anställda i över 130 länder. SAP står för Systems, Applications, and Products in Data Processing (SAP 2013a).

Till deras kunder hör bland annat Vodafone, Delta Airlines och BMW (SAP 2013b).

SAPs huvudprodukt är SAP ERP (Enterprise Resource Planning). Den innehåller allting som ett företag behöver för att kunna spara, hantera, och läsa information. Det kan t.ex. användas för företagets finanser, human resources, produktion och försäljning. Figur 2 visar olika moduler i SAP ERP (SAP u.å.).

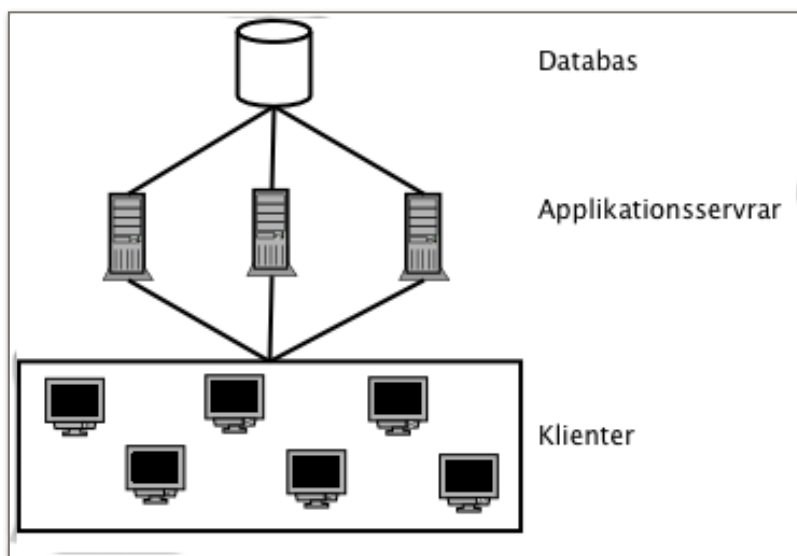
	Self Services							
Analytics	Strategic Enterprise Management		Financial Analytics		Operations Analytics		Workforce Analytics	
Financials	Corporate Governance		Financial Accounting		Management Accounting		Financial Supply Chain Management	
Human Capital Management	Employee Relationship Management		Employee Lifecycle Management		Employee Transaction Management		Workforce Deployment	
Operations: Value Generation	Purchasing	Inventory Management	Manufacturing	Distribution	Sales Order Management	Service Order Management		
Operations: Support	Product Structure Management		Project Management		Quality Management		Asset Management	
Corporate Services	Travel Management		Environment, Health & Safety		Incentive and Commission Management		Real Estate Management	
Solution and Integration Platform	People Integration		Information Integration		Process Integration		Application Platform	

Figur 2. SAP moduler (SAP u.å.b).

SAP som det fungerar idag kom först till användning i mitten av 1990-talet med SAP R/3 och ABAP Objects (Keller & Kruger 2011, s. 24).

## 2.2 Struktur

SAP består av en databas där informationen sparas, en eller flera applikationsserver där programvara körs som kommer åt informationen i databasen och en GUI-klientapplikation som används för att komma åt programmen som körs på applikationsservern, som ses i figur 3 nedan (Schreckenbach 2011, s. 32-33).



Figur 3. SAP struktur med databas, applikationsservrar, och klienter.

### 2.2.1 Databas

Varje SAP-installation har en databasserver. Denna brukar vanligtvis kallas SAP "central instance" (Schreckenbach 2011, s. 32-33). SAP kan köra flera olika typer av databaser, som väljs på basis av hur den kommer att användas. En databastyp kan till exempel vara optimerad för att kunna ha många olika användare som skall kunna komma åt information på samma gång, medan en annan databastyp kan vara optimerad för att användare skall kunna hantera en stor mängd information på samma gång (SAP 2013c).

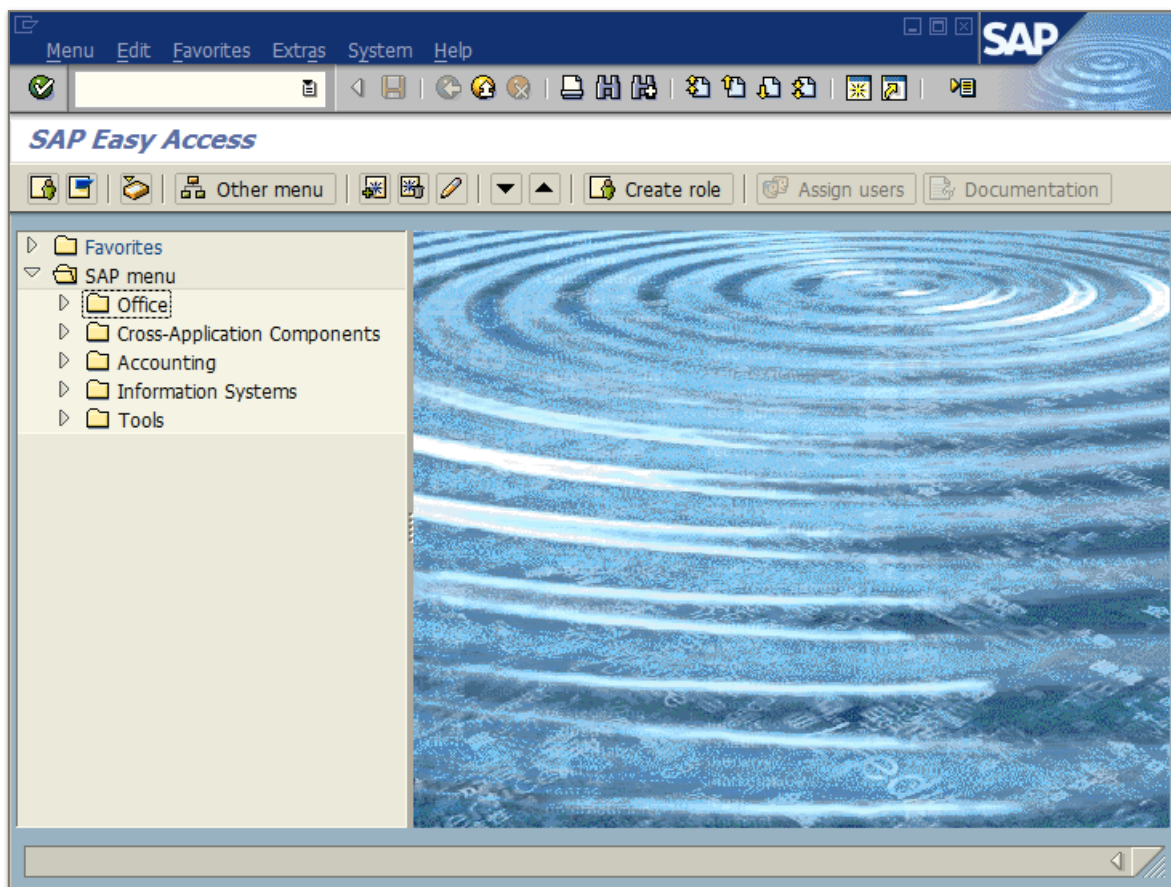
I databasen sparas all programdata som t.ex. beställningar, räkningar, SAP systeminställningar, SAP användare och användarinställningar, samt all källkod för programmen som körs på applikationsserverna (Schreckenbach 2011, s. 36).

### 2.2.2 Applikationsserver

En SAP installation har en eller flera applikationsservrar. Dessa brukar vanligtvis kallas SAP “dialog instances” (Schreckenbach 2011, s. 32-33). Vanligtvis körs databasen och applikationsservern på samma serverdator, men i mer krävande datormiljöer körs databasen på en skild serverdator (Schreckenbach 2011, s. 46-47).

### 2.2.3 Klientapplikation

För att användare skall kunna använda programmen på applikationsservern, används en klientapplikation. Klientapplikationen har ett grafiskt användargränssnitt och körs på användarens dator. Figur 4 visar en bild av ingångssidan till klientapplikationen. Överst i fönstret finns det menyer. Till vänster finns det snabbänkar till SAP-program som körs på applikationsservern. Vilka SAP-program som finns i menyn beror på vilka inställningar SAP-installationen har. I figur 4 är området till höger tomt. Vad som är till höger i fönstret beror också på vilka inställningar SAP-installationen har.



Figur 4. SAP Easy Access klientapplikation, ingångssidan.

Klientapplikationen skickar användarens kommandon till applikationsservern, och applikationsservern skickar information tillbaka till klientapplikationen (Schreckenbach 2011, s. 525).

Alla program i SAP har en programkod. Program som användare skapar skall vanligtvis ha en programkod som börjar med antingen Z eller Y, men kan i specialfall börja med andra bokstäver om SAP AG har gett rättigheter till det. Program som SAP AG har skapat kan börja med vilka bokstäver som helst, men SAP AG följer vanligtvis ett namnsättningssystem som gör att det kan vara lättare att hålla reda på vilket program som hör vart. Till exempel många av de program som har att göra med programutveckling i SAP har en programkod som börjar med SE.

För att köra program i klientapplikationen kan man antingen söka efter dem i listan över alla program eller lägga till dem i listan över användarens favoritprogram, som ses till vänster i figur 4. Ett annat sätt att starta program är att skriva programmets programkod i textfältet som ses uppe till vänster i figur 4. Via textfältet kan man snabbt gå till olika program, förutsatt att man kommer ihåg programkoden.

Genom att använda en webbläsare går det också att använda programmen på applikationsservern utan att installera en klientapplikation. För detta finns det flera olika lösningar som väljs beroende på vilka behov som finns. För att använda standardlösningen, SAP GUI för HTML, behöver man köra SAP ITS (Internet Transaction Server). Man behöver inte installera någonting speciellt på användarens dator utan man kommer åt SAP direkt via en vanlig webbläsare.

SAP GUI för HTML kan vara en bra lösning då man vill komma åt SAP utan att installera klientapplikationen men ändå ha ett användargränssnitt som ser ut som klientapplikationen. Men det har också några nackdelar jämfört med att använda klientapplikationen. Exempel på funktioner som saknas är drag-and-drop, historia, och utskrift. SAP GUI för HTML använder också mer nätverkskapacitet än den vanliga klienten (SAP 2001a) (SAP 2001b) (SAP u.å.c) (SAPDev u.å.).

## 2.3 Programmeringsspråket ABAP

ABAP står idag för “Advanced Business Application Programming”, men ursprungligen stod det för “Allgemeiner Berichts-Aufbereitungs Prozessor”, vilket är tyska för “Allmän Rapportskrivnings Processor” (Keller & Kruger 2011, s. 23).

ABAP är ett programmeringsspråk som skapades och utvecklades av SAP AG, och används endast i SAP-produkter. I början av 1980-talet skapades ABAP för att kunna användas till att skapa rapporter med men i början av 90-talet vidareutvecklades ABAP för att också kunna användas till att skapa hela program. Efter att ABAP blev lämpat för att skapa hela program användes ABAP också för all inbyggd funktionalitet i SAP ERP, och det enda som idag inte är programmerat i ABAP är SAPs system core, som är programmerat i C (Keller & Kruger 2011, s. 23-24).

Även om ABAP nu kan användas till att skapa hela program är det fortfarande specialiserat för datahantering med egenskaper som till exempel inbyggd effektiv tillkomst till databaser och hantering av dess data (Keller & Kruger 2011, s. 25).

I slutet av 90-talet ändrades SAP till att bli ett objektorienterat programmeringsspråk, då ABAP Objects utvecklades för att ABAP programmerare kunde börja dra nytta av olika fördelar som objektorienterad programmering ger. Programutvecklare kunde nu välja om de ville programmera med eller utan ABAP Objects (Keller & Kruger 2011, s. 24, 26).

Möjligheten att programmera utan ABAP Objects betyder att äldre program fortfarande fungerar som de gjorde förut, och programmerare kan fortsätta programmera som de gjorde tidigare utan att lära sig att använda ABAP Objects. Även om det fortfarande är möjligt att programmera utan ABAP Objects så finns det en rad fördelar med den objektorienterade programmeringsmodellen som gör att programutvecklare kan vilja välja att programmera med ABAP Objects, och vissa av de nyare delarna av SAP, som t.ex. Web Dynpro, är gjorda för att fungera bäst med Objects (Keller & Kruger 2011, s. 26-29).

Även om ABAP Objects nu var ett välanpassat programmeringsspråk hade det den nackdelen att det endast användes i SAP, och var därför relativt okänt för personer som inte var SAP-programutvecklare. Därför lade SAP AG i början av 2000-talet till möjligheten att skapa SAP program i Java, då Java redan var ett väl använt programmeringsspråk i företagsvärlden (Keller & Kruger 2011, s. 30-31).

Programutvecklare kan nu välja om de vill utveckla i ABAP eller Java. Båda språken har samma status i SAP, och kan överlag användas för samma saker. När grupper väljer om de kommer att programmera SAP i ABAP eller Java, finns det ändå några olika frågor att ta i beaktande, t.ex. om de behöver göra ändringar i nuvarande program, och programmerarnas nuvarande kunskaper i de olika språken och kostnader för utbildning.

För vissa personer kan ABAP vara svårare att lära sig än Java, delvis p.g.a. att ABAP har många fler nyckelord än Java, 700 mot 50 i Java. ABAP har också den nackdelen att ABAP har både delar som är moderna och delar som är gamla, och blandar dem ihop, vilket betyder att det fortfarande innehåller vissa saker som idag vore onödiga, men som har sparats för att behålla bakåtkompatibilitet i SAP. Eftersom Java används i många andra miljöer än endast i SAP och är mer likt andra moderna programmeringsspråk, kan det också vara lättare att hitta information om det.

ABAP å andra sidan har också en rad fördelar. Eftersom ABAP är optimerat för SAP kan det i många fall ge prestandaförbättringar jämfört med Java. ABAP-utveckling är också djupt integrerat i SAP, och alla verktyg som man behöver för att utveckla, testa, debugga, och köra finns färdigt i SAP (Keller & Kruger 2011, s. 31-40) (SAP 2011).

## **3 Verktyg och språk**

### **3.1 SAP utvecklingsverktyg**

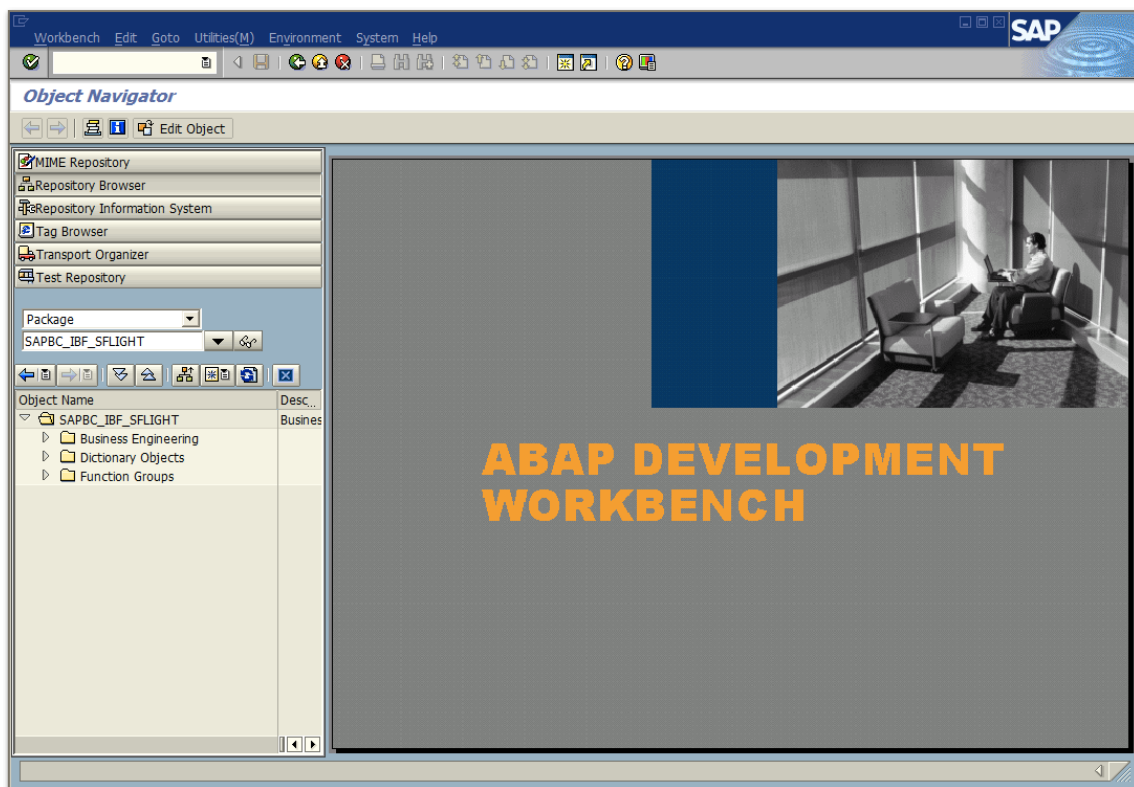
Alla utvecklingsverktyg som man behöver för att skapa SAP-program finns i SAP. SAP-programmet Object Navigator är huvudprogrammet i SAP ABAP-programmering, och via det kommer man åt de flesta funktioner som man behöver. Via Object Navigator kan man skriva, testa, och debugga kod. I Object Navigator finns det också funktioner för att skapa databastabeller, menyer, fönster, och de flesta andra objekt som man kan använda i ett SAP-program.

SAP använder ett system där alla ändringar som görs sparas i ett så kallat "transport request", och Object Navigator har också funktioner för att skapa och spara dessa (Keller & Kruger 2011, s. 26, s. 57-60).

### 3.1.1 Object Navigator

Huvudprogrammet, som fungerar ungefär som en IDE i andra programmeringsspråk, heter Object Navigator. I Object Navigator kan man ändra alla ABAP-filer och deras komponenter, men det finns också andra program som används till att ändra andra sorts filer. Object Navigator har programkoden SE80.

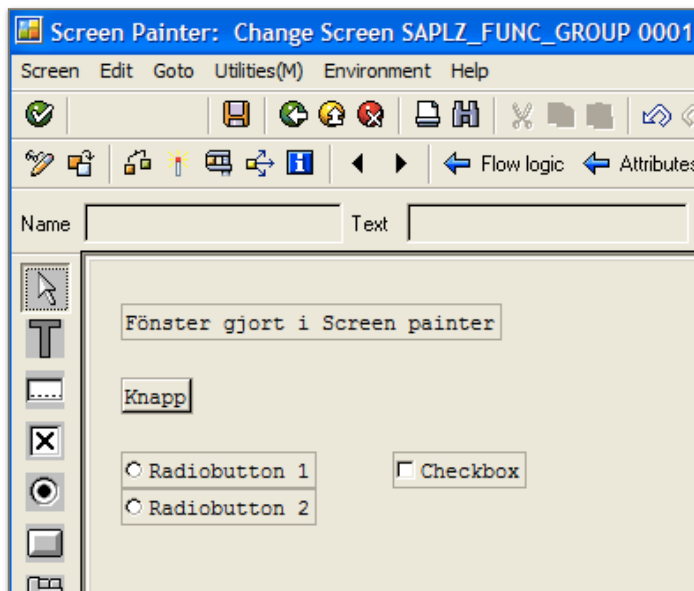
Object Navigators fönster, som ses nedan i figur 5, är delat i två delar, delen till vänster är navigeringsområdet där man väljer vilka projekt och vilka filer man vill redigera, och delen till höger är redigeringsområdet där man t.ex. kan skriva kod och ändra filers egenskaper (Keller & Kruger 2011, s. 59).



Figur 5. ABAP Development Workbench Object Navigator.

### 3.1.2 Screen Painter

Då man vill skapa fönster med grafiska användargränssnitt kan man antingen skapa dem direkt i kod, eller använda programmet Screen Painter. Screen Painter är en WYSIWYG editor där man kan skapa fönster med grafiska användargränssnitt. I figur 6 är Screen Painter med några av de olika komponenterna som man kan använda.

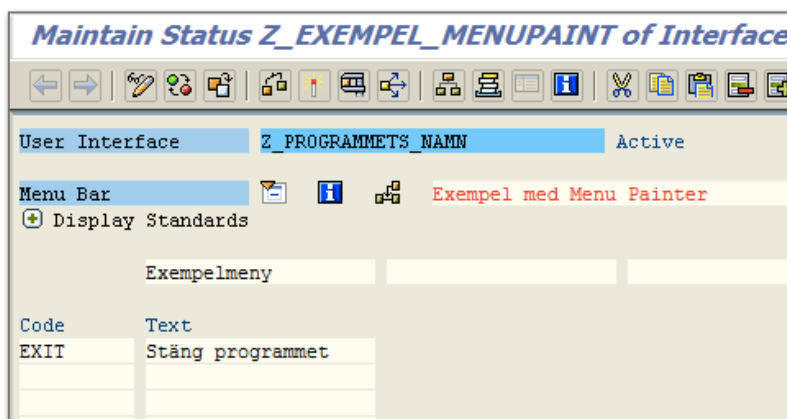


Figur 6. Screen Painter.

Man kan antingen komma åt Screen Painter från Object Navigator, eller med programmets programkod, SE51. Screen Painter skiljer sig från de flesta SAP-program i att det inte körs inne i SAP-klientapplikationen, utan öppnas istället i ett skilt fönster.

### 3.1.3 Menu painter

Menu Painter är programmet i SAP som man kan använda för att skapa menyer. Menu Painter har programkoden SE41. I figur 7 ses Menu Painter och en menyflik.



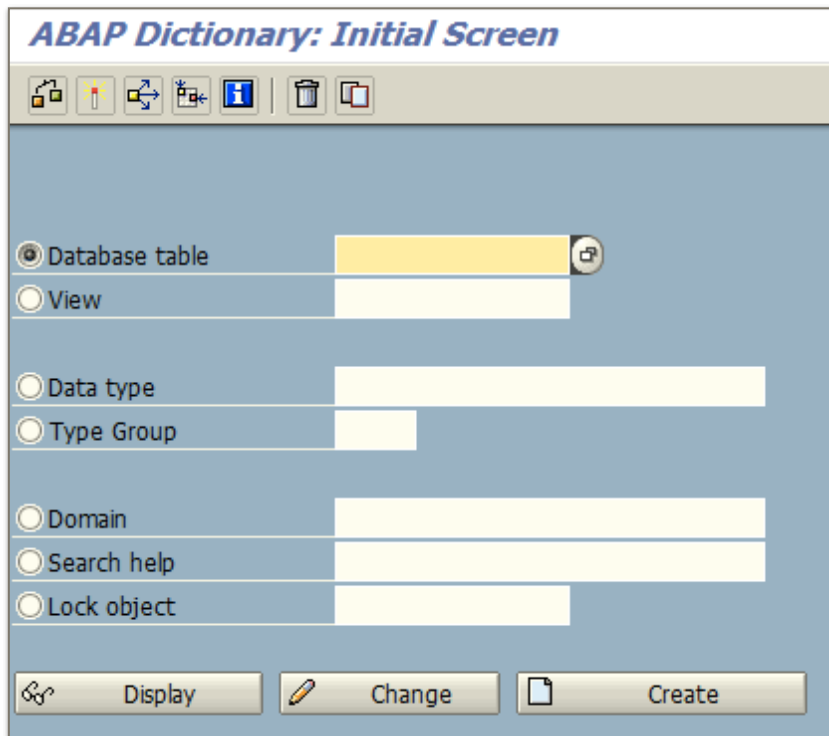
Figur 7. Menu Painter.

i exemplet i figur 7 skapas en menyflik med titeln “Exempelmeny” som innehåller ett kommando med texten “Stäng programmet”.



### 3.1.4 ABAP Dictionary

SAPs ABAP Dictionary innehåller alla definitioner för dataobjekt i SAP, som till exempel databastabeller och datatyper. I SAP finns det också ett program, som ses i figur 8. där man kan se, skapa och ändra objekt i ABAP Dictionary. Programmets programkod är SE11.



Figur 8. ABAP Dictionary.

För att använda ABAP Dictionary skriver man namnet på vad man vill söka efter, ändra, eller skapa, och trycker sedan på en av knapparna nederst i fönstret. Som i de flesta textfält i SAP går det också att söka efter möjliga sökord genom att trycka på den runda knappen till höger om textfältet, som ses i figur x ovan.

### 3.1.5 Number Range Object

Även om databasfunktionerna i SAP är mycket liknande som SQL, finns det ändå några skillnader. Som ett exempel kan vi se på MySQL-koden nedan:

```
CREATE TABLE Exempeltabell
(
  ID int NOT NULL AUTO_INCREMENT,
  Text varchar(255) NOT NULL,
  PRIMARY KEY (ID)
)

INSERT INTO Exempeltabell (Text) VALUES ('Text till tabellen')
```

*Kod 1. MySQL-kodexempel med AUTO\_INCREMENT.*

SQL-koden skapar en tabell med namnet Exempeltabell som har två kolumner, ID och Text. Eftersom kolumnen ID skapas med nyckelorder AUTO\_INCREMENT, kan man nu lägga till rader till tabellen genom att endast skriva ett värde för kolumnen Text, och man behöver inte skriva ett värde för kolumnen ID då det kommer automatiskt att växa med 1.

I SAP finns det inget nyckelord som fungerar som nyckelorder AUTO\_INCREMENT från MySQL. Ett sätt som man kan lösa detta i SAP är genom att skapa en funktion som läser databastabellen, tar det senaste värdet från ID-kolumnen, och lägger till 1.

Ett annat sätt att lösa problemet är genom att använda en SAP-funktion som heter Number Ranges. För detta finns det ett SAP-program som heter Number Range Object Maintenance, som har programkod SNRO, som ses i figur 9.

<b>Number Range Object: Create</b>	
Object	ZNUMMER Number range object has no intervals
Short text	Exempel nummer
Long text	Exempel på automatisk numrering
<b>Interval characteristics</b>	
Subobject data element	
To-year flag	<input type="checkbox"/>
Number length domain	NUM10
No interval rolling	<input type="checkbox"/>
<b>Customizing specifications</b>	
Number range transaction	
Warning %	1,0
Main memory buffering	<input checked="" type="checkbox"/> No. of numbers in buffer 10
<b>Group specification</b>	
Group table	<input type="text"/> <input type="button" value="Maintain text"/>
Subobject field in group table	<input type="text"/>
Fld. NoRangeElement	<input type="text"/> <input type="button" value="Delete group ref."/>
Fields int./ext. no.range no.	<input type="text"/>
Fld. NoRangeNo	<input type="text"/>
Display element text	<input type="checkbox"/>

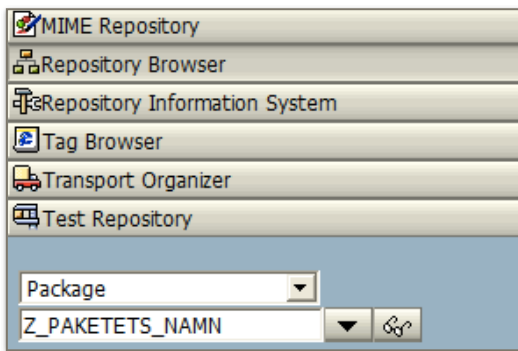
Figur 9. Number Range Object: Create.

I Number Range Object Maintenance-programmet kan man se, skapa, och ändra Number Range-objekt. Med programmet kan man välja kod och namn för objektet, välja vilka sorts nummer det skall innehålla, var det skall börja och sluta, och så vidare.

### 3.2 ABAP Programmering

Det finns flera olika sätt att skapa nya program i SAP, men det går att göra allting från Object Navigator.

Först måste man skapa ett nytt paket. Paket i SAP är liknande som att skapa nya paket i andra utvecklingsmiljöer, paketen innehåller projektets alla filer. För att antingen öppna ett gammalt paket eller skapa ett nytt paket i Object Navigator går man till Repository Browser. I listmenyn väljer man Package, och i textfältet under menyn skriver man paketets namn. I SAP, alla paket som skapas av användare i SAP måste börja med Y eller Z, och får endast innehålla bokstäver, siffror, och understräck. I figur 10 ses menyn i Object Navigator, och ett paket med namnet "Z\_PAKETETS\_NAMN" är öppnat.



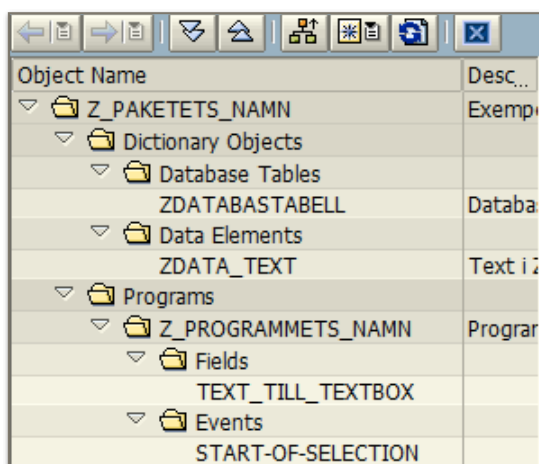
Figur 10. Object Navigator, skapa nytt paket.

Då man skapar ett nytt paket måste man också specificera paketets egenskaper, som t.ex. beskrivning.

Varje gång då man skapar någonting i SAP Developer Workbench, behöver man lägga till det till en Transport Request. En Transport Request är en sorts lista som innehåller information om alla ändringar som har gjorts i SAP.

Object Navigator ber automatiskt om paketets egenskaper och Transport Request då man skapar paketet.

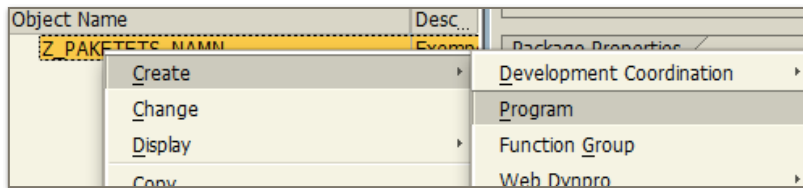
Paketet syns nu i Repository Browser, med alla objekt som hör till paketet. I bilden nedan ses paketet med ett program som innehåller en variabel och en event, och ett databasobjekt som innehåller en databastabell och ett Data element.



Figur 11. Repository Browser.

### 3.2.1 Hello world

För att skapa ett nytt program, högerklickar man på paketets namn i Repository Browser, och väljer Create > Program, som i figur 12.



Figur 12. Skapa nytt program i Object Navigator.

Object Navigator ber sedan om vilket paket programmet skall läggas till, och vilket Transport Request som skall användas.

Då man har skapat programmet har man en fil man skriver ABAP kod. Exemplet nedan är ett Hello World-program skrivet i ABAP.

```

*&-----
*
*& Report  Z_PROGRAMMETS_NAMN
*&
*&-----
*
*&
*&
*&-----
*

REPORT  z_programmets_namn.

DATA text_till_textbox TYPE string VALUE 'Hello from the message box!'.

START-OF-SELECTION.
  WRITE 'Hello world!'.
  MESSAGE text_till_textbox TYPE 'I'.

```

Kod 2. Hello world i ABAP.

I ABAP, alla rader som börjar med ett '\*'-tecken är kommentarer. Kommentarererna i koden ovan är automatiskt gjorda av Object Navigator.

Efter det kommer det en rad som beskriver att filen är en rapport som heter z\_programmets\_namn. Orsaken till varför det kallas rapport istället för t.ex. program, är att i äldre versioner av SAP användes ABAP endast till att skriva rapporter, och för att äldre

program skall vara kompatibla med nya versioner av SAP används fortfarande gamla nyckelord.

Efter programmets namn finns en punkt. I ABAP skall alla satser sluta med en punkt, på liknande vis som alla satser i t.ex. Java eller C++ skall sluta med ett semikolon.

I ABAP har versaler och gemener ingen betydelse, så programmerare kan välja var versaler och gemener används för att göra det så läsvänligt som möjligt. Object Navigator har en funktion som heter Pretty Printer som kan användas för att automatiskt formatera texten med versaler, gemener, och indentering. I exemplet ovan har Pretty Printer formaterat koden så att nyckelorden är skrivna med versaler, variabler är skrivna med gemener, och textsträngar blir inte ändrade.

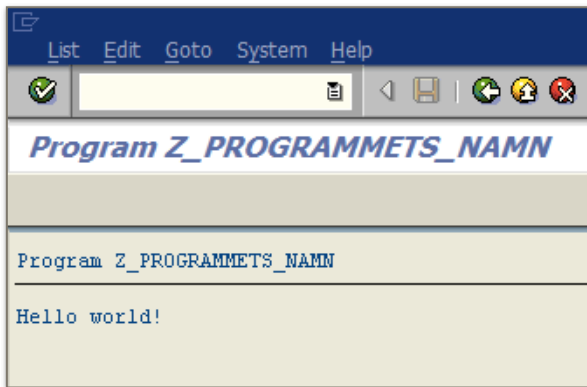
I nästa rad skapas en variabel med namnet `text_till_textbox` av typen `string`, som ges värdet "Hello from the message box!". Textsträngar i SAP skrivs inom enkla citationstecken. Som alla andra satser, slutar den med en punkt.

Nästa rad innehåller nyckelordet "START-OF-SELECTION", som fungerar på ett liknande sätt som main-funktionen i t.ex. Java eller C++. Det är här som SAP börjar då man kör programmet.

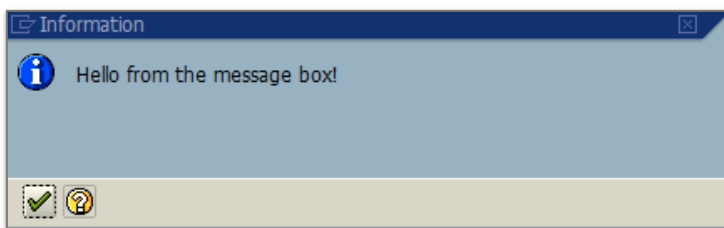
Nästa rad innehåller nyckelordet "WRITE", följt av en textsträng. Den här satsen fungerar på ett liknande sätt som `System.out.println()` fungerar i Java. Den skriver textsträngen i programmets fönster.

Den sista raden innefaller nyckelordet "MESSAGE" som ges värdet av variabeln "text\_till\_textbox" och typen "I". Den här satsen fungerar på ett liknande vis som `JOptionPane.showMessageDialog()` i Java. SAP öppnar en textbox, som innehåller texten som sparats i "text\_till\_textbox". Typen "I" gör att textboxen visas som information, i motsats till t.ex. typen "E" som gör att textboxen visas som ett felmeddelande.

Då man aktiverar och kör programmet i Object Navigator visas textboxen med texten "Hello from the message box!" och huvudfönstret med texten "Hello world!", som ses i figurerna 13 och 14.



Figur 13. Hello world version 1.



Figur 14. Hello world version 2.

### 3.2.2 Databastabell

För att skapa en ny databastabell högerklickar man på paketets namn i Repository Browser, och väljer Create > Dictionary Object > Database Table. Efter det skriver man tabellens namn och specificerar dess egenskaper.

Efter det kan man lägga till nya kolumner till tabellen. Det görs i fliken Fields. Under Field skrivs kolumnens namn, och under Data element skrivs kolumnens typ.

 A screenshot of the 'Fields' tab in the Object Navigator. The window title is 'Transp. Table ZDATABASTABELL New(Revised)'. The 'Short Description' is 'Databastabell'. The 'Fields' tab is active, showing a table with the following columns: Field, Key, Inti..., Data element, Data Type, Length, Decim..., and Short Description.
 

Field	Key	Inti...	Data element	Data Type	Length	Decim...	Short Description
CLIENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MANDI	CLNI	3	0	Client
ID	<input checked="" type="checkbox"/>	<input type="checkbox"/>		NUMC	4	0	ID
TEXT	<input type="checkbox"/>	<input type="checkbox"/>	ZDATA_TEXT		0	0	

Figur 15. Databastabellens egenskaper i Object Navigator.

Tabellen i figur 15 innehåller tre kolumner, som använder två olika sätt att skapa kolumnens typ. Kolumnen CLIENT av typen MANDT är en SAP-specifik kolumn som skall finnas i alla applikationsdatabastabeller. Kolumnen ID är tabellens primärnyckel och innehåller nummer som är fyra tecken långa. ID-kolumnens typ specificeras i samma fönster under Data Type, Length, och Short Description.

Den tredje kolumnen, TEXT, är av typen ZDATA\_TEXT. ZDATA\_TEXT är inte en av SAPs vanliga datatyper, det är istället en specialgjord databastyp. Om man vill skapa en egen datatyp så skriver man först namnet på datatypen i “Data element” kolumnen i tabellen i figur 15, och sen dubbelklickar man på namnet som man skrev. Då går SAP till ett fönster där man skapar datatypen.

The screenshot shows the SAP Data Element creation window for 'ZDATA\_TEXT'. The window has a title bar 'Data element ZDATA\_TEXT New(Revised)' and a 'Short Description' field containing 'Text i ZDATA'. Below this are four tabs: 'Attributes', 'Data Type', 'Further Characteristics', and 'Field Label'. The 'Data Type' tab is active, showing the 'Elementary Type' section with 'Domain' selected. The 'Domain' is set to 'CHAR20' with a 'Character field length' of 20. Below this, the 'Data Type' is 'CHAR' (Character String) and the 'Length' is '20'. Other options like 'Predefined Type', 'Reference Type', and 'Reference to Predefined Type' are visible but not selected.

Figur 16. Data element.

I figur 16 skapas datatypen ZDATA\_TEXT som en textsträng som kan vara 20 tecken lång. Under fliken “Field Label” fylls i vilken titel som kolumnen skall ha då den syns till exempel i en tabell.

I det här exemplet kunde man också ha valt att specificera TEXT-kolumnens typ direkt i tabellen i figur 15 istället för att skapa en egen databastyp, men i det här fallet skapades en ny datatyp endast för att göra det som ett exempel.



### 3.3.3 Spara data till databastabell

För att spara data till en databastabell skapas ett nytt program med koden nedan.

```
REPORT  Z_SPARA_TILL_DATABAS.

DATA:  databastabell TYPE zdatabastabell,
       databastabell_tabell TYPE HASHED TABLE
       OF zdatabastabell WITH UNIQUE KEY id.

START-OF-SELECTION.

databastabell-id = '0001'.
databastabell-text = 'Första raden'.
INSERT databastabell INTO TABLE databastabell_tabell.

databastabell-id = '0002'.
databastabell-text = 'Andra raden'.
INSERT databastabell INTO TABLE databastabell_tabell.

TRY.
  DELETE FROM zdatabastabell.
  INSERT zdatabastabell FROM TABLE databastabell_tabell.
  IF sy-subrc = 0.
    MESSAGE 'Tabellen updaterad.' TYPE 'I'.
  ENDIF.
CATCH cx_sy_open_sql_db.
  MESSAGE 'Tabellen kunde inte uppdateras.' TYPE 'I'.
ENDTRY.
```

*Kod 3. Spara data till databastabell.*

Först skapas variablerna “databastabell” och “databastabell\_tabell” för databastabellen som data skall sparas till.

De tre första raderna efter START-OF-SELECTION lägger en rad till databastabellen.

I TRY-CATCH-satsen sparas databastabellen till SAP.

### 3.3.4 Hämta data från databastabell

För att hämta data från en databastabell skapas ett nytt program med koden nedan:

```
REPORT  z_hamta_fran_databas.

DATA  databastabell TYPE zdatabastabell.

START-OF-SELECTION.
  SELECT * INTO databastabell FROM zdatabastabell.
  WRITE: / databastabell-id, databastabell-text.
ENDSELECT.
```

*Kod 4. Hämta data från databastabell.*

Först skapas en variabel för databastabellen som data skall hämta från.

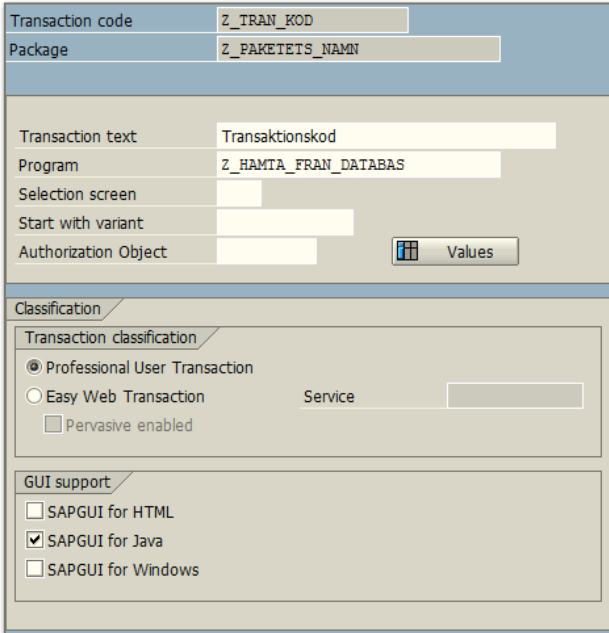
I de tre raderna efter START-OF-SELECTION hämtas data från databastabellen och skriver datan med WRITE-funktionen.

I både exemplet med att spara data till en databastabell och i exemplet med att hämta data från en databastabell går det att se att ABAP använder SQL-kommandon för databashantering.

### 3.3.5 Skapa transaktionskod

Även om det går att köra programmen direkt via Object Navigator, måste man skapa en transaktionskod för att användare skall kunna köra programmen.

För att göra detta, högerklickar man på paketets namn i Repository Browser, och väljer Create > Other (1) > Transaction. I exemplet nedan är transaktionskoden som användare skulle använda för att starta programmet Z\_TRAN\_KOD, och när transaktionen startas går den till programmet Z\_HAMTA\_FRAN\_DATABAS i paketet Z\_PAKETETS\_NAMN.



The screenshot shows the SAP transaction code creation dialog box. The fields are filled with the following information:

- Transaction code: Z\_TRAN\_KOD
- Package: Z\_PAKETETS\_NAMN
- Transaction text: Transaktionskod
- Program: Z\_HAMTA\_FRAN\_DATABAS
- Selection screen: (empty)
- Start with variant: (empty)
- Authorization Object: (empty) with a 'Values' button.

The 'Classification' section is expanded, showing:

- Transaction classification:
  - Professional User Transaction
  - Easy Web Transaction
  - Pervasive enabled
- Service: (empty)

The 'GUI support' section is also expanded, showing:

- SAPGUI for HTML
- SAPGUI for Java
- SAPGUI for Windows

Figur 17. Transaktionskod.

## 4 Utförande

I detta examensarbete har applikationen skapats i SAP NetWeaver Application Server ABAP 7.03 SP04, som är en utvecklingsversion av SAP som finns gratis tillgängligt på SAPs webbsida. Efter detta behöver applikationen flyttas över till Celsas SAP-installation. På grund av att applikationen är inte skapad i Celsas SAP är vissa delar, t.ex. Celsas databastabeller och menyer inte exakt lika som det som finns i Celsas SAP. Istället är de förenklade versioner av vad som finns i Celsas SAP.

För att skapa applikationen skapas först ett nytt SAP-programpaket. Till paketet läggs ett program med en main-funktion som visar programmets huvudskärm.

```
PROGRAM z_main.

CLASS cl_main DEFINITION.
  PUBLIC SECTION.
    CLASS-METHODS main.
ENDCLASS.
CLASS cl_main IMPLEMENTATION.
  METHOD main.
    CALL FUNCTION 'Z_MAINSCREEN'.
  ENDMETHOD.
ENDCLASS.
```

*Kod 5. Programmets main-funktion.*

Förutom main-funktionen skapas det en transaktionskod som går till programmets main-funktion, som användaren använder då den vill köra programmet.

The screenshot shows the SAP configuration dialog for a transaction code. The fields are filled with the following values:

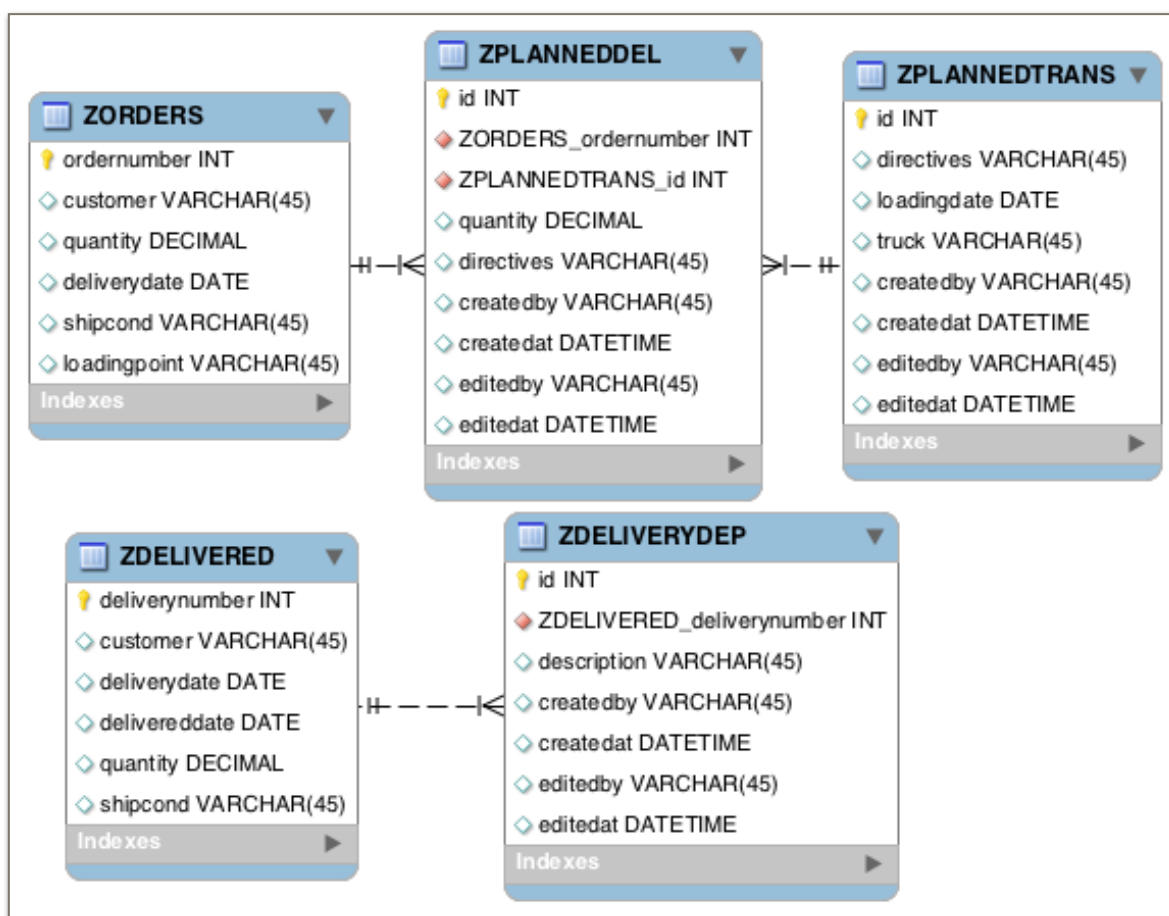
Transaction code	z_CELSA
Package	Z_CELSAPLANNING
Transaction text	Transport Planning for Celsa
<input type="checkbox"/> OO transaction model	
Class Name	CL_MAIN
Method	MAIN
<input checked="" type="checkbox"/> Local in program	Z_MAIN
Authorization Object	
	<input type="button" value="Values"/>

*Figur 18. Programmets transaktionskod.*

## 4.1 Databas

Programmet skall delvis använda Celsas nuvarande databas där det finns information om till exempel beställningar, kunder och material. Till dessa skall det också skapas databastabeller för planerade leveranser, lastningar, och leveranssäkerhet. Då varje beställning kan finnas levereras i flera lastningar, och varje lastning kan innehålla flera leveranser, kopplas de två tabellerna samman med en många-till-många relation. Resten av datan, som t.ex. leveransadress, kund, osv. hämtas från Celsas nuvarande tabeller.

Diagrammet nedan visar applikationens databastabeller. De tre övre tabellerna hör till programmets transportplanering, och de två nedre hör till programmets leveranssäkerhet. Tabellerna för transportplanering och tabellerna för leveranssäkerhet är inte hopkopplade. Som det tidigare konstaterades så är tabeller inte exakt lika som de som finns i Celsas SAP-installation. Tabellerna ZORDERS och ZDELIVERED är förenklade versioner av liknande tabeller som finns i Celsas SAP-installation.



Figur 19. EER-diagram.

Tabellen ZORDERS kommer från Celsas nuvarande tabeller för bl.a. beställningar och materialdata.

Tabellen ZPLANNEDDEL är för planerade leveranser, och där sparas det information om vilken beställning det finns i leveransen, hur mycket som skall levereras, och vilken transport leveransen skall lastas på.

Tabellen ZPLANNEDTRANS är för planerade lastningar, och där sparas det information om när transporten skall lastas, och på vilken bil det skall lastas.

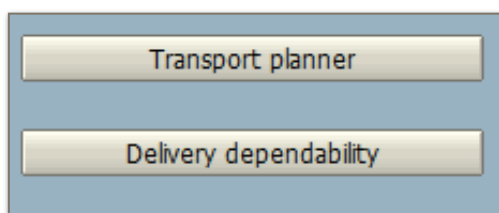
I både ZPLANNEDDEL och ZPLANNEDTRANS finns det information om när den planerade lastningen eller leveransen skapades och av vem, när och av vem leveransen eller lastningen har senast ändrats, och det går att skriva lastnings- och leveransdirektiv om lastningarna och leveranserna.

Tabellen ZDELIVERED kommer från Celsas nuvarande tabeller för bl.a. leveranser, transporter, och materialdata.

Programmet för leveranssäkerhet har en tabell som inte är kopplat till applikationens andra andra tabeller, ZDELIVERYDEP. Den innehåller leveransnummer, text med information om varför beställningen blev försenad, vem som skrev orsaken och när personen skrev orsaken, samt när och av vem den har senast ändrats. Som med tabellerna för transportplanering, hämtas resten av informationen från Celsas nuvarande tabeller.

## 4.2 Programmets huvudsida

Då programmet har flera olika funktioner, skapas det en huvudsida som fungerar som ingångspunkt till programmet där användare lätt kan komma åt programmets olika funktioner. I huvudprogrammet finns det knappar som går till programmet som används för transportplanering, och programmet som används för att se leveranssäkerhet.



Figur 20. Programmets huvudsida.

Huvudsidans användargränssnitt är skapat i Screen Painter, och består endast av tre olika knappar. Knapparna läggs till i Screen Painter, och förutom visuell formatering ges de namn, text, och funktionskod. Knapparnas "namn" används av vissa funktioner i SAP, "texten" är den som finns på knappen, och "funktionskoden" används då man i programkoden vill läsa vilken knapp som användaren tryckte på.

För att programmet skall kunna läsa vilken knapp som användaren tryckte på, skapas skärmen med en så kallad PAI event (Process After Input). SAP går till skärmens PAI efter att användaren t.ex. tryckt på en knapp. I skärmens Flow Logic läggs det till en rad där programmet går till programmets user\_command\_0001-modul.

```
PROCESS AFTER INPUT.
MODULE user_command_0001.
```

*Kod 6. Flow Logic till huvudskärmen.*

user\_command\_0001 gör att då användaren trycker på en knapp, går programmet till den knappens skärm.

```
MODULE user_command_0001 INPUT.
CASE sy-ucomm.

    WHEN 'MAINBUT1'.
        CALL FUNCTION 'Z_TRANSPLAN'.
    WHEN 'MAINBUT2'.
        CALL FUNCTION 'Z_DEPENDABILITY'.

ENDCASE.
```

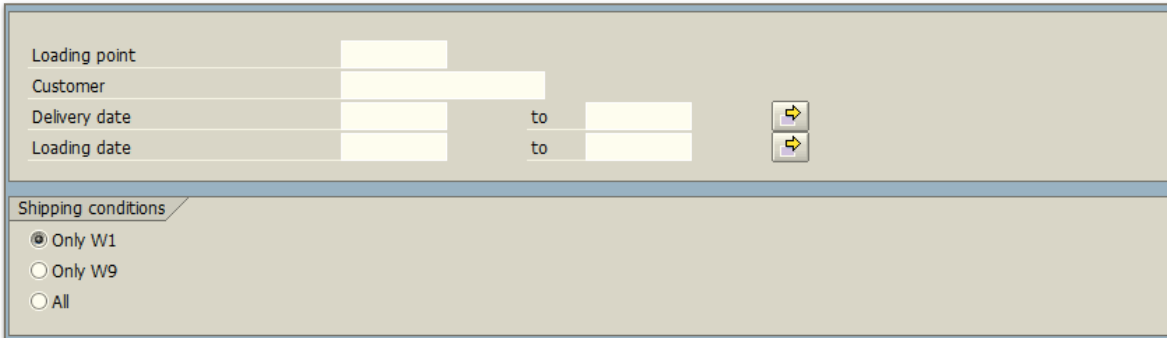
*Kod 7. Modulen user\_command\_0001.*

I koden ovan, CASE sy-ucomm är satsen som SAP går till då användaren har tryckt på en knapp. MAINBUT1-2 är de olika knapparnas funktionskoder. Kommandot CALL FUNCTION öppnar programmets gång till programmets olika delar.

### 4.3 Program för transportplanering

Programmet för transportplanering består av två olika huvudfönster: fönstret där man söker efter beställningar, och fönstret där man ser beställningar.

Då programmet först startar går man till fönstret där man söker efter beställningar, so ses i figur 21. Här går det att specificera leveransdatum som hämtas från beställningen, lastningsdatum som hämtas från planeringen, lastningsplats, leveransklass, och kund.



The screenshot shows a search window with the following fields and options:

- Loading point: [text input]
- Customer: [text input]
- Delivery date: [text input] to [text input]
- Loading date: [text input] to [text input]

There are two arrow buttons to the right of the date fields. Below these fields is a section titled "Shipping conditions" with three radio button options:

- Only W1
- Only W9
- All

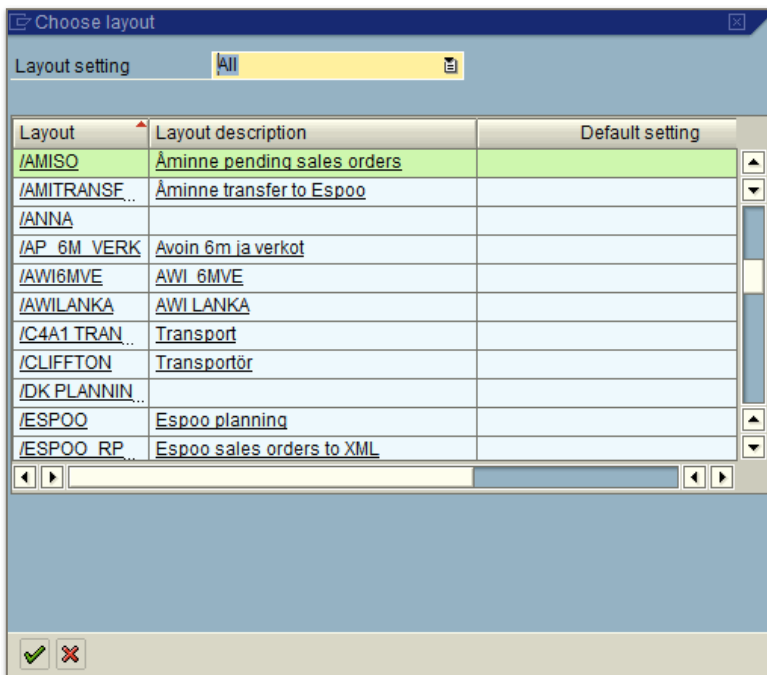
*Figur 21. Sökning efter beställningar till transportplanering.*

Programmet är skapat som en Function Group, och skärmen är en vanlig SELECTION-SCREEN. Därefter skapas en Function Module, som visar skärmen.

Då man har valt vilka beställningar man vill söka efter tas man till fönstret där man ser beställningar. Det här fönstret består av en meny där man kommer åt olika funktioner, och en tabell där informationen visas.

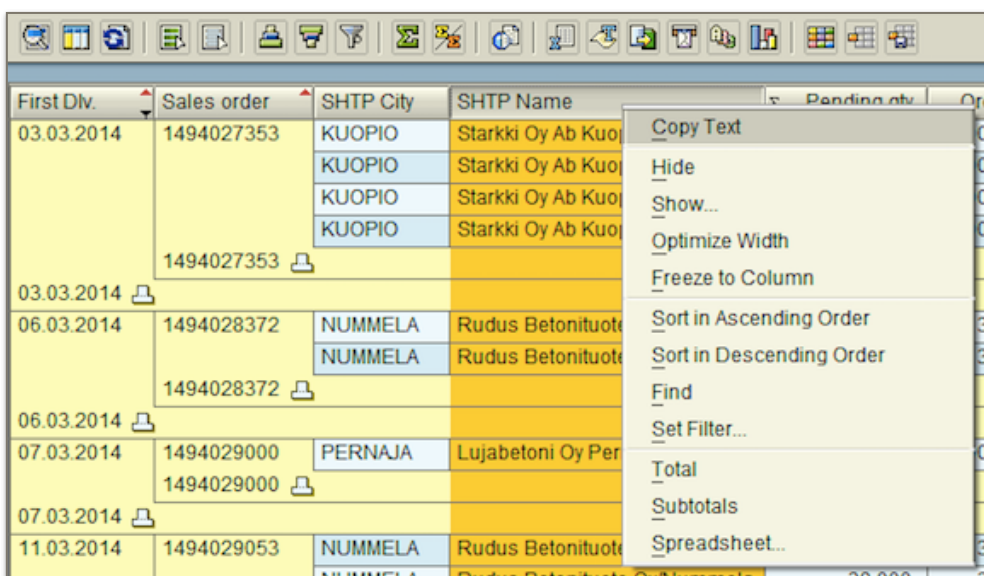
Menyer och tabeller som används till programmet då det är flyttat till Celsas SAP är Celsas egna menyer och tabeller som används i alla deras program. Därför har det här programmet skapas med endast en simpel tabell som ett exempel.

SAP har en Layout-funktion, som ses i figur 22, som gör att man kan spara och öppna olika vyer, där man kan t.ex. välja vilka kolumner som skall synas, sortering, och filter. för båda programmen som gjordes i detta arbete behövs det huvudsakligen två olika layouts, en som är gjord så att datan skall vara enkel att läsa i SAP, och en som används då datan skall skrivas ut till Excel-dokument.



Figur 22. Layout för tabeller.

I menyn som ses i överst i figur 23 finns det knappar bland annat för välja, ändra, och skapa nya layouter som ändrar hur tabellen är utformad, vilka kolumner den innehåller, osv. Till tabellen skapas det layouter för att se alla beställningar, endast beställningar som redan är planerade, endast beställningar som är oplanerade, endast beställningar som är förberedda för lastning, och endast beställningar som inte är förberedda för lastning. Där finns också knappar för att uppdatera informationen i tabellen, och för att spara informationen till dokument.



Figur 23. Menyner.



Till programmets meny läggs det också till knappar för att ändra och skapa nya planerade leveranser, och kombinera leveranserna till transporter.

#### **4.5 Program för leveranssäkerhet**

Lika som programmet för transportplanering består också programmet för leveranssäkerhet av två olika huvudfönster: fönstret där man söker efter beställningar, och fönstret där man ser beställningar. De skapas också på samma sätt som till programmet för transportplanering.

I fönstret där man söker efter beställningar finns det fält för lastningsplats och leveransdatum. Leveranssäkerhetsrapporten endast används för leveranser där Celsa arrangerar transporten.

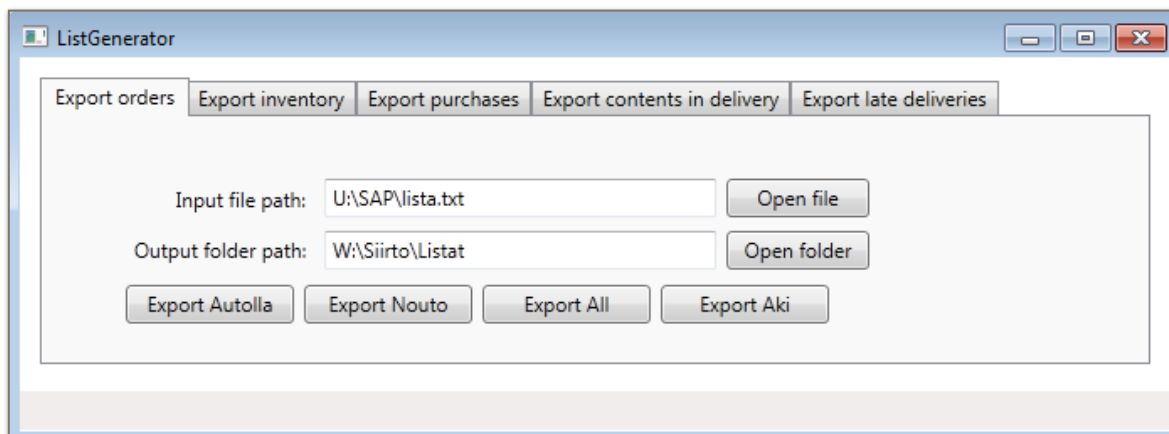
Som i programmet för transportplanering, består programmet för leveranssäkerhets andra fönster av menyer och en tabell, som båda i Celsas SAP använda Celsas menyer och tabell.

Menyn innehåller samma knappar som i programmet för transportplanering, men knapparna för transportplanering är ersatta med en knapp för leveranssäkerhet.

Programmet för leveranssäkerhets huvudfunktion är att det går att skriva tilläggsinformation om varför en leverans blev försenad. För att göra detta väljer man leveransen i tabellen och trycker på knappen där man kan skriva tilläggsinformation. Då öppnas ett fönster där man kan skriva informationen, som sparas i programmets databastabell.

#### **4.6 Skriva ut rapporter i Excel-format**

Då formateringen av Excel-dokument i SAP är över detta arbetets gränser, sparas informationen från SAP till CSV-filer, och sedan anpassas ett av företagets nuvarande program, ListGenerator, som ses i figuren nedan, för att kunna läsa in CSV-filerna och skapa Excel-dokument i Open Office XML Spreadsheet-format.



*Figur 24. Programmet ListGenerator.*

För att kunna skriva ut rapporter till Excel-format går användaren först till programmet som användaren vill skriva ut data från, och söker efter data på samma vis som man vanligtvis gör. Efter det väljer användaren att använda en layout som är gjord för data som skrivs ut till Excel-format, och väljer därefter att spara all data till en textfil.

Efter att textfilen har sparats från SAP öppnar användaren filen i ListGenerator-programmet och väljer vilken sorts rapport som skall skapas. ListGenerator-programmet skapar då ett Excel-dokument.

## **5 Resultat och diskussion**

Som resultat av examensarbetet finns en applikation som är gjord i en utvecklingsversion av SAP, som kan flyttas till Celsas SAP-installation så att det kan användas av Celsa. Delar som behöver ändras i applikationen då den flyttas är bland annat databaskopplingar, menyfunktioner, och numreringar till databastabellernas primärnycklar.

Enligt expeditionsavdelningen i Celsa skulle applikationen i detta arbete fylla dess syfte och passa in i deras nuvarande arbetssystem, men det finns några olika funktioner som skulle kunna läggas till applikationen i en möjlig vidareutveckling av den.

En av de funktioner som kan läggas till om applikationen vidareutvecklas är att ändra den delen av applikationen som används för att skriva ut rapporter, så att användare inte längre behöver använda tilläggsapplikationen ListGenerator för att skapa Excel-dokument. Möjligheter som kan utredas för att skriva ut rapporter som godkänns av transportföretaget

direkt från SAP är till exempel några av SAPs inbyggda system för att skriva ut rapporter. Om inget av de inbyggda systemen skulle vara lämpade, går det till exempel att skapa ett programbibliotek till SAP som skapar filer i Open Office XML Spreadsheet-format, på samma sätt som tilläggsapplikationen nu gör.

En annan typ av funktioner som kan läggas till i möjlig vidareutveckling av applikationen är möjligheten att skapa leverans- och transportdokument direkt via applikationen för transportplanering. Just nu är den endast gjord för att planera lastningar, men det skulle också vara möjligt att använda det för att skapa SAP-dokument. Med denna funktion kunde skapandet av SAP-dokument också effektiveras jämfört med Celsas nuvarande system.

På liknande sätt som applikationen för transportplanering kunde vidareutvecklas för att skapa SAP-dokument, kunde den också vidareutvecklas för att skriva ut lastningslistor och fraktsedlar.

## Källförteckning

- Celsa Group (2013a). *About us*. <http://www.celsagroup.com/secciones/about/presentation.aspx> (hämtat: 26.11.2013)
- Celsa Group (2014). *Where we are*. <http://www.celsagroup.com/secciones/about/where.aspx> (hämtat: 1.02.2014)
- Celsa Steel Service (2013b). *Yleistä*. <http://celsa-steelservice.fi/yleista/> (hämtat: 26.11.2013)
- Keller, H. & Kruger, S. (2011). *ABAP Objects - ABAP programming in SAP NetWeaver*. Bonn: Galileo Press
- SAP AG (2013a). *SAP Company Information*. <http://global.sap.com/corporate-en/our-company/index.epx> (hämtat: 26.11.2013)
- SAP AG (2013b). *SAP Customer Testimonials, Stories Spotlight*. <http://global.sap.com/customer-testimonials/index.epx> (hämtat: 26.11.2013)
- SAP AG (u.å.). *Overview ERP Software*. <http://www.sap.com/pc/bp/erp/software/overview.html> (hämtat: 24.02.2014)
- SAP AG (2001a). *SAP GUI for HTML*. <http://help.sap.com/printdocu/core/print46c/en/data/pdf/BCFESITSSAPGUIHTML/BCFESITSSAPGUIHTML.pdf> (hämtat: 14.02.2014)
- SAP AG (2001b). *SAP GUI for HTML*. [http://help.sap.com/erp2005\\_ehp\\_04/helpdata/en/4d/aeae42cd7fb611e10000000a155106/content.htm](http://help.sap.com/erp2005_ehp_04/helpdata/en/4d/aeae42cd7fb611e10000000a155106/content.htm) (hämtat: 14.02.2014)
- SAP AG (u.å.b). *Training offerings for mySAP ERP*. <https://websmp106.sap-ag.de/~sapidp/011000358700003272282004> (hämtat: 24.02.2014)
- SAP AG (u.å.c). *What is Object Orientation*. [http://help.sap.com/saphelp\\_nw70/helpdata/en/c3/225b5654f411d194a60000e8353423/content.htm](http://help.sap.com/saphelp_nw70/helpdata/en/c3/225b5654f411d194a60000e8353423/content.htm) (hämtat: 22.02.2014)
- SAP DB Library (2013c). *The SAP DB Database System*. <http://www.sapdb.org/7.4/hthelp/97/026b3ae886616ae10000000a114084/frameset.htm> (hämtat: 27.11.2013)
- SAP Insider (2011). *How to Get the Most Out of ABAP and Java in the Context of SAP Technology: Part I*. <http://sapinsider.wispubs.com/Assets/Articles/2011/July/How-To-Get-The-Most-Out-Of-ABAP-And-Java-In-The-Context-Of-SAP-Technology-Part-1> (hämtat: 11.02.2014)
- SAPDev (u.å.). *SAP GUI for HTML - running SAP on a PC that does not have the SAP client installed*. [http://www.sapdev.co.uk/webapps/web\\_htmlgui.htm](http://www.sapdev.co.uk/webapps/web_htmlgui.htm) (hämtat: 17.02.2014)
- Schreckenbach, S. (2011). *SAP Administration - Practical Guide*. Bonn: Galileo Press
- World Steel Association (2012). *Top steel-producing companies*. <http://www.worldsteel.org/dms/internetDocumentList/downloads/statistics/Topsteelproducers/document/Top%20steel%20producers.pdf> (hämtat: 29.11.2013)