

Outlook Addin for Platina

Aron Grahn

Luleå tekniska universitet
Högskoleingenjörsprogrammet
Datateknik
LTU Skellefteå
Avdelningen för Mobila nätverk och programutveckling

Outlook Addin for Platina™

”A part of the Office Integration for Platina™”



Aron Grahn

22 november 2006

Förord

Syfte med rapporten

Syftet med denna rapport är att ge en beskrivning av det examensarbete som jag utfört under hösten 2006. Jag kommer att gå igenom vissa delar av den kod som skrivits de senaste 10 veckorna och även visa några bilder från det färdiga programmet.

Ett stort tack till:

Jimmy Öhlin, Chef på utvecklaravdelningen

- För att han gav mig erbjudande att få göra examensarbete här på EBI Systems.

Staffan Nilsson, Examinator på LTU

- För att han hjälpte mig komma igång med examensarbetet.

Alla på EBI Systems utvecklaravdelning i Skellefteå

- För alla tips på hur jag skulle lösa diverse kodning, samt all övrig hjälp som tex. konfigurering av programvara och felsökning av kod.

Outlook Addin for Platina™

Aron Grahn

Abstract

This project was performed during 10 weeks in the autumn of 2006.

The goal was to combine Microsoft Outlook 2003 with EBI Systems platform: Platina.

For those who don't know anything about Microsoft Outlook I give you the definition:

Microsoft® Office Outlook® 2003 is the personal information manager and communication program that gives you a unified place to manage e-mail, calendars, contacts, and other personal and team information.*[1] For more information about Platina read appendix F.

The goal was to make MS Outlook show files and functions from Platina, make Outlook to communicate with Platina and save mail from Outlook directly into Platina database.

To make this possible there had to be a lot of reading on how MS Outlook was build, and also how Platina was build. The biggest part of my code was written in C#, but also JavaScript, VB-Script, Active-X and HTML-programming was needed to make this addin function.

Outlook Addin for Platina™

Aron Grahn

Sammanfattning

Examensarbetet handlade om att integrera Microsoft Outlook 2003 med EBI Systems plattform Platina. För de som inte vet vad Microsoft Outlook är så kommer här Microsofts definition:

Microsoft® Office Outlook® 2003 är ett personligt informationshanterings- och kommunikationsprogram där du hanterar e-post, kalendrar, kontakter och annan personlig och grupprelaterad information.*[2] Vill du veta mer om Platina så finns det information i Bilaga E.

Målet med examensarbetet var att få tillgång till filer och funktioner från Platina i MS Outlook. Användaren av Outlook skulle även kunna spara e-post direkt till Platina databasen på olika sätt, till ärenden, till handlingar och till vanliga mappar.

För att kunna göra detta krävdes djupare förståelse om hur både MS Outlook är uppbyggt och hur Platina är uppbyggt. Större delen av koden är skriven i C#. Har också arbetat med JavaScript, VB-Script, Active-X och vanlig HTML-programmering.

Förklaringar och begrepp

Här är en förklaring av ord och begrepp som kommer att användas i rapporten.

Platina – Det program som jag arbetar mot. Se bilaga E.

MS – Förkortning av namnet Microsoft.

Form – Ett grafiskt gränssnitt i MS Windows som används för att visa data (formulär).

Addin – Engelska för tillägg. I detta fall är det ett tillägg till programmet Outlook.

Aspx – Active Server Page. En webbsida som innehåller html kod och script.

Event – Engelska för händelse. För oss betyder det att en viss händelse har inträffat. Man kan välja att vänta på en viss händelse, och när det inträffar så exekveras en viss kod.

EventHandler – Koden som exekveras när en händelse inträffat.

Toolbar – Verktygsfält med knappar och text.

Pollning – Innebär att man undersöker en viss sak vid olika tidpunkter, i detta fall undersöker jag innehållet i en mapp.

Innehåll

FÖRORD	2
SYFTE MED RAPPORTEN.....	2
ETT STORT TACK TILL:.....	2
ABSTRACT	3
SAMMANFATTNING	4
FÖRKLARINGAR OCH BEGREPP	5
INNEHÅLL	6
1 INLEDNING	7
1.1 BAKGRUND.....	7
1.2 MÅL.....	7
2 GENOMFÖRANDE	8
2.1 INITIALT ARBETE	8
2.1.1 Outlook.....	8
2.1.2 Platina.....	8
2.1.3 Programmeringsverktyg.....	8
2.2 PROGRAMMERING C#.....	9
2.2.1 Mapplista-fönstret.....	9
2.2.2 Windows Forms	11
2.2.3 Menyner	13
2.3 UTVECKLING OCH TESTER	14
2.3.1 Mappar.....	14
2.3.2 Forms.....	14
2.3.3 Menyner	15
3 RESULTAT	16
4 DISKUSSION	18
5 REFERENSER	18
BILAGOR	19
A. UPPDATERA KATALOGSTRUKTUREN	19
B. BILDER PÅ FORMS.....	20
C. SKAPA MENY I E-POST FÖNSTRET.....	21
D. FÅNGA EVENT, VÄNTA 500 MILLISEKUNDER.....	22
E. PLATINA ÖVERSIKT	23
F. PLATINA INFORMATION ENGLISH	24

1 Inledning

1.1 Bakgrund

EBI Systems är ett innovativt och entreprenördrivet mjukvaruföretag som ser mycket ljust på framtiden. EBI Systems affärsidé är att leverera lösningar som på ett smartare sätt löser ledande företag och organisationers behov av att:

- Integrera och relatera data mellan olika interna och externa system
 - Ge alla medarbetare rätt information vid rätt tidpunkt i rätt format
- EBI Systems gör detta genom att erbjuda en väl paketerad och flexibel mjukvarulösning av hög kvalitet i form av PLATINA™.

Sommaren 2007 släpper EBI Systems deras nyaste version av Platina, och programmerarna arbetar för fullt på att få klart den nya versionen. Det är här jag kommer in i bilden pga. att företaget har bestämt att när de släpper den nya versionen skall den kunna integreras med Microsoft Office-paketet. De kunder som använder Microsoft Office på sina datorer skall kunna installera ett addin-paket som ger dem möjligheten att via Microsofts produkter få tillgång till funktioner i Platina. Officepaketet består av flera olika program där Outlook är ett av dessa. Word som är ett annat program i Officepaketet hade de redan en programmerare som börjat arbeta med. Jag fick uppgiften att börja med programmering av Outlook. Officepaketet är släppt i flera versioner, men EBI Systems har valt att inrikta sig på version 2003 och nyare. Programmeringen skulle huvudsakligen göras i c# vilket passade mig alldeles utmärkt eftersom jag i en projektkurs tidigare under utbildningen programmerat i det språket. Detta examensarbete speglade väldigt bra den utbildning jag gått. Utan tvekan valde jag att göra detta examensarbete under 10 veckor hösten 2006.

1.2 Mål

Målet med detta examensarbete var att på 10 veckor göra en addin till Outlook som skulle kunna tillämpas när de släpper den nya versionen av Platina. Vad denna addin skulle göra skulle bestämmas under arbetets gång. På det viset fick jag hela tiden nya uppgifter att klara av.

2 Genomförande

2.1 Initialt arbete

2.1.1 Outlook

Outlook var för mig ett sedan tidigare bekant program som jag använt för att skicka och ta emot e-post. Däremot hade jag aldrig programmerat mot Outlook. Vid start av projektet visste jag ingenting om vilka objekt och vilka möjligheter som fanns att göra rent programmeringsmässigt. Efter att ha installerat Office-paketet, SQL-server, Platina, MS Visual Studio .Net 2003, MS Visual Studio .Net 2005, .Net framework 2.0 och nyaste Platina klienten var det dags att studera strukturen i Outlook. De två första dagarna gick åt till att installera alla program på datorn och göra en enkel addin för att se att det fungerade att programmera mot Outlook. Dag tre sökte jag på Internet efter manualer om c# programmering mot Outlook. Det visade sig att det fanns mycket bra information. Bästa informationen hittade jag på MSDN *[3].

2.1.2 Platina

Eftersom jag aldrig tidigare använt Platina så började jag med att undersöka hur det var uppbyggt. Jag utforskade Platinas programkatalog och hittade mängder av c#- kod. Att förstå sig på all kod var till en början väldigt svårt. Istället började jag studera koden som rörde den bit jag för tillfället höll på med. Koden var väldigt bra strukturerad, vilket gjorde det var enkelt att hitta det man ville lära sig mer om.

I början hade jag svårt att förstå hur man skulle använda handlingar och ärenden i Platina. För att klara av min uppgift så räckte det med att jag visste hur man sparade filer till ärenden och handlingar. Hur handlingar och ärenden används är svårt att förklara eftersom alla företag använder dessa på olika sätt.

Det är helt enkelt upp till användaren hur ärenden och handlingar används i Platina.

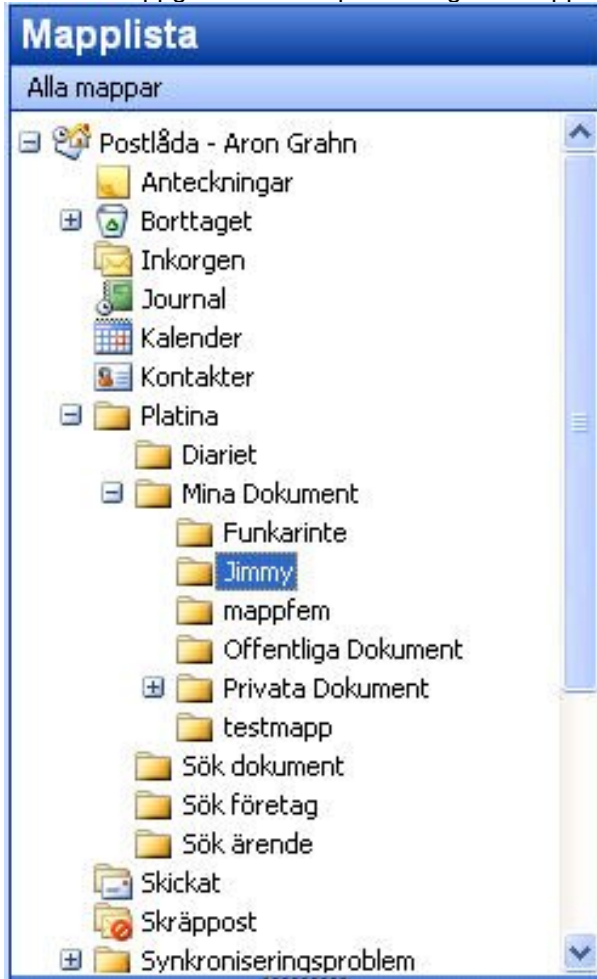
2.1.3 Programmeringsverktyg

MS Visual Studio .Net 2003 och MS Visual Studio .Net 2005 var de programmeringsverktyg jag använde mig av. 2005 versionen användes då jag skrev huvudprogrammet som innehåller alla forms och anrop. Huvudprogrammets kod kommer det exempel på längre fram i rapporten. 2003 versionen användes för att skapa ett Project "Outlook" i Platinas programkatalog. Där placerade jag kod för att visa webbsidor. På detta sätt kunde jag använda befintliga JavaScript som fanns i Platinas programkatalog. Många aspx-sidor skapades i 2003 versionen, den mest komplicerade var den första jag skulle göra som skulle visa en lista på filer då man klickade på en katalog.

2.2 Programmering c#

2.2.1 Mapplista-fönstret

Min första uppgift var att skapa kataloger i "Mapplista" fönstret.



Bilden visar katalogen Platina som jag själv skapat. Alla mappar under Mina dokument har jag läst ut ur Platinas databas.

Uppgift ett var att skapa katalogen "Platina". I kod ser det ut såhär:

```
private void createPlatinaFolder()
{
    Outlook.Namespace space = oAppobj.GetNamespace("MAPI");
    Outlook.MAPIFolder inbox =
space.GetDefaultFolder(Outlook.OlDefaultFolders.olFolderInbox) as
Outlook.MAPIFolder;
    Outlook.MAPIFolder newF = null;
    if (inbox != null && inbox.Parent is Outlook.MAPIFolder)
    {
        Outlook.MAPIFolder root = (Outlook.MAPIFolder)inbox.Parent;
        newF = root.Folders.Add("Platina", missing);
    }
}
```

Denna kod skulle exekveras varje gång man startade Outlook, men metoden fungerade inte om katalogen redan fanns.

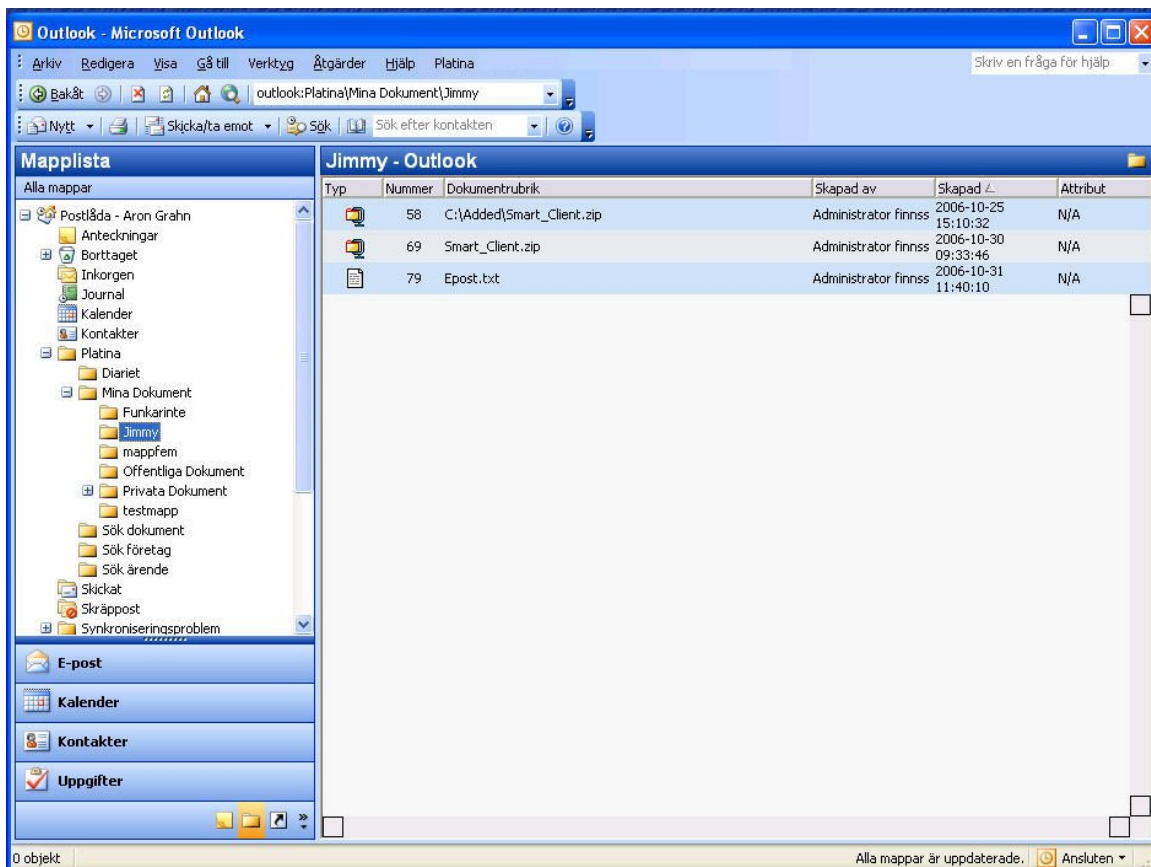
Därför fick jag lägga in en kontroll som tog bort Platina-mappen om den redan fanns när Outlook startades. Då skrevs nästa metod, som körs vid start av Outlook, men före createPlatinaFolder():
`private void deletePlatinaFolder()`

Vid start av min addin i detta läge skapades "Platina"-katalogen. Nu var det dags att fylla den med underkataloger. Sök dokument, sök ärende och sök företag kopplades till aspx-sidor som jag senare skulle komma att programmera i 2003 versionen av Visual Studio. Diariet är även det en aspx-sida från samma projekt.

Exempel på hur man kopplar en mapp till att visa innehåll som en webbsida:

```
newF = _Folder.Folders.Add("LTUs hemsida", missing);  
newF.WebViewURL = "http://www.ltu.se";  
newF.WebViewOn = true;
```

Katalogen "mina dokument" är en katalog som ska spegla alla filer och mappar som finns i Platinas databas. Filerna valdes att visas upp på en aspx-sida, naturligtvis skriven i .Net 2003. För att rita mapparna däremot skulle det göras anrop direkt från huvudprogrammet till Platina-servern. Eftersom katalogstrukturen kan ändras undertiden man sitter i Outlook så blev jag tvungen att lägga in en kontroll som uppdaterar alla mappar då man klickar på "mina dokument" eller "Platina" katalogen. För att användaren av Outlook skulle få se dessa nya ändringar så blev man tvungen att först ta bort alla mappar i mina dokument och sedan rita upp dessa igen. Detta gjordes genom ett event som initieras vid start av Outlook. Se bilaga A.



Denna bild visar innehållet i mappen "jimmy". Visar ikon, filnamn, id-nummer i databasen, vem som är skaparen samt datum för skapande. Allt visas snyggt på en aspx-sida som jag skrivit.

2.2.2 Windows Forms

Efter fyra veckors jobb med att få mapplista-fönstret att fungera med aspx-sidor kom nästa utmaning. När jag drog ett e-post meddelande till mappen "diariet" som jag tidigare skapat så skulle man få upp ett formulär (en form) som skulle fyllas i innan man fick spara e-posten som en handling.

The diagram shows a Windows Forms dialog box with two tabs: "Diarium DD" and "Postlista DD". The "Diarium DD" tab is selected and contains a text area with the following text: "Lista med -mail (ska bli html i platina -x Bifogade filer)". To the right of this text area are several controls: a "Verk kod" dropdown menu, a "Handlingstyp" dropdown menu, a large text area labeled "Initierare", a "Rubrik" dropdown menu, and an "Innehåll" text area. At the bottom right of the dialog is a "Spara" button.

Figuren ovan visar hur formuläret skulle se ut.

Diarium skulle innehålla de diariet som fanns att välja mellan i databasen.

Postlista skulle innehålla de postlistor som fanns i databasen.

Verksamhets-koder skulle visas i en lista och när man valde en verksamhetskod så fick man veta en identifierare som man i sin tur använde för att fylla listan handlingstyp med aktuella värden.

Olika verksamhetskoder hade olika handlingstyper.

Nedan visas metoden som anropas efter man valt en verksamhetskod. Lagg märke till att "verkskodid"-variabeln används vid anropet för att plocka ut de val som finns för den valda verksamhetskoden.

```
private void handlingstypDraw()
{
    comboBox4.Items.Clear();
    PostList.wsPostList ws = new PostList.wsPostList();
    ws.UseDefaultCredentials = true;
    PostList.DsPostList res =
ws.ClientSearchDeedTypes(verkskodid, "");
    handlingslista = new String[res.DeedType.Rows.Count];
    int i = 0;
    foreach (DataRow row in res.DeedType.Rows)
    {
        comboBox4(row[4].ToString());
        handlingslista[i] = (row[4].ToString() + "nummer" +
row[0].ToString());
        i++;
    }
}
```

The screenshot shows a Windows application window titled "Diarieför som handling". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main content area is a light beige color and contains several form fields and buttons. On the left side, there are three dropdown menus: "Diarium" (set to "Övningsdiarium"), "Postlista" (set to "Skellefteå"), and "Koppla dokument till Diarie" (containing "Fil: dumdum.txt" and "Mail: hhhh"). Below these are two buttons: "ta bort" and "lägg till". On the right side, there are three more dropdown menus: "Verks.kod" (set to "Övn"), "Handlingstyp" (set to "Beslut"), and "Rubrik" (set to "E-post: hhhh"). Below these is a text field for "Innehåll" containing "hhh". There are two buttons: "Visa initierare" and "Spara handling". At the bottom of the window, there is a section for "Post 1 av 1" with fields for "Namn" (Aron Grahn), "Pers./org.nr", "Tel.nr", "E-post", "Adress", "Postnr.", "Postadress", and "Land" (with a "[Välj land]" dropdown). There are also buttons for "Fler namn", "Ta bort namn", "Sök", "<--", "-->", and "Lista".

Ovan visas den färdiga formen.

Jag har lagt till att man kan plocka bort och lägga till filer, vilket inte fanns med som krav från början.

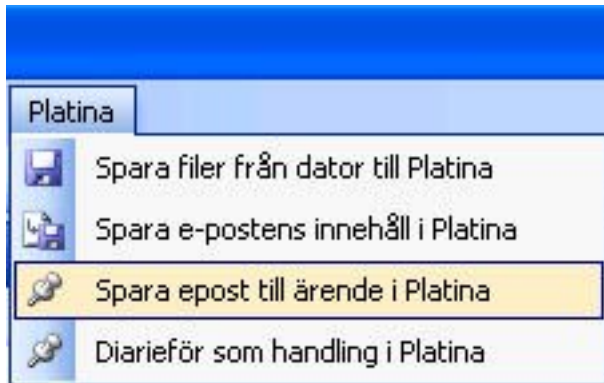
Om man fått viktig e-post som man vill spara till databasen och e-posten innehåller filer som man kanske inte vill ladda upp, så ska det naturligtvis finnas möjlighet att kunna plocka bort dessa filer.

Har även gjort fler forms som används för andra anrop mot databasen. Se exempel Bilaga B.

2.2.3 Menyer

Efter drygt sex veckors programmering var det dags att skapa en meny högst uppe bland standardmenyerna i Outlook. Det skulle även läggas till en Platina meny i det fönster som öppnas då man dubbelklickar på ett e-post meddelande.

För exempel på hur jag skapade upp menyn i "e-post fönstret" se Bilaga C.



Ovan visas den färdiga menyn i Outlook.

Menyn ovan skapas då man startar Outlook, och finns den redan vid start så tas den bort.

Samma princip gäller då man skapar menyn i e-post fönstret.

Problem som dök upp under den här kategorin var att man får inte skapa "Platina" menyn som finns i e-post fönstret förrän e-post-fönstret är öppet.

Detta löste jag genom att fånga det event som inträffar då man öppnar ett e-postfönster.

Det går inte att skapa menyn direkt man fått "nytt fönster event", eftersom den fortfarande då inte hunnit börja ladda, utan jag fick lägga till en timer som väntade 500 millisekunder från det eventet kom. Detta gjorde att jag var garanterad att fönstret fått tid att öppnas, så att jag kunde lägga till menyn. Se kod Bilaga D.

2.3 Utveckling och tester

2.3.1 Mapper

Jag har tidigare beskrivit min första uppgift som var att skapa mappar i mappfönstret. Mapparna i sin tur var länkade till aspx-sidor som jag fick sätta mig ner och skriva under vecka tre och fyra. Jag hade läst en kurs med HTML programmering på universitet så helt nybörjare på området var jag inte. Eftersom jag kände mig lite osäker valde jag att fräscha upp minnet på detta område, vilket gjordes genom att söka information på www.google.se.

På webbsidan hade jag först tänkt använda mig av treeview som är en komponent i .Net som visar en mappstruktur. Vid klick i mappstrukturen kunde den visa filer. Kompilering av detta projekt skapade en .dll fil som innehöll koden. Jag provade att skriva ett program och kompilerade det. När jag öppnade sidan i Internet Explorer visades kataloger precis som det var tänkt. Däremot när jag visade sidan i MS Outlook fungerade det inte alls. Efter att pratat med några andra programmerare kom vi fram till att det var Outlooks inbyggda säkerhet som var för hög. Outlook vägrar visa en webbsida som läser in en .dll fil. Jag fick helt enkelt hitta någon annan lösning på problemet. Lösningen blev att skapa en lista med JavaScript och active-x som gjorde det möjligt att vid dubbelklick på en fil plocka ut filen ur databasen. Detta fungerade utmärkt i Outlook.

Mappen diariet visste jag egentligen inte vad den skulle göra först. Jag hittade några vb-script på Internet som kunde integreras med Outlook. En enkel aspx-sida med VB-script skrevs. När man öppnade aspx-sidan (klickade på diarie-mappen) skapades ett e-post meddelande med adress, ärende och text som sedan skickades. Detta gjordes utan att man behövde mata in några värden. Vi hade tänkt utnyttja denna kod på det sätt att när man drog ett e-post meddelande till mappen så skulle detta meddelande skickas vidare till en annan e-postadress och därifrån skulle den laddas upp till databasen.

Denna lösning fungerade bra, men visade sig att det egentligen inte var detta man ville uppnå med diarie-mapp funktionen. Meningen var egentligen att man skulle dra e-post precis som vi tänkt från början till mappen, och då skulle man få upp ett formulär som man fyllde i och sparade till databasen. Detta blev mitt första formulär som jag började med vecka fem.

2.3.2 Forms

När ett e-post meddelande släpptes i diariet skulle det öppnas ett formulär som läste in olika värden från databasen. Användaren skulle sedan välja vilka värden som skulle användas innan man sparade e-posten till databasen.

Det var inte hållbart att varje gång man skulle testa formuläret så skulle man först starta upp Outlook, sedan dra ett e-post meddelande till mappen sedan vänta några sekunder innan man fick fram formuläret. Detta var en procedur som tog 15 sekunder.

Därför skrevs ett testprogram, som inte laddade någon av den gamla koden, utan direkt vid start av Outlook bara startade formuläret, nu sparade jag in 10 sekunder vid varje test av formulär.

Den grafiska biten av mitt första formulär gjordes på mindre än 10 minuter. Sedan började jag skriva kod till alla komponenter så att formuläret fick den funktionalitet som jag önskade.

Under detta steg stötte jag på ett pinsamt problem. Jag läste ut namn på diarium, postlistor och verksamhetskoder och detta fungerade. När användaren valt verksamhetskod så skulle dess ID användas till att hämta ut aktuella handlingskoder som i sin tur skulle skrivas ut i en egen lista på formuläret. Vid klick på en verksamhetskod gjordes ett korrekt anrop, jag fick värden, men dessa skrevs inte ut i formuläret. Jag började söka information på Internet om vad som kunde vara fel, men hittade absolut inget fel i min skrivna kod. Jag tyckte det skulle vara för pinsamt att fråga

någon på företaget hur man skriver ut text i en textlista, eftersom det verkligen är en lätt uppgift i vanliga fall. Jag provade lägga till extra uppdateringar av listan under körning, men det hjälpte inte heller. Då fick jag en idé!

Efter att ha lagt till värden i listan gjordes en metod som hämtade ut innehållet i listan och skrev ut det på skärmen i en ny MessageBox. Det visade sig att värden som jag lagt dit verkligen fanns där, vartenda ett, men de syntes bara inte. Jag fortsatte söka information på Internet vad som skulle kunna vara fel, jag tyckte fortfarande att det skulle vara pinsamt att fråga om något så enkelt. Några timmar med informationssökning på Internet och jag visste fortfarande inte vad som var fel, läget började bli desperat. Det blev dags för nästa desperata handling! Jag kopierade alla komponenter som fanns på formuläret, skapade en helt ny form där de gamla komponenterna fick plats. Koden från det gamla formuläret användes och kompilerades och resultatet blev förvånande. Jag hade inte ändrat någon kod överhuvudtaget, men nu fungerade formuläret felfritt, vad som var fel har jag inte kunnat lista ut.

2.3.3 Meny

Mitt första formulär hade jag fått att fungera felfritt under sjätte veckan av examensarbetet. Min nya utmaning var att göra menyer. Uppgift ett där var att diarieformuläret skulle kunna nås via meny, man ska inte behöva dra ett e-post meddelande till diariemappen för att få upp formuläret. Först skapade jag menyer med ikoner och text. Eftersom alla menyer utom en utgick ifrån att man markerat ett e-post meddelande fick jag skriva en funktion som kontrollerade om man markerat ett e-post meddelande. Om inget e-post meddelande var markerat så kunde inte funktionerna exekveras. Valde man diarieför så öppnades formuläret som jag sedan tidigare gjort, och information från e-post meddelandet lästes in till formuläret. Jag behövde inte skriva om någon större del kod för att få detta att fungera. Sedan var det dags att göra några nya formulär som hanterade de andra valen i meny. Nu var jag van programmeringen och det var bara en veckas jobb att få de övriga valen i meny att fungera. Nästa steg var att skapa en Platina meny då man öppnar ett nytt fönster som är av typen e-post meddelande. Man vill ju självklart inte stänga ner e-post meddelandet och sen klicka på inkorgen, markera e-posten och sedan klicka på diarieför för att kunna diarieföra. Även detta gick bra att programmera.

Sedan provade jag att lägga till en egen "toolbar" som är en grafisk meny med val. Knapparna på denna toolbar använde samma kod som valen i meny. Detta fungerade utmärkt, men jag ansåg att detta inte var en bra lösning. En toolbar kan flyttas runt i programmet och har därför ingen fast position. Snyggast och mest praktiskt enligt mig var en meny som alltid finns på samma plats. Att infoga toolbars är ofta irriterande för användaren, därför valde jag att inte ha någon toolbar.

Totalt nio veckor programmering gick åt för att min Addin skulle bli funktionsduglig. En vecka till att skriva om koden på ett bättre sätt är inte inräknat i de nio veckorna.

3 Resultat

Resultatet blev förutom några tusen rader kod även en fungerandes Addin.

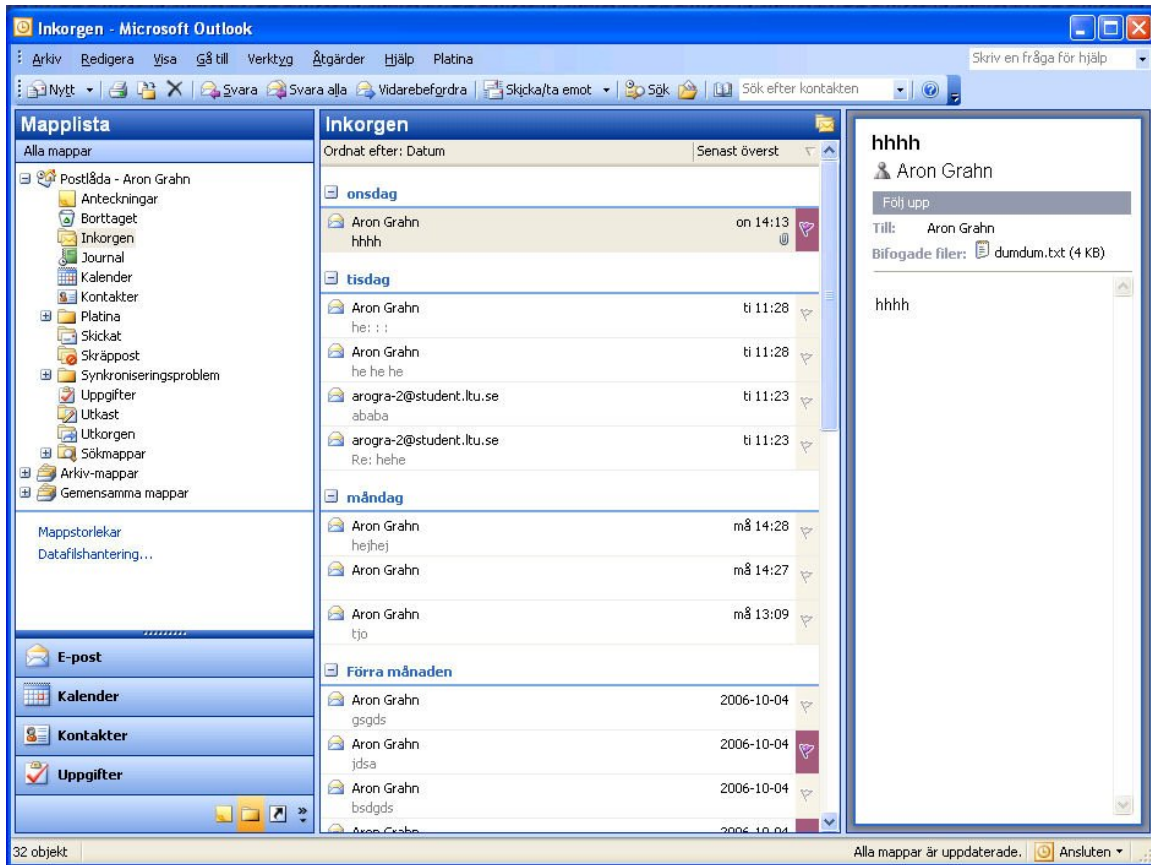


Bild på Outlook 2003 med min addin installerat.

Outlook har utökats med en meny "Platina" med funktioner.

Även mapplistan har blivit utökad med en mapp "Platina" som ger tillgång till alla filer i Platinas databas, samt tillgång till vissa metoder.

Snyggt och enkelt, en ovan användare ska inte ha några problem att kunna använda Outlooks nya funktioner.

För att visa om man diarefört ett e-post meddelande eller inte så markeras meddelandet med en lila flagga i Outlook om det är diarefört. Den nyaste versionen av min addin ändrar dessutom även titeln på e-posten har om den är diareförd.

Mapplista

Sök dokument - DocSearch

Vänlig sökning | Sök | Delsök | Rensa | Visa | Visa 100 | Dölj automatiskt | Oppna fönster | Sökning färdig, Antal poster: 86

Typ	Nummer	Dokumentrubrik	Skapad av	Skapad	Attribut
	113	Documentrubrik epost.txt	Administrator finnis	2006-11-01	
	114	dumdum.txt	Administrator finnis	2006-11-01	
	115	Epost.txt	Administrator finnis	2006-11-01	
	116	dumdum.txt	Administrator finnis	2006-11-01	
	117	Epost.txt	Administrator finnis	2006-11-01	
	118	Epost.txt	Administrator finnis	2006-11-01	
	119	Epost.txt	Administrator finnis	2006-11-01	
	121	Epost.txt	Administrator finnis	2006-11-01	
	122	ids.txt	Administrator finnis	2006-11-01	
	123	ids.txt	Administrator finnis	2006-11-01	
	124	Epost.txt	Administrator finnis	2006-11-01	
	127	Epost.txt	Administrator finnis	2006-11-01	
	130	Epost.txt	Administrator finnis	2006-11-01	
	131	ids.txt	Administrator finnis	2006-11-01	
	137	Epost.txt	Administrator finnis	2006-11-03	
	3	word[2]	Administrator	2006-09-18	
	53	C:\Added\IP.gif	Administrator finnis	2006-10-25	
	54	C:\Added\jabis.gif	Administrator finnis	2006-10-25	
	55	C:\Added\jabis-06_fra.gif	Administrator finnis	2006-10-25	
	58	C:\Added\Smart_Client.zip	Administrator finnis	2006-10-25	
	67	Smart_Client.zip	Administrator finnis	2006-10-30	
	68	Smart_Client.zip	Administrator finnis	2006-10-30	
	69	Smart_Client.zip	Administrator finnis	2006-10-30	
	40	C:\Added\Examensrapport_PyMTP_2006.pdf	Administrator finnis	2006-10-20	
	56	C:\Added\Labb.rar	Administrator finnis	2006-10-25	
	60	C:\Added\3DWNS_CU_	Administrator finnis	2006-10-25	
	61	C:\Added\3DWWE_CU_	Administrator finnis	2006-10-25	
	62	C:\Added\3DGARRO_CU_	Administrator finnis	2006-10-25	
	63	C:\Added\3DWWWE_CU_	Administrator finnis	2006-10-25	
	64	3DGN0_CU_	Administrator finnis	2006-10-25	
	65	ucFileList.dll	Administrator finnis	2006-10-25	
	80	dla	Administrator finnis	2006-10-31	
	125	dla.jpg	Administrator finnis	2006-11-01	
	87	Epost.txt	Administrator finnis	2006-11-01	

Dokument - Web Page Dialog

Vad vill du göra med dokumentet?

Bilden ovan visar en bild på webbsidan "Sök dokument". Vid dubbelklickning på fil i resultatlistan så kommer ett formulär med tre alternativ. "Visa", "Checka ut" eller "Egenskaper". Alla dessa tre val fungerar, det vill säga att du kan välja visa dokument om du vill läsa den, eller checka ut den om du vill göra ändringar i den eller helt enkelt bara ändra egenskaper på den. Väljer du checka ut filen så får du en fråga om du vill spara filen tillbaka till databasen efter det att du har läst eller ändrat i texten Vill man inte spara ändringar väljer man "ängra utcheckning".

4 Diskussion

Nu har Outlook blivit utökad med de metoder i Platina som man vill kunna nå direkt från Outlook. Det skulle inte vara några problem att lägga till fler val i menyerna och implementera nya metoder.

Mitt examensarbete slutar här, men utvecklingen av denna addin kommer att fortsätta. Addinen kommer med stor sannolikhet att behöva utökas i framtiden eftersom många kunder kommer att ha egna önskemål på funktionalitet i Outlook.

Förhoppningsvis är min programmering kompatibel med de nyare versionerna av MS Outlook. Hade jag haft mer tid hade jag programmerat snyggare forms. Då kan det vara klokt att fråga någon på företaget som arbetar med design om råd.

Vad som var synd var att inte folder-eventet fungerade. Pollning av diarie-mappen var lite av en speciallösning, men samtidigt tror jag att väldigt få eller ingen kommer använda den funktionen. Jag tror att nästan ingen kommer att använda sig av Platina-mappen med dess underkataloger. Det användaren däremot kommer att få användning av är den enkla Platina-menyn.

Annat som jag skulle vilja programmera om är uppdateringen av mappar, om den skulle kunna gå lösa på något annat sätt. För närvarande tar det nästan 3 sekunder att uppdatera hela strukturen. Men det är databasanropen som tar tid och inte självaste uppritningen av mappar.

Borttagning av mappar var något som Outlook inte kunde hantera på ett tillfredställande sätt. Vid borttagning av en mapp hamnade den direkt i papperskorgen. Därifrån var man tvungen att ta bort den igen för att den skulle försvinna. Den nyaste versionen av Outlook verkar ha möjlighet till "hard-delete" som innebär att mappen aldrig hamnar papperskorgen utan kan tas bort direkt. Det är väldigt störande då Outlook börjar tömma papperskorgen eftersom borttagningen av varje fil/mapp visas några millisekunder på skärmen. Om man hunnit klicka fem gånger på uppdatering av mappsystelet innebär det att det finns cirka 30 mappar i papperskorgen som ska tas bort samtidigt. Detta visar sig genom att ett formulär blinkar till i cirka en sekund. Bäst hade varit om det gått att ta bort mapparna utan att gå via papperskorgen, men i Outlook 2003 är detta omöjligt.

För övrigt är jag nöjd med programmet, det fungerar precis som det var tänkt.

5 Referenser

- *[1] Microsofts hemsida (2006-11-29)
<http://office.microsoft.com/en-us/outlook/HA010714981033.aspx>
- *[2] Microsofts hemsida (2006-11-29)
<http://office.microsoft.com/sv-se/outlook/HA010714981053.aspx>
- *[3] Microsofts hemsida (2006-11-29)
<http://msdn.microsoft.com/office/program/outlook/>
- *[4] EBI Systems hemsida (2006-11-29)
<http://www.ebi-systems.com/>

Bilagor

A. Uppdatera katalogstrukturen

```
private void folderEvent() //aktiveras då man bläddrar bland kataloger
{
    _Explorer = oAppobj.ActiveExplorer();
    Outlook.ExplorerEvents_Event explorerEvents =
(Outlook.ExplorerEvents_Event)_Explorer;
    explorerEvents.FolderSwitch += new
Outlook.ExplorerEvents_FolderSwitchEventHandler(checkEvent);
}
public void checkEvent() //undersöker vilken katalog man klickat på.
{
    string folderPath =
oAppobj.ActiveExplorer().CurrentFolder.FullFolderPath;
    if (folderPath.EndsWith("Platina"))
    {
        fillMyDocuments();
    }
    if (folderPath.EndsWith("Mina Dokument"))
    {
        fillMyDocuments();
    }
}
```

B. Bilder på Forms



The dialog box has a blue title bar with the text 'Spara filer till Platina' and standard window control buttons. The main content area is light gray and contains the following text:

Steg 1: Välj filer att lägga till i Platina:

File: dumdum.txt
Mail: hhhh

Steg 2: Tryck på Klar och välj i vilken mapp du vill lägga filerna:

At the bottom, there are four buttons: 'Lägg till', 'Ta bort', 'Avbryt', and 'Klar'.

Bilden ovan visar form för att spara e-post till en vanlig Platina-mapp.



The dialog box has a blue title bar with the text 'Spara epost till Ärende' and standard window control buttons. The main content area is light gray and contains the following elements:

Sök efter befintligt ärende

Diarium
Övningsdiarium

Fritext
[Empty text box]

Diariernr
[Empty text box] [Sök button]

Resultat:
mju - mju
hadghsa - hadghsa
hehe - hehe

At the bottom, there are three buttons: 'Avbryt', 'Ny sökning', and 'Ok'.

Bilden ovan visar form för att spara e-post till ärende.

C. Skapa meny I e-post fönstret.

```
private void AddMailMenuBar()
{
    try
    {
        menuBar =
this.oAppobj.ActiveInspector().CommandBars.ActiveMenuBar;
        newMenuBar =
(Office.CommandBarPopup)menuBar.Controls.Add(
            Office.MsoControlType.msoControlPopup, missing,
            missing, missing, false);
        if (newMenuBar != null)
        {
            newMenuBar.Caption = "Platina";
            newMenuBar.Tag = menuTag;
            buttonFour =
(Office.CommandBarButton)newMenuBar.Controls.
            Add(Office.MsoControlType.msoControlButton,
missing, missing, 1, true);
            buttonFour.Style =
Office.MsoButtonStyle.msoButtonIconAndCaption;
            buttonFour.Caption = "Diarieför";
            buttonFour.FaceId = 279;
            buttonFour.Click += new
Office._CommandBarButtonEvents_ClickEventHandler(buttonFour_Click);
            newMenuBar.Visible = true;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void RemoveMailMenubar()
{
    // If the menu already exists, remove it.
    try
    {
        Office.CommandBarPopup foundMenu =
(Office.CommandBarPopup)

this.oAppobj.ActiveInspector().CommandBars.ActiveMenuBar.
        FindControl(Office.MsoControlType.msoControlPopup,
            missing, menuTag, true, true);
        if (foundMenu != null)
        {
            foundMenu.Delete(true);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

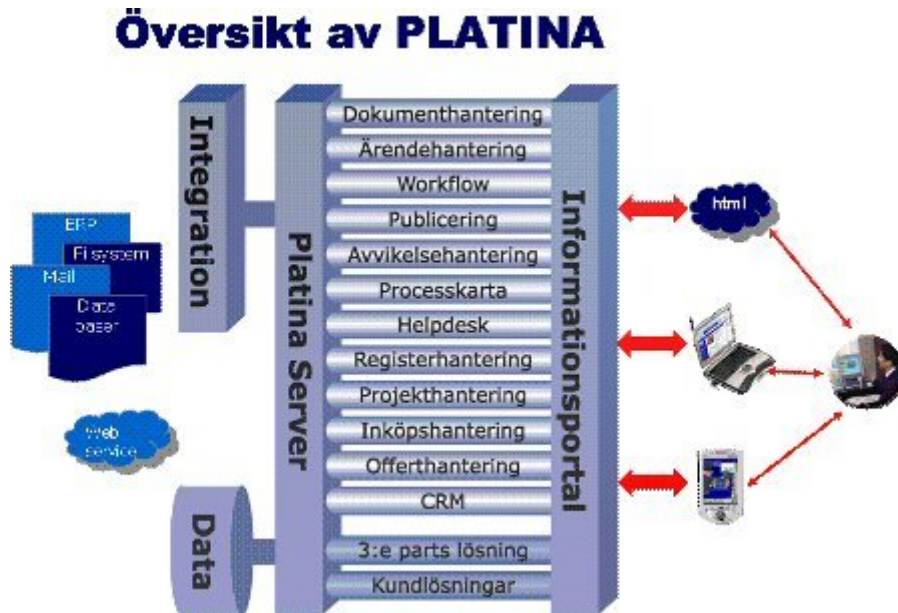
D. Fånga event, vänta 500 millisekunder.

```
//körs vid start av Outlook, skapar event då nytt fönster öppnad
public void windowMenu()
{
    myInspectors = (Outlook.InspectorsClass)oAppobj.Inspectors;
    myInspectors.NewInspector += new
Microsoft.Office.Interop.Outlook.InspectorsEvents_NewInspectorEventHand
ler(myInspectors_NewInspector);
}
//undersöker vilken typ av fönster som öppnades
private void
myInspectors_NewInspector(Microsoft.Office.Interop.Outlook.Inspector
Inspector)
{
    try
    {
        object Item = Inspector.CurrentItem;
        // Check the ItemType
        if (Item is Outlook.MailItem)
        {
            m_mail = (Outlook.MailItem)Item;
            if (m_mail.Sent == true)
            {
                waitsome();//det var rätt typ, men vänta lite.
            }
        }
    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
//väntar 500ms
private void waitsome()
{
    windowClock.Dispose();
    windowClock.Interval = 500;
    windowClock.Enabled = true;
    windowClock.Tick += new System.EventHandler(windowEvent);
    windowClock.Start();
}
//Nu har jag väntat 500ms, stanna klockan, ta bort befintlig menubar
och lägg dit den nya.
private void windowEvent(object source, EventArgs e)
{
    windowClock.Stop();
    RemoveSubMenubar();
    AddSubMenuBar();
}
```

E. Platina översikt

Nedan är en översikt av Platina plockad från EBI Systems hemsida. *[4]

De kraftfulla applikationerna i produktfamiljen PLATINA™ hjälper användare i sitt dagliga arbete. Genom att förenkla och snabba upp åtkomst av information som är nödvändig för att utföra viktiga arbetsuppgifter så förbättras produktivitet, effektivitet och arbetsinnehåll.



PLATINA™ består av en basplattform (PLATINA™ Server). Denna innehåller grundläggande funktioner för integration av data och information med bakomliggande system, en unik funktion för informationssynkronisering. De olika applikationerna görs sedan åtkomliga för användaren på ett mycket användarvänligt sätt via en helt webb-baserad informationsportal.

EBI Systems har med hjälp av kunder och partners utvecklat en rad innovativa och paketerade produkter för informationshantering, kvalitet- och processledning och kundvård. Klicka på respektive produkt i menyn ovan för mer information!

Förutom de färdiga produkterna är det lätt att med PLATINA™ Utvecklingsplattform använda välkända utvecklingsverktyg såsom t.ex. Visual Basic, C++ eller C# för att kundanpassa de olika applikationerna utifrån unika behov. Det är också möjligt för er själva eller våra utvalda partners att själva utveckla helt nya kundspecifika verksamhetssystem på ett snabbt och kostnadseffektivt sätt.

PLATINA™ är byggd på en modern arkitektur baserad på standarder såsom XML och Web Services. Systemet är från grunden utvecklat i och baserat på Microsofts nya koncept för affärslogik, .NET. Det säkerställer en skalbar, integrerad och lättimplementerad lösning med hög säkerhet för att hantera alla typer av data och information.

F. Platina information English

This is a page from EBI Systems homepage. *[4]

Information systems for enhanced efficiency!

EBI Systems develops and markets the packaged information and knowledge management software solution PLATINA™. The system helps you automate and streamline your business processes by providing relevant, up to date, and correct information to the users.

When using PLATINA™ all important information of the organisation like documents, cases, customer information, proposals, project information etc. is centrally stored in an information warehouse and presented in a modern web based user interface - integrated with other internal and external systems. By using PLATINA™ you can increase the effectiveness and speed up your business processes. The results are higher customer satisfaction, enhanced quality and improved profitability. PLATINA™ is a powerful and integrated family of applications for:

- Case Management
- Document Management
- Workflow
- Helpdesk
- Deviation Management
- Customer Relationship Management
- Project Management

PLATINA™ can also be used as a comprehensive .NET development platform to rapidly create company specific system solutions and extended information systems.

EBI Systems is an innovative and customer focused software company with leading experience and knowledge in delivering scalable solutions based on the Microsoft .NET framework, XML and Web Services.

