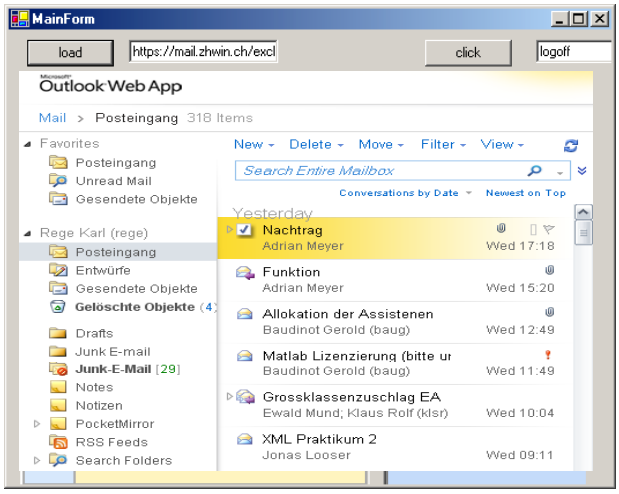
Zürcher Hochschule
für Angewandte Wissenschaften

Praktikum 8

Fortsetzung NextChange

Aufgabe 1 – Internet Explorer Einbindung und Fernsteuerung

Es gibt auch die Möglichkeit, den Kalender zentral auf einem Server zu führen. Dabei stellt sich jedoch das Problem des entfernten Zugriffs. Eine einfache Möglichkeit ist die Darstellung des Kalender in einem Browser Fenster wie es z.B. in Outlook Web Access realisiert wurde (`mail.zhaw.ch`). Diese Lösung hat jedoch den Nachteil, dass der Benutzer u.U. mehrere Anwendungen gleichzeitig bedienen und sich auch mehrmals anmelden muss. Es gibt jedoch die Möglichkeit in .NET den Browser als sog. ActiveX Element einzubinden und fernzusteuern (siehe Script). In Visual Studio steht bereits ein vorbereitetes WebBrowser Control zur Verfügung.



Nach dem Einbinden erfolgt der Aufruf einer Seite einfach mittels:

```
webBrowser1.Navigate (url);
```

Da jedoch der Outlook Passwort geschützt ist (man kann nur auf den eigenen Kalender zugreifen), wird man beim ersten Aufruf auf eine Login-Seite verwiesen, in der man das Passwort eingeben muss. Falls der OWA Teil einer laufenden Anwendung sein soll ist dies lästig. Um diese Passwort-Eingabe zu automatisieren, möchten wir das Login Formular deshalb automatisch ausfüllen (mit <StudentenName> und Passwort). Dafür

müssen wir einen DocumentComplete Handler folgendermassen anmelden (vor dem Navigate-Aufruf) .

```
webBrowser1.DocumentCompleted += new
WebBrowserDocumentCompletedEventHandler(webBrowser1_DocumentCompleted);
```

Die mitgegebene Methode wird nach erfolgtem Laden der Seite aufgerufen.

Sie muss folgende Signatur besitzen:

```
private void webBrowser1_DocumentCompleted(Object sender,
            WebBrowserDocumentCompletedEventArgs e)
```

In dieser Methode kann mittels:
`e.Url.ToString().StartsWith("https://mail.zhaw.ch/OWA/auth/logon.aspx"` überprüft werden ob wir uns auf der Login Seite befinden.

Wurde die Seite vollständig geladen können wir mittels:

```
HtmlDocument doc = webBrowser1.Document;
```

auf das HTML Dokument zugreifen.

Die Textfelder des Passwort-Formulars greifen wir folgendermassen zu:

```
HtmlElement e1 = doc.GetElementById("username");
e1.SetAttribute("value","<StudentenName>");
HtmlElement e2= doc.GetElementById("password");
e2.SetAttribute("value", "******");
```

Das Formular wird durch einfaches aktivieren des Submit-Knopfes abgeschickt:

```
foreach (HtmlElement elem in doc.GetElementsByTagName("input"))
{
if (elem.GetAttribute("value") == "Anmelden")
      {elem.InvokeMember("click");}
}
```

Aufgabe 2 – Low Level Programmierung

Für einen Kalender ist die exakte Uhrzeit wichtig. Leider sind die Uhren der meisten Computer ziemlich ungenau und sie sollte deshalb periodisch überprüft und allenfalls korrigiert werden. Es gibt im Internet für diesen Zweck sogenannte NIST Internet Time Server, die auf Anfrage die exakte Zeit liefern. Eine Zusammenstellung der verfügbaren Time Server findet man unter http://tf.nist.gov/tf-cgi/servers.cgi#

Leider kann in .NET nicht direkt die Uhr der eigenen Maschine gesetzt werden. Es gibt jedoch in der Kernel32.dll einen entsprechenden Aufruf (siehe Zusatzdokument oder: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/setsystemtime.asp)

Schreiben Sie eine Methode, welche die Uhr mit der NIST Zeit synchronisiert.

Hinweis: es muss folgendes `struct` mit den zu C++ analogen Datentypen als **ref** dem Aufruf übergeben werden.

```
public struct SystemTime                        // win32 struct
{
      public short wYear;
      public short wMonth;
      public short wDayOfWeek;
      public short wDay;
      public short wHour;
      public short wMinute;
      public short wSecond;
      public short wMilliseconds;
};
```

Hinweise

- Sie müssen den Aufruf unter Admin Rechten ausführen; Rechts-Klick und runas
- Oder mittels Anleitung http://www.codeproject.com/KB/cs/zetaimpersonator.aspx (Quelle finden Sie auf der .NET Seite).

Aufgabe 3 – COM Einbindung und Fernsteuerung

Sie möchten, dass Sie die Anwendung freundlich begrüsst. Binden Sie einen Agenten über die COM Schnittstelle ein.

Eventuell müssen Sie die COM Komponente zuerst noch Installieren.

Im Fall dass Sie mit dem Werkzeug die Agentenschnittstelle erzeugen müssen Sie Microsoft.Agent.Server und

Microsoft.Agent.Object einbinden. Die verfügbaren Agenten finden Sie im Verzeichnis `c:\windows\msagent\chars`. In Windows 7 wird der Agent Server nicht mehr standardmässig eingebunden. Er muss separat installiert werden (Link auf Homepage).

Falls Sie weitere Agenten suchen: http://www.msagentring.org

Hinweise

- Eine Wrapper Klasse erzeugen Sie sich indem Sie mittels Rechtsklick auf das Project → `Add Reference` → `COM` → `Microsoft Agent Server 2.0`
- Neue Agenten Files (z.B. Merlin.acs) können einfach `c:\winows\msagent\chars` Verzeichnis kopiert werden.