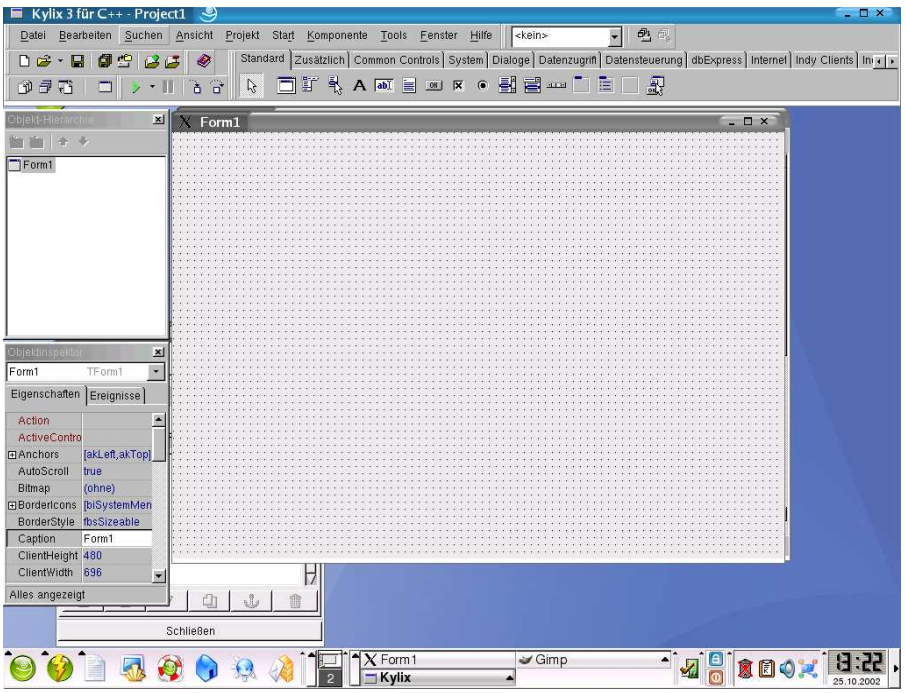
Borland®

IBM DevDays 2002

Rapid Application Development mit Kylix und DB2

Borland-Gastvortrag

Jochen Stärk <jstaerk@borland.com>



Die leere Kylix 3 C++-IDE

Borland®

Die IDE

Die Kylix-Entwicklungsumgebung (IDE) ist seit Version 3 nicht mehr auf Delphi beschränkt: Es existiert parallel eine Delphi- und eine C++-IDE. Beide IDEs sind in drei wichtige Bereiche aufgeteilt: Die Komponentenpalette, die Formularfenster und der Objektinspektor. In der Komponentenpalette wählen Sie visuelle Komponenten wie Buttons, Bilder, Eingabefeder u.ä., die Sie dann auf dem „Formular“ platzieren. Es gibt auch unsichtbare Komponenten wie Timer, die nur zur Entwurfszeit auf dem Formular erscheinen. Diese unsichtbaren Komponenten können Sie dann zum Beispiel im Quelltext ansprechen.

Die Formulare stellen die Fenster Ihrer späteren Applikation schon zur Entwurfszeit dar. Allerdings ist „Formular“ bei Kylix eher ein genereller Begriff für Fenster als eine Einschränkung auf Datenbanken.

Im Objektinspektor ändern Sie die Eigenschaften und Ereignisse von Komponenten. Sie wählen dafür die Komponente aus und ändern die Eigenschaften im Objektinspektor. Wenn Sie möchten, können Sie diese Eigenschaften auch im Quelltext ändern. Im Ereignisreiter können Sie Behandlungsroutinen für die Ereignisse, die das Objekt aufruft, anlegen.

Erstellen einer Web-Anwendung

Es soll eine Human-Resources-Anwendung zur internen Stellenausschreibung erstellt werden, die Datensätze in eine DB2-Datenbank aufnimmt und auf Wunsch per Browser oder Client-Programm wiedergibt.

```
Voraussetungen: Kylix 3 Ent, DB2 8.1
Datenbank „Borland“, Tabelle „Applications“, s.u.
create table applications (id integer not null,name
varchar(200),email varchar(100), resume blob (16000 K) not logged,
picture blob (16000 K) not logged, status integer)
```

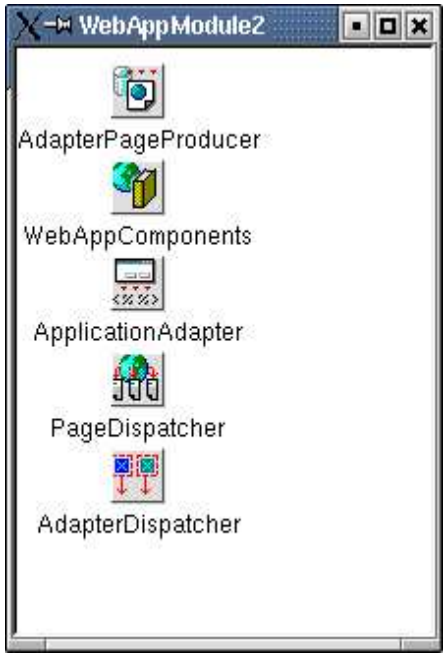
-Starten Sie DB2 (`db2start`)
-Starten Sie die Delphi-IDE (`startdelphi`)

-Erstellen Sie eine Neue Web-Snap-Anwendung mit Datei/Neu/Weitere/Web Snap/Web Snap Anwendung und wählen Sie Web-Debugger-Anwendung. Der Klassenname ist beliebig, zum Beispiel HumanResources.

-Wählen Sie in den Seitenoptionen als Typ AdapterPageProducer. Bestätigen Sie die Seitenoptionen und die Erstellungsoptionen.

Der Web App Debugger ist ein kleiner Web Server, der neben dem Browser mit HTTP-Daten zu beliefern, auch Debuginformationen protokolliert. Er erlaubt es auch, der Web-Anwendung ein sichtbares Fenster zuzuweisen. Dieses Fenster sieht man jetzt in Form1. Web-App-Debugger-Anwendungen können in der Regel innerhalb weniger als 2 Minuten in CGI-Anwendungen oder Apache-DSO-Module umgewandelt werden.

Die leere Web-Snap-Anwendung selbst besteht aus Komponenten wie dem AdapterPageProducer, die zur Laufzeit nicht sichtbar sind.



Die leere Web-Snap-Anwendung

-Fügen Sie aus dem dbExpress-Reiter eine SQLConnection ein. Schalten Sie dessen Connectionname auf DB2-Connection und das Loginprompt auf false. Passen Sie die Parameter (Params) entsprechend Ihren Gegebenheiten an, verwenden Sie bitte die Borland-Datenbank (siehe oben).

-Schalten Sie die Connected-Eigenschaft auf True.

-Fügen Sie aus dem dbExpress-Reiter ein SQLClientDataSet ein.

-Schalten Sie dessen DBConnection auf SQLConnection1

-Wählen Sie den CommandType ctTable

Kylix durchsucht nun die Datenbank nach allen Tabellen, um sie in die Auswahlbox CommandText einzufügen. Alternativ könnten Sie auch in CommandType ctQuery und in CommandText `SELECT * FROM APPLICATIONS` angeben.

-Wählen Sie aus CommandText die Tabelle Applications. Schalten Sie die Active-Eigenschaft probeweise auf true.

-Schalten Sie nun die Active-Eigenschaft wieder auf False und die Connected-Eigenschaft der Connection ebenfalls auf false.

Sie haben jetzt die Verbindung zu und Zugriff auf die Datenbank. Wir möchten nun mit einem Formular Datensätze einfügen können.

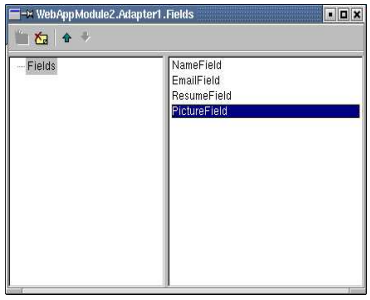
-Platzieren Sie einen Websnap/Adapter auf dem WebAppModule und doppelklicken Sie ihn.

Sie haben jetzt die Verbindung zu und Zugriff auf die Datenbank. Wir möchten nun mit einem Formular Datensätze einfügen können. Dafür verwenden wir einen Adapter, dem wir im Feldeditor durch Klick auf das „neuer Eintrag“-Icon neue Formularfelder zuweisen können.

-Legen Sie zwei AdapterFields, ein AdapterMemoField und ein AdapterFileField an. Änderen Sie deren Namen anschließend im Objektinspektor in „NameField“, „EmailField“, „ResumeField“ bzw. „PictureField“.

-Rechtsklicken Sie auf das Icon des Adapters im Web Module und starten Sie den Aktionseditor

-Die Aktionen die ich im Web Module ausführen kann, zum Beispiel ein Submit, lege ich im Aktionseditor an.



Der Feldeditor des Adapters

-Legen Sie einen Adapteraction an und benennen Sie sie zum Beispiel Submit. Wechseln Sie im Objektinspektor in den Ereignisse-Reiter und doppelklicken Sie in den Bereich neben onExecute.

Erklärung: Kylix legt jetzt eine Ereignisbehandlung an, die aufgerufen wird, sobald die Formulardaten empfangen wurden. Löschen Sie Begin und End und tragen Sie folgendes ein:

```
var
        tname, email, resume : String;
        Stream, inStream : TStream;
begin
SQLConnection1.Connected:=true;
SQLClientDataSet1.Active:=true;
Adapter1.UpdateRecords();
```

```
tname:=NameField.ActionValue.Values[0];
email:=EMailField.ActionValue.Values[0];
resume:=ResumeField.ActionValue.Values[0];
inStream:=PictureField.ActionValue.Files[0].Stream;
With SQLClientDataSet1 do
begin
  Insert;
  FieldByName('id').Value:=1;
  FieldByName('name').Value:=tname;
  FieldByName('email').Value:=email;

  Stream:=CreateBlobStream(SQLClientDataSet1.FieldByName('picture'),bmReadWrite
);
  try
    Stream.CopyFrom(inStream,0);
  finally
    Stream.Free;
    inStream.Free;
  end;
  inStream:=TStringStream.Create(ResumeField.ActionValue.Values[0]);
  Stream:=CreateBlobStream(SQLClientDataSet1.FieldByName('resume'),bmReadWrite)
;
  try
    Stream.CopyFrom(inStream,length(ResumeField.ActionValue.Values[0]));
  finally
    Stream.Free;
    inStream.Free;
  end;
  Post;
  ApplyUpdates(-1);
end;
SQLConnection1.Connected:=false;
SQLClientDataSet1.Active:=false;
end;
```

Erklärung: Dieser onSubmit-Handler aktiviert zunächst das SQLClientDataSet und die Connection, und holt sich mit UpdateRecords die aktuellen Werte der Input-Felder. Danach wird mit insert ein neuer Datensatz angelegt und dessen Felder mit FieldByName gefunden und belegt. Es werden Streams für die Blob-Felder Resume und Picture BlobStreams angelegt, das für Picture wird mit dem FileStram der hochgeladenen Datei, das von Resume mit dem StringStream des Bewerbungstextes belegt. Die Änderungen werden per Post in das SQLClientDataset und per ApplyUpdates in die Datenbank geschrieben.

Der Adapter ist jetzt korrekt eingestellt und muss noch angezeigt werden.

-Dafür doppelklicken Sie auf das Icon des AdapterPageProducers im WebModule.

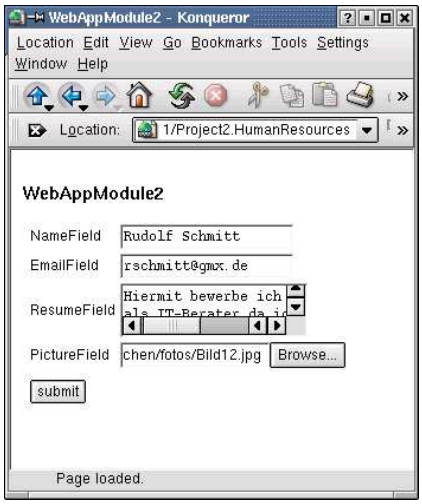
-Legen Sie ein AdapterForm an. Wählen Sie das AdapterForm aus und legen Sie darunter eine AdapterFieldGroup an.

-Ändern Sie die Adater-Eigenschaften der AdapterfieldGroup in Adapter1. Legen Sie unter AdapterForm eine neue AdapterCommandGroup und eine AdapterErrorList an.

-Ändern Sie Adapter in AdapterErrorList in Adapter1 und die DisplayComp von AdapterCommandGroup in AdapterFieldGroup1.

Erklärung: Die AdapterCommandGroup beinhaltet die Buttons des Formulars während die AdapterFieldGroup die Input-Felder anzeigt. AdapterErrorList zeigt im fehlerfall die Fehler an.

Das Hochladen und Einfügen neuer Einträge ist damit geschafft. Sie können das Projekt nun sichern und ausführen. Durch Klick auf Start wird die Anwendung beim Web App Debugger nebenbei registriert, sodass Sie, nachdem Sie den WepAppDebugger in Tool/Web-Anwendungs-Debugger aufgerufen und mit Klick auf Start gestartet haben, die angegebene Standard-URL aufrufen können und dort Ihren Projektnamen bereit zum Starten finden.



Der fertige erste Teil der WebAnwendung

Den Erfolg des Eintrags können Sie zum Beispiel in der db2-Kommandozeile mit

```
CONNECT TO BORLAND
SELECT * FROM APPLICATIONS
```

erfragen.

Nun könnte man auf die Idee kommen, die Bewerber auch per Web anzeigen lassen zu können – Für die Statistik, für die entsprechenden Abteilungsleiter und Betriebsräte oder

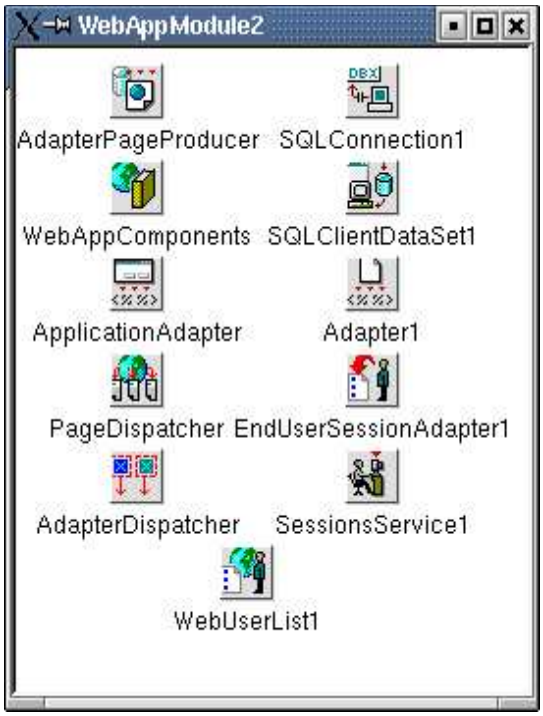
andere. Dazu müsste man Logins vergeben, da die Bewerber einen berechtigten Anspruch auf Vertraulichkeit Ihrer Daten haben. Dazu kann man eine WebUserList und, zur Dauerhaftigkeit des Logins, einen SessionService und einen EndUserSessionAdapter benutzen. (Fügen Sie diese Komponenten bitte ein).

-Ändern Sie die LoginPage des EndUserSessionAdapter in „login“ (diese Seite werden wir noch erstellen) und doppelklicken Sie auf WebUserList.

-Geben Sie einen Benutzer mit einem Passwort Ihrer Wahl ein (bspw. ibm/ibm).

Jetzt erstellt man eine Login-Seite mit Datei/Neu/Weitere/WebSnap/WebSnap Seitenmodul. Wählen Sie erneut als Typ AdapterPageProducer und benennen Sie die Seite „login“. Klicken Sie auf OK.

Die neue Seite ist jetzt erstellt und muss nur noch mit einem LoginFormAdapter bestückt werden. PasswordRequired können Sie getrost auf true setzen.



Die Webanwendung mit allen benötigten Komponenten

-Doppelklicken Sie dann auf denAdapterPage-Producer, neuer Eintrag-AdapterForm und unter dem AdapterForm neuer Eintrag-AdapterFieldGroup.

-Darunter erzeugen Sie bitte zwei AdapterDisplayFields. Diese benennen Sie im Objektinspektor zum Beispiel Benutzer und Passwort und setzen deren FieldNames dementsprechend auf UserName bzw. Password.

-Fügen Sie unter dem Adapterform noch eine AdapterErrorList und eine AdapterCommandGroup ein und setzen Sie deren Adapter- bzw. DisplayComponent-Eigenschaft auf LoginFormAdapter1 bzw. AdapterFieldGroup1.

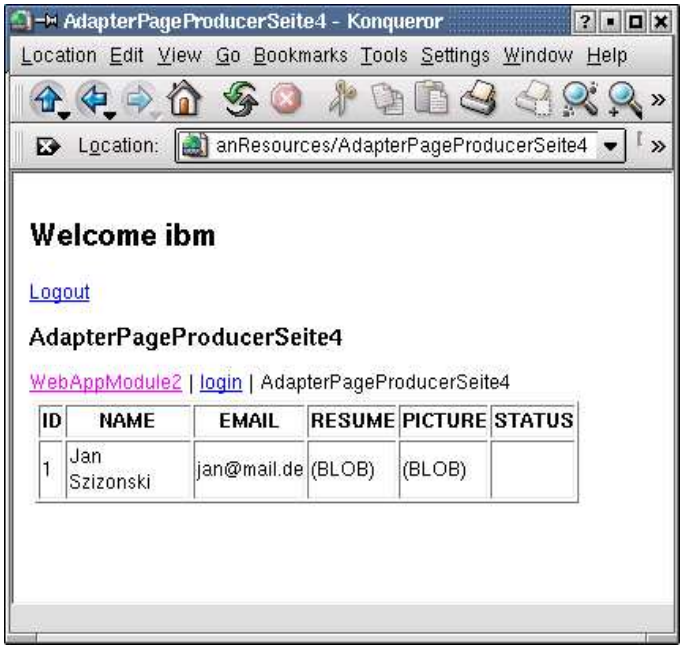
Diese Login-Seite nützt ohne zugriffsgeschützte Seite noch wenig. Erstellen Sie daher mit Datei/Neu/Weitere/Web-Snap/Web Snap-Seitenmodul eine weitere Seite des Typs AdapterPageProducer mit dem Namen Liste, klicken Sie diesmal jedoch die Anmeldung-

erfolderlich-Checkbox an.

-Aktivieren Sie das ursprüngliche Web-App-Modul und markieren Sie bei gedrückter Shift-Taste die SQLConnection und das SQLClientDataSet, und kopieren Sie diese mit STRG+C. Im neuen AdapterPageProducer können Sie sie dann mit STRG+V einfügen. Setzen Sie hier bitte die Eigenschaften Connected und Active auf True.

-Fügen Sie einen DataSetAdapter ein und setzen Sie die DataSet-Eigenschaft auf DataSet1. Doppelklicken Sie auf den AdapterpageProducer und fügen Sie ein AdapterForm und (hierachisch) darunter ein Adaptergrid hinzu. Wählen Sie als Adapter erneut DataSetAdapter1.

Ein AdapterGrid zeigt die Einträge z.B. eines DataSets tabellarisch an.



Die Bewerberliste nach einem erfolgreichen Login

Sie können nun alles Speichern und zuerst in der Entwicklungsumgebung starten um dann die entsprechende Web-App-Debugger-Seite per Browser anzusprechen. Klicken Sie auf „Liste“ kommen Sie jetzt, sofern nicht bereits eingeloggt, automatisch auf die Login-Seite, die ursprüngliche Bewerbungsseite ist immer noch jedem zugänglich.

Die Anwendung

Kylix ist aber sehr viel mehr als nur ein Tool um Webanwendungen zu erstellen, die stärkste Seite ist vielleicht das Erstellen von QT-Anwendungen.

Die folgende Anwendung wird in der C++-IDE programmiert:

-Kopieren Sie wieder mit STRG+C die SQLConnection und das SQLClientDataSet. Starten Sie die C++ IDE und fügen Sie sie mit STRG+V ein.

Die C++-IDE versteht allerdings auch Pascal – noch besser als C++Builder Delphi versteht. So enthält die C++-IDE alle Komponenten der Pascal-IDE, i.d.R. auch allle nachinstallierten Komponenten.

Wichtige Komponenten sind die Controls, die sichtbaren Komponenten, z.B. Button, Label, Image, MainMenu und viele mehr.

Klicken Sie auf das Formular und wählen Sie Ereignisse. Erzeugen Sie einen EventHandler für das OnClose-Ereignis mit dem Code

```
SQLConnection1->Close();
```

Wenn Sie kompilieren erhalten Sie übrigens native ELF-Dateien, die sie auf fast jedem x86-Linux-Rechner ausführen können, der die QT installiert hat. Sie können solche Programme auch problemlos unter Gnome laufen lassen – eine installierte QT vorausgesetzt.

Fügen Sie aus Zusätzlich ein Image ein und ziehen es auf der rechten oberen Ecke des Formulars auf. Doppelklicken Sie es um ein Bild, z.B. ein Logo zu laden.

Stellen Sie sicher, dass die Connection connected und das SQLClientDataSet aktiviert ist. Fügen Sie aus dem Tab Datenzugriff eine DataSource und aus Datensteuerung ein DBGrid ein. Setzen Sie die DataSet-Eigenschaft der DataSource auf SQLClientDataSet1.

Setzen Sie die DataSource-Eigenschaft des DBGrids auf SQLClientDataSet1.

Sie können schon zur Entwurfszeit sehen, wie das Grid später zur Laufzeit aussehen wird. In diesem Fall werden die Spalten zu breit angezeigt. Durch doppelklick auf das SQLClientDataSet und rechtsklick/alle Felder hinzufügen merkt sich Kylix, welche Felder es anzeigen soll. Durch markieren eines Eintrags und setzen des Width-Wertes kann bspw. Eine standardbreite von 120 Pixeln festgelegt werden.

-Fügen Sie aus Datensteuerung einen DBNavigator ein. Setzen Sie dessen DataSource-Eigenschaft auf DataSource1 und Flat auf true.

Wie Sie sich erinnern können, cached das SQLClientDataSet die geänderten Daten. Um die Änderungen zu übernehmen führen wir ein ApplyUpdates aus.

-Platzieren Sie einen Speedbutton aus dem zusätzlich-Menü auf Ihrem Formular und setzen Sie dessen Flat-Eigenschaft ebenfalls auf True. Ändern Sie die Caption in „Updates übernehmen“ und doppelklicken Sie ihn. Fügen Sie den Code `SQLClientDataSet1->ApplyUpdates(-1);` ein.

Sollten Sie aus welchen Gründen auch immer Ihre Datenbank auf Platte speichern wollen, genügt dafür seit Kylix 1 die Zeile:

```
SQLClientDataSet1->SaveToFile(“/tmp/applicants.xml“,dfXML);
```

Benutzen Sie einen weiteren Speedbutton und setzen Sie dessen Flat-Eigenschaft auf True. Ändern Sie die Caption in „Updates übernehmen“ und doppelklicken Sie ihn. Fügen Sie die obige Codezeile ein.

Man kann diese Anwendung leicht erweitern– zum Beispiel um den entsprechenden Bewerbern eMails zu schicken.

Um die Bewerber zu benachrichtigen, lassen wir uns direkten Zugriff auf die Spalten geben.

-Doppelklicken Sie auf das SQLClientDataSet und rechtsklicken Sie auf das Fenster zur Spaltenauswahl. Lassen Sie „alle Felder hinzufügen“.

-Wählen Sie aus dem Indy-Clients-Menü einen SMTP-Client aus und lassen ihn auf Ihr Formular fallen.

-Benutzen Sie einen letzten Speedbutton und setzen Sie dessen Flat-Eigenschaft auf True. Ändern Sie die Caption in „benachrichtigen“ und doppelklicken Sie ihn. Fügen Sie

```
IdSMTP1->QuickSend(NULL, "127.0.0.1","Ihre Bewerbung",SQLClientDataSet1EMAIL-
>AsString(),"de@borland.com","Wir würden uns freuen, Sie zu einem Gespräch
begrüßen zu dürfen.");
```

hinzu.

-Fügen Sie nun ein Datensteuerung/DBImage ein und verknüpfen es mit der DataSource. Wählen Sie als DataField „Picture“.

Die erstellte Anwendung ist, ganz nebenbei bemerkt, C++-Builder-Kompatibel, d.h., ein Windows-Client könnte man einfach dadurch erstellen, dass man diese Anwendung in C++Builder 6 Enterprise lädt und kompiliert.

Die kostenlose Kylix 30-Tage Version und eine sogenannte Open-Edition, mit der sie kostenlos Open-Source-Programme entwickeln dürfen, finden Sie unter www.borland.com/kylix.

Vielen Dank für Ihre Aufmerksamkeit

Die Anwendung – mit einem weiteren Image (Hintergrund),align=alClient und Form1 Contraints maxHeight=435