



Cebit 2003 Kylix/CBuilder-Script
<jstaerk@borland.com>

Jochen Stärk

Borland Kylix for C++/CBuilder Präsentation Cebit 2003

Inhaltsverzeichnis

Borland Kylix for C++/CBuilder Präsentation Cebit 2003.....	1
Verwendete Konfiguration.....	1
FTP-Videoplayer.....	2
Rundgang durch die Komponentenpalette.....	3
Ein CLX-Projekt.....	5
WebSnap.....	6
SMS -Web-Service.....	8
(optional) Web-Snap-Zugriffskontrolle.....	10
(optional) Umwandlung der AppDebugger-Anwendung in ein CGI.....	11
(optional) VCL-Konvertierung.....	13

Verwendete Konfiguration

Die Präsentation wird ausgeführt auf einem SuSE-Linux 8.1 Computer mit installiertem Vmware 3.2 und wahlweise Windows 2000 oder XP. Windows gibt Laufwerk C der virtuellen Platte zum lesen/schreiben frei, welches unter Linux als /windows/c/ gemountet ist. Linux gibt das Home-Verzeichnis des Benutzers ebenfalls zum lesen und schreiben frei, welches als Netzlaufwerk als e: verbunden ist. Das Linux-Home-Verzeichnis und das virtuelle C haben jeweils ein Unterverzeichnis „temp“, in dem die Dateien gespeichert werden sollen.

VMWare wird bei Beginn der Präsentation noch Vollbild ausgeführt.

Im CBuilder ist unbedingt ein neues, leeres CLX-Projekt zu zeigen.

Linux hat einen FTP-Server gestartet, der im root-Verzeichnis ein Video mit Namen



Cebit 2003 Kylix/CBuilder-Script

Jochen Stärk

<jstaerk@borland.com>

tarkan.mpg für Anonymous lesbar enthält.

Der Samba-Server von Linux läuft.

Der Linux-Interbase-Server läuft.

Die WSDL von <SMSServiceServer>/**test/sms/wsdngen.php** ist auf dem Windows-Laufwerk im temp-Verzeichnis

Windows setzt ein CBuilder 6 Update 2 ein. Sowohl CBuilder als auch Kylix haben eine Project Conversion Utility (http://www.borland.com/products/downloads/download_cbuilder.html unter dem Link „updates“) installiert.

FTP-Videoplayer

1. Fügen Sie einen Standard I Button, System I Media Player, Win32 I Progress Bar und Indy Clients I IdFTP ein.
2. Ändern Sie dessen Host auf Ihren Linux-FTP-Server und den Benutzernamen in „anonymous“. Achtung: Im Normalfall (Bridged Networking) entspricht die Linux-IP nicht Ihrer Windows-IP
3. Doppelklicken Sie auf den Button und geben Sie als Ereignisbehandlung ein:

```
IdFTP1->Connect();  
ProgressBar1->Max=IdFTP1->Size("tarkan.mpg");  
IdFTP1->Get("tarkan.mpg", "c:\\temp\\download.mpg", true);  
IdFTP1->Disconnect();  
MediaPlayer1->FileName="c:\\temp\\download.mpg";  
MediaPlayer1->Open();
```

<jstaerk@borland.com>

4. Geben Sie für das onWork-Ereignis des IdFTP1-Objekts folgenden Code vor:

```
ProgressBar1->Position=AWorkCount ;
```

1. Starten Sie das Programm und drücken Sie den Knopf, um die Datei herunter zu laden. Klicken Sie dann auf den Play-Knopf des MediaPlayers um das Video zu sehen.



Rundgang durch die Komponentenpalette

1. Gehen Sie zurück zu CBuilder für Windows
2. Ziehen Sie aus der Komponentenpalette eine Standard | ActionList auf Ihr Formular
3. Ziehen Sie ein Standard | Edit auf Ihr Formular
4. Ziehen Sie ein Standard | Main Menu auf Ihr Formular
5. Doppelklicken Sie die ActionList
6. Rechtsklicken Sie in den weißen Bereich unter Kategorien und wählen Sie Neue Standardaktion
7. Wählen Sie mindestens TEditCut und TEditPaste
8. Schließen Sie das Standardaktionen-Hinzufügen und das ActionList-Fenster
9. Doppelklicken Sie auf MainMenu
10. Geben Sie „Datei“ ein und bestätigen Sie mit Enter

<jstaerk@borland.com>

11. Klicken Sie auf den gepunkteten Bereich unter Datei und wählen Sie im Objektinspektor Action/EditCut
12. Klicken Sie auf den gepunkteten Bereich unter Ausschneiden und wählen Sie im Objektinspektor Action/EditPaste
13. Klicken Sie auf den gepunkteten Bereich unter Einfügen und geben im Objektinspektor als Caption „Schließen“ an.
14. Doppelklicken Sie auf Schließen und geben Sie „Close();“ ein.
15. Fügen Sie ein Zusätzlich | Image ein.
16. Setzen Sie ggf. Eigenschaften | Stretch auf true.
17. Doppelklicken Sie auf das Bild und laden ein Beliebige Beispielbild.
18. Fügen Sie einen Dialoge | Opendialog ein.
19. Klicken Sie an einer leeren Stelle im Formular.
20. Fügen Sie durch Doppelklick an der entsprechenden Stelle im Objektinspektor (Rechts neben Ereignisse | OnClick) folgenden Event-Handler ein:

```
if (OpenDialog1->Execute())
{
    Image1->Picture->LoadFromFile(AnsiString(OpenDialog1->FileName));
    Form1->Edit1->Text=OpenDialog1->FileName;
}
```
21. Die Anwendung sollte zur Entwicklungs- bzw. Laufzeit jetzt in etwa so aussehen:



Wenn Sie jetzt auf das Formular klicken, können Sie sich eine Datei aussuchen, durch die die Bilddatei ersetzt wird. Außerdem wird der entsprechende Dateiname im Edit-Feld angezeigt. Das Menü wird bei einer Textauswahl im Edit-Feld Ausschneiden aktivieren und bei Text in der Zwischenablage einfügen. Beenden können Sie das Programm zum Beispiel durch Datei | Schließen.

22. Speichern Sie das Projekt in temp/imageViewer.

Ein CLX-Projekt

1. Erstellen Sie ein neues CLX-Projekt (Datei/Neu/CLX-Anwendung)
2. Werfen Sie ein Button auf das Formular des CBuilders, als Ereignisbehandlung (Doppelklick auf den Button) geben Sie beispielsweise `ShowMessage("Hallo Welt");` ein.
3. Speichern Sie das Projekt unter `e:\temp\buttonprojw.bpr`. Danach
4. wechseln Sie nun per `STRG+ALT` in den Fenstermodus von `vmWare` und
5. starten Sie in `Kylix | Tools` die „Project Conversion Utility“.
6. Wählen Sie als Eingabedatei `temp/buttonprojw.bpr` und als Ausgabedatei `temp/buttonprojk.bpr`.



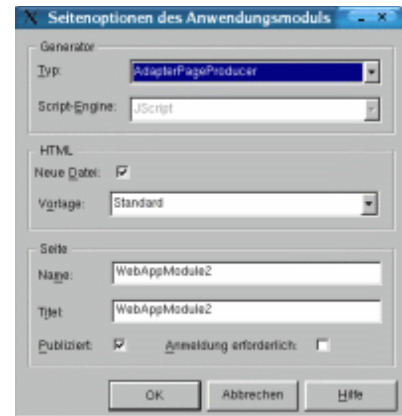
Die Linux-Version der Project Conversion Utility wandelt CBuilder-Projektdateien in Kylix-for-C++-Projektdateien um, die Windows-Version Kylix-for-C++-Projektdateien in Kylix-Projektdateien. Sie wandelt keine Quelldateien, etwa von VCL in CLX o.ä. um. Bei der Umwandlung kann man ohne Aufwand durchaus auch einfach die Quelldateien übernehmen und die Projektdatei in Kylix neu anlegen, das PCU ist also kein „muss“. Die Tatsache, dass Delphi-CLX-Projektdateien hundertprozentig mit den Kylix-for-Delphi-Projektdateien übereinstimmen könnte ein Hinweis sein, dass eine solche Übereinstimmung auch im C++-Bereich erreicht werden kann.



7. Öffnen Sie das konvertierte Projekt
8. Starten Sie (F9)

WebSnap

1. In Kylix wählen Sie bitte Datei | Neu | Weitere | Web Snap | Web Snap Anwendung
2. Wählen Sie eine Web Debugger Anwendung mit dem Klassennamen Beispiel
3. Wählen Sie in den Seitenoptionen als Typ Adapterpageproducer



4. Fügen Sie Ihrem Formular zwei Labels hinzu.
5. Setzen Sie die Caption des ersten Labels (Label1) auf Web Snap Server und leeren Sie die Caption des zweiten Labels, vergrößern aber dessen Anzeigebereich
6. Fügen Sie Ihrem Seitenmodul eine dbExpress/SQLConnection hinzu
7. Ändern Sie deren Connectionname auf Interbase, LoginPrompt auf False, und in den Params die Database auf /opt/interbase/examples/employee.gdb (Mysql (Voraussetzung: Installiertes Mysql-Devel-Packet):ConnectionName MySqlConnection, Params/Hostname: localhost, Params/Database: test, Params/User_Name: root, Params/Password=bspw. masterkey)
8. Setzen Sie connected auf True
9. Fügen Sie ein DBExpress/SQLClientDataSet ein
10. Setzen Sie dessen DBConnection auf die eben erstellte DbConnection1
11. Setzen Sie CommandType auf ctTable und CommandText auf eine beliebige Tabelle (Achtung: Mysqls-Test-Datenbank ist per Default leer)
12. Setzen Sie active auf True
13. Fügen Sie einen Websnap/DatasetAdapter hinzu
14. Ändern Sie die Dataset-Eigenschaft des DatasetAdapter auf SQLClientDataSet1.

<jstaerk@borland.com>

15. Doppelklicken Sie auf den Adapterpageproducer

16. Klicken Sie auf Neuer Eintrag, Adapterform

17. Lassen Sie Adapterform markiert und klicken sie auf Neuer Eintrag

18. Wählen Sie ein Adaptergrid

19. Setzen Sie im Objektinspektor dessen Adapter auf DatasetAdapter1

20. Setzen Sie die den onBeforeGetContent-Handler des Adapters z.B. auf

```
for (int i=0; i<10; i++)  
{  
    Form1->Label2->Caption=Form1->Label2->Caption+"Test";  
}
```

Setzen Sie einen Breakpoint auf die mittlere Zeile.

21. Speichern Sie Ihr Projekt z.B. in ~/temp/Dienstag/websnap/

22. Compilieren Sie und starten Sie es in Kylix (F9) Starten Sie Tools/Web Anwendungen
Debugger

23. Starten Sie ggf. den Server

24. Klicken Sie auf die Standard-URL (möglicherweise möchten Sie vorher in
Server/Optionen als Browser konqueror eingeben)

Sie sollten jetzt an Ihren Breakpoint gelangen. Durch STRG+Klick auf die Variable i können Sie sehen, in welcher Iteration sie sich befinden. Durch Drücken von F9 sollten Sie zur nächsten Iteration gelangen.

SMS -Web-Service

1. Erstellen Sie ein neues CLX-Projekt in CBuilder. (Es ist belanglos, ob es ein CLX- oder VCL-Projekt im CBuilder oder in Kylix for C++ ist)

<jstaerk@borland.com>

2. Platzieren Sie eine Web Services/HTTPRIO-Komponente,
3. eine Edit-Box mit Namen nr
4. eine Edit-Box mit Namen text
5. und ein Button auf Ihrem Formular
6. Wählen Sie als bitte eine lokale Datei. Sie können zu deren Erzeugung <SMSServiceServer>/test/sms/wsdngen.php verwenden. Kopieren Sie die lokale Adresse in die Zwischenablage



Die WSDL sagt CBuilder, welche Funktionen der Service zur Verfügung steht. Dieser spezielle Service steht nur zur Cebit zur Verfügung – normalerweise gibt es ihn nur als puren HTTP-Gateway und kostet Geld (als Dienstleister nutzen wir 3mobile, www.3mobile.de).

7. Wählen Sie als Service lserviceService und als Port lservicePort
8. Wählen Sie Datei/Neu/Weitere/Web Services/WSDL Importer
9. Fügen Sie die WSDL-Adresse aus der Zwischenablage ein. Klicken Sie auf Weiter und Fertigstellen.



Es wird jetzt ein Interface generiert, wie die Programmiersprache den Service ansprechen soll.

10. Doppelklicken Sie auf den Button um ein Ereignishandler zu generieren.
11. Lassen Sie den sms-Unit-Header einschließen (z.B. durch Datei/Unit Header einschließen).

<jstaerk@borland.com>

12. Verwenden Sie die Zeilen

```
_di_Iservice sendService;  
HTTPRIO1->QueryInterface(sendService);  
sendService->sendSMS(nr->Text, text->Text);
```

Um SMSe mit dem Service zu verschicken.

13. Bitte verwenden Sie keine Umlaute.

14. Verwenden Sie als Empfängernummer bitte ausschließlich internationale Nummern, wie „49177123456“

(optional) Web-Snap-Zugriffskontrolle

1. Laden Sie das Web-Snap-Projekt, deaktivieren Sie ggf. den Break-Point
2. Kopieren Sie SQLConnection1, SQLClientDataSet1 und DataSetAdapter1 in die Zwischenablage
3. Klicken Sie Datei/Neu/Weitere/WebSnap/WebSnap Seitenmodul
4. Type: Adapterpageproducer
5. Wählen Sie diesmal: Anmeldung erforderlich
6. Fügen Sie die SQLConnection1, das SQLClientDataSet1 und den DataSetAdapter1 ein.
7. Wählen Sie im CommandText des SQLClientDataSets eine andere Tabelle und aktivieren Sie das SQLClientDataSet wieder
8. Doppelklicken Sie auf Adapterpageproducer
9. Neuer Eintrag/AdapterForm
10. Darunter: Neuer Eintrag/AdapterGrid
11. Wählen Sie als Adapter wieder DataSetadapter
12. Fügen Sie im WebAppmodul eine Websnap/WebUserList hinzu.
13. Fügen Sie eine SessionService und einen
14. EndUserSessionAdapter ein.

<jstaerk@borland.com>

15. Ändern Sie die LoginPage-Eigenschaft des EndUserSessionAdapters auf „login“

(Diese Seite werden wir noch erstellen)

16. Doppelklicken Sie die WebUserList und wählen Sie Neue hinzufügen

17. Geben Sie im Objektinspektor als Username und Passwort bspw. jeweils test ein.

18. Wählen Sie Datei/Neu/Weitere/Web Snap/Web Snap Seitenmodul

19. Wählen Sie als Namen „login“

20. Wählen Sie als Typ wiederum AdapterpageProducer

21. WebSnap/Loginformadapter

22. Doppelklicken Sie auf den AdapterPageProducer

23. Neu/AdapterForm

24. Darunter: Neu/AdapterErrorlist

25. Fügen Sie zum Anzeigen der Eingabefelder des Logins ebenfalls unterhalb des Adapterforms ein: Neu/AdapterFieldGroup

26. Adapter-Eigenschaft der Adapterfieldgroup im Objektinspektor:

LoginFormAdapter1

27. Ebenfalls unterhalb des Adapterforms zum Anzeigen des Submit-Buttons:

Neu/AdapterCommandGroup

28. Wählen Sie im Objektinspektor für deren DisplayComponent

AdapterFieldGroup1

29. Klicken Sie im WebAnwendungsDebugger wieder auf DefaultURL, und wählen Sie Ihr Projekt

(optional) Umwandlung der AppDebugger-Anwendung in ein CGI

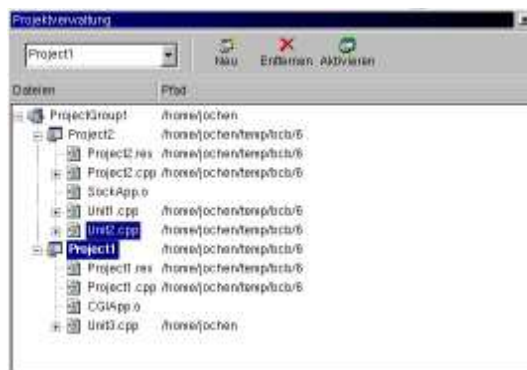
(Voraussetzung: Schreibzugriff auf das CGI-bin-Verzeichnis des Webserver.

Ein Link auf /usr/local/kylix3/bin/libsqlib.so, /usr/local/kylix3/bin/libmidas.so.1 und /usr/local/kylix3/bin/libjs.borland.so in /usr/lib)

(optional) Umwandlung der AppDebugger-Anwendung in ein CGI -

<jstaerk@borland.com>

1. Öffnen Sie das WebSnap-Projekt
2. Gehen Sie in Ansicht | Projektverwaltung
3. Klicken Sie auf Neu | Web Snap | Web Snap Anwendung
4. Wählen Sie als Servertyp CGI und in den Seitenoptionen als Typ wieder Adapterpageproducer
5. Bringen Sie die Projektverwaltung in den Vordergrund
6. Klappen Sie die Web-App-Debugger-Anwendung und die CGI-Anwendung „auf“
7. Ziehen Sie Unit2 mit gedrückter Maustaste vom Web-App ins CGI-Projekt (Bestätigen Sie, dass Sie dem Projekt hinzufügen möchten).



8. Löschen Sie Unit3 aus dem Projekt.
9. Speichern Sie das Projekt und die Projektgruppe
10. Entfernen Sie ggf. alle Referenzen zum Formular Unit1 aus Unit2.
11. Kompilieren Sie das Projekt.
12. Kopieren Sie die erstellte Binärdatei (z.B. Project2) in das Apache-CGI-Bin-Verzeichnis.
13. Kopieren Sie ebenfalls die HTML-Dateien der Units (*.html).
14. Rufen Sie Ihr CGI auf, z.B. „http://localhost/cgi-bin/Project2“.

(optional) Umwandlung der AppDebugger-Anwendung in ein CGI -

(optional) VCL-Konvertierung

1. Starten Sie ein neues Projekt im CBuilder mit Datei | Neu | Anwendung,
2. werfen wieder einen Button auf das Formular,
3. schreiben Sie wieder einen Hallo-Welt-Ereignishandler und
4. Speichern Sie das Projekt unter e:\temp\buttonprojw2.bpr.
5. Wechseln Sie mit STRG+ENTER nach Kylix.
6. starten Sie in Kylix | Tools die „Project Conversion Utility“.
7. Wählen Sie als Eingabedatei „temp/buttonprojw2.bpr“ und als Ausgabedatei „temp/buttonprojk2.bpr“.
8. Ändern Sie im konqueror durch F2 den Dateinamen von Unit2.dfm in Unit2.xfm
9. Öffnen Sie Unit1.cpp in einem Editor (Rechtsklick | Öffnen mit | Kate) und ändern Sie die Zeile `#pragma resource *.dfm` in `#pragma resource *.xfm` in Unit1.cpp.
10. Ändern Sie ebenfalls `#include <vcl.h>` in `#include <clx.h>` und speichern Sie diese Änderungen.
11. Öffnen Sie buttonprojk2.bpr in Kylix.
12. Ändern Sie bitte vcl.h auch im Projektquelltext (Projekt | Quelltext anzeigen) in clx.h
13. Zeigen Sie Unit1.h und kompilieren Sie. Kylix fügt hier beim Speichern die benötigten CLX-Units ein.
14. Löschen Sie in Unit1.h `#include <Controls.hpp>`, `#include <StdCtrls.hpp>`, `#include <ExtCtrls.hpp>`, und `#include <Graphics.hpp>` und
15. Ändern Sie `#include <Forms.hpp>` in `#include <QForms.hpp>`
16. Kompilieren Sie



Beachten Sie, dass Sie sich viel Mühe sparen, wenn Sie von Anfang an durch Verwendung von CLX denselben Source-Code für Windows und Linux verwenden können.



Borland®

Cebit 2003 Kylix/CBuilder-Script

Jochen Stärk

<jstaerk@borland.com>

