

combit List & Label

Beispiel für COBOL

Vorbemerkung

Die nachfolgenden Beispiele stellen keine fertig lauffähigen Quelltexte dar, es fehlen beispielsweise die Routinen zum Öffnen und Schließen eines List & Label Jobs, jegliche Fehlerbehandlung u.Ä. Sie dienen lediglich als Vorlage, um einen Eindruck davon zu bekommen, wie man sich die Ansteuerung von List & Label unter COBOL vorstellen kann.

Die verwendeten List & Label Konstanten müssen aus dem C/C++ Headerfile "cmbtLL??" (?? steht für die List & Label Version, also bspw. 14, 15, ...) auf COBOL Syntax umgesetzt werden. Das Headerfile befindet sich im Visual C++ Unterverzeichnis und kann mit jedem Texteditor geöffnet werden.

Die darin enthaltenen Konstantendefinitionen der Art

```
#define LL_PROJECT_LABEL (1)
```

müssen auf COBOL-Syntax umgesetzt werden, also bspw. zu

```
78 LL-PROJECT-LABEL value 1.
```

Möglicherweise gibt es in der COBOL-Entwicklungsumgebung bereits ein Tool, das automatisch eine derartige Umsetzung vornehmen kann (Stichwort "H2CPY").

Aufruf des Designers

```
*****
*   Dieses COBOL Beispiel startet den Designer von List & Label. Es
*   werden 2 Felder angemeldet: 'Artikelnummer' und 'Bezeichnung'
*
*   Copyright (c) combit GmbH
*****
[...]
```

```
MOVE "LLDEMO.LBL" & X"00" TO PROJEKTNAME
MOVE LL-PROJECT-LABEL TO PROJEKTTYP
MOVE "Etiketten Designer" & X"00" TO FENSTERTITEL
```

```
* Melde alle verfügbaren Variablen an List & Label an
* (hier bspw. Artikelnummer + Bezeichnung):
* übergebe zum Design lediglich Beispielinhalte
```

```
MOVE "Artikelnummer" TO VARIABLE-NAME
MOVE "Beispielartikelnummer" TO VARIABLE-INHALT
CALL WINAPI "LIDefineVariableA"
    USING BY VALUE JOB-HANDLE
        BY REFERENCE VARIABLE-NAME
        BY REFERENCE VARIABLE-INHALT
    RETURNING RC-INT
END-CALL
MOVE "Bezeichnung" TO VARIABLE-NAME
MOVE "Beispielbezeichnung" TO VARIABLE-INHALT
CALL WINAPI "LIDefineVariableA"
    USING BY VALUE JOB-HANDLE
        BY REFERENCE VARIABLE-NAME
        BY REFERENCE VARIABLE-INHALT
    RETURNING RC-INT
END-CALL
```

```
* Starte Designer
```

```
CALL WINAPI "LIDefineLayoutA"
    USING BY VALUE JOB-HANDLE
        BY VALUE WINDOW-HANDLE
        BY REFERENCE FENSTERTITEL
        BY VALUE PROJEKTTYP
        BY REFERENCE PROJEKTNAME
    RETURNING RC-INT
END-CALL
```

```
[...]
```

Druck eines Etiketts

```
*****
*   Dieses COBOL Beispiel druckt 1 Etikett mit List & Label. Die Daten-
*   Felder werden in diesem Fall aus einem Array INHALT geholt
*   (Normalerweise würde hier jeweils ein Datensatz-Zugriff erfolgen).
*
*   Copyright (c) combit GmbH
*****
[...]
```

* *Starte Preview-Druck eines Etiketts "LLDEMO.LBL":*

```
MOVE "LLDEMO.LBL" & X"00" TO PROJEKTNAME
MOVE LL-PRINT-PREVIEW TO DRUCKOPTIONEN
MOVE LL-PROJECT-LABEL TO PROJEKTTYP
MOVE LL-BOXTYPE-BRIDGEMETER TO BOXTYP
MOVE "Etikettendruck" & X"00" TO FENSTERTITEL
```

```
CALL WINAPI "LIPrintWithBoxStartA"
  USING BY VALUE JOB-HANDLE
        BY VALUE PROJEKTTYP
        BY REFERENCE PROJEKTNAME
        BY VALUE DRUCKOPTIONEN
        BY VALUE BOXTYP
        BY VALUE WINDOW-HANDLE
        BY REFERENCE FENSTERTITEL
  RETURNING RC-INT
END-CALL
```

* *Melde alle Variablen des zu druckenden Datensatzes an (hier bspw. Artikelnummer + Bezeichnung):*
* *hole dabei Feldinhalt aus Datenquelle (hier zum Test bspw. das Array INHALT)*

```
MOVE "Artikelnummer" TO VARIABLE-NAME
MOVE INHALT(1) TO VARIABLE-INHALT
CALL WINAPI "LIDefineVariableA"
  USING BY VALUE JOB-HANDLE
        BY REFERENCE VARIABLE-NAME
        BY REFERENCE VARIABLE-INHALT
  RETURNING RC-INT
END-CALL
MOVE "Bezeichnung" TO VARIABLE-NAME
MOVE INHALT(2) TO VARIABLE-INHALT
CALL WINAPI "LIDefineVariableA"
  USING BY VALUE JOB-HANDLE
        BY REFERENCE VARIABLE-NAME
        BY REFERENCE VARIABLE-INHALT
  RETURNING RC-INT
END-CALL
```

* *Drucke Etikett*

```
CALL WINAPI "LIPrint"
  USING BY VALUE JOB-HANDLE
  RETURNING RC-INT
END-CALL
```

** Beende Ausdruck*

```
MOVE 0 TO PAGES  
CALL WINAPI "LIPrintEnd"  
    USING BY VALUE JOB-HANDLE  
        BY VALUE PAGES  
    RETURNING RC-INT  
END-CALL
```

** Zeige Preview*

```
MOVE "." & X"00" TO PFAD.  
CALL WINAPI "LIPreviewDisplayA"  
    USING BY VALUE JOB-HANDLE  
        BY REFERENCE PROJEKTNAME  
        BY REFERENCE PFAD  
    BY VALUE WINDOW-HANDLE  
    RETURNING RC-INT  
END-CALL
```

[...]