

Portierung von Unique 4GL-Anwendungen der Versionen 3.12 und 4.02 zur Version 5.12

Die Portierung von Unique 4GL-Anwendungen zur Version 5.12 umfasst die Bereiche Anwendungslayout, Datenbankzugriffe, allgemeine 4GL-Funktionalität und Unique Konfiguration. Dieses Dokument soll die Dokumentation von Unique CONCEPT nicht ersetzen, sondern kann nur mit dieser zusammen eingesetzt werden. Querverweise zum Reference manual und Operations guide zeigen auf die vollständigen Erklärungen zu den angesprochenen Themen. Der Abschnitt New and changed functionality in den Release notes sollte in jedem Fall vorher gelesen werden. Lesen Sie bitte dieses Dokument von vorne an durch, da dieses Dokument als Schritt für Schritt-Anleitung konzipiert ist.

Anwendungslayout

Portieren ohne Konvertierung zur Element-Syntax

Vorteile

- Ein Quelltext für die Versionen 3.12, 4.02 und 5.12
- Keine (oder sehr geringe) Änderungen am Quelltext notwendig
- Geringer Aufwand für die Portierung

Nachteile

- Forms Painter kann nicht benutzt werden, da der Forms Painter nur auf der Element-Syntax arbeitet und alte Syntax automatisch konvertiert.
- Neue GUI-Controls und 4GL-Funktionen können nicht oder nur begrenzt benutzt werden, z.B. Tab-Forms, LDT-System (Sprachunabhängigkeit), Elementtypen Line, Scrollbars,...
- Anwendungslayout entspricht dem Stil der 3.12 oder 4.02 Umgebung und nicht den Konventionen von Windows 95/NT 4.0

Maßnahmen

1. Einrichtung eines Unique-Systems mit anderem object-path und application-path als für Version 3.12 oder 4.02. Für Version 5.12 sind in jedem Fall neue Kompilate notwendig.
2. Änderung der Unique System Database (\$UNIQUE_HOME/setup/uqsystem.ini). Siehe Operations Guide: Unique System Database.
Im Abschnitt [SETTINGS] der uqsystem.ini den Parameter line-gap setzen. line-gap definiert, welchen Zeilenabstand die Felder haben. Standardmäßig ist line-gap=0 gesetzt. Wenn Sie Feldrahmen benutzen, z.B. LINE oder ENGRAVED, werden sich die Felder vertikal überschneiden. Mit line-gap=4 ist für alle Rahmen genügend Platz vorhanden, aber auch ein Wert von 2 oder 3 kann bei sparsamer Verwendung von Rahmen ausreichen.

Wenn in einem 4GL-Programm die stream-Deklaration line-gap benutzt wird, bleibt die Einstellung des 4GL-Programmes erhalten.

Damit die Änderung dieses Parameters beachtet wird, müssen alle 4GL-Programme nach einem Neustart neu kompiliert werden. (Nur die Parameter im Abschnitt [SETTINGS] erfordern eine erneute Kompilierung, alle anderen Parameter erfordern nur einen Neustart von Unique.)

3. Abhängig von der benutzten Bildschirmauflösung, des benutzten Grafiktreibers und der benutzten Bildschirmschriften können alle Fenster von Unique kleiner als in Version 4.02 wirken. Es können auch die Proportionen auf dem Bildschirm nicht stimmen. Zum Beispiel können die Texte in einzelnen Feldern größer sein als die Rahmen der Felder.

Für eine Anpassung muß wie im vorigen Abschnitt die Unique System Database geändert werden (\$UNIQUE_HOME/setup/uqsystem.ini). Ändern Sie zunächst das Formular, das für die Größenberechnung benutzt wird.

Ein Formular definiert ein internes Koordinatensystem. Die Basiseinheit dieses Koordinationsystems ist eine Zelle (entspricht einem Zeichen in Version 4.02). Die physikalische Größe einer Zelle ist frei definierbar. Die Größe einer Zelle wird in der Unique System Database (der uqsystem.ini) für jedes einzelne Formular festgelegt. Alle Größenangaben innerhalb von Unique 4GL Programmen basieren fest auf Zellen.

Standardmäßig sind drei Formulare definiert: das Formular OnlineDefault für alle Dialogprogramme, XtraDefault für alle Reports und UqEditor für den Unique EDITOR.

Wichtig sind zunächst nur die ersten beiden Zahlen einer Formular-Definition: Die erste Zahl gibt die Höhe einer Zelle in Points an (1 Point entspricht 1/72 Zoll), die zweite Zahl gibt die Breite einer Zelle in Points an. Erhöhen Sie die Werte zum Beispiel auf OnlineDefault=12,5.5,... . Siehe Operations Guide: Unique System Database: [FORMULAR] Section.

```
[FORMULAR]
OnlineDefault=10,4.5,20,pixel
XtraDefault=12,6,10,high
UqEditor=11,5.5,0,pixel
```

Eine Änderung der Formular-Definition wird nach dem nächsten Neustart von Unique aktiv. Mit einer Änderung der vordefinierten Formulare ändern Sie auch das Grunderscheinungsbild von Unique, da alle internen Unique-Applikationen auch diese Formulare benutzen.

Sie können sich auch eigene Formulare definieren, indem Sie eine Zeile wie die folgende in den Abschnitt [FORMULAR] einfügen.

```
MyDefault=12,5.5,20,pixel
```

In Unique 4GL-Programmen können Sie dann ein bestimmtes Formular auswählen. Geben Sie auf Stream-Ebene dafür die Deklaration `formular an`, z.B.:

```
stream="Main "
..
formular('MyDefault')
..
Siehe auch Hinweis b.
```

4. Die Schriften sind nach der Änderung der Formulare aber gleichgeblieben. Das heißt, in einem jetzt größeren Fenster wirken die Texte möglicherweise teilweise zu klein. Wenn Sie größere Schriften benutzen möchten, können Sie die Schriften systemweit ändern. Dafür sind die Abschnitte [SCREEN-FONTS], [PRINTER-FONTS] und [STYLES] in der Unique System Database zuständig. In den Abschnitten [SCREEN-FONTS] und [PRINTER-FONTS] sind die Bildschirm- bzw. Druckerschriften definiert. Einige Standardschriften sind mit eigenem Namen aufgeführt, für die Abwärtskompatibilität sind aber auch alle Schriftnummern wie sie in Version 4.02 benutzt wurden definiert.

Ein Eintrag kann folgendermaßen aussehen:

uqSystem = MS Sans Serif , -8,0,0,0,100,0,0,0,1,2,0,34

Die wichtigsten Parameter sind der Schriftname MS Sans Serif und die Punktgröße -8, die meistens negativ angegeben wird. Um die Schrift etwas zu vergrößern, ändern Sie die -8 in eine -9. Eine genaue Beschreibung finden Sie im Operations Guide: Unique System Database: [FONTS] Section.

Wenn Sie den Schriftnamen oder die Schriftgröße einer Schrift in der Unique System Database ändern,

ändern Sie das Aussehen aller auch der Unique-internen Anwendungen, die diese Schrift benutzen. Die vordefinierten Schriften sollten Sie deshalb auch auf keinen Fall löschen!

Meistens wird bei einem Feld oder Element eine bestimmte Schrift gar nicht mitangegeben. In diesem Fall wird die Schrift wie auch der Rahmen sowie die Vorder- und Hintergrundfarbe aus dem Style abgeleitet. Bei einem Unique 4GL-Programm, das noch gar keine Styles benutzt, werden Standard-Styles abhängig vom Element- oder Feldtyp benutzt. Der Name für den benutzen Standardstyle beginnt mit OL- für alle Dialogprogramme und mit XT- für alle Reports, gefolgt vom Element- bzw. Feldtyp, z.B. PUSH-BUTTON. In einem Dialogprogramm würde also der Style OL-PUSH-BUTTON automatisch für einen Push-Button benutzt.

Ein Style ist eine Zusammenfassung von vier visuellen Eigenschaften, die einem Element, einer Form oder einem Stream über eine einzelne Deklaration zugewiesen werden kann. Styles werden ebenfalls in der Unique System Database definiert. Ein Eintrag kann folgendermaßen aussehen:

OL-PUSH-BUTTON = BLUE, colour(BLACK), uqSystem, embossed

Hier wurde eine Style namens OL-PUSH-BUTTON mit der Hintergrundfarbe BLUE, der Vordergrundfarbe BLACK, der Schrift uqSystem und dem Rahmentyp EMBOSSED definiert.

Siehe auch Reference manual:Declarations and Definitions:Definitions:Style definitions und Operations Guide: Unique System Database: [STYLES] Section

5. Unterschiedlich große Schriften oder Farbwechsel innerhalb von Dialogprogrammen oder Reports wurden bis zur Version 4.02 über stream-Deklarationen colour und font und über DIO-Makros gelöst. DIO-Makros werden in den Forms zwischen start-form und endform geschrieben z.B. :

```
stream="Screen"
colour(BLACK on LIGHT WHITE)
font(2)
```

```
form="Screen"
```

```
start-form
```

```
^FN,18; ^CL,RED on LIGHT WHITE;
```

Überschrift in Arial 24pt und Rot auf Weiß

```
^FN,2; ^CL,BLACK on LIGHT WHITE;
```

Standardschrift in Standardfarbe

```
endform
```

In diesen Fällen wird meist der Hintergrund teilweise in einer falschen Farbe angezeigt. Das liegt daran, daß mit Version 5.12 auch für einzelne Forms die Vorder- und Hintergrundfarbe definiert werden kann (Beachten Sie hierzu die Hinweise h und i.) Um die richtigen Farben dargestellt zu bekommen, müssen Sie die Deklaration background und colour für die Form so definieren, daß diese Deklarationen nur mit der Version 5.12 (oder höher) kompiliert werden. Falls Sie keine Form-Deklaration (form="Screen") definiert haben, sondern start-form direkt auf die Stream-Deklarationen folgt, müssen Sie auch die Form-Deklaration einfügen.

Für Kompatibilität zu Version 4.02:

```
stream="Screen"
colour(BLACK on LIGHT WHITE)
font(2)
```

```
form="Screen"
```

```
$if $UQVERNO1=5
```

```
background(colour(LIGHT WHITE))
```

```
colour(BLACK)
```

```
$endif
```

```
start-form
```

```
^FN,18; ^CL,RED on LIGHT WHITE;
```

Überschrift in Arial 24pt und Rot auf Weiß

```
^FN,2; ^CL,BLACK on LIGHT WHITE;
```

Standardschrift in Standardfarbe

```
endform
```

Für Kompatibilität zu Version 3.12 (und 4.02)

```
stream="Screen"
colour(BLACK on LIGHT WHITE)
font(2)
```

```
$if $UQVERNO1=5
```

```
form="Screen"
```

```
background(colour(LIGHT WHITE))
```

```
colour(BLACK)
```

```
$endif
```

```
start-form
```

```
^FN,18; ^CL,RED on LIGHT WHITE;
```

Überschrift in Arial 24pt und Rot auf Weiß

```
^FN,2; ^CL,BLACK on LIGHT WHITE;
```

Standardschrift in Standardfarbe

```
endform
```

Hinweise für die Erhaltung der Kompatibilität mit Version 3.12 und 4.02

- a. Benutzen Sie nicht mehr als 21 Zeilen Maskeninhalte in einem 4GL-Programm. Auf einem Terminal bei Version 3.12 und 4.02 und bei Standard-VGA-Auflösung unter Version 4.02 läßt sich nicht mehr auf einem Bildschirm anzeigen.

- b. Um Programmcode, der nur bei bestimmten Unique 4GL Versionen ausgeführt werden soll, zu definieren, benutzen Sie die Kompilerdirektive `$if $UQVERNO`. Wenn Sie einen Programmteil nur für Version 5.xx ausführen möchten, definieren Sie `$if $UQVERNO1=5 ... $endif`. Die 1 bewirkt, daß nur das erste Zeichen der Konstante `$UQVERNO` ausgewertet wird. Vermeiden Sie zu genaue Abfragen, z.B. `$if $UQVERNO=5.12-04`, denn das nächste Update kommt bestimmt. Siehe Reference manual:File directives.
- c. In Version 4.02 konnten Sie in der `defaults.ini` Standardattribute für Streams, Forms und Felder ablegen, z.B. `line-gap`, `colour`, `frame`. Diese Datei wird nicht mehr unterstützt. Stattdessen werden `Formulare` und `Styles` benutzt. Siehe Abschnitte 3 und 4.

Hinweis zu Unique II-Syntax

- d. Unique 4GL Version 5.12 unterstützt weiterhin recht viel Unique II-Syntax, z.B. `start-fields.. end-fields`. Bei einer Anpassung des Quelltextes oder bei entsprechenden Compiler-Warnungen sollten Sie die Unique II-Syntax, die seit Version 3.0 (1991) veraltet ist, ersetzen.

Hinweis zur globalen und lokalen Unique System Databases

- e. Die Unique System Database im Setup-Verzeichnis (`$UNIQUE_HOME/setup/uqsystem.ini`) gilt für alle Unique-Systeme und alle Unique 4GL-Anwendungen. Auch das Erscheinungsbild von XTRA oder QUICK ändert sich bei einer Anpassung der globalen Unique System Database. Wenn Sie Einstellungen nur für ein System eintragen möchten, benutzen Sie eine lokale `uqsystem.ini`. Hierzu kopieren Sie die Datei `$UNIQUE_HOME/setup/uqsystem.ini` in ein Verzeichnis, daß im `application-path` des Unique Systems eingetragen ist. (Schenken Sie dem Hinweis in der `uqsystem.ini` keine Beachtung, daß die `uqsystem.ini` auch im `source-path` stehen darf.)

In dieser lokalen Unique System Database können Sie dann alle Änderungen an Styles, Formularen oder Schriften vornehmen und auch neue Einträge einfügen.

Konvertierung zur Element-Syntax mit Benutzung von ISUS (Intelligent Source Update System)

Vorteile

- Alle neuen GUI-Controls und 4GL-Funktionen können vollständig benutzt werden, z.B. Tab-Forms, LDT-System(Sprachunabhängigkeit), Elementtypen Line, Scrollbars, ..
- Die Konvertierung kann weitestgehend von ISUS übernommen werden.
- Anwendungslayout kann den Konventionen von Windows 95/NT 4.0 angepasst werden.

Nachteile

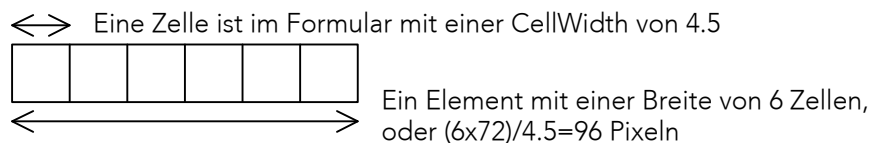
- Quelltext nicht mehr lauffähig mit Versionen 3.12 und 4.02.
- Änderungen am Quelltext notwendig
- Kompilerkonstanten, z.B. \$constant MeineKonstante Hallo, und Kompilerdirektive \$if ... und Include-Anweisung \$include können je nach ISUS-Optionen verloren gehen.

Maßnahmen

5. Führen Sie zuerst die gleichen Schritte wie im Abschnitt Portieren ohne Konvertierung zur Element-Syntax aus. Bevor die Programme zur Element-Syntax konvertiert werden, sollten die Programme durch Änderung der Konfiguration möglichst gut an die Version 5.12 angepasst sein. Das vermeidet oder verringert den später deutlich höheren Aufwand, diese Programme manuell anzupassen.
6. Entscheiden Sie sich für die geeigneten ISUS-Optionen.

Hinweise zu unit-types in der 4GL und Standardgrößen im Forms Painter

- f. Die Deklaration unit-type(CHAR) auf Applikationsebene ist mit Version 5.00 eingeführt worden, wird in Version 5.12 aber nicht mehr benutzt. Stattdessen werden Formulare (im Englischen Formulars nicht Forms) benutzt. Siehe Operations Guide: Unique System Database: [FORMULAR] Section.
- g. Im Forms Painter lassen sich Standardgrößen für neue Elemente definieren. Diese Größen werden nicht in Zellen, sondern in Pixel (in der Dokumentation auch als Point bezeichnet) angegeben. Die Größe einer Zelle ist variabel und basiert auf der Definition des Koordinationsystems des benutzten Formulars. Die Größe eines Pixels hingegen ist immer 1/72 Zoll. Die Breite einer Zelle in Pixeln läßt sich berechnen als $(\text{Breite in Zellen} \times 72) / \text{CellWidth}$. CellWidth ist der zweite Parameter einer Formular-Definition in der Unique System Database. Siehe Operations Guide: Unique System Database: [FORMULAR] Section.



Hinweise zu geänderten Deklarationen colour und background

- h. Die Deklarationen colour und background können ab Version 5.12 für alle Streams, Forms und Elemente definiert werden. Die Deklaration colour für Streams ist allerdings nur für die Abwärtskompatibilität gedacht.
- i. Es gibt eine stärkere Trennung von colour und background. colour bezieht sich nur noch auf die Vordergrundfarbe, background auf den Hintergrund, der in einer Farbe, mit einem Farbverlauf oder einer Bitmap dargestellt werden kann. Die alte Syntax, bei der über colour sowohl Vorder- wie Hintergrundfarbe definiert wird, wird weiterhin unterstützt. Siehe Unique 4GL Reference Manual: Element declarations.
- j. Falsche Formgrößen bei unsichtbaren Forms erst mit 5.12 sichtbar.

