

Programmierung
einer einfachen
Textbaustein-Verarbeitung
mit
Team Developer 5.2

Inhaltsverzeichnis

Einleitung	2
Beschreibung der Anwendung	3
Das Bearbeitungsfenster Basistextbausteine	4
Das Bearbeitungsfenster Textbausteinverarbeitung	5
About.....	7
Aufbau und Codierung der Beispielanwendung	8
Datenbank.....	8
Codierung	8
SAM_AppStartup	8
MDI: Menüaufbau und -aussehen.....	8
Das Arbeitsfenster.....	10
Codierung des (neuen) Grid-Controls.....	11
Nachrichtenauswertung	11
Funktionen	11
Neuen Datensatz in Grid einfügen.....	16
Rich Text Information einem Datensatz zuordnen	16
Zusammenfassung	18
Arbeiten mit Textbausteinen - Textbausteinverarbeitung.....	18
Textbausteine einfügen.....	18
Sonderzeichen suchen und manuell ersetzen	20
Fertige Texte ansehen und drucken.....	21
Zusammenfassung	23

Einleitung

Mit der Version 5.2 des Team Developer wird das neue Rich-Text-Control eingeführt. Mit diesem Control wird die Möglichkeit eröffnet, formatierte Texte in SQLWindows-Anwendungen zu erfassen, in einer Datenbank zu speichern und wieder auszulesen und - im Zusammenhang mit dem Report Builder 5.2 auf einem Drucker oder in eine entsprechende PDF-Datei auszugeben.

Beim RTF-Format handelt es sich um einen Microsoft Standard, der beispielsweise von den Anwendungen Wordpad oder Microsoft Word unterstützt wird. Der RFT-Standard bietet unter anderem die Möglichkeit

- ▶ unterschiedliche Fonts, -größen und -attribute
- ▶ Tabellen und Abbildung
- ▶ Anstriche und Nummerierung
- ▶ usw.

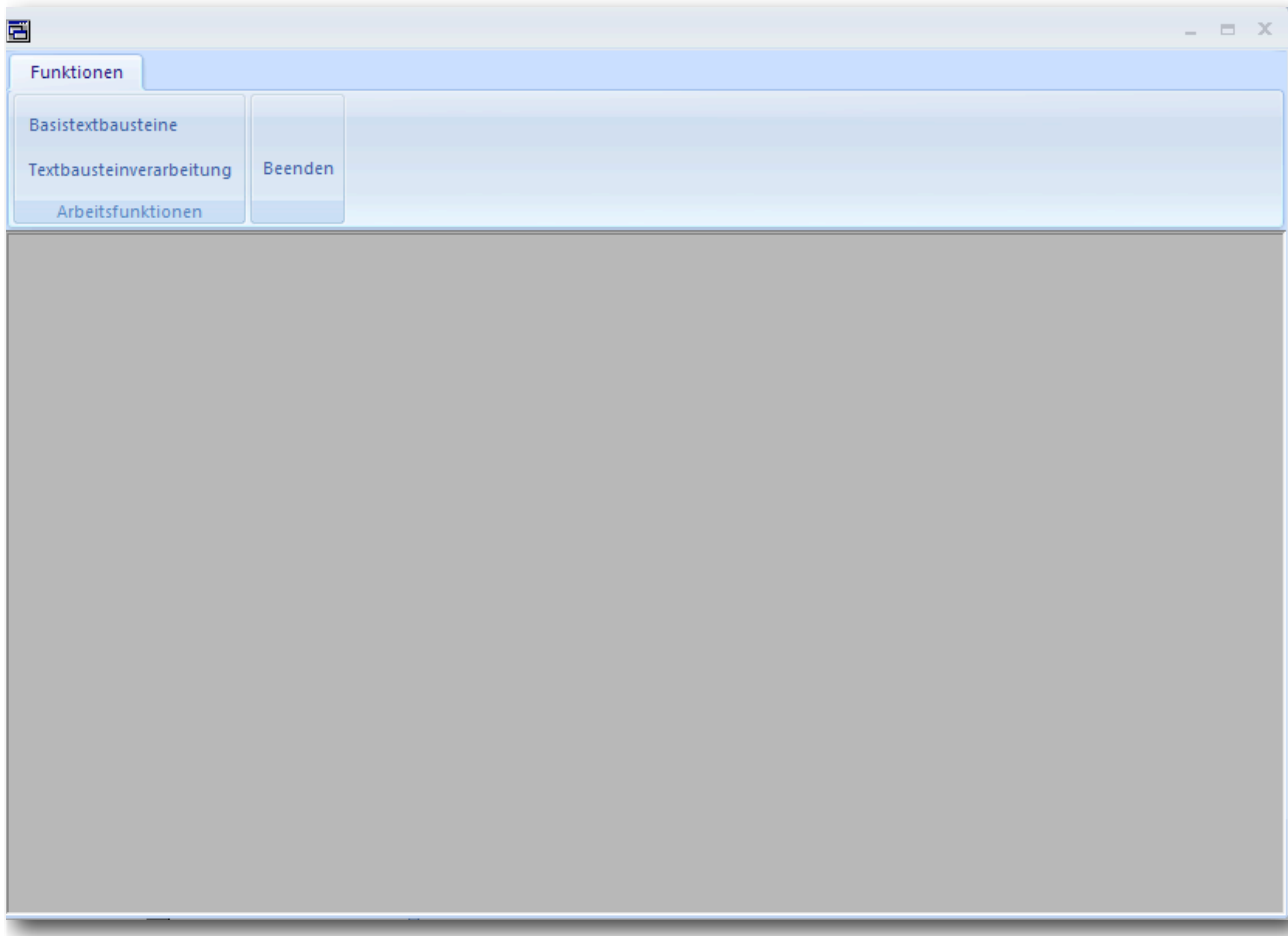
in einem Textelement zu verwalten.

Bei der Beschäftigung mit dem Rich Text Control wurde schon nach kurzer Zeit deutlich, dass es sich um ein sehr mächtiges Werkzeug handelt und ich mich daher auf die Frage vorbereiten sollte, ob es auch möglich ist, die Funktionalität für den Endanwender einzuschränken. So kam ich auf die Idee, eine fiktive Textbausteinverarbeitung zu erstellen, bei der es in zwei unterschiedliche Bearbeitungsfenstern möglich sein sollte, Textbausteine mit mächtiger Funktionalität zu erstellen und in einem anderen Fenster dieser Textbausteine mit eingeschränkter Bearbeitungsfunktionalität für andere Anwender zur Verfügung zu stellen. Auf die explizite Implementierung eines Rechtekonzepts habe ich allerdings verzichtet. In diesem Papier soll gezeigt werden, wie in einer SQLWindows-Anwendung Rich Text Elemente angelegt und gespeichert und wie in einer weiteren Anwendungsmaske die vorhandenen Textbausteine zu einem Dokument zusammengeführt, bearbeitet und gedruckt werden können.

Beschreibung der Anwendung

Die Anwendung stellt zwei Bearbeitungsfenster zur Verfügung:

- ▶ Basistextbausteine und
- ▶ Textbausteinverarbeitung

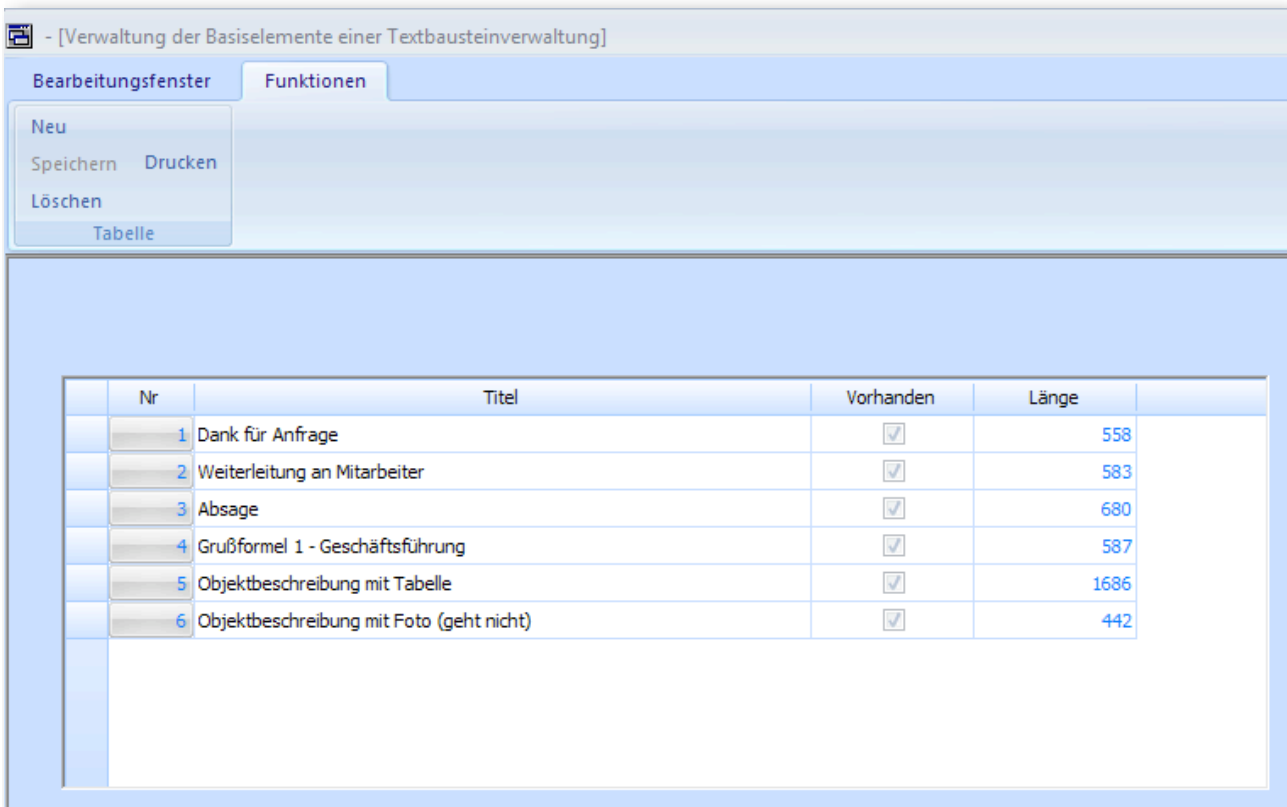


Hauptmenü der Beispielanwendung

Die Abbildung zeigt das Menü in Form einer Ribbon Bar. Die Ribbon Bar selbst besteht aus einem Popup-Menü mit dem Namen Funktionen. In diesem Popup sind wiederum das Popup Arbeitsfunktionen und das Menu Item Beenden dargestellt. Das Popup Menü Arbeitsfunktionen bietet die beiden Menü Items Basistextbausteine und Textbausteinverarbeitung

1. Mit dem Bearbeitungsfenster Basistextbausteine können Textbausteine angelegt, mit einer Bezeichnung versehen und gespeichert werden.
2. Im Bearbeitungsfenster Textbausteinverarbeitung können ausgewählte Textbausteine zusammengefügt, überarbeitet und ausgegeben werden.

Das Bearbeitungsfenster Basistextbausteine



Das Arbeitsfenster Basistextbausteine

Die Abbildung zeigt das zum Bearbeitungsfenster zugehörige Menü Funktionen. Im Arbeitsbereich wird das neue Grid-Control mit den Spalten Nr. Titel, Vorhanden und Länge angezeigt.

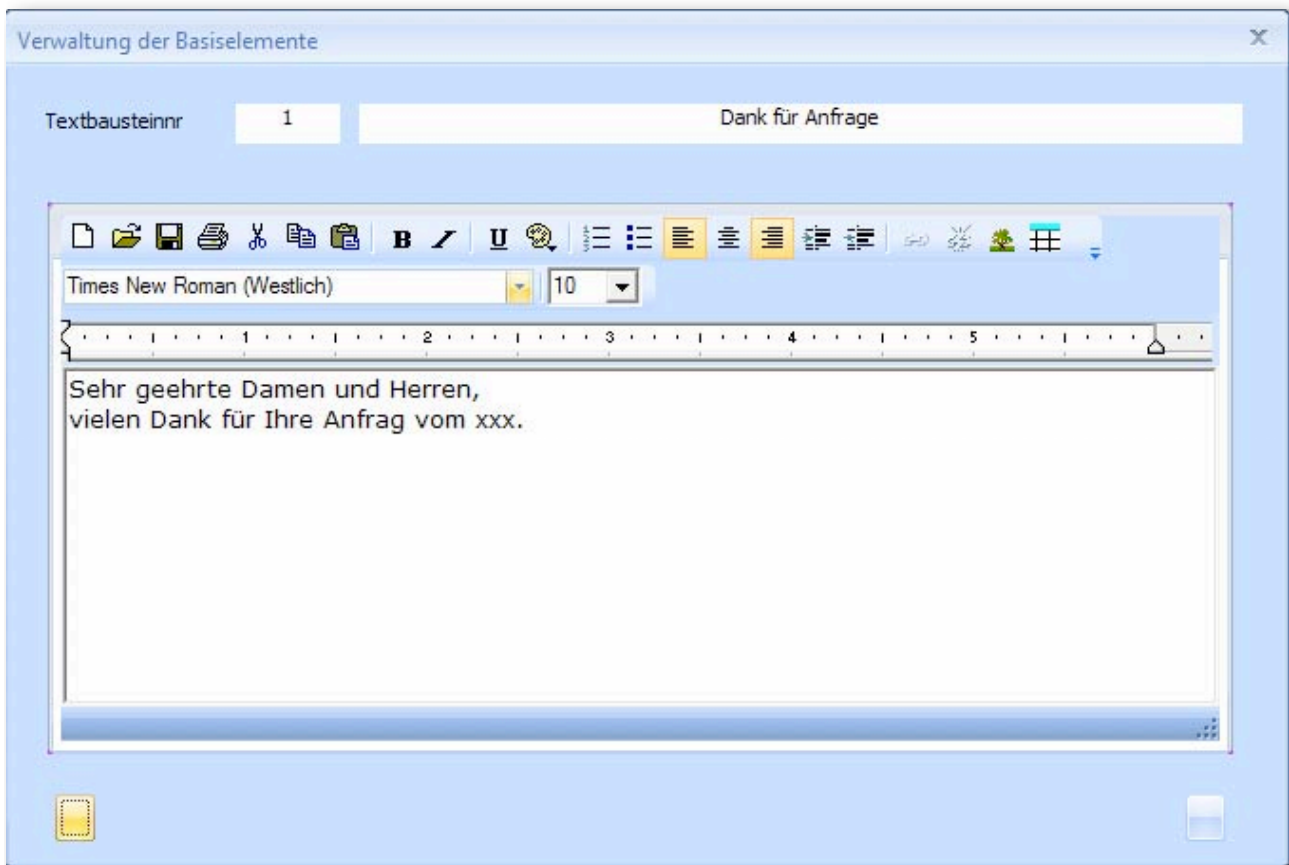
Die Spalte Nr ist als Push Button-Spalte realisiert. Nach Anklicken der Schaltfläche wird ein Dialog zur Bearbeitung des entsprechenden Textbausteins angezeigt.

Die Spalte Vorhanden ist als klassische Check Box-Spalte

Im Menü Funktionen können folgende Funktionen aufgerufen werden:

- ▶ Neu: Einfügen einer neuen Zeile im Grid
- ▶ Speichern: Speichern einer neuen Zeile oder geänderter Informationen in einer vorhandenen Tabellenzeile
- ▶ Löschen: Löschen eines markierten Zeile im Grid
- ▶ Drucken: Anzeige der im neuen Grid integrierten Druckvorschau

Nach Anklicken einer Schaltfläche in der Grid-Spalte Nr wird der Bearbeitungsdialog für den entsprechenden Eintrag angezeigt.



Der Dialog Verwaltung der Basiselemente

Der Dialog Verwaltung der Basiselemente zeigt die Basisdaten aus dem zugrunde liegenden Grid und das neue RTF-Control. Das RTF-Control besteht aus folgenden Bestandteilen:

- ▶ der Toolbar mit Bedienelementen zum Laden, Speichern und Drucken von Rich Text Informationen, Formatierung von Texten wie fett, unterstrichen, Ausrichtung (links, rechts, zentriert), Nummerierung und Anstrichen, usw.
- ▶ Einer Toolbar für die Auswahl von Fonts
- ▶ Einem Lineal und
- ▶ dem Eingabe- und Anzeigebereich von „rich Text“

Im Eingabebereich kann der Anwender seine Textbausteine eingeben und die entsprechenden Formatierungen seines Textes vornehmen. Durch Anklicken der Schaltfläche Speichern werden die Eingabe in der Datenbank SQLBase gespeichert und der Dialog wird geschlossen. Bei Abbrechen werden mögliche Änderungen verworfen und der Dialog beendet.

Das Bearbeitungsfenster Textbausteinverarbeitung

Im Bearbeitungsfenster Textbausteinverarbeitung können die angelegten Textbausteine in jeder beliebigen Kombination eingefügt, editiert und gedruckt werden. Auf das Speichern dieser Texte wurde in diesem Beispiel verzichtet.

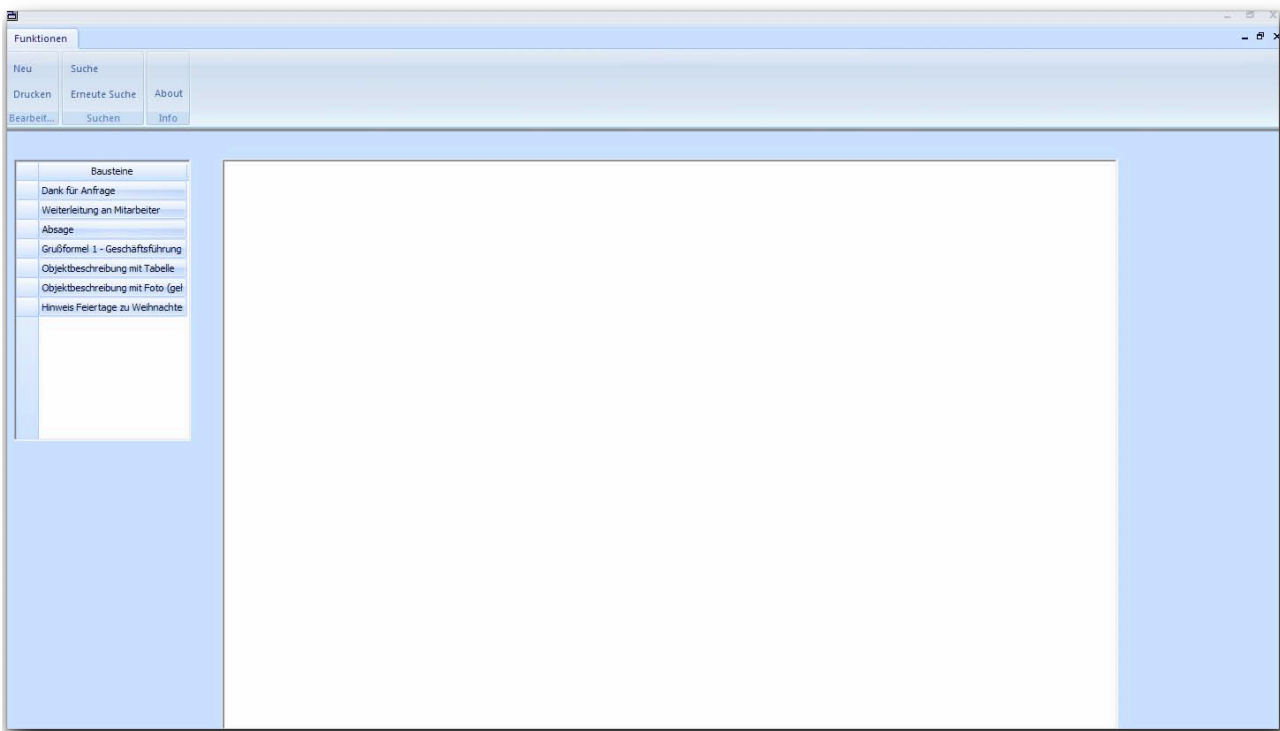
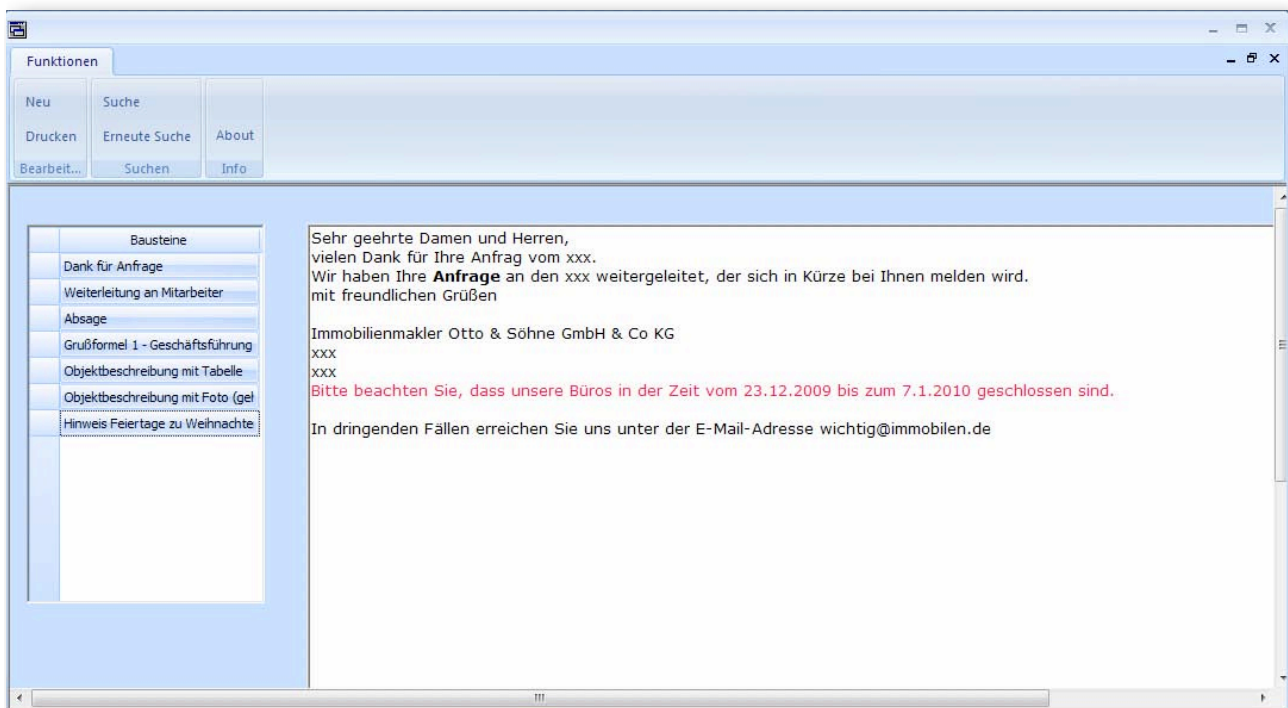


Abbildung Textbausteinverarbeitung

In der Maske Textbausteinverarbeitung gibt es zwei Bedienelemente:

- ▶ ein Grid, in der in der Spalte Bausteine alle vorhandenen Textbausteine angezeigt werden und
- ▶ ein Rich Text Bereich, in den die ausgewählten Textbausteine eingefügt werden

Wählen Sie die Textbausteine, die eingefügt werden sollen, aus. Das Bearbeitungsfenster sollte danach etwa wie folgt aussehen:

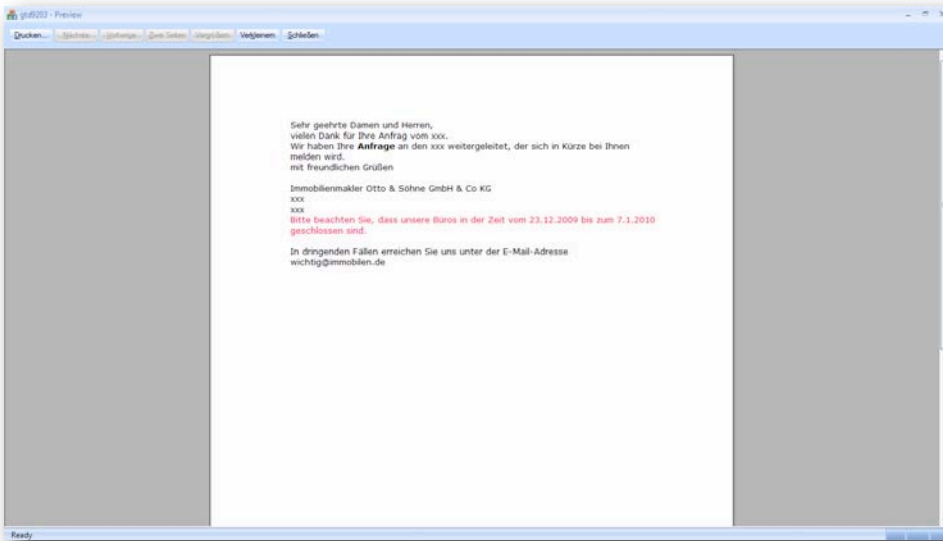


Bearbeitetes Dokument

Im Rich Text Control befinden sich die ausgewählten Textbausteine, die an der jeweiligen Cursorposition eingefügt wurden. Die Formatierungen aus den Bausteinen wurden übernommen.

Sollten Sie weitere Formatierungen vornehmen wollen, bietet Ihnen das Kontextmenü innerhalb des RTF-Controls die entsprechenden Optionen

- ▶ Mit der Menüfunktion Neu wird der bestehende Inhalt gelöscht. Es können wieder neue Textbausteine eingefügt werden.
- ▶ Mit der Menüfunktion Drucken wird das Preview des aktuellen Dokuments angezeigt.



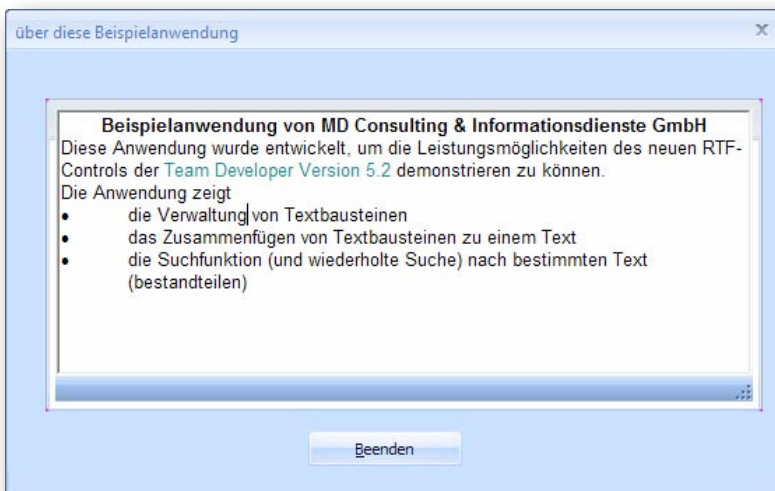
Preview des RTF-Controls nach Einfügen der Textbausteine

Die Abbildung zeigt die im RTF-Control integrierte Druckvorschau. Die Vorschau kann vergrößert und verkleinert werden. Über die Schaltfläche Drucken wird die Ausgabe auf den voreingestellten Standarddrucker ausgelöst.

Mit Schließen wird die Druckvorschau beendet.

About

Durch Anklicken des Menüpunktes About in der Ribbon Bar wird der About Dialog angezeigt, der ein nicht-editierbares RTF-Control mit dem entsprechenden Inhalt anzeigt.



Der About-Dialog mit einem nicht-editierbaren RTF-Control

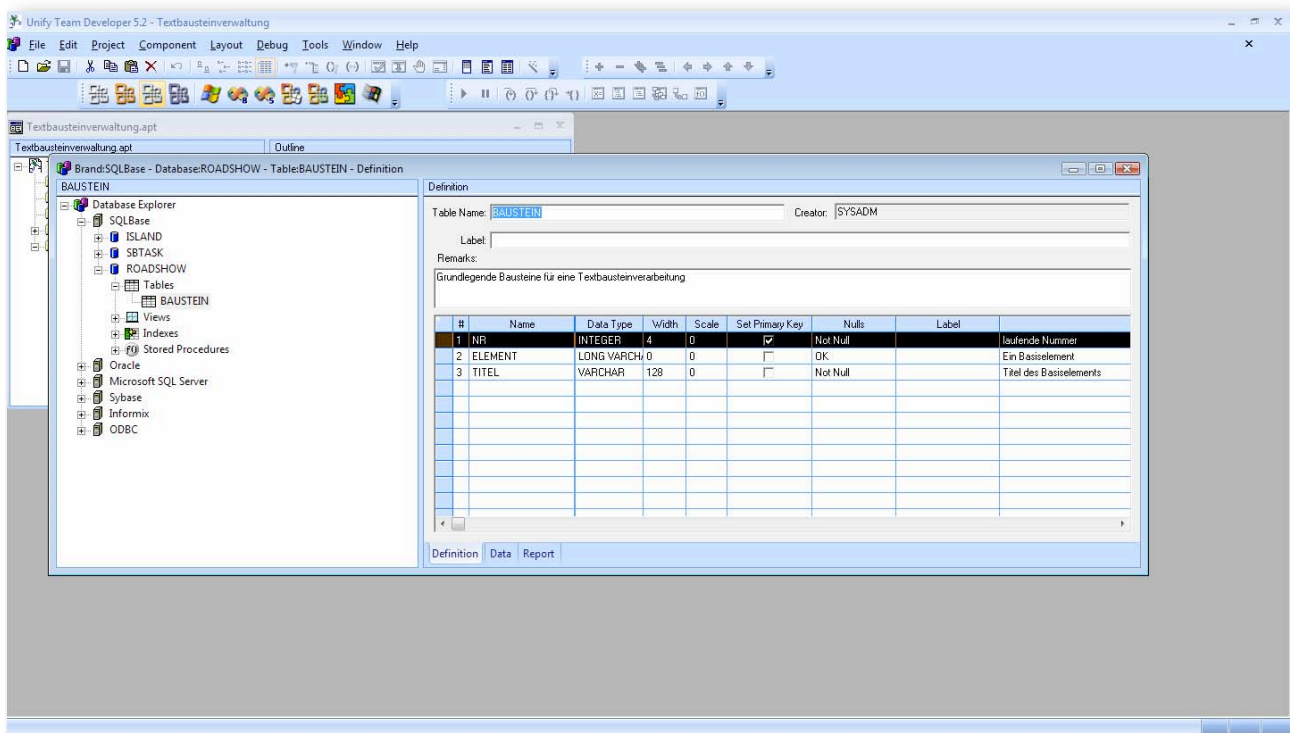
Bei dem RTF-Control wurden sämtliche Zusatzfenster abgeschaltet und zusätzlich die Eigenschaft editable auf No gesetzt.

Nachdem die Anwenderfunktionalität der Beispielanwendung vorgestellt wurde, soll nun gezeigt werden, wie die Anwendung erstellt wurde.

Aufbau und Codierung der Beispielanwendung

Datenbank

Die nachfolgende Abbildung zeigt die Datenstruktur der Beispielanwendung (mit Hilfe des Database Explorers in SQLWindows)



Die Datenbank der Beispielanwendung

Die Anwendung speichert ihre Daten in der Tabelle BAUSTEIN. Die Tabelle hat drei Spalten

- ▶ die Spalte NR, eine INTEGER-Spalte, die immer Werte einhalten muss (NOT NULL)
- ▶ die Spalte ELEMENT, die als LONG VARCHAR definiert ist und
- ▶ die Spalte TITEL, die als VARCHAR-Spalte mit der maximalen Länge von 128 Spalten hinterlegt ist

Es gibt für diese Beispielanwendung nur diese eine Tabelle, in der die Basis-Textbausteine gespeichert werden.

Codierung

SAM_AppStartup

Folgender Code ist in der SAM_AppStartup-Sektion hinterlegt:

```
On SAM_AppStartup
  Set SqlDatabase = „ROADSHOW“
  Set bMenuBarAsRibbonBar = TRUE
```

Code in der SAM_AppStartup-Sektion

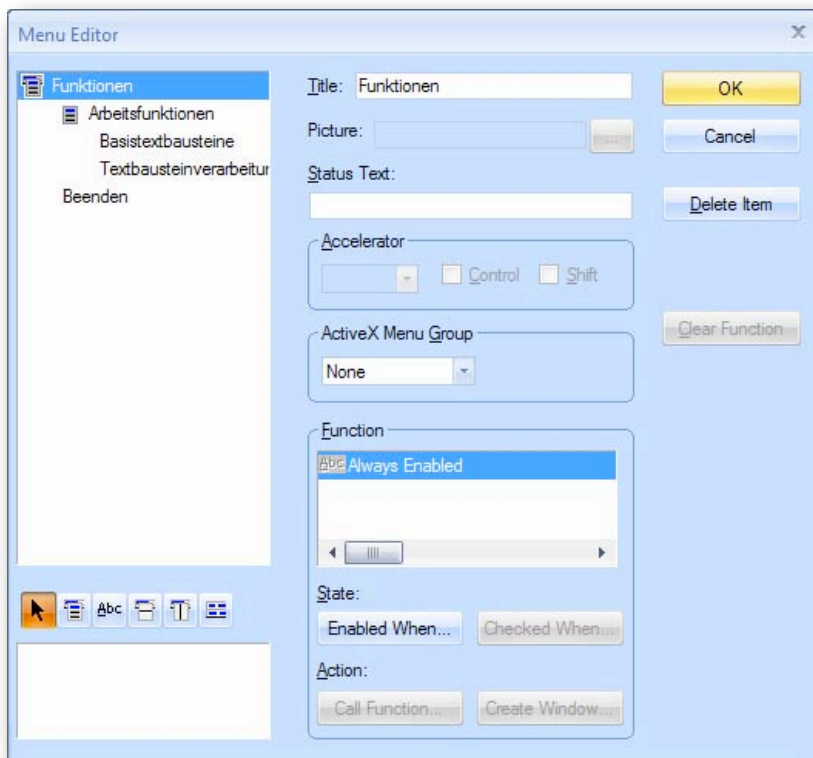
Die mit der Version 5.x eingeführte Systemvariable bMenuBarAsRibbonBar wird auf TRUE gesetzt, um die Anzeige der definierten Menüstruktur als Ribbon Bar zu erreichen.

Wenn diese Systemvariable nicht oder auf FALSE gesetzt wird, wird kein Ribbon Bar Menü, sondern das klassische Menü angezeigt.

MDI: Menüaufbau und -aussehen

- ❖ Es wird von Unify empfohlen, in Zukunft bei der Definition eines Menüs den integrierten Menüeditor zu verwenden.

Die nachfolgende Abbildung zeigt den Menüeditor zusammen mit dem Menü des MDI.

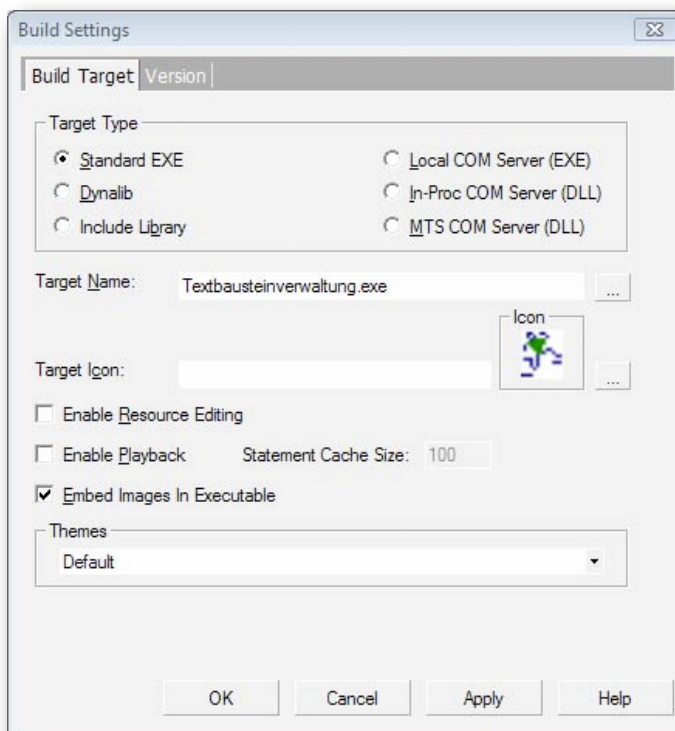


Der Menüeditor mit der Menüstruktur des MDI der Beispielanwendung

Es gibt eine Ribbon Bar Funktionen, die wiederum ein Popup Menü Arbeitsfunktionen enthält. Als Menüitems sind in diesem Beispiel die Funktionen Basistextbausteine, Textbausteinverarbeitung und Beenden vorgesehen.

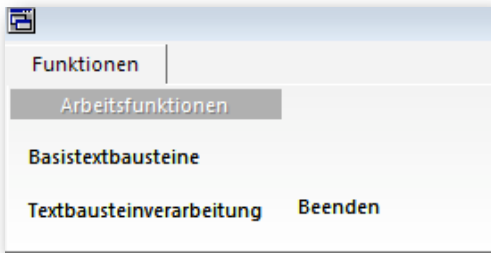
Deutlich zu sehen, ist das Datenfeld Picture mit der Schaltfläche, über die ein entsprechendes Picture aus dem Datensystem ausgewählt und eingebunden werden kann.

- ❖ Es gibt eine neue Option in den Build Settings, die bewirkt, dass alle in SQLWindows verwendeten Abbildungen in die zu erstellende EXE-Datei mit eingebunden werden. Diese Option erweist sich insbesondere für ein vereinfachtes Deployment einer fertigen Anwendung als besonders wertvoll



Build Settings: Einbinden von Abbildungen in die EXE-Datei

Die durch den Menü-Editor (siehe oben) definierte Ribbon Bar hat folgendes Aussehen.



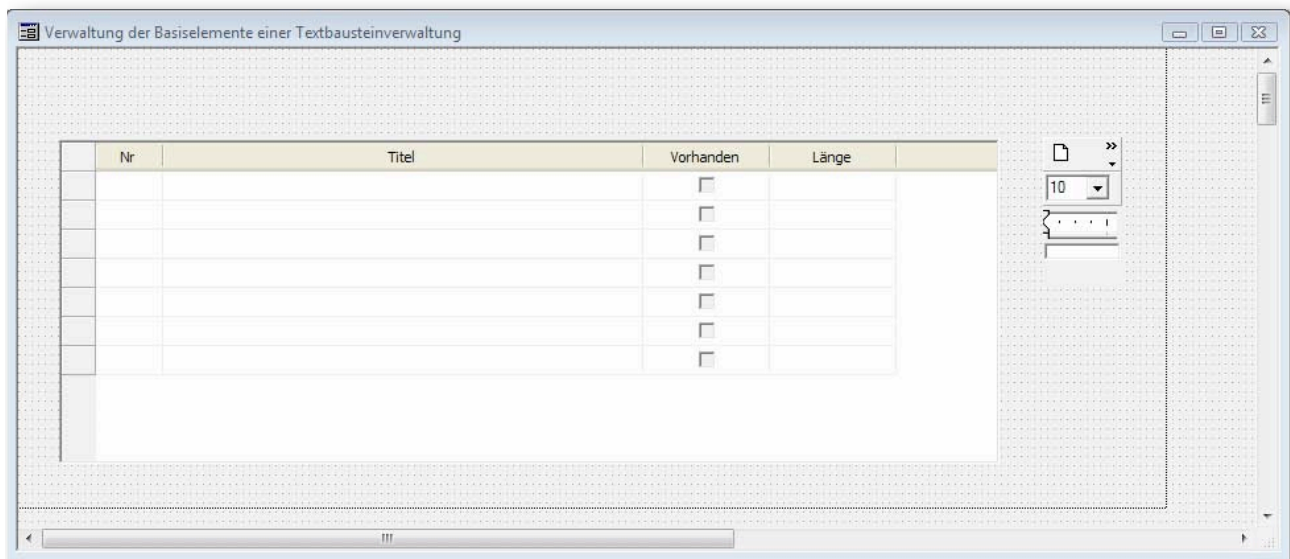
Die Ribbon Bar als Abbild der hinterlegten Menüstruktur

Es gibt nur eine Ribbon Bar beim MDI der Beispielanwendung Arbeitsfunktionen.

- ▶ Gruppirt - als weiteres Popup Menü mit dem Namen Arbeitsfunktionen wurden die Menüitems Basistextbausteine und Textbausteinverarbeitung. Die beiden Menüitems werden unter dem Gruppierungstitel Arbeitsfunktionen angezeigt. Grundsätzlich sind für eine Gruppe drei Einträge vorgesehen. Gibt es mehr Einträge innerhalb einer Gruppierung (d.i. ein Popup-Menü) wird eine neue Spalte in der Gruppierung angelegt.
- ▶ Als weiteres Menüitem - ohne Gruppierung - ist das Menüitem Beenden vorgesehen.

Das Arbeitsfenster

In der Layoutsicht hat das Arbeitsfenster folgendes Aussehen



Layoutsicht auf des Arbeitsfenster Basistextbausteine

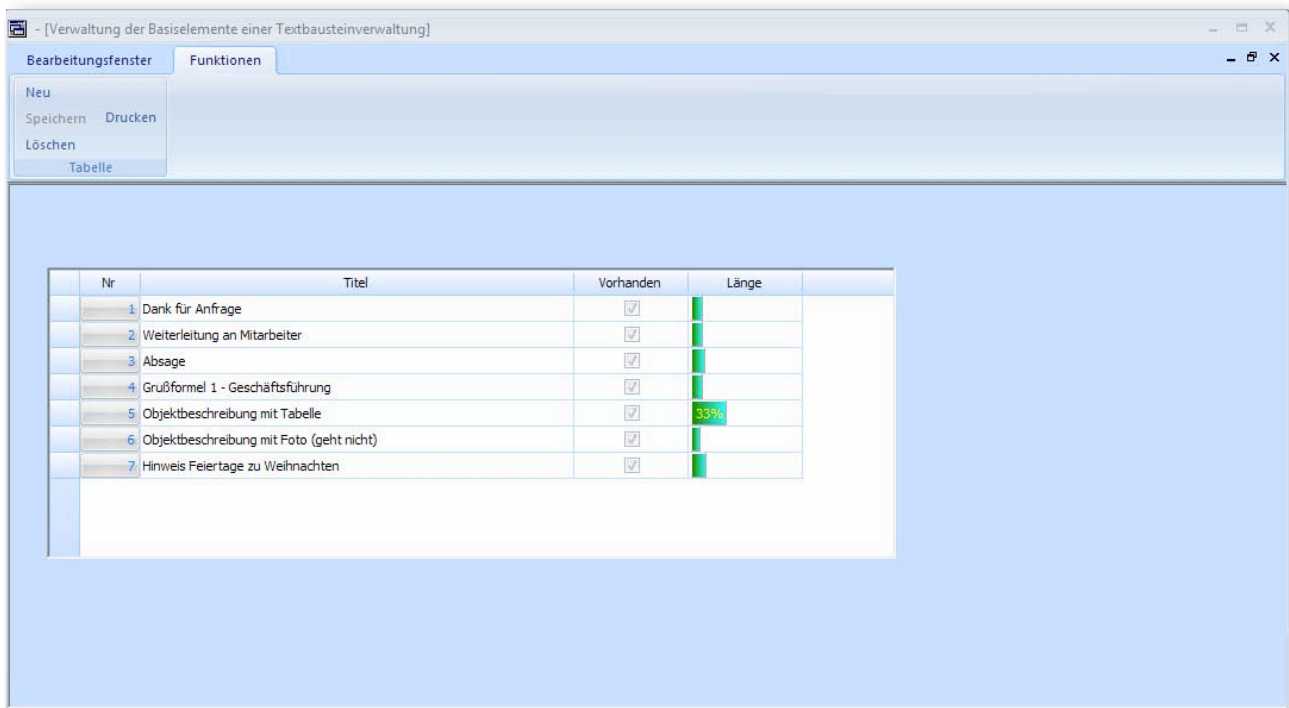
Folgende Controls befinden sich im Arbeitsfenster Basistextbausteine

- ▶ das (neue) Grid-Control mit den Spalten Nr, Titel, Vorhanden und Länge und
- ▶ ein Rich Text Control (, dass allerdings während der Ausführung nicht angezeigt wird.

Innerhalb des neuen Grid-Controls werden neue Spaltentypen verwendet:

- ▶ Die Spalte Nr ist vom Typ Push Button und
- ▶ Die Spalte Länge ist von Typ Progress

Diese Definition und andere Aspekte, auf die weiter unten eingegangen wird, führen dazu, dass dieses Arbeitsfenster während der Ausführung folgendes Aussehen hat.



Das Arbeitsfenster Basistextbausteine während der Ausführung

Die Spalte Nr ist als Schaltflächen-Spalte und die Spalte Länge zeigt die Anzahl Zeichen des hinterlegten RTF-Codes in Form eines Fortschrittsbalkens an.

- ❖ Im Grid werden nicht editierbare Spalten standardmäßig hellblau angezeigt. Sollte diese Einstellung nicht angemessen sein, stehen nach wie vor die entsprechende Funktionen als der GTBLEX.apl zur Verfügung, mit denen eine andere Schriftfarbe definiert werden kann.

Codierung des (neuen) Grid-Controls

Nachrichtenauswertung

```
On SAM_CreateComplete
  Call VorhandeneTextbausteineAuslesen( )
On PM_Neu
  Call NeueTabellenzeileEinfügen( )
On PM_Speichern
  Call DatenSpeichern( )
On SAM_FetchRowDone
  Call TextbausteinÜberprüfen( colNr ,lParam)
On PM_Drucken
  Call DruckvorschauAnzeigen( )
```

Message Action Sections beim Grid Control

Das Grid Control wertet die Standard-Nachrichten SAM_CreateComplete, SAM_FetchRowDone und die Programmierer-Nachrichten PM_Neu, PM_Speichern und PM_Drucken aus.

Funktionen

Im folgenden wird der Source Code der Funktionen, die beim Grid hinterlegt sind, dargestellt und - wenn notwendig - erläutert.

Die Funktion VorhandeneTextbausteineAuslesen wird aufgerufen, wenn das neue Grid auf dem Formular während der Ausführung fertig gezeichnet wurde.

Function: VorhandeneTextbausteineAuslesen

Description: Auslesen aller Textbausteine aus der Datenbank

Returns

Parameters

Static Variables

Local variables

Sql Handle: hSql

String: sStatement

Actions

Call SalWaitCursor(TRUE)

Call SalTblReset(hWndForm)

Call SalGridSetCellProp(hWndForm, GRIDAREA_Column, colAnzahl, lParam, CellProp_Progress_RangeBegin , 0)

Call SalGridSetCellProp(hWndForm, GRIDAREA_Column, colAnzahl, lParam, CellProp_Progress_RangeEnd , 5000)

Call SalGridSetCellProp(hWndForm, GRIDAREA_Column, colAnzahl, lParam, CellProp_Progress_ClrBegin , COLOR_DarkGreen)

Call SalGridSetCellProp(hWndForm, GRIDAREA_Column, colAnzahl, lParam, CellProp_Progress_ClrEnd , COLOR_Cyan)

Set sStatement = 'SELECT NR, TITEL FROM BAUSTEIN ORDER BY 1 INTO :colNr, :colTitel'

Call SqlConnect(hSql)

Call SalTblPopulate(hWndForm, hSql, sStatement, TBL_FillAll)

Call SqlDisconnect(hSql)

Call SalWaitCursor(FALSE)

Die Funktion VorhandeneTextbausteineAuslesen

Beim Auslesen der Daten, welche auch beim neuen Grid mit der Funktion SalTblPopulate vorgenommen werden kann, wird festgelegt, dass die Tabellenspalte, die über die Eigenschaften vom Typ Progress definiert wurde, als Bereich von 0 bis 5000 interpretiert wird. Das es bei der anzuzeigenden Zahl um die Anzahl Zeichen geht, die als Rich Text hinterlegt sind, zeigt der Fortschrittsbalken also an, wie viele Zeichen in Prozent bereits in das entsprechende Rich Text-Objekt eingegeben worden sind.

Mit dem Parameter CellProp_Progress_ClrBegin bzw. CellProp_Progress_ClrBegin kann der Anwendungsprogrammierer den Farbverlauf innerhalb des Fortschrittsbalkens definieren. Die beiden Codezeilen bewirken also, dass die Darstellung im Fortschrittsbalken von Dunkelgrün nach Cyan verläuft. Eine Gleichsetzung von Beginn- und Endfarbe bewirkt, dass die Fortschrittsbalken ohne Verlauf dargestellt werden.

Function: NeueTabellenzeileEinfügen

Description: Einfügen einer neuen Zeile im Grid und Positionierung auf die zweite Spalte

Returns

Parameters

Static Variables

Local variables

Number: nZeile

Actions

Set nZeile = SalTblInsertRow(hWndForm, 0)

Call SalTblSetFocusCell(gridÜbersicht, nZeile, colTitel, NUMBER_Null, NUMBER_Null)

Set bNew = TRUE

Set bSpeichern = TRUE

Die Funktion NeueTabellenzeileEinfügen

In der Funktion NeueTabellenzeilenEinfügen sind keine wesentlich neuen Punkte codiert. Die beiden bool'schen Variablen bNew und bSpeichern wurden beim Form Window definiert: sie bewirken die Grau- bzw. Schwarz-Steuerung der entsprechenden Menüpunkte in der Ribbon Bar.

Function: DatenSpeichern

Description:

Returns

Parameters

Static Variables

Local variables

Sql Handle: hSql

String: sStatement

Actions

Call SqlConnect(hSql)

If SalTblAnyRows(hWndForm, ROW_New , NUMBER_Null)

Set colNr = NächsterWert(hSql)

Set sStatement = 'INSERT INTO BAUSTEIN (NR, TITEL) VALUES (:colNr, :colTitel)'

Call SqlPrepare(hSql, sStatement)

Call SalTblDoInserts(hWndForm, hSql, TRUE)

If SalTblAnyRows(hWndForm, ROW_Edited, NUMBER_Null)

Set sStatement = 'UPDATE BAUSTEIN SET TITEL = :colTitel WHERE NR = :colNr'

Call SqlPrepare(hSql, sStatement)

Call SalTblDoUpdates(hWndForm, hSql, TRUE)

Set bSpeichern = FALSE

Call SqlCommit(hSql)

Call SqlDisconnect(hSql)

Die Funktion DatenSpeichern

Bei der Funktion DatenSpeichern wurde die Standard-Verarbeitung für die Ausführung von INSERT- bzw. UPDATE-Anweisungen (ohne ROWID) codiert. Zudem wird der Menüpunkt Speichern deaktiviert, da er nur bei Änderungen im Grid oder neuen Datenzeilen aktiviert wird.

Function: NächsterWert

Description:

Returns

Number:

Parameters

Sql Handle: hSql

Static Variables

Local variables

String: sStatement

Number: nReturn

Number: nSchlüssel

Actions

Set sStatement = 'SELECT MAX(NR) FROM BAUSTEIN INTO :nSchlüssel'

Call SqlPrepareAndExecute(hSql, sStatement)

Call SqlFetchNext(hSql, nReturn)

Set nSchlüssel = nSchlüssel + 1

Return nSchlüssel

Die Funktion NächsterWert

Die Funktion NächsterWert erzeugt den nächsten Primärschlüsselwert für die INSERT-Operation. Da programmtechnisch sichergestellt ist, dass immer nur ein neuer Datensatz nach einander eingefügt werden kann, läuft auch diese Routine einwandfrei durch.

Function: TextbausteinÜberprüfen

Description:

Returns

Parameters

Number: p_nNr

Number: p_lParam

Static Variables

Local variables

Long String: lsElement

Sql Handle: hSql

String: sStatement

Number: nReturn

Number: nLänge

Actions

Call SqlConnect(hSql)

Set sStatement = 'SELECT ELEMENT FROM BAUSTEIN WHERE NR = :p_nNr INTO :lsElement'

Call SqlPrepareAndExecute(hSql, sStatement)

Call SqlSetLongBindDatatype(Bindvariable.Eins, Datentyp.Binär)

Call SqlFetchNext (hSql, nReturn)

Call SqlDisconnect(hSql)

Set nLänge = SalRTFTextPaste(rtc1, lsElement)

If nLänge <= 1

Set colVorhanden = FALSE

Else If nLänge > 1

Set colVorhanden = TRUE

Set colAnzahl = nLänge

Die Funktion TextbausteinÜberprüfen

In der Funktion TextbausteinÜberprüfen werden Informationen in Bezug auf einen möglicherweise bereits hinterlegten Textbaustein ermittelt.

Es wird ermittelt

- ▶ ob bereits ein Textbaustein angelegt und
- ▶ wenn ja, wie viele Zeichen der Textbaustein enthält.

Die Spalte colVorhanden ist als Check Box Spalte definiert, wobei der Check-Wert TRUE ist und der Uncheck-Wert mit FALSE definiert wurde.

Bei der Ausführung der Routine wird der entsprechende Rich Text bei Eintreffen der SAM_FetchRowDone-Nachricht in die Long String Variable lsElement eingelesen.

Beim Auslesen von Rich Text Informationen muss die Funktion SqlSetLongDatatype aufgerufen werden, um festzulegen, welcher Datentyp verwendet werden soll. Bei der Funktion SqlSetLongDatatype definiert der erste Parameter das n-te Element von Bind-Variablen in der SELECT-Anweisung, während über den zweiten Parameter der zu verwendende Datentyp spezifiziert wird.

- ❖ Beim Lesen von gespeicherten Rich Text Informationen aus einer SQLBase Datenbank muss immer der Datentyp 23, d.i. Datentyp.Binär verwendet werden. Es ist sicherlich etwas gewöhnungsbedürftig, dass die RTF-Informationen in einer Long Varchar-Spalte als binäre Informationen gespeichert werden.
- ❖ Um zu vermeiden, um Source Code mit so genannten „magical numbers“ zu arbeiten - Zahlen, deren Bedeutung möglicherweise einem anderen Programmierer nicht (leicht) zu erschließen ist, wurde in diesem Beispielcode mit Enumerations - numerischen Konstantengruppen gearbeitet. Es wurden die Enumerations Bindvariable und Datentyp angelegt.


```

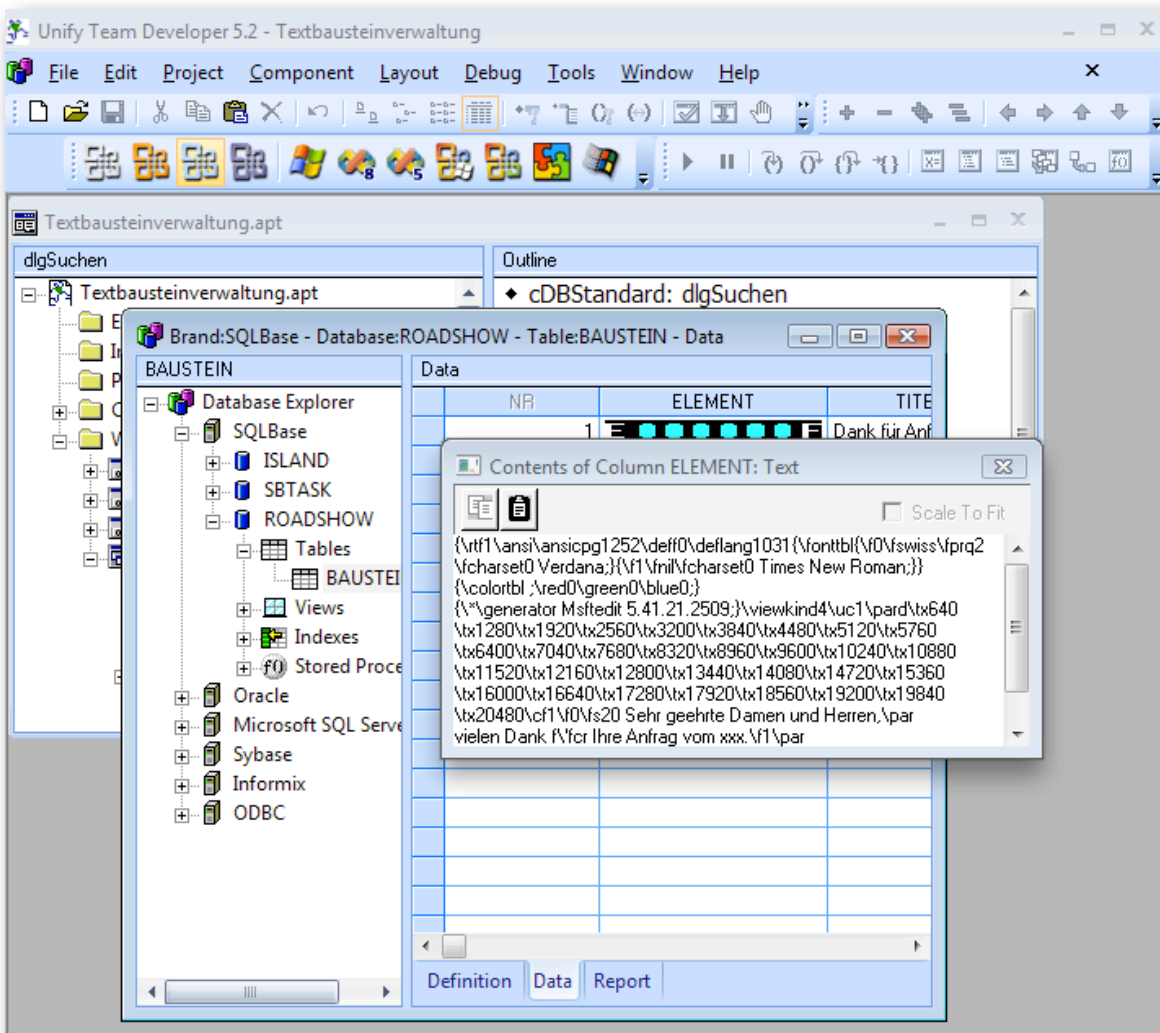
Enum: Datentyp
Attributes
Item: String      =      22
Item: Binär
Item: Long
Enum: Bindvariable
Attributes
Item: Eins        =      1
Item: Zwei
Item: Drei
Item: Vier
Item: Fünf
Item: Sechs
Item: Sieben
Item: Acht
Item: Neun

```

Eigene Enumerations im Beispielcode

Bei der Enumeration Datentyp wurde als erster Wert der Wert 22 angesetzt. Alle nachfolgenden Items einer Enumeration erhalten den nächst höheren Wert zugewiesen. Daher ist der Wert von Datentyp.Binär gleich 23.

Die Implementierung ist durchgängig gehalten. Wenn wir uns beispielsweise den vorhandenen Datenbestand über den Database Explorer von SQLWindows anschauen, sehen wir in der Data-Sicht nach Anklicken der entsprechenden Rich Text Spalte den hinterlegten Rich Text.



Anzeige der Rich Text Information im Database Explorer von SQLWindows

Aus der Abbildung soll deutlich werden, dass die Verarbeitung von Rich Text Informationen durchgängig über die Werkzeuge SQLBase, Report Builder und SQLWindows über die Interpretation als binäre Daten vorgenommen wird.

Function: DruckvorschauAnzeigen

Description:

Returns

Parameters

Static Variables

Local variables

Actions

Call SalGridPrintPreview(hWndForm)

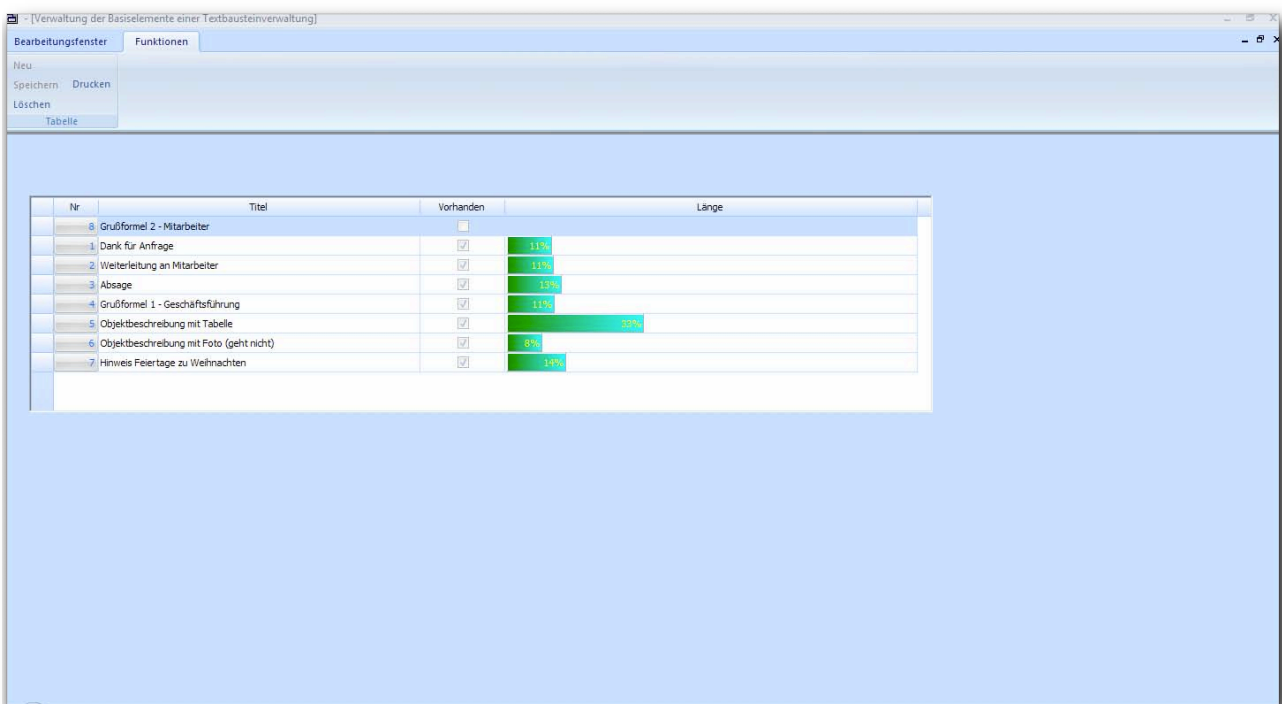
Die Funktion DruckvorschauAnzeigen

Die Funktion DruckvorschauAnzeigen ruft die neue Grid-Funktion SalGridPrintPreview auf. Diese Funktion arbeitet nur mit dem neuen Grid und nicht mit dem alten (Child) Table Window.

Neuen Datensatz in Grid einfügen

Über den Menüeintrag Neu in der Ribbon bar wird eine neue Zeile als oberste Zeile im Grid angelegt. Der Anwender kann jetzt lediglich einen Namen für den Textbaustein eingeben, da alle anderen Angaben beim Auslesen der Daten nach dem Speichern automatisch berechnet werden. Programmtechnisch wird der nächste Primärschlüsselwert über die Funktion NächsterWert gebildet, die dann bei der INSERT-Anweisung neben dem eingegebenen Titel gespeichert wird.

Nach dem Speichern des neuen Datensatzes werden alle Daten erneut in das Grid eingelesen.

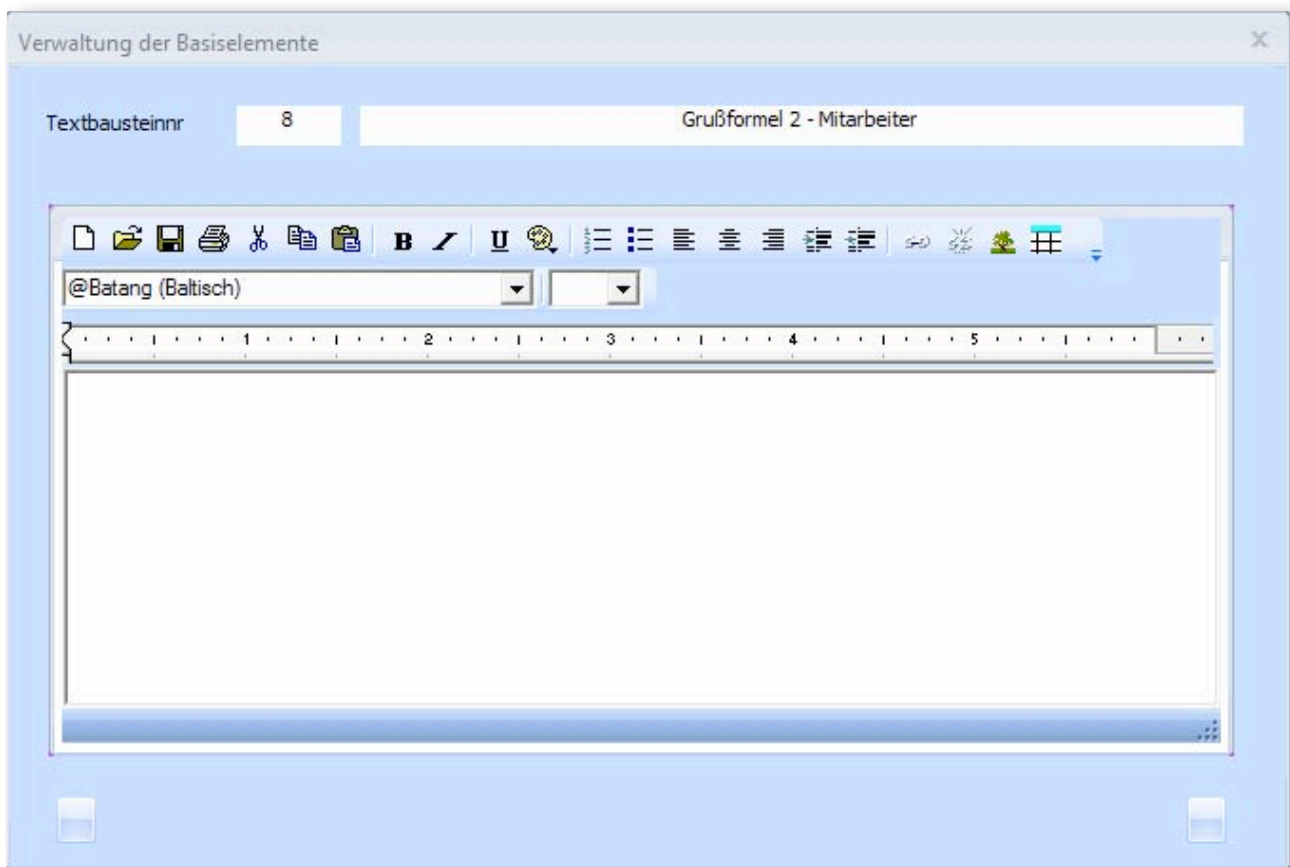


Erneutes Einlesen der Daten in das Grid

Der neue Eintrag Grußformel 2 - Mitarbeiter enthält noch keinen Rich Text. Aus diesem Grund ist die Check Box bei diesem Eintrag nicht gesetzt und auch die Anzahl Zeichen in der Spalte Länge enthält keine Fortschrittsinformation.

Rich Text Information einem Datensatz zuordnen

Um den Rich Text anzulegen bzw. einen vorhandenen Rich Text zu bearbeiten, muss die entsprechende Schaltfläche in der Spalte Nr angeklickt werden. Danach öffnet sich der Dialog für die Bearbeitung von Rich Text-Informationen.



Der Dialog Verwaltung der Basiselemente ohne einen Eintrag

Angezeigt werden die Nr, der Textbaustein-Titel und des RTF-Control mit allen Hilfsfenstern.

Der Anwender kann jetzt einen Textbaustein mit allen möglichen Formatierungsoptionen anlegen und beim entsprechenden Datensatz als Rich Text speichern.

Function: TextbausteinSpeichern

Description: Umwandeln des Inhalts des RFT-Controls in Long String und binäres Speichern eines Textbausteins

Returns

Parameters

Static Variables

Local variables

String: sStatement

Sql Handle: hSql

Long String: lsElement

Number: nPuffer

Actions

Set nPuffer = PUFFER_MaximaleLänge

Call SalRTFTextSelectAll(rtc1)

Call SalRTFTextCopy(rtc1, lsElement, nPuffer)

Call SqlConnect(hSql)

Set sStatement = 'UPDATE BAUSTEIN SET ELEMENT = :lsElement WHERE NR = :nNr'

Call SqlPrepare (hSql, sStatement)

Call SqlSetLongBindDatatype(Bindvariable.Eins, Datentyp.Binär)

Call SqlExecute(hSql)

Call SqlCommit(hSql)

Call SqlDisconnect(hSql)

Die Funktion TextbausteinSpeichern

Beim Speichern von Informationen aus einem RTF-Control muss ebenfalls zwischen dem Übersetzen der SQL-Anweisung (SqlPrepare) und der Ausführung (SqlExecute) die Funktion SqlSetLongBindDatatype aufgerufen werden.

Die Logik der Funktion besteht darin, den Inhalt eines RTF-Controls zu markieren, um danach den markierten Bereich in eine Long String Variable zu übertragen. Diese Information wird im binären Format bei der Ausführung der UPDATE-Anweisung verwendet.

Zusammenfassung

Der Umgang mit dem neuen RTF-Control im Zusammenhang mit der Datenbank SQLBase (11.5) ist damit erklärt. Folgende Punkte können als einfache Checkliste angesehen werden.

- ▶ Die Tabellenspalten in einer SQLBase-Datenbank müssen als LONG VARCHAR-Spalten definiert werden.
- ▶ Zum Schreiben von RTF-Informationen müssen die Informationen zunächst aus dem Control extrahiert - Markieren und Kopieren mit den entsprechenden SalRTFText-Funktionen und einer Variablen vom Typ Long String zugewiesen werden.
- ▶ Bei INSERT- und UPDATE-Operationen muss vor dem SqlExecute die Funktion SqlSetLongBindDatatype angewendet werden, wobei der erste Parameter numerisch auf die entsprechende Bind-Variable in der SQL-Aweisung zeigt, während der zweite Parameter immer 23 (Datentyp.Binär) sein muss.
- ▶ Beim Lesen von RTF-Informationen aus der Datenbank muss vor der Ausführung der SqlFetchNext-Anweisung wiederum die Funktion SqlSetLongBindDatatype mit der entsprechenden Festlegung - siehe oben - ausgeführt werden.

Im nächsten Abschnitt sollen die gespeicherten Informationen verwendet werden, um dynamisch aus den vorhandenen Elementen einen neuen Text zu erstellen.

Arbeiten mit Textbausteinen - Textbausteinverarbeitung

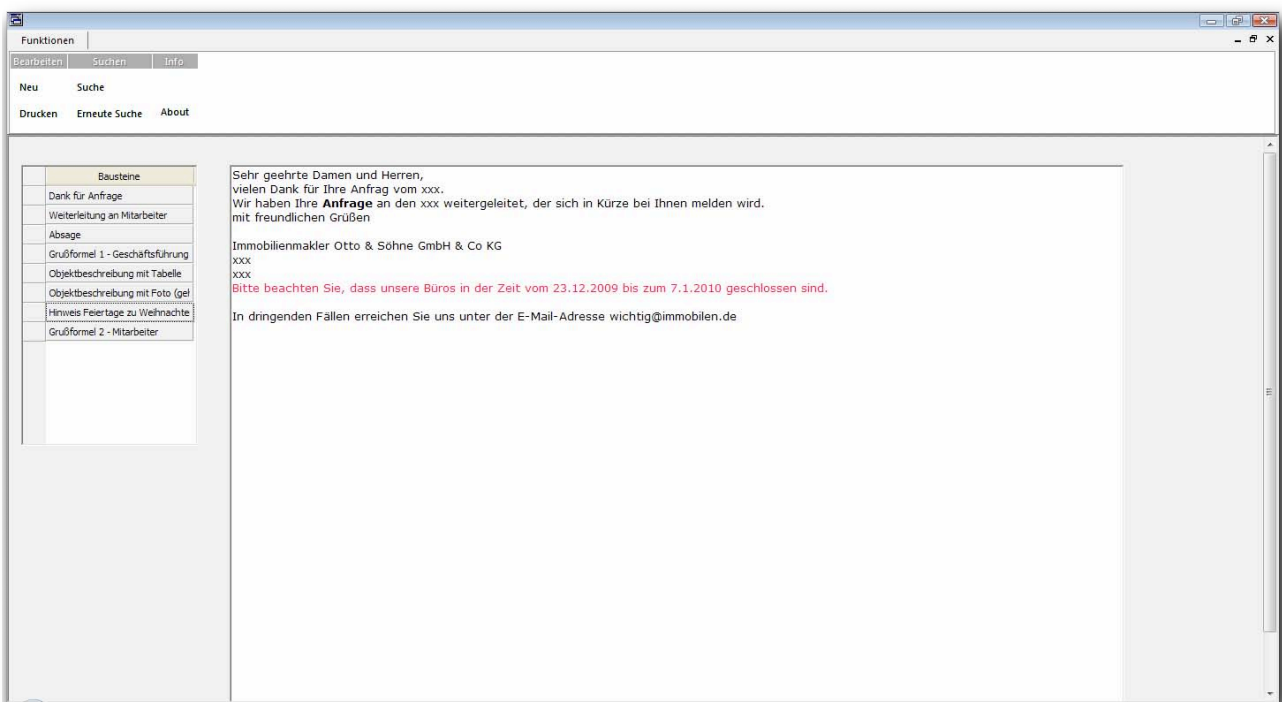
Im Bearbeitungsfenster Textbausteinverarbeitung sollen die Basiselemente durch einen Anwender nach Belieben zusammengestellt werden. Zunächst soll gezeigt werden, wie in einem RTF-Control beliebige Informationen zu einem Text zusammengeführt werden können.

Danach soll gezeigt werden, wie mit der entsprechenden Suchfunktion in den Textbausteinen hinterlegte Informationen gesucht und markiert werden können, damit der Anwender die entsprechenden Platzhalter überschreiben kann. Hierfür wird eine einfache Suchfunktion implementiert, die wiederholt aufgerufen werden kann.

Als letztes soll gezeigt werden, wie das Ergebnis der Textbausteinverarbeitung ausgegeben werden kann. Die Möglichkeit, das erstellte Dokument zu speichern wird hier nicht noch einmal behandelt, da diese Funktion bereits im ersten Abschnitt ausführlich vorgestellt und erläutert wurde.

Textbausteine einfügen

Die nachfolgende Abbildung zeigt das Bearbeitungsfenster, nachdem eine Reihe von Textbausteinen eingefügt wurden. In der linken Seite des Fensters befindet sich ein Grid mit einer Spalte, in der die Titel der Textbausteine angezeigt werden. Die Spalte ist vom Typ Push Button definiert. Wenn der Anwender auf eine Schaltfläche klickt, wird der entsprechende RTF-Text aus der Datenbank geladen und an der aktuellen Cursor-Position in das RTF-Control eingefügt.

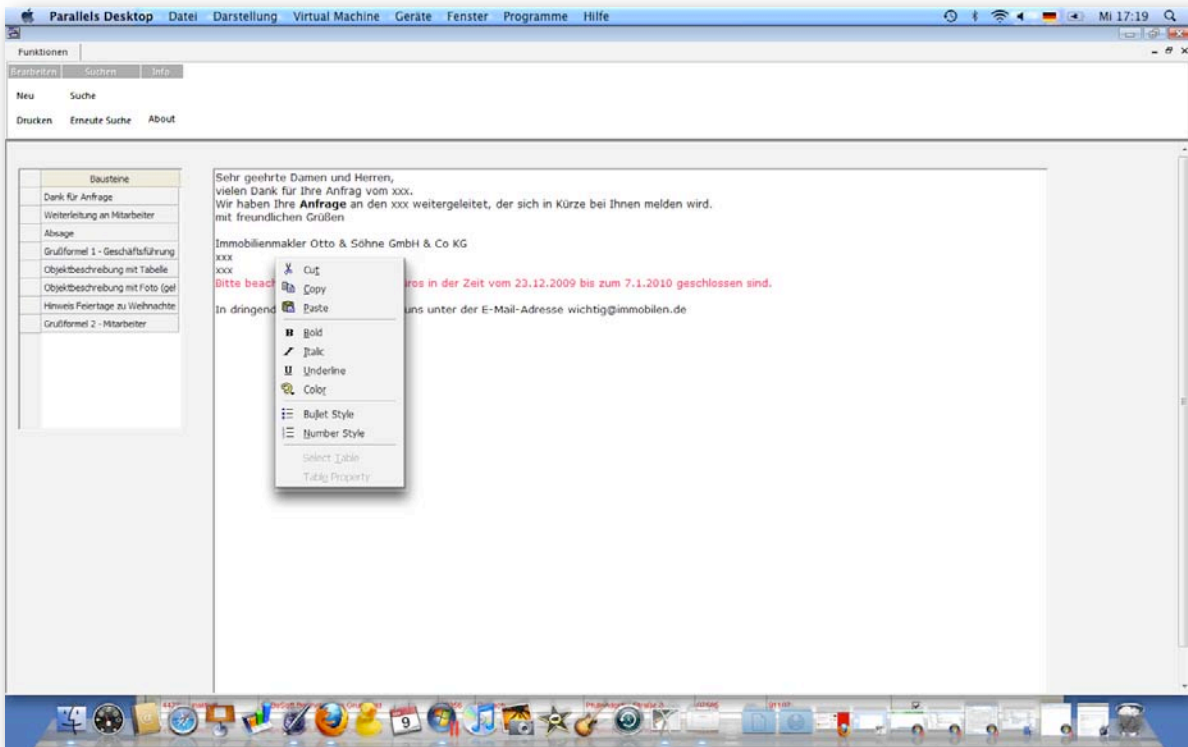


Ein neues Dokument mit einer Reihe von Textbausteinen

Die übernommenen Textbausteine enthalten nach wie vor die Formatierungen, die bei der Definition der Textbausteine festgelegt wurden.

Es wurden dem Anwender in diesem Arbeitsfenster explizit die Möglichkeiten der freien Textgestaltung über die entsprechenden Toolbars vorenthalten. Hiermit soll erreicht werden, dass nicht jeder Anwender wie Kraut und Rüben beispielsweise die unterschiedlichsten Zeichenfonts in einem Dokument verwenden kann.

Allerdings sollen schon einige einfache Möglichkeiten der Textgestaltung zur Verfügung stehen: wählen Sie über die rechte Maustaste das Kontextmenü des Rich Text Controls aus. Sie sehen, dass durchaus individuell notwendigen Gestaltungsfunktionen dem Anwender zur Verfügung gestellt werden können. Es steht natürlich frei, diese Funktionen auch über entsprechende Schaltfläche bzw. Menüitems in der Ribbon Bar zusätzlich abzubilden.



Kontextmenü im Rich Text Control

Function: TextbausteinÜbernehmen

Description: Übernehmen und Einfügen eines Textbausteins an die Cursorposition des RTF-Controls

Returns

Parameters

Static Variables

Local variables

Sql Handle: hSql

String: sStatement

Long String: lsElement

Long String: lsAltesElement

Number: nReturn

Number: nPuffer

Actions

```
Set sStatement = 'SELECT ELEMENT FROM BAUSTEIN WHERE NR = :gridAuswahl.colnNr INTO :lsElement'
```

```
Call SqlConnection( hSql )
```

```
Call SqlPrepareAndExecute( hSql, sStatement )
```

```
Call SqlSetLongBindDatatype( Bindvariable.Eins, Datentyp.Binär )
```

```
Call SqlFetchNext ( hSql, nReturn )
```

```
Call SqlDisconnect( hSql )
```

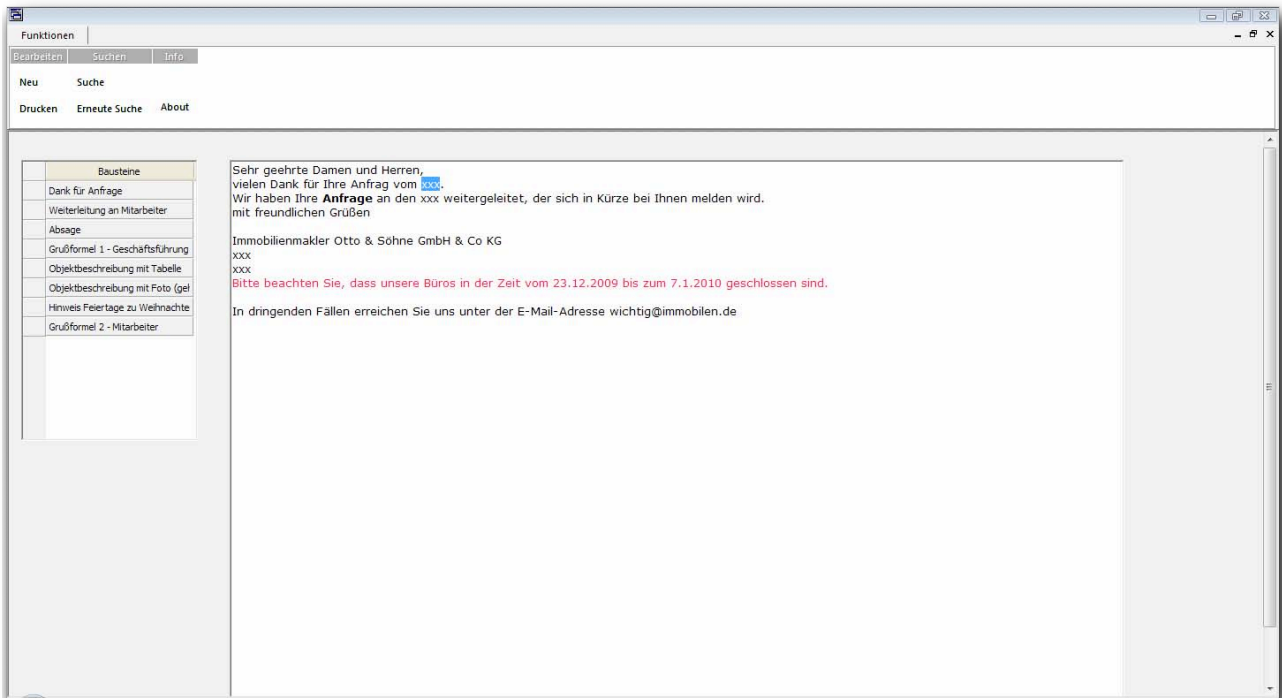
```
Call SalRTFTextPaste( rtc1, lsElement )
```

Die Funktion TextbausteinÜbernehmen

Das Auslesen des Rich Text-Inhalts aus der Datenbank ist nunmehr bekannt. In der letzten Zeile wird der lediglich der Inhalt der Variablen lsElement an der aktuellen Cursorposition in das Rich Text Control eingefügt.

Sonderzeichen suchen und manuell ersetzen

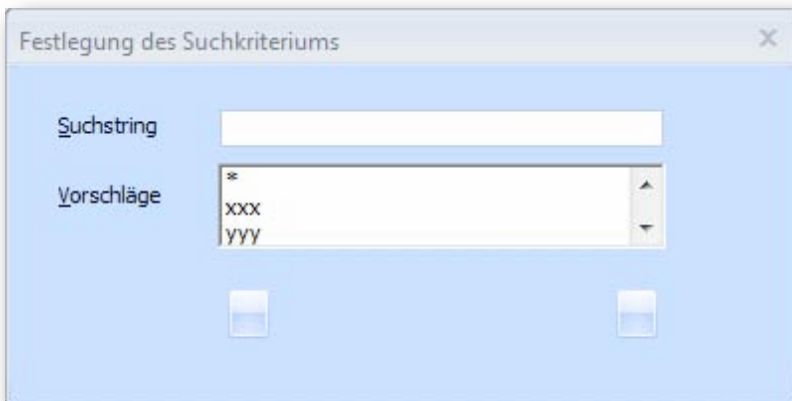
Im gegebenen Beispiel wurde die Zeichenfolge „xxx“ als Platzhalter definiert. In diesem Abschnitt geht es darum, diese Zeichenfolge in dem angelegten Text zu finden und zu markieren. Hierdurch hat der Anwender eine einfache Funktion, um Platzhalter manuell zu ersetzen.



Suchen nach Platzhaltern im Text und Markieren

Die Abbildung zeigt ein Rich Text Control, in dem vier Mal die Zeichenfolge „xxx“ (aus den vorhandenen Textbausteinen) enthalten ist. Es geht nun darum, eine Funktion zu implementieren, mit der der Anwender in einem Text die entsprechenden Platzhalter findet und markiert, damit die Platzhalter nur noch überschrieben werden müssen.

Durch Anklicken des Menüpunkts Suchen in der Ribbon Bar wird der Dialog zur Festlegung des Suchkriteriums aufgerufen.



Dialog zur Festlegung des Suchkriteriums

Um die Eingabe eines Suchkriteriums - sprich Platzhalter im Text - zu vereinfachen, wurde im Dialog noch eine Listbox mit möglichen Vorschlägen für Platzhalter implementiert. Ein Doppelklick auf einen Eintrag in die Listbox bewirkt, dass das Suchkriterium festgelegt, der Dialog geschlossen und die Suche nach dem ersten Platzhalter im Rich Text Control ausgeführt wird. Wenn ein Treffer gefunden wurde, wird er in seiner gesamten Länge markiert. Der Anwender kann den Platzhalter einfach überschreiben.

Die nachstehende Funktion Suchen führt die Suche nach dem ersten Treffer im Rich Text Control durch.

Function: Suchen

Description:

Returns

Parameters

String: sSuchstring

Static Variables

Local variables

Number: nPosition

Number: nLänge

Actions

Call SalRTFTextSelect(rtc1, 0, 0)

Set nLänge = SalStrLength(sSuchstring)

Set nPosition = SalRTFTextFind(rtc1 , sSuchstring , 0)

Call SalRTFTextSelect(rtc1, nPosition, nPosition + nLänge)

Die Funktion Suchen

Zunächst wird automatisch an den Anfang des Dokuments positioniert, um danach mit der Funktion SalRTFTextFind im vorhandenen Text nach dem Platzhalter zu suchen. Wenn die Suche erfolgreich war, wird der gefundene Platzhalter-Text markiert, sodass er sofort durch den Anwender überschrieben werden kann.

Function: Weitersuchen

Description:

Returns

Parameters

Static Variables

Local variables

Actions

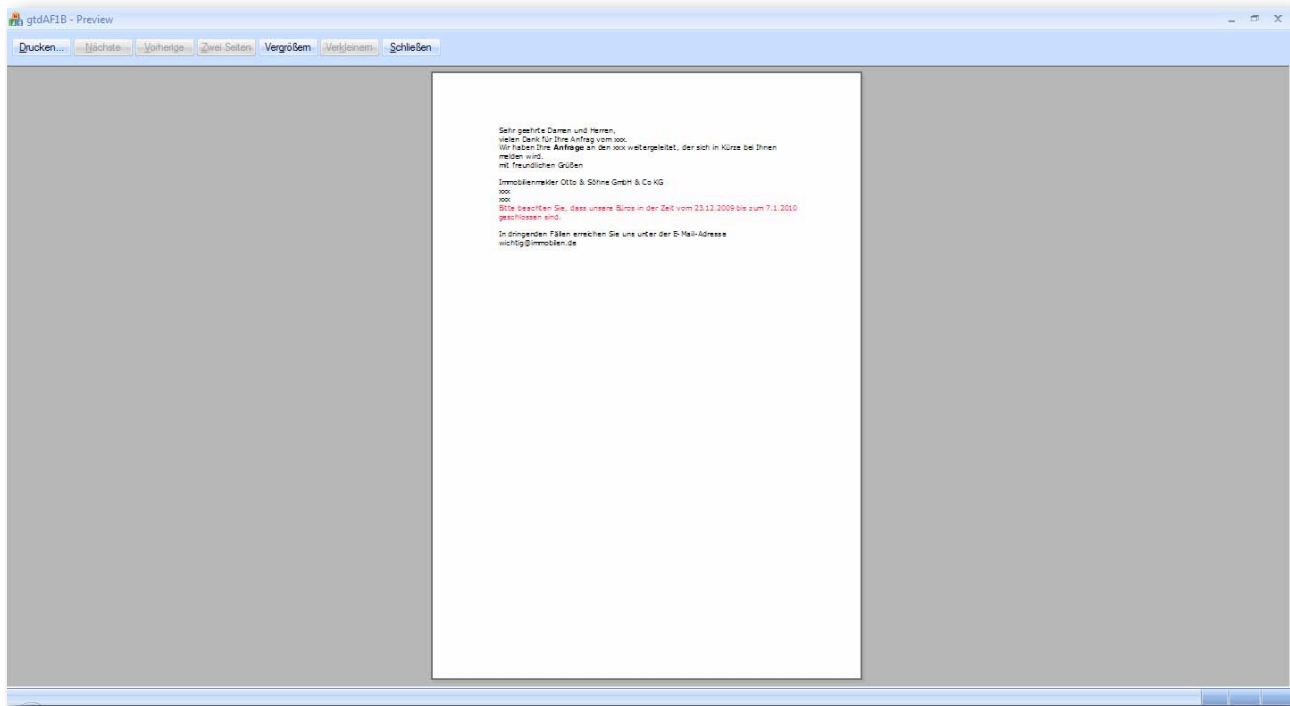
Call SalRTFTextFindNext(rtc1)

Die Funktion Weitersuchen

Die Funktion SalRTFTextFindNext bietet die komfortable Möglichkeit, ab der letzten Position im Rich Text Control nach der bisherigen Vorgabe weiter zu suchen. Die Kombination von SalRTFTextFind und SalRTFTextFindNext bietet somit eine einfache Möglichkeit, einen bestehenden Text nach vorgegeben Kriterien zu durchsuchen.

Fertige Texte ansehen und drucken

Ein auf die beschriebene Art und Weise erstelltes Dokument soll gedruckt werden. Durch Anklicken des Menüeintrags Drucken in der Ribbon Bar wird die im Rich Text Control integrierte Funktion für die Vorschau des Dokuments aufgerufen. Die Vorschau auf das Dokument wird angezeigt.



Vorschau auf das mit dem Rich Text Control erstellte Dokument

Die im Control integrierte Druckvorschau ermöglicht es, das Dokument zu vergrößern und zu verkleinern und das Dokument auf den Standarddrucker auszugeben. Durch Anklicken der Schaltfläche Schließen wird die Druckvorschau wieder geschlossen und zur Beispielanwendung zurück geschaltet.

Nachfolgend wird der Code für die Anzeige der Druckvorschau dargestellt.

Function: Drucken

Description:

Returns

Parameters

Static Variables

Local variables

Actions

Call SalRTFDocumentPreview(rtc1, Preview.Dokument)

Die Funktion Drucken

Zusammenfassung

Das Rich Text Control im Team Developer 5.2 bietet eine komfortable Möglichkeit, die Erfassung, Bearbeitung, Speicherung und Ausgabe von formatierten Texten in SQLWindows-Anwendungen vorzunehmen.

Für die programmtechnische Verarbeitung gibt es eine Vielzahl von Funktionen, die durch ihren Namen eine entsprechende Gliederung vorgeben:

- ▶ SalRTFDocument ... - Funktionen für die Bearbeitung des Dokuments
- ▶ SalRTFParagraph... - Funktionen für die Bearbeitung von Absätzen
- ▶ SalRFTText... - Funktionen für die Bearbeitung von Zeichen(folgen)

Da das Rich Text Control auch in der Version 5.2 von Report Builder gibt, können Rich Text formatierte Texte nunmehr auch in QRP-Dateien hinterlegt und damit die Ausgabe (auf Drucker oder in PDF-Dateien) in bestehende Anwendungen integriert werden.