DLL verwendet BTree-Tabellen[1] mit den Spalten «ID» und «Text», welche mit MaintainText4Form.exe gepflegt werden.

1. Class Tool4Form

```
Tool4Form :: Public Class Tool4Form
| enc :: Public Shared ReadOnly enc As System.Text.Encoding = System.Text.Encoding.UTF8
| exe4pdf :: Public ReadOnly exe4pdf As String = ""
| pdf :: Public ReadOnly pdf As String = ""
| F1 :: Public Sub F1(ByRef frm As System.Windows.Forms.Form, Optional name As String = "")
| GetText :: Public Function GetText(id As String, Optional notext As String = "") As String
| New :: Public Sub New(StartupPath As String, ProductName As String)
| Start4Process ::  Public Sub Start4Process(ByRef frm As System.Windows.Forms.Form, start As String, Optional param As String= "")
| Text2Form :: Public Sub Text2Form(ByRef frm As System.Windows.Forms.Form, ByRef tip As System.Windows.Forms.ToolTip)
```

2. BTree-Tabellen

Die Namensgebung für die Tabellen setzen sich aus «Produktname» und «Sprachcode», getrennt durch einen Punkt sowie der Dateiendung «.tab» zusammen. Für Sprachcode realisiert sind: «e», «f», «g», «i». Der Name für ein Produkt «Test.exe» lautet somit für Deutsch: «Test.g.tab». Der Name für die Hilfe[2] dazu lautet: «Test.g.pdf».

[1] BTree gelangt erst DLL-Intern zur Anwendung.
[2] Ist fakultativ, gelangt aber nur zur Anwendung, wenn die Sprachcode beider Dateien übereinstimmen.

Standard für Tabelleninhalt[3]

ID[4]	Text
#	Überschrift für Fenster# ^Hilfe[5]
#.Name	Text für Name in Fenster# ^ToolTip[6]^Hilfe[5]
beliebig	Freier Inhalt für Anwendungsprogramm

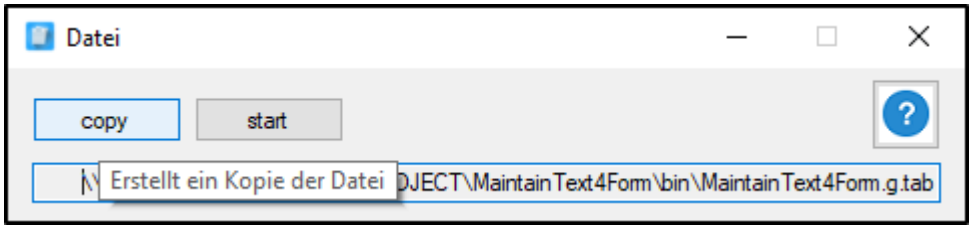
PDF-Dokument

Das Programm zur Anzeige des PDF-Dokuments wird automatisch ermittelt und ist danach im Feld «exe4pdf» gespeichert. Prioritär ist es das dem PDF zugeordnete, sofern es sich um «chrome», «msedge» oder «acrord32» handelt[7].

3. Anwendungsbeispiel 1

Nachfolgend wird die Tabelle der Anwendung «MaintainText4Form» verwendet. Im Programmcode auf Seite 4 sind nur notwendige, bzw. gewünschte[8] Elemente aufgeführt und mit → markiert.

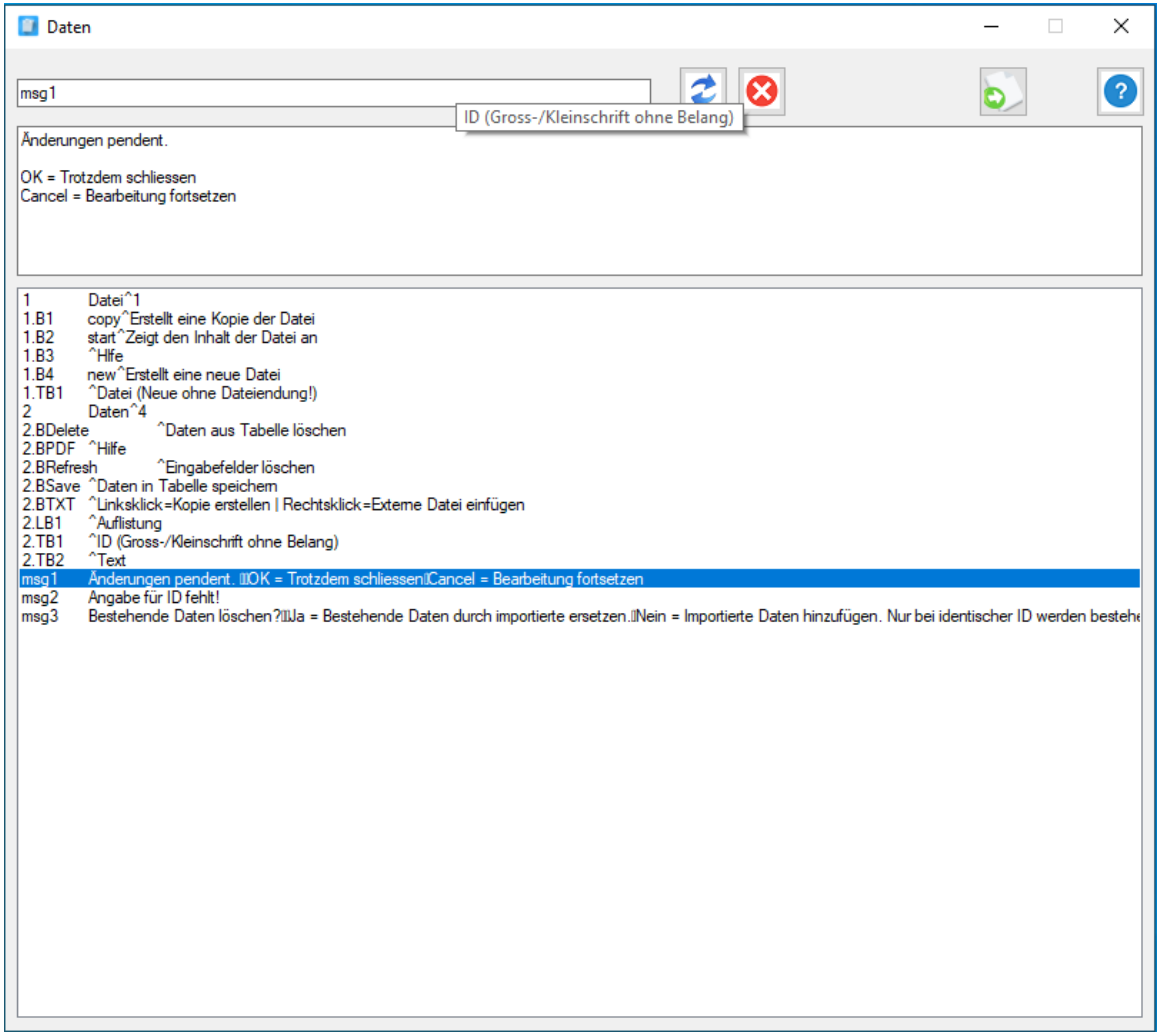
3.1 BTree-Tabelle für Sprache «g»



Texte werden aus der aufgeführten Datei geladen. Bleibt der Cursor auf einem Objekt stehen, wird der ToolTip angezeigt (hier ist es auf [copy] der Fall).

[3] Zeilenumbrüche sind intern durch Chr(1) repräsentiert.
[4] # für Nummer des Fensters (letzte numerische Stellen im Namen des Fensters). Gross-/Kleinschrift ohne Bedeutung.
[5] Hilfe fakultativ. Mit Angabe der Seitennummer zeigt [F1] direkt diese an.
[6] ^ToolTip fakultativ
[7] Ansonsten wird es, wegen der Parametrisierung, eines der Dreien (Reihenfolge wie aufgeführt).
[8] Nur die Texte für die MsgBox sind gewünschte Elemente.

3.2 BTree-Tabelle für Sprache «g» angewendet



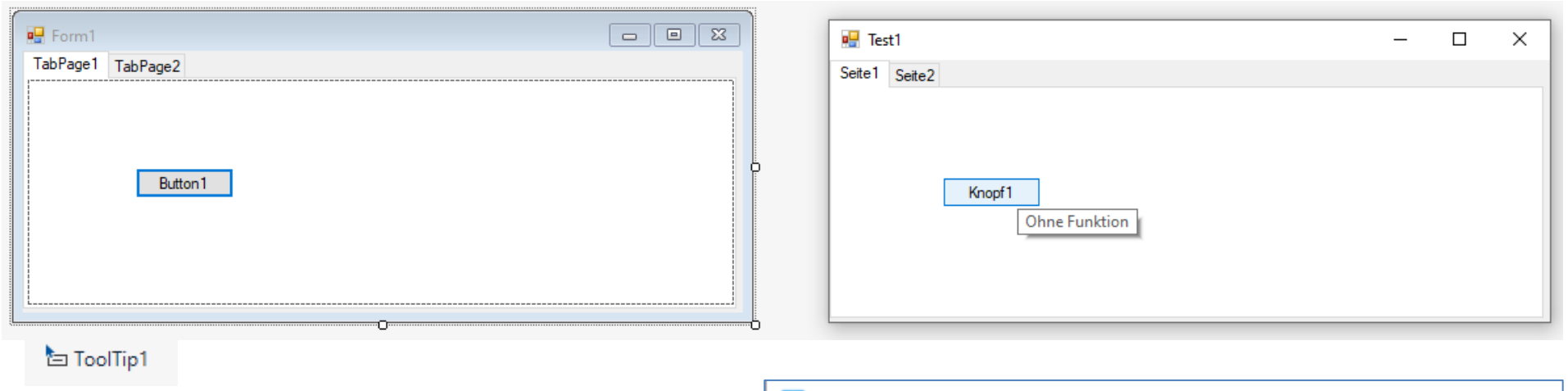
Texte sind aus der Datei gemäss Punkt 3.1 geladen. Bleibt der Cursor auf einem Objekt stehen, wird der ToolTip angezeigt.

3.3 Programmcode

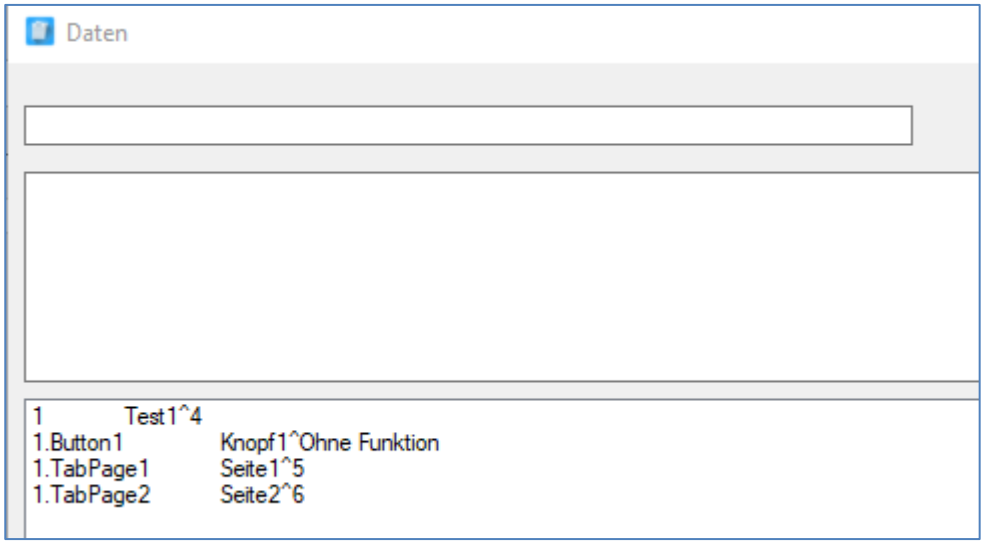
```
Public Class Form1
→       Public t4f As Tool4Form.Tool4Form
        Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
→               t4f = New Tool4Form.Tool4Form(Application.StartupPath, Application.ProductName)
→               t4f.Text2Form(Me, ToolTip1)
        End Sub
        Private Sub PDF_Click(sender As Object, e As EventArgs) Handles B3.Click
→               t4f.Start4Process(Me, t4f.pdf)
         End Sub
        Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles Me.KeyDown
→               If e.KeyValue = Keys.F1 Then t4f.F1(Me)
         End Sub
End Class


Public Class Form2
        Private Sub Form2_Load(sender As Object, e As EventArgs) Handles Me.Load
→               Form1.t4f.Text2Form(Me, ToolTip2)
        End Sub
        Private Sub PDF_Click(sender As Object, e As EventArgs) Handles BPDF.Click
→               Form1.t4f.Start4Process(Me, Form1.t4f.pdf)
        End Sub
        Private Sub Form2_KeyDown(sender As Object, e As KeyEventArgs) Handles Me.KeyDown
→               If e.KeyValue = Keys.F1 Then Form1.t4f.F1(Me)
        End Sub
        Private Sub BSave_Click(sender As Object, e As EventArgs) Handles BSave.Click
                 TB1.Text = TB1.Text.Trim
                If TB1.Text.Length = 0 Then
→                       MsgBox(Form1.t4f.GetText("msg2", "Angabe für Key fehlt"))
                        TB1.Focus()
                        Exit Sub
                End If
```

```
        End Sub
End Class
```

4. Anwendungsbeispiel 2



Das Beispiel zeigt eine Testsituation mit Verwendung von «TabControl Class». In solchen Fällen ist es wahrscheinlich sinnvoll, die [F1]-Taste der aktiven «TabPage» zuzuweisen. Falls eine «TabPage» keine Seitennummer hat, wird automatisch jene von «Form1» massgebend.



4.1 Programmcode

```
Public Class Form1
→       Public t4f As Tool4Form.Tool4Form

        Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
→           t4f = New Tool4Form.Tool4Form(Application.StartupPath, Application.ProductName)
→           t4f.Text2Form(Me, ToolTip1)
    End Sub

    Private Sub TabControl1_KeyDown(sender As Object, e As KeyEventArgs) Handles TabControl1.KeyDown
            Dim tc As TabControl = CType(sender, TabControl)
→           If e.KeyValue = Keys.F1 Then t4f.F1(Me, tc.Controls(tc.SelectedIndex).Name)
    End Sub
End Class
```