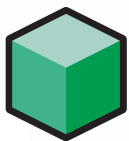




SPRACHREFERENZ 5.5



REALbasic®

Copyright 2004 REAL Software, Inc. und Application Systems Heidelberg

Alle Rechte vorbehalten. Jede auch auszugsweise Vervielfältigung der Dokumentation oder des REALbasic-Softwareprogramms wird strafrechtlich verfolgt. Die Übertragung des Softwareprogramms auf Datenträger jeglicher Art zu einem anderen Zwecke als dem der Datensicherung ist nicht gestattet. Die Rechte am Softwareprogramm REALbasic und an der Dokumentation liegen bei Application Systems Heidelberg und REAL Software, Inc.

Der rechtmäßige Erwerb des Handbuchs und der Original-CD erlaubt die Nutzung des Programms analog der Benutzung eines Buchs. Entsprechend der Unmöglichkeit, daß ein Buch zugleich an verschiedenen Orten von mehreren Personen gelesen wird, darf das Softwareprogramm REALbasic nicht gleichzeitig von verschiedenen Personen an verschiedenen Orten benutzt werden.

Einschränkung der Gewährleistung

Inhaltliche Änderungen des Handbuchs und des Softwareprogramms behalten wir uns ohne Ankündigung vor. Es wird keine Haftung für die Richtigkeit des Inhalts des Handbuchs oder Schäden, die sich aus dem Gebrauch des Softwareprogramms ergeben, übernommen. Für Hinweise auf Fehler sind wir jederzeit dankbar.

Warenzeichen

Innerhalb dieses Handbuchs wird auf Warenzeichen Bezug genommen, die nicht explizit als solche ausgewiesen sind. Aus dem Fehlen einer Kennzeichnung kann also nicht geschlossen werden, daß ein Name frei von den Rechten Dritter ist.

Apple, Macintosh, Power Macintosh und Mac OS sind geschützte Warenzeichen von Apple.

REALbasic Sprachreferenz

Deutsche Übersetzung von O. Buchmann, T. Hoffmann, V. Ritzhaupt, M. Taubert

Postanschrift: Application Systems Heidelberg Software GmbH
Pleikartsförsterhof 4/1
D – 69124 Heidelberg, Deutschland

Telefon: 06221 300002

Telefax: 06221 300389

Web Site: <http://www.application-systems.de/realbasic>
<http://www.realsoftware.com>

Support: realbasic@application-systems.de

Verkauf: sales@application-systems.de

Einführung

Diese Sprachreferenz bietet Ihnen schnellen Zugriff zu Informationen über alle Sprachelemente von REALbasic: Objekte, Schlüsselwörter, Operatoren, Konstanten, Fehlermeldungen, Laufzeitfehler, eingebaute Methoden und Funktionen und so weiter. Wo immer möglich werden die zu den Objekten gehörenden Eigenschaften und Methoden dokumentiert und mit Querverweisen zu zugehörigen Objekten versehen.

Aktualisierte Ausgaben dieser Sprachreferenz stellen wir in unregelmäßigen Abständen als PDF-Datei im Internet unter www.realsoftware.de zur Verfügung.

Es gibt darüberhinaus ein Tutorial, einen QuickStart-Guide und ein Entwicklerhandbuch, die Ihnen die Konzepte bei der Entwicklung von REALbasic-Programmen näherbringen.

Konventionen

Namen von Eigenschaften, die **fett** gedruckt sind, sind zur Laufzeit read-only. Das bedeutet, daß Ihr Programm diese Eigenschaften auslesen, aber nicht deren Werte ändern kann.

Optionale Parameter in den Ausdrücken sind in eckige Klammern eingeschlossen – [].

Ausdrücke, die blau unterlegt sind (in der PDF-Version der Sprachreferenz) sind Hyperlinks zu anderen Stellen des Dokuments.

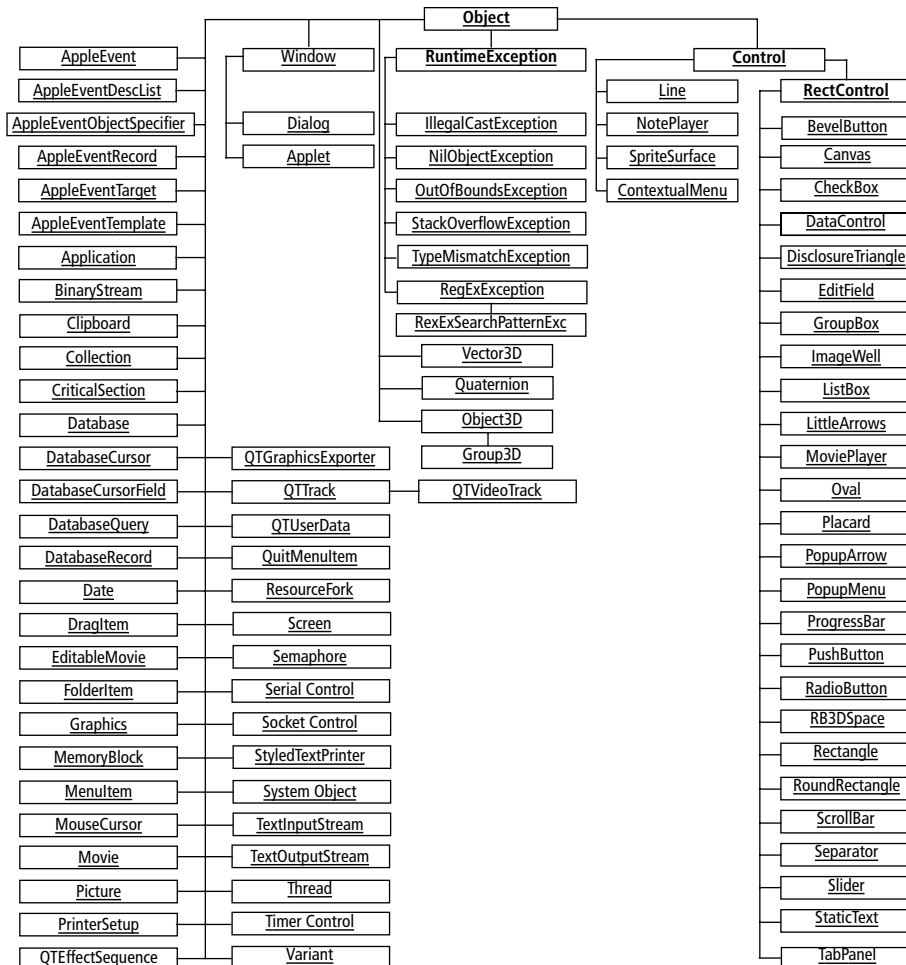
In einigen Fällen wurde Code, der für das Beispiel irrelevant ist, aus Gründen der Übersichtlichkeit weggelassen. Weggelassener Code wird durch Punkte angezeigt.

Die Klassen-Hierarchie

Objekte sind in REALbasic in einer Hierarchie angeordnet. Das bedeutet, daß ein Objekt die Eigenschaften, Methoden und Events seiner übergeordneten Klasse (den Klassen der Hierarchie darüber) erbt und eventuell Eigenschaften, Methoden und Events besitzt, die für dieses Objekt einzigartig sind.

Die folgende Abbildung zeigt die REALbasic-Klassenhierarchie. Ein PushButton ist beispielsweise eine Unterklasse eines RectControl, welches eine Unterklasse eines Controls, welches wiederum eine Unterklasse eines Objects ist (der Basisklasse aller Objekte).

Objektklassen, die in folgender Abbildung fett gedruckt sind, sind nicht direkt verwendbar. Sie können nur auf deren Unterklassen zugreifen.



Cross-Plattform-Entwicklung

Verwenden von Target-Flags

REALbasic ermöglicht es Ihnen (abhängig von der Version), aus ein und dem selben Projekt Programme für Mac OS, Windows und Linux zu compilieren. Mit Hilfe der folgenden globalen boole'schen Konstanten kann Ihr Programm feststellen, auf welcher Plattform gerade Code ausgeführt wird. So können Sie plattformspezifischen Code ausführen.

Boole'sche Konstante	Beschreibung
<u><i>DebugBuild</i></u>	Die Anwendung läuft innerhalb der REALbasic-Entwicklungsumgebung, Sie haben also Debug/Run aufgerufen.
<u><i>RBVersion</i></u>	Liefert die REALbasic-Versionsnummer des Compilers zurück.
<u><i>Target68K</i></u>	Die compilierte Anwendung läuft momentan mit 68K-Macintosh-Code auf einem 68K-Rechner oder auf einem PPC-Rechner im Emulationsmodus. Nur aus Kompatibilitätsgründen zu früheren Versionen von REALbasic vorhanden. REALbasic kann ab Version 4.0 keine 68K-Applikationen mehr erzeugen. Die letzte Version, die über einen 68K-Compiler verfügte, ist die Version 3.5.2.
<u><i>TargetCarbon</i></u>	Die compilierte Anwendung läuft momentan mit Carbon/Mac OS X Code.
<u><i>TargetHasGUI</i></u>	Die compilierte Anwendung hat eine grafische Benutzeroberfläche, ist also keine ConsoleApplication oder ServiceApplication.
<u><i>TargetLinux</i></u>	Die compilierte Anwendung läuft unter Linux.
<u><i>TargetMachO</i></u>	Die compilierte Anwendung läuft unter Mac OS X und verwendet das Objektcode-Format des Mac OS-Kernels.
<u><i>TargetMacOS</i></u>	Die compilierte Anwendung läuft momentan mit Macintosh-Code, PPC-Classic oder Mac OS X.
<u><i>TargetPPC</i></u>	Die compilierte Anwendung läuft momentan mit PPC-Macintosh-Code im Classic Mac OS.
<u><i>TargetWin32</i></u>	Die compilierte Anwendung läuft momentan auf einem Windows-Rechner.

Bedingte Compilierung

Wenn ein Teil Ihres Codes nur für eine bestimmte Plattform compiliert werden soll (beispielsweise AppleScript-spezifischer Code oder Toolbox-Aufrufe), können Sie den #If-Befehl verwenden:

```
#If TargetBoolean
    //OS-spezifischer Code
    //in der erzeugten Applikation
    //nur auf der Zielplattform
#Else //(optional)
    //(Optional) anderer Plattform-spezifischer Code
#Endif
```

Sprachreferenz

Es folgt eine alphabetisch sortierte Auflistung der REALbasic-Sprachelemente.

- Operator

Wird verwendet, um die Differenz zweier Zahlen zu berechnen.

Syntax

Ergebnis=Ausdruck1 - Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Zahl	Die Differenz von Ausdruck1 und Ausdruck2.
<i>Ausdruck1</i>	Zahl	Beliebiger numerischer Ausdruck.
<i>Ausdruck2</i>	Zahl	Beliebiger numerischer Ausdruck.

Beispiel

Dieses Beispiel speichert die Differenz zweier Zahlen in einer Variable:

```
Dim x as Integer
x=50-30 //x ergibt 20
```

#If ...#Else...#Endif Befehl

Wird zur Steuerung bedingter Compilierung verwendet.

Syntax

#If TargetBoolean [Then]

//OS-spezifischer Code

Teil	Typ	Beschreibung
<i>TargetBoolean</i>	<u>Boole'sche</u> Konstante	<u>TargetMacOS</u> , <u>TargetPPC</u> , <u>Target68K</u> , <u>TargetCarbon</u> , <u>TargetWin32</u> , oder <u>DebugBuild</u> Konstante, die das Betriebssystem festlegt, für welches der Code, der folgt, eingebunden werden soll.

[#Else]

//Für anderes OS spezifischer Code

[#ElseIf TargetBoolean]

// Spezifischer Code für ein weiteres OS

#Endif

Teil	Typ	Beschreibung
<i>TargetBoolean</i>	<u>Boole'sche</u> Konstante	Konstante, die das Betriebssystem festlegt, für welches der Code, der folgt, eingebunden werden soll. Normalerweise eine der folgenden Konstanten: <u>TargetMacOS</u> , <u>TargetPPC</u> , <u>TargetCarbon</u> , <u>TargetMachO</u> , <u>TargetWin32</u> , <u>TargetLinux</u> , <u>DebugBuild</u> oder <u>RBVersion</u> .

Hinweise:

Verwenden Sie bedingte Compilierung, um plattformspezifische Anweisungen wie Toolbox-Aufrufe, AppleEvent-Routinen oder Routinen für Konsolen-Applikationen für Mac OS X oder Linux, zu isolieren. Der Code, der auf den #If-Befehl folgt, wird nur in der Version für das zugehörige Betriebssystem eingebunden. Target68K, TargetPPC und TargetCarbon sind Untermengen von TargetMacOS.

Die optionale ElseIf-Anweisung gibt Ihnen die Möglichkeit, Codebausteine zu verwenden, die alle möglichen Fälle abdecken. Wie zum Beispiel der folgende Baustein:

```
#If TargetMacOS
// Mac OS-spezifischer Code
#ElseIf TargetWin32
// Windows-spezifischer Code
```

```
#ElseIf TargetLinux
// Linux-spezifischer Code
#endif
```

Der TargetBoolean-Parameter muß entweder eine Konstante sein oder ein Ausdruck, der sich nach True und False auflösen läßt. Sie können zum Beispiel die Konstante RBVersion oder RBVersionString benutzen, um herauszufinden, welche REALbasic-Version installiert ist.

Beispiel

Folgendes Beispiel weist der (benutzerdefinierten) Separator-Eigenschaft der App-Klasse in ihrem Open-Event-Handler Werte zu. Dies wird verwendet, um volle Pfadnamen festzulegen, die sowohl auf dem Macintosh als auch unter Windows korrekt sind.

```
#If TargetMacOS
    Separator=":"
#Endif
#If TargetWin32
    Separator="\\"
#Endif
#If TargetMacOS
    Separator = ":"
#ElseIf TargetWin32
    Separator = "\\"
#ElseIf TargetLinux
    Separator = "/"
#Endif
```

Siehe auch

[Declare](#) Befehl, [AppleEvent](#) Klasse; [RBVersion](#), [RBVersionString](#), [TargetMacOS](#), [TargetPPC](#), [Target68K](#), [TargetHasGui](#), [TargetMachO](#), [TargetLinux](#), [TargetCarbon](#), [TargetWin32](#), [DebugBuild](#) Konstanten.

_ Schlüsselwort

Der Unterstrich als Schlüsselwort wird verwendet, wenn eine Zeile Code in der nächsten Zeile fortgesetzt wird.

Syntax

_ (als letztes Zeichen in der Zeile)

Hinweise

Benutzen Sie den Unterstrich als letztes Zeichen in einer Zeile Code, wenn diese Zeile in der nächsten Zeile fortgesetzt werden soll. Der Compiler wird die Zeile nach dem Unterstrich als fortgesetzte Zeile interpretieren. Der Code-Editor wird die fortgesetzte Zeile automatisch einrücken, wie in dem Beispiel-Code gezeigt.

Möchten Sie einen langen String umbrechen und in der folgenden Zeile fortsetzen, so benutzen Sie den Unterstrich außerhalb der Anführungszeichen(siehe Beispiel).

Unter MacOS(X) können Sie zum Einfügen des Unterstrichs die Tastenkombination Auswahl+Return benutzen. Der Cursor wird dann automatisch in der nächsten Zeile eingerückt.

Beispiel

Das folgende Beispiel benutzt eine fortgeführte Zeile, um einen langen String in zwei Zeilen darzustellen. Der Unterstrich bewirkt, daß die folgende Zeile an die aktuelle Zeile angehängt wird, wenn sie interpretiert wird und das + verbindet die zwei Strings.

```
Dim SaveChangesDialog as New MessageDialog
SaveChangesDialog.Explanation="Hinweis: Wenn Sie nicht abspeichern, werden Sie " _
+"Ihre gesamte Arbeit seit dem letzten Abspeichern verlieren."
```

#Pragma Direktive

Werden verwendet, um die Ausführungsgeschwindigkeit des Codes zu beschleunigen, indem bestimmte automatische Tasks abgeschaltet werden. Wenn in einer Methode eine Pragma-Direktive verwendet wird, bleibt der automatische Task bis zum Ende der Methode abgeschaltet, in der er aufgerufen wurde (oder wird aktiviert, indem er erneut aufgerufen und True festgelegt wird). Dies gilt jedoch nicht für Methoden, die innerhalb dieser Methode aufgerufen werden.

Syntax

#pragma Directive

oder

#pragma Directive **True** oder **False**

Direktive	Beschreibung
<i>BackgroundTasks</i>	Aktiviert oder deaktiviert die automatische Rechenzeitabgabe an Hintergrund-Threads. Zusätzlich können Sie bei der Pragma-Direktive <u>True</u> oder <u>False</u> festlegen. Wenn Sie diese Direktive auf <u>False</u> setzen, ist das das Gleiche wie die Verwendung von DisableBackgroundTasks.
<i>BoundsChecking</i>	Schaltet die Überprüfung des Wertebereichs von Arrays ein oder aus. Zusätzlich zur Pragma-Direktive können Sie noch <u>True</u> oder <u>False</u> festlegen. Wenn Sie <u>False</u> verwenden, ist das das Gleiche wie die Verwendung von DisableBoundsChecking.
<i>DisableAutoWaitCursor</i>	Wird verwendet, um das automatische Anzeigen des Mauszeigers als Uhr oder drehendes Rad auszuschalten. Ab Version 5.0 ist der Wirkungsbereich nicht mehr global. Der Warten-Mauszeiger wird in der Methode deaktiviert, die das Pragma aufruft, bis diese zu Ende ist. Wenn z.B. DisableAutoWaitCursor in einem PushButton aufgerufen wird, der in einer Schleife läuft, wird der Warten-Mauszeiger nur so lange die Schleife läuft deaktiviert und bis das Action-Event vollständig ausgeführt wurde.
<i>DisableBackgroundTasks</i>	Schaltet die Verwaltung automatischer Hintergrundtasks für den Programmcode nach #pragma aus. Dies bleibt bis zum Ende der Methode ausgeschaltet. Es hindert REALbasic daran, die WaitNext-Event-Toolbox-Routine aufzurufen, so daß keine Zeit an andere Applikationen abgegeben wird. Dies kann dadurch sehr rechenintensive Operationen beschleunigen, hindert REALbasic aber daran, den Mauszeiger als Uhr darzustellen und kann das Neuzeichnen von Bedienelementen im Hintergrund oder die Ausführung anderer Threads verhindern.
<i>DisableBoundsChecking</i>	Schaltet die Überprüfung des Wertebereichs von Arrays für den Programmcode nach #pragma aus. Funktioniert nur bei eindimensionalen boole'schen Arrays.
<i>NilObjectChecking</i>	Steuert das Verhalten, ob Objekte automatisch auf <u>Nil</u> getestet werden sollen, bevor auf deren Eigenschaften zugegriffen wird oder deren Methoden aufgerufen werden. Zusätzlich zur Pragma-Direktive können Sie noch <u>True</u> oder <u>False</u> festlegen.
<i>StackOverflowChecking</i>	Steuert das Verhalten, ob auf Stapelüberläufe (Stack Overflows) getestet werden soll. Zusätzlich zur Pragma-Direktive können Sie noch <u>True</u> oder <u>False</u> festlegen.

Hinweise

Bei vier Pragma-Direktiven müssen Sie True oder False übergeben, um eine Option ein- oder auszuschalten. Da diese Pragmas ein- oder ausgeschaltet werden können, können Sie nun Dinge erledigen wie Hintergrund-Tasks innerhalb einer inneren Schleife deaktivieren, sie aber in der äußeren Schleife aktiv lassen:

```

For y = 0 To Height
  #pragma BackgroundTasks False
  For x = 0 To Width
    //...einige Pixel verarbeiten...
  Next
Next
#pragma BackgroundTasks True
Next

```

& Operator

Wird verwendet, um hexadezimale, oktale, binäre und Farb-Konstanten zu repräsentieren.

Syntax

&h als Präfix für Hexadezimalzahlen

&o als Präfix für Oktalzahlen

&b als Präfix für Binärzahlen

&cRRGGBB für Farben

Teil	Typ	Beschreibung
RR	String	Der Rotanteil der Farbe, hexadezimal von 00 bis FF
GG	String	Der Grünanteil der Farbe, hexadezimal von 00 bis FF
BB	String	Der Blauanteil der Farbe, hexadezimal von 00 bis FF

Hinweise

Der &c-Operator bietet eine alternative Möglichkeit, eine Farbe über ihre RGB-Anteile anzugeben. Seine Funktionalität entspricht der RGB-Funktion, mit dem Unterschied, daß Sie die Anteile hexadezimal(00-FF) angeben und nicht dezimal (0-255). Im Code-Editor werden die Farbanteile entsprechend ihrer Grundfarbe angezeigt. Farben können auch in HSV oder CMY angegeben werden.

Ein einfacher Weg, eine gewünschte Farbe zu bestimmen, ist es, in der Farbpalette einen leeren Eintrag doppelzuklicken und in der Farbauswahl die gewünschte Farbe diesem Eintrag zuzuweisen. Jetzt können Sie diesen Eintrag in den Code-Editor ziehen und ihn dort im Code fallen lassen, wo Sie die Farbe benötigen. Sie können den Eintrag in der Farbpalette auch im Eigenschaften-Fenster, in einer Farbeigenschaft eines Elements, fallen lassen, um diese zuzuweisen. Sie ziehen mit diesem Eintrag einmal den echten Farbwert und zusätzlich eine hexadezimale Repräsentation der Farbe. Sie können die hexadezimale Repräsentation in jedem Texteditor fallen lassen, der Drag&Drop unterstützt.

Beispiele

Dieses Beispiel erlaubt einem Benutzer, die Farbe zu bestimmen, mit der das Objekt **Rectangle1** gefüllt wird.

```
Dim c as Color
Dim b as Boolean
c=&cFFFFFF //setzt die Standardfarbe, die in der Farbauswahl angezeigt wird
b=SelectColor(c,"Select a Color")
Rectangle1.FillColor=c
```

Dieses Beispiel zeigt eine Hexadezimal-Konstante

&hA4

Dieses Beispiel zeigt eine Oktal-Konstante:

&o527

Dieses Beispiel zeigt eine Binär-Konstante:

&b10010101

Siehe auch

[Bin.](#), [Oct.](#), [Hex.](#), [RGB.](#), [HSV.](#), [CMY](#) Funktionen; [Color](#) Klasse.

' Operator

Wird verwendet, um dem Code Kommentare beizufügen.

Syntax

' Diese Zeile berechnet die Mehrwertsteuer.

Hinweis

REALbasic ignoriert alle Kommentare, die Sie nach dem Befehl ' eingeben. Kommentare werden auch nicht in die fertige Anwendung übernommen.

Sie können auch Rem oder zwei Schrägstriche (//) verwenden, um ihren Code zu kommentieren. Kommentare können in einer separaten Zeile oder auch auf der gleichen Zeile wie Ihr ausführbarer Code stehen, solange sie rechts vom ausführbaren Code stehen. Gültige Kommentare erscheinen im Code-Editor in rot.

Der Tastatur-Shortcut Befehl-' (Befehl-Shift-#) schaltet die selektierten Code-Zeilen zwischen Kommentar- und Normal-Status hin- und her. Wenn die Einfügemarke irgendwo in der Mitte der Zeile ist, wenn Sie Befehl-' drücken, wird die gesamte Zeile auskommentiert. Wenn Sie nur den Text rechts von der Einfügemarke auskommentieren möchten, drücken Sie ' oder verwenden Sie Rem oder //.

Beispiel

Dieses Beispiel verwendet Kommentare, um die Ergebnisse von boole'schen Vergleichen zu dokumentieren:

```
Dim a As Boolean
Dim b As Boolean
Dim c As Boolean
Dim d As Boolean
a=True
b=False
d=a Or b ' Ergibt True
d=b And c ' Ergibt False
```

Siehe auch

Rem Befehl, // Operator.

* Operator

Wird verwendet, um zwei Zahlen zu multiplizieren.

Syntax

Ergebnis=Ausdruck1 * Ausdruck2

Teil	Art	Beschreibung
Ergebnis	Zahl	Das Produkt aus Ausdruck1 und Ausdruck2.
Ausdruck1	Zahl	Beliebiger numerischer Ausdruck.
Ausdruck2	Zahl	Beliebiger numerischer Ausdruck.

Beispiele

Dieses Beispiel speichert das Produkt zweier Zahlen in einer Variablen:

```
Dim x as Integer
x=50*30 //x ergibt 1500
```

+ Operator

Wird verwendet, um zwei Zahlen zu addieren oder zwei Strings (Zeichenketten) aneinanderzuhängen.

Syntax

Der + Operator hat zwei mögliche Syntax-Formen:

Ergebnis=Ausdruck1 + Ausdruck2

Teil	Art	Beschreibung
Ergebnis	Zahl	Die Summe von Ausdruck 1 und Ausdruck2.
Ausdruck1	Zahl	Beliebiger numerischer Ausdruck.
Ausdruck2	Zahl	Beliebiger numerischer Ausdruck.

Ergebnis=Ausdruck1 + Ausdruck2

Teil	Art	Beschreibung
Ergebnis	String	Ausdruck2 wird an Ausdruck1 angehängt.
Ausdruck1	String	Beliebiger String-Ausdruck.
Ausdruck2	String	Beliebiger String-Ausdruck.

Beispiele

Dieses Beispiel addiert mehrere Zahlen:

```
Dim x as Integer
x=1+2+3
```

Dieses Beispiel hängt zwei Strings aneinander und speichert das Ergebnis in einer Variablen:

```
Dim Name as String
Name="REAL"+"basic" //Name = REALbasic
```

Siehe auch

Str Funktion.

/ Operator

Wird verwendet, um eine Fließkomma-Division zweier Zahlen durchzuführen.

Syntax

Ergebnis = Ausdruck1 / Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Zahl	Die Division von Ausdruck1 und Ausdruck2.
<i>Ausdruck1</i>	Zahl	Beliebiger numerischer Ausdruck.
<i>Ausdruck2</i>	Zahl	Beliebiger numerischer Ausdruck.

Hinweis

Verwenden Sie diesen Operator, wenn Sie eine Division benötigen, die den Nachkommateil berücksichtigt.

Beispiel

Dieses Beispiel speichert die Division zweier Zahlen in einer Variablen:

```
Dim x As Double  
x=50.56/30.34 //x ergibt 1.6664469
```

// Operator

Wird verwendet, um dem Code Kommentare beizufügen.

Syntax

// Diese Zeile berechnet die Mehrwertsteuer.

Hinweis

REALbasic ignoriert alle Kommentare, die Sie hinter dem Operator // eingeben. Kommentare werden auch nicht in die fertige Anwendung übernommen.

Sie können auch Rem oder den Apostroph (') verwenden, um ihren Code zu kommentieren. Kommentare können in einer separaten Zeile oder auch auf der gleichen Zeile wie Ihr ausführbarer Code stehen, solange sie rechts vom ausführbaren Code stehen. Gültige Kommentare erscheinen im Code-Editor in rot.

Der Tastatur-Shortcut Befehl-' schaltet die selektierten Code-Zeilen zwischen Kommentar- und Normal-Status hin- und her. Wenn die Einfügemarke irgendwo in der Mitte der Zeile ist, wenn Sie Befehl-' drücken, wird die gesamte Zeile auskommentiert. Wenn Sie nur den Text rechts von der Einfügemarke auskommentieren möchten, drücken Sie ' oder verwenden Sie Rem oder //.

Beispiel

Dieses Beispiel verwendet Kommentare, um die Ergebnisse von boole'schen Vergleichen zu dokumentieren:

```
Dim a As Boolean  
Dim b As Boolean  
Dim c As Boolean  
Dim d As Boolean  
a=True  
b=False  
d=a Or b // Ergibt True  
d=b And c // Ergibt False
```

Siehe auch

Rem Befehl, ' Operator.

< Operator

Wird verwendet, um festzustellen, ob ein alphabetischer oder quantitativer Ausdruck kleiner als ein anderer ist. String-Vergleiche sind case-insensitiv.

Syntax

Ergebnis=Ausdruck1 < Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Boolean	Ergibt <u>True</u> , wenn Ausdruck 1 kleiner Ausdruck 2 ist.
<i>Ausdruck1</i>	<u>String</u> , Zahl oder <u>Datum</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.
<i>Ausdruck2</i>	<u>String</u> , Zahl oder <u>Datum</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.

Hinweis

Verwenden Sie StrComp, um case-sensitive String-Vergleiche durchzuführen.

Beispiel

Dieses Beispiel verwendet den <-Operator, um a und b zu vergleichen.

```
Dim a,b as Integer
If Editfield1.text <> "" and EditField2.text <> "" then
a=Val(Editfield1.text)
b=Val(EditField2.text)
If a<b then
MsgBox "A ist kleiner als B!"
else
Beep
end if
else
MsgBox "Bitte geben Sie in beide Kästchen Zahlen ein!"
end if
```

Siehe auch

> ,>= ,<= ,= , <> und StrComp Operatoren.

<= Operator

Wird verwendet, um festzustellen, ob ein alphabetischer oder quantitativer Ausdruck kleiner oder gleich einem anderen ist. String-Vergleiche sind case-insensitiv.

Syntax

Ergebnis=Ausdruck1 <= Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Boolean	Ergibt <u>True</u> , wenn Ausdruck 1 kleiner oder gleich Ausdruck 2 ist.
<i>Ausdruck1</i>	<u>String</u> , Zahl oder <u>Datum</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.
<i>Ausdruck2</i>	<u>String</u> , Zahl oder <u>Datum</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.

Hinweis

Verwenden Sie StrComp, um case-sensitive String-Vergleiche durchzuführen.

Beispiel

Dieses Beispiel verwendet den <= Operator, um a und b zu vergleichen.

```
Dim a,b as Integer
If Editfield1.text <> "" and EditField2.text <> "" then
a=Val(Editfield1.text)
b=Val(EditField2.text)
If a<=b then
MsgBox "A ist kleiner oder gleich B!"
else
```



```

        Beep
    end if
else
    MsgBox "Bitte geben Sie in beide Kästchen Zahlen ein!"
end if

```

Siehe auch

>, >=, <, <=, <> und [StrComp](#) Operatoren.

<> Operator

Wird verwendet, um festzustellen, ob ein Ausdruck ungleich einem anderen ist. [String](#)-Vergleiche sind case-insensitiv.

Syntax

Ergebnis=Ausdruck1 <> Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Boolean	Ergibt True , wenn Ausdruck 1 ungleich Ausdruck 2 ist.
<i>Ausdruck1</i>	String , Zahl , Object oder Datum	Ein Objekt oder beliebiger alphabetischer oder quantitativer Ausdruck.
<i>Ausdruck2</i>	String , Zahl , Object oder Datum	Ein Objekt oder beliebiger alphabetischer oder quantitativer Ausdruck.

Hinweise

Die Datentypen von **Ausdruck1** und **Ausdruck2** müssen zueinander passen. Der <> Operator wird außerdem bei Objekten verwendet, um zu testen, ob diese [Nil](#) sind. Wenn Sie zum Beispiel die NewMemoryBlock-Funktion verwenden, um ein neues Objekt zu erzeugen, sollten Sie überprüfen, ob das neue Objekt Nil ist, bevor Sie fortfahren.

Verwenden Sie [StrComp](#), um case-sensitive String-Vergleiche durchzuführen.

Beispiele

Dieses Beispiel verwendet den <> Operator, um auf leere EditFields zu überprüfen.

```

Dim a,b as Integer
If Editfield1.text <> "" and EditField2.text <> "" then
a=Val(Editfield1.text)
b=Val(EditField2.text)
If a<b then
    MsgBox "A ist kleiner als B!"
else
    Beep
end if
else
    MsgBox "Bitte geben Sie in beide Kästchen Zahlen ein!"
end if

```

Folgendes Beispiel überprüft, ob ein FolderItem Nil ist, bevor es dieses der Movie-Eigenschaft eines MoviePlayer-Steuerelements zuweist.

```

Dim f As FolderItem
f=GetOpenFolderItem("video/quicktime")
If f<>Nil then
    MediaPlayer1.movie=f.OpenAsMovie
End if

```

Siehe auch

>, >=, <, <=, = und [StrComp](#) Operatoren.

= Operator

Wird verwendet, um festzustellen, ob ein Ausdruck gleich einem anderen ist, oder um einer Variable oder Eigenschaft einen Wert zuzuweisen. [String](#)-Vergleiche sind case-insensitiv.

Syntax

Ergebnis=Ausdruck1 = Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Boolean</u>	Ergibt <u>True</u> , wenn Ausdruck 1 gleich Ausdruck 2 ist.
<i>Ausdruck1</i>	<u>String</u> , Zahl, <u>Boolean</u> oder <u>Color</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.
<i>Ausdruck2</i>	<u>String</u> , Zahl, <u>Boolean</u> oder <u>Color</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.

Ergebnis=Wert

Teil	Typ	Beschreibung
Ergebnis	Beliebig	Die Eigenschaft, Variable, Array oder Array-Element mit zugewiesenem Wert.
Wert	Beliebig	Der Wert, der dem Ergebnis zugewiesen wird.

Hinweise

Die Datentypen von **Ausdruck1** und **Ausdruck2** müssen zueinander passen.

Verwenden Sie StrComp, um case-sensitive String-Vergleiche durchzuführen.

Beispiel

Folgendes Beispiel überprüft, ob zwei Strings gleich sind.

```
Dim a,b as String
If Editfield1.text <> "" and EditField2.text <> "" then
a=Editfield1.text
b=EditField2.text
If a=b then
MsgBox "A ist gleich B!"
else
Beep
end if
else
MsgBox "Bitte geben Sie in beide Kästchen etwas ein!"
end if
```

Siehe auch

>, >=, <, <=, <> und StrComp Operatoren.

> Operator

Wird verwendet, um festzustellen, ob ein alphabetischer oder quantitativer Ausdruck größer als ein anderer ist. String-Vergleiche sind case-insensitiv.

Syntax

Ergebnis=Ausdruck1 > Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Boolean</u>	Ergibt <u>True</u> , wenn Ausdruck 1 größer als Ausdruck 2 ist.
<i>Ausdruck1</i>	<u>String</u> , Zahl oder <u>Datum</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.
<i>Ausdruck2</i>	<u>String</u> , Zahl oder <u>Datum</u>	Beliebiger alphabetischer oder quantitativer Ausdruck.

Hinweise

Die Datentypen von **Ausdruck1** und **Ausdruck2** müssen zueinander passen.

Verwenden Sie StrComp, um case-sensitive String-Vergleiche durchzuführen.

Beispiel

Folgendes Beispiel überprüft, ob eine Zahl größer als die andere ist.

```
Dim a,b as Integer
If Editfield1.text <> "" and EditField2.text <> "" then
a=Val(Editfield1.text)
b=Val(EditField2.text)
```

```

If a>b then
    MsgBox "A ist größer als B!"
else
    Beep
end if
else
    MsgBox "Bitte geben Sie in beide Kästchen Zahlen ein!"
end if

```

Siehe auch

>=, <=, <, =, <> und [StrComp](#) Operatoren.

>= Operator

Wird verwendet, um festzustellen, ob ein alphabetischer oder quantitativer Ausdruck größer oder gleich einem anderen ist. [String](#)-Vergleiche sind case-insensitiv.

Syntax

Ergebnis = Ausdruck1 >= Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Boolean	Ergibt True , wenn Ausdruck 1 größer oder gleich Ausdruck 2 ist.
<i>Ausdruck1</i>	String , Zahl oder Datum	Beliebiger alphabetischer oder quantitativer Ausdruck.
<i>Ausdruck2</i>	String , Zahl oder Datum	Beliebiger alphabetischer oder quantitativer Ausdruck.

Hinweise

Die Datentypen von **Ausdruck1** und **Ausdruck2** müssen zueinander passen.

Verwenden Sie [StrComp](#), um case-sensitive String-Vergleiche durchzuführen.

Beispiel

Folgendes Beispiel überprüft, ob eine Zahl größer oder gleich einer anderen ist.

```

Dim a,b as Integer
If EditField1.text <> "" and EditField2.text <> "" then
    a=Val(EditField1.text)
    b=Val(EditField2.text)
    If a>=b then
        MsgBox "A ist größer oder gleich B!"
    else
        Beep
    end if
else
    MsgBox "Bitte geben Sie in beide Kästchen Zahlen ein!"
end if

```

Siehe auch

>, <=, <, =, <> und [StrComp](#) Operatoren.

\ Operator

Wird verwendet, um eine [Integer](#)-Division zweier Zahlen vorzunehmen.

Syntax

Ergebnis = Ausdruck1 \ Ausdruck2

Teil	Art	Beschreibung
<i>Ergebnis</i>	Zahl	Die Integer -Division aus Ausdruck1 und Ausdruck2.
<i>Ausdruck1</i>	Zahl	Beliebiger numerischer Ausdruck.
<i>Ausdruck2</i>	Zahl	Beliebiger numerischer Ausdruck.

Hinweis

Verwenden Sie diesen Operator, wenn Sie eine Division benötigen, die den Nachkommateil des Ergebnisses nicht errechnet.

Beispiele

Dieses Beispiel speichert das Ergebnis einer Division zweier Zahlen in einer Variablen:

```
Dim x as Integer
x=50.56\30.34 //x ergibt 1
```

Siehe auch

[/,Mod Operatoren](#).

Abs Funktion

Gibt den Absolut-Wert einer Zahl zurück.

Syntax

Ergebnis=**Abs**(Wert)

Teil	Art	Beschreibung
<i>Wert</i>	<u>Double</u>	Die Zahl, deren Absolutwert Sie ermitteln wollen.
<i>Ergebnis</i>	<u>Double</u>	Der Absolutwert von <i>Wert</i> .

Hinweis

Die **Abs**-Funktion gibt das positive Äquivalent des angegebenen Wertes zurück.

Beispiele

Diese Beispiele verwenden die **Abs**-Funktion, um den Absolutwert der angegebenen Zahlen zu ermitteln.

```
Dim d as Double
d=Abs(23.9) //gibt 23.9 zurück
d=Abs(-23.9) //gibt 23.9 zurück
```

Siehe auch

[Sign](#) Funktion

Abschließende Gänsefüßchen fehlen Fehler

Die abschließenden Gänsefüßchen am Ende eines Strings fehlen.

Beispiel

Das abschließende Gänsefüßchen bei folgendem String, der an den MsgBox-Befehl übergeben wird, fehlt:

```
MsgBox "Hello world
```

Abstrakte Klasse kann nicht vererbt werden Fehler

Sie haben versucht, eine Instanz aus einer abstrakten Klasse zu erzeugen. Sie können nur den New Operator zum Erzeugen einer neuen Instanz eines schon in einem Fenster vorhandenen Steuerelements, eines dem Projekt über „Ablage/Neues Fenster“ hinzugefügten Fensters oder eines schon vorhandenen Menüeintrags verwenden. Das vorhandene Objekt dient als Vorlage für die neue Instanz.

Beispiel

So nicht:

```
Dim w as Window
w=New Window
```

Verwenden Sie stattdessen den New Operator mit einer Instanz der Window-Klasse, die Sie über Ablage/Neues Fenster erzeugen.

```
Dim w as FindWindow
w=New FindWindow
```

Siehe auch

[New Operator](#).

Acos Funktion

Gibt den Arkus-Cosinus des Wertes zurück. Der Arkus-Cosinus ist der Winkel, dessen Cosinus *Wert* ist. Der gelieferte Winkel wird im Bogenmaß übergeben.

Syntax

Ergebnis = **Acos**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der Arkus-Cosinus von <i>Wert</i> .
<i>Wert</i>	<u>Double</u>	Der Wert, dessen Arkus-Cosinus Sie ermitteln wollen.

Hinweis

Die **Acos**-Funktion gibt den Winkel (im Bogenmaß) des übergebenen Cosinus zurück. Um das Ergebnis ins Gradmaß umzurechnen, multiplizieren Sie es mit 180/Pi.

Beispiele

Dieses Beispiel verwendet die **Acos**-Funktion, um den Arkus-Cosinus einer Zahl zu ermitteln.

```
Dim d as Double
Const Pi=3.14159265358979323846264338327950
d=Acos(.5) //ergibt 1.0471976
d=Acos(.5)*180/Pi //ergibt 60
```

Siehe auch

[Cos](#) Funktion.

AddressBook Klasse

Bietet Zugriff auf das Adreßbuch von Mac OS X. Sie können in dem Adreßbuch lesen und schreiben.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
CurrentUser	<u>AddressBookContact</u>	Liefert den Adreßbucheintrag des aktuellen Benutzers.
HasUnsavedChanges	<u>Boolean</u>	Ist <u>True</u> , wenn Änderungen noch nicht gespeichert sind. Benutzen Sie die Save-Methode zum Speichern.

Methoden

Name	Parameter	Beschreibung
Add	Eintrag vom Typ <u>AddressBookRecord</u>	Fügt den neuen Adreßbucheintrag in das Adreßbuch ein.
Contacts		Liefert ein Array mit allen Adreßbucheinträgen vom Typ <u>AddressBookContact</u> .
Find	Abfrage als <u>String</u>	Liefert einen <u>AddressBookRecord</u> . Find sucht über alle Felder aller Einträge im Adreßbuch und gibt ein Array mit allen zutreffenden Einträgen zurück.
Groups		Liefert ein Array mit allen Gruppen. Elemente sind vom Typ <u>AddressBookGroup</u> .
Remove	Eintrag vom Typ <u>AddressBookRecord</u>	Entfernt den Eintrag aus dem Adreßbuch.
Save		Speichert alle Änderungen im Adreßbuch. Liefert ein <u>Boolean</u> zurück. <u>True</u> , wenn erfolgreich.

Beispiel

Es gibt zwei Wege, eine Instanz dieser Klasse zu erzeugen. Sie können den New-Operator oder die AddressBook-Eigenschaft des System-Objekts benutzen.

```
Dim buch As New AddressBook
// oder
Dim buch As AddressBook
buch=System.AddressBook
```

Zugriff auf die Kontakte des Adreßbuchs erhalten Sie über die **Contacts**-Methode. Dieses Beispiel füllt ein Array mit allen Kontakten des Benutzers.

```
Dim buch as AddressBook
buch=System.AddressBook
Dim Kontakte() as AddressBookContact
Kontakte=buch.Contacts
```

Ist das Array gefüllt, können Sie die Anzahl mit der Ubound-Funktion ermitteln und über die Eigenschaften der AddressBookContact-Klasse auf die einzelnen Felder zugreifen. Die AddressBookData-Klasse kann die meisten Felder automatisch in einen String umwandeln, wenn es sich um Felder handelt, die nur einen Wert aufnehmen können.

Zum Beispiel können Sie das Array von AddressBookContact-Elementen durchlaufen und die Firmennamen in einer ListBox ausgeben.

```
Dim i as Integer
Dim buch as AddressBook
buch=System.AddressBook
Dim Kontakte() as AddressBookContact
Kontakte=buch.Contacts
For i =0 to Ubound(Kontakte)
  ListBox1.Addrow Kontakte(i).CompanyName
Next
```

So erreichen Sie den Eintrag für den aktuellen Benutzer.

```
Dim buch as AddressBook
buch=System.AddressBook
Msgbox buch.CurrentUser.FirstName
```

Hier ist eine weitere Möglichkeit, in der über die Value-Methode der AddressBookData-Klasse auf Felder zugegriffen wird.

```
Dim buch as AddressBook
buch=System.AddressBook
Msgbox buch.CurrentUser.FirstName.Value
```

Siehe auch

AddressBookRecord, AddressBookGroup, AddressBookContact, AddressBookData, AddressBookAddress Klassen.

AddressBookAddress Klasse

Enthält die Felder, die zusammen eine vollständige Adresse für das Mac OS X-Adreßbuch ergeben.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
City	<u>String</u>	Die Stadt des Adreßeintrags
Country	<u>String</u>	Das Land des Adreßeintrags
CountryCode	<u>String</u>	Der Ländercode des Adreßeintrags
State	<u>String</u>	Das Bundesland des Adreßeintrags
StreetAddress	<u>String</u>	Die Straßenangabe des Adreßeintrags
Zip	<u>String</u>	Die Postleitzahl des Adreßeintrags

Methoden

Name	Parameter	Beschreibung
Value	Eigenschaftname als <u>String</u>	Liefert den gewünschten Wert vom Typ <u>Variant</u> .
Value	Eigenschaftname als <u>String</u> , <u>Assigns</u> neuerWert als <u>Variant</u>	Weist der angegebenen Eigenschaft einen Wert zu.

Beispiel

Dieses Beispiel liest den Adreßbucheintrag für den aktuellen Benutzer und schreibt ihn in ein EditField.

```
Dim buch as AddressBook
Dim adresse as AddressBookAddress
Dim daten as AddressBookData
Dim s as String

buch=System.AddressBook
daten=buch.CurrentUser.Addresses
If daten.count >0 then
  Adresse=daten.value(0)
  s=adresse.StreetAddress+Adresse.City+", "+Adresse.state+ " "+adresse.Zip
else
  s="Zähler ist Null!"
end if
EditField1.text=s
```

Siehe auch

AddressBookRecord, AddressBookGroup, AddressBookContact, AddressBookData, AddressBook Klassen.

AddressBookContact Klasse

Enthält einen Kontakteintrag des Mac OS X-Adreßbuchs.

Übergeordnete Klasse

AddressBookRecord

Eigenschaften

Name	Typ	Beschreibung
Addresses	<u>AddressBookData</u>	Die Adressen
AIMScreenNames	<u>AddressBookData</u>	Die AIM-Namen
Birthday	<u>AddressBookData</u>	Der Geburtstag
CompanyName	<u>AddressBookData</u>	Der Firmenname
EmailAddresses	<u>AddressBookData</u>	Die E-Mail-Adressen
FirstName	<u>AddressBookData</u>	Der Vorname
HomePage	<u>AddressBookData</u>	Die Internetseite
ICQNumbers	<u>AddressBookData</u>	Die ICQ-Nummern
JabberScreenNames	<u>AddressBookData</u>	Der Jabber-Namen
JobTitle	<u>AddressBookData</u>	Der Beruf
LastName	<u>AddressBookData</u>	Der Nachname
MiddleName	<u>AddressBookData</u>	Der Namenszusatz
MSNScreenNames	<u>AddressBookData</u>	Die MSN-Namen
Note	<u>AddressBookData</u>	Die Notizen des Kontakteintrags
PhoneNumbers	<u>AddressBookData</u>	Telefonnummern
VCard	<u>String</u>	Die VCard des Kontakteintrags
YahooScreenNames	<u>AddressBookData</u>	Die Yahoo-Namen

Konstruktor

Name	Parameter	Beschreibung
Constructor	[vcard als <u>String</u>]	Erzeugt einen Adreßbucheintrag; Wird vcard angegeben, werden diese Daten verwendet und voreingetragen.

Hinweise

Die Methoden der AddressBookData-Klasse geben Ihnen Zugriff auf Bezeichnungen, Namen, Werte und weitere Eigenschaften der AddressBookContact-Einträge. Benutzen Sie diese, um Werte auszulesen oder zu setzen.

Beispiel

Diese Methode liest die E-Mail-Adresse des aktuellen Benutzers unter Verwendung der Count und Value-Methoden der AddressBookData-Klasse.

```
Dim buch as AddressBook
Dim i as Integer
Dim meinKontakt as AddressBookContact
buch=System.AddressBook
meinKontakt=buch.CurrentUser
For i=0 to meinKontakt.emailAddresses.Count
    ListBox1.AddRow meinKontakt.emailAddresses.Value(i)
Next
```

Die folgende Methode setzt einen neuen Wert im Feld FirstName.

```
Dim buchas AddressBook
Dim i as Integer
Dim meinKontakt as AddressBookContact
buch=System.AddressBook
meinKontakt=buch.CurrentUser
meinKontakt.FirstName="Boris"
//oder über die AddressBookData.Value-Methode
meinKontakt.FirstName.Value="Boris"
```

Siehe auch

AddressBookRecord, AddressBookContact, AddressBookAddress, AddressBookData, AddressBook Klassen.

AddressBookContact Klasse

Enthält einen Kontakt des Mac OS X-Adreßbuchs.

Übergeordnete Klasse

AddressBookRecord

Eigenschaften

Name	Typ	Beschreibung
Addresses	<u>AddressBookData</u>	Die Adresse
AIMScreenNames	<u>AddressBookData</u>	Die AIM-Namen
Birthday	<u>AddressBookData</u>	Der Geburtstag
CompanyName	<u>AddressBookData</u>	Der Firmenname
EmailAddresses	<u>AddressBookData</u>	Die Email-Adressen
FirstName	<u>AddressBookData</u>	Der Vorname
HomePage	<u>AddressBookData</u>	Die Internetseite
ICQNumbers	<u>AddressBookData</u>	Die ICQ-Nummern
JabberScreenNames	<u>AddressBookData</u>	Die Jabber-Namen
JobTitle	<u>AddressBookData</u>	Der Beruf
LastName	<u>AddressBookData</u>	Der Nachname
MiddleName	<u>AddressBookData</u>	Der Namenszusatz

Name	Typ	Beschreibung
MSNScreenNames	AddressBookData	Die MSN-Namen
Note	AddressBookData	Die Notiz
PhoneNumbers	AddressBookData	Die Telefonnummern
VCard	String	Die VCard
YahooScreenNames	AddressBookData	Die Yahoo-Namen

Konstruktor

Name	Parameter	Beschreibung
Constructor	[vcard as String]	Erzeugt einen Kontakt; Wird vcard angegeben, werden diese Daten verwendet und voreingetragen.

Hinweise

Die Methoden der [AddressBookData](#)-Klasse, geben Ihnen Zugriff auf Bezeichnungen, Namen, Werte und weitere Eigenschaften, der AddressBookContact-Einträge. Benutzen Sie diese, um Werte auszulesen oder zu setzen.

Beispiel

Diese Methode ermittelt die E-Mail-Adressen des aktuellen Benutzers. Sie verwendet hierfür die Methoden Count und Value der [AddressBookData](#)-Klasse.

```
Dim buch as AddressBook
Dim i as Integer
Dim meinKontakt as AddressBookContact
buch=System.AddressBook
meinKontakt=buch.CurrentUser
For i=0 to meinKontakt.emailAddresses.Count
    ListBox1.AddRow meinKontakt.emailAddresses.Value(i)
Next
```

Die folgende Methode setzt einen neuen Wert im Feld FirstName.

```
Dim buchas AddressBook
Dim i as Integer
Dim meinKontakt as AddressBookContact
buch=System.AddressBook
meinKontakt=buch.CurrentUser
meinKontakt.FirstName="Boris"
//oder über die AddressBookData.Value-Methode
meinKontakt.FirstName.Value="Boris"
```

Siehe auch

[AddressBookRecord](#), [AddressBookContact](#), [AddressBookAddress](#), [AddressBookData](#), [AddressBook](#) Klassen.

AddressBookData Klasse

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
ActualLabel	Index als Integer	Liefert die Bezeichnung der Daten als String , wie sie vom System genannt werden.
Count		Liefert die Anzahl der Werte und Bezeichnungen als Integer .
Label	Index as Integer	Liefert die Bezeichnung als String , die zu den Daten gehört. Mögliche Bezeichnungen sind: Home, Home Fax, Main, Mobile, Pager, Work Fax, Work. Für andere Bezeichnungen wird ein leerer String zurückgegeben.
Name		Liefert den Namen der Eigenschaft als String . Zum Beispiel "FirstName".

Name	Parameter	Beschreibung
Text		Liefert die Daten des Objekts als <u>String</u> . Ist nur ein Wert vorhanden, zum Beispiel der Vorname (AddressBookContact.FirstName), wird dieser zurückgegeben. Enthält die Kontakteigenschaft mehr als einen String, wird nur der erste zurückgegeben.
Value	Index als <u>Integer</u> =0	Liefert den Wert als <u>Variant</u> vom Typ <u>String</u> oder <u>AddressBookAddress</u> . Das Array ist null-basiert.
Value	Index als <u>Integer</u> =0, Assigns neuerWert als <u>Variant</u>	Weist dem Element mit angegebenen Index einen neuen Wert zu.

Beispiel

Die AddressBookData-Klasse kann benutzt werden, um mehr Informationen über die einzelnen Felder eines Objekts vom Typ AddressBookContact zu erhalten. Die Name-Methode liefert den Namen der Eigenschaft, die die Daten darstellen. Über die Label-Methode erhalten Sie die Bezeichnung, unter mit der diese Daten dargestellt werden, zum Beispiel Home, Work, etc. Die Label-Eigenschaft ist nur für AddressBookData-Objekte relevant, die mehrere Daten enthalten können.

Dieses Beispiel zeigt die Telefonnummern des aktuellen Benutzers in einer zweispaltigen Listbox an. Die erste Spalte zeigt die Bezeichnung der Adressbuchdaten („Home“, „Work“, „Mobile“, etc.), die zweite die Telefonnummer.

```
Dim buch as New AddressBook
Dim daten as AddressBookData
Dim c,i as Integer
daten = buch.CurrentUser.PhoneNumbers
c=daten.count-1
If daten.Count > 0 then
  For i=0 to c
    ListBox1.Addrow daten.Label(i)
    ListBox1.Cell(i,1)=daten.Value(i)
  next
Else
  MsgBox "Keine Telefonnummern!"
End if
```

Diese Methode benutzt die Value-Methode, um die E-Mail-Adresse des aktuellen Benutzers zu bekommen.

```
Dim buch as AddressBook
Dim meinKontakt as AddressBookContact
buch=System.AddressBook
meinKontakt=buch.CurrentUser
Msgbox meinKontakt.emailAddresses.Value(0)
```

Siehe auch

AddressBookRecord, AddressBookContact, AddressBookAddress, AddressBookGroup, AddressBook Klassen.

AddressBookGroup Klasse

Verwaltet die Gruppen des Adreßbuchs.

Übergeordnete Klasse

AddressBookRecord

Eigenschaften

Name	Typ	Beschreibung
Name	<u>String</u>	Der Name der Gruppe

Methoden

Name	Parameter	Beschreibung
Add	Eintrag als AddressBookRecord	Fügt den Eintrag oder die Gruppe dem Adreßbuch hinzu. Wenn Eintrag eine Gruppe ist, wird diese als Untergruppe hinzugefügt. Ist Eintrag ein Kontakt, wird dieser als Mitglied der Gruppe hinzugefügt.
Contacts		Liefert ein Array vom Typ AddressBookContact , das alle Kontakte der Gruppe enthält.
Groups		Liefert ein Array vom Typ AddressBookGroup , das alle Untergruppen der Gruppe enthält.
Remove	Eintrag als AddressBookRecord	Entfernt den Eintrag aus der Gruppe.

Beispiel

Dieses Beispiel zeigt alle Gruppen des aktuellen Benutzers in einer [ListBox](#) an.

```
Dim buch as Addressbook
Dim gruppen() as AddressBookGroup
Dim i as Integer

buch=System.AddressBook
gruppen=buch.Groups
For i=0 to Ubound(gruppen)
    ListBox1.addrow gruppen(i).name
Next
```

Dieses Beispiel zeigt zwei Kontakte der ersten Gruppe in einer [ListBox](#) an.

```
Dim buch as Addressbook
Dim gruppen() as AddressBookGroup
Dim kontakte() as AddressBookContact
Dim i as Integer

buch=System.AddressBook
gruppen=buch.Groups // Array mit Gruppen; Eine Gruppe muß existieren

kontakte=gruppen(0).contacts // Array mit Kontakten der ersten Gruppe
For i=0 to Ubound(kontakte)
    ListBox1.addrow kontakte(i).CompanyName
    ListBox1.cell(i,1)=kontakte(i).LastName
Next
```

Siehe auch

[AddressBookRecord](#), [AddressBookGroup](#), [AddressBookAddress](#), [AddressBookData](#), [AddressBook](#) Klassen.

AddressBookRecord Klasse

Speichert Informationen über einen Adreßbucheintrag und besitzt Methoden, um Werte auszulesen, zu schreiben und zu löschen. Dies ist die Basisklasse für die Klassen [AddressBookContact](#) und [AddressBookGroup](#).

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
CreationDate	Date	Liefert das Erstellungsdatum.
ModificationDate	Date	Liefert das Datum der letzten Modifikation.
UniqueID	String	Liefert die eindeutige ID des Eintrags.

Methoden

Name	Parameter	Beschreibung
Value	Eigenschaftname als String	Liefert den Wert der Eigenschaft als Variant .

Name	Parameter	Beschreibung
Value	Eigenschaftname als <u>String</u> Assigns neuerWert als <u>Variant</u>	Setzt einen neuen Wert vom Typ <u>Variant</u> für die angegebene Eigenschaft.
RemoveValue	Eigenschaftname als <u>String</u>	Löscht den Wert der angegebenen Eigenschaft.

Siehe auch

[AddressBook](#), [AddressBookAddress](#), [AddressBookGroup](#), [AddressBookContact](#), [AddressBookData](#); [System](#) Objekt.

AddressOf Operator

Liefert einen Zeiger auf eine Methode. Dieser Operator kann mit der Declare-Anweisung benutzt werden, um Funktionen in externen Bibliotheken zu verwenden.

Syntax

Zeiger = **AddressOf** MethodenName

Teil	Beschreibung
Zeiger	Zeiger auf <i>MethodenName</i>
MethodenName	Die Methode, für die Sie einen Zeiger benötigen.

Siehe auch

[MemoryBlock](#) Funktion, [Declare](#)-Anweisung.

ADSP4DServer Klasse

Wird verwendet, um unter Mac OS „Classic“ eine 4th Dimension/4D Server Datenquelle über ADSP zu öffnen. Bitte verwenden Sie diese Klasse anstelle von [Open4DDatabaseByADSP](#) oder [Select4DDatabasebyADSP](#).

Übergeordnete Klasse

[Database](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Task</i>	<u>String</u>	Beschreibung der Verbindung. Der Task-Parameter ist der Name des Prozesses, der auf dem 4D-Server erzeugt wird, wenn Sie eine Verbindung aufbauen. Dieser wird in dem 4D-Server-Fenster angezeigt.

Hinweise

Um auf einen 4D-Server zugreifen zu können, muß er derart konfiguriert sein, daß Verbindungen grundsätzlich zulässig sind und der Benutzer einer Gruppe angehört, die entsprechende Zugriffsrechte besitzt, um Verbindungen aufzubauen.

Damit Sie den ADSP4DServer benutzen können, muß das korrekte 4D-Server-Plugin im Plugins-Verzeichnis von REALbasic installiert sein. REAL Software bietet separate Plugins für den 4D-Server in den Versionen 6.0, 6.5 und 6.7 an. Die REALbasic CD enthält alle Datenbank-Plugins und aktuelle Software erhalten Sie unter <http://www.realsoftware.com>. Wenn Sie eine 4D-Datenquelle unter Windows verwenden, sollten Sie die 4DOpen.DLL-Datei (von 4D bereitgestellt) in das gleiche Verzeichnis wie Ihre Applikation kopieren. Das stellt sicher, daß die Applikation diese Library finden kann und daß die passende Version der Library verwendet wird.

Sie finden eine Reihe von Database-Plugins auf der REALbasic-CD, neuere Versionen könnte es aber auf der REAL Software Web-Site unter www.realsoftware.com geben.

Beispiel

Einloggen in eine Datenbank unter Verwendung der Benutzer- und Passwort-Eigenschaften der [Database](#)-Klasse.

```
Dim db as ADSP4DServer60
db=New ADSP4DServer60
db.userName="Alfred"
db.password="MonoBloc"
db.connectionString="theString"
db.task="myTask"
```

```
If db.Connect then
// Ihr Programm beginnt hier...
else
MsgBox "Verbindung fehlgeschlagen"
end if
```

Siehe auch

[TCP/IP4DServer](#), [Database Klassen](#).

And Operator

Wird verwendet, um einen logischen Vergleich zweier boole'scher Ausdrücke vorzunehmen.

Syntax

Ergebnis=Ausdruck1 **And** Ausdruck2

Teil	Beschreibung
<i>Ergebnis</i>	Ein <u>boole'scher</u> Wert.
<i>Ausdruck1</i>	Ein beliebiger <u>boole'scher</u> Ausdruck.
<i>Ausdruck2</i>	Ein beliebiger <u>boole'scher</u> Ausdruck.

Hinweis

Wenn beide Ausdrücke True (Wahr) ergeben, dann ist das Ergebnis True. Wenn einer der Ausdrücke False (Falsch) ergibt, ist das Ergebnis False.

Beispiele

Dieses Beispiel verwendet den **And**-Operator, um logische Vergleiche durchzuführen.

```
Dim a As Boolean
Dim b As Boolean
Dim c As Boolean
Dim d As Boolean
a=True
b=True
d=a And b //ergibt True
d=a And c //ergibt False
```

Siehe auch

[Not Operator](#), [Or Operator](#).

App Objekt

Eine Instanz der Application-Klasse, die die Anwendung repräsentiert.

Syntax

App.Eigenschaft=Wert

App.Methode

Hinweise

Das **App**-Objekt repräsentiert die Anwendung selbst (im Gegensatz zu einem Fenster oder einem Steuerelement). Das erlaubt Ihnen, auf die Objekt-Eigenschaften, Events und Methoden der Anwendung zuzugreifen, ohne explizit eine Referenz darauf zu speichern. Wenn Sie eine Klasse erzeugt haben, die eine Unterklasse von „Application“ ist, wird das **App**-Objekt eine Referenz auf diese Klasse zurückgeben.

Beispiele

Dieses Beispiel ändert den Mausfeil (MouseCursor) des Programms in die Armbanduhr:

```
App.MouseCursor=WatchCursor
```

Hinweis: Der Zugriff auf den Resourcefork des Programmes wird nur auf dem Macintosh unterstützt. Prüfen Sie den Wert der Konstante TargetMacOS, bevor Sie versuchen, einen Resourcefork zu öffnen.

Das folgende Beispiel benutzt die DoEvents-Methode. Wenn Sie PushButtons einmal mit, einmal ohne DoEvents-Methoden, erzeugen, sehen Sie, wie REALbasic mit anderen Aufgaben umgeht, während die Hauptschleife läuft.

```
Dim t as Integer
t=Ticks

While t+600>Ticks
  App.DoEvents
Wend
```

Siehe auch

[Application](#), [MDIWindow](#) Klassen; [System](#) Objekt.

Append Methode

Hängt ein neues Element an das Ende eines eindimensionalen Arrays an und erhöht die Länge des Arrays um Eins.

Syntax

array.**Append** Wert

Teil	Beschreibung
<i>array</i>	Das Array, an das der Wert angehängt werden soll.
<i>Wert</i>	Der Wert, der dem neuen Array-Element zugewiesen werden soll.

Hinweis

Die **Append**-Methode funktioniert nur mit eindimensionalen Arrays.

Beispiel

In diesem Beispiel wird ein neues Element an das Ende des „aNames“-Arrays angehängt.

```
aNames.append "Kurt"
```

Siehe auch

[Dim](#) Ausdruck; [Array](#), [Split](#), [Join](#), [Ubound](#) Funktionen; [Insert](#), [Redim](#), [Remove](#), [Pop](#), [IndexOf](#), [Insert](#), [Shuffle](#), [Sort](#) Methoden; [ParamArray](#) Schlüsselwörter.

AppleEvent Klasse

AppleEvent-Objekte können verwendet werden, um mit anderen Macintosh-Programmen und der Macintosh-Systemsoftware zu kommunizieren. Wenn Sie Ihr Programm für Nicht-Macintosh-Betriebssysteme compilieren, überprüfen Sie unbedingt die globalen boole'schen Konstanten [TargetMacOS](#) und [TargetWin32](#). Diese Konstanten werden automatisch in der fertigen Applikation gesetzt und liefern [True](#) zurück, wenn das Programm auf dem jeweiligen Betriebssystem läuft.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>BooleanParam</i>	ParameterName als String	Ein boole'scher Wert, der als Parameter dem AppleEvent übergeben wird.
<i>DescListParam</i>	ParameterName als String	Ein AppleEventDescList -Objekt, das als Parameter dem AppleEvent übergeben wird.
<i>DoubleParam</i>	ParameterName als String	Ein Double, das als Parameter dem AppleEvent übergeben wird.
<i>EnumeratedParam</i>	ParameterName als String	Ein vier Zeichen langer String , der als Parameter dem AppleEvent übergeben wird.
<i>FolderItemParam</i>	ParameterName als String	Ein FolderItem , das als Parameter dem AppleEvent übergeben wird.
<i>IntegerParam</i>	ParameterName als String	Ein Integer (übergeben als String) der als Parameter dem AppleEvent übergeben wird.

Name	Art	Beschreibung
<i>MacTypParam</i>	ParameterName als <u>String</u>	Ein vier Zeichen langer <u>String</u> , der als Parameter dem AppleEvent übergeben wird.
<i>ObjectSpecifierParam</i>	ParameterName als <u>String</u>	Ein <u>AppleEventObjectSpecifier</u> , der als Parameter dem AppleEvent übergeben wird.
<i>Ptr</i>	<u>Integer</u>	Ein Pointer auf eine darunterliegende AppleEvent-Datenstruktur für die Verwendung mit Declare-Befehlen.
<i>RecordParam</i>	ParameterName als <u>String</u>	Ein <u>AppleEventRecord</u> , der als Parameter dem AppleEvent übergeben wird.
<i>ReplyBoolean</i>	<u>Boolean</u>	Ein Reply in <u>boole'scher</u> Form. <u>True</u> bedeutet, daß die Applikation die Nachricht erfolgreich bearbeitet hat.
<i>ReplyDescList</i>	<u>AppleEventDescList</u>	Ein Reply in <u>AppleEventDescriptorList</u> -Form.
<i>ReplyDouble</i>	<u>Double</u>	Ein Reply in <u>Double</u> -Form.
<i>ReplyEnumerated</i>		
<i>ReplyFolderItem</i>	<u>FolderItem</u>	Ein Reply in <u>FolderItem</u> -Form.
<i>ReplyInteger</i>	<u>Integer</u>	Ein Reply in <u>Integer</u> -Form.
<i>ReplyMacType</i>	<u>String</u>	
<i>ReplyObjectSpecifier</i>	<u>AppleEventObjectSpecifier</u>	Ein Reply in <u>AppleEventObjectSpecifier</u> Form.
<i>ReplyPtr</i>	<u>Integer</u>	Ein Reply in Pointer-Form auf die darunterliegende AppleEvent-Struktur zur Verwendung mit Declare-Befehlen.
<i>ReplyRecord</i>	<u>AppleEventRecord</u>	Ein Reply in <u>AppleEventRecord</u> Form.
<i>ReplyString</i>	<u>String</u>	Ein Reply in <u>String</u> -Form.
<i>SingleParam</i>	ParameterName als <u>String</u>	Ein Single, der als Parameter dem AppleEvent übergeben wird.
<i>StringParam</i>	ParameterName als <u>String</u>	Ein <u>String</u> , der als Parameter dem AppleEvent übergeben wird.
<i>TimeOut</i>	<u>Integer</u>	Timeout-Zeit in Sekunden -1 = Voreinstellung verwenden -2 = Kein Timeout

Methoden

Name	Parameter	Beschreibung
<i>LoadFromTemplate</i>	Template als <u>AppleEventTemplate</u>	Lädt ein AppleEvent Template.
<i>Send</i>		Gibt <u>True</u> zurück, wenn das AppleEvent erfolgreich gesendet wurde und <u>False</u> , falls nicht. <u>True</u> bedeutet nicht, daß die Applikation, die die Nachricht empfing, diese erfolgreich bearbeitet hat. Verwenden Sie hierfür die ReplyBoolean-Eigenschaft.
<i>SetNullParam</i>	ParameterName als <u>String</u>	Setzt den durch das Schlüsselwort angegebenen Parameter auf Null.

Hinweis

AppleEvent-Objekte werden verwendet, um Informationen zwischen Ihrer und anderen Macintosh-Applikationen oder dem Mac OS zu senden und zu empfangen. Um ein AppleEvent von Ihrer Applikation an eine andere zu senden, müssen Sie ein AppleEvent mittels der Funktion NewAppleEvent erzeugen, die Eigenschaften des AppleEvents mit den nötigen Daten versorgen und dann die Send-Funktion des AppleEvents aufrufen, um ihn zu versenden.

AppleEvents können ebenso von Ihrem Programm empfangen werden. Wenn Ihr Programm ein AppleEvent erhält, wird das HandleAppleEvent vom Application-Objekt aufgerufen und das AppleEvent wird diesem Event als Parameter übergeben.

Um mehr Informationen über AppleEvents zu erhalten, schauen Sie unter Application-Klasse nach.

Antworten auf ein AppleEvent

Wenn ein AppleEvent empfangen wird (über einen AppleEvent-Handler), können die Eigenschaften ReplyBoolean, ReplyInteger und ReplyString verwendet werden, um automatisch eine Antwort an die Applikation zurückzusenden, die das AppleEvent verschickt hat.

Wenn Sie ein AppleEvent (über die Send-Methode) senden, können die Eigenschaften ReplyBoolean, ReplyInteger, und ReplyString verwendet werden, um irgendeine Antwort zu erhalten, die die Zielapplikation zurückgeschickt hat, nachdem sie das AppleEvent empfangen hatte.

Z.B. zeigt folgende Zeile, daß die Applikation die AppleEvent-Nachricht erfolgreich bearbeitet hat:

```
e.ReplyBoolean=True
```

Um festzulegen, ob die andere Applikation eine Nachricht erfolgreich bearbeitet hat, verwenden Sie Code wie folgenden:

```
Dim as as AppleEvent
Dim s as String
:
:
If as.Send then //AE erfolgreich gesendet
    If ae.replyBoolean then
        s = "Ja (bearbeitet)"
    else
        s = "Ja (nicht bearbeitet)"
    end if
else
    s = "Nein"
end if
```

Mehr Informationen über AppleEvents erhalten Sie im Apple's Developer Bereich im Internet.

Beispiele

Dieses Beispiel verwendet ein AppleEvent, um den Finder anzuweisen, das Kontrollfeld „Erscheinungsbild“ zu öffnen:

```
Dim a as AppleEvent
a = NewAppleEvent("aevt", "odoc", "MACS")
a.FolderItemParam("----")= ControlPanelsFolder.Child("Erscheinungsbild")
If Not a.Send Then
    MsgBox "Das Kontrollfeld Erscheinungsbild konnte nicht geöffnet werden."
End if
```

Im nächsten Beispiel wird SimpleText (welches in diesem Fall schon gestartet sein muß) aufgetragen, zwei Dokumente („Mein Dokument“ und „Mein zweites Dokument“) zu öffnen, die sich im gleichen Ordner wie das REALbasic-Projekt befinden müssen:

```
Dim a as AppleEvent
Dim list as AppleEventDescList
a = NewAppleEvent("aevt", "odoc", "ttxt")
list = New AppleEventDescList
list.AppendFolderItem GetFolderItem("Mein Dokument")
list.AppendFolderItem GetFolderItem("Mein zweites Dokument")
a.DescListParam("----") = list
If Not a.Send Then
    MsgBox "Das AppleEvent konnte nicht verschickt werden."
End If
```

Dieses Beispiel zeigt die Mac OS-Version an, unter der das Programm läuft:

```
Dim ae as AppleEvent
Dim obj as AppleEventObjectSpecifier
Dim version as string

If TargetMacOS then
    ae = NewAppleEvent("core", "getd", "MACS")
    obj = GetPropertyObjectDescriptor(nil, "ver2")
    ae.ObjectSpecifierParam("----") = obj

    If Not ae.send() then
        MsgBox "Das Event konnte nicht gesendet werden."
    else
        version = ae.ReplyString
        If (version <> "") then
            MsgBox "Die Systemversion ist "" + version + ""."
        else
            MsgBox "Diese Systemversion unterstützt dieses Event nicht."
```



```

    end if
  end if
else
  MsgBox "Dies ist kein Macintosh!"
end if

```

Siehe auch

[AppleEventDescList](#) Klasse, [AppleEventObjectSpecifier](#) Klasse, [Application](#) Objekt, [NewAppleEvent](#) Funktion, [#If...#Else](#) Befehl, [TargetMacOS](#), [TargetPPC](#), [TargetWin32](#), [DebugBuild](#) Konstanten.

AppleEventDescList Klasse

Wird verwendet, um komplexe Informationen mittels AppleEvent an andere Applikationen zu senden.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>Count</i>	Integer	Die Anzahl der Einträge der Liste.
<i>FolderItemItem</i>	Index als Integer	Ein FolderItem -Feld, das mit dem AppleEvent übergeben wird.
<i>IntegerItem</i>	Index als Integer	Ein Integer -Feld, das mit dem AppleEvent übergeben wird.
<i>RecordItem</i>	Index als Integer	Ein AppleEventRecord -Feld, das mit dem AppleEvent übergeben wird.
<i>StringItem</i>	Index als Integer	Ein StringItem -Feld, das mit dem AppleEvent übergeben wird.

Methoden

Name	Parameter	Beschreibung
<i>AppendBoolean</i>	Wert als Boolean	Fügt einen neuen boole'schen Wert zum Array hinzu.
<i>AppendDescList</i>	Wert als AppleEventDescList	Fügt einen neuen AppleEventDescList -Wert zum Array hinzu.
<i>AppendFolderItem</i>	Wert als FolderItem	Fügt der Eigenschaft FolderItem-Array ein neues FolderItem hinzu.
<i>AppendInteger</i>	Wert als Integer	Fügt der Eigenschaft IntegerItem-Array einen neuen Integer hinzu.
<i>AppendObjectSpecifier</i>	Wert als AppleEventObjectSpecifier	Fügt dem Array ein neues AppleEventObjectSpecifier -Objekt hinzu.
<i>AppendRecord</i>	Wert als AppleEventRecord	Fügt der Eigenschaft RecordItem-Array einen neuen AppleEventRecord hinzu.
<i>AppendString</i>	Wert als String	Fügt der Eigenschaft StringItem-Array einen neuen String hinzu.
<i>BooleanItem</i>	Index als Integer	Gibt ein boole'sches Index-Objekt zurück.
<i>DescListItem</i>	Index als Integer	Gibt ein AppleEventDescList -Index-Objekt zurück.
<i>ObjectSpecifierItem</i>	Index als Integer	Gibt ein AppleEventObjectSpecifier -Index-Objekt zurück.

Beispiele

Beispiele finden Sie unter [AppleEvent](#) Klasse.

Siehe auch

[AppleEvent](#) Klasse, [AppleEventRecord](#) Klasse, [NewAppleEvent](#) Funktion.

AppleEventObjectSpecifier Klasse

Ein [AppleEventObjectSpecifier](#) spezifiziert ein Objekt, auf das ein [AppleEvent](#) angewendet werden kann.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Keine.

Events

Keine.

Methoden

Keine.

Hinweis

Ein AppleEventObjectSpecifier kann ein beliebiges Objekt, wie ein Dokument oder einen PushButton, repräsentieren. AppleEventObjectSpecifiers werden von verschiedenen Funktionen zurückgegeben und werden an die ObjectSpecifierParam-Eigenschaft eines AppleEvent-Klassenobjekts übergeben.

Beispiele

Siehe AppleEvent-Klasse.

Siehe auch

AppleEvent Klasse, GetIndexedObjectDescriptor, GetNamedObjectDescriptor, GetOrdinalObjectDescriptor, GetPropertyObjectDescriptor, GetRangeObjectDescriptor, GetStringComparisonObjectDescriptor, GetTestObjectDescriptor Funktionen.

AppleEventRecord Klasse

Ein Record, das eine Reihe unterschiedlicher Datentypen enthält. AppleEventRecords können mit Hilfe von AppleEvents übergeben werden.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>BooleanParam</i>	ParameterName als <u>String</u>	Ein <u>boole'scher</u> Ausdruck, der als Parameter dem AppleEventRecord übergeben wird.
<i>DescListParam</i>	ParameterName als <u>String</u>	Ein <u>AppleEventDescList</u> -Objekt, das als Parameter dem AppleEventRecord übergeben wird.
<i>DoubleParam</i>	ParameterName als <u>String</u>	Ein <u>Double</u> , das als Parameter dem AppleEventRecord übergeben wird.
<i>EnumeratedParam</i>	ParameterName als <u>String</u>	Ein vier Zeichen langer String, der als Parameter dem AppleEventRecord übergeben wird.
<i>FolderItemParam</i>	ParameterName als <u>String</u>	Ein <u>FolderItem</u> , das als Parameter einem AppleEventRecord übergeben wird.
<i>IntegerParam</i>	ParameterName als <u>String</u>	Ein Integer (übergeben als ein <u>String</u>), der als Parameter einem AppleEventRecord übergeben wird.
<i>MacTypeParam</i>	ParameterName als <u>String</u>	Ein vier Zeichen langer <u>String</u> , der als Parameter einem AppleEventRecord übergeben wird.
<i>ObjectSpecifierParam</i>	ParameterName als <u>String</u>	Ein <u>AppleEventObjectSpecifier</u> , der als Parameter einem AppleEventRecord übergeben wird.
<i>RecordParam</i>	ParameterName als <u>String</u>	Ein <u>AppleEventRecord</u> , das als Parameter einem AppleEventRecord übergeben wird.
<i>SingleParam</i>	ParameterName als <u>String</u>	Ein Single, das als Parameter einem AppleEventRecord übergeben wird.
<i>StringParam</i>	ParameterName als <u>String</u>	Ein <u>String</u> , der als Parameter einem AppleEventRecord übergeben wird.

Methoden

Name	Parameter	Beschreibung
<i>LoadFromTemplate</i>	Template als <u>AppleEventTemplate</u>	Lädt ein AppleEventTemplate.
<i>SetNullParam</i>	ParameterName als <u>String</u>	Setzt den durch das Schlüsselwort angegebenen Parameter auf Null.

Siehe auch

AppleEvent Klasse.

AppleEventTarget Klasse

Wird verwendet, um Apple-Events an Applikationen auf anderen Rechnern zu schicken.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Remote</i>	<u>Boolean</u>	<u>True</u> , falls es sich um einen anderen Rechner handelt.
<i>Computer</i>	<u>String</u>	Name des Rechners.
<i>PortType</i>	<u>String</u>	Creator-Code, der das Programm identifiziert.
<i>Zone</i>	<u>String</u>	AppleTalk-Zone.

Methoden

Name	Parameter	Beschreibung
<i>NewAppleEvent</i>	EventClass als <u>String</u> , EventID als <u>String</u>	Erzeugt einen neuen AppleEvent. Liefert ein <u>AppleEvent</u> -Objekt zurück.

AppleEventTemplate Klasse

Ein Template-Objekt zum Erzeugen von AppleEvent-Objekten.

Übergeordnete Klasse

Object

Eigenschaften

Name	Parameter	Beschreibung
<i>IntegerParam</i>	Name als <u>String</u>	Ein <u>Integer</u> , der dem Apple-Event als Parameter übergeben wird.
<i>StringParam</i>	Name als <u>String</u>	Ein <u>String</u> , der dem Apple-Event als Parameter übergeben wird.
<i>FolderItemParam</i>	Name als <u>String</u>	Ein <u>FolderItem</u> , das dem Apple-Event als Parameter übergeben wird.

AppleMenuFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Apple-Menü zu bekommen. Unter Windows bietet es Zugriff auf das Programme-Verzeichnis im Start-Menü-Verzeichnis.

Syntax

Ergebnis = **AppleMenuFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein <u>FolderItem</u> , das den Ordner Apple-Menü repräsentiert.

Hinweise

Verwenden Sie die Funktion **AppleMenuFolder**, wenn Sie auf einem Macintosh Zugriff auf den Ordner Apple-Menü im Systemordner benötigen oder das FolderItem benötigen, das die Objekte im Start-Menü unter Windows enthält.

Die Funktion **AppleMenuFolder** bietet eine Möglichkeit, auf die Einträge im Apple-Menü zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

Unter Linux liefert AppleMenuFolder Nil zurück.

Beispiele

Dieses Beispiel listet alle Einträge des Apple-Menüs in einer ListBox auf.

```
Dim i as Integer
Dim f as FolderItem
f=AppleMenuFolder
For i=1 to f.Count
    ListBox1.AddRow f.Item(i).Name
Next
Exception err as NilObjectException
MsgBox "Es gibt keinen Apple-Menü-Ordner!"
```

Siehe auch

ControlPanelsFolder, DesktopFolder, ExtensionsFolder, FontsFolder, PreferencesFolder, ShutDownItemsFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

Application Klasse

Die Application-Klasse verfügt über Eigenschaften, Methoden und Events, die zentraler Bestandteil einer Applikation sind.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>BalloonHelpVisible</i>	<u>Boolean</u>	Nur unter Windows unterstützt. Wenn dies False ist, ist die Ballon-Hilfe deaktiviert, True aktiviert sie.
<i>BugVersion</i>	<u>Integer</u>	Bug-Version der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.
<i>DockItem</i>		(nur Mac OS X). Erlaubt Ihnen die Manipulation des Dock-Eintrags, das zu der Applikation gehört. Die DockItem-Eigenschaft besitzt zwei Methoden, UpdateRow und ResetIcon. ResetIcon setzt das Icon in seinen ursprünglichen Zustand zurück (Standardaussehen). Das Aussehen des Icons können Sie mit den Methoden der <u>Graphics</u> -Klasse ändern. Da ein Mac OS X-Icon automatisch skaliert wird, sollten Sie das Icon als ein Pixel-Icon in der Größe 128*128 entwerfen. Um das Icon neu zu zeichnen, rufen Sie die UpdateNow-Methode auf. Sie können auch mit Hilfe der ClearRect-Eigenschaft der <u>Graphics</u> -Klasse mit einem leeren Icon beginnen. Alles, was man mit einem <u>Graphics</u> -Objekt machen kann, kann man mit dem Grafik-Objekt im Dock machen (Zeichnen in einem Bild oder ein Group 2D verwenden). Die DockItem-Eigenschaft der <u>Window</u> -Klasse erlaubt Ihnen, den Dock-Eintrag jedes einzelnen Fensters zu steuern.
<i>LongVersion</i>	<u>Integer</u>	Lange Version der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.
<i>MajorVersion</i>	<u>Integer</u>	Lange Version der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.

Name	Art	Beschreibung
<i>MenuBar</i>	<u>MenuItem</u>	Repräsentiert die globale Menüleiste der Applikation. Die Kinder sind die Menüs. Sie können die komplette Menüleiste ersetzen, indem Sie MenuBar ein neues MenuItem zuweisen. Jedem Fenster kann eine eigene Menüleiste über seine MenuBar-Eigenschaft zugewiesen werden. Wenn einem Fenster keine MenuBar zugewiesen ist, wird die Menüleiste der Applikation verwendet.
MinorVersion	<u>Integer</u>	Kurze Version der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.
<i>MouseCursor</i>	<u>MouseCursor</u>	Der Mauszeiger, der dargestellt wird, so lange das Programm läuft. Wenn die Application-Klasse MouseCursor nicht <u>Nil</u> ist, werden die Nicht-Nil-MouseCursor-Eigenschaften, die zu irgendeinem <u>Window</u> oder <u>Control</u> gehören, ignoriert. Namen sind: WatchCursor, IBeamCursor und ArrowCursor. MouseCursors können ebenso aus Resource-Dateien zugewiesen werden.
NonReleaseVersion	<u>Integer</u>	NonRelease-Version der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.
PackageInfo	<u>Integer</u>	Zusatzinfo der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.
RegionCode	<u>Integer</u>	Region-Code der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird. Unter Windows nicht unterstützt.
ResourceFork	<u>ResourceFork</u>	Wird verwendet, um an den Resourcefork der Applikation zu kommen. Wird nur auf dem Macintosh unterstützt. Prüfen Sie den Wert von <u>TargetMacOS</u> , bevor Sie versuchen, auf die Ressourcen der Applikation zuzugreifen.
ShortVersion	<u>Integer</u>	Kurze Version der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.
StageCode	<u>Integer</u>	Stage-Code der Applikation, passend zur Versionsinformation, die in der Dialogbox für die Compiler-Einstellungen vorgenommen wird.

Methoden

Name	Parameter	Beschreibung
<i>NewDocument</i>		Ruft den Event-Handler NewDocument auf.
<i>OpenDocument</i>	File als <u>FolderItem</u>	Ruft den Event-Handler OpenDocument auf.

Events

Name	Parameter	Beschreibung
<i>Activate</i>		Die Applikation wurde eingeblendet.
<i>Close</i>		Die Applikation wird beendet.
<i>Deactivate</i>		Die Applikation wurde ausgeblendet.
<i>EnableMenuItems</i>		Wird ausgeführt, wenn der Anwender in die Menüleiste klickt, um Ihnen die Gelegenheit zu geben, zu entscheiden, welche Menüeinträge anwählbar sein sollen. Wird ebenso aufgerufen, wenn die Applikation gestartet wird und kein Default-Fenster in der Dialogbox für die Projekt-Einstellungen angewählt ist und wenn das letzte offene Fenster geschlossen wurde.
<i>HandleAppleEvent</i>	Event als <u>AppleEvent</u> , EventClass als <u>String</u> , EventID als <u>String</u>	Ein <u>AppleEvent</u> wurde empfangen. Geben Sie <u>True</u> zurück, um das <u>AppleEvent</u> anzunehmen und <u>False</u> , um es abzuweisen.
<i>NewDocument</i>		Der Anwender hat die Applikation durch Doppelklick auf das Programmicon gestartet oder die Methode NewDocument wurde aufgerufen.
<i>Open</i>		Die Applikation wird gerade gestartet.

Name	Parameter	Beschreibung
<i>OpenDocument</i>	File als <u>FolderItem</u>	Der Anwender hat auf ein Dokument einen Doppelklick ausgeführt, dessen Creator auf den der Applikation paßt oder die Methode OpenDocument wurde aufgerufen.
<i>UnhandledException</i>	error als <u>RuntimeException</u>	Erhält anderweitig entgangene Exceptions – Exceptions, die in einem Exception-Block mit einem der Runtime-Exception-Handlers hätte abgefangen werden können. Wenn der passende Exception-Block fehlt, löst der RuntimeException-Error dieses Events aus und erlaubt es Ihnen, hier Exceptions zu verarbeiten. Siehe auch das Beispiel. Liefert ein Boolean. Wenn es True liefert, wird der Standard-Dialog, der eine unhandled exception anzeigt und das Programm beendet, unterdrückt. Übergeben Sie False, um das Standardverhalten von REALbasic wiederherzustellen. Siehe auch RuntimeException-Klasse.

Hinweise

REALbasic fügt eine Klasse basierend auf der **Application**-Klasse dem Projekt-Fenster hinzu, wenn Sie ein neues Projekt erzeugen. Diese neue Unterklasse gibt Ihnen die Möglichkeit, auf die Methoden und Events des **Application**-Objekts zuzugreifen.

Die neue Unterklasse hat ihren eigenen Menü-Handler, welcher verwendet werden kann, um Menüeinträge zu verwalten, wenn kein Fenster offen ist oder für Menüeinträge, die immer den selben Menü-Handler aufrufen, egal welches Fenster das oberste ist. Sie können die globale Menüleiste ändern, indem Sie der MenuBar-Eigenschaft eine andere Menüleiste zuweisen.

Informieren Sie sich auch bei der Control-Klasse, wie man die Cursor-Form ändert und dem Projekt einen neuen Cursor hinzufügt.

Die App-Funktion gibt eine Referenz auf das **Application**-Objekt zurück. Für weitere Informationen schlagen Sie bei der App-Funktion nach.

Resource-Dateien (Dateien vom Typ „rsrc“), die mit ResEdit oder etwas ähnlichem erzeugt wurden, können dem Projekt durch Draggen der Resource-Datei in das Projekt-Fenster hinzugefügt werden. Auf diese Ressourcen kann man dann über die ResourceFork-Eigenschaften des **Application**-Objekts zugreifen. Die Ressourcen aller Resource-Dateien werden in die erzeugte Applikation eingebunden. Im Falle eines Konflikts überschreiben spätere Resource-Dateien frühere, wobei die Dateien in der Reihenfolge, in der sie im Projektfenster erscheinen, geschrieben werden.

Die Ressourcen der Applikation sind nur lesbar. Wenn Sie in Ressourcen schreiben wollen, müssen Sie eine externe Resource-Datei verwenden. Mehr Informationen finden Sie im Abschnitt ResourceFork-Klasse.

Hinweis: Zugriff auf den Resourcefork wird nur auf dem Macintosh unterstützt. Prüfen Sie den Wert von TargetMacOS, bevor Sie versuchen, auf den Resourcefork der Applikation zuzugreifen.

Beispiel

Folgender Code im Action-Event eines PushButtons führt zu einem OutOfBoundsException-Laufzeitfehler, wenn der Zähler i den Wert von FontCount erreicht.

```
Dim i as Integer
For i=1 to FontCount
  ListBox1.Addrow Font(i)
Next
```

Da es innerhalb dieser Methode keinen Exception-Handler gibt, wird die Runtime-Exception an die Application-Klasse übergeben, die das UnhandledException-Event auslöst.

Folgender Code im UnhandledException-Event der App-Klasse fängt die unhandled OutOfBoundsException auf. Natürlich fängt es alle unhandled OutOfBoundsExceptions der gesamten Applikation auf, so daß es nicht weiß, wo der Fehler auftrat. Sie können stattdessen einen Exception-Block innerhalb des Action-Events des PushButtons plazieren, so daß Sie etwas genauere Diagnose-Möglichkeiten anbieten können.

```
Function UnhandledException(error as RuntimeException) as Boolean
  If error IsA OutOfBoundsException then
    MsgBox "Ein OutOfBounds Exception Fehler ist aufgetreten!"
  End if
  Return True
End Function
```

Das folgende Beispiel benutzt die DoEvents-Methode. Wenn Sie einen PushButton erzeugen, einmal mit und einmal ohne DoEvents-Methoden, so sehen Sie, wie REALbasic mit weiteren Aufgaben umgeht, während die Hauptschleife läuft.

```
Dim t as Integer
t=Ticks

While t+600>Ticks
  App.DoEvents
Wend
```

Siehe auch

[App](#), [System](#), [ResourceFork](#) Objekte, [AppleEvent](#), [MDIWindow](#) Klassen.

ApplicationSupportFolder Funktion

Liefert ein [FolderItem](#), das auf den Application Support-Ordner im aktiven System-Ordner verweist. Unter Windows verweist es, wenn vorhanden, auf das Verzeichnis „C:\Dokumente und Einstellungen\Anwendername\Lokale Einstellungen\Anwendungsdaten“, sonst auf das Windows-Verzeichnis.

Syntax

Ergebnis = **ApplicationSupportFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Ein FolderItem , das auf den Application Support Ordner verweist.

Hinweis

Der Application Support Ordner kann Code oder Dateien beinhalten, die von anderen Anwendungen benötigt werden. Diese Dateien sollten normalerweise nicht dorthin geschrieben werden, nachdem sie bereits installiert wurden. Löscht man Dateien aus diesem Ordner, verlieren andere Applikationen normalerweise bestimmte Funktionalität (im Gegensatz zum Löschen von Dateien im Preferences-Ordner, die nicht unentbehrlich sein sollten).

Der Application Support Ordner wird von Mac OS 8 und neuer unterstützt.

Unter Linux liefert ApplicationSupportFolder ein FolderItem für das „Home“-Verzeichnis des Benutzers zurück. Normalerweise erzeugt eine Applikation ein Unterverzeichnis, wenn Konfigurationsdateien oder ähnliches gespeichert werden müssen. Diese Unterverzeichnisse tragen denselben Namen wie die Applikation. „MeinProgramm“ würde zum Beispiel das Verzeichnis „/home/username/MeinProgramm“ benutzen.

Beispiel

Folgendes Beispiel zeigt den vollständigen Pfad des Application Support Ordners an, falls ein solcher existiert.

```
Dim f as FolderItem
f=ApplicationSupportFolder
If f <> Nil then
  MsgBox f.absolutePath
else
  MsgBox "Keinen Application Support Ordner gefunden."
end if
```

Siehe auch

[DesktopFolder](#), [ControlPanelsFolder](#), [ExtensionsFolder](#), [FontsFolder](#), [PreferencesFolder](#), [ShutDownItemsFolder](#), [StartupItemsFolder](#), [SystemFolder](#), [TemporaryFolder](#), [TrashFolder](#) Funktionen.

ArcShape Klasse

Wird zum Zeichnen von Bögen in einer Vektorgrafik-Umgebung verwendet.

Übergeordnete Klasse

[OvalShape](#)

Eigenschaften

Name	Typ	Beschreibung
ArcAngle	<u>Double</u>	Der Winkel des Bogen im Bogenmaß.
StartAngle	<u>Double</u>	Der Start-Winkel im Bogenmaß.

Hinweise

Ein Start-Winkel von 0 bedeutet genau östlich, bzw. ein positiver X-Wert, wobei Y=0 ist. Positive Winkel sind im Uhrzeigersinn, so wie bei Object2D.rotation. Der Wert von ArcAngle kann positiv oder negativ sein, ausgehend vom Start-Winkel. Wenn der Winkel ausgefüllt ist, wird ein Keil erzeugt; dies kann zum Beispiel für das Produzieren von Tortengrafiken nützlich sein.

Beispiele

Folgende Methode erzeugt einen Bogen, wenn sie im Paint-Event eines Fensters aufgerufen wird.

```
Dim a as ArcShape
a=New ArcShape
d=New Group2D
a.arcAngle=1.57
a.startAngle=-1.57
a.fillcolor=RGB(255,0,127)
g.drawobject d,100,80
```

Dieser Bogen sieht so aus:



Siehe auch

CurveShape, FigureShape, FolderItem, Group2D, Graphics, Object2D, OvalShape, Picture, PixmapShape, RectShape, RoundRectShape, StringShape Klassen.

Array Funktion

Weist den fortlaufenden Elementen jeweils einen Wert aus einer Liste zu.

Syntax

Ergebnis=Array(ElementListe)

Teil	Typ	Beschreibung
Ergebnis	Jeder gültige Datentyp für ein Array	Name des Arrays, welches mit den Werten der ElementList gefüllt wurde.
ElementList	Derselbe wie Ergebnis	Eine komma-getrennte Liste mit Werten, mit denen das Ergebnis befüllt wird. Der Datentyp der Listenelemente muß mit dem Datentyp des Ergebnis-Arrays übereinstimmen.

Hinweise

Die Array Funktion bietet die gleiche Funktionalität wie die separate Zuweisung von Werten für jedes einzelne Element des Arrays, die mit Element Null beginnt. Befinden sich mehr Elemente in dem Array als in der Liste, werden diese nicht verändert.

Beispiel

Das folgende Beispiel erzeugt ein Array und füllt die ersten drei Elemente.

```
Dim aNames() as String
//using the Array function
aNames=Array("Fred","Ginger","Stanley")
```


Siehe auch

[Dim](#) Ausdruck; [Split](#), [Join](#), [Ubound](#) Funktionen; [Append](#), [Insert](#), [Redim](#), [Remove](#), [Pop](#), [IndexOf](#), [Insert](#), [Shuffle](#), [Sort](#) Methoden; [ParamArray](#) Schlüsselwort.

Arraygrenzen müssen Integer sein Fehler

Sie haben in einem Dim-Befehl beim Deklarieren eines Arrays keinen Integer verwendet. Sie können Integer-Konstanten verwenden, um Arraygrenzen festzulegen aber keine Konstanten anderen Typs.

Beispiele

```
Dim aPicture (5.2) as String
```

```
Dim aProps (True) as Boolean
```

Asc Funktion

Gibt den ASCII-Wert des ersten Buchstabens eines [Strings](#) als [Integer](#) zurück.

Syntax

Ergebnis=[Asc](#)([string](#))

Teil	Art	Beschreibung
<i>Ergebnis</i>	Integer	Der ASCII Wert des ersten Zeichens von <i>string</i> .
<i>string</i>	String	Ein beliebiger gültiger String-Ausdruck.

oder

Ergebnis=[StringVariable](#).[Asc](#)

Teil	Art	Beschreibung
<i>Ergebnis</i>	Integer	Der ASCII Wert des ersten Zeichens von <i>string</i> .
<i>string</i>	String	Ein beliebiger gültiger String-Ausdruck.
<i>StringVariable</i>	String	Ein beliebige Variable vom Typ String .

Hinweise

Die **Asc**-Funktion gibt den Zeichen-Code des ersten Zeichens des übergebenen [Strings](#) zurück. Zeichen von 0 bis 127 gehören zum Standard-ASCII-Zeichensatz. Sie sind auf praktisch jeder Plattform identisch. Asc liefert den Zeichen-Code für das Encoding, das dem String zugrunde liegt. Ist der String mit MacRoman kodiert, so erhalten Sie den Zeichen-Code entsprechend der MacRoman-Kodierung.

Wenn Sie den ASCII-Code des ersten Bytes eines Strings statt des ersten Zeichens benötigen, verwenden Sie hierzu bitte die [AscB](#)-Funktion.

Beispiele

Dieses Beispiel verwendet die Funktion **Asc**, um den ASCII-Wert eines Zeichens zu ermitteln.

```
Dim a as Integer  
a = Asc("@") //ergibt 64
```

Dieses Beispiel ermittelt den Zeichen-Code für das „≥“ Symbol:

```
Dim s as String  
Dim n as Integer  
s="≥"  
n=s.Asc
```

Siehe auch

[Chr](#), [InStr](#), [Left](#), [Len](#), [Mid](#), [Right](#) Funktionen; [TextEncoding](#) Klasse.

AscB Funktion

Gibt den Wert des ersten Bytes eines Strings als Integer zurück.

Syntax

Ergebnis=**AscB**(string)

oder

Ergebnis=**StringVariable.AscB**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Der Wert des ersten Bytes von <i>string</i> .
<i>string</i>	<u>String</u>	Ein beliebiger gültiger String-Ausdruck.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ String.

Hinweis

Die **AscB**-Funktion gibt den Code des ersten Bytes des übergebenen Strings zurück. Wenn Sie den Character-Code des ersten Zeichens eines Strings statt des ersten Bytes benötigen, verwenden Sie hierzu bitte die Asc-Funktion.

AscB sollte anstelle von Asc verwendet werden, wenn der String aus binären Daten besteht oder wenn Ihr Programm auf einem System mit Ein-Byte-Zeichensatz (z.B. US-System) läuft und Sie Case-Sensitivität wollen.

Beispiele

Dieses Beispiel verwendet die Funktion **AscB**, um den Wert eines Zeichens zu ermitteln.

```
MsgBox Str(AscB("a")) //ergibt 97
MsgBox Str(AscB("A")) //ergibt 65
```

Siehe auch

Asc, ChrB, InStrB, LeftB, LenB, MidB, RightB Funktionen.

Asin Funktion

Gibt den Arkus-Sinus des übergebenen Wertes zurück.

Syntax

Ergebnis=**Asin**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der Arkus-Sinus von Wert.
<i>Wert</i>	<u>Double</u>	Der Wert, dessen Arkus-Sinus Sie ermitteln wollen. Wert ist der <i>Sinus</i> des Winkels, den Sie möchten und muß zwischen -1 und 1 sein.

Hinweis

Der Arkus-Sinus ist der Winkel dessen Sinus *Wert* entspricht. Die **Asin**-Funktion gibt den Winkel (im Bogenmaß) des übergebenen Sinus zurück. Um den Arkus-Sinus im Gradmaß zu bekommen, müssen Sie das Ergebnis mit 180/Pi multiplizieren.

Beispiele

Folgendes Beispiel verwendet die **Asin**-Funktion, um den Arkus-Sinus einer Zahl zu ermitteln.

```
Dim d as Double
Const Pi=3.14159265358979323846264338327950
d=Asin(.5) //ergibt 0.5235988
d=Asin(.5)*180/Pi //ergibt 30
```

Siehe auch

Sin Funktion.

Assigns Schlüsselwort

Wird verwendet, um anzuzeigen, daß ein Parameter an eine Methode über den Assignment-Operator übergeben wird.

Syntax

Assigns Parameter As DataType

Teil	Typ	Beschreibung
<i>Parameter</i>		Der Name des zu deklarierenden Parameters.
<i>DataType</i>	Beliebiger gültiger Datentyp	Der Datentyp des Parameters.

Hinweise

Verwenden Sie das **Assigns**-Schlüsselwort, wenn Sie dem Parameter einer Methode einen Wert mit dem Gleichheitszeichen zuweisen wollen und nicht als ein Argument.

Das **Assigns**-Schlüsselwort kann in einer Method-Deklarations-Dialogbox für den einzigen der Methode übergebenen Parameter erscheinen, oder, wenn mehr als ein Parameter übergeben wurde, für den letzten Parameter. Wenn **Assigns** verwendet wird, müssen Sie beim Aufruf der Methode eine andere Syntax verwenden. Der Wert für den Parameter muß unter Verwendung des Assignment-Operator und nicht als ein Argument übergeben werden.

Beispiel

Folgende Deklaration verwendet **Assigns** für den letzten Parameter:

```
Sub theVolume(a as Integer, b as Integer, Assigns c as Integer)
```

Wenn diese Methode aufgerufen wird, werden nur die Werte für a und b als Argumente übergeben; der Wert für c wird unter Verwendung des Assignment-Operator übergeben. Zum Beispiel:

```
theVolume(5,4)=10
```

Wenn Sie es mit folgender Syntax versuchen:

```
theVolume(5,4,10)
```

erhalten Sie einen Syntax-Fehler.

Siehe auch

Assigns kann nur auf den letzten Parameter angewendet werden Fehler.

Assigns kann nur auf den letzten Parameter angewendet werden Fehler

Sie haben das Assigns-Schlüsselwort in einer Methoden-Deklaration für einen anderen als den letzten Parameter verwendet. Wenn es nur einen einzigen Parameter gibt, ist die Verwendung von Assigns erlaubt.

Beispiel

Folgende Deklaration verwendet das Assigns-Schlüsselwort für den ersten Parameter.

```
Sub SquareIt(Assigns x as Double, y as Double)
```

Siehe auch

Assigns Schlüsselwort.

Atan Funktion

Gibt den Arkus-Tangens des übergebenen Wertes zurück. Der Arkus-Tangens ist der Winkel, dessen Tangens *Wert* entspricht.

Syntax

Ergebnis = **Atan**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der Arkus-Tangens von Wert.
<i>Wert</i>	<u>Double</u>	Der Wert, dessen Arkus-Tangens Sie ermitteln wollen.

Hinweis

Die **Atan**-Funktion gibt den Winkel (im Bogenmaß) der übergebenen Zahl zurück. Um den Arkus-Tangens im Gradmaß zu bekommen, müssen Sie das Ergebnis mit 180/Pi multiplizieren.

Beispiele

Dieses Beispiel verwendet die **Atan**-Funktion, um den Arkus-Tangens einer Zahl zu ermitteln.

```
Dim d as Double
Const Pi=3.14159265358979323846264338327950
d=Atan(1) //ergibt 0.785398 (Pi/4 im Bogenmaß)
d=Atan(1)*180/Pi // ergibt 45
```

Siehe auch

[Tan](#), [Atan2](#) Funktionen.

Atan2 Funktion

Gibt den Arkus-Tangens des Punktes mit den Koordinaten x und y zurück. Der Arkus-Tangens ist der Winkel zur x-Achse, zu einer Linie, die durch den Ursprung (0,0) und durch den Punkt mit den Koordinaten x,y geht.

Syntax

Ergebnis=**Atan2**(y,x)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der Arkus-Tangens des Punktes (y, x) im Bogenmaß.
<i>y</i>	<u>Double</u>	y-Koordinate des Punktes.
<i>x</i>	<u>Double</u>	x-Koordinate des Punktes.

Hinweis

Das Ergebnis wird im Bogenmaß geliefert. Um es ins Gradmaß zu konvertieren, multiplizieren Sie es mit 180/Pi. Die umgekehrten Operationen werden mit Cos und Sin vorgenommen. Wenn Sie also einen Winkel haben und ein x/y-Paar auf der Linie benötigen, die durch 0,0 und x,y beschrieben wird, können Sie dies so erreichen:

```
x=Cos(Winkel)*Radius
y=Sin(Winkel)*Radius
```

Beispiele

Dieses Beispiel verwendet die **Atan2**-Funktion, um den Arkus-Tangens eines Punktes direkt oberhalb des Ursprungs zu finden.

```
Dim d as Double
const Pi=3.14159265358979323846264338327950
d=Atan2(1,0) //ergibt 1.57
d=Atan2(1,0)*180/Pi //ergibt 90
```

Siehe auch

[Tan](#), [Atan](#) Funktionen.

AutoDiscovery Klasse

Die **AutoDiscovery**-Klasse benutzt die [EasyUDPSocket](#)-Klasse, um automatisch nach anderen REALbasic-Applikationen im Netzwerk zu suchen.

Übergeordnete Klasse

[EasyUDPSocket](#)

Methoden

Name	Parameter	Beschreibung
GetMemberList		Liefert ein String -Array mit IP-Adressen der Maschinen, die zu der aktuellen Gruppe gehören.

Name	Parameter	Beschreibung
SendMessageToGroup	Befehl als <u>Integer</u> Daten als <u>String</u>	Sendet eine Nachricht, die den Befehl und die Daten enthält, an die Gruppe.
UpdateMemberList		Löscht die interne Liste der verbundenen Gruppenmitglieder und ermittelt die Mitglieder in dem Netzwerk neu.

Events

Name	Parameter	Beschreibung
MemberJoined	IP-Adresse als <u>String</u>	Die Maschine mit der IP-Adresse wurde gefunden.
MemberLeft	IP-Adresse als <u>String</u>	Die Maschine mit der IP-Adresse ist verschwunden.

Hinweise

Die **AutoDiscovery**-Klasse gibt Ihnen die Möglichkeit, andere Maschinen im Netzwerk zu finden, die ebenfalls über die EasyUDP-Socket-Klasse verbunden sind. Diese werden anhand des Gruppennamen ermittelt, der über die Register-Funktion einer EasyUDP-Socket zugewiesen werden kann. Wird ein neues Gruppenmitglied entdeckt, wird ein MemberJoined-Event mit der IP-Adresse der Maschine ausgelöst. Dies geschieht auch, wenn Ihre Applikation die Register-Funktion aufruft. Verschwindet ein Gruppenmitglied, wird ein MemberLeft-Events, ebenfalls mit der IP-Adresse, ausgelöst. Sie erhalten eine Liste der momentan vorhanden Gruppenmitglieder über die GetMemberList-Methode. Diese können Sie über die UpdateMemberList-Methode aktualisieren. Dabei wird die interne Liste gelöscht und alle Gruppenmitglieder neu ermittelt.

Die **AutoDiscovery** -Klasse soll die Netzwerkkommunikation so einfach wie möglich gestalten. Deshalb wird hier ein proprietäres Protokoll verwendet. Damit sind Sie nicht in der Lage, andere Dienste im Netzwerk zu ermitteln, zum Beispiel iChat. Hierfür benutzen Sie bitte die generischen Klassen wie TCPSocket, UDPSocket, HTTPSocket und so weiter.

Beispiel

Als Gruppenmitglied registrieren. Um einer Gruppe beizutreten, müssen die gleichen Gruppennamen bei der Registrierung verwendet werden.

```
AutoDiscovery1.Register("MyGroup")  
AutoDiscovery1.UpdateMemberList
```

Ermittelt die aktuelle Mitgliederliste und zeigt diese in einer ListBox an.

```
Dim s(-1) as String  
Dim i as Integer  
s= AutoDiscovery1.GetMemberList  
For i=0 to Ubound(s)  
    ListBox1.addrow s(i)  
Next
```

Eine Gruppe verlassen.

```
AutoDiscovery1.Unregister("MyGroup")
```

Eine Nachricht an die Gruppe senden.

```
AutoDiscovery1.SendMessageToGroup(100,"Hello world")
```

Empfangen einer Nachricht über den ReceivedMessage-Event:

```
Sub ReceivedMessage (FromIP as String, Command as Integer, Data as String)  
    MsgBox FromIP + " sent us " + Str(command) + ": " + Data  
End Sub
```

Siehe auch

EasyUDPSocket, UDPSocket-Klassen.

Beep Methode

Lässt den System-Warnton erklingen.

Syntax

Beep

Hinweis

Die **Beep**-Methode spielt den Warnton, den Sie im Monitore&Ton-Kontrollfeld eingestellt haben.

Beispiel

Dieses Beispiel spielt den Warnton dreimal ab.

```
Beep
Beep
Beep
```

BevelButton Steuerelement



Hiermit kann man einen BevelButton (Multifunktionsbutton) erzeugen. Ein BevelButton kann Text, Grafik, ein Popup-Menü oder Kombinationen verschiedener dieser Interface-Elemente gleichzeitig beinhalten.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Typ	Beschreibung
<i>Bevel</i>	<u>Integer</u>	Der Bevelstil. Die Bevel-Eigenschaft beeinflusst das Aussehen und, in manchen Fällen, die Form eines BevelButtons. Beispiele der möglichen Bevel-Stile finden Sie im Benutzerhandbuch. 0 - Kleiner Bevel 1 - Normaler Bevel 2 - Großer Bevel 3 - Abgerundeter Bevel (nur Max OS X) 4 - Kein Bevel (nur Windows XP; Hinweise beachten) 5 - Rund (nur Max OS X) 6 - Rund groß (nur Mac OS X) 7 - Aufklapdreieck (nur Mac OS X) Für Bevel-Stile, die es nur unter Mac OS X gibt, wird auf anderen Plattformen der Kleine Bevel benutzt.
<i>Bold</i>	<u>Boolean</u>	Weist dem Button-Text das Textattribut fett zu.
<i>ButtonType</i>	<u>Integer</u>	0 - Button. Bleibt in der Gedrückt-Position, bis die Maustaste losgelassen wird. 1 - Toggles. Bleibt in der Gedrückt-Position, bis er erneut angeklickt wurde. 2 - Sticky. Bleibt in der Gedrückt-Position, nachdem er angeklickt wurde.
<i>Caption</i>	<u>String</u>	Der Text des Buttons.
<i>CaptionAlign</i>	<u>Integer</u>	Die Ausrichtung des Buttontextes. 0 - Linksbündig 1 - Rechtsbündig 2 - Systemausrichtung 3 - Zentriert
<i>CaptionDelta</i>	<u>Integer</u>	Abstand des Textes in Pixeln vom linken Rand des Buttons.

Name	Typ	Beschreibung
<i>CaptionPlacement</i>	<u>Integer</u>	Ausrichtung der Textes relativ zur Grafik (Icon). 0 - Systemausrichtung 1 - Normal 2 - Rechts von der Grafik 3 - Links von der Grafik 4 - Unter der Grafik 5 - Über der Grafik
<i>Icon</i>		Name der Grafik, die als Icon verwendet wird. Ziehen Sie die Grafik in das Projektfenster oder importieren Sie die Datei über den Menüpunkt „Ablage/Importieren“. Es werden auch Grafiken unterstützt, die zur Laufzeit mit der <u>NewPicture</u> Funktion erzeugt werden. Transparenz wird ebenfalls unterstützt. Setzen Sie hierfür die Transparenz-Eigenschaft des <u>Picture</u> -Objekts auf 1 oder über die Mask-Eigenschaft. Beachten Sie hierzu auch das erste Beispiel.
<i>IconAlign</i>	<u>Integer</u>	Ausrichtung der Grafik innerhalb des Bevel-Buttons 0 - Systemausrichtung 1 - Zentriert 2 - Linksbündig 3 - Rechtsbündig 4 - Oben 5 - Unten 6 - Oben links 7 - Unten links 8 - Oben rechts 9 - Unten rechts
<i>IconDx</i>	<u>Integer</u>	Abstand in Pixeln nach links oder rechts, je nach Ausrichtung. Falls diese zentriert ist, bewirkt IconDx nichts.
<i>IconDy</i>	<u>Integer</u>	Abstand in Pixeln nach oben oder unten, je nach Ausrichtung. Falls diese zentriert ist, bewirkt IconDy nichts.
<i>Italic</i>	<u>Boolean</u>	Weist dem Button-Text das Textattribut kursiv zu.
<i>LockBottom</i>	<u>Boolean</u>	Bestimmt, ob die untere Kante des Steuerelements einen konstanten Abstand zum unteren Ende des Fensters, in dem es sich befindet, beibehalten soll.
<i>LockLeft</i>	<u>Boolean</u>	Bestimmt, ob die linke Kante des Steuerelements einen konstanten Abstand zum linken Rand des Fensters, in dem es sich befindet, beibehalten soll. LockLeft bewirkt nichts, falls LockRight <u>True</u> ist.
<i>LockRight</i>	<u>Boolean</u>	Bestimmt, ob die rechte Kante des Steuerelements einen konstanten Abstand zum rechten Rand des Fensters, in dem es sich befindet, beibehalten soll.
<i>LockTop</i>	<u>Boolean</u>	Bestimmt, ob die obere Kante des Steuerelements einen konstanten Abstand zum oberen Rand des Fensters, in dem es sich befindet, beibehalten soll. LockTop bewirkt nichts, falls LockBottom <u>True</u> ist.
<i>HasMenu</i>	<u>Integer</u>	Bestimmt, ob der BevelButton ein Popup-Menü besitzt. 0 - Kein Menü 1 - Normales Menü 2 - Menü rechts
<i>MenuValue</i>	<u>Integer</u>	Die Nummer des Menüeintrags, den der Benutzer wählt. Der erste Menüeintrag hat die Nummer Null. Ein Separator kann nicht gewählt werden, „zählt“ aber als Menüeintrag.
<i>TextFont</i>	<u>String</u>	Name des Fonts, der für den Button-Text verwendet wird.
<i>TextSize</i>	<u>Integer</u>	Größe des Fonts, der für den Button-Text verwendet wird.
<i>Underline</i>	<u>Boolean</u>	Weist dem Button-Text das Textattribut unterstrichen zu.

Name	Typ	Beschreibung
<i>Value</i>	<u>Boolean</u>	Wenn <u>True</u> , erscheint der Button anfangs, als wenn er gedrückt wäre.

Methoden

Name	Parameter	Beschreibung
<i>AddRow</i>	Text als <u>String</u>	Fügt eine Zeile am Ende des Menüs hinzu und verwendet den Text als Menüeintrag.
<i>AddSeparator</i>		Fügt eine Zeile am Ende des Menüs hinzu und erzeugt einen Separator als Menüeintrag.
<i>DeleteAllRows</i>		Löscht alle Zeilen des Menüs des aktuellen BevelButtons.
<i>InsertRow</i>	Row als <u>Integer</u> Text als <u>String</u>	Fügt eine Zeile an der Position <i>Row</i> ein und verwendet <i>Text</i> als Menüeintrag.
<i>List</i>	Row als <u>Integer</u>	Liefert den Text des Menüeintrags als String zurück. Der erste Menüeintrag hat die Nummer Null.
<i>RemoveRow</i>	Row als <u>Integer</u>	Entfernt die Zeile <i>Row</i> aus dem Menü. Der erste Menüeintrag hat die Nummer Null.

Hinweise

Ein Bevel-Button kann eine Grafik, einen Text oder beides darstellen.

Die erste Zeile eines Menüs hat den Index Null.

Wenn die Caption-Eigenschaft ein Und-Zeichen enthält, wird dieses nur dargestellt, wenn ein weiteres Und-Zeichen folgt. Wenn Sie also beispielsweise die Caption-Eigenschaft auf „Robert & Petra“ setzen, wird nur „Robert Petra“ angezeigt. Sie müssen sie auf „Robert && Petra“ setzen. Das Und-Zeichen wird unter Windows für Tastaturkürzbehele verwendet.

Dies wurde so gelöst, damit sich Macintosh- und Windows-Programme gleich verhalten.

Die „Kein Bevel“-Option für die Bevel-Eigenschaft ist für das XP-Betriebssystem bestimmt. Der Knopf erscheint ohne Rahmen, bis der Mauszeiger den Bereich des Knopfs betritt. Dann erscheint ein Rahmen um den Knopf herum. Für alle anderen Betriebssysteme verhält sich die „Kein Bevel“-Option wie die Option „Kleiner Bevel“.

Beispiele

Das folgende Beispiel plziert ein Icon auf einem Bevel-Button, richtet es aus und legt die Ausrichtung des Button-Textes fest. Die Grafik „databasequery“ wurde in das Projektfenster gedraggt.

```
me.icon=databasequery
me.iconalign=6
me.captionalign=0
me.captionplacement=2
me.icondx=2
me.icondy=1
```

Das folgende Beispiel erzeugt ein BevelButton-Menü

```
Dim s as string
Dim i as integer
Dim last as integer
s="Macintosh,Windows,Unix, - , Andere"
last=CountFields(s,",")
me.caption="Plattform"
me.captionalign=0 //linksbündig
me.hasMenu=2 //Menü rechts
for i=1 to last
me.addRow NthField(s,",",i)
Next
```

Wenn der Benutzer den „Andere“-Menüpunkt wählt, bekommt MenuValue den Wert 4, da der Separator als ein Menüpunkt zählt, auch wenn er nicht ausgewählt werden kann.

Folgende Zeile Code im Action-Event-Handler des BevelButton setzt die Beschriftung des BevelButton auf den Wert, den der User auswählt:

```
Me.caption=Me.list(Me.MenuValue)
```

Siehe auch

PushButton, PopupMenu, ContextualMenu Objekte.

Bezeichner erwartet Fehler

REALbasic erwartet in einem Dim-Befehl einen Variablennamen.

Beispiele

Es folgen zwei Fehlerarten bei einem Dim-Befehl.

```
Dim As Boolean //kein Variablenname
Dim Double as Double //'Double' ist reserviert
```

Bin Funktion

Liefert die Binärversion der übergebenen Zahl als String.

Syntax

Ergebnis = **Bin**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Wert, ins Binäre konvertiert.
<i>Wert</i>	<u>Integer</u>	Die Zahl, die ins Binäre konvertiert werden soll.

Hinweise

Wenn der Wert keine ganze Zahl ist, wird der Dezimalteil abgeschnitten. Sie können Binär-, Hexadezimal- oder Oktal-Zahlen festlegen, indem Sie der Zahl den &-Operator und den Buchstaben für die Zahlen-Basis voranstellen. Der Buchstabe „b“ legt die Zahl als Binär-, „h“ als Hexadezimal- und „o“ als Oktal-Zahl fest.

Beispiele

```
Dim binVersion As String
binVersion=Bin(15) //liefert "1111"
binVersion=Bin(15.5) //liefert "1111"
binVersion=Bin(75) //liefert "1000000"
binVersion=Bin(&hF) //liefert "1111"
```

Siehe auch

& Operator; Oct, Hex Funktionen.

BinaryStream Klasse

Objekte der BinaryStream-Klasse werden verwendet, um Daten in eine Binärdatei zu schreiben oder aus einer zu lesen. Die Vorteile bei der Verwendung von BinaryStreams sind, daß man an beliebiger Stelle in der Datei lesen und schreiben kann. Text-Dateien müssen kontinuierlich von Anfang an gelesen werden.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>EOF</i>	<u>Boolean</u>	Der Datenstrom hat das Ende der Datei erreicht.
<i>Length</i>	<u>Integer</u>	Die Länge des Data-Fork der Datei.
<i>LittleEndian</i>	<u>Boolean</u>	Wenn True, wird als Byte-Reihenfolge Low-Byte, High-Byte genommen.
<i>Position</i>	<u>Integer</u>	Die aktuelle Position im Binary-Stream (Datenstrom). Der Wert dieser Eigenschaft wird automatisch bei allen Lese- und Schreiboperationen erhöht.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Schließt den Datenstrom (und folglich auch die Datei, die als Stream geöffnet wurde).

Name	Parameter	Beschreibung
<i>Read</i>	Count als <u>Integer</u> [Encoding als <u>TextEncoding</u>]	Liest <i>Count</i> Zeichen aus dem Stream, beginnend an der Eigenschaft Position und gibt diese als <u>String</u> zurück. Der optionale Parameter <i>Encoding</i> erlaubt es Ihnen das Text-Encoding für den zurückgelieferten String festzulegen. Verwenden Sie das Encodings-Objekt, um das Text-Encoding festzulegen. Wenn Sie diesen Parameter weglassen (oder <u>Nil</u> übergeben), ist das Encoding des zurückgelieferten Strings <u>Nil</u> .
<i>ReadBoolean</i>		Liest einen Boolean-Wert aus dem Stream und gibt es als ein <u>Boolean</u> zurück.
<i>ReadByte</i>		Liest ein Byte aus dem Stream und gibt es als ein <u>Integer</u> zurück.
<i>ReadDouble</i>		Liest ein Doppelwort aus dem Stream und gibt es als <u>Double</u> zurück.
<i>ReadLong</i>		Liest ein Langwort (4 Bytes) aus dem Stream und gibt dies als <u>Integer</u> zurück.
<i>ReadSingle</i>		Liest ein Wort aus dem Stream und gibt es als <u>Double</u> zurück.
<i>ReadPString</i>	[Encoding als <u>TextEncoding</u>]	Liest einen Pascal-String aus dem Stream und gibt ihn als <u>String</u> zurück. Der optionale Parameter <i>Encoding</i> erlaubt es Ihnen das Encoding für den zurückgelieferten String festzulegen. Verwenden Sie das Encodings-Objekt, um das Text-Encoding festzulegen. Wenn Sie diesen Parameter weglassen (oder <u>Nil</u> übergeben), ist das Encoding des zurückgelieferten Strings <u>Nil</u> .
<i>ReadShort</i>		List ein Kurzwort (2 Bytes) aus dem Stream und gibt es als <u>Integer</u> zurück.
<i>Write</i>	Data als <u>String</u>	Schreibt <i>Data</i> in die Datei, beginnend an der Eigenschaft Position.
<i>WriteBoolean</i>	Wert als <u>Boolean</u>	Schreibt den übergebenen Wert als ein Boolean in den Stream.
<i>WriteByte</i>	Wert als <u>Integer</u>	Schreibt den übergebenen Wert als ein Byte in den Stream.
<i>WriteDouble</i>	Value als <u>Double</u>	Schreibt den übergebenen Wert als ein Doppelwort in den Stream.
<i>WriteLong</i>	Wert als <u>Integer</u>	Schreibt den übergebenen Wert als ein Langwort (4 Bytes) in den Stream.
<i>WriteSingle</i>	Value als <u>Double</u>	Schreibt den übergebenen Wert als ein Wort in den Stream.
<i>WritePString</i>	Text als <u>String</u>	Schreibt den übergebenen Text als Pascal-String in den Stream.
<i>WriteShort</i>	Wert als <u>Integer</u>	Schreibt den übergebenen Wert als ein Kurzwort (2 Bytes) in den Stream.

Hinweis

Was bedeutet die Eigenschaft LittleEndian?

Manche Betriebssysteme speichern Binärwerte in umgekehrter Reihenfolge als das Mac OS. Wenn Sie die Methoden *ReadShort* oder *ReadLong* verwenden, um Daten aus einer Datei zu lesen, die im Little-Endian-Format vorliegt, bekommen Sie falsche Werte. *REALbasic* liest Daten im Big-Endian-Format. Die meisten Macintosh-Dateien liegen im Big-Endian-Format vor.

Wenn Sie eine Datei lesen wollen, die im Little-Endian-Format vorliegt, müssen Sie diese Eigenschaft auf *True* setzen, bevor Sie damit beginnen, die Datei einzulesen. Dies gilt auch für das Schreiben von Daten mit *WriteShort* und *WriteLong*.

Der Wert 258 wird beispielsweise im Big-Endian-Format (wie im Mac OS) so gespeichert:

01 02

Im Little-Endian-Format würde er so gespeichert:

02 01

Wenn die Eigenschaft *LittleEndian* falsch gesetzt wäre, würden Sie als Wert 513 erhalten.

Da *REALbasic* die Eigenschaft *LittleEndian* bereitstellt, können Sie Ihren Code plattformunabhängig gestalten. Setzen Sie die *Little-Endian*-Eigenschaft auf True, wenn das Dateiformat wirklich *LittleEndian* ist (z.B. bei GIF-Dateien), andernfalls lassen Sie ihn auf False.

Die **BinaryStream**-Klasse implementiert bereits das Klassen-Interface zum Lesen und Schreiben. Implementieren Sie Ihr eigenes Interface, so müssen Sie eine *Read*- und *Write*-Methode implementieren, die der folgenden Beschreibung entspricht.

Ergebnis=**Read** (Anzahl, [Kodierung])

Teil	Typ	Beschreibung
Ergebnis	<u>String</u>	Der Text, der von der Methode zurückgegeben wird.
Anzahl	<u>Integer</u>	Anzahl der Bytes, die gelesen werden sollen.

Teil	Typ	Beschreibung
Kodierung	TextEncoding	Ein optionaler Parameter, der die Kodierung des einzulesenden Texts bestimmt.

Das Klassen-Interface zum Lesen enthält zwei Methoden.

Name	Parameter	Beschreibung
EOF		Liefert True wenn der Stream das Ende der Datei erreicht hat.
ReadError		Liefert True wenn ein Fehler aufgetreten ist.

Write(Daten)

Teil	Typ	Beschreibung
Daten	String	Der Text, der in die Datei geschrieben werden soll.

Das Klassen-Interface zum Schreiben besitzt die folgenden Methoden:

Name	Parameter	Beschreibung
Flush		Leert alle internen Puffer und schreibt die Daten auf das Volume.
WriteError		Liefert True wenn ein Fehler während einer Schreiboperation auftrat.

Weitere Informationen zum Thema Klassen-Interface finden Sie im Benutzerhandbuch.

Beispiel

Dieses Beispiel liest jedes Byte-Paar aus einer Datei und schreibt sie in umgekehrter Reihenfolge in eine neue Datei. Der Anwender wählt die Quelldatei mit der Dateiauswahlbox und speichert die neue Datei mit Hilfe der „Sichern als“-Dialogbox.

```
Dim WriteToFile as BinaryStream
Dim ReadFromFile as BinaryStream
Dim f as FolderItem
f=GetOpenFolderItem("text")
If f <> Nil Then
    ReadFromFile=f.OpenAsBinaryFile(false)
    ReadFromFile.littleEndian=true
    f=GetSaveFolderItem("", "")
    If f <> Nil Then
        WriteToFile=f.CreateBinaryFile("text")
        While Not ReadFromFile.EOF
            WriteToFile.WriteShort ReadFromFile.ReadShort
        wend
        WriteToFile.close
    End If
    ReadFromFile.close
End If
```

Siehe auch

[TextInputStream](#), [TextOutPutStream](#), [FolderItem](#) Klassen.

Bitwise Klasse

Führt Bit-Operationen mit Integeren durch. Die Methoden der **Bitwise**-Klasse ersetzen die Funktionen [BitwiseAnd](#), [BitwiseOr](#) und [BitwiseXor](#).

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
<i>BitAnd</i>	Wert1 als Integer Wert2 als Integer	Liefert einen Integer . Führt eine bitweise Und-Verknüpfung aus Wert1 und Wert2 durch.
<i>BitOr</i>	Wert1 als Integer Wert2 als Integer	Liefert einen Integer . Führt eine bitweise Oder-Verknüpfung aus Wert1 und Wert2 durch.

Name	Parameter	Beschreibung
<i>BitXor</i>	Wert1 als <u>Integer</u> Wert2 als <u>Integer</u>	Liefert einen <u>Integer</u> . Führt eine bitweise XOr-Verknüpfung aus Wert1 und Wert2 durch.
<i>OnesComplement</i>	Wert als <u>Integer</u>	Führt eine Komplementärberechnung für <i>Wert</i> durch. Jedes Bit des Integer wird umgedreht. Null wird zu Eins und umgekehrt.
<i>ShiftLeft</i>	Wert als <u>Integer</u> shift als <u>Integer</u> [,numBits als <u>Integer</u>]	Shiftet <i>Wert</i> um <i>shift</i> nach links. Shiftet nur Bits innerhalb der Feldgröße <i>numBits</i> . Die Voreinstellung von <i>numBits</i> ist 32Bit.
<i>ShiftRight</i>	Wert als <u>Integer</u> shift als <u>Integer</u> [,numBits als <u>Integer</u>]	Shiftet <i>value</i> um <i>shift</i> nach rechts. Shiftet nur Bits innerhalb der Feldgröße <i>numBits</i> . Die Voreinstellung von <i>numBits</i> ist 32Bit.

Hinweise

Die Funktionen BitwiseAnd, BitwiseOr und BitwiseXor sind veraltet. Benutzen Sie stattdessen die entsprechende Methode der **Bitwise**-Klasse.

Sie müssen keine Instanz der Bitwise-Klasse erzeugen, um auf ihre Methoden zugreifen zu können. Es ist ein spezielles Objekt, wie System oder Application, die immer existieren.

Die BitAnd-Methode liefert ein integer, bei dem die zwei Integer-Argumente bitweise verglichen sind und an jede Stelle, bei der es eine Übereinstimmung gab, eine 1 geschrieben wurde. Unterscheiden sich die Bits, wird eine 0 geschrieben.

Die BitOr-Methode liefert ein Integer, bei dem die zwei Integer-Argumente bitweise verglichen sind und an jede Stelle, bei der eines der Bits den Wert 1 hatte, eine 1 geschrieben wurde. Hatte keines der Bits den Wert 1, wurde eine 0 geschrieben.

Die BitXor-Methode liefert ein Integer, bei dem die zwei Integer-Argumente bitweise verglichen sind und an jede Stelle, bei der die Bits einen ungleichen Wert hatten, eine 1 geschrieben wurde.

Das Einer-Komplement wird häufig dazu verwendet, um positive und negative Werte zu ermöglichen. Positive Werte beginnen mit einer Null und negative mit einer Eins. Das einzige Problem hierbei ist, daß eine Null doppelt interpretiert wird.

Dezimal	Einer-Komplement	Dezimal mit Vorzeichen
0	000	0
1	001	1
2	010	2
3	011	3
4	100	-3
5	101	-2
6	110	-1
7	111	-0

Beispiel

```
Dim i as Integer
i=Bitwise.bitand(5,3) //liefert 1
i=Bitwise.bitor(5,3) //liefert 7
i=Bitwise.bitxor(5,3) //liefert 6
```

Siehe auch

BitwiseAnd, BitwiseOr, BitwiseXor Funktionen.

BitwiseAnd Funktion

Führt eine bitweise Und-Verknüpfung (BitWiseAnd) der zwei übergebenen Werte aus und gibt das Ergebnis zurück. Diese Funktion ist veraltet. Bitte benutzen Sie die entsprechende Methode der Bitwise-Klasse.

Siehe auch

Bitwise-Klasse.

BitwiseOr Funktion

Führt eine bitweise Oder-Verknüpfung (BitwiseOr) der beiden übergebenen Werte aus und gibt das Ergebnis zurück. Diese Funktion ist veraltet. Bitte benutzen Sie die entsprechende Methode der [Bitwise](#)-Klasse.

Siehe auch

[Bitwise Klasse](#).

BitwiseXor Funktion

Führt eine bitweise Entweder-Oder-Verknüpfung (BitwiseXor) der beiden übergebenen Werte aus und gibt das Ergebnis zurück. Diese Funktion ist veraltet. Bitte benutzen Sie die entsprechende Methode der [Bitwise](#)-Klasse.

Siehe auch

[Bitwise Klasse](#).

Boolean Datentyp

Boolean ist ein wesentlicher Datentyp in REALbasic. Ein **Boolean** kann nur den Wert True oder False (Wahr oder Falsch) annehmen.

Boolean-Werte sind zunächst False und können mit dem True-Schlüsselwort von REALbasic auf True eingestellt und mit dem False-Schlüsselwort wieder zurück auf False geändert werden. Manche Eigenschaften von Objekten in REALbasic sind boole'sche Werte. Im Eigenschaften-Fenster erscheinen diese als Checkboxes. Zum Beispiel haben die meisten Steuerelemente eine Enabled-Eigenschaft, die vom Typ **Boolean** ist.

Die [VarType](#)-Funktion liefert 11, wenn eine Boolean-Variable übergeben wird.

Beispiel

Folgender Befehl deaktiviert ein EditField, so daß man nicht in dieses hineinkommt und es nicht selektieren kann (der Anwender kann es mit der Tab-Taste nicht anspringen).

```
EditField1.Enabled=False
```

Siehe auch

[True](#), [False](#), [VarType](#) Funktionen; [Dim](#)-Ausdruck.

Bounds3D Klasse

Repräsentiert einen abgrenzenden Raum, nützlich für einfache Tests auf Kollision und Sichtbarkeit in [RB3DSpace](#)-Steuerelementen. Ein **Bounds3D** kann entweder als eine Achsen-ausgerichtete umgrenzende Box und/oder eine umgrenzende Kugel repräsentiert werden.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Center</i>	Vector3D	Das Zentrum der Kugel oder der Box.
<i>Maximum</i>	Vector3D	Das Maximum x, y, und z oder die umgrenzende Box.
<i>Minimum</i>	Vector3D	Das Minimum x, y, und z oder die umgrenzende Box.
<i>Radius</i>	Double	Der Radius der umgrenzenden Kugel.

Name	Typ	Beschreibung
<i>Type</i>	<u>Integer</u>	Der Typ des umgrenzenden Objekts 1 = Kugel 2 = Box 0 = Ungültig.

Methoden

Name	Parameter	Beschreibung
<i>Intersects</i>	Anderer als Bounds3D	Liefert ein <u>Boolean</u> . Testet, ob dieser umgrenzende Raum einen anderen überlappt. Im Zweifelsfall liefert es <u>True</u> .
<i>InView</i>	Rb3D als <u>RB3DSpace</u>	Liefert ein <u>Boolean</u> . Testet, ob der Bounds3D-Raum zumindest teilweise im sichtbaren Bereich des RB3DSpace <i>RB3D</i> ist. Im Zweifelsfall liefert es <u>True</u> .
<i>ContainsPoint</i>	Pt als <u>Vector3D</u>	Liefert ein <u>Boolean</u> . Testet, ob der umgrenzende Raum den 3D-Punkt <i>Pt</i> enthält.
<i>LineIntersection</i>	Pt1 als <u>Vector3D</u> Pt2 als <u>Vector3D</u>	Liefert ein <u>Vector3D</u> . Sucht den Punkt, an dem die Linie durch <i>Pt1</i> und <i>Pt2</i> den umgrenzenden Raum schneidet. Wenn er ihn nicht schneidet, liefert es <u>Nil</u> .
<i>LineSegmentIntersection</i>	Pt1 als <u>Vector3D</u> Pt2 als <u>Vector3D</u>	Liefert ein <u>Vector3D</u> . Sucht den Punkt, an dem eine Linie durch <i>Pt1</i> und <i>Pt2</i> den umgrenzenden Raum schneidet. Wenn der Schnittpunkt nicht zwischen <i>Pt1</i> und <i>Pt2</i> ist, liefert es <u>Nil</u> .

Konstruktor-Syntax

Diese Tabelle zeigt, wie man entweder eine umgrenzende Box oder eine Kugel erzeugt.

Syntax	Parameter	Beschreibung
<i>New Bounds3D</i>	Zentrum als <u>Vector3D</u> Radius als <u>Double</u>	Der Constructor für eine Bounds3D -Kugel.
<i>New Bounds3D</i>	Minimum als <u>Vector3D</u> Maximum als <u>Vector3D</u>	Der Constructor für eine Bounds3D -Box.

Hinweise

Wie in der Methoden-Tabelle hingewiesen wurde, sind manche Testes nicht exakt; in schwer zu entscheidenden Grenzfällen liefert **Bounds3D** True, wodurch angezeigt wird, daß zwei Objekte überlappen oder ein Objekt sichtbar ist (InView). Das bedeutet, daß ein Rückgabewert von True anzeigen kann, daß genauere Tests nötig sind. Ein Rückgabewert von False jedoch zeigt an, daß keine weiteren Tests nötig sind.

Derzeit behandelt der Linien-Schnittpunkt-Test alle **Bounds3D**-Räume als umgrenzende Kugeln.

Siehe auch

RB3DSpace, Vector3D Klassen.

Break Ausdruck

Läuft die Applikation aus der IDE heraus, kann die Applikation mit einem **Break** angehalten werden und die IDE wechselt in den Debugger.

Syntax

Break

Hinweise

Der **Break**-Ausdruck hat denselben Effekt, als würden Sie einen Breakpoint im Code-Editor setzen. Break-Ausdrücke werden in Standalone-Applikationen ignoriert.

Siehe auch

[Function](#), [Sub](#) Ausdrücke.

ByRef Schlüsselwort

Wird verwendet, um einen Parameter als Referenz zu übergeben.

Syntax

ByRef Parameter

Teil	Typ	Beschreibung
<i>Parameter</i>	Beliebiger Datentyp	Parameter, der als Referenz übergeben werden soll. Parameter kann ein Array sein.

Hinweis

Normalerweise übergeben Sie Parameter an Methoden als Wert (by value). Wenn Sie dies so machen, erhält die Methode eine Kopie der Daten in dem Objekt, das Sie übergeben. Ihre Methode erhält die Daten und kann Operationen damit ausführen.

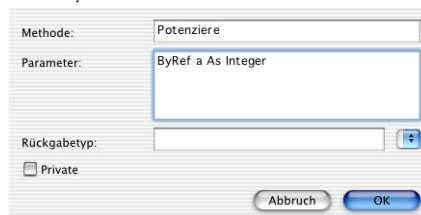
Um einen Parameter als Referenz zu deklarieren, benutzen Sie das ByRef-Schlüsselwort, wenn Sie die Methode im Dialog „Neue Methode“ deklarieren. Wenn Sie bei der Definition der Methode dem Parameter das Schlüsselwort **ByRef** voranstellen, übergeben Sie die Informationen als Referenz. Wenn Sie die Information als Referenz übergeben, übergeben Sie lediglich einen Zeiger auf das Objekt, das die Informationen beinhaltet. Der praktische Vorteil dieser Technik ist, daß die Methode die Werte von jedem Parameter ändern kann.

Wenn Sie die Parameter als Wert (by value) übergeben, können Sie dies nicht, da die Parameter nur eine Kopie der Daten selbst darstellen.

Wenn es sich bei einem Parameter um ein Array handelt, funktioniert ByRef genau so wie bei normalen Parametern. Ihre Methode kann das komplette Array ersetzen oder einzelne Elemente.

Beispiel

Die folgende Methodendeklaration verwendet ByRef:



Wenn Sie nun OK klicken, um die Methode zu speichern, zeigt Ihnen der Sub-Ausdruck im Code-Editor, daß der Parameter als Referenz übergeben wird.

```
Protected Sub powers(ByRef a As Integer)
End Sub
```

Die Potenziere-Methode bekommt einen Parameter, einen Integer a, übergeben, was ByRef genannt wird.

Die Methode ist einfach:

```
a=a*a
```

Potenzen wird folgendermaßen aufgerufen:

```
Dim a as integer
```

```
a=3
Potenziere a
Editfield1.text=Str(a)
```

Das `EditField` zeigt den Wert 9 an.

Siehe auch

[ByVal](#) Schlüsselwort.

ByVal Schlüsselwort

Wird verwendet, um einen Parameter als Wert zu übergeben.

Syntax

ByVal Parameter

Teil	Typ	Beschreibung
<i>Parameter</i>	Beliebiger Datentyp	Parameter, der als Wert übergeben werden soll.

Hinweis

ByVal kann in der Methoden-Deklarations-Dialogbox verwendet werden, um festzulegen, daß ein Parameter als Wert (by value) übergeben werden soll. Normalerweise werden Parameter als Wert übergeben. Daher ist das Schlüsselwort **ByVal** optional.

Per `ByVal` übergebene Parameter werden wie lokale Variablen innerhalb der Methode behandelt – genau so wie Variablen, die mit dem `Dim`-Befehl erzeugt wurden. Das bedeutet, daß Sie die Werte der Parameter selbst verändern können anstatt zuerst den Parameter einer lokalen Variablen zuzuweisen. Wenn Sie z.B. einen Wert im Parameter „x“ übergeben, können Sie den Wert von „x“ entweder erhöhen oder erniedrigen anstatt den Wert von x einer lokalen Variablen zuzuweisen, die mit einem `Dim`-Befehl erzeugt wurde.

Siehe auch

[ByRef](#) Schlüsselwort.

Call Befehl

Erlaubt Ihnen, eine Funktion aufzurufen, ohne sich um den Wert zu kümmern, den diese zurückliefert.

Syntax

Call FunktionsName

Teil	Beschreibung
<i>FunktionsName</i>	Beliebige Funktion

Hinweise

Verwenden Sie den **Call**-Befehl nur, wenn Sie eine Funktion aufrufen wollen, deren Rückgabewert Sie nicht verwenden. Das erlaubt es Ihnen, die Funktion ihre Arbeit verrichten zu lassen, ohne extra eine lokale Variable definieren zu müssen, die das Funktionsergebnis speichert.

Beispiel

Die folgenden Aufrufe rufen die `SelectColor`-Funktion auf, ohne deren Ergebnisse ([Boolean](#)) zu verwenden.

```
Dim c as Color
c=CMY(.35,.9,.6) //wähle Default-Farbe aus der Farbwahl aus
Call SelectColor(c,"Wähle eine Farbe")
Rectangle1.FillColor=c
```

Siehe auch

[Function](#) Befehl.

Canvas Steuerelement



Canvas-Steuerelemente sind sehr nützlich, wenn Sie Ihre eigenen grafischen Steuerelemente einbauen möchten, da Sie die Zeichen-Befehle der Graphics- oder der Object2D-Klasse verwenden können, um innerhalb des Canvas-Bereichs zu zeichnen. Es kann auch verwendet werden, um vorhandene Grafiken, wie z.B. das ImageWell, anzuzeigen.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>AcceptFocus</i>	<u>Boolean</u>	Falls <u>True</u> , kann das Steuerelement den Fokus erhalten. Es kann die <u>GotFocus</u> - und <u>LostFocus</u> -Events akzeptieren und einen Fokus-Rahmen darstellen, wenn <u>UseFocusRing</u> auf <u>True</u> gesetzt ist. Die Voreinstellung ist <u>False</u> .
<i>AcceptTabs</i>	<u>Boolean</u>	Falls <u>AcceptTabs</u> und <u>AcceptFocus</u> <u>True</u> sind, löst das Drücken der Tab-Taste das <u>KeyDown</u> -Event zur weiteren Verarbeitung aus. Wenn es <u>False</u> ist, löst das Drücken der Tab-Taste das <u>KeyDown</u> -Event nicht aus; das Drücken von Tab löst das <u>LostFocus</u> -Event aus und selektiert das nächste Objekt im Fenster, das den Fokus besitzen kann.
<i>Backdrop</i>	<u>Picture</u>	Das Bild, das automatisch in diesem Bereich gezeichnet wird. Muß auf dem Macintosh ein PICT und unter Windows ein BMP sein, außer, der Anwender hat QuickTime installiert.
Graphics	<u>Graphics</u>	Erlaubt Zugriff auf alle <u>Graphics</u> -Methoden, um in diesem Bereich zu zeichnen. Egal, welche Objekte mit der <u>Graphics</u> -Methode gezeichnet werden, sie werden in einer separaten Ebene gezeichnet, nachdem das <u>Backdrop</u> -Image gezeichnet worden ist. Dies erlaubt es Ihnen, Objekte auf ein <u>Backdrop</u> -Image zu legen.
<i>UseFocusRing</i>	<u>Boolean</u>	Falls <u>True</u> , zeigt das Canvas mit einem Rahmen um seinen Rand an, daß es den Fokus besitzt; falls <u>False</u> , ändert sich das Aussehen des Objekts nicht, wenn es den Fokus bekommt. Das Setzen dieser Eigenschaft auf <u>True</u> hat keinen Effekt unter Windows. Die Voreinstellung ist <u>True</u> .

Events

Name	Parameter	Beschreibung
<i>EnableMenuItems</i>		Das Canvas-Steuerelement hat den Fokus erhalten. Menu-Handler werden aufgerufen, wenn das Canvas-Steuerelement den Fokus erhalten hat.
<i>GotFocus</i>		Das Canvas-Steuerelement hat den Fokus erhalten.
<i>KeyDown</i>	Taste als <u>String</u>	Die Taste, die an das Event übergeben wurde, wurde gedrückt, während das Canvas-Steuerelement den Fokus hatte.
<i>LostFocus</i>		Das Canvas-Steuerelement hat den Fokus verloren.
<i>MouseDown(Function)</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Canvas-Bereichs an der Position x,y gedrückt. Geben Sie <u>True</u> zurück, wenn Sie das <u>MouseDown</u> -Event auswerten wollen.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Canvas-Steuerelements gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im <u>MouseDown</u> -Event <u>True</u> zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Canvas-Bereichs an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im <u>MouseDown</u> -Event <u>True</u> zurückliefern.

Name	Parameter	Beschreibung
<i>Paint</i>	g als <u>Graphics</u>	Der Bereich muß neu gezeichnet werden. Der übergebene Parameter bietet Ihnen Zugriff auf die Graphics-Klassen-Methoden, um zu zeichnen. Verwenden Sie anstelle des Open-Event den Paint-Event, wenn Sie möchten, daß Objekte, die mit den Graphics-Klassen-Methoden gezeichnet werden, permanent sind.

Methoden

Name	Parameter	Beschreibung
<i>Scroll</i>	DeltaX als <u>Integer</u> , DeltaY als <u>Integer</u> , [Left als <u>Integer</u>], [Top als <u>Integer</u>], [Breite als <u>Integer</u>], [Höhe als <u>Integer</u>], [ScrollControls als <u>Boolean</u>]	DeltaX und DeltaY sind die Anzahl von Pixeln, die horizontal und vertikal, relativ zur aktuellen Position des Bildes im Canvas-Steuerelement, gescrollt werden soll. Mit positiven Werten scrollt man nach rechts und unten, während man mit negativen Werten nach links oder oben scrollt. Die Parameter Left, Top, Width und Height (Links, Oben, Breite und Höhe) geben den Ausschnitt des Bildes an, der gescrollt werden soll. Werden diese nicht angegeben, wird das gesamte Bild gescrollt. Mit ScrollControls wird festgelegt, ob Steuerelemente, die auf einem Canvas-Steuerelement positioniert sind, mitgescrollt werden sollen. Der Wert von ScrollControl hat als Default den Wert <u>True</u> .

Hinweis

Koordinaten, die Canvas-Events übergeben werden, sind im Canvas-Objekt lokal. Um mit der Scroll-Methode das Bild in einem Canvas-Steuerelement zu scrollen, müssen Sie die letzten Scroll-Werte für die Achsen, an denen Sie scrollen, speichern, damit Sie den Umfang, um den gescrollt werden soll, berechnen können. Dies können Sie durch Hinzufügen einer Eigenschaft zu dem Fenster erreichen, das das Canvas-Steuerelement enthält oder durch Erzeugen einer neuen Klasse, die auf dem Canvas-Steuerelement basiert, die Eigenschaften enthält, um den letzten X- und den letzten Y-Scrollwert zu speichern. Wenn der Parameter ScrollControls True ist, wird jedes Steuerelement auf dem Canvas-Steuerelement mitgescrollt. Das erlaubt die Implementation eines scrollenden Feldes mit Steuerelementen.

Beispiele

Dieses Beispiel zeichnet ein 3D-Rechteck mit erhöhtem Aussehen.

```
Sub Paint(g as Graphics)
    g.forecolor=RGB(255, 255, 255) //weiß
    g.drawline 1,1,canvas1.width,1
    g.drawline 1,canvas1.height-1,1,1
    g.forecolor=RGB(140,140,140) //dunkelgrau
    g.drawline canvas1.width-1,2,canvas1.width-1,canvas1.height
    g.drawline 1,canvas1.height-1,canvas1.width,canvas1.height-1
    //füllt das hellgraue Rechteck
    g.forecolor=RGB(239,239,239)
    g.fillrect 2,2,canvas1.width-3,canvas1.height-3
End Sub
```

Dieses Beispiel weist der Backdrop-Eigenschaft ein Bild zu, das in das Projekt-Fenster gedraggt wurde:

```
me.backdrop=OSLogo
```

Sie können dann die Methoden der Graphics-Klasse verwenden, um das Bild in beliebiger Weise zu modifizieren. Beispielsweise fügt folgender Code im Paint-Event-Handler des Objekts dem Bild eine Beschriftung hinzu:

```
me.Graphics.TextFont="Helvetica"
me.Graphics.TextSize=14
me.Graphics.ForeColor=RGB(255,255,255)
me.Graphics.DrawString "Joe Bob",10,100
```

Wenn Sie stattdessen die Grafik der BackDrop-Eigenschaft mittels NewPicture wie unten gezeigt zuweisen, können Sie die Grafik auf Pixel-Ebene mit der RGBSurface-Eigenschaft des Picture-Objekts manipulieren.

```
Me.backdrop=NewPicture(160,120,32)
Me.backdrop.Graphics.DrawPicture Honey,0,0
```

Sehen Sie auch bei den Beispielen der Control-Klasse nach, die Ihnen einen Eindruck vermitteln, wie man von einem Canvas-Control draggt und das Beispiel der ImageWell-Klasse, welches Drag&Drop auf und von einer ImageWell.Image-Eigenschaft zeigt.

Einen Fokus-Rahmen unter Windows simulieren

Leider erscheint der Fokus-Rahmen nicht in Win32-Programmen, wenn ein Canvas-Steuerelement den Fokus bekommt, die Events GotFocus und LostFocus reagieren aber normal. Sie können somit einen Fokus-Rahmen recht einfach simulieren.

Verwenden Sie hierzu im GotFocus-Event:

```
#If TargetWin32 then
    Me.Graphics.forecolor=HighlightColor //oder FrameColor, was immer Sie wollen
    Me.Graphics.DrawRect 0,0,me.width-1,me.height-1
#endif
```

Verwenden Sie hierzu im LostFocus-Event:

```
#If TargetWin32 then
    Me.Graphics.forecolor=RGB(178,178,178) //grau
    Me.Graphics.DrawRect 0,0,me.width-1,me.height-1
#endif
```

Der simulierte Fokus-Rahmen sieht etwa so aus:



Siehe auch

Graphics, Picture Klassen; NewPicture Funktion.

CDBl Funktion

Diese Funktion entspricht der Val-Funktion, wird jedoch verwendet, wenn man einen String übergeben muß, der ein anderes Zeichen als den Punkt (.) als Dezimaltrenner verwendet. Es wird das Zeichen verwendet, das im Zahlenformat-Kontrollfeld eingestellt ist.

Mehr Informationen dazu finden Sie unter der Val-Funktion.

Siehe auch

CStr, Str, Val Funktionen.

Ceil Funktion

Gibt den angegebenen Wert, gerundet auf den nächstgrößeren Integer, zurück.

Syntax

Ergebnis = **Ceil** (Wert)

Teil	Art	Beschreibung
<i>Wert</i>	<u>Double</u>	Der Wert, den Sie Ceil runden möchten.
<i>Ergebnis</i>	<u>Double</u>	Der Ceil-gerundete <i>Wert</i> .

Hinweis

Die **Ceil**-Funktion gibt den übergebenen Wert, gerundet auf den nächstgrößeren Integer, zurück.

Beispiele

Dieses Beispiel verwendet die **Ceil**-Funktion um die Ceil-Rundung einer Zahl zurückzugeben.

```
Dim d as double
d=Ceil(1.234) //ergibt 2
```

Siehe auch

Floor, Round Funktionen.

ChasingArrows Steuerelement



Fügt einem Fenster ChasingArrows (sich verfolgende, drehende Pfeile) hinzu. Unter Mac OS X ist dies ein animiertes Zahnrad.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach. Verwenden Sie die Visible-Eigenschaft des Steuerelements, um es anzuzeigen oder um es zu verstecken. Sobald es sichtbar ist, wird es animiert, die Pfeile drehen sich also.

Checkbox Steuerelement



Das Steuerelement für Standard-Checkboxes.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>Bold</i>	<u>Boolean</u>	Weist der Beschriftung der Checkbox den Schriftstil fett zu.
<i>Caption</i>	<u>String</u>	Der Textinhalt der Checkbox.
<i>DataField</i>	<u>String</u>	Nur relevant, wenn die CheckBox in Verbindung mit einem <u>DataControl</u> verwendet wird, um den Inhalt von Feldern einer Datenbanktabelle anzuzeigen. Der Name eines Feldes in der Tabelle, auf die das <u>DataControl</u> verweist.
<i>DataSource</i>	<u>DataControl</u>	Das <u>DataControl</u> , das auf eine Tabelle in einer Datenbank verweist, dessen Felder dargestellt werden. Verwenden Sie die DataField-Eigenschaft, um ein in der CheckBox anzuzeigendes Feld zu verbinden. In der IDE erscheint ein Popup-Menü mit den Feldnamen im Eigenschaftfenster. Wenn Sie es im Code einstellen, erwartet es einen String. Weisen Sie ihn der Name-Eigenschaft eines <u>DataControl</u> zu. Hinweis: Datenbank-Felder können auch mittels <u>EditField</u> -, <u>ListBox</u> - und <u>StaticText</u> -Steuerelementen dargestellt werden.
<i>Italic</i>	<u>Boolean</u>	Weist der Beschriftung der Checkbox den Schriftstil kursiv zu.
<i>TextFont</i>	<u>String</u>	Name des Fonts, mit dem der Knopftext dargestellt werden soll.
<i>TextSize</i>	<u>Integer</u>	Größe des Fonts, mit dem der Knopftext dargestellt werden soll.
<i>Underline</i>	<u>Boolean</u>	Weist der Beschriftung des Knopfes den Schriftstil unterstrichen zu.
<i>Value</i>	<u>Boolean</u>	Der Wert der Checkbox, wenn das zugehörige Fenster geöffnet wird.

Events

Name	Parameter	Beschreibung
<i>Action</i>		Das Steuerelement wurde angeklickt.
<i>GotFocus</i>		(Win32 und Linux) Das Steuerelement hat den Fokus bekommen und besitzt einen Selektionsrahmen um die Beschriftung.
<i>LostFocus</i>		(Win32 und Linux) Das Steuerelement hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y gedrückt. Wenn Sie MouseDown auswerten möchten, liefern Sie <u>True</u> zurück. Das Action-Event wird nicht ausgeführt und der Status des Objekts ändert sich nicht

Name	Parameter	Beschreibung
<i>MouseUp</i>	x als <u>Integer</u> y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht <u>True</u> im MouseDown-Event liefern. Der Rückgabewert wird ignoriert.

Hinweis

Wenn die Caption-Eigenschaft ein Und-Zeichen enthält, wird dieses nur dargestellt, wenn ein weiteres Und-Zeichen folgt. Wenn Sie also beispielsweise die Caption-Eigenschaft auf „Butter & Brot“ setzen, wird nur „Butter Brot“ angezeigt. Sie müssen sie auf „Butter && Brot“ setzen.

Dies wurde so gelöst, damit sich Macintosh- und Windows-Programme gleich verhalten. Das Und-Zeichen wird unter Windows für Tastaturkurzbefehle verwendet. Näheres finden Sie im Entwicklerhandbuch unter „Verwenden von Konstanten, um Menüs und Menüeinträgen Windows-Tastatur-Shortcuts hinzuzufügen“.

Beispiel

Folgender Code im Action-Event einer CheckBox prüft den Wert der CheckBox:

```
If CheckBox1.value then
    EditField1.text="Wahr"
else
    EditField1.text="Falsch"
end if
```

Siehe auch

RadioButton Steuerelement.

Chr Funktion

Gibt das Zeichen zurück, dessen ASCII-Wert übergeben wurde.

Syntax

Ergebnis = **Chr**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Das Zeichen, dessen ASCII-Wert übergeben wurde.
<i>Wert</i>	<u>Integer</u> , <u>Single</u> oder <u>Double</u>	Der numerische Wert (Zeichen-Code) des gewünschten Zeichens, in einem Bereich zwischen 0-127.

Hinweis

Die **Chr** Funktion liefert das Zeichen, dessen Zeichen-Code angegeben wurde. Diese Funktion ist nur für ASCII-Zeichen gültig. Das Ergebnis ist undefiniert, wenn Sie einen Zeichen-Code außerhalb des Bereichs von 0 bis 127 angeben.

Sie sollten grundsätzlich die Chr-Methode der TextEncoding-Klasse verwenden. Dort geben Sie die Kodierung und den Zeichen-Code in dieser Kodierung an. Bitte beachten Sie die Beispiele für die TextEncoding-Klasse und die letzte Zeile im folgenden Beispiel. Die **Chr**-Funktion funktioniert für den speziellen Fall, in dem die Kodierung ASCII und der Wert kleiner 128 ist.

Die **Chr**-Funktion liefert einen Ein-Byte- String, wenn sie auf Systemen ausgeführt wird, die nur Ein-Byte-Strings unterstützen. Andernfalls liefert sie einen Zwei-Byte- String. Benötigen Sie einen Ein-Byte-String unter einem Zwei-Byte-System, so benutzen Sie bitte die Funktion ChrB.

Beispiele

Diese Beispiele verwenden die **Chr**-Funktion, um die Zeichen zu erzeugen, deren ASCII-Werte übergeben werden.

```
Dim Tab, CR as String
Tab=Chr(9) //erzeugt ein Tabulator-Zeichen
CR=Chr(13) //erzeugt ein Carriage Return
```

Siehe auch

Asc, InStr, Left, Len, Mid, Right, EndOfLine, Encoding Funktionen; TextEncoding Klasse; Encodings Objekt.

ChrB Funktion

Gibt einen Single-Byte-String des Werts zurück, der übergeben wurde.

Syntax

Ergebnis=ChrB(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der Single-Byte-String, dessen Wert übergeben wurde.
<i>Wert</i>	<u>Integer</u> , <u>Single</u> oder <u>Double</u>	Der Wert des gewünschten Zeichens.

Hinweis

Die **ChrB**-Funktion gibt einen Single-Byte-String des Werts zurück, der übergeben wurde. **ChrB** sollte anstelle von Chr verwendet werden, wenn es sich bei *Wert* um binäre Daten handelt. **ChrB** sollte anstelle von Chr verwendet werden, wenn *Wert* aus binären Daten besteht.

Wenn Sie einen Single-Byte-String benötigen, wenn das Programm auf einem System mit Single-Byte Systemsoftware läuft und einen Doppel-Byte-String, wenn es auf einem Doppel-Byte-System läuft, verwenden Sie bitte die Chr Funktion.

Beispiele

Diese Beispiele verwenden die **ChrB** Funktion, um die Zeichen mit den übergebenen ASCII-Werten zu ermitteln.

```
Dim s as string
s=ChrB(32) //liefert ein Leerzeichen
s=ChrB(13) //liefert ein Carriage Return
```

Siehe auch

AscB, Chr, InStrB, LeftB, LenB, MidB, RightB, EndOfLine, Encoding Funktionen; TextEncoding Klasse; Encodings Objekt.

ClearFocus Methode

Entfernt den Fokus von dem Steuerelement, das den Fokus hat, so daß kein Steuerelement mehr den Fokus hat.

Syntax

ClearFocus

Hinweise

Der Aufruf von ClearFocus hat den gleichen Effekt wie der Aufruf der Window-Klassen-Methode SetFocus. Sie können den Fokus über die SetFocus-Eigenschaft der RectControl-Klasse setzen. Bitte beachten Sie, daß die SetFocus-Methode der RectControl-Klasse nichts bewirkt, wenn das Steuerelement, der sie zugeordnet ist, nicht in der Lage ist, den Fokus auf der Plattform, auf der der Code läuft, zu bekommen.

Beispiel

Folgender Code im Action-Event eines PushButton entfernt den Fokus von dem Objekt im Fenster, das den Fokus besitzt.

```
ClearFocus
```

Siehe auch

Window, RectControl Klassen.

Clipboard Klasse

Wird verwendet, um Informationen in die Zwischenablage zu schreiben und von dort zu lesen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Picture</i>	<u>Picture</u>	Das Bild in der Zwischenablage.
<i>PictureAvailable</i>	<u>Boolean</u>	Ergibt <u>True</u> , wenn die Zwischenablage ein Bild enthält.
<i>Text</i>	<u>String</u>	Der Text in der Zwischenablage.

Methoden

Name	Parameter	Beschreibung
<i>AddMacData</i>	Data als <u>String</u> , MacTyp als <u>String</u>	Kopiert die angegebenen Binärdaten im angegebenen Format in die Zwischenablage.
<i>Close</i>		Schließt die Zwischenablage. Die Zwischenablage muß innerhalb des Event-Handlers, in dem sie geöffnet wurde, wieder geschlossen werden, oder es wird ein Fehler auftreten.
<i>MacData</i>	MacTyp als <u>String</u>	Gibt die Binärdaten aus der Zwischenablage als einen <u>String</u> im angegebenen Format (Typ) zurück.
<i>MacDataAvailable</i>	MacTyp als <u>String</u>	Ergibt True, falls die Zwischenablage Daten des angegebenen Typs enthält.
<i>SetText</i>	<u>String</u>	Schreibt den angegebenen Text in die Zwischenablage.
<i>TextAvailable</i>	<u>Boolean</u>	Ergibt <u>True</u> , wenn die Daten in der Zwischenablage Text sind.

Hinweis

Um Text aus der Zwischenablage zu lesen oder ihn dorthin zu schreiben, müssen Sie ein Objekt vom Typ Clipboard erzeugen und dann auf die *Text*-Eigenschaft dieses Objekts zugreifen. Sie können ein Bild in die Zwischenablage befördern, indem Sie die *Picture*-Eigenschaft des Clipboard-Objekts setzen. Mit den Methoden *SetText* und *AddMacData* können Sie Text oder Binärdaten in die Zwischenablage schreiben.

Beispiele

Dieses Beispiel holt den Text aus der Zwischenablage und kopiert ihn in eine Variable.

```
Dim c as Clipboard
Dim s as String
c=New Clipboard
s=c.text
c.close
```

Dieses Beispiel schreibt Text in die Zwischenablage.

```
Dim c as Clipboard
c=New Clipboard
c.text="Hallo Welt!"
c.close
```

Dieses Beispiel überprüft, ob sich in der Zwischenablage Text befindet. Falls ja, kopiert es diesen Text in ein EditField.

```
Dim c as Clipboard
c=New Clipboard
If c.TextAvailable then
    editfield1.text=c.Text
End if
c.close
```

Folgendes Beispiel kopiert ein PICT-Bild in die Zwischenablage:

```
Dim c as Clipboard
c=New Clipboard
If ImageWell1.image <> Nil then
    c.picture=ImageWell1.image
    c.close
else
    MsgBox "Kein Bild verfügbar!"
end if
```

Wenn Sie mehr als ein Objekt in der Zwischenablage ablegen möchten, können Sie die Eigenschaften der Klasse nicht verwenden, um einen Text oder Grafik zusätzlich in die Zwischenablage abzulegen. Im folgenden Beispiel wird nur ein Text in der Zwischenablage abgelegt:

```
Dim c as New Clipboard
c.text="This is the first item being put on the clipboard!"
c.text="This is the second item being put on the clipboard"
c.close
```

Sie müssen stattdessen die Texte zusammenfügen und dann als einzelnen Text in der Zwischenablage ablegen.

CMY Funktion

Liefert ein Color, das auf dem CMY-Farbmodell (cyan, magenta, yellow) basiert.

Syntax

Ergebnis = **CMY**(cyan, magenta, yellow)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Ein Objekt, das die Farbe repräsentiert, die sich aus den angegebenen Werten für <i>cyan</i> , <i>magenta</i> und <i>yellow</i> ergibt.
<i>cyan</i>	<u>Double</u>	Der Wert von Cyan in der Farbe (0-1).
<i>magenta</i>	<u>Double</u>	Der Wert von Magenta in der Farbe (0-1).
<i>yellow</i>	<u>Double</u>	Der Wert von Yellow in der Farbe (0-1).

Hinweis

Die **CMY** Funktion liefert ein Color, das auf den übergebenen Werten für Cyan, Magenta und Yellow basiert. Diese Werte sind Doubles zwischen 0 und 1 und werden intern als drei Bytes gespeichert. Sie können ebenso entweder das RGB- oder HSV-Modell nehmen, um eine Farbe zuzuweisen.

Beispiele

Dieses Beispiel verwendet die **CMY**-Funktion, um der fillcolor-Eigenschaft eines Rectangle-Steuerelements ein Color zuzuweisen.

```
rectangle1.fillcolor=CMY(.35,.9,.6)
```

Siehe auch

Color Datentyp, HSV, RGB, SelectColor Funktionen; &c Operator.

Collection Klasse

Dient zum Speichern einer Menge zusammengehörender Elemente in einer größeren Struktur. Das Konzept der Collection ähnelt einem Array, mit dem Unterschied, daß jedes Element einer Collection einen anderen Datentyp haben kann. Eine Collection kann man als ein Array von Variants ansehen. Darüber hinaus kann jedes Element einer Collection einen Namen bekommen. Auf Elemente einer Collection kann über den Namen oder die Nummer zugegriffen werden.

Die Dictionary-Klasse bietet die gesamte Funktionalität der Collection-Klasse und besitzt weitere Vorteile: Bei der Collection-Klasse hängt die Zeit, um ein Objekt zu finden, von der Anzahl der Objekte in der Collection ab, da die Suche sequentiell erfolgt. Die Dictionary-Klasse verwendet eine Hash-Tabelle, bei der die Zeit relativ unabhängig von der Anzahl der Objekten ist. Sie wurde für schnelle Suche entwickelt. Außerdem ist der Schlüssel-Parameter in der Dictionary-Klasse ein Variant und in der Collection-Klasse ein String. Daher ist es empfehlenswert, sofern möglich, die Dictionary-Klasse der Collection-Klasse vorzuziehen.

Die Collection-Klasse ist nur aus Kompatibilitätsgründen zu früheren Versionen und zu VisualBasic-Projekten vorhanden.

Übergeordnete Klasse

Object

Methoden

Name	Parameter	Beschreibung
<i>Add</i>	Value als <u>Variant</u> Key als <u>String</u>	Value wird der Collection als letztes Element hinzugefügt; wenn Key angegeben wird, kann dieser verwendet werden, um mit der Item-Eigenschaft auf dieses Element zu verweisen.
<i>Count</i>		Anzahl der Elemente in der Collection. Liefert einen <u>Integer</u> -Wert.
<i>Item</i>	Index als <u>Integer</u> ODER Key als <u>String</u>	Verweist auf ein Element der Collection. Liefert ein <u>Variant</u> . Der Index ist 1-basiert.
<i>Remove</i>	Index als <u>Integer</u> ODER Key als <u>String</u>	Entfernt das angegebene Element aus der Collection. Der Index ist 1-basiert.

Beispiel

Das folgende Beispiel erzeugt eine Collection, füllt diese mit Strings und numerischen Werten und zeigt jedes Element in einem EditField oder Canvas-Steuerelement an (das Bild „lois“ wurde zum Projekt hinzugefügt). Beachten Sie, daß eine Collection sehr stark einem Datensatz einer Datenbank ähnelt.

```
Dim c as Collection
c=New Collection
c.add 1,"ID"
c.add "Lois Lane", "Name"
c.add "Reporter", "JobTitle"
c.add 85000,"Salary"
c.add lois,"Picture"
Editfield1.text=c.item("ID")
Editfield2.text=c.item(2) // liefert "Lois Lane"
Editfield3.text=c.item("JobTitle")
Editfield4.text=c.item("Salary")
Canvas1.backdrop=c.item("Picture")
```

Wenn Sie die Methoden Item oder Remove verwenden wollen, um auf einen Eintrag zu verweisen, schließen Sie die Parameter, die dieser Methode übergeben werden, in Klammern ein. Dies ist nötig, da der Compiler nicht weiß, welchen Datentyp Sie übergeben. Verwenden Sie also zum Beispiel

```
c.remove("Name")
```

anstelle von

```
c.remove "Name"
```

um den Datensatz „Lois Lane“ aus der Collection zu entfernen.

Siehe auch

Variant, Dictionary Klassen; VarType Funktionen, Nil Schlüsselwort.

Color Datentyp

Ein Color ist ein wesentlicher Datentyp, der aus drei Bytes besteht, die eine Farbe definieren. Eine Farbe kann über die RGB-, HSV-, oder CMY-Farbmodelle oder den &c-Operator definiert werden. Für jedes Farbmodell werden drei Eigenschaften gesetzt, die zusammen die gewünschte Farbe beschreiben. Die Werte von jeder der neun Eigenschaften können dann ausgelesen werden.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
Blue	<u>Integer</u>	Der Blauanteil (0-255) der Farbe (RGB).
Green	<u>Integer</u>	Der Grünanteil (0-255) der Farbe (RGB).
Red	<u>Integer</u>	Der Rotanteil (0-255) der Farbe (RGB).
Hue	<u>Double</u>	Der Farbton (0-1) der Farbe (HSV).

Name	Typ	Beschreibung
Saturation	<u>Double</u>	Die Sättigung (0-1) der Farbe (HSV).
Value	<u>Double</u>	Die Helligkeit (0-1) der Farbe (HSV).
Cyan	<u>Double</u>	Der Cyan-Anteil (0-1) der Farbe (CMY).
Magenta	<u>Double</u>	Der Magenta-Anteil (0-1) der Farbe (CMY).
Yellow	<u>Double</u>	Der Yellow-Anteil (0-1) der Farbe (CMY).

Hinweis

Colors werden meist verwendet, um Eigenschaften des Typs *Color* eine Farbe zuzuweisen. Verwenden Sie die RGB-, HSV-, oder CMY-Funktion, um einem Objekt eine Farbe zuzuweisen oder verwenden Sie das Format:

```
&cRRGGBB
```

wobei *RR* der RGB-Wert von Rot in Hexadezimal ist, *GG* der Wert von Grün in Hexadezimal und *BB* der Wert von Blau in Hexadezimal ist. Um die Hex-Werte einfach zu erhalten, verwenden Sie die Neue Konstante-Dialogbox, um eine Color-Konstante über den eingebauten Farb-Picker zu erzeugen. Wählen Sie die gewünschte Farbe visuell aus. Wenn Sie dies machen, errechnet REALbasic die Hex-Werte für das RGB-Modell und fügt diese in den Werte-Bereich des Dialogs ein.

Ein einfacher Weg, die gewünschte Farbe zu erzeugen ist, die Farben-Palette zu öffnen und Farben über den Farb-Picker auszuwählen. Führen Sie einfach einen Doppelklick auf einen leeren Eintrag aus und wählen Sie eine Farbe mit dem Farb-Picker. Draggen Sie dann eine Farbe aus der Farben-Palette in die Code-Zeile, in der Sie die Farbe zuweisen möchten oder auf die Eigenschaft im Eigenschaften-Fenster. Ein Draggen beinhaltet beides, die (sichtbare) Farbe und ihre RGB-Werte in hexadezimal. Sie können Die Farbe auch auf den Schreibtisch oder in einen Texteditor ziehen, der Drag&Drop unterstützt.

Sie können mit der VarType-Funktion feststellen, ob eine Eigenschaft oder eine Variable ein **Color** ist. Falls es so ist, liefert VarType 16.

Beispiele

Folgendes Beispiel verwendet das RGB-Modell, um die Vordergrundfarbe eines Canvas-Steuerelements zu setzen und ein Quadrat mit der aktuellen Vordergrundfarbe zu zeichnen. Der Code wird in das Paint-Event des Canvas-Steuerelements plaziert.

Folgendes Beispiel weist eine Farbe direkt unter Verwendung des RGB-Farbmodells zu. Die RGB-Werte müssen in Hexadezimal angegeben werden.

```
Canvas1.graphics.foreColor=&cFF0000
Canvas1.graphics.drawrect 0,0,50,50
```

Folgendes Beispiel verwendet das CMY-Modell, um die Füllfarbe eines Rectangle-Steuerelements, r, zu setzen.

```
r.fillcolor=CMY (.1,.3,.5)
```

Folgendes Beispiel invertiert die Füllfarbe:

```
Dim c as Color
c=RGB(255-r.fillcolor.red,255-r.fillcolor.green, 255-r.fillcolor.blue)
r.fillColor=c
```

Siehe auch

RGB, HSV, CMY, SelectColor, DarkTingeColor, FillColor, FrameColor, HighlightColor, LightBevelColor, LightTingeColor, TextColor, Variant, VarType Funktionen; &c Operator; Dim Ausdruck.

ComboBox Steuerelement



Ein Steuerelement, das die Eingabe von Daten und auch die Auswahl von Daten aus einer Liste, ähnlich zum PopupMenu, bietet. Es verhält sich wie ein einzeliges EditField, dem eine Liste mit Einträgen angehängt ist. Die **ComboBox** kann unter Macintosh, Windows und Linux einen Fokus erhalten.

Übergeordnete Klasse

PopupMenu

Da es sich hierbei um ein RectControl handelt, beachten Sie bitte auch dessen Eigenschaften und Events.

Eigenschaften

Name	Typ	Beschreibung
<u>Text</u>	<u>String</u>	Liefert oder setzt den Text des aktuellen Eintrags

Events

Name	Beschreibung
GotFocus	Das Steuerelement hat den Fokus bekommen. Ist der Benutzer über den Tabulator auf das Element gesprungen, wird der aktuelle Eintrag hervorgehoben. Dieses Event tritt unter Macintosh, Windows und Linux auf.
LostFocus	Das Steuerelement hat den Fokus verloren. Dieses Event tritt unter Macintosh, Windows und Linux auf.
TextChanged	Der Text in der ComboBox wurde verändert.

Beispiele

Dieser Code im Open-Event-Handler füllt eine ComboBox und setzt den initialen Wert auf den aktuellen Monat:

```
Dim s as String
Dim i,last as Integer
Dim d as New Date
s="Januar,Februar,März,April,Mai,Juni,Juli,August,September,Oktober,November,Dezember"
last=CountFields(s,",")
For i=1 to last
    me.addRow NthField(s,",",i)
Next
me.ListIndex=d.Month-1
```

Die ListIndex-Eigenschaft enthält den Index des aktuell ausgewählten Eintrags, gibt aber nicht an, ob der Benutzer Text eingegeben hat. Verwenden Sie die Text-Eigenschaft, wenn Sie die aktuelle Auswahl oder den eingegebenen Text ermitteln möchten. Die folgende Zeile im TextChanged-Event zeigt den momentan ausgewählten Eintrag, beziehungsweise den eingegebenen Text an.

```
StaticText1.Text=Me.Text
```

Dieses Beispiel im Open-Event fügt der **ComboBox** einen Eintrag hinzu.

```
Me.addrow "Oktober"
```

Dieses Beispiel füllt die RowTag-Eigenschaft mit einer fortlaufenden Nummer:

```
Dim i,nItems as Integer
nItems=Me.ListCount-1
For i=0 to nItems
    Me.rowTag(i)=i
Next
```

Da RowTag ein Variant ist, müssen Sie diesen erst in ein Integer konvertieren, wenn Sie es mit einem Integer vergleichen möchten. Dies können Sie mit einer einfachen Zuweisung vornehmen:

```
Dim RecID as Integer
RecID=ComboBox1.Rowtag(1)
```

Nun vergleichen Sie RecID mit anderen Integern.

Dieses Beispiel öffnet ein neues Fenster, wenn ein Eintrag ausgewählt wurde.

```
Sub Change()
Dim w As ListEditorWindow
If ComboBox1.text="Edit List..." Then
    w=New ListEditorWindow
End If
End Sub
```

Das folgende Beispiel ändert den ausgewählten Eintrag einer **ComboBox**:

```
ComboBox1.listIndex=3
```

Dies zeigt den RowTag des ausgewählten Eintrags an:

```
EditField1.text=ComboBox1.rowtag(ComboBox1.listindex)
```

Siehe auch

[BevelButton](#), [ContextualMenu](#), [PopupMenu](#), [EditField](#) Steuerelemente; [RectControl](#) Klasse

ConsoleApplication Klasse

Wird verwendet, um Konsolen-Applikationen unter Windows, Mac OS X und Linux zu starten.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
ExecutableFile	FolderItem	Verweist auf die ausführbare Datei; Selbst wenn diese in einem Bundle liegt.

Methoden

Name	Parameter	Beschreibung
DoEvents	[milliseconds als Integer]	Gibt die Kontrolle an REALbasic zurück, damit es ausstehende Events verarbeiten kann. Der optionale Parameter gibt an, für wie lange Sie den aktuellen Thread „schlafen“ legen möchten. Wenn alle Threads „schlafen“, legt REALbasic sich selbst „schlafen“ und gibt die Kontrolle an das System. Damit können Sie Applikationen schreiben, die nicht 100% CPU-Auslastung erzeugen, wenn sie eine Schleife abarbeiten. Wenn Sie Null Millisekunden angeben, wird der nächste wartende Thread ausgeführt. Ein negativer Wert verhindert das „Schlafen“. Voreingestellt sind 10 Millisekunden.

Events

Name	Parameter	Beschreibung
Run	args als String	Liefert ein Integer . Args ist ein Array von Kommandozeilen-Parametern, die an die Applikation übergeben werden. Der erste Parameter ist immer die Applikation selbst. Der Rückgabewert wird an das System weitergegeben, als Rückgabewert der gesamten Applikation.
UnhandledException		Wird bei einer RuntimeException ausgelöst, die nicht anderweitig behandelt wurde. Dieses Event erlaubt Ihnen, letzte Aufräumarbeiten durchzuführen, allerdings läuft die Applikation nach diesem Event nicht weiter. Sie wird beendet, wenn dieser Event abgearbeitet ist.

Hinweise

Eine Konsolen-Applikation unterscheidet sich von einem normalen REALbasic-Projekt, da sie keine grafischen Elemente besitzt und ausschließlich von der Konsole aus gestartet wird. Unter Mac OS X läuft eine Konsolen-Applikation in einem Terminal, unter Windows in der Kommandozeile und unter Linux in einem Terminal-Fenster oder der Kommandozeile.

Um eine Konsolen-Applikation zu erzeugen, wählen Sie aus dem Menü den Punkt „**Ablage/Neu**“ und wählen hier „**Kommandozeilen-Applikation**“. Dies erzeugt ein Projekt, das keine Objekte für Fenster oder Menüs enthält. Es ist nur eine App-Klasse vorhanden, die von **ConsoleApplication** abgeleitet ist, anstelle von [Application](#).

Da eine Konsolen-Applikation keine Fenster oder Menüs besitzt, geschieht die Kommunikation mit dem Benutzer über die [Print](#) und [Input](#) Befehle oder über die Klassen [StandardInputStream](#) und [StandardOutputStream](#), die eine äquivalente Funktionalität bieten.

Wenn Sie eine GUI-Applikation erzeugen, beginnt ihre Ausführung mit dem Open-Event der [Application](#)-Klasse und endet mit einem Aufruf der Quit-Methode oder dem Anwählen des Menüpunkts „Beenden“. Ihre Applikation verbleibt im Speicher, bis die Quit-Methode aufgerufen wird.

Eine Konsolen-Applikation verhält sich anders. Die Ausführung beginnt im Run-Event der **ConsoleApplication**-Klasse und endet, wenn Sie den Run-Event verlassen oder die Quit-Methode aufrufen. In anderen Worten heißt das, daß die gesamte Applikation innerhalb des Run-Events läuft.

Wenn Ihre Applikation sich eher wie eine GUI-Applikation verhalten soll, die so lange läuft, bis der Benutzer mit der Applikation interagiert, so müssen Sie eine While-Schleife in den Run-Event einfügen. Damit kann sich die Applikation erst beenden, wenn der Benutzer explizit eine Aktion ausführt, die die Quit-Methode aufruft oder die While-Schleife verläßt.

Eine ServiceApplication ist ein spezieller Typ von Konsolen-Applikation, der ohne Benutzer-Interaktion arbeitet. Einige typische Beispiele sind HTTP-, FTP- oder WebDAV-Server, die auch dann laufen, wenn kein Benutzer in dem System eingeloggt ist.

Siehe auch

Input, Print, StandardInputStream, StandardOutputStream, ServiceApplication, StdErr, StdIn, StdOut, TargetHasGUI.

Const Befehl

Legt einen Wert als eine Konstante fest.

Syntax

Const Konstantenname=Wert

Hinweise

Der **Const**-Befehl kann anstelle des Dim-Befehls verwendet werden, gefolgt von einem Zuweisungsbefehl, wenn Sie sicher sind, daß der Wert der Variablen in der Methode nicht geändert werden soll. Die Verwendung des Befehl **Const** anstelle von Dim bietet einen bequemen Weg, solche Werte zu verwalten.

Ein **Const**-Befehl kann nicht innerhalb einer bedingten Struktur wie einem If-Befehl oder einem Schleifenkonstrukt platziert werden, Sie können es aber hinter Zeilen mit ausführbaren Code platzieren.

Natürlich sind Konstanten, die auf diese Art deklariert werden, lokal für die Methode. Sie können Konstanten auch in einem Fenster, einer Klasse oder einem Modul deklarieren. Dies wird im Benutzerhandbuch beschrieben. Konstanten, die zu einem Fenster oder einer Klasse gehören, können öffentlich (public), geschützt (protected) oder privat (private) deklariert werden. Module können globale Konstanten enthalten, die von überall in der Applikation erreichbar sind. Globale Konstanten können zum Lokalisieren einer Applikation verwendet werden.

Beispiele

Folgende Konstante wird für die Beschriftung eines Knopfes verwendet.

```
Const Accept="OK"
BevelButton1.caption=Accept
```

Folgendes Beispiel trägt für die Methode den Wert von „Pi“ fest.

```
Const Pi=3.141592653589793223846264338327950
```

Siehe auch

Dim Befehl.

ContextualMenu Steuerelement



ContextualMenu-Steuerelemente werden verwendet, um Kontextmenüs anzuzeigen und zu verwalten.

Übergeordnete Klasse

Control

Da dies ein Steuerelement (Control) ist, schauen Sie bei der Control-Klasse wegen der anderen Eigenschaften und Events nach, die alle Control-Objekte gemeinsam haben.

Events

Name	Parameter	Beschreibung
<i>Action</i>	Item als <u>String</u>	Dieser Eintrag des Kontextmenüs wurde ausgewählt.

Eigenschaften

Name	Typ	Beschreibung
<i>UseCCM</i>	<u>Boolean</u>	Falls <u>True</u> , wird der Hilfe-Eintrag dargestellt. Falls <u>False</u> , wird der Hilfe-Eintrag weggelassen. Dies hat unter Linux und Windows keinen Effekt.

Methoden

Name	Parameter	Beschreibung
<i>AddRow</i>	Item als <u>String</u>	Hängt einen Eintrag in eine neue Zeile am Ende der Liste an.
<i>AddSeparator</i>		Hängt eine Trennlinie an das Ende der Liste an.
<i>DeleteAllRows</i>		Löscht alle Zeilen in der Liste.
<i>Open</i>		Zeigt das Kontextmenü an.

Hinweis

Kontextmenüs werden auf die gleiche Art wie PopupMenu-Steuerelemente erzeugt. Wenn Sie ein statisches Set von Kontextmenüeinträgen haben, können Sie das Menü im Open-Event-Handler des Kontextmenüs zusammenbauen. Wenn sich die Einträge ändern, werden Sie das Kontextmenü während der Laufzeit zusammenbasteln müssen – nachdem Sie isCMMClick in einem MouseDown oder MouseEnter Event-Handler testen. Um das Kontextmenü anzuzeigen, verwenden Sie dessen Open-Methode.

Es gibt zwei Wege, Kontextmenüs zu implementieren, wenn Sie mehr als ein Steuerelement in einem Fenster haben, das Kontextmenü erfordert.

Die einfachste Methode ist, jedem Steuerelement, das ein Kontextmenü erfordert, ein ContextualMenu-Steuerelement hinzuzufügen. Bei der zweiten Methode müssen Sie eine Eigenschaft vom Typ Control oder RectControl dem Fenster hinzufügen und eine Referenz auf das Objekt, das die Anzeige eines Kontextmenüs erfordert, speichern. Dann kann ein einzelnes ContextualMenu-Steuerelement verwendet werden, um die Kontextmenüs anzuzeigen und zu steuern, da es die Eigenschaft verwenden kann, die Sie hinzugefügt haben, um zu entscheiden, welches Objekt der Anwender anklickt. Die zweite Methode ist etwas komplizierter, aber dafür effizienter, wenn Sie verschiedene Steuerelemente in einem Fenster haben, die alle das gleiche Kontextmenü benötigen.

Beispiel

Folgender Code in einem Open-Event eines ContextualMenus zeigt das Steuerelement:

```
Dim s as String
Dim i, last as Integer
s="Frühling,Sommer,Herbst,Winter"
last=CountFields(s,",")
For i=1 to last
Me.addRow NthField(s,",",i)
Next
```

Folgender Code im Mousedown-Event eines Canvas-Steuerelements zeigt ein Kontextmenü, wenn der Benutzer die ctrl- und die Maus-Taste gedrückt hält:

```
If isCMMClick then
  ContextualMenu1.open
  Return True
End if
```

Wenn der Benutzer einen Menüeintrag auswählt, wird das Action-Event des ContextualMenus ausgeführt.

Siehe auch

Control Klasse, isCMMClick Funktion.

Control Klasse

Control-Klassen enthalten NotePlayer und ContextualMenu Steuerelemente. Die RectControl-Klasse ist eine Unterklasse der Control-Klasse.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Handle</i>	<u>Integer</u>	Liefert ein Handle auf das Steuerelement. Diese Eigenschaft bietet die gleiche Funktionalität wie <code>MacControlHandle</code> , die Namensänderung zeigt jedoch die Plattformunabhängigkeit an. Jedes sichtbare Win32-Steuerelement sollte nun ein Handle besitzen.
<i>MacControlHandle</i>	<u>Integer</u>	Liefert ein Handle auf das Steuerelement. Kann von Toolbox-Programmierern verwendet werden, um die Steuerelemente direkt zu steuern. Diese Eigenschaft ist in REALbasic aus Kompatibilität zu früheren Versionen noch vorhanden. Sie wird vielleicht in künftigen Versionen nicht mehr unterstützt. Sie sollten in Ihrem Code an ihrer Stelle <i>Handle</i> verwenden.
<i>MouseX</i>	<u>Integer</u>	Die x-Koordinate der Maus in Pixeln. Gemessen von der linken oberen Ecke des Fensters.
<i>MouseY</i>	<u>Integer</u>	Die y-Koordinate der Maus in Pixeln. Gemessen von der linken oberen Ecke des Fensters.
Name	<u>String</u>	Der Name des Steuerelements.
TabPanelIndex	<u>Integer</u>	Wenn das Steuerelement auf das Panel eines <u>TabPanel</u> -Steuerelements platziert wurde, liefert es die Panel-Nummer. Das erste Panel hat die Nummer 1. Wenn das Steuerelement kein <u>TabPanel</u> ist, liefert es Null. Diese Eigenschaft ist lesbar, Sie können sie aber nicht verwenden, um ein Steuerelement in ein anderes Panel zu bewegen. Hinweis: Diese Eigenschaft wird eventuell in künftigen Versionen durch eine andere Art, wie der Ort eines Steuerelements festgelegt wird, ersetzt. Beachten Sie, daß Sie eventuell Ihren Code dann später einmal modifizieren müssen, wenn Sie <code>TabPanelIndex</code> verwenden.

Events

Name	Parameter	Beschreibung
<i>Close</i>		Das Fenster ist dabei, geschlossen zu werden.
<i>Open</i>		Das Fenster ist dabei, geöffnet zu werden.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Entfernt ein Steuerelement. Das Entfernen eines Steuerelements entfernt das Steuerelement dauerhaft aus dem Speicher und man kann nicht wieder darauf zugreifen. Sie können sowohl indizierte als auch nicht-indizierte Steuerelemente entfernen. Wenn Sie ein indiziertes Steuerelement entfernen, verschieben sich die Indizes der folgenden Steuerelemente nach unten, so daß die Indizes bei Null beginnen und durchgehend numeriert sind.
<i>Refresh</i>		Zeichnet alle Bestandteile des Steuerelements neu.
<i>RefreshRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u>	Zeichnet nur den angegebenen Bereich des Steuerelements neu.

Hinweis

Ändern des Maus-Cursors

Die `MouseCursor`-Eigenschaft bestimmt, welcher Cursor angezeigt werden soll, wenn sich die Maus über dem Steuerelement befindet, vorausgesetzt, daß die Eigenschaften von `MouseCursor` der Klassen Application und des übergeordneten Window nicht Nil sind. Sie können beispielsweise verschiedenen Steuerelementen verschiedene `MouseCursor` innerhalb eines Fensters zuweisen und die Form des Mauszeigers wird sich immer ändern, wenn sie über einem bestimmten Steuerelement ist.

Wenn Sie benutzerdefinierte Mauszeiger auf Fenster- oder Programm-Ebene benötigen, müssen Sie so vorgehen wie es im Abschnitt bei der MouseCursor-Klasse beschrieben ist.

Es gibt drei eingebaute Cursor-Formen: Armbanduhr (WatchCursor), Einfügemarke (IBeamCursor) und Pfeil (ArrowCursor). Um dem Projekt einen anderen Cursor hinzuzufügen, muß man folgendes unternehmen:

1. Starten Sie ResEdit.
2. Erzeugen Sie eine neue Resource-Datei.
3. Fügen Sie der Datei eine neue Resource vom Typ CURS hinzu.
4. Verwenden Sie den Editor von ResEdit, um das Aussehen des gewünschten Cursors zu erzeugen oder fügen Sie einen Cursor aus einer anderen CURS-Resource ein. Sie können pro CURS-Resource-ID einen Cursor erzeugen.
5. Speichern Sie die neue Resource-Datei.
6. Beenden Sie ResEdit.
7. Draggen Sie die neue Resource-Datei in Ihr Projekt-Fenster.

Sie können dann z.B. auf die CURS-Resource-ID 1027 zugreifen, indem Sie folgende Anweisung in das MouseEnter-Event des Steuerelements einfügen:

```
Me.MouseCursor=App.ResourceFork.GetCursor(1027)
```

Dies wird den Mauszeiger auf die angegebene CURS-Resource ändern, wenn der Benutzer den Mauszeiger auf das Steuerelement bewegt.

Benutzerdefinierte Maus-Cursor unter Windows

Wenn Sie benutzerdefinierte Maus-Cursor in der Windows-Version Ihres Programms benötigen, können Sie dies mit einer leicht abgewandelten Form dieser Technik erreichen. Dies ist voll kompatibel zu der Macintosh-Variante des Programms.

1. Erzeugen Sie in einem Resource-Editor eine CURS-Resource, die nur einen Cursor enthält. Geben Sie dieser Resource-Datei einen Namen, der auf den Typ des Cursors, der darin enthalten ist, hinweist.
2. Wiederholen Sie diesen Prozeß für jeden benutzerdefinierten Cursor.
3. Draggen Sie alle Resource-Dateien in das Projekt-Fenster. Diese speziellen Resource-Dateien werden im Projekt-Fenster mit einem Pfeil-Icon angezeigt.
4. Sie können dann auf jeden benutzerdefinierten Cursor über den Namen zugreifen, z.B.:

```
Me.MouseCursor=Stift
```

Dragging

Es gibt drei mögliche Dinge, die der Anwender aus einem Steuerelement draggen (ziehen) kann: Einen Text, ein Bild oder ein FolderItem. Damit der Anwender etwas draggen kann, muß das Steuerelement (im passenden Event-Handler) ein DragItem-Objekt erzeugen. Welcher der passende Event-Handler ist, hängt von der Art des Objektes ab: Für ein Canvas-Steuerelement wäre derMouseDown-Event-Handler passend, da der Anwender die Maustaste drücken muß, um etwas draggen zu können.

Das DragItem repräsentiert die Daten, die der Anwender draggt. DragItem-Objekte haben Eigenschaften, die Text-, Bild- und FolderItem-Informationen aufnehmen können. Eine oder mehrere dieser Eigenschaften müssen mit den Werten besetzt werden, die der Anwender draggen möchte. Wenn Sie ein neues DragItem mit der NewDragItem-Methode anlegen, können Sie den Verschieberahmen (drag rectangle) definieren, der während des Verschiebens erscheint.

EditFields und ListBoxes unterstützen von sich aus bereits das Dragging. Dies bedeutet, daß sie ein neues DragItem und auch den Verschieberahmen anlegen können. EditFields übernehmen automatisch die Text-Eigenschaften des DragItems, während der Anwender Text draggt. ListBoxes haben einen DragRow-Event-Handler, in dem Sie die gewünschten Daten in ihr DragItem übernehmen können.

Dropping

Bevor ein Steuerelement ein Objekt annimmt, müssen Sie die Methoden AcceptFileDrop, AcceptPictureDrop und/oder AcceptTextDrop verwenden, um die Arten der Objekte anzugeben, die auf das Objekt gedroppt werden dürfen. Dies wird üblicherweise im Open-Event-Handler erledigt. Wenn ein annehmbares Objekt auf das Steuerelement gezogen wird, wird der DropObject-Event-Handler gestartet und ihm das DragItem übergeben, das auf das Steuerelement gezogen wurde. Zu diesem Zeitpunkt können Sie die DragItem-Methode verwenden, um festzustellen, was für Daten verfügbar sind und auf diese zugreifen. Wenn Sie nur eine bestimmte Art

von Daten (wie z.B. Text) zum Draggen zulassen, müssen Sie die Art der Daten auch nicht überprüfen. Sie können den Text einfach aus der Text-Eigenschaft des DragItems holen. Siehe „DragItem Klasse“ für weitere Informationen und Beispiele.

Beispiele

Dieses Beispiel zeigt, wie der DropObject-Event-Handler eines [EditFields](#) programmiert sein kann, um eines oder mehrere Simple-Text-Dokumente oder anderen Text akzeptieren und verarbeiten zu können:

```
Sub DropObject(Obj as DragItem)
  If Obj.TextAvailable then
    EditField1.Text=Obj.Text
  ElseIf Obj.FolderItemAvailable Then
    Do
      //Lade den Styled Text aus dem
      //Dokument in das EditField1
      Obj.FolderItem.OpenStyledEditField EditField1
    Loop Until Not Obj.NextItem
  End if
End Sub
```

Siehe auch

[NotePlayer](#) Steuerelement, [RectControl](#), [Object](#) Klassen.

ControlPanelsFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Ordner „Kontrollfelder“ des Systemordners zu bekommen. Unter Windows gibt der Aufruf von ControlPanelsFolder Nil zurück.

Syntax

Ergebnis = **ControlPanelsFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Ein FolderItem das den Kontrollfelder-Ordner repräsentiert.

Hinweis

Verwenden Sie die **ControlPanelsFolder**-Funktion, wenn Sie auf Objekte aus dem Kontrollfelder-Ordner zugreifen müssen.

Die Funktion **ControlPanelsFolder** bietet eine Möglichkeit, auf die Einträge Ordner „Kontrollfelder“ zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird. Wenn Ihr Programm unter Windows verwendet wird, können Sie den Wert der globalen Konstante [TargetWin32](#) überprüfen, bevor Sie **ControlPanelsFolder** aufrufen. Da **ControlPanelsFolder** unter Windows Nil liefert, sollte man auf einen Nil-Wert testen, bevor man versucht, auf das FolderItem zuzugreifen, das von **ControlPanelsFolder** zurückgeliefert wird.

ControlPanelsFolder liefert unter Linux [Nil](#) zurück.

Beispiele

Bei der Beschreibung der Funktion [AppleMenuFolder](#) finden Sie ein passendes Beispiel.

Siehe auch

[DesktopFolder](#), [ExtensionsFolder](#), [FontsFolder](#), [ApplicationSupportFolder](#), [PreferencesFolder](#), [ShutDownItemsFolder](#), [StartupItemsFolder](#), [SystemFolder](#), [TemporaryFolder](#), [TrashFolder](#) Funktionen.

ConvertEncoding Funktion

Bietet eine schnelle Lösung, um einen String bekannter Kodierung in eine andere zu konvertieren, ohne daß man zuvor ein [TextConverter](#)-Objekt erzeugen muß.

Syntax

Ergebnis = **ConvertEncoding**(Str, NewEncoding)

oder

Ergebnis=Str.ConvertEncoding(NewEncoding)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der konvertierte String, der die <i>NewEncoding</i> -Textkodierung verwendet.
str	<u>String</u>	Der zu konvertierende String.
newEncoding	TextEncoding	Die für die Konvertierung zu verwendende Kodierung.

Wenn Sie Text in eine Datei schreiben, der von einer anderen Applikation gelesen wird, die eine bestimmte Kodierung erwartet, können Sie die Methode **ConvertEncoding** verwenden, bevor Sie mit der Write-Methode den Text in die Datei schreiben. Hier ist ein Beispiel, das den Text eines EditFields nach MacRoman konvertiert.

```
Dim file As FolderItem
Dim fileStream As TextOutputStream
file=GetSaveFolderItem("plain/text","My Info")
fileStream=file.CreateTextFile
fileStream.Write ConvertEncoding(namefield.Text,Encodings.MacRoman)
fileStream.Close
```

Beispiel

Das folgende Beispiel verwendet das Encodings-Objekt, um den Text in einem EditField nach UTF8 zu konvertieren.

```
Dim s as String
s=ConvertEncoding(Editfield1.text,Encodings.UTF8)
s=EditField1.Text.ConvertEncoding(Encodings.UTF8)
```

Siehe auch

TextEncoding, TextConverter, TextOutputStream Klassen; DefineEncoding, Encoding Funktionen; Encodings Objekt.

Cos Funktion

Gibt den Cosinus des angegebenen Winkels zurück.

Syntax

Ergebnis=Cos(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der Cosinus von <i>Wert</i> .
<i>Wert</i>	<u>Double</u>	Der Wert (im Bogenmaß), dessen Cosinus Sie ermitteln wollen.

Hinweis

Die **Cos**-Funktion gibt den Cosinus des Winkels (im Bogenmaß) der übergebenen Zahl zurück. Wenn der Winkel im Gradmaß vorliegt, multiplizieren Sie ihn mit Pi/180, um ihn ins Bogenmaß zu konvertieren.

Beispiele

Dieses Beispiel verwendet die Funktion **Cos**, um den Cosinus eines Winkels zu ermitteln.

```
Dim d as Double
Const PI=3.14159265358979323846264338327950
d=Cos(45*Pi/180) //ergibt 0.707
```

Siehe auch

Acos Funktion.

CountFields Funktion

Liefert die Anzahl an Feldern (Werten) des übergebenen Strings zurück, die durch ein angegebenes Trennzeichen getrennt sind.

Syntax

Ergebnis=CountFields(source, separator)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Anzahl der Felder in <i>source</i> , die durch <i>separator</i> getrennt sind

Teil	Art	Beschreibung
<i>source</i>	<u>String</u>	Der zu prüfende String.
<i>separator</i>	<u>String</u>	Das Zeichen, das die Felder in <i>source</i> voneinander trennt.

Hinweise

Die Funktion **CountFields** wird verwendet, um spaltenweise Daten aus einer Textdatei einzulesen, die durch ein bestimmtes Zeichen voneinander getrennt werden.

Wenn der Separator in *Source* nicht gefunden wird, liefert **CountFields** 1. Wenn *Source* leer ist, liefert **CountFields** Null.

Beispiele

Dieses Beispiel liefert 5 zurück.

```
Dim count as integer
Dim s as String
s="Peter*Pan*11.03.69*5125554323*Männlich"
count=CountFields(s, "*")
```

Folgendes Beispiel liefert drei, da es das leere „Feld“ nach dem (unnötigen) letzten Feldbegrenzer mitzählt.

```
Dim count as Integer
Dim s as String
s="Franz*Schmidt*"
count=CountFields(s, "**")
```

Sehen Sie auch beim Beispiel nach, das zeigt, wie man das PopupMenu Steuerelement aufruft.

Siehe auch

TextInputStream Object Beispiel, nthField Funktion.

CriticalSection Klasse

Wird verwendet, um einen kritischen Code-Bereich in einer Multithreading-Umgebung zu schützen.

Übergeordnete Klasse

Object

Methoden

Name	Parameter	Beschreibung
<i>Enter</i>		Beginn des kritischen Code-Bereichs.
<i>Leave</i>		Ende des kritischen Code-Bereichs.

Hinweis

CriticalSection ähnelt einer Semaphore, mit dem Unterschied, daß CriticalSection normalerweise verwendet wird, um einen kritischen Bereich des Programmcodes zu schützen. Die Methoden *Enter* und *Leave* für den kritischen Code-Bereich arbeiten ähnlich wie Signal/Release. Der Hauptunterschied ist, daß CriticalSection.Enter mehrere Male vom gleichen Thread aufgerufen werden kann (mit entsprechend vielen dazugehörigen Leave-Aufrufen), während Semaphore.Signal nur einmal aufgerufen werden kann.

Siehe auch

Semaphore Klasse.

CStr Funktion

Wird benutzt, um das angegebene Argument in einen String zu konvertieren. Sie können hier auch Klassen als Argument angeben, sofern diese den Convert Operator implementiert haben, der einen String zurückliefert.

Beachten Sie auch die Str Funktion für weitere Informationen.

Syntax

Ergebnis=**CStr**(Daten)

Teil	Typ	Beschreibung
Ergebnis	<u>String</u>	Die String-Repräsentation des Daten-Arguments.
Daten	Variable oder eine Klasse mit einer String-Repräsentation	Die Variable oder Klasse deren String-Repräsentation Sie benötigen.

Beispiel

Dieses Beispiel liefert die String-Repräsentation von zwei Zahlen und einem Boolean.

```
Dim s as String
Dim d as New Date
s=CStr(1) //liefert "1" als einen String
s=CStr(d.Day) //liefert die Tag-Zahl als String
s=CStr(True) //liefert "True"
```

Siehe auch

CDBl, Str, Val Funktionen.

CurveShape Klasse

Wird zum Zeichnen von Linien und Kurven in einer Vektorgrafik-Umgebung verwendet.

Übergeordnete Klasse

Object2D

Eigenschaften

Name	Typ	Beschreibung
<i>ControlX</i>	<u>Double</u>	Parameter ist Index as Integer. Ein Null-basiertes Array von Kontrollpunkten (horizontale Position).
<i>ControlY</i>	<u>Double</u>	Parameter ist Index as Integer. Ein Null-basiertes Array von Kontrollpunkten (vertikale Position).
<i>Order</i>	<u>Integer</u>	Die Anzahl von Kontrollpunkten außerhalb der Kurve, die verwendet werden. Es ist einer der folgenden Werte: 0—Eine gerade Linie von x,y nach x2,y2 wird verwendet. 1—Eine quadratische Bezier-Kurve wird unter Verwendung eines Kontrollpunktes gezeichnet. 2—Eine kubische Bezier-Kurve wird unter Verwendung zweier Kontrollpunkte gezeichnet.
<i>Segments</i>	<u>Integer</u>	Anzahl von geraden Linien-Segmenten, die für die Annäherung einer Kurve verwendet werden. Der Defaultwert von Null weist REALbasic an, eine 'sensible' Anzahl von Segmenten zu verwenden.
<i>X2</i>	<u>Double</u>	Die horizontale Position des Endes der Linie oder Kurve.
<i>Y2</i>	<u>Double</u>	Die vertikale Position des Endes der Linie oder Kurve.

Hinweise

Bei CurveShapes ist der Defaultwert für die Border-Eigenschaft 100 und der Defaultwert für Fill ist 0.

Wenn Sie Order auf Null setzen, erhalten Sie eine gerade Linie von x,y nach x2, y2. Sie verwenden Kontrollpunkte, wenn Sie die Linie auf eine bestimmte Weise „verbiegen“ möchten. Wenn Sie **Order** auf 1 setzen, krümmt sich die Linie auf dem Weg von X,Y nach X2,Y2 zu ControlX(0),ControlY(0) hin. Das ist genau das, was in dem Beispiel passiert. Je weiter der Kontrollpunkt entfernt ist, desto mehr wird die Linie in diese Richtung abgelenkt.

Wennn Order auf 2 gesetzt wird, gibt es zwei Kontrollpunkte. Die Linie verbiegt sich zuerst Richtung ControlX(0),ControlY(0) und dann Richtung ControlX(1),ControlY(1). An den Endpunkten zeigt die Linie direkt auf den zugehörigen Kontrollpunkt. Die Kurve befindet sich immer innerhalb des Polygons, das die Endpunkte und die Kontrollpunkte (wenn es welche gibt) bildet.

Beispiel

Folgende Methode zeichnet, wenn sie sich im MouseDown-Event eines Fensters befindet, eine einfache kleine Kurve, wenn der Anwender die Maustaste drückt. Der negative Wert von ControlY(0) plaziert den Kontrollpunkt über die imaginäre gerade Linie von x,y nach x2,y2.

```
Dim c As CurveShape
c=New CurveShape
c.controlx(0)=120
c.controly(0)=-40
c.order=1
c.x=10
c.y=10
c.x2=250
c.y2=10
Graphics.DrawObject c,x,y
```



Siehe auch

[FigureShape](#), [FolderItem](#), [Group2D](#), [Graphics](#), [Object2D](#), [OvalShape](#), [Picture](#), [PixmapShape](#), [RectShape](#), [RoundRectShape](#), [StringShape](#) Klassen.

DarkBevelColor Funktion

Liefert die momentan gewählte Farbe des Erscheinungsbild-Managers zurück, die zum Zeichnen dunkler Bereiche bei Unterteilungslinien und GroupBoxen verwendet wird.

Syntax

Ergebnis=DarkBevelColor

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Die zum Zeichnen verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit [Canvas](#) Steuerelementen wichtig, um deren Aussehen mit denen des Erscheinungsbild-Managers gleich zu halten. Wenn Sie Steuerelemente wie [Unterteilungslinien](#) oder [GroupBox](#)-Objekte zeichnen, sollten Sie diese Farbe zum Zeichnen dunkler Bereiche (üblicherweise die rechte und die untere Seite des Objekts) verwenden.

Dieser Wert kann vom Anwender geändert werden, weshalb Sie diesen Wert im Paint-Event-Handler abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

[DarkTingeColor](#), [FillColor](#), [FrameColor](#), [HighlightColor](#), [LightBevelColor](#), [LightTingeColor](#), [TextColor](#), [RGB](#), [CMY](#), [HSV](#) Funktionen; [Color](#) Datentyp.

DarkTingeColor Funktion

Liefert die momentan gewählte Farbe des Erscheinungsbild-Managers zurück, die zum Zeichnen von dunklen Linien an rechter und unterer Seite in Rahmen verwendet wird.

Syntax

Ergebnis=DarkTingeColor

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Die zum Zeichnen verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit Canvas Steuerelementen wichtig, um deren Aussehen mit denen des Erscheinungsbild-Managers gleich zu halten. Wenn Sie Steuerelemente wie Unterteilungslinien oder GroupBox-Objekte zeichnen, sollten Sie diese Farbe zum Zeichnen heller Bereiche (üblicherweise die rechte und die untere Seite des Objekts) verwenden. In einem Checkbox Steuerelement wird diese Farbe verwendet, um den hervorgehobenen Eindruck hervorzurufen.

Dieser Wert kann vom Anwender geändert werden, weshalb Sie diesen Wert im Paint-Event-Handler abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

DarkBevelColor, FillColor, FrameColor, HighlightColor, LightBevelColor, LightTingeColor, TextColor Funktionen.

DataAvailableProvider Interface Klasse

Wird verwendet, um benutzerdefinierte Objektverknüpfungen zu programmieren.

Eigenschaften

Name	Typ	Beschreibung
<i>dataAvailable</i>	<u>Boolean</u>	<u>True</u> , falls Daten verfügbar sind.

Siehe auch

DataNotifier Interface Klasse.

Database Klasse

Ein Database-Objekt repräsentiert eine offene Datenbank, auf die mit den Befehlen des in REALbasic integrierten Datenbank-Front-Ends zugegriffen werden kann. Sie können eine REAL-Datenbank mittels der REALdatabase-Klasse öffnen oder erzeugen. Verwenden Sie die ADSP4DServer, TCP4DServer, ODBCDatabase, OpenBaseDatabase, MySQLDatabase oder PostgreSQLDatabase Klassen, um Database-Objekte zurückzuliefern.

Für jede Database außer der REAL-Datenbank müssen Sie das passende Datenbank-Plugin im REALbasic-Plugins-Ordner installieren, um andere Datenquellen zu verwenden. Database-Plugins können unter Umständen häufiger aktualisiert worden sein als REALbasic selbst. Sie können die aktuellen Versionen aller Datenbank-Plugins über www.realsoftware.com erhalten.

FrontBase kann ebenso über ein Plugin eines Fremdanbieters angesteuert werden. Das FrontBase-Plugin ist kostenlos und befindet sich auf der REALbasic-CD.

Die Standard-Version von REALbasic unterstützt die gesamte Datenbank-Funktionalität, beschränkt Sie aber auf Datenbanken mit höchstens 50 Datensätzen und beinhaltet nicht die Möglichkeit, eigenständige Datenbank-Applikationen zu erzeugen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>DatabaseName</i>	<u>String</u>	Der Name der zu öffnenden Datenbank.
<i>Error</i>	<u>Boolean</u>	True, falls die Datenbank-Engine einen Fehler gemeldet hat.
<i>ErrorCode</i>	<u>Integer</u>	Der Fehlercode, der von der Datenbank-Engine gemeldet wurde. Fehlercodes und Fehlermeldungen sind je nach Engine unterschiedlich.
<i>ErrorMessage</i>	<u>String</u>	Der Text der Fehlermeldung, den die Datenbank-Engine lieferte. Fehlercodes und Fehlermeldungen sind je nach Engine unterschiedlich.
<i>Host</i>	<u>String</u>	Der Datenbank-Host-Name oder IP-Adresse des Servers, mit dem verbunden werden soll.
<i>Password</i>	<u>String</u>	Das Paßwort, das für den Zugriff auf die Datenbank benötigt wird.
<i>UserName</i>	<u>String</u>	Der Anwender-Name, der für den Zugriff auf die Datenbank benötigt wird.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Schließt die Datenbank.
<i>Commit</i>		Speichert Änderungen an Datensätzen in der Datenbank. Wenn Sie die Anwendung beenden, nachdem Datensätze geändert wurden, führt REALbasic ein implizites Commit durch. Verwenden Sie Commit und Rollback, um Transaktionen zu verarbeiten.
<i>Connect</i>		Verbindet mit dem Datenbank-Server und öffnet die Datenbank für den Zugriff. Liefert ein <u>Boolean</u> . Bevor Sie mit Datenbankoperationen fortfahren, sollten Sie testen, ob Connect True liefert.
<i>FieldSchema</i>	TableName als <u>String</u>	Liefert einen <u>DatabaseCursor</u> mit Informationen über alle Felder der Tabelle. Näheres siehe Hinweise.
<i>InsertRecord</i>	TableName als <u>String</u> Data als <u>DatabaseRecord</u>	Fügt Data als letzten Datensatz in die Tabelle <i>TableName</i> ein.
Property	Name als String	Ermittelt die durch Name spezifizierte Datenbank-Eigenschaft der Datenquelle. Derzeit nur für 4D-Server unterstützt und zum Ermitteln des ConnectionStrings. Siehe <u>Beispiel</u> .
<i>Rollback</i>		Widerruft eine Reihe von Änderungen, die an Datensätzen vorgenommen wurden. Benutzen Sie diese oder die Commit-Methode, um eine Transaktion abzuschließen oder zu verwerfen.
<i>SQLExecute</i>	SQL-Anweisung als <u>String</u>	Die SQL-Anweisung ist eine gültige SQL-Anweisung(select ausgenommen), die von der Datenquelle unterstützt wird. Benutzen Sie die SQL-Select-Methode, wenn Sie Abfragen durchführen wollen.
<i>SQLSelect</i>	SQLAbfrage als <u>String</u>	Eine SQL-Select-Abfrage. SQLAbfrage ist eine gültige Abfrage, die ein <u>RecordSet</u> zurückliefert. In den REALbasic-Versionen 4.0 und kleiner wird ein <u>DatabaseCursor</u> zurückgeliefert.
<i>TableSchema</i>		Liefert einen <u>DatabaseCursor</u> mit einer Liste aller Tabellen der Datenbank. Siehe Hinweise.

Hinweise

SQL für eine REALdatenbank-Datenquelle

Die REALdatenbank Daten-Quelle unterstützt eine Untermenge von SQL/92 und SQL/99, inklusive Abfragen, die ein Self-Join oder Aggregat-Funktionen enthalten. Für mehr Informationen beachten Sie bitte den Anhang A in der gedruckten Sprachreferenz oder der PDF-Variante.

Mit einer Datenquelle eine Verbindung aufnehmen

Folgende neue Unterklassen der Database-Klasse wurden in Version 4.5 von REALbasic hinzugefügt. Die entsprechenden Funktionen der früheren REALbasic-Versionen sind veraltet:

Klasse	Veraltete Funktionen
<i>ADSP4DServer</i>	Wird anstelle von <u>Open4DDatabasebyADSP</u> verwendet.
<u>MySQLDatabase</u>	Bietet Unterstützung für MySQL-Datenbanken; dies war bisher nur mit Plugins von Fremdanbietern möglich.
<i>ODBCDatabase</i>	Wird anstelle von <u>OpenODBCDatabase</u> verwendet.
<i>OpenBaseDatabase</i>	Wird anstelle von <u>OpenOpenBaseDatabase</u> verwendet.
<i>PostgreSQLDatabase</i>	Wird anstelle von <u>OpenPostgreSQLDatabase</u> verwendet.
<i>REALdatabase</i>	Wird anstelle von <u>OpenREALdatabase</u> und <u>NewREALdatabase</u> verwendet.
<i>TCPIP4DServer</i>	Wird anstelle von <u>Open4DDatabasebyTCPIP</u> verwendet.

Die oben aufgelisteten Funktionen sind aus Kompatibilitätsgründen in der aktuellen REALbasic-Version noch vorhanden, wir empfehlen jedoch, die neuen Unterklassen der Database-Klasse zu verwenden. Sie besitzen zwei große Vorteile:

- Wenn Sie dieses Schema verwenden, können wir Funktionen anbieten, die bei bestimmten Datenbanken einzigartig sind; einige Datenbank-Klassen unterstützen unterschiedliche Dinge.
- Es ist nun möglich, Unterklassen der Database-Klasse zu erzeugen und damit etwas Nützliches anzustellen. Das war bisher nicht möglich, da alle Funktionen, die zum Öffnen einer Datenquelle verwendet wurden, ein Database-Objekt lieferten.

Wenn Sie eine Verbindung zu einer beliebigen Datenquelle herstellen, sollten Sie zuvor einen Test wie folgenden durchführen, bevor Sie fortfahren:

```
Dim d as Database
.
If db.Connect() then
//Verbindung ist erfolgreich
else
//Verbindung schlug fehl
end if
```

Normalerweise werden Sie die Eigenschaften Username, Host und Password der Database-Klasse verwenden, um eine Verbindung herzustellen. Bestimmte Anforderungen unterscheiden sich von Datenquelle zu Datenquelle.

FieldSchema und TableSchema

FieldSchema liefert ein RecordSet mit fünf Feldern: ColumnName (String), FieldType (Integer), IsPrimary (Boolean), und NotNull (Boolean) und Length in Bytes (Integer) für Textfelder. FieldType (Datentyp) verwendet folgende Werte. Nicht alle Datenquellen unterstützen alle diese Feldtypen. Außerdem haben einige Feldtypen implementationsspezifische Features, wie z.B. numerische Genauigkeit und Maximallänge. Die REALdatenbank unterstützt eine Untermenge der Datentypen: Integer, VarChar, Double, Date, Time, Timestamp, Boolean and Binary. Lesen Sie hierzu den Anhang A in der gedruckten Sprachreferenz oder der PDF-Variante. Verwenden Sie eine andere Datenquelle, beachten Sie bitte deren Dokumentation.

Von FieldSchema gelieferte Feldtypen

FieldType	Wert	Beschreibung
Null	0	Zeigt das Fehlen eines Wertes an, d.h., es fehlt ein Wert.
Byte	1	Speichert die Byte-Repräsentation eines Character-Strings.
SmallInt	2	Ein numerischer Datentyp ohne Nachkommateil. Die maximale Anzahl von Stellen ist implementationsabhängig, ist jedoch meist kleiner oder gleich INTEGER. Die REAL-Database unterstützt 2-Byte-SmallInts, die es erlauben, Werte im Bereich von +/- 32767 zu speichern. Wenn Sie eine andere Datenquelle verwenden, schauen Sie in deren Dokumentation nach.
Integer	3	Ein numerischer Datentyp ohne Nachkommateil. Die maximale Anzahl von Stellen ist implementationsabhängig. Die REAL-Database unterstützt 4-Byte-Integer, die es erlauben, Werte im Bereich von +/- 2.000.000.000 zu speichern. Wenn Sie eine andere Datenquelle verwenden, schauen Sie in deren Dokumentation nach
Char	4	Speichert alphabetische Daten, für die Sie die maximale Anzahl von Zeichen für das Feld festlegen, also CHAR(20) für ein Feld mit 20 Zeichen. Wenn ein Datensatz weniger als die Maximalzahl von Zeichen für das Feld enthält, werden die verbleibenden Zeichen mit Leerzeichen gefüllt.
Text oder VarChar	5	Speichert alphabetische Daten, bei denen die Anzahl der Zeichen von Datensatz zu Datensatz variieren, bei denen Sie aber nicht wollen, daß unbenutzte Zeichen mit Leerzeichen gefüllt werden. VARCHAR(20) legt zum Beispiel ein VARCHAR-Feld mit der Maximallänge von 20 Zeichen fest. Die REAL-Database unterstützt den VarChar-Feldtyp, nicht den Text-Typ. Die REAL-Database verwendet als Maximallänge für ein VARCHAR-Feld 8174 Bytes; wenn Sie in einem CREATE TABLE oder ALTER TABLE Befehl eine Maximallänge festlegen, wird es ignoriert.
Float	6	Speichert Fließkommazahlen mit einer Genauigkeit, die Sie festlegen können, z.B. FLOAT (5). Die REAL-Database verwendet 4-Byte-Fließkommazahlen.

FieldType	Wert	Beschreibung
<i>Double</i>	7	Speichert Fließkommazahlen mit doppelter Genauigkeit. Die REAL-Datenbank verwendet 8-Byte Doubles.
<i>Date</i>	8	Speichert die Werte Jahr, Monat und Tag eines Datums. Der Jahr-Wert ist vier Stellen lang, die Werte für Monat und Tag zwei Stellen.
<i>Time</i>	9	Speichert die Werte Stunden, Minuten und Sekunden einer Uhrzeit. Die Stunden und Minuten sind zwei Stellen lang. Die Sekunden bestehen auch aus zwei Stellen, können aber auch einen optionalen Nachkommateil haben, z.B. 09:55:25.248. Die Standardlänge des Nachkommateils ist null.
<i>TimeStamp</i>	10	Speichert Tag- und Uhrzeit-Information. Die Länge der Komponenten eines TimeStamp sind die gleichen wie für Time und Date, außer daß die Standardlänge für den Nachkommateil der Zeit sechs und nicht null Stellen ist. Wenn ein TimeStamp-Wert keinen Nachkommateil besitzt, ist seine Länge 19 Stellen. Wenn es einen Nachkommateil besitzt, hat es die Länge 20 plus der Länge des Nachkommateils.
<i>Currency</i>	11	Speichert einen Dezimalwert mit einem Dollarzeichen.
<i>Boolean</i>	12	Speichert den Wert TRUE oder FALSE.
<i>Decimal</i>	13	Speichert einen numerischen Wert, der einen Vor- und einen Nachkommateil haben kann. Sie legen die Gesamtzahl der Stellen und die Anzahl der Nachkommastellen fest. Z.B. legt DECIMAL (5,2) ein Dezimalfeld fest, das Werte bis 999,99 enthalten kann. DECIMAL (5) legt ein Feld fest, das Werte bis 99.999 enthalten kann.
<i>Binary</i>	14	Speichert Code, Bilder und hexadezimale Daten. Schlagen Sie in der Dokumentation Ihrer Datenquelle wegen der Maximalgröße des Binary-Feldes nach.
<i>Long Text (BLOB)</i>	15	Speichert ein Text-Objekt. Schlagen Sie in der Dokumentation Ihrer Datenquelle wegen der Maximalgröße des BLOB-Feldes nach.
<i>Long Var-Binary (BLOB)</i>	16	Speichert ein Binär-Objekt. Die REAL-Datenbank unterstützt BLOBs bis 2 GByte. Wenn Sie eine andere Datenquelle verwenden, schlagen Sie in deren Dokumentation wegen der Maximalgröße nach.
<i>MacPICT</i>	17	Speichert ein Macintosh-PICT-Bild.
<i>Unknown</i>	18	Nicht erkannter Datentyp.

Die Werte von IsPrimary (Primärschlüssel) und AllowNull (leeren Feldeintrag erlauben) werden durch den SQLSelect-Befehl gesetzt, der die Tabelle erzeugt.

TableSchema liefert ein RecordSet mit dem Feld TableName (String).

Die Property-Methode

Der Zweck der Property-Methode besteht darin, verschiedene Informationen über die Datenquelle zu ermitteln. Wenn Sie z.B. 4th Dimension als Datenquelle verwenden, können Sie der Property-Methode den String „ConnectionString“ übergeben und sie wird den Connection-String für die 4D-Server-Datenbank zurückliefern. Dieser enthält den Namen der Datenbank. ConnectionString wird von 4th Dimension unterstützt, jedoch nicht von anderen Datenquellen.

SQLSelect und SQLExecute Methoden

Sie können die Methoden SQLSelect und SQLExecute verwenden, um mit Ihrer Datenquelle mit SQL-Befehlen zu kommunizieren. Verwenden Sie die SQLSelect-Methode für Select-Abfragen und die SQLExecute-Methode für alles andere. REALbasic unterstützt derzeit eine Untermenge von SQL. Folgende Tabelle listet die unterstützten SQL-Befehle auf:

Befehl	Beispiel	Beschreibung
<i>SELECT</i> <i>WHERE</i> <i>ORDER BY</i> <i>GROUP BY</i>	SELECT * from Kunden	Liefert eine Gruppe von Datensätzen, bekannt als ein Cursor in der SQL-Sprache. REALbasic verwendet den Ausdruck RecordSet. Sie können die Spalten (Felder) festlegen, die Tabelle(n), Suchbedingungen, Gruppierung und Spaltensortierung.

Befehl	Beispiel	Beschreibung
<i>CREATE TABLE</i>	CREATE TABLE Rechnungen (id integer not null, KuNr integer not null, Betrag double, Datum date, primary key (id))	Erzeugt eine Tabelle und legt die Felder und deren Attribute fest.
<i>ALTER TABLE</i>	ALTER TABLE <Tabellenname> ADD <Feldname> <field type> [NOT NULL]	Fügt einer vorhandenen Tabelle ein Feld hinzu und legt optional als Eigenschaft Not Null fest.
<i>UPDATE</i>	UPDATE Kunden SET Stadt='Heidelberg', Telefon='06221-300002' WHERE KuNr='0121'	Aktualisiert vorhandenen Datensatz.
<i>INSERT</i>	INSERT INTO pubs (pub_id, pub_name) VALUES (481, "REALbasic erforschen")	Fügt einen Datensatz in eine festgelegte Tabelle ein. Alle Werte für alle Spalten müssen festgelegt werden Sie können nicht einen Datensatz einfügen, jedoch nur einige der Spalten festlegen.
<i>DELETE</i>	DELETE FROM Autoren WHERE au_name ="Marx"	Löscht den Datensatz, der auf die WHERE-Bedingung paßt.
<i>COMMIT</i>	COMMIT	Speichert Änderungen (Einfügen, Löschen, Aktualisierung) in der Datenbank.
<i>ROLLBACK</i>	ROLLBACK	Verwirft Änderungen (Einfügen, Löschen, Aktualisierung) in der Datenbank.
<i>Mengenfunktionen: Min, Max, Avg, Count, Sum</i>	Select Count (*) from Kunden Select Sum(Gesamtsumme) from Rechnung Select Name from Rechnung where Gesamtsumme = (select max(Gesamtsumme) from Rechnung)	Mengenfunktionen fassen eine Eigenschaft einer Menge von Datensätzen in einem Wert zusammen. Mengenfunktionen werden zusammen mit dem SELECT-Befehl und einer optionalen WHERE-Klausel verwendet.

SQL Aggregat-Funktionen

Die folgenden Aggregat-Funktionen werden unterstützt. Jede Funktion berechnet einen Wert aus einer Reihe von Zeilen, die über den WHERE-Ausdruck in der SELECT-Abfrage ausgewählt werden, in der sich auch die Aggregat-Funktion befindet:

Funktion	Definition
AVG	Liefert den Mittelwert eines Ausdrucks.
COUNT	Liefert die Anzahl der Zeilen, die von dem Ausdruck berührt wurden
COUNT (*)	Liefert die Anzahl der Zeilen, die von dem Ausdruck berührt wurden. Es wird keine Auflistung der Felder angegeben und Null-Werte und Duplikate sind in dem Ergebnis nicht berücksichtigt.
MAX	Liefert den maximalen Wert der angegebenen Felder.
MIN	Liefert den minimalen Wert der angegebenen Felder.
SUM	Liefert die Summe der angegebenen Felder.

Datum/Zeit und andere Funktionen

Die folgenden Datums- und Zeit-Funktionen werden von der REALdatenbank-Datenquelle unterstützt.

Funktion	Definition
CURRENT_DATE	Liefert das aktuelle Datum im SQL-Datumformat (YYYY-MM-DD).
CURRENT_TIME	Liefert die aktuelle Zeit im SQL-Zeitformat (HH:MM:SS).
CURRENT_TIMESTAMP	Liefert den aktuellen Zeitpunkt als SQL-TimeStamp (YYYY-MM-DD HH:MM:SS).
LAST_ROWID	Liefert den Wert der letzten _rowID für die angegebene Tabelle. Dieser Wert wird für einige Arten von Relationalen-Joins benötigt.

Beispiele

Argument	Beschreibung														
<i>Spaltenliste</i>	Liste von Feldern durch Kommas getrennt. Die Verwendung des Asterisk (*) anstelle der Feldnamen ruft alle Felder der Tabelle ab. Wenn ein Spaltenname Leerzeichen enthält, muß er in eckigen Klammern eingeschlossen werden, z.B. muß [PLZ Ort] für ein Feld namens „PLZ Ort“ verwendet werden. Ebenso müssen Sie die Klammern in dieser Form verwenden, wenn Ihr Spaltenname Zeichen mit ASCII-Code größer 127 verwendet. Wenn <i>Tabellenliste</i> auf mehrere Tabellen verweist, verwenden Sie die Syntax <i>Tabellenname.Feldname</i> , um auf dieses Feld zu verweisen.														
<i>Tabellenliste</i>	Liste der Tabellen durch Komma getrennt. Wenn ein Tabellenname Leerzeichen enthält oder Zeichen mit ASCII-Code größer 127 verwendet, muß er in eckige Klammern eingeschlossen werden, z.B. [Produkte Gruppen]														
<i>Suchbedingung</i>	Ausdruck, der eine Untermenge von Datensätzen in einer Tabelle oder mehreren Tabellen festlegt. Muß durch ein Boolean auf den Wert True für die gewünschten Datensätze getestet werden. Wenn <i>Tabellenliste</i> auf mehrere Tabellen verweist, müssen Sie die Syntax <i>Tabellenname.Feldname</i> bei der <i>Suchbedingung</i> verwenden, z.B. Kunden.KuNr=Rechnungen.KuNr. <i>Suchbedingung</i> kann folgende Vergleichssymbole verwenden: <table> <tr> <th>Vergleich</th><th>Symbol</th></tr> <tr> <td>Gleichheit</td><td>=</td></tr> <tr> <td>Nicht gleich</td><td>< ></td></tr> <tr> <td>Kleiner als</td><td><</td></tr> <tr> <td>Kleiner oder gleich</td><td>< =</td></tr> <tr> <td>Größer als</td><td>></td></tr> <tr> <td>Größer oder gleich</td><td>> =</td></tr> </table> Sie können durch OR, AND und NOT kombinierte Suchbedingungen verwenden, z.B. "Kunden.KuNr=56 AND Rechnungen.KuNr=56". Um eine SQL-Wildcard zu verwenden, verwenden Sie den LIKE-Operator, z.B. SELECT * FROM Kunden WHERE Name LIKE "Schmid%"	Vergleich	Symbol	Gleichheit	=	Nicht gleich	< >	Kleiner als	<	Kleiner oder gleich	< =	Größer als	>	Größer oder gleich	> =
Vergleich	Symbol														
Gleichheit	=														
Nicht gleich	< >														
Kleiner als	<														
Kleiner oder gleich	< =														
Größer als	>														
Größer oder gleich	> =														
<i>Gruppierungsbedingung</i>	Feld oder Felder, in denen Sie die Datensätze im Cursor gruppieren wollen. Wenn eine GROUP BY Bedingung verwendet wird, werden die Datensätze in Gruppen durch die Felder in dieser Bedingung organisiert. Die Gruppen erscheinen in alphabetischer Reihenfolge.														
<i>Sortierbedingung</i>	Feld oder Felder in denen Sie die individuellen Datensätze im <u>RecordSet</u> sortieren möchten. Verwenden Sie ORDER BY anstelle von GROUP BY, wenn das Feld unüblich zum Definieren von Gruppen ist, so wie z.B. Nachname. Normalerweise erscheinen die Datensätze in aufsteigender Reihenfolge. Wenn Sie es in absteigender wünschen, fügen Sie den Befehl DESC ein, also z.B. 'ORDER BY Rechnungen.Datum DESC'.														

Erzeugen einer REAL-Datenbank

Folgender Code erzeugt eine REAL-Datenbank und verwendet SQLExecute, um eine Tabelle zu erzeugen:

```
Dim db as REALDatabase
Dim f as FolderItem
Dim result as Boolean
```

```
f=New FolderItem("mydb")
db=New REALDatabase
db.databaseFile=f
result=db.CreateDatabaseFile
If db.Connect() then
    db.SQLExecute("create table invoices(id integer not null, Cust_ID integer not null,
    Amount double, Date date, primary key (id))")
else
    MsgBox "Datenbank nicht erzeugt"
end if
```

Folgendes Beispiel fügt einen Datensatz in diese Tabelle ein:

```
Dim r as DatabaseRecord
Dim mybool as Boolean
Dim mydate as Date
.
.
r=New DatabaseRecord
r.IntegerColumn("id")=4
r.IntegerColumn("Cust_ID")=2
r.DoubleColumn("Amount") =9.98
mybool=ParseDate("10/22/98",mydate)
r.DateColumn("Date") = mydate
db.InsertRecord("invoices",r)
```

RecordSets

Folgendes Beispiel erzeugt ein RecordSet, das alle Zeilen und Spalten für einen einzelnen Kunden nach Menge sortiert enthält:

```
Dim rs as DatabaseCursor
.
.
rs = db.SQLSelect("select * from invoices where cust_id=02 ORDER BY amount DESC")
```

Folgender Select-Befehl liefert alle Datensätze und Spalten der Kunden-Tabelle in der der Nachname des Kunden mit einem „L“ beginnt.

```
rs=db.SQLSelect("select * from Customers WHERE customer.name LIKE 'L%'")
```

Dieser Select-Befehl liefert alle Datensätze und Spalten, in denen der Nachname des Kunden nicht mit dem Buchstaben „L“ beginnt.

```
rs=db.SQLSelect("select * from customer WHERE customer.name NOT LIKE 'L%'")
```

Folgendes Beispiel liefert ein RecordSet, das Informationen über die Felder der Kundentabelle enthält:

```
Dim rs as RecordSet
dim dbFile as FolderItem
dim db as Database
.
.
dbFile = GetFolderItem("test.rdb")
db=OpenREALDatabase(dbFile)
rs = db.FieldSchema("customers")
```

Empfangen des ConnectionStrings von einem 4D-Server

Sie können die Methode Property verwenden, um denConnectionString von einem 4D Server anzufordern. Nachdem Sie den ConnectionString haben, können Sie direkt mit Open4DDatabasebyADSP oder Open4DDatabasebyTCPIP auf die Datenbank zugreifen anstatt daß Sie diese aus der Dialogbox auswählen müssen, die Ihnen durch Select4DDatabaseByADSP präsentiert wird.

```
Dim db as Database
Dim theString as String
db=Select4DDatabaseByADSP ("getting the connectionstring","Dave","Gorilla")
If db.Connect() then
    theString=db.Property("connectionstring")
end if
db.close
```

Siehe auch

DatabaseRecord, RecordSet, DatabaseField, ADSP4DServer, MySQLDatabase, ODBCDatabase, OpenBaseDatabase, OracleDatabase, PostGRESQlDatabase, REALdatabase, TCPIP4DServer Klassen; OpenCSVcursor, OpenDBFCursor, OpenDTFDatabase, OpenOracleDatabase Funktionen.

DatabaseCursor Klasse

Wird zum Erzeugen und Verwalten von Datenbank-Cursoren verwendet. Diese Klasse ist obsolet und wurde durch `RecordSet` ersetzt. Sie wird nur aus Kompatibilitätsgründen zu früheren REALbasic-Versionen beibehalten. Bitte verwenden Sie stattdessen die `RecordSet`-Klasse. Funktionen, die in früheren Versionen einen **DatabaseCursor** lieferten, liefern nun ein `RecordSet`.

Hinweise

In SQL-Jargon ist ein *Cursor* eine Gruppe von Datensätzen. In diesem Kontext bedeutet das Wort also nicht Mauszeiger oder Schreibcursor einer Textverarbeitung. Die Bedeutung des Wortes „Cursor“ im Zusammenhang mit den Funktionen `OpenCSVCursor` und `OpenDFBCursor` unterscheidet sich völlig von der Bedeutung im Zusammenhang mit der `MouseCursor` Klasse.

REALbasic verwendet den Ausdruck *RecordSet*, um eine Verwechslung zu vermeiden. Wenn Sie jedoch ein SQL-Buch lesen, werden Sie auf den Ausdruck „Cursor“ stoßen. Denken Sie einfach daran, daß es sich hierbei um eine Gruppe von Datensätzen in einer SQL-Datenbank handelt – und nicht um Ihren Schreibcursor.

Siehe auch

`RecordSet` Klasse.

DatabaseCursorField Klasse

Ein **DatabaseCursorField** wurde verwendet, um den Inhalt von Datenbankfeldern zu ermitteln. Diese Klasse ist obsolet und wurde durch die `DatabaseField`-Klasse ersetzt.

Siehe auch

`DatabaseField`, `RecordSet` Klassen.

DatabaseField Klasse

Wird verwendet, um auf Werte von Feldern in einem Datensatz in einer Datenbank-Tabelle zuzugreifen. Verwenden Sie diese Klasse anstelle der `DatabaseCursorField` Klasse.

Übergeordnete Klasse

`Object`

Eigenschaften

Name	Typ	Beschreibung
<i>BooleanValue</i>	<u>Boolean</u>	Wird zum Setzen und Ermitteln von Werten des Feldtyps <u>Boolean</u> verwendet. Bitte verwenden Sie stattdessen die Value-Eigenschaft.
<i>DateValue</i>	<u>Date</u>	Wird zum Setzen und Ermitteln von Werten des Feldtyps <u>Date</u> verwendet. Bitte verwenden Sie stattdessen die Value-Eigenschaft.
<i>DoubleValue</i>	<u>Double</u>	Wird zum Setzen und Ermitteln von Werten des Feldtyps <u>Double</u> verwendet. Bitte verwenden Sie stattdessen die Value-Eigenschaft.
<i>IntegerValue</i>	<u>Integer</u>	Wird zum Setzen und Ermitteln von Werten des Feldtyps <u>Integer</u> verwendet. Bitte verwenden Sie stattdessen die Value-Eigenschaft.
<i>StringValue</i>	<u>String</u>	Wird zum Setzen und Ermitteln von Werten des Feldtyps <u>String</u> /Character verwendet. Bitte verwenden Sie stattdessen die Value-Eigenschaft.
<i>Value</i>	<u>Variant</u>	Wird zum Setzen und Ermitteln von Werten eines Feldes eines beliebigen Datentyps verwendet. Setzen Sie dies auf <u>Nil</u> , um das Feld auf NULL zu setzen. Der Wert NULL wird derzeit in den Plugins für MySQL, OpenBase und PostgreSQL unterstützt.

Hinweis

Wenn Sie der Datum-Eigenschaft einen Wert zuweisen, wird die Zeit ebenfalls zugewiesen, wie bei einem `DatabaseRecord`. Wenn Sie die Datum-Eigenschaft auslesen, bekommen Sie ein `Date`-Objekt, das ebenfalls auch die Uhrzeit enthält. Für Felder vom Typ Datum

ist die Uhrzeit auf 00:00:00 (Mitternacht) gesetzt. Für Felder vom Typ Uhrzeit ist das Datum auf den ersten Januar 0001 gesetzt. Felder vom Typ TimeStamp enthalten sowohl Uhrzeit, als auch Datum.

Beispiele

Folgendes Beispiel verwendet die Value-Eigenschaft, um die Werte von Feldern unterschiedlicher Datentypen zu setzen.

```
Dim MyDB as Database
Dim rs as RecordSet
rs = MyDB.SQLSelect("select * from mytable")
rs.Edit
rs.field("MyName").value = Nil // Dieses Feld NULLen,
//arbeitet nicht bei allen Database-Plugins
rs.field("MyID").value = 23
rs.field("MyText").value = "Test"
rs.update
MyDB.Commit
```

Folgende Methode füllt eine ListBox mit einem RecordSet. Sie verwendet die Eigenschaften *Name* und *StringValue*, um die Feldnamen und Werte zu ermitteln:

```
Sub populateCursor(d as DatabaseCursor)
    dim i as integer
    dim v as string

    ListBox1.deleteAllRows
    ListBox1.columncount = d.fieldcount
    ListBox1.addrow d.IdxField(1).Name
    ListBox1.cellBold(Listbox1.lastIndex, 0) = true
    for i = 2 to d.fieldcount
        ListBox1.cell(ListBox1.lastIndex,i-1)=d.IdxField(i).name
        ListBox1.cellBold(Listbox1.lastIndex, i-1) = true
    next
    While not d.eof
        v = d.IdxField(1).StringValue
        ListBox1.addrow v
        for i = 2 to d.fieldcount
            ListBox1.cell(ListBox1.lastIndex,i-1)=d.IdxField(i).StringValue
        next
        d.MoveNext
    wend
End Sub
```

Siehe auch

Database, DatabaseRecord, RecordSet Klassen.

DatabaseQuery Steuerelement

Das Steuerelement, mit dem Sie Datenbankabfragen durchführen. Sie verwenden das DatabaseQuery-Steuerelement, indem Sie es mit einer ListBox verknüpfen, welche zur Anzeige der Ergebnisse der SQL-Abfragen verwendet wird.

Alternativ dazu können Sie Datenbanken ohne DatabaseQuery-Steuerelement aufbauen, indem Sie nur die Datenbank-Befehle von REALbasic verwenden.

Da Object die übergeordnete Klasse von DatabaseQuery ist, können Sie eine Instanz des Steuerelements im Programmcode erzeugen und müssen es nicht unbedingt auf einem Fenster ablegen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
Database	<u>Database</u>	Die Datenbank, in der eine Abfrage ausgeführt werden soll.
SQLQuery	<u>String</u>	Der Text der SQL-Abfrage, die auf Database angewendet werden soll.

Events

Name	Parameter	Beschreibung
<i>QueryComplete</i>		Wird ausgeführt, wenn die Abfrage beendet ist.

Methoden

Name	Parameter	Beschreibung
<i>RunQuery</i>		Führt die durch die <i>SQLQuery</i> -Eigenschaft festgelegte Abfrage aus.

Beispiele

Siehe Beispiel im Kapitel „Datenbankprogrammierung mit REALbasic“ im Entwickler-Handbuch.

Siehe auch

[Database](#), [RecordSet](#), [DatabaseField](#), [DatabaseRecord](#), [ADSP4DServer](#), [MySQLDatabase](#), [ODBCDatabase](#), [OpenBaseDatabase](#), [PostgreSQLDatabase](#), [REALdatabase](#), [TCPIP4DServer](#) Klassen; [OpenCSVCursor](#), [OpenDBFCursor](#), [OpenDTFDatabase](#), [OpenOracleDatabase](#) Funktionen; [DataControl](#) Steuerelement.

DataBaseRecord Klasse

Wird verwendet, um neue Datensätze zu erzeugen. Mit der Column-Methode werden die Felder des Datensatzes gefüllt. Rufen Sie *Column* für jede Spalte des Datensatzes auf.

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
<i>BlobColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf das angegebene BLOB.
<i>BooleanColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf den angegebenen boole'schen Wert.
<i>Column</i>	Name als String	Spalte in der aktuellen Tabelle.
<i>DateColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf den angegebenen Datumswert.
<i>DoubleColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf den angegebenen Double-Wert.
<i>IntegerColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf den angegebenen Integer-Wert.
<i>JPEGColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf das angegebene JPEG-Bild.
<i>MacPictColumn</i>	Name als String	Der Parameter ist der Spaltenname. Er setzt das angegebene Feld auf das angegebene-Macintosh PICT Bild.

Hinweise

Zuweisungen zur *DateColumn*-Eigenschaft speichern sowohl die Uhrzeit als auch das Datum, um die Feldtypen SQL TimeStamp und Time (als auch Date) zu unterstützen. Denken Sie daran, daß, wenn der Datums-Teil der 1. Januar 0001 ist, nur die Uhrzeit nach SQL konvertiert wird (z.B. „18:54:00“), wogegen, wenn das Datum ein anderes ist, es in der vollen Form nach SQL konvertiert wird (z.B. „2000-5-30 18:54:00“).

Beispiel

Folgendes Beispiel erzeugt eine neue REAL-Datenbank mit einer Mitarbeiter-Tabelle und fügt einen Datensatz in die Tabelle ein.

```
Dim mybool as Boolean
Dim mydate as Date
Dim rec as DatabaseRecord
```

```
dbFile = GetFolderItem("test.rdb")
db = NewREALDatabase(dbFile)
db.SQLExecute("create table employees(id integer not null, name varchar, jobtitle varchar, DOB
date, Salary double, primary key (id))")
rec = New DatabaseRecord

rec.IntegerColumn("id") =1
rec.Column("name") = "Lois Lane"
rec.Column("jobtitle")="Pundit"
mybool=ParseDate("1Jan1950",mydate)
rec.DateColumn("DOB")=mydate
rec.DoubleColumn("Salary")=105000
db.InsertRecord("employees",rec)
```

Siehe auch

[Database](#), [RecordSet](#), [DatabaseField](#), [ADSP4DServer](#), [TCPIP4DServer](#), [REALdatabase](#), [ODBCDatabase](#), [OpenBaseDatabase](#), [PostgreSQLDatabase](#) Klassen; [OpenCSVCursor](#), [OpenDBFCursor](#), [OpenDTFDatabase](#), [OpenOracleDatabase](#) Funktionen.

DataControl Steuerelement



Mit dem DataControl Steuerelement kann der Anwender durch die Datensätze einer Datenbank blättern, indem er auf Navigationsknöpfe klickt. Es stehen Knöpfe zur Verfügung, um auf den ersten, den vorigen, den nächsten oder den letzten Datensatz zu springen. Außerdem wird die Caption-Eigenschaft als Text ausgegeben.

Wenn der Anwender auf einen der Knöpfe klickt, werden in das Steuerelement eingebaute Events ausgelöst. Die zur Verfügung stehenden Methoden erlauben das programmgesteuerte Navigieren durch die Datensätze.

Das **DataControl** ist dafür konzipiert worden, Daten unter Verwendung von [EditFields](#), [ListBoxes](#), [StaticTexts](#) und [PopupMenu](#)s anzuzeigen. Sie können auch eigene Klassen erzeugen, die auf einer dieser Klassen basieren und diese in Verbindung mit einem **DataControl** verwenden. Jedes dieser Steuerelemente (oder deren Unterklasse) besitzt zwei Eigenschaften, die für das **DataControl** spezifisch sind: *DataSource* und *DataField*. *DataSource* erwartet ein DataControl-Objekt und *DataField* den Namen eines Feldes der Tabelle, mit der das DataControl verbunden ist. Sie erzeugen ein Datenbank-Interface, indem Sie ein **DataControl** mit der Tabelle einer Datenbank verbinden und jedes Datenfeld, das angezeigt werden soll, mit einem [EditField](#)-, [ListBox](#)-, [StaticText](#)-, oder [PopupMenu](#)-Objekt verbinden, indem Sie dessen *DataField*- und *DataSource*-Eigenschaften entsprechend belegen.

Übergeordnete Klasse

[RectControl](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Caption</i>	String	Dieser Text wird zwischen den beiden Gruppen von Navigationsknöpfen angezeigt.
<i>Cursor</i>	RecordSet	Der Datenbank-Cursor, der durch das DataControl verwaltet wird. Bitte verwenden Sie stattdessen die Eigenschaft RecordSet.
<i>DataBase</i>	Database	Die Datenbank, mit der das Steuerelement verbunden ist.
<i>ReadOnly</i>	Boolean	Wenn Sie diese Eigenschaft auf True setzen, kann der Anwender keine Datensätze hinzufügen, verändern oder löschen.
<i>RecordLocked</i>	Boolean	Wenn Sie diese Eigenschaft auf True setzen, sind die Datensätze gesperrt, so daß keine Änderungen daran vorgenommen werden können; bei False ist der Datensatz nicht gesperrt.
<i>RecordSet</i>	RecordSet	Das RecordSet (der Cursor), der durch das DataControl verwaltet wird.
<i>SQLQuery</i>	String	SQL-Abfrage, die einen Datenbank-Cursor liefert.
<i>TableName</i>	String	Name einer Tabelle der <i>Database</i> .

Events

Name	Parameter	Beschreibung
<i>Delete</i>		Tritt auf, falls der Datensatz als gelöscht validiert wurde. Zu diesem Zeitpunkt wurde der Datensatz bereits gelöscht.
<i>Insert</i>		Tritt auf, nachdem der neue Datensatz validiert wurde und bevor die Eingabefelder geleert werden.
<i>MoveFirst</i>		Tritt auf, nachdem der Navigationsknopf „Erster Datensatz“ gedrückt wurde und bevor der Datensatzzeiger auf den ersten Datensatz bewegt wird. Indem Sie <u>True</u> zurückgeben, verhindern Sie das Wechseln auf den ersten Datensatz.
<i>MoveLast</i>		Tritt auf, nachdem der Navigationsknopf „Letzter Datensatz“ gedrückt wurde und bevor der Datensatzzeiger auf den letzten Datensatz bewegt wird. Indem Sie <u>True</u> zurückgeben, verhindern Sie das Wechseln auf den letzten Datensatz.
<i>MoveNext</i>		Tritt auf, nachdem der Navigationsknopf „Nächster Datensatz“ gedrückt wurde und bevor der Datensatzzeiger auf den nächsten Datensatz bewegt wird. Indem Sie <u>True</u> zurückgeben, verhindern Sie das Wechseln auf den nächsten Datensatz.
<i>MovePrevious</i>		Tritt auf, nachdem der Navigationsknopf „Voriger Datensatz“ gedrückt wurde und bevor der Datensatzzeiger auf den vorigen Datensatz bewegt wird. Indem Sie <u>True</u> zurückgeben, verhindern Sie das Wechseln auf den vorigen Datensatz.
<i>Reposition</i>	Index als <u>Integer</u>	Tritt nach der Initialisierung auf und wenn der Datensatzzeiger auf einen anderen Datensatz bewegt wurde. Der Parameter Index enthält die Datensatznummer nach der Reposition.
<i>Update</i>		Tritt auf, wenn der Datensatz zum Aktualisieren validiert ist. Zu diesem Zeitpunkt wurde der Datensatz bereits aktualisiert.
<i>Validate</i>	Aktion als <u>Integer</u>	Tritt auf, wenn Sie den aktuellen Datensatz mittels <u>EditFields</u>, die eine Eingabe direkt verarbeiten, modifiziert haben. Indem Sie <u>False</u> (Vorgabe) zurückgeben, werden die Änderungen in den Datensatz übernommen. Rückgeben von <u>True</u> verhindert die Aktualisierung. Der Parameter Aktion legt fest, welche Aktion ausgeführt wird: 0=AddNew (Neuen Datensatz hinzufügen) 1=Update (Datensatz aktualisieren) 2=Delete (Datensatz löschen)

Methoden

Methode	Parameter	Beschreibung
<i>Cursor</i>		Liefert ein <u>DatabaseCursor</u> -Objekt zurück.
<i>Delete</i>		Löscht den aktuell ausgewählten Datensatz. Wenn möglich, verbleibt der Cursor in der selben Zeile. Falls der letzte Datensatz gelöscht wird, wird der Cursor auf den vorigen Datensatz gesetzt. Delete löst folgende Events aus: Validate Reposition Delete
<i>FieldCount</i>		Liefert die Anzahl der Felder in <i>TableName</i> zurück.
<i>Insert</i>		Fügt normalerweise einen neuen Datensatz ein, schließt den Cursor und leert die Eingabefelder. Dieses Verhalten kann im Insert-Event modifiziert werden. Insert löst folgende Events aus: Validate (und Schließen des Cursors) Insert (und Löschen der Eingabefelder)

Methode	Parameter	Beschreibung
<i>MoveFirst</i>		Springt auf den ersten Datensatz im <u>DatabaseCursor</u> .
<i>MoveLast</i>		Springt auf den letzten Datensatz im <u>DatabaseCursor</u> .
<i>MoveNext</i>		Springt auf den nächsten Datensatz im <u>DatabaseCursor</u> .
<i>MovePrevious</i>		Springt auf den vorigen Datensatz im <u>DatabaseCursor</u> .
<i>MoveTo</i>	Index als <u>Integer</u>	Springt auf den Datensatz <i>Index</i> im <u>DatabaseCursor</u> .
<i>RecordCount</i>		Liefert die Anzahl der Records in <i>TableName</i> .
<i>RecordSet</i>		Liefert ein RecordSet.
<i>Row</i>		Liefert den aktuellen Datensatz im <u>DatabaseCursor</u> .
<i>RowCount</i>		Liefert die Datensatzanzahl von <i>TableName</i> .
<i>RunQuery</i>		Füllt den <u>DatabaseCursor</u> mit Daten, indem <i>SQLQuery</i> ausgeführt wird.
<i>Update</i>		Aktualisiert den ausgewählten Datensatz, der Cursor bleibt in der selben Zeile. Update löst folgende Events aus: Validate Update

Hinweise

Ein DataControl-Steuerelement in einem Fenster sieht so aus:



Sie verwenden ein DataControl, indem Sie ein Fenster erzeugen, das EditFields, ListBoxes, Popup-Menüs oder StaticText-Steuerelemente verwendet, um die Daten anzuzeigen und zu bearbeiten. Jedes dieser Steuerelemente besitzt zwei Eigenschaften, die nur in Verbindung mit einem DataControl von Bedeutung sind.

Damit das DataControl mit einer REAL Datenbankquelle zusammenarbeiten kann, muß die Tabelle über einen Primärschlüssel verfügen.

Bei einem Primärschlüssel handelt es sich um ein Feld, das nicht leer sein darf und dessen Werte eindeutig sein müssen. Eine einfache Möglichkeit, den Primärschlüssel mit Daten zu bestücken, ist die fortlaufende Numerierung. Siehe Beispiel weiter unten.

Mit Hilfe von Objektverknüpfungen können Objekte wie z.B. PushButtons mit einem DataControl verbunden werden. Unterstützte Aktionen sind: Add (Hinzufügen), Update (Aktualisieren) und Delete Record (Datensatz löschen). Diese können ausgeführt werden, wenn ein PushButton gedrückt wird. Um die Objektverknüpfung einzurichten, halten Sie die Shift- und Befehlstaste gedrückt, klicken auf das Objekt (z.B. den PushButton) und ziehen die Maus bei weiterhin gedrückten Tasten auf das DataControl.

Wenn Sie ein Interface mit Hilfe eines DataControl bauen, ist es hiermit einfach, Datensätze zu bearbeiten und zu speichern. Wenn der Anwender die Änderungen speichert, werden die Werte in der Datenbank automatisch durch die Daten der Steuerelemente aktualisiert, die mit dem DataControl verbunden sind. Wenn Sie andere Werte aktualisieren wollen, müssen Sie auf konventionelle Weise die RecordSet-Methode des DataControl verwenden.

Beispiel

Die folgende Beispieldatenbank verwendet EditFields zur Anzeige und ein **DataControl** zur Navigation.

Jedes EditField ist mit dem **DataControl** verbunden, indem im Eigenschaftfenster die **DataSource**- und **DataField**-Eigenschaften entsprechend gesetzt wurden. Nach dem Setzen der **DataSource**-Eigenschaft erscheint in der **DataField**-Eigenschaft ein PopUp-Menü mit den Datenfeldern der Tabelle, die über Database festgelegt wurde. Die folgende Abbildung zeigt, daß das EditField für „First Name“ mit der Spalte „first“ der Tabelle verbunden ist, zu der das **DataControl** gehört.

Der Primärschlüssel „ID“ ist in diesem Fall mit einem EditField names „IDField“ verbunden.

Der Knopf „Add Record“ ist mittels Objektverknüpfung mit dem **DataControl** verbunden. Um die Verknüpfung herzustellen, halten Sie die Shift- und Befehlstaste gedrückt, klicken auf den PushButton und ziehen bei weiterhin gedrückten Tasten die Maus zum **DataControl**. Es erscheint der Dialog „Neue Verknüpfung“. Wählen Sie hier die Aktion „Datensatz einfügen when addButton gedrückt“.

Gehen Sie analog vor, um den „Delete“-Knopf an das **DataControl** zu binden und verwenden Sie dort die Aktion „Datensatz löschen when DeleteButton gedrückt.“

Verwenden Sie im Validate-Event des **DataControl** folgenden Code, um den Primärschlüssel um 1 zu erhöhen:

```
Dim dbc as RecordSet
If action = 0 then
    dbc = Me.database.SQLSelect("Select Max(id) from " + me.tableName)
    idfield.text = Str(dbc.idxField(1).integerValue + 1)
end if
```

Siehe auch

Database, DatabaseRecord, RecordSet Klassen.

Datagram Klasse

Wird verwendet, um Daten mit Hilfe der [UDPSocket](#)-Klasse zu senden und zu empfangen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Adresse</i>	String	Die Adresse des entfernten Rechners, an den Sie Daten senden oder von dem Sie Daten empfangen.
<i>Daten</i>	String	Die Daten, die Sie senden oder empfangen.

Hinweise

Ein **Datagram** besteht aus zwei Teilen, der IP-Adresse des entfernten Rechners, an den Sie Daten senden und den Daten selbst. Wenn Sie versuchen, Daten zu senden, müssen Sie die Information in Form eines Datagram festlegen. Diese Information ist normalerweise die Remote-Adresse des Rechners, der Ihr Paket empfangen soll, der Port, an den es gesendet werden soll und die Daten, die Sie an den entfernten Rechner senden wollen. Weitere Informationen, wie man das Datagram verwendet, finden Sie im Abschnitt [UDPSocket](#).

Siehe auch

[UDPSocket](#), [SocketCore](#) Klassen.

DataNotificationReceiver Interface Klasse

Wird verwendet, um benutzerdefinierte Objektverknüpfungen zu erzeugen.

Methoden

Name	Parameter	Beschreibung
<i>Reload</i>		Lädt Data Notifier nach Änderungen erneut.
<i>Commit</i>		Sichert Änderungen der Daten.

DataNotifier Interface Klasse

Wird verwendet, um benutzerdefinierte Objektverknüpfungen zu erzeugen.

Methoden

Name	Parameter	Beschreibung
<i>addDataNotificationReceiver</i>	receiver als DataNotificationReceiver	Fügt der Interface-Klasse einen Data Notification Receiver hinzu.
<i>removeDataNotificationReceiver</i>	receiver als DataNotificationReceiver	Entfernt einen Data Notification Receiver aus der Interface-Klasse.

Date Klasse

Ein **Date**-Objekt speichert die Anzahl der Sekunden seit dem 1. Januar 1904. Die Eigenschaften eines **Date** erlauben Ihnen, nur einen Tag-Wert, ein Datum und eine Uhrzeit oder nur eine Uhrzeit zu setzen oder zu ermitteln.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
AbbreviatedDate	<u>String</u>	Beispiel: Mit, 31. Dez 1997.
<i>Day</i>	<u>Integer</u>	Der Tag (Zahl) im Monat des Datums.
DayOfWeek	<u>Integer</u>	1=Sonntag, 7=Samstag.
DayOfYear	<u>Integer</u>	Der Tag des Jahres.
<i>Hour</i>	<u>Integer</u>	Die Stunde aus der Uhrzeit des Datums.
LongDate	<u>String</u>	Beispiel: Mittwoch, 31. Dezember 1997
LongTime	<u>String</u>	Beispiel: 14:32:40 Uhr.
<i>Minute</i>	<u>Integer</u>	Die Minute aus der Uhrzeit des Datums.
<i>Month</i>	<u>Integer</u>	Der Monat des Datums als Zahlwert.
<i>Second</i>	<u>Integer</u>	Die Sekunde aus der Uhrzeit des Datums.
ShortDate	<u>String</u>	Beispiel: 31.12.1997.
ShortTime	<u>String</u>	Beispiel: 14:32 Uhr.
<i>TotalSeconds</i>	<u>Double</u>	Die Anzahl von Sekunden seit dem 1. Januar 1904. TotalSeconds ist die „Haupt“-Eigenschaft, aus der alle anderen Formen von Datum und Uhrzeit abgeleitet sind. Ein negativer Wert von TotalSeconds bedeutet ein Datum oder eine Zeit vor dem 1. Januar 1904.
WeekOfYear	<u>Integer</u>	Die Kalenderwoche des Datums.
<i>Year</i>	<u>Integer</u>	Das Jahr des Datums.

Hinweise

Wenn Sie ein Date-Objekt instanzieren, wird es mit dem aktuellen Datum und Uhrzeit initialisiert. Wenn Sie das Datum oder Datum/Uhrzeit setzen und ein ungültiges Format verwenden, wird REALbasic eine UnsupportedFormatException auslösen.

Auf die Datums-Eigenschaften von FolderItems kann mit CreationDate und ModificationDate zugegriffen werden. Sie können aktuelles Datum und Uhrzeit abfragen, indem Sie ein neues **Date** anlegen und die Werte der Eigenschaften von Year, Month, Day, Hour, Minute und Second auslesen.

Im folgenden Code speichert das Variant den Wert der TotalSeconds-Eigenschaft als ein Double, zusammen mit der Typ-Information, daß es sich um ein Date handelt (die Type-Eigenschaft des Variants ist 7):

```
Dim v as Variant
Dim d as Date
d=New Date
v=d
```

Obwohl Date eine Klasse ist, die von Object abgeleitet ist, identifiziert VarType ein Date als einen Date-Datentyp (Type=7) und nicht als ein Object (Type=9).

Verwenden Sie die Funktion ParseDate, um einen Datums-String in einen Date-Wert zu konvertieren.

Die Eigenschaft TotalSeconds ist die „Haupt“-Eigenschaft, die das Datum und die Zeit eines Date speichert. Die anderen Eigenschaftswerte werden von TotalSeconds abgeleitet. Wenn Sie den Wert der Eigenschaft TotalSeconds ändern, ändern sich die Werte der Eigenschaften von Year, Month, Day, Hour, Minute und Second auf die Sekunde, auf die TotalSeconds zutrifft. Umgekehrt, wenn Sie eine dieser Eigenschaften ändern, ändert sich TotalSeconds ebenfalls entsprechend.

Die Date-Eigenschaften, die formatierte Datums- oder Zeitinformationen zurückliefern, werden von den Einstellungen beeinflusst, die der Anwender im Kontrollfeld Datum & Uhrzeit vorgenommen hat.



Wenn Sie das exakte Aussehen der Datum/Zeit-Informationen steuern müssen, ist es das Beste, die Informationen selbst zu extrahieren und die Formatierung mittels [String](#)-Manipulations-Funktionen selbst vorzunehmen.

Beispiele

Dieses Beispiel erzeugt ein Date-Objekt und stellt das aktuelle Datum in einer MsgBox dar.

```
Dim d as Date
d=New Date
MsgBox d.shortdate
```

Sie können das Beispiel verkürzen, indem Sie die Instanzierung mit dem Dim-Ausdruck verbinden:

```
Dim d as New Date
MsgBox d.shortdate
```

Dieses Beispiel setzt ein Datum-Objekt auf das Datum vom 10 Februar 1954.

```
Dim d as New Date
d.Year=1954
d.Month=2
d.Day=10
MsgBox d.ShortDate
```

Siehe auch

[ParseDate](#), [Ticks](#), [Microseconds](#) Funktionen; [FolderItem](#) Klasse.

DebugBuild Konstante

Wird verwendet, um festzustellen, ob es sich bei der momentanen Laufzeitumgebung um die IDE handelt.

Syntax

Ergebnis=**DebugBuild**

Teil	Typ	Beschreibung
<i>Ergebnis</i>	Boolean	True, wenn die Applikation innerhalb von REALbasic läuft, Sie also Debug/Programm starten gewählt haben.

Beispiel

Folgendes Beispiel zeigt eine Messagebox nur in der IDE an:

```
#if DebugBuild then
  MsgBox "Sie befinden sich in der IDE"
#endif
```

Siehe auch

[TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetWin32](#), [TargetLinux](#), [RBVersion](#), [RBVersionString](#) Konstanten; [#If](#) Ausdruck.

DebugDumpObjects Methode

Wird zum Aufspüren von Speicherlecks verwendet.

Syntax

DebugDumpObjects(filename)

Teil	Art	Beschreibung
<i>filename</i>	<u>String</u>	Der Name der zu verwendenden Textdatei

Hinweis

REALbasic verwaltet Speicher selbstständig. Dies bedeutet, daß es Speicher für neue Objekte belegt, den Einsatz momentan verwendeter Objekte protokolliert und Objekte aus dem Speicher entfernt, sobald diese nicht mehr benötigt werden.

Es kann Situationen geben, in denen REALbasic Objekte nicht aus dem Speicher entfernt, woraufhin eine Anwendung zur Laufzeit immer mehr Speicher belegt. Ihr Programmcode kann solche Situationen provozieren. Sehen Sie sich hierzu das Beispiel bei StackOverflowException an.

Die **DebugDumpObjects** Methode kann Ihnen helfen herauszufinden, ob Ihre Anwendung ein solches Speicherleck aufweist. Wenn Sie **DebugDumpObjects** aufrufen, schreibt diese Methode eine Textdatei, die Sie im Ordner ihrer Anwendung oder im Ordner von REALbasic (falls Sie ihre Anwendung aus der Entwicklungsumgebung heraus starten) finden. Diese Textdatei enthält Angaben darüber, wie viel Speicher momentan von Objekten wie etwa Fenstern, Steuerelementen, Klassen usw. belegt wird. Diese Textdatei zählt ebenso alle Objekte auf, die momentan Speicher verwenden. Sie können ein Speicherleck entdecken, indem Sie diese Methode an verschiedenen Stellen ihrer Anwendung aufrufen und die dabei geschriebenen Textdateien miteinander vergleichen und so herausfinden, ob die Objekte im Speicher auch diejenigen sind, die Sie dort erwarten sollten.

Beispiel

Folgende Code-Zeile erzeugt die Textdatei wie oben beschrieben:

```
DebugDumpObjects ("dumpfile")
```

Declare Befehl

Wird verwendet, um Toolbox-Aufrufe zu erzeugen. Es werden derzeit Rechner mit PPC und X86 unterstützt. REALbasic Versionen vor 3.5.2 (inklusive) unterstützen außerdem 68K-Rechner. DLLs können aus einer erzeugten Windows-Applikation aufgerufen werden. „shared libraries“ werden nur unter Macintosh und Linux unterstützt.

Syntax

Declare Sub/Function Name Lib Fragment [Alias AliasName] ([Parameters]) [As ReturnType]

Teil	Typ	Beschreibung
<i>Name</i>	Name	Name des Toolbox-Aufrufs. Verwenden Sie <i>Name</i> in Ihrem Code, um auf den Toolbox-Aufruf zu verweisen.
<i>Fragment</i>	<u>String</u>	Library, in der der Toolbox-Aufruf zu finden ist.
<i>AliasName</i>	<u>String</u>	Optional: Wenn der Toolbox-Aufruf den gleichen Namen wie eine REALbasic-Methode hat, deklarieren Sie den Aufruf mit einem anderen Namen und verwenden das Alias, um auf die eigentliche Toolbox-Routine zu verweisen.
<i>Parameters</i>		Parameter des Toolbox-Aufrufs.
<i>ReturnType</i>		Optional: Der Datentyp des Wertes, der vom Aufruf zurückgeliefert wird, falls es sich um eine Funktion handelt.
<i>EntryPoint</i>	<u>String</u>	Optional: Wenn Sie 68K-Mac-Unterstützung einbauen möchten, binden Sie den InLine68K-Parameter ein.

Hinweise

Funktionsnamen für das OS (Mac oder Win32) sind case-sensitiv. Zum Beispiel arbeitet SetLocalTime unter Win32, SetlocalTime jedoch nicht.

Declare-Befehle müssen die CarbonLib aufrufen, um unter Mac OS X zu laufen. Wenn sie das nicht tun, läuft die Applikation in der „Classic“-Umgebung unter Mac OS X.

Ein Wert, der einem Declare-Befehl übergeben wird, muß eine Konstante oder eine lokale Variable sein. Es kann nicht das Ergebnis eines Funktionsaufrufs oder einer Eigenschaft sein. Um einen Wert zu übergeben, deklarieren Sie einfach eine Variable und setzen Sie die Eigenschaft des Funktionsergebnisses mit der Variablen gleich.

Der Declare-Befehl erlaubt die Verwendung von Konstanten für Lib-Namen.

Wenn Sie eine Carbon-Applikation im Mach-O-Format erzeugen, akzeptiert der Declare-Ausdruck den Namen des Frameworks statt eines absoluten Pfads. Wenn Sie ein Declare auf das CoreMIDI-Framework anwenden, sieht der Compiler, daß kein absoluter Pfad angegeben ist und wandelt die Angabe in den Pfad „/System/Library/Frameworks/CoreMIDI.framework/CoreMIDI“ um.

Beispiele

Folgendes Beispiel verwendet den Macintosh-Speech-Manager um den Text eines EditFields vorzulesen. Beachten Sie, daß bedingte Comilierung sowohl für den Declare-Befehl als auch für den Toolbox-Aufruf selbst verwendet wird. Normalerweise sollten Sie, wenn die Funktionalität auf beiden Plattformen gegeben ist, die gleichen Namen für den Code verwenden und dann durch Auswerten der TargetBoolean-Konstante die korrekte Version für die aktuelle Plattform aufrufen.

Normalerweise erwartet die Windows-Version eines Toolbox-Aufrufs andere Parameter als die Mac-Version.

```
Dim s as String
Dim i as Integer
#If TargetMacOS
    Declare Function SpeakString lib "SpeechLib" (SpeakString as pstring) as Integer
#Endif

s=EditField1.text
#If TargetMacOS
    i=SpeakString(s)
#Else
    MsgBox "Sprachausgabe wird nur auf dem Mac unterstützt!"
#Endif
```

Namenskonflikte

Wenn SpeakString der gleiche Name wie der einer REALbasic-Methode wäre, müssten Sie das Alias-Schlüsselwort verwenden, um auf den Toolbox-Aufruf zu verweisen. Dies zueig folgendes Beispiel:

```
Declare Function MySpeakString lib "SpeechLib" Alias "SpeakString" (SpeakString as pstring) as integer
.
.
i=MySpeakString(s)
```

Folgendes Beispiel zeigt die maximale Anzahl von Ticks zwischen Mausclicks an, die als zwei Klicks und nicht als Doppelklick gewertet werden. Über diesem Wert werden zwei Klicks als separate Maus-Events behandelt. Auf dem Mac kann diese Zeit im Maus-Kontrollfeld eingestellt werden.

```
Dim DoubleClickTime as Integer
#If TargetMacOS
    Declare Function GetDbtTime Lib "InterfaceLib" () as Integer Inline68K("2EB802F0")
    doubleClickTime = GetDbtTime()
#endif

#If targetWin32
    Declare Function GetDoubleClickTime Lib "User32.DLL" () as Integer
    doubleClickTime = GetDoubleClickTime()
#endif
MsgBox Str(doubleclicktime)
```

Um den Code so zu modifizieren, daß er unter Carbon läuft, ersetzen Sie „InterfaceLib“ durch „CarbonLib“:

```
Dim DoubleClickTime as Integer
#If TargetCarbon
    Declare Function GetDbtTime Lib "CarbonLib" () as Integer
    doubleClickTime = GetDbtTime()
#endif
MsgBox Str(doubleclicktime)
```


Das folgende Beispiel ermittelt die aktuelle ProzessID und verdeutlicht die Benutzung von Declare für Mach-O, Mac OS „Classic“, Windows und Linux.

```
Function GetPid() As Integer
    Dim pid as Integer
    Dim psn as MemoryBlock

    #if TargetWin32
        On Windows, we're going to use the GetCurrentProcessId
        function to figure out our pid
        Declare Function GetCurrentProcessId Lib "Kernel32" () as Integer
        ' Get the pid according to windows
        id = GetCurrentProcessId

    #elseif TargetMachO
        ' For a Mach-O app, we can just use the system's
        ' getpid function in the kernel framework, which
        ' is automatically included in the system framework.
        Declare Function getpid Lib "/System/Library/Frameworks/System.framework/System" _
        () as Integer
        ' Get the pid according to mach
        pid = getpid
    #elseif TargetLinux
        ' On Linux, we get the pid from the libc library. It
        ' is worth noting that I am hard coding the version
        ' of libc in this call. It requires libc version 2.3.2
        Declare Function getpid Lib "libc-2.3.2.so" () as Integer
        ' Get the pid according to the kernel
        pid = getpid

    #elseif TargetCarbon or TargetPPC
        ' On Carbon PEF or Classic we will use the lower
        ' 32 bits of the ProcessSerialNumber.
        Declare Function GetCurrentProcess Lib "CarbonLib" (psn as Ptr) as Integer

        ' Make a 64 bit structure
        psn = New MemoryBlock( 8 )
        ' Get the PSN from it
        Call GetCurrentProcess( psn )
        ' And store off the lower 32 bits (the upper
        ' 32 seemed to always be 0 anyways)
        pid = psn.Long( 4 )
    #endif
    ' Return the pid we found (note, if we didn't
    ' find a pid, it'll return 0)
    Return pid
End Function
```

Siehe auch

[TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetLinux](#), [TargetWin32](#), [TargetMachO](#), Konstanten.

DecodeBase64 Funktion

Decodiert einen Base64-String zurück in seine Ursprungsform.

Syntax

Ergebnis=DecodeBase64(str)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Der decodierte Wert von <i>str</i> .
<i>str</i>	String	Der zu decodierende Base64- String .

Hinweise

Die **DecodeBase64**-Funktion führt die umgekehrte Operation von [EncodeBase64](#) aus. Eine übergebener codierter String, der an **DecodeBase64** übergeben wird, liefert den Ursprungs-String.

Siehe auch

[EncodeBase64](#) Funktion.

DecodeQuotedPrintable Funktion

Decodiert einen String in einen „normalen“ Text, der zuvor mit [EncodeQuotedPrintable](#) codiert wurde.

Syntax

Ergebnis=**DecodeQuotedPrintable**(str)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Der Ergebnis aus der Verarbeitung von <i>str</i> .
<i>str</i>	String	Der String-Ausdruck, der zurück in seine Originalform decodiert werden soll.

Hinweise

DecodeQuotedPrintable konvertiert Texte mit undruckbaren Zeichen (also Steuerzeichen, Returns und Tabs) zurück in die Originalform. Es wurde für die Verarbeitung von E-Mails und Usenet-News entwickelt und macht die Kodierung, die [EncodeQuotedPrintable](#) durchgeführt hat, rückgängig.

Siehe auch

[EncodeQuotedPrintable](#) Funktion.

DefineEncoding Funktion

Liefert einen [String](#) mit den gleichen Daten wie der übergebene String, jedoch mit der gewünschten Kodierung. Diese Funktion ist hilfreich, wenn Sie einen String haben, dessen Kodierung Ihnen, nicht aber REALbasic bekannt ist. REALbasic ist die Kodierung der Texte, die es selbst erzeugt, bekannt. Für diese wird DefineEncoding nicht benötigt.

Syntax

Ergebnis=**DefineEncoding**(str, enc)

oder

Ergebnis=str.**DefineEncoding**(enc)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Das Ergebnis der Kodierung von <i>str</i> unter Verwendung der durch <i>enc</i> festgelegten Kodierung.
<i>str</i>	String	Der zu codierende String .
<i>enc</i>	TextEncoding	Das für die Kodierung von <i>str</i> zu verwendende TextEncoding.

Hinweise

In der gedruckten und in der PDF-Version der Sprachreferenz finden Sie im Anhang A eine Liste der *Base*-Werte, die Sie verwenden müssen, wenn Sie ein [TextEncoding](#)-Objekt mit Hilfe der [GetTextEncoding](#)-Funktion erzeugen.

Beispiele

Das folgende Beispiel verwendet 8 Bytes aus einem [MemoryBlock](#) und setzt deren Kodierung auf UTF16.

```
EditField1.text=DefineEncoding(MyMemoryBlock.StringValue(0,8),Encodings.UTF16 )
```

Dieses Beispiel benutzt **DefineEncoding** beim Lesen von einer [TCPSocket](#).

```
EditField1.Text = DefineEncoding( TCPSocket1.ReadAll, Encodings.ASCII )
```

Siehe auch

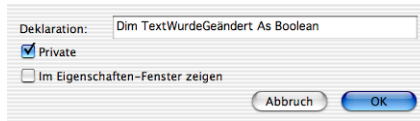
[TextEncoding](#) Klasse; [GetTextEncoding](#), [ConvertEncoding](#), [Encoding](#) Funktionen; [Encodings](#) Objekt.

Deklaration Enthält ein reserviertes Wort

Sie haben einen Dim-Befehl beim Deklarieren einer Eigenschaft in der Neue-Eigenschaft-Dialogbox verwendet.

Beispiel

Das Dim-Schlüsselwort sollte bei folgender Deklaration „Dim TextHasChanged as Boolean“, weggelassen werden:



Der New Operator kann nicht mit einem Klassen-Interface verwendet werden Fehler

Sie haben den New-Operator mit einem Klassen-Interface verwendet. Neue Klassen-Interfaces erzeugt man mit dem Menüpunkt **Ablage/Neues Klassen-Interface**.

Beispiel

Das Klassen-Interface Interface1 wurde bereits dem Projekt hinzugefügt.

```
Dim c as Interface1
c=New Interface1
```

DesktopFolder Funktion

Wird benutzt, um Zugriff auf die Schreibtischobjekte zu bekommen. Unter Windows greift es auf das Desktop-Verzeichnis zu.

Syntax

Ergebnis = **DesktopFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein <u>FolderItem</u> , das ein Schreibtischobjekt repräsentiert.

Hinweis

Verwenden Sie die Funktion **DesktopFolder**, wenn Sie auf Schreibtischobjekte zugreifen müssen. Obwohl Schreibtischobjekte nicht in einem Ordner zu sein scheinen, werden diese in einem Ordner „Desktop Folder“ verwaltet.

Volumes und der Papierkorb finden Sie nicht im Schreibtischordner, sehr wohl aber Druckersymbole. Auf den Papierkorb können Sie mit der Funktion TrashFolder zugreifen.

Die Funktion **DesktopFolder** bietet eine Möglichkeit, auf die Einträge des Schreibtischordners zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

Unter Windows wird auf das Desktop-Verzeichnis zugegriffen. Unter Linux wird das Desktop-Verzeichnis des WindowManagers benutzt. Dies ist entweder ein Verzeichnis namens „Desktop“ oder „gnome-desktop“, welches sich im Home-Verzeichnis befindet.

Beispiele

Bei der Beschreibung der Funktion AppleMenuFolder finden Sie ein passendes Beispiel.

Siehe auch

AppleMenuFolder, ApplicationSupportFolder, ControlPanelsFolder, ExtensionsFolder, FontsFolder, PreferencesFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

Destructoren können keine Parameter haben Fehler

Sie können für den Destructor einer Klasse keine Parameter definieren. Die Dialogbox für die Methoden-Deklaration erlaubt dies nicht.

Ein Destructor hat den gleichen Namen wie die Klasse, mit einem vorangestelltem '∼'-Zeichen. Er wird automatisch ausgeführt, wenn die Klasse verlassen wird.

Dictionary Klasse

Ein Objekt, das eine Liste von Schlüssel-Wert-Paaren enthält. Sowohl Schlüssel als auch Werte sind Variants.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>BinCount</i>	<u>Integer</u>	Die Anzahl der Bins, die die Hash-Tabelle verwendet. Dies ist ein Maß für die Größe der Hash-Tabelle. Normalerweise müssen Sie sich um Bins und Hash-Tabellen nicht kümmern. BinCount wäre für Sie nur hilfreich, wenn Sie sehr große Dictionarys hätten, die aus tausenden Einträgen bestünden und Sie versuchen wollten, die Geschwindigkeit zu optimieren. Wenn Sie BinCount auf einen positiven Wert setzen, wird das Dictionary diese Anzahl an Bins verwenden, egal wie viele Objekte sich im Dictionary befinden. Wenn Sie BinCount auf kleiner oder gleich Null setzen, nimmt das Dictionary einen Wert, von dem es denkt, daß es zu der Anzahl der Objekte im Dictionary paßt. Es ändert die Anzahl der Bins dynamisch, wenn Objekte dem Dictionary hinzugefügt oder aus diesem entfernt werden (oder bis Sie BinCount einen positiven Wert zuweisen). Letzteres ist das voreingestellte Verhalten.
<i>Count</i>	<u>Integer</u>	Die Anzahl der Schlüssel-Werte-Paare im Dictionary.
<i>Value</i>	<u>Variant</u>	Erwartet einen Parameter, <i>Key als Variant</i> . Weist dem Schlüssel-Eintrag im Dictionary einen Wert zu oder liefert den Wert, der zu dem Schlüssel-Objekt gehört. Wenn Sie vorhaben, einen Wert für einen Schlüssel zu lesen, den es im Dictionary nicht gibt, tritt ein KeyNotFoundException Fehler auf.

Methoden

Name	Parameter	Beschreibung
<i>Clear</i>		Löscht alle Schlüssel-Wert-Paare im Dictionary.
<i>HasKey</i>	Schlüssel als <u>Variant</u>	Liefert ein Boolean. Liefert True, falls der Schlüssel im Dictionary vorkommt und False, falls nicht.
<i>Key</i>	Index als <u>Integer</u>	Liefert ein Variant. Liefert den Wert von <i>Key</i> für das <i>Index^{te}</i> sequentielle Objekt im Dictionary. Das erste Objekt hat die Nummer Null. Wenn es kein <i>Index^{tes}</i> Objekt im Dictionary gibt, erzeugt ein Aufruf einen OutOfBoundsException-Fehler.
<i>Remove</i>	Schlüssel als <u>Variant</u>	Weist dem Schlüssel-Eintrag im Dictionary einen Wert zu oder liefert den Wert, der zu dem Schlüssel-Objekt gehört. Wenn Sie vorhaben, einen Wert für einen Schlüssel zu lesen, den es im Dictionary nicht gibt, tritt ein KeyNotFoundException Fehler auf.

Hinweise

Die Dictionary-Klasse bietet die gesamte Funktionalität der Collection-Klasse und besitzt weitere Vorteile: Bei der Collection-Klasse hängt die Zeit, um ein Objekt zu finden, von der Anzahl der Objekte in der Collection ab, da die Suche sequentiell erfolgt. Die Dictionary-Klasse verwendet eine Hash-Tabelle, bei der die Zeit relativ unabhängig von der Anzahl der Objekte ist. Sie wurde für schnelle Suche entwickelt. Außerdem ist der Schlüssel-Parameter in der Dictionary-Klasse ein Variant und in der Collection-Klasse ein String. Andererseits kann die Collection verschiedene Werte pro Schlüssel speichern. Wenn Sie einem Schlüssel einen Wert zuweisen, der bereits im Dictionary vorhanden ist, ersetzt der neue Wert den alten.

Beispiel

Die folgende For-Schleife lädt die Werte eines Dictionary in eine ListBox:

```
Dim d as Dictionary
Dim i as Integer
d=New Dictionary
d.Value(1)="ID"
d.Value(2)="Lois Lane"
d.Value(3)="Reporter"
d.Value(4)=84000
For i=1 to d.Count
    ListBox1.Addrow d.value(i)
Next
```

Folgendes Beispiel macht es sich zu Nutzen, daß der Schlüssel ein Variant ist und nicht unbedingt eine Zahl. In diesem Beispiel sind die Schlüssel Farben und die Werte sind Beschreibungen.

```
Dim d as Dictionary
d=New Dictionary
d.Value(RGB(255,0,0))="Das ist pures Rot."
d.Value(RGB(0,255,255))="Das ist pures Cyan."
MsgBox d.value(d.Key(1)) //liefert Cyan
Exception err as RuntimeException
If err IsA KeyNotFoundException then
    MsgBox "Schlüssel nicht im Dictionary"
End if
If err IsA OutOfBoundsException then
    MsgBox "Der Index des Schlüssels ist außerhalb des Wertebereichs!"
End if
```

Falls der Index, der an die Key-Methode in folgender Zeile übergeben wurde:

```
MsgBox d.value(d.Key(1)) //liefert Cyan
```

kein Element des Dictionarys ist, tritt eine OutOfBoundsException auf und der Exception Block am Ende der Methode wird aufgerufen. Sie können diesen Wert im Dictionary auch durch Übergabe der Value-Methode des Schlüssels anstelle des Index erhalten:

```
MsgBox dict.value(RGB(0,255,255))
```

In diesem Fall, wenn Sie einen Schlüssel, der sich nicht im Dictionary befindet übergeben, tritt eine KeyNotFoundException auf und der Exception Block wird aufgerufen und zeigt die passende Fehlermeldung an.

Anstelle des Exception Blocks können Sie zuerst die HasKey-Methode verwenden, bevor Sie versuchen, auf den Wert zuzugreifen:

```
If d.HasKey(RGB(0,255,255)) then
    MsgBox d.value(RGB(0,255,255))
Else
    MsgBox "Schlüssel nicht im Dictionary!"
End If
```

Siehe auch

KeyNotFoundException Fehler; Variant, Collection Klassen.

Die Größe eines Arrays muss eine Konstante oder eine Zahl sein Fehler

Wenn Sie mit dem Dim-Befehl ein Array deklarieren und die Größe festlegen, muß die Größe des Arrays entweder mit einer Zahl oder einer Konstante festgelegt werden.

Beispiel

Es wird versucht, die Größe eines Arrays mit einer Variable festzulegen:

```
Dim j as Integer
j=5
Dim a(j) as Integer
```

Siehe auch

Dim Befehl.

Die Zählervariable und die Variable nach Next müssen übereinstimmen Fehler

Wenn Sie in einer For-Schleife sowohl Zählervariable als auch eine Variable im Next-Befehl verwenden, müssen diese übereinstimmen. Die Variable im Next-Befehl kann weggelassen werden.

Beispiel

In folgender For-Schleife ist die Zählervariable i und daher muß die Variable im Next-Befehl ebenfalls i sein.

```
Dim i,j as Integer
For i=1 to 10
    j=j+1
Next j
```

Siehe auch

For Befehl.

Dies ist kein Array, aber Sie benutzen es wie eines Fehler

Sie haben ein Objekt einer Methode oder Funktion übergeben, die ein Array erwartet, aber keines erhielt. Dies tritt auch auf, wenn Sie versuchen, einem Element eines angeblichen Arrays, das aber keines ist, einen Wert zuzuweisen.

Beispiele

```
Dim aInts(4) as Integer
Dim f as FolderItem
Redim f(10) //kein Array
```

```
Dim a as String
a(1)="Ted" //kein Array
```

Siehe auch

Dim Befehl, Redim Operator.

Diese Form eines Array kann nicht sortiert werden Fehler

Sie haben versucht, ein Boolean-Array zu sortieren.

Beispiel

Sie haben versucht, ein Array von Objekten an Stelle eines Arrays von Buchstaben oder Zahlen zu sortieren. Zum Beispiel kann ein Array aus Date-Objekten nicht sortiert werden, jedoch ein Array aus den TotalSeconds-Eigenschaften eines Date-Objekt, da die TotalSeconds-Eigenschaft ein Double ist.

Dieses Beispiel-Array kann nicht sortiert werden.

```
Dim d(2) as Date
d(0)=New Date
d(1)=New Date
d(2)=New Date
d(0).year=1965
d(1).year=1970
d(2).year=1950
d.Sort
```

Die letzte Zeile des Beispiels produziert die Fehlermeldung.

```
Dim theTruth(3) as Boolean
theTruth(0)=True
theTruth(1)=True
theTruth(2)=False
theTruth(3)=False
theTruth.Sort
```

Siehe auch

Sort Methode.

Diese globale Variable hat den selben Namen wie eine globale Funktion Fehler

Sie haben den Namen einer Methode oder Funktion in einem Modul als Name einer Eigenschaft in einem Modul wiederverwendet.

Beispiel

Sie definieren 'myAlert' als eine Methode in einem Modul und definieren außerdem 'myAlert' als eine Eigenschaft in diesem Modul.

Diese globale Variable hat den selben Namen wie eine Klasse Fehler

Sie haben eine globale Variable in einem Modul erzeugt, die den gleichen Namen wie eine der Klassen in Ihrem Projekt verwendet.

Diese lokale Variable oder Konstante hat den selben Namen wie ein Declare in dieser Methode Fehler

Sie haben den Namen einer lokalen Variablen oder lokalen Konstanten in einem Declare-Befehl in einer Methode verwendet.

Siehe auch

Declare Befehl.

Diese(r) lokale Variable/Parameter hat den selben Namen wie eine Konstante Fehler

Sie haben eine lokale Variable in einer Methode oder als Parameter in einem Sub oder Function verwendet, die den gleichen Namen hat wie eine Konstante.

Beispiel

Sie können nicht den gleichen Namen für eine Konstante und eine lokale Variable verwenden:

```
Const Version="10B"  
Dim Version as String
```

Siehe auch

Const, Dim Befehle.

Diese Methode, Eigenschaft oder Variable existiert nicht Fehler

Sie haben eine Syntax verwendet, die impliziert, daß ein Ausdruck eine Methode, Eigenschaft oder Variable ist, die aber nicht existiert. Dies tritt auch auf, wenn Sie in einem Zuweisungsbefehl eine Variable verwenden, aber das Gleichheitszeichen weglassen, so daß es aussieht, als wäre es eine Methode oder Eigenschaft.

Beispiele

Das Gleichheitszeichen wurde weggelassen:

```
Dim myString as String  
myString "Columbus"
```

Es wurde eine Methode aufgerufen, die nicht definiert wurde:

```
myCustomMethod(7,5,3)  
Dim aFiles(3) as FolderItem  
aFiles.List //so etwas gibt es nicht
```

Es wurde ein Wert einer Variable zugewiesen, die nicht mit einem Dim-Befehl deklariert wurde:

```
i=10
```

Dies ist der Action-Event-Handler eines PushButtons; der Anwender möchte die Caption-Eigenschaft des PushButton anzeigen.

```
MsgBox caption //müßte me.caption heißen
```

Es wurde versucht, im Code eine Eigenschaft zu setzen, die nur im Eigenschaften-Fenster gesetzt werden kann. Folgende Code-Zeile versucht, den Wert der Bold-Eigenschaft eines MenuItems zu setzen, REALbasic erkennt die Bold-Eigenschaft jedoch nicht im Code.

```
SearchFind.Bold=True
```

Es wurde versucht, Me in einer Methode oder Funktion zu verwenden, die dem Objekt nicht hinzugefügt wurde. Folgende Methode versucht, die Caption-Eigenschaft des aufrufenden PushButton zu liefern, das ist jedoch nicht erlaubt.

```
Function myCaption() as String
    Return Me.Caption
End Function
```

Sie können auf die Eigenschaften des aufrufenden Objekts mittels Me nur innerhalb eines Event-Handlers zugreifen, der zu diesem Objekt gehört.

Es wurde versucht, einen Wert einer Konstante zuzuweisen:

```
Const SerialPort=1080
SerialPort=1040
```

Der Programmierer hatte vor, aNames als ein Array im Dim-Befehl zu deklarieren, vergaß dies aber:

```
Dim aNames as String
aNames.Append "Walter Kerr"
```

Der Versuch, einer privaten oder geschützten Eigenschaft einen Wert zuzuweisen, wie das Beispiel zeigt. myPrivateProperty ist eine private Eigenschaft des Moduls des Fensters.

```
myPrivateProperty=56
```

Diese Methode benötigt weniger Parameter als übergeben wurden Fehler

Sie haben einer Methode zu viele Parameter übergeben. Eine andere Möglichkeit ist, daß die benötigte Anzahl Parameter übergeben haben, diese aber nicht mit Kommas getrennt haben.

Beispiel

Es wurde die benötigte Anzahl Parameter übergeben, aber es wurde ein Komma vergessen:

```
ListBox1.AddRow 2 "Chicago"
```

Es wurden zu viele Parameter übergeben:

```
ListBox1.insertRow 2,"Charlie","Dallas"
```

Die selbstgeschriebene Funktion „myFunktion“ benötigt zwei Parameter:

```
Dim d as Double
d=myFunktion(5,6,8)
```

Diese Eigenschaft besitzt den gleichen Namen wie eine Klassen-Methode Fehler

Sie haben den Namen einer Methode in einer Klasse als Namen für eine Eigenschaft in der Klasse wiederverwendet.

Diese Eigenschaft hat den selben Namen wie ein Event Fehler

Sie haben den Namen eines Events als Namen für eine Eigenschaft verwendet. Sie müssen die Eigenschaft umbenennen.

Beispiel

Es wurde „Open“ sowohl als Name eines Events als auch einer Eigenschaft in einer Klasse verwendet.

Diese Eigenschaft hat den selben Namen wie eine Methode Fehler

Sie haben den Namen einer Methode als den Namen einer Eigenschaft wiederverwendet.

Diese Eigenschaft ist geschützt. Sie kann nur innerhalb ihrer Klasse verwendet werden Fehler

Sie haben versucht, auf eine Eigenschaft von außerhalb der Klasse, der sie gehört, zuzugreifen. Eine geschützt Eigenschaft kann nur innerhalb ihrer Eltern-Klasse, Fenster oder Modul verwendet werden.

Sie können eine Eigenschaft entschützen, indem Sie den Haken in der Geschützt-Checkbox in der Eigenschaften-Deklarations-Diaglogbox entfernen.

Diese Methode benötigt mehr Parameter als übergeben wurden Fehler

Sie haben nicht alle benötigten Parameter einer Methode übergeben. Eine andere Möglichkeit ist, daß Sie die benötigte Anzahl Parameter übergeben haben, diese aber nicht mit Kommas getrennt haben.

Beispiele

Es wurde die benötigte Anzahl Parameter übergeben, aber ein Komma vergessen:

```
ListBox1.InsertRow 2 "Chicago"
```

Es wurden zu wenige Parameter übergeben:

```
ListBox1.InsertRow "Chicago"
```

```
Dim c as Color  
c=RGB(100,100) //liefert "RGB benötigt 3 Parameter"
```

In den folgenden Beispielen benötigt myFunction zwei Parameter und liefert ein Double:

```
d=myFunction(5,6) //korrekt  
d=myFunction (5,6) //auch korrekt  
d=myFunction 5,6 //falsch  
d=myFunction //auch falsch
```

Diese Methode ist geschützt. Sie kann nur innerhalb ihrer Klasse verwendet werden Fehler

Sie haben versucht, auf eine geschützte Methode von außerhalb der Klasse, zu der sie gehört, zuzugreifen. Auf eine geschützte Methode kann nur von innerhalb ihres Elternobjekts zugegriffen werden – ihrer Klasse, Fenster oder Modul.

Sie können den Schutz einer Methode aufheben, indem Sie das Häkchen in der „Geschützt“-Box in der Methoden-Deklarationsbox entfernen.

Diese Methode liefert keinen Wert zurück Fehler

Sie haben in einem Ausdruck eine Methode verwendet, als wenn sie einen Wert zurückliefern würde, was sie aber nicht macht.

Beispiel

die selbstgeschriebene Methode myColor ist keine Funktion, daher ist die Syntax im Aufruf dieser Methode nicht korrekt:

```
Dim c as Color  
c=myColor //liefert kein Color
```

Diese Zeile ist zu komplex Fehler

Sie haben eine Code-Zeile geschrieben, die REALbasic nicht verstehen kann. Sie müssen die Zeile vereinfachen.

Dieses #else hat kein vorhergehendes passendes #if Fehler

Ein zusätzliches `#else`-Schlüsselwort wurde in einem `#If`-Befehl verwendet (bedingte Compilierung) oder ein `#else`-Befehl wurde verwendet, wo kein `#if` voranging.

Beispiel

Ein `#else`-Schlüsselwort wurde an Stelle eines `Else`-Schlüsselworts in einem `If`-Befehl verwendet:

```
Dim theNumber As Integer
Dim digits As Integer
theNumber=33
If theNumber<10 Then
    digits=1
#else //müßte else sein
    digits=3
End If
```

Siehe auch

`#If`, `If` Befehle.

Dieses #endif hat kein vorhergehendes passendes #if Fehler

Ein `#If`-Befehl wurde in einer `#if...#endif` Struktur weggelassen.

Beispiel

Der `If`-Befehl in diesem Beispiel müßte `#If` sein.

```
If TargetMacOS then
    Beep
#endif
```

Siehe auch

`#If` Befehl.

Dieses Array hat mehr Dimensionen als Sie angegeben haben Fehler

Beim Verweisen auf ein Element in einem mehrdimensionalen Array haben Sie nicht für alle Dimensionen Indizes mitgeliefert.

Beispiel

```
Dim a(5,4) as String
a(1)="Peter"
```

Siehe auch

`Dim` Befehl.

Dieses Array hat weniger Dimensionen als Sie angegeben haben Fehler

Sie haben versucht, eine eingebaute Array-Methode, die ein eindimensionales Array benötigt, auf ein mehrdimensionales Array anzuwenden.

Beispiel

Sie können ein zweidimensionales Array nicht sortieren:

```
Dim aInts(3,3) as Integer
Dim i,j as Integer
For i=0 to 2
    For j=0 to 2
        aInts(i,j)=i*j
    Next
Next
aInts.Sort //funktioniert nicht mit zweidimensionalen Arrays
```

Sie Verwenden die Syntax, die darauf hinweist, daß Sie auf ein Element in einem zweidimensionalen Array verweisen, wenn dieses jedoch als eindimensionales Array deklariert wurde:

```
Dim aInts(3) as Integer  
aInts(0,0)=3 //falsch
```

Sie haben versucht, ein eindimensionales Array mit dem Redim-Befehl in ein zweidimensionales zu ändern:

```
Dim aInts(3) as Integer  
  
Redim aInts(5,5)
```

Siehe auch

Append, Insert, Remove, Sort, Dim, Redim Methoden.

Dieser Klasse fehlt eine oder mehrere Methoden eines Interfaces, das sie erbt Fehler

Eine oder mehrere Methoden dieses Interfaces konnte in dieser Klasse nicht gefunden werden, obwohl sie das Interface implementieren müßte.

Wenn Sie die Eigenschaften einer Klasse im Eigenschaften-Fenster bearbeiten, können Sie auch ein Klassen-Interface für diese Klasse angeben. Wenn Sie dies tun, muß die Klasse die entsprechenden Methoden des Interfaces selbst implementieren. Diese Methoden müssen von der Deklaration, sprich Parameter-Anzahl und -Typen, mit denen im Interface übereinstimmen. Wenn Sie eine oder mehrere Methoden vergessen, erhalten Sie diesen Fehler. Wenn Sie die Applikation in der IDE ausführen, erhalten Sie ein Fenster, das Ihnen alle Methoden zeigt, die die Klasse implementiert haben muß.

Dieser Name wird bereits verwendet Fehler

Sie haben einen Variablennamen in einem Dim-Befehl verwendet, der schon in einem anderen Dim-Befehl oder als eine Eigenschaft definiert wurde. Oder, Sie haben versucht, eine Eigenschaft mit dem gleichen Namen wie eine andere Eigenschaft zu deklarieren.

Beispiel

```
Dim a,b as Double  
Dim b as Double //wird bereits verwendet
```

Siehe auch

Dim Befehl.

Dieses Objekt steht im Konflikt zu einem anderen Objekt gleichen Namens Fehler

Tritt auf, wenn Sie eine überladene Methode aufrufen und REALbasic nicht entscheiden kann, welche Version sie aufrufen wollen. Das passiert, wenn zwei Methoden den gleichen Namen haben, die gleiche Anzahl Parameter und die Datentypen der Parameter übereinstimmen.

Dieser Fehler tritt auch auf, wenn ein Methodennamen mit einem anderen Projektobjekt wie einem Fenster, Menüeintrag, Steuerelement usw. in Konflikt gerät. Wenn Sie z.B. eine globale Funktion namens Bla und ein Fenster namens Bla haben.

Beispiel

Die Methode myOverLoadedMethod erwartet drei Integer-Parameter, wurde aber im Code-Editor eines Fensters zweimal definiert (um unterschiedliche Funktionen auszuführen) Ein Aufruf von myOverLoadedMethod produziert diesen Fehler. Die Lösung ist, die zweite Instanz der myOverLoadedMethod zu löschen oder umzubenennen.

Instanz 1

```
Sub myOverLoadedMethod(a as Integer, b as Integer, c as Integer)  
Dim d as Integer  
d=a*b/c  
MsgBox str(d)  
End Sub
```

Instanz 2

```
Sub myOverLoadedMethod(x as Integer, y as Integer, zas Integer)
    Dim d as Integer
    d=x/z*y
    MsgBox str(d)
End Sub
```

Aufruf der Methode

```
Dim a,b,c as Integer
a=5
b=10
c=15
myOverloadedMethod(a,b,c)
```

Dim Befehl

Legt eine lokale Variable anhand der Angaben von Datentyp, Name und Größe (falls es sich um ein Array handelt) an.

Syntax

Der Befehl **Dim** kennt zwei Syntax-Formen:

Dim *variableName* [,*variableNameN*] **As** *dataTyp*[= *InitialValue*]

Teil	Art	Beschreibung
<i>variableName</i>	Variablenname	Der Name der neuen Variable.
<i>variableNameN</i>	Variablenname	Optional. Die Namen anderer Variablen, die Sie mit dem gleichen Datentyp anlegen möchten. Die Namen werden durch Kommata getrennt.
<i>dataTyp</i>	Datentyp als Name	Der Datentyp der Variable.

Dim *arrayName*(*size* [,*sizeN*]) **as** *dataTyp*

Teil	Art	Beschreibung
<i>arrayName</i>	Variablenname	Der Name des neuen Arrays.
<i>size</i>	<u>Integer</u>	Die Größe (Anzahl der Elemente) bei einem Array.
<i>sizeN</i>	<u>Integer</u>	Optional. Die Größe der nächsthöheren Dimension eines Arrays, falls Sie ein mehrdimensionales Array anlegen möchten.
<i>dataTyp</i>	Datentyp als Name	Der Datentyp des Arrays.

Hinweis

Bei der ersten Syntax-Form wird der **Dim**-Befehl verwendet, um neue Variablen anzulegen. Eine Variable ist ein Objekt, das im Arbeitsspeicher Werte aufnehmen kann. Bei der zweiten Syntax-Form wird der **Dim**-Befehl verwendet, um ein neues Array der angegebenen Größe anzulegen. Ein Array ist einfach eine Variable, die mehrere Werte des gleichen Datentyps aufnehmen kann. Geben Sie mehr als eine Größe an, so wird ein mehrdimensionales Array angelegt.

Die maximale Dimension in REALbasic ist 20.

Sie können optional einen initialen Wert angeben, wenn Sie eine Variable mit einem Dim-Ausdruck deklarieren. Ist die Variable von einem der folgenden Datentypen: String, Integer, Single, Double, Boolean oder Color, so können Sie, entsprechend dem folgenden Beispiel, initiale Werte zuweisen:

```
Dim a as Integer =5
Dim b as Double =87.87
Dim s as String="Howdy"
Dim c as Color = &cFF4B51
```

Das folgende Beispiel initialisiert drei Variablen mit einem Wert:

```
Dim x, y, z As Integer = 10
```

Ist die Variable vom Typ Objekt, können Sie den New-Operator mit dem Dim-Ausdruck verbinden. In diesem Beispiel wird ein Date-Objekt deklariert und instanziiert.

```
Dim d as New Date
MsgBox d.shortdate
```

Ansonsten benötigen Sie eine weitere Zeile für die Instanzierung, wie das folgende Beispiel zeigt:

```
Dim d as Date
d=New Date
MsgBox d.shortdate
```

Versteht der Konstruktor des Objekts Parameter, können Sie diese direkt mitangeben:

```
Dim mb as New MemoryBlock(512)
```

Der **Dim**-Befehl kann sich nicht innerhalb einer bedingten Struktur (ein **If**- oder **Case**-Befehl) oder einer Schleifen-Struktur wie **While** oder **For** befinden. Normalerweise kann er sich aber hinter anderen Zeilen mit ausführbarem Code befinden.

Mit dem **Dim**-Befehl in einem Modul angelegte Variablen haben einen lokalen Einsatzbereich. Dies bedeutet, daß sie bei jedem Aufruf der Methode neu angelegt und beim Beenden der Methode entfernt werden. Der Wert einer lokalen Variable kann nur in der gleichen Methode verwendet werden.

Eigenschaften, die innerhalb eines Moduls erzeugt wurden, haben jedoch einen globalen Einsatzbereich und auf sie kann innerhalb der gesamten Applikation zugegriffen werden.

Wenn Sie die Größe des Arrays zu dem Zeitpunkt, zu dem Sie es erzeugen, nicht kennen, können Sie es als ein Null-Array deklarieren, das ist ein Array ohne Elemente und später dann zum Vergrößern den Redim-Befehl verwenden. Sie machen das, indem Sie einen Index von -1 übergeben. Das bedeutet „ein Array ohne Elemente“. Zum Beispiel erzeugt der Befehl:

```
Dim aNames(-1) as String
```

das Array namens **nNames** ohne Elemente. Wenn Ihr Programm eine Liste von Namen speichern muß, die der Anwender eingibt, können Sie mit dem Festlegen der Größe so lange warten, bis Sie wissen, wie viele Namen der Anwender eingegeben hat. Sie erreichen das Gleiche, indem Sie das **-1** weglassen. Folgender Ausdruck ist äquivalent:

```
Dim aNames() as String
```

Wenn Sie einem Array Werte zuweisen, wie zum Beispiel mit der Split-Funktion, müssen Sie die Anzahl der Elemente nicht im Voraus kennen.

Sie können einem Array auch ein Array zuweisen, wie das folgende Beispiel zeigt:

```
Dim myArray() as String
Dim newArray() as String
myArray(0)="Anthony"
myArray(1)="Aardvark"
myArray(2)="Accountant"
newArray()=myArray //newArray ist nun das gleiche wie myArray
```

Beispiele

Dieses Beispiel benutzt den Befehl **Dim**, um eine Integer-Variable mit dem Namen „Alter“ anzulegen und ihr den Wert 33 zuzuweisen.

```
Dim Alter As Integer
Alter=33
```

Dieses Beispiel des **Dim**-Befehls legt zwei String-Variablen an:

```
Dim Vorname, Nachname As String
```

Dieses Beispiel benutzt den Befehl **Dim**, um ein Array „Namen“ mit 11 Elementen anzulegen (denken Sie daran, daß Arrays immer ein „nulltes“ Element haben).

```
Dim Namen(10) As String
```

Dieses Beispiel benutzt den Befehl **Dim**, um ein Array „Namen“ mit nur einem Element anzulegen.

```
Dim Namen(0) As String
```

Dieses Beispiel benutzt den Befehl **Dim**, um ein zweidimensionales Array mit dem Namen „Namen“ anzulegen, das 11 Zeilen und 4 Spalten aufnehmen kann.

```
Dim Namen(10,3) As String
```

Siehe auch

Append , Insert , Redim , Remove , Sort Methoden; UBound Funktion; Collection Klasse.

DisclosureTriangle Steuerelement



Wird verwendet, um einem Fenster ein DisclosureTriangle-Element (Aufklappdreieck) hinzuzufügen. Ein DisclosureTriangle kann im zugeklappten Zustand entweder nach rechts oder nach links zeigen. Wenn der Anwender auf das DisclosureTriangle-Steuerelement klickt, wird der Wert von **Value** automatisch umgeschaltet.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Typ	Beschreibung
<i>Facing</i>	<u>Integer</u>	0 - Spitze nach rechts 1 - Spitze nach links
<i>Value</i>	<u>Boolean</u>	<u>True</u> bedeutet aufgeklappt (Spitze nach unten); <u>False</u> bedeutet eingeklappt, also Spitze nach rechts oder links, je nach <i>Facing</i> .

Siehe auch

RectControl Klasse.

DocumentsFolder Funktion

Wird benutzt, um Zugriff auf den Dokumente-Ordner des Anwenders zu bekommen. Unter Windows XP liefert ein Aufruf von DocumentsFolder den Ordner „Eigene Dateien“ im Verzeichnis des aktuellen Anwenders, z.B. „C:\Dokumente und Einstellungen\Anwendername\Eigene Dateien“.

Syntax

Ergebnis=**DocumentsFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein <u>FolderItem</u> , das den Dokumente-Ordner repräsentiert.

Hinweis

Verwenden Sie die Funktion **DocumentsFolder**, um auf den Dokumente-Ordner des derzeitigen Anwenders zuzugreifen. Die Funktion **DocumentsFolder** bietet eine Möglichkeit, auf die Einträge des Dokumente-Ordners zuzugreifen, die auch unter anderen Sprachversionen des Betriebssystems und sowohl auf dem Macintosh als auch unter Windows funktioniert. Unter Linux wird versucht, das Verzeichnis „Documents“ im Home-Verzeichnis zu lesen. Wenn dieses Verzeichnis nicht existiert, wird ein FolderItem des Home-Verzeichnisses zurückgeliefert.

Beispiele

Bei der Beschreibung der Funktion AppleMenuFolder finden Sie ein passendes Beispiel.

Siehe auch

AppleMenuFolder, ApplicationSupportFolder, ControlPanelsFolder, ExtensionsFolder, DesktopFolder, FontsFolder, PreferencesFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

Do...Loop Befehl

Wiederholt eine Reihe von Anweisungen, bis eine Bedingung erfüllt ist.

Syntax

Do [*Until Bedingung*]

[Anweisungen]

[Exit]

[Anweisungen]

Loop [Until *Bedingung*]

Teil	Beschreibung
<i>Do</i> [Until]	Startet die Schleife. Wenn die optionalen Parameter <i>Until</i> und <i>Bedingung</i> angegeben sind, wird <i>Bedingung</i> benutzt, um das Ende der Schleife festzulegen.
<i>Bedingung</i>	Ein gültiger <u>boole'scher</u> Ausdruck. Nimmt dieser den Wert <i>True</i> an, so wird die Schleife beendet.
<i>Exit</i>	Optional. Beendet die Schleife vorzeitig.
<i>Anweisungen</i>	Befehle, die wiederholt werden (solange <i>Bedingung</i> <u>False</u> ist).
<i>Loop</i> [Until]	Schließt die Schleife. Falls die optionalen Parameter <i>Until</i> und <i>Bedingung</i> angegeben sind, wird <i>Bedingung</i> geprüft.

Hinweis

Der Befehl **Do...Loop** führt die Befehle *Anweisungen* solange aus, bis *Bedingung* den Wert *True* annimmt oder der Exit Befehl ausgeführt wird.

Sollen *Anweisungen* nur ausgeführt werden, wenn *Bedingung* bereits den Wert *True* hat, so sollten Sie am Anfang der Schleife *Bedingung* überprüfen. Sollen *Anweisungen* mindestens einmal ausgeführt werden, so prüfen Sie *Bedingung* einfach am Ende der Schleife. Wenn *Bedingung* nicht verwendet werden soll, um die Schleife zu beenden, so müssen Sie den Exit Befehl innerhalb von **Do...Loop** aufrufen, da Ihre Schleife sonst nie beendet werden kann.

Do...Loop-Befehle können beliebig ineinander verschachtelt werden. Zu jedem **Loop**-Befehl gehört ein vorausgehendes **Do**. Wenn die Loop-Schleife läuft, übernimmt sie die Benutzerschnittstelle, um den Anwender daran zu hindern, mit Menüs oder Steuerelementen zu arbeiten. Normalerweise macht das nichts aus. Wenn jedoch die Schleife sehr lange läuft, sollten Sie deren Code in einen separaten Thread verschieben, damit sie im Hintergrund läuft.

Beispiel

Dieses Beispiel verwendet **Do...Loop**, um den Wert einer Variable zu erhöhen. Überschreitet dieser Wert bereits beim Start der Schleife den Wert 100, so wird der Wert der Variable nicht erhöht.

```
Do Until x<100
    x=x+1
Loop
```

Im folgenden Beispiel wird die Schleife mindestens einmal ausgeführt, da *Bedingung* erst am Ende der Schleife überprüft wird.

```
Do
    x=x+1
Loop Until x<100
```

Siehe auch

For...Next , While...Wend Befehle; Thread Klasse.

Double Datentyp

Ein **Double** ist ein wesentlicher Datentyp in REALbasic. Ein **Double** ist eine Zahl, die einen Dezimalwert besitzen kann, es ist eine reelle Zahl. Sie kann Werte zwischen 2.2250738585072013 e-308 und 1.7976931348623157 e+308 annehmen. In anderen Sprachen würde das **Double** von REALbasic als Realzahl mit doppelter Präzision bezeichnet. Da **Double** Zahlen sind, können Sie mathematische Berechnungen mit diesen durchführen. **Double** belegen 8 Byte Speicher.

Die VarType-Funktion liefert als Wert 5, wenn ein **Double** übergeben wird.

Beispiel

```
Dim d as Double
d=3.14159265358979323846264338327950
```

Siehe auch

Integer, Single, String, Boolean, Color Datentypen; +, -, *, /, <, <=, =, >=, >, <>, \, Mod, Val, Str, VarType, IsNumeric Funktionen; Dim Ausdruck.

DragItem Klasse

Wird verwendet, um Daten zu übertragen, die der Anwender verschoben hat.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Destination</i>	<u>Object</u>	Der Ort des fallengelassenen Objekts, wenn es auf den Schreibtisch oder auf einer Applikation fallengelassen wird. Da <i>Destination</i> vom Typ Objekt ist, ist es in der Lage, jeden Typ eines REALbasic-Objektes anzunehmen. Derzeit funktioniert <i>Destination</i> nur im Macintosh-Finder und liefert ausschließlich ein <u>FolderItem</u> . Es wird entweder Mac OS 8-9 oder Mac OS X 10.2 oder neuer benötigt. Um zu gewährleisten, daß Ihre Applikation auch mit künftigen Versionen von REALbasic kompatibel ist, sollten Sie den gelieferten Objekttyp mittels <u>IsA</u> überprüfen, bevor Sie ihn einem <u>FolderItem</u> zuweisen. Unter Windows wird das Draggen von Objekten auf den Schreibtisch nicht unterstützt; daher funktioniert diese Eigenschaft nicht unter dieser Bedingung.
<i>DropHeight</i>	<u>Integer</u>	Höhe des Objektes, das gedroppt wurde.
<i>DropLeft</i>	<u>Integer</u>	Abstand von der linken Seite des Steuerelements, wo das Objekt gedroppt wurde.
<i>DropTop</i>	<u>Integer</u>	Abstand von der oberen Rand des Steuerelements, wo das Objekt gedroppt wurde.
<i>DropWidth</i>	<u>Integer</u>	Breite des Objektes, das gedroppt wurde.
<i>FolderItem</i>	<u>FolderItem</u>	Das verschobene <u>FolderItem</u> .
<i>FolderItemAvailable</i>	<u>Boolean</u>	Die FolderItem-Eigenschaft enthält ein Ordnerobjekt.
<i>MacData</i>	<u>String</u>	Der Dateityp der Daten, die gedraggt wurden. Type ist ein vierbuchstabiger Resource-Typ oder vom Programmierer definierter vierbuchstabiger Code. Wird nur auf dem Macintosh unterstützt und innerhalb der REALbasic-Applikation unter Windows.
<i>MacDataAvailable</i>	<u>Boolean</u>	True, wenn das DragItem Daten des angegebenen Typs enthält. Der Typ wird als String an MacData übergeben. Type ist ein vierbuchstabiger Resource-Typ oder vom Programmierer definierter vierbuchstabiger Code. Der Typ wird an MacData oder PrivateMacData als ein String übergeben. Wird nur auf dem Macintosh unterstützt.
<i>MouseCursor</i>	<u>MouseCursor</u>	Die Form des Mauszeigers, die dieser während des Verschiebens hat.
<i>Picture</i>	<u>Picture</u>	Das zu verschiebende Bild.
<i>PictureAvailable</i>	<u>Boolean</u>	Die Picture-Eigenschaft enthält ein Bild.
<i>PrivateMacData</i>	<u>String</u>	Daten (des angegebenen Typs), die verschoben werden. Type ist ein vierbuchstabiger Resource-Typ oder vom Programmierer definierter vierbuchstabiger Code. Diese Daten können nicht an eine weitere Anwendung verschoben werden. Wird nur auf dem Macintosh unterstützt.
<i>Text</i>	<u>Text</u>	Der verschobene Text.
<i>TextAvailable</i>	<u>Boolean</u>	Die Text-Eigenschaft enthält Text.

Methoden

Name	Parameter	Beschreibung
<i>AddItem</i>	Left als <u>Integer</u> , Top als <u>Integer</u> , Breite als <u>Integer</u> , Height als <u>Integer</u>	Fügt dem verschobenen Objekt ein weiteres Objekt hinzu. Die Parameter geben den während des Verschiebens angezeigten Rahmen an.
<i>Drag</i>		Gestattet das Verschieben.
<i>NextItem</i>		Gibt <u>True</u> zurück, wenn ein weiteres Objekt verschoben wurde und weist den DragItem-Eigenschaften dessen Werte zu.

Hinweis

Wenn der Anwender Daten aus einem Fenster oder Steuerelement verschieben möchte, muß ein **DragItem**-Objekt erzeugt werden, das die Daten enthält, die dem Objekt übermittelt werden, auf das der Anwender Daten verschiebt. REALbasic bietet Ihnen echtes Macintosh Drag-und-Drop. Dies bedeutet, daß es Ihnen die Möglichkeit gibt, daß der Empfänger der Daten alles sein kann, was Daten der Art annehmen kann, die den verschobenen entsprechen. Text kann beispielsweise auf den Schreibtisch gezogen werden, um dort einen Text-Clip zu erzeugen. Ebenso kann Text an eine beliebige Anwendung übergeben werden, die (wie z.B. SimpleText oder der Notizblock) verschobene Texte annehmen. Bilddaten können auf den Schreibtisch gezogen werden, um dort Grafik-Clip-Dateien zu schreiben. Andere Datenarten können mit Hilfe der MacData Eigenschaft transportiert werden.

Standardmäßig kann ein **DragItem**-Objekt nur eine Datenzeile je Datentyp enthalten. Mit Hilfe der *AddItem* Methode können Sie dem **DragItem**-Objekt weitere Zeilen hinzufügen, so daß auch mehrere Einträge des gleichen Typs enthalten sein können. Wenn Sie z.B. möchten, daß ein zu verschiebendes Objekt zwei verschiedene Bilder beinhaltet, weisen Sie der Bildeigenschaft ein Bild zu, während Sie danach für das andere Bild mit Hilfe der *AddItem*-Methode eine weitere Datenzeile hinzufügen und dann der Bildeigenschaft das zweite Bild zuweisen.

MacData und *PrivateMacData* erlauben Ihnen die Unterstützung von Drag und Drop von fremden Datenformaten. Sie legen das Datenformat durch Verwendung eines vier-buchstabigen Types entsprechend einem Macintosh-Resource-Type fest. *MacData* und *PrivateMacData* liefern das gedraggte Objekt als ein String zurück. Das Steuerelement oder das Fenster, das das Objekt erhält, ist selbst dafür zuständig, was es damit zu machen hat. Das bedeutet normalerweise, daß Sie Toolbox-Aufrufe oder ein Plugin verwenden müssen, das für den Umgang mit diesem Datenformat geschaffen ist.

Sie können ebenso *MacData* oder *PrivateMacData* verwenden, um internes Drag&Drop zu steuern. Wenn Sie einen vier-buchstabigen Type (keinen schon vorhandenen Macintosh Resource-Type) erfinden, können Sie für ein Steuerelement verhindern, daß Text von fremden Quellen darauf gedraggt werden kann. Siehe das Beispiel mit dem Draggen von einer ListBox in einer andere.

Für EditFields wird Drag-und-Drop nur unterstützt, wenn deren Eigenschaft „Multiline“ True (angekreuzt) ist. Für EditFields müssen Sie keine DragItems anlegen, REALbasic erledigt diese Aufgabe für Sie selbständig.

ListBoxen erlauben das Verschieben von Zeilen, sobald die EnableDrag-Eigenschaft der ListBox True (angekreuzt) ist. Wie bei EditFields erzeugt REALbasic das DragItem für die ListBox automatisch, aber Sie müssen dieses DragItem im Event-Handler „DragRow“ vorbelegen. Siehe „ListBox Steuerelement“ für weitere Informationen über Auswahllisten.

Die Eigenschaften DropWidth und DropHeight haben unter Win32 keine Bedeutung und liefern 0 zurück, außer sie werden mittels NewDragItem gesetzt.

Beispiele

Dieses Beispiel zeigt, wie ein Bild aus einem Canvas verschoben werden kann. Das Beispiel verwendet das Me-Schlüsselwort, um auf das Canvas zu verweisen. Als Besonderheit verwendet das Beispiel einen benutzerdefinierten Cursor für das Draggen. Das Beispiel geht davon aus, daß eine Resource-Datei namens Resources, die eine CURS-Resource mit ID 1044 enthält, in das Projekt-Fenster gedraggt wurde.

```
Function MouseDown(X As Integer, Y As Integer) As Boolean
    Dim d as DragItem
    d=NewDragItem(Me.left,Me.top,Me.width,Me.height)
    d.picture=Me.Backdrop
    d.MouseCursor=app.ResourceFork.getcursor(1044)
    d.Drag //Erlaubt das Draggen
End Function
```

Siehe auch das Beispiel für ImageWell, welches Drag&Drop zwischen zwei ImageWells und PICT-Dateien zeigt oder für Clips auf dem Schreibtisch und das ListBox-Steuerelement, welches ein Beispiel für Drag&Drop zwischen zwei ListBoxen zeigt.

Folgendes Beispiel implementiert ein „Anfasser“-Interface, in welchem ein Benutzer Zeilen aus der ListBox1 in die ListBox2 (und umgekehrt) draggen kann, jedoch keine der beiden ListBoxen akzeptiert Text von anderen Steuerelementen oder von Programmen außerhalb der REALbasic-Anwendung. Es verwendet PrivateMacData, damit verhindert wird, daß Text von außerhalb in das Programm gedraggt werden kann.

Um Drag&Drop in beide Richtungen zu erlauben, sind beide ListBoxen auf gleiche Weise aufgebaut. Die EnableDrag-Eigenschaft bei-der ListBoxen ist auf True gesetzt. Das DragRow-Event sieht folgendermaßen aus:

```
Function DragRow (drag as DragItem, row as Integer) As Boolean
    Drag.PrivateMacData("hehe")=Me.List(Row)+Chr(13) //Hole den Text
    Me.RemoveRow(Row)
    Return True //Erlaube Dragging
End function
```

Um zu erreichen, daß jede Listbox den gedraggten Text akzeptiert, muß folgende Anweisung in deren Open-Event eingefügt werden:

```
Me.AcceptMacDataDrop("hehe")
```

Der DropObject-Event-Handler sieht folgendermaßen aus:

```
Sub DropObject (obj as DragItem)
    If obj.MacDataAvailable("hehe") then
        Me.AddRow(obj.MacData("hehe"))
    end if
End Sub
```

Folgendes Beispiel erlaubt dem Anwender, mehrere Text-Dateien vom Schreibtisch auf ein EditField zu ziehen. Das Beispiel fügt den Inhalt aller Textdateien in das EditField, indem es den Inhalt jeder einzelnen Datei an die Text-Eigenschaft des EditFields hängt.

Um das Draggen mehrerer Dateien zu unterstützen, wird die Funktion NextItem verwendet, um festzustellen, ob es weitere Dateien gibt, die abgelegt werden müssen.

```
Dim textStream as TextInputStream
If obj.folderItemAvailable then
    Do
        textStream=obj.FolderItem.Openastextfile
        Me.text=Me.text+textstream.readall+Chr(13)
    loop until not obj.NextItem
End if
```

Das EditField besitzt folgende Code-Zeile

```
Me.AcceptFileDrop("text")
```

in ihrem Open-Eventhandler, wobei "text" im Dateitypen-Dialog als Dateien vom Macintosh-Type TEXT definiert sind.

Dynamische Steuerelemente können nur als Teil eines Steuerelement-Arrays erzeugt werden Fehler

Sie haben versucht, zur Laufzeit aus einem vorhandenen Steuerelement ein neues zu erzeugen.

Beispiel

Ein PushButton Steuerelement CancelButton existiert bereits in dem Fenster.

```
Dim p as CancelButton
p=New CancelButton
```

Siehe auch

New Operator; Beschreibung, wie man ein Array aus Steuerelementen erzeugt (im Entwicklerhandbuch).

EasyTCPSocket Klasse

Kommuniziert mit anderen REALbasic Applikationen über das TCP-Protokoll. Unerfahrene Benutzer können mit dieser Klasse einfacher über Netzwerk kommunizieren als mit der TCPsocket-Klasse.

Übergeordnete Klasse

TCPsocket

Events

Name	Parameter	Beschreibung
Error	Code als <u>Integer</u>	Ein Fehler ist aufgetreten. Der Code-Parameter enthält einen Fehlercode, entsprechend der Fehler-Konstanten der <u>SocketCore</u> -Klasse.
ReceivedMessage	Command als <u>Integer</u> , Data als <u>String</u>	Eine empfangene Nachricht, die eine Befehls-ID und einen dazugehörigen Text enthält.

Das DataAvailable-Event der TCPSocket ist nicht verfügbar.

Methoden

Name	Parameter	Beschreibung
SendMessage	Command als <u>Integer</u> , Data als <u>String</u>	Sendet eine Nachricht, die eine Befehls-ID und einen zugehörigen Text enthält.
WaitForConnection	Timeout als <u>Integer</u>	Wartet auf eine Verbindung, bis Timeout erreicht ist. Liefert <u>True</u> wenn eine Verbindung aufgebaut wurde, andernfalls <u>False</u> .
WaitForMessage	Command als <u>Integer</u>	Wartet auf eine Nachricht mit der angegebenen Befehls-ID und liefert den zugehörigen Text.

Obwohl Sie Zugriff auf die Read-, Write- und ReadAll-Methoden der TCPSocket-Klasse haben, sollten Sie diese nicht aufrufen. Tun Sie dies, wird eine RuntimeException mit entsprechender Fehlermeldungen ausgelöst.

Klassen Konstanten

Diese Konstanten geben an, welcher Fehler aufgetreten ist. Sie entsprechen den Fehler-Konstanten der SocketCore-Klasse. Wenn ein Fehler auftritt, prüfen Sie den Parameter „Code“ gegen diese Konstanten, um den Fehler zu ermitteln.

Fehler-Code	Klassenkonstante
100	OpenDriverError
102	LostConnection
103	NameResolutionError
105	AddressInUseError
106	InvalidStateError
107	InvalidPortError
108	OutOfMemoryError

Hinweise

Der Command-Parameter kann verwendet werden, um die Daten zu identifizieren, die geschickt wurden. Zum Beispiel können Sie einen Befehl mit der ID 100 senden, um festzulegen, daß diese Daten einem FolderItem entsprechen. Oder Sie definieren die ID 101, um die Daten als Benutzernamen zu klassifizieren. Die Festlegung der IDs und deren Sinn ist Ihnen überlassen.

Befehl-IDs unterhalb Null sind für die interne Benutzung reserviert. Wenn Sie Nachrichten verschicken, sollten Sie keine Befehl-IDs unterhalb Null verwenden, da diese mit anderen Klassen interagieren könnten, was eventuell nicht beabsichtigt ist.

Die **EasyTCPSocket**-Klasse erlaubt Ihnen Verbindungen zu erstellen und über das TCP-Protokoll mit anderen Computern zu kommunizieren. Der Hauptunterschied zwischen der **EasyTCPSocket**- und der TCPSocket-Klasse besteht darin, daß die **EasyTCPSocket**-Klasse über Nachrichten kommuniziert. Bei beiden Klassen müssen Sie, wenn Sie Daten an einen entfernten Computer schicken wollen, die SendMessage-Methode verwenden.

Wenn Sie Daten empfangen, erhalten Sie kein DataAvailable-Event, wie es bei der TCPSocket-Klasse der Fall ist. Stattdessen erhalten Sie die vollständige Nachricht über das ReceivedMessage-Event. Deshalb sollten Sie nicht die grundlegenden Methoden Read oder ReadAll der TCPSocket-Klasse verwenden. Wenn Sie eine Nachricht empfangen, die über ein EasyTCPSocket verschickt wurde, können Sie diese im ReceivedMessage-Event verarbeiten.

Für Synchronisationszwecke gibt es eine WaitForConnection-Methode, mit der Sie für eine bestimmte Zeit auf eine Verbindung warten können. Sie liefert True, wenn eine Verbindung zustande kam, andernfalls False.

REALbasic ruft intern die DoEvents-Methode der Application-Klasse auf, damit das GUI weiterhin bedienbar bleibt und nicht blockiert. Für synchrone Kommunikation können Sie die WaitForMessage-Methode verwenden. Diese Methode wartet auf eine Nachricht, die eine bestimmte Befehls-ID enthält. Wird eine solche Nachricht empfangen, liefert die Methode den zugehörigen Text zurück. Wird eine Nachricht empfangen, die nicht die erforderliche Befehls-ID enthält, so wird diese verworfen. Auch diese Methode ruft intern die DoEvents-Methode auf, damit das GUI bedienbar bleibt.

Die **EasyTCPSocket**-Klasse ist nur für einfache Netzwerkcommunication zwischen REALbasic-Applikationen entwickelt. Es ist nicht für die Entwicklung von eigenen Protokollen vorgesehen. Es kann auch nur mit Applikationen kommuniziert werden, die ebenfalls die **EasyTCPSocket**-Klasse verwenden.

Beispiele

Verbinden mit einem Netzwerk. Diese Beispiel verwendet ein EasyTCPSocket namens „Connector“.

```
Connector.Address = "localhost"
Connector.Port = 1270
Connector.Connect
```

Lauschen auf eine Nachricht. Hier wird ein EasyTCPSocket namens „Listener“ verwendet.

```
Listener.Port = 1270
Listener.Listen
```

Eine Nachricht versenden. Das verwendete EasyTCPSocket heißt „Connector“ und im EditField2 ist die Befehls-ID, im EditField3 der zugehörige Text, enthalten.

```
Connector.SendMessage(Val(EditField2.Text), EditField3.Text)
```

Siehe auch

TCPSocket, EasyUDPSocket, AutoDiscovery Klassen.

EasyUDPSocket Klasse

Bietet die Kommunikation über das UDP-Protokoll in den Modi „unicasting“, „multicasting“ und „broadcasting“. Diese Klasse bietet eine einfache Abstraktion der UDPSocket-Klasse, die es unerfahrenen Benutzern einfacher macht, über UDP zu kommunizieren.

Übergeordnete Klasse

UDPSocket

Events

Name	Parameter	Beschreibung
Error	Code als <u>Integer</u>	Ein Fehler ist aufgetreten. Dieses Event ersetzt das Error-Event der <u>TCP</u> - <u>Socket</u> . Es funktioniert genauso, aber dieses Event übergibt den Fehler-Code. Damit müssen Sie nicht die LastErrorCode-Eigenschaft überprüfen.
ReceivedMessage	FromIP als <u>String</u> Command als <u>Integer</u> Data als <u>String</u>	Eine vollständige Nachricht, die eine Befehls-ID mit zugehörigem Text enthält und die vom Computer mit der IP-Adresse FromIP verschickt wurde.

Das DataAvailable-Event der UDPSocket-Klasse ist nicht verfügbar.

Methoden

Name	Parameter	Beschreibung
Bind	Port als <u>Integer</u>	Bindet den Socket an den angegebenen Port.

Name	Parameter	Beschreibung
Register	GroupName als <u>String</u>	Registriert <i>GroupName</i> als ein Erkennungsmerkmal für Nachrichten, die mit <code>SendMessageToGroup</code> verschickt werden. <i>GroupName</i> kann ein beliebiger String sein. REALbasic wird diesen String in eine gültige Klasse D-IP-Adresse umwandeln (224.0.0.0 - 239.255.255.255). Wenn Sie zum Beispiel Register („Mein cooles Programm“) aufrufen, wird REALbasic eine Klasse D-IP-Adresse hierfür speichern. Wenn Sie allerdings eine gültige Klasse D-IP-Adresse angeben, wird diese ohne Modifikation verwendet.
SendMessageToGroup	GroupName als <u>String</u> , Command als <u>Integer</u> , Data als <u>String</u>	Sendet eine Nachricht mit einer Befehls-ID Command und einem Text Data and die angegebene Gruppe <i>GroupName</i> .
SendMessageToIndividual	IPAddress als <u>String</u> , Command als <u>Integer</u> , Data als <u>String</u>	Sendet eine Nachricht mit einer Befehls-ID Command und einem Text Data an die angegebene IP-Adresse <i>IPAddress</i> .
Unregister	GroupName als <u>String</u>	Entfernt die Registrierung der Gruppe <i>GroupName</i> .

Obwohl Sie Zugriff auf die Read- und Write-Methoden der `UDPSocket`-Klasse haben, sollten Sie diese nicht verwenden. Rufen Sie diese auf, wird eine `RuntimeException` mit entsprechenden Fehlermeldungen ausgelöst.

Hinweise

Die **EasyUDPSocket**-Klasse bietet eine einfache Abstraktion der `UDPSocket`-Klasse, die es unerfahrenen Benutzern einfacher macht, über UDP zu kommunizieren. Mit der **EasyUDPSocket**-Klasse können Sie Nachrichten an einzelne Computer oder an eine Gruppe von Computern schicken. Ein großer Vorteil der **EasyUDPSocket**-Klasse ist, daß sie „multicasting“ unterstützt. In der Regel ist es ärgerlich, herauszufinden, welche IP-Adresse für „multicasting“ genutzt werden kann.

Der Command-Parameter kann verwendet werden, um die Daten zu identifizieren, die geschickt wurden. Zum Beispiel können Sie einen Befehl mit der ID 100 senden, um festzulegen, daß diese Daten einem FolderItem entsprechen. Oder Sie definieren die ID 101, um die Daten als Benutzernamen zu klassifizieren. Die Festlegung der IDs und deren Sinn ist Ihnen überlassen.

Befehl-IDs unterhalb Null sind für die interne Benutzung reserviert. Wenn Sie Nachrichten verschicken, sollten Sie keine Befehl-IDs unterhalb Null verwenden, da diese mit einen Klassen interagieren können, was eventuell nicht beabsichtigt ist.

Um eine Nachricht an eine Gruppe zu schicken, geben Sie den Gruppennamen an, den Sie vorher mit der Register-Methode registriert haben. Das heißt, Sie können den String „MeineLokaleGruppe“ als Parameter übergeben, wenn Sie diese Gruppe vorher registriert haben. Um eine Nachricht an einen einzelnen Computer zu schicken, verwenden Sie bitte dessen IP-Adresse und die `SendMessageToIndividual`-Methode.

Sie können auch eine gültige Klasse D-IP-Adresse angeben; REALbasic wird diese verwenden. Eine weitere Hilfsfunktion ist die `Bind`-Methode. Diese Methode konfiguriert ein Socket (Sie müssen keine RouterHops oder `SendToSelf`-Eigenschaften einstellen) und bindet ihn an den angegebenen Port.

Eines sollten Sie sich merken; Bei einer **EasyUDPSocket** ist `SendToSelf` in der Voreinstellung aktiviert. Sie können dies überschreiben, indem Sie der Eigenschaft `False` zuweisen, nachdem Sie die `Bind`-Methode aufgerufen haben. Obwohl Sie weiterhin Multicast-Gruppen beitreten und austreten können, empfehlen wir Ihnen, die Register- und Unregister-Methoden nur für Ihre **EasyUDPSocket**-Applikationen zu verwenden.

Beispiele

An einen Port binden. `EditField1` enthält die Portnummer:

```
EasyUDPSocket1.Bind(Val(EditField1.Text))
```

Registrieren eine Gruppe. Alle Teilnehmer müssen denselben Namen verwenden.

```
EasyUDPSocket1.Register("MyGroup")
```

Eine Nachricht an eine Gruppen verschicken. EditField1 enthält den Text der Nachricht.

```
EasyUDPSocket1.SendMessageToGroup("MyGroup",100,EditField1.Text)
```

Eine Nachricht an ein Individium einer Gruppe verschicken. Die IP-Adresse des Individuum und der Port werden als erster Parameter angegeben. Der zweite Parameter ist die Befehls-ID und EditField1 enthält den Text der Nachricht.

```
Dim port as String
port = Str(EasyUDPSocket1.Port)
EasyUDPSocket2.SendMessageToIndividual("192.168.1.152"+"."+port,100, EditField1.Text )
```

Eine Nachricht über den ReceivedMessage-Event empfangen:

```
Sub ReceivedMessage (fromIP as String, Command as Integer, Data as String)
MsgBox "Received " + Str(command) + ": " + data + " from " + fromIP +EndOfLine
End Sub
```

Siehe auch

[UDPSocket](#), [AutoDiscovery](#), [SocketCore](#) Klassen.

EditableMovie Klasse

Wird verwendet, um QuickTime-Movies (Filme) zu erzeugen, zu verändern und zu speichern.

Übergeordnete Klasse

[Movie](#)

In früheren REALbasic-Versionen war **EditableMovie** von Object abgeleitet. Da es nun von Movie abgeleitet ist, können Sie ein **EditableMovie** unter Verwendung des MediaPlayer-Steuerelements anzeigen.

Eigenschaften

Name	Typ	Beschreibung
<i>Duration</i>	Double	Dauer des Filmes (in Sekunden)
EOF	Boolean	True, wenn <i>Position</i> das Ende des Filmes ist.
<i>Handle</i>	Integer	Liefert ein Handle auf das Editable Movie, meist für Toolbox-Aufrufe verwendet, die ein Handle auf einen Film benötigen.
Picture	Picture	Bild des Filmes an der aktuellen Position.
<i>Position</i>	Double	Aktuelle Position des Filmes (in Sekunden).
Poster	Picture	Titelbild des Filmes.
<i>SaveOnClose</i>	Boolean	Falls True, aktualisiert und speichert das EditableMovie die Datei, wenn der Bereich verlassen wird (so wie in früheren Versionen von REALbasic). Falls False, werden keine Aktualisierungen an der EditableMovie-Datei vorgenommen, bis Sie die CommitChanges-Methode aufrufen.
<i>Sellength</i>	Double	Die Länge (in Sekunden) des gewählten Segements des EditableMovies.
<i>SelStart</i>	Double	Die Start-Position (in Sekunden) des gewählten Segments der EditableMovies.
TimeDuration	Integer	Dauer des Filmes in TimeScale-Einheiten.
<i>TimeScale</i>	Integer	Legt die TimeScale (Zeitbasis) des Filmes in Einheiten 1/TimeScale pro Sekunde fest.
<i>TimeValue</i>	Integer	Aktuelle Position des Filmes in TimeScale-Einheiten.
TrackCount	Integer	Anzahl der Spuren im Film.
<i>UserData</i>	QTUserData	UserData enthält zusätzliche Informationen über den Film, Credits, Copyright-Informationen usw. Die QTUserData -Klasse wird verwendet, um die individuellen Teile der Daten zu verwalten.

Methoden

Name	Parameter	Beschreibung
<i>AppendMovieSegment</i>	SourceMovie als <u>Movie</u> , SourcePosition als <u>Double</u> , SourceDuration als <u>Double</u> , ShowProgress als <u>Boolean</u>	Hängt <i>SourceMovie</i> an das Ende des aufrufenden EditableMovie an. <i>SourcePosition</i> ist der Anfang (in Sekunden) des Segments in <i>SourceMovie</i> , das an den aufrufenden Film angehängt werden soll. <i>SourceDuration</i> ist die Länge (in Sekunden) in <i>SourceMovie</i> des Segments, das angehängt werden soll. Falls der Vorgang zeitintensiv ist, zeigt QuickTime einen modalen Dialog mit einem Fortschrittsbalken an, wenn <i>ShowProgress</i> <u>True</u> ist.
<i>Clear</i>		Löschen die aktuelle Auswahl des EditableMovie . Verwenden Sie die Eigenschaften <i>SelStart</i> und <i>SelLength</i> von EditableMovie , um die Auswahl festzulegen.
<i>CommitChanges</i>		Liefert ein <u>Boolean</u> . Es liefert <u>True</u> , wenn das Übernehmen erfolgreich war. Übernimmt Änderungen am EditableMovie manuell. Sie können dies ebenso mit dem Destructor der Klasse machen, wie in vorigen Versionen von REALbasic. Wenn Sie <i>SaveOnClose</i> auf <u>True</u> setzen, speichert es die Änderungen automatisch, wenn das EditableMovie den Bereich verläßt.
<i>Copy</i>		Kopiert die aktuelle Auswahl des EditableMovie in die Zwischenablage. Verwenden Sie die Eigenschaften <i>SelStart</i> und <i>SelLength</i> von EditableMovie , um die Auswahl festzulegen.
<i>Cut</i>		Schneidet die Auswahl des EditableMovie aus und kopiert sie in die Zwischenablage. Verwenden Sie die Eigenschaften <i>SelStart</i> und <i>SelLength</i> von EditableMovie , um die Auswahl festzulegen.
DataSize	<i>StartTime</i> als <u>Integer</u> <i>DurationTime</i> als <u>Integer</u>	Liefert einen <u>Integer</u> , die Anzahl von Bytes im durch <i>StartTime</i> und <i>DurationTime</i> festgelegten Segment.
<i>Flatten</i>	<i>Destination</i> als <u>FolderItem</u> , <i>ResourceFork</i> als <u>Boolean</u> , <i>FastStart</i> als <u>Boolean</u> , <i>ShowProgress</i> als <u>Boolean</u>	Packt das EditableMovie und speichert es in <i>Destination</i> . Macintosh-Anwender haben die Möglichkeit, den <i>ResourceFork</i> durch Setzen von <i>ResourceFork</i> auf <u>True</u> zu packen; Windows-Anwender packen immer den <i>Data-Fork</i> ein. Dieser Parameter wird in Programmen für Windows ignoriert. Setzen Sie <i>FastStart</i> auf <u>True</u> , um die "FastStart"-Funktion von QuickTime fürs Internet-Streaming zu setzen. Falls der Vorgang zeitintensiv ist, zeigt QuickTime einen modalen Dialog mit einem Fortschrittsbalken an, wenn <i>ShowProgress</i> <u>True</u> ist.
<i>InsertMovieSegment</i>	<i>sourceMovie</i> als <u>Movie</u> , <i>sourcePosition</i> als <u>Double</u> , <i>sourceDuration</i> als <u>Double</u> , <i>destinationPosition</i> als <u>Double</u> , <i>showProgress</i> als <u>Boolean</u>	Fügt von <i>SourceMovie</i> ein Segment an der durch <i>DestinationPosition</i> festgelegten Position ein. Das Segment in <i>SourceMovie</i> , das eingefügt werden soll, beginnt bei <i>SourcePosition</i> und wird fortgeführt durch <i>SourceDuration</i> . <i>SourcePosition</i> , <i>SourceDuration</i> und <i>DestinationPosition</i> sind in Sekunden anzugeben. Falls der Vorgang zeitintensiv ist, zeigt QuickTime einen modalen Dialog mit einem Fortschrittsbalken an, wenn <i>ShowProgress</i> <u>True</u> ist.
<i>NewSoundTrack</i>		Erzeugt und liefert einen <u>QTSoundTrack</u> . Sie können keinen neuen <u>QTSoundTrack</u> mit der <i>New</i> -Funktion erzeugen.
<i>NewVideoTrack</i>	<i>Width</i> als <u>Integer</u> <i>Height</i> als <u>Integer</u> <i>TimeScale</i> als <u>Integer</u>	<i>Breite</i> und <i>Höhe</i> in Pixeln. <i>TimeScale</i> ist Frames per second (Bilder pro Sekunde). Erzeugt einen neuen Video-Track und liefert ein <u>QTVideoTrack</u> .

Name	Parameter	Beschreibung
<i>Paste</i>		Fügt den Inhalt der Zwischenablage (vom Aufruf von Cut oder Copy) im aufrufenden EditableMovie an der Position SelStart ein. Verwenden Sie die Eigenschaften SelStart und SelLength um die Auswahl festzulegen. Wenn SelLength größer als 0 ist, überschreibt der Inhalt der Zwischenablage den ausgewählten Inhalt des EditableMovie .
<i>RestoreRedoEditState</i>		Bringt das EditableMovie in den Zustand zurück, in dem es sich befand, bevor SetRedoEditState aufgerufen wurde.
<i>RestoreUndoEditState</i>		Bringt das EditableMovie in den Zustand zurück, bevor SetUndoEditState aufgerufen wurde.
<i>ScaleMovieSegment</i>	StartPosition als <u>Double</u> , OldDuration als <u>Double</u> , NewDuration als <u>Double</u> , ShowProgress als <u>Boolean</u>	Ändert die Dauer des gewählten Segments im EditableMovie . Hinweis: Ein SelStart von 0 und eine SelLength von myEditableMovie.Duration kann verwendet werden, um den gesamten Film in der Länge zu skalieren. Eine Änderung der Dauer führt dazu, daß sich die Tonhöhe ebenfalls ändert. Wenn Sie den Film z.B. länger machen, nimmt die Tonhöhe ab. StartPosition ist der Anfang des zu skalierenden Segements. OldDuration ist die aktuelle Dauer des Segements und NewDuration ist die Dauer des Segments, auf die skaliert werden soll. Falls der Vorgang zeitintensiv ist, zeigt QuickTime einen modalen Dialog mit einem Fortschrittsbalken an, wenn <i>ShowProgress</i> <u>True</u> ist.
<i>SetRedoEditState</i>		Verwenden Sie die Methode, um einen Bearbeitungs-Status für das EditableMovie zu speichern. Das EditableMovie kann zu diesem Status jederzeit durch den Aufruf von RestoreRedoEditState zurückkehren, vorausgesetzt, der Bereich des EditableMovie wurde nicht verlassen.
<i>SetUndoEditState</i>		Verwenden Sie die Methode, um einen Bearbeitungs-Status für das EditableMovie zu speichern. Das EditableMovie kann zu diesem Status jederzeit durch den Aufruf von EditableMovie.RestoreUndoEditState zurückkehren, vorausgesetzt, der Bereich des EditableMovie wurde nicht verlassen.
<i>SoundTrack</i>	Index als <u>Integer</u>	Liefert den <u>QTSoundTrack</u> , der zu dem übergebenen Index gehört. Der Index ist ein 1-basiertes Array. Liefert einen <u>QTSoundTrack</u> .
<i>VideoTrack</i>	Index als <u>Integer</u>	Liefert den Video-Track, der zu dem übergebenen Index gehört. Der Index ist ein 1-basiertes Array. Liefert einen <u>QVideoTrack</u> .
<i>Track</i>	Index als <u>Integer</u>	Index ist ein 1-basiertes Array. Liefert einen <u>QTrack</u> .

Hinweise

Der *TimeScale*-Parameter legt die Anzahl von Zeiteinheiten fest, die jede Sekunde durchlaufen. Zum Beispiel setzt ein TimeScale-Wert von 60 die Zeitbasis auf sechzigstel Sekunden. Die Parameter *TimeValue* und *TimeDuration* werden in TimeScale-Einheiten angegeben.

Die Position oder die TimeValue-Eigenschaften können zum Positionieren des Films verwendet werden; an dieser Stelle enthält die Picture-Eigenschaft das Bild von dieser Stelle des Films.

Beispiele

Wenn man folgenden Code im Action-Event eines PushButtons ablegt, wird ein vorhandener Film als EditableMovie geöffnet. Der Code aktualisiert ImageWell und EditField mit dem aktuellen Bild und der aktuellen Position des Filmes.

```
Dim f as FolderItem
Dim m as EditableMovie
f=GetFolderItem("Honey.Mov")
m=f.OpenEditableMovie
```



```

m.position=movieplayer1.position
ImageWell1.image=m.picture
ImageWell2.image=m.poster
EditField1.text=str(m.position)

```

Folgendes Beispiel öffnet einen vorhandenen Film als ein EditableMovie und zeigt ihn in einem MoviePlayer-Steuerelement namens DerSpieler an.

```

Dim f As FolderItem
Dim derEFilm as EditableMovie
f=GetOpenFolderItem("video/quicktime")
If f<>Nil and f.exists then
  derEFilm=f.OpenEditableMovie
  If derEFilm<>Nil then
    DerSpieler.movie=derEFilm
  end if
end if

```

Das folgende Beispiel erzeugt eine neue Video-Spur (Track) aus zwei Bildern, p1 und p2, die in das Projekt-Fenster gedraggt wurden.

```

Dim sequence as QTEffectSequence
Dim i,effectID as Integer
Dim theEffect as QTEffect
Dim track as QTVideoTrack
Dim m as EditableMovie
Dim f as FolderItem
f=GetFolderItem("Movie")
m=f.CreateMovie
If radiobutton1.value then
  theEffect=GetQTCrossFadeEffect
else
  theEffect=GetQTSMPTEEffect(val(EffectsList.cell(EffectsList.ListIndex,1)))
end if
sequence=New QTEffectSequence(theEffect,p1,p2,96)
track=m.NewVideoTrack(p1.width, p1.height, 32)

```

Siehe auch

QTrack, QTUserData, QTVideoTrack, FolderItem Klassen.

EditField Steuerelement



Das Standard-EditField. Ein EditField-Steuerelement kann eine Zeile Text oder mehrere Zeilen Text enthalten, je nachdem, wie der Wert der Eigenschaft *MultiLine* gesetzt ist. Der Text kann ferner nur aus einem Font, einer Fontgröße, einem Stil oder aus verschiedenen Fonts, Fontgrößen und Stilen („Styled Text“) bestehen. Styled Text kann mit Hilfe der StyledTextPrinter-Klasse gedruckt werden. Text mit Stil-Eigenschaften kann unabhängig von einem EditField über die StyledText Klasse verwendet werden.

EditFields, ListBoxen und Canvas-Steuerelemente sind die einzigen Steuerelemente, die den Fokus besitzen können (unter Mac OS).

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>AcceptTabs</i>	<u>Boolean</u>	Wenn <u>True</u> , wird ein Tabulatorzeichen in das Feld eingefügt, statt daß der Einfügekursor zum nächsten Feld wechselt.

Name	Art	Beschreibung
<i>Alignment</i>	<u>Integer</u>	Stellt die Absatzausrichtung für den gesamten Inhalt des EditFields ein. 0: Standard-Ausrichtung 1: Linksbündig 2: Zentriert 3: Rechtsbündig Verwenden Sie SelAlignment, um die Ausrichtung für bestimmte Absätze einzustellen. Derzeit ist die Standard-Ausrichtung die gleiche wie linksbündig.
<i>BackColor</i>	<u>Color</u>	Die Hintergrundfarbe (Standard ist Weiß).
<i>Bold</i>	<u>Boolean</u>	Stellt den Text fett dar.
Border	<u>Boolean</u>	Zeichnet einen Rahmen um den Text, wenn <u>True</u> .
DataField	<u>String</u>	Nur relevant, wenn das EditField in Verbindung mit einem <u>DataControl</u> verwendet wird, um den Inhalt von Feldern einer Datenbanktabelle anzuzeigen. Der Name eines Feldes in der Tabelle, auf die das <u>DataControl</u> verweist.
<i>DataSource</i>	<u>DataControl</u>	Das <u>DataControl</u> , das auf eine Tabelle in einer Datenbank verweist, dessen Feld mittels eines <u>EditFields</u> dargestellt wird. Verwenden Sie die DataField-Eigenschaft, um jedes anzuzeigende Feld mit einem EditField zu verbinden. In der IDE erscheint ein Popup-Menü mit den Feldnamen.
<i>Italic</i>	<u>Boolean</u>	Stellt den Text kursiv dar.
<i>LimitText</i>	<u>Integer</u>	Gibt an, wie viele Zeichen in dieses Eingabefeld eingegeben werden dürfen. Ein Wert von Null begrenzt den Text nicht. Funktioniert nur, wenn MultiLine nicht gewählt ist.
<i>LiveUpdate</i>	<u>Boolean</u>	Wenn LiveUpdate True ist, wird der Inhalt des EditField ständig aktualisiert. Dies ist notwendig, wenn das EditField mit einem anderen Steuerelement in Verbindung steht.
MultiLine	<u>Boolean</u>	Bricht Text automatisch auf mehrere Zeilen um und erlaubt es, mehr als 32K an Text einzugeben. Das seitensweise Blättern und das Springen an Anfang und Ende mit Hilfe der Tastatur wird hierbei ebenso unterstützt. Ein horizontaler Scrollbalken wird automatisch hinzugefügt, wenn MultiLine True ist. MultiLine muß auf True gesetzt sein, um Text mit Stilinformationen und automatisches Drag&Drop zu unterstützen. MultiLine-Editfields unterstützen alle Zeilentrenner (Return, Linefeed oder Return + Linefeed), um das Ende eines Absatzes zu markieren. Wenn MultiLine ausgeschaltet ist, ist der Inhalt für EditFields auf dem Mac auf 32 K Text beschränkt.
<i>Password</i>	<u>Boolean</u>	Statt der eingegebenen Zeichen erscheinen fette Punkte und die Menüeinträge zum Kopieren und Ausschneiden werden automatisch ausgeschaltet. Die MultiLine-Eigenschaft muß False sein, damit ein EditField als Paßwort-Feld verwendet werden kann.
<i>ReadOnly</i>	<u>Boolean</u>	Wenn True, kann der Text nicht verändert werden.
Scrollbar- Horizontal	<u>Boolean</u>	Wenn True, wird eine horizontale Scrollleiste dargestellt und der Text wird am rechten Rand des EditFields nicht umbrochen.
Scrollbar- Vertical	<u>Boolean</u>	Wenn True, wird eine vertikale Scrollleiste dargestellt. In früheren REALbasic-Versionen hieß die Eigenschaft nur „Scrollbar“.
<i>ScrollPosition</i>	<u>Integer</u>	Null-basierter Index der obersten sichtbaren Linie. Ermitteln Sie ScrollPosition, um die erste sichtbare Linie zu bestimmen. Schreiben Sie in Scrollposition, um das EditField zu scrollen. Wenn Sie ScrollPosition auf die letzte Zeile oder darunter setzen, wird der Scrollbar ans untere Ende gesetzt, es wird aber nicht unbedingt die letzte Zeile am oberen Ende des EditFields angezeigt. ScrollPosition ist 0, wenn die MultiLine-Eigenschaft False ist. Dies kann in der IDE eingestellt werden.
<i>ScrollPositionX</i>	<u>Integer</u>	Scrollt das EditField horizontal an die Position X Pixel von der linken Kante des EditFields.

Name	Art	Beschreibung
<i>SelAlignment</i>	<u>Integer</u>	Stellt die Absatzausrichtung für den selektierten Text ein. 0: Standard-Ausrichtung 1: Linksbündig 2: Zentriert 3: Rechtsbündig -1: Gemischt – Die Selektion umfaßt mehrere Absätze mit verschiedener Ausrichtung. Derzeit ist die Standard-Ausrichtung das gleiche wie linksbündig.
<i>SelBold</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „Fett“ für den ausgewählten Text auszulesen oder neu festzulegen.
<i>SelCondense</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „komprimiert“ für den ausgewählten Text auszulesen oder neu festzulegen.
<i>SelExtend</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „gesperrt“ für den ausgewählten Text auszulesen oder neu festzulegen.
<i>SelItalic</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „kursiv“ für den ausgewählten Text auszulesen oder neu festzulegen.
<i>SelLength</i>	<u>Integer</u>	Die Anzahl ausgewählter Zeichen. Ein SelLength von 0 bedeutet, daß nichts selektiert ist und die Einfügemarke sichtbar ist. Ein Wert größer als die Anzahl der Buchstaben im EditField bedeutet, daß die Selektion von SelStart bis zum Ende des EditFields vorgenommen wird.
<i>SelOutline</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „Konturschrift“ für den ausgewählten Text auszulesen oder neu festzulegen. Wird unter Windows oder Mac OS X nicht unterstützt.
<i>SelShadow</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „Schattiert“ für den ausgewählten Text auszulesen oder neu festzulegen. Wird unter Windows oder Mac OS X nicht unterstützt.
<i>SelStart</i>	<u>Integer</u>	Die Position des ersten ausgewählten Zeichens. Ein Wert von 0 bedeutet vor dem ersten Zeichen.
<i>SelText</i>	<u>String</u>	Der momentan ausgewählte Text.
<i>SelTextColor</i>	<u>Color</u>	Wird verwendet, um die Farbe des ausgewählten Textes abzufragen oder neu festzulegen. Wenn es in einem EditField ohne Stilinformationen verwendet wird, ändert es die Farbe des gesamten Textes in diesem EditField.
<i>SelTextFont</i>	<u>String</u>	Wird verwendet, um die Schriftart des ausgewählten Textes abzufragen oder neu festzulegen. Wenn es in einem EditField ohne Stilinformationen verwendet wird, ändert es die Schriftart des gesamten Textes in diesem EditField.
<i>SelTextSize</i>	<u>Integer</u>	Wird verwendet, um die Schriftgröße des ausgewählten Textes abzufragen oder neu festzulegen. Wenn es in einem EditField ohne Stilinformationen verwendet wird, ändert es die Textgröße des gesamten Textes in diesem EditField.
<i>SelUnderline</i>	<u>Boolean</u>	Wird verwendet, um die Textstil-Eigenschaft „unterstrichen“ für den ausgewählten Text auszulesen oder neu festzulegen.
Styled	<u>Boolean</u>	Wenn True, werden Textstile zugelassen. Um Styled Text verwenden zu können, muß die MultiLine-Eigenschaft auf True gesetzt sein.
<i>Text</i>	<u>String</u>	Der angezeigte Text. Wenn Sie der Text-Eigenschaft einen String zuweisen, wird der gesamte Inhalt des EditFields dadurch ersetzt. Der Font, die Größe und Farbe sind gleich und entsprechen den zuletzt mit den Eigenschaften TextFont, TextSize und TextColor eingestellten Werten, oder, wenn diese nicht verwendet wurden, den Einstellungen aus der IDE. (Die Verwendung von SelTextFont etc. kann das gesamte Aussehen des Textes ändern – immer, wenn das Feld nicht styled ist – es ändert aber nicht die Voreinstellungen, die verwendet werden, wenn der gesamte Text über die Text-Eigenschaft ersetzt wird.)
<i>TextColor</i>	<u>Color</u>	Die Textfarbe (Standard ist Schwarz).

Name	Art	Beschreibung
<i>TextFont</i>	<u>String</u>	Der Name der Schriftart, die zum Anzeigen des Textes verwendet wird. Wenn Sie TextFont eine andere Schriftart zuweisen, ändert das die Schriftart vom gesamten Text in diesem EditField. Es stellt außerdem die Voreinstellung wieder her, die verwendet wird, wenn die Text-Eigenschaft später verwendet wird, um den Inhalt des Textes im EditField zurückzusetzen.
<i>TextSize</i>	<u>Integer</u>	Die Schriftgröße des darzustellenden Textes.
<i>TextStyleData</i>	<u>String</u>	Die Textstil-Informationen für das Eingabefeld. Diese Informationen können in einer „styl“-Resource gesichert und mit der SetTextAndStyle Methode wieder eingestellt werden.
<i>Underline</i>	<u>Boolean</u>	Wendet den Schriftstil „unterstrichen“ auf den Text an.
<i>UseFocusRing</i>	<u>Boolean</u>	Wenn <u>True</u> , wird das EditField umrahmt und zeigt so, daß es den Fokus hat. Wenn <u>False</u> , ändert sich das Aussehen des Objekts nicht, wenn es den Fokus besitzt.

Events

Name	Parameter	Beschreibung
<i>GotFocus</i>		Der Einfügeschursor wurde in den Eingabebereich bewegt.
<i>KeyDown</i>	Key als <u>String</u>	Der Anwender hat diese Taste gedrückt, während der Eingabebereich aktiv war. Liefert ein Boolean. Wenn True zurückgeliefert wird, heißt das, daß Key nicht weiter verarbeitet werden muß.
<i>LostFocus</i>		Der Einfügeschursor hat den Eingabebereich verlassen.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Steuerelements an der Position x,y gedrückt. Geben Sie <u>True</u> zurück, wenn Sie selbst MouseDown auswerten möchten. Das Rückgeben von <u>True</u> verhindert, daß das EditField den Mausklick auswertet.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Steuerelements an der Position x,y losgelassen. Dieses Event tritt nicht auf, bis Sie <u>True</u> im MouseDown-Event zurückgeben.
<i>SelChange</i>		Die Auswahl des markierten Textes wurde geändert.
<i>TextChange</i>		Die Text-Eigenschaft des Feldes wurde geändert.

Methoden

Name	Parameter	Beschreibung
<i>BindValue</i>	Wert als <u>StringProvider</u>	Wird bei der Implementierung benutzerdefinierter Objektverknüpfungen verwendet.
<i>CharPosAtLineNum</i>	LineNumber als <u>Integer</u>	Liefert (als <u>Integer</u>) in einem MultiLine-EditField die Zeichenposition des ersten Zeichens in der Zeile LineNumber. Die Zeichen sind vom Start bis zum Ende des EditFields fortlaufend numeriert. Das erste Zeichen trägt die Nummer 1. Die erste Zeile trägt die Nummer 0. Wenn MultiLine nicht <u>True</u> ist, liefert diese Funktion 0, ohne Rücksicht auf <i>LineNumber</i> .
<i>CharPosAtXY</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Liefert (als <u>Integer</u>) das Zeichens an den Pixelkoordinaten X, Y relativ zum EditField. Die Zeichen sind durchgehend vom Start bis zum Ende des EditField numeriert. Das erste Zeichen hat die Nummer 1.
<i>Copy</i>		Kopiert den selektierten Text inklusive Stilinformationen in die Zwischenablage.
<i>LineNumAtCharPos</i>	CharPosition als <u>Integer</u>	Liefert (als <u>Integer</u>) die Zeilennummer, in der das Zeichen an <i>CharPosition</i> erscheint. Die Zeichen sind fortlaufend durchnumeriert, das erste Zeichen trägt die Nummer 1. Die erste Zeile trägt die Nummer 0. Wenn diese Funktion mit ScrollPosition eingesetzt wird, kann man das EditField an eine bestimmte Stelle im Text scrollen lassen.
<i>Paste</i>		Fügt den Text mit oder ohne Stilinformationen aus der Zwischenablage an der Einfügemarke im EditField zum vorhandenen Text ein.

Name	Parameter	Beschreibung
<i>SetFocus</i>		Aktiviert den Eingabebereich, verschickt alle KeyDown-Events und bewegt den Einfügecursor im Feld.
<i>SetTextAndStyle</i>	Text als <u>String</u> , TextStyleData als <u>String</u>	Setzt den Text in die Text-Eigenschaft des EditFields und wendet die Textstile aus TextStyleData darauf an. Der TextStyleData-String ist mit der „styl“-Resource kompatibel.
<i>StyledTextPrinter</i>	g als <u>Graphics</u> , Width als <u>Integer</u>	Wird verwendet, um ein <u>StyledTextPrinter</u> -Objekt zu erzeugen, um den Inhalt eines EditFields als Styled Text zu drucken. Die MultiLine-Eigenschaft des Eingabefeldes muß auf True gesetzt sein, um Styled Text drucken zu können. Liefert ein StyledTextPrinter-Objekt. Width (in Pixeln) ist die Breite des bedruckbaren Bereichs.
<i>ToggleSelectionBold</i>		Wechselt zwischen fetter und vorheriger Darstellung des ausgewählten Textes.
<i>ToggleSelectionCondense</i>		Wechselt zwischen unterstrichener und vorheriger Darstellung des ausgewählten Textes. Wird unter Windows und Mac OS X nicht unterstützt.
<i>ToggleSelectionExtend</i>		Wechselt zwischen gesperrter und vorheriger Darstellung des ausgewählten Textes. Wird unter Windows und Mac OS X nicht unterstützt.
<i>ToggleSelectionItalic</i>		Wechselt zwischen kursiver und vorheriger Darstellung des ausgewählten Textes.
<i>ToggleSelectionOutline</i>		Wechselt zwischen Konturschrift und vorheriger Darstellung des ausgewählten Textes. Wird unter Windows und Mac OS X nicht unterstützt.
<i>ToggleSelectionShadow</i>		Wechselt zwischen schattierter und vorheriger Darstellung des ausgewählten Textes. Wird unter Windows und Mac OS X nicht unterstützt.
<i>ToggleSelectionUnderline</i>		Wechselt zwischen unterstrichener und vorheriger Darstellung des ausgewählten Textes.

Hinweis

Arbeiten mit Textstilen in EditFields

Um Text mit Textstilen in einem EditField anzuzeigen, muß die Eigenschaft „Style“ für das EditField aktiviert sein. Die Eigenschaften *SelBold*, *SelTextColor*, *SelCondense*, *SelExtend*, *SelItalic*, *SelOutline*, *SelShadow* und *SelUnderline* können verwendet werden, um sowohl den Textstil des ausgewählten Textes abzufragen wie auch diesen neu festzulegen (Die Eigenschaften *Outline*, *Shadow*, *Condense* und *Extend Style* sind nur unter Mac OS 8-9). Diese Eigenschaften werden automatisch von REALbasic zur Laufzeit behandelt, wenn Text in einem EditField ausgewählt wird. Um den Stil festzulegen, müssen Sie lediglich der entsprechenden Eigenschaft den Wert True zuzuweisen. Um z.B. den ausgewählten Text im EditField „Editfield1“ fett darzustellen, können Sie folgenden Befehl verwenden:

```
Editfield1.SelBold=True
```

Sie können mit der gleichen Eigenschaft feststellen, ob der ausgewählte Text in einem bestimmten Schriftstil vorliegt. Der folgende Quelltext gibt einen Warnton aus, wenn der ausgewählte Text fett dargestellt wird:

```
If EditField1.SelBold Then
    Beep
End If
```

Die Eigenschaften *SelBold*, *SelCondense*, *SelExtend*, *SelItalic*, *SelOutline*, *SelShadow* und *SelUnderline* haben den Wert „True“, wenn alle ausgewählten Zeichen den zu prüfenden Textstil haben. Andernfalls haben diese Eigenschaften den Wert False.

Auf die gleiche Weise können Sie mit Hilfe der Eigenschaften *SelTextFont* und *SelTextSize* die Schriftart und die Schriftgröße des ausgewählten Textes ändern oder abfragen. Um die Schriftart des ausgewählten Textes einzustellen, setzen Sie die Eigenschaft *SelTextFont* auf den Namen der zu verwendenden Schriftart. Wenn im ausgewählten Text mehr als eine Schriftart verwendet wird, liefert *SelTextFont* einen leeren String zurück. Wenn im ausgewählten Text mehr als eine Schriftgröße verwendet wird, hat die Eigenschaft *SelTextSize* für diesen Text den Wert Null.

Wenn im markierten Text mehrere Schriftarten, Schriftgrößen oder Textstile verwendet wurden, so können Sie mit den Eigenschaften *SelStart* und *SelLength* diese zeichenweise selektieren, um festzustellen, welcher Stil, Font und Größe verwendet wird. Bitte beachten Sie, daß, wenn ein EditField mehr als einen Stil, Font oder Fontgröße enthält und Sie den Wert der Text-Eigenschaft neu zuweisen, Sie die Stil-Informationen des darin enthaltenen Strings verlieren. Der gesamte Text wird dann im gleichen Stil formatiert. Um den Text in einem EditField mit Stil-Informationen zu aktualisieren, ist es am sichersten, die Eigenschaften SelStart, SelLength und SelText zu verwenden, um die Textselektionen zu aktualisieren.

Mit der *SelTextColor*-Eigenschaft können Sie auch die Farbe des ausgewählten Textes bestimmen oder einstellen. Dieser Quelltext prüft, ob der ausgewählte Text Weiß ist und stellt ihn dann auf Schwarz ein:

```
Dim White, Black as Color
White=RGB(255,255,255)
Black=RGB(0,0,0)
If EditField1.SelTextColor=White Then
    EditField1.SelTextColor=Black
End If
```

Um Styled Text eines EditFields zu drucken, erzeugen Sie mit der *StyledTextPrinter*-Methode ein *StyledTextPrinter*-Objekt. Benutzen Sie dann die *DrawBlock*-Methode von *StyledTextPrinter*. Ein Beispiel hierzu finden Sie bei der Beschreibung von *StyledTextPrinter*.

Styled Text unter Windows

Folgende Styles werden auf der Windows-Plattform nicht unterstützt: Outline, Shadow, Condensed, und Extended.

Einzeilige EditFields

Einzeilige EditFields unter Mac OS 9 (oder älter) sind auf 32 KB Textinhalt begrenzt. Sie unterstützen die ASCII-Zeichen-Codes über 127 und den Symbol-Font.

Benutzen von StyledText

Mit der *StyledText*-Klasse können Sie Stil-Text unabhängig vom EditField verwenden. Der Stil-Text ist über die Text-Eigenschaft des *StyledText*-Objekts erreichbar und der Stil kann über die Methoden Bold, Italic, Underline, Font, Size und TextColor gesetzt werden. Um einen Stil-Text darzustellen, können Sie das *StyledText*-Objekt der *StyledText*-Eigenschaft eine EditFields zuweisen. Alle Änderungen, die Sie vornehmen, während der Text angezeigt wird, werden augenblicklich übernommen.

Die Eigenschaften des **EditFields** zum Setzen und Lesen von Stil-Attributen werden bei der Anzeige berücksichtigt, solange das *StyledText*-Objekt in dem EditField angezeigt wird.

Für weitere Informationen lesen Sie bitte die Beschreibung der *StyledText*-Klasse.

Eingabefilter (Input Entry Filters)

Sie können die Mask-Eigenschaft benutzen, um Benutzereingaben zeichenweise zu filtern oder Formatierungen vorzunehmen. Sie können beispielsweise einen Filter zu Eingabe von Telefonnummer erzeugen, indem Sie nur Klammern, Leerzeichen, einen Bindestrich und, über das Zeichen "#", die Eingabe auf Ziffern beschränken.

Die folgende Tabelle zeigt Ihnen alle Zeichen, die Sie zum Definieren eines Filter benutzen können.

Filter-Zeichen	Beschreibung
#	Platzhalter für ein einzelnes Zeichen. Der Benutzer kann an dieser Stelle nur ein Zeichen eingeben.
.	Dezimal-Platzhalter. An dieser Stelle wird das Dezimaltrennzeichen eingefügt, das in den Landeseinstellungen definiert wird.
,	Tausender-Trennzeichen. An dieser Stelle wird das Tausender-Trennzeichen eingefügt, das in den Landeseinstellungen definiert ist.
:	Zeit-Trennzeichen. An dieser Stelle wird das Zeit-Trennzeichen eingefügt, das in den Landeseinstellungen definiert ist.
/	Datum-Trennzeichen. An dieser Stelle wird das Datum-Trennzeichen eingefügt, das in den Landeseinstellungen definiert ist.
\	Das Filter-Escape-Zeichen. Das nachfolgende Zeichen wird als Zeichen gewertet. Damit kann man die Zeichen '#', '&', 'A', '?' (und so weiter) verwenden.
&	Platzhalter für ein Zeichen. Gültige ASCII-Werte sind 32-126 und die Nicht-ASCII-Zeichen 128-255

Filter-Zeichen	Beschreibung
>	Konvertiert alle nachfolgenden Zeichen in Großbuchstaben. Dies funktioniert über die ASCII-Begrenzung hinaus. Sprich ü wird zu Ü.
<	Konvertiert alle nachfolgenden Zeichen in Kleinbuchstaben. Dies funktioniert über die ASCII-Begrenzung hinaus. Sprich Ü wird zu ü.
A	Alphanumerischer Platzhalter, der eine Eingabe erzwingt. Zum Beispiel "AAA" erzwingt die Eingabe von drei alphabetischen Zeichen.
a	Alphanumerischer Platzhalter, der keine Eingabe erzwingt.
9	Numerischer Platzhalter, der keine Eingabe erzwingt.
C	Ein Platzhalter für Zeichen oder Leerzeichen, deren Eingabe optional ist. Verhält sich wie der '&'-Platzhalter.
?	Alphabetischer Platzhalter.
Any literal	Alle anderen Symbole, die zur Formatierung angezeigt werden.
~	Für zukünftige Benutzung reserviert. Die Benutzung löst eine Ausnahme (Exception) aus. Benutzen Sie stattdessen \~.

Hier sind einige Beispiele:

```
EditField1.Mask="###-##-####" //US Social Security number
EditField1.Mask="(###) ###-####" //US Phone number, with area code
```

Wenn der Benutzer ein unzulässiges Zeichen eingibt, wird ein `ValidationError`-Event ausgelöst. Das eingegebene Zeichen und die Position werden an diesen Event-Handler übergeben, wo Sie die Fehleingabe behandeln können.

Um einen Filter zu löschen, weisen Sie der Mask-Eigenschaft einen leeren String zu:

```
EditField1.Mask=""
```

Einem EditField Text hinzufügen

Wenn Sie an ein EditField Text anhängen möchten, kann es sein, daß das EditField störend aufflackert, da REALbasic das EditField neu zeichnet, um den Text darzustellen:

```
EditField1.Text=EditField1.Text+"mein neuer Text"
```

Dies passiert, da der Inhalt des gesamten EditFields neu gezeichnet werden muß. Um dieses Flackern zu vermeiden, sollten Sie den Cursor an das Ende des Eingabefeldes setzen und dann erst den Text an das EditField anhängen:

```
EditField1.SelectStart=Len(EditField1.Text)
EditField1.SelectText="mein neuer Text"
```

Textkodierung unter Carbon

Unter Carbon/Mac OS X speichern EditFields den gesamten Text intern in Unicode, welches in der Lage ist, eine Mixtur aus Fonts unterschiedlicher Schreibsysteme zu repräsentieren. Wenn Sie den Text über die Text- oder SelText-Eigenschaft extrahieren, wird der Text in UTF-8 zurückgeliefert.

Siehe auch

[Font](#), [FontCount](#) Funktionen; [StyledTextPrinter](#), [StyledText](#), [StyleRun](#), [Paragraph](#), [Range](#) Klassen.

Eigenschaft hat keinen Wert Fehler

Sie haben versucht, einen Wert aus einer Eigenschaft zu lesen, die keinen Wert hatte.

Ein Erweiterter Parameter kann keinen Standardwert haben Fehler

Sie haben einen Standardwert zugewiesen, obwohl Sie das `Extends`-Schlüsselwort verwenden.

Beispiel

Die folgende Parameterdeklaration ist ungültig.

```
Extends C as Color=$cFF0000, Amt as Integer
```

Siehe auch

Extends-Schlüsselwort.

Ein Assigns-Parameter kann keinen Standardwert haben Fehler

Sie haben einen Standardwert angegeben, obwohl Sie das Assigns-Schlüsselwort verwenden. Dies ist ungültig.

Beispiel

Die folgende Methodendeklaration ist ungültig.

```
Sub theVolume(a As Integer, b As Integer, Assigns c As Integer=10)
```

Sie müssen die Zuweisung (=10) entfernen.

Siehe auch

Assigns-Schlüsselwort.

Ein Interface dieser Klasse ist nicht vom Typ Klassen-Interface Fehler

Im Eigenschaften-Fenster eines Klassen-Interfaces haben Sie den Namen einer Klasse in das Interface-Feld eingegeben, die nicht vom Typ Klassen-Interface ist. Dies tritt auf, wenn Sie den Namen einer „normalen“ Klasse an Stelle des Namens eines Klassen-Interfaces verwenden.

Eine Methode mit einem Parameter-Array, kann keine optionalen Parameter haben Fehler

Ein Parameter nach einem Parameter-Array Fehler

Sie haben einen weiteren Parameter deklariert, nachdem Sie einen Parameter als ParamArray deklariert haben. Dies ist nicht erlaubt. Geben Sie nur den Parameter an, den Sie als ParamArray deklariert haben.

Beispiel

In der folgenden Deklaration ist der Parameter „m“ unzulässig.

```
Sub myMethod(ParamArray n as Integer, m as Integer)
```

Siehe auch

ParamArray-Schlüsselwort.

Ein ParamArray kann keine Arrays enthalten Fehler

Sie haben versucht, ein Array als ParamArray zu deklarieren.

Beispiel

Diese Methodendeklaration ist ungültig, da ein Array als ParamArray deklariert wird. Dies ist unzulässig. Sie können eine unbegrenzte Anzahl von Elementen als Parameter übergeben, wenn Sie entweder ein Array als Argument benutzen oder ein Argument als ParamArray deklarieren. Sie können nicht beides gleichzeitig benutzen.

```
Sub meineMethode(ParamArray b() As String)
```

Siehe auch

ParamArray Schlüsselwort, Dim Anweisung

Ein Parameter-Array kann keinen Standardwert haben Fehler

Wenn Sie einen Parameter als ParamArray deklarieren, können Sie diesem keinen Standardwert zuweisen.

Beispiel

Das folgende Beispiel zeigt eine ungültige Methodendeklaration:

```
Sub AddiereZahlen(ParamArray Nums as Integer = 10)
```

Sie müssen die Zuweisung am Ende der Deklaration entfernen.

Siehe auch

[ParamArray](#) Schlüsselwort

Ein Parameter-Array kann nicht als Referenz übergeben werden Fehler

Sie haben versucht einen Parameter mit „ParamArray“ und „ByRef“ zu benutzen. Parameter-Arrays können nicht als Referenz an Methoden übergeben werden.

Ein Referenz-Parameter kann keinen Standardwert haben Fehler

In der Methodendeklaration haben Sie einen Parameter als Referenz (ByRef) deklariert. Sie versuchen diesem Parameter einen Standardwert zuzuweisen.

Beispiel

Die folgende Parameterdeklaration ist ungültig:

```
ByRef a As Integer = 5
```

Sie können entweder die Zuweisung oder das Schlüsselwort „ByRef“ entfernen.

Siehe auch

[ByRef](#) Schlüsselwort

EmailAttachment Klasse

Wird verwendet, um einen Anhang (Attachment) für eine E-Mail aufzubewahren.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>ContentEncoding</i>	String	Content-Encoding des Anhangs.
<i>Data</i>	String	Der Inhalt des Anhangs, wenn ein solcher erhalten wird.
<i>MacCreator</i>	String	Vierbuchstabiger Macintosh-Creator-Code der Anhang-Datei.
<i>MacType</i>	String	Vierbuchstabiger Macintosh-Type-Code.
<i>MIMEType</i>	String	MIME-Typ des Anhangs.
<i>Name</i>	String	Name der angehängten Datei.

Methoden

Name	Parameter	Beschreibung
<i>LoadFromFile</i>	Datei als FolderItem	Hängt die übergebene Datei namens <i>File</i> an.
<i>SaveToFile</i>	Datei als FolderItem	Speichert den Anhang in eine Datei <i>File</i> . Liefert ein Boolean ; True falls erfolgreich.

Beispiel

Folgendes Beispiel übernimmt den Pfad auf eine Datei und fügt diese als einen E-Mail-Anhang an. Der Pfad wurde in ein [EditField](#) namens fileFld eingegeben.

```
Dim file as EmailAttachment
```

```

Dim mail as EmailMessage
If fileFld.text <> "" then
    file = New EmailAttachment
    file.loadFromFile GetFolderItem(fileFld.text)
    mail.Attachments.Append file
end if

```

Siehe auch

EmailMessage, EmailHeaders, POP3Socket, POP3SecureSocket, SMTPSocket, SMTPSecureSocket, TCPsocket, SocketCore Klassen.

EmailHeaders Klasse

Wird zum Aufbewahren der Header einer E-Mail verwendet.

Übergeordnete Klasse

InternetHeaders

Methoden

Name	Parameter	Beschreibung
<i>AppendHeader</i>	Name als <u>String</u> , Value als <u>String</u>	Hängt einen neuen Header an.
<i>Count</i>		Liefert die Anzahl der Header als ein <u>Integer</u> zurück.
<i>Delete</i>	Name als <u>String</u> ODER Name als <u>String</u> , Index als <u>Integer</u>	Löscht den angegebenen E-Mail-Header.
<i>Name</i>	Index als <u>Integer</u>	Liefert als <u>String</u> den Namen des angegebenen Headers.
<i>SetHeader</i>	Name als <u>String</u> , Value als <u>String</u>	Legt den Namen und Wert eines Headers fest. E-Mails, die einen Anhang haben, besitzen zusätzlich eine MIME-Version des Headers.
<i>Source</i>		Liefert einen <u>String</u> , der den rohen Quelltext des Headers enthält.
<i>Value</i>	Name als <u>String</u> ODER Name als <u>String</u> , Index als <u>Integer</u> ODER Index als <u>Integer</u>	Liefert einen <u>String</u> , der den Wert des angegebenen Headers enthält.
<i>ValueCount</i>	Name als <u>String</u>	Liefert einen <u>Integer</u> . Liefert die Anzahl der Header-Werte, die den Namen haben, der durch <i>Name</i> festgelegt wird. Dies ist eingebaut, da es möglich ist, mehr als einen Header mit dem gleichen Namen zu haben.

Hinweis

Vor REALbasic Version 5.5 wurde diese Klasse von der Object-Klasse abgeleitet. Sie ist nun eine Unterklasse von InternetHeaders. Bitte lesen Sie auch die Beschreibung von InternetHeaders für deren Methoden.

Beispiel

Folgender Code setzt den Header einer zu verschickenden E-Mail.

```

Dim mail as EmailMessage
mail = New EmailMessage
mail.headers.AppendHeader "X-Mailer","REALbasic SMTPSocket Demo"

```

Siehe auch

InternetHeaders, EmailMessage, EmailAttachment, POP3Socket, POP3SecureSocket, SMTPSocket, SMTPSecureSocket, TCPsocket, SSLsocket, SocketCore Klassen.

EmailMessage Klasse

Wird zum Aufbewahren des Inhalts einer E-Mail-Nachricht und seiner Anhänge, falls vorhanden, verwendet.

Übergeordnete Klasse

Object

Eigenschaften

Name	Parameter	Beschreibung
Attachments	<u>EmailAttachment</u>	Das Array aus EmailAttachments, das an diese E-Mail-Nachricht angehängt ist.
BCCAddress	<u>String</u>	Liefert eine komma-separierte Liste von <u>Strings</u> . Liefert die BCC-Adressen für diese Mail.
BodyEnriched	<u>String</u>	Der Haupttext der E-Mail-Nachricht im Enriched-Format. Manche E-Mail-Programme wie Mail verwenden dieses Format, wenn sie Mails mit Stilinformationen senden.
BodyHTML	<u>String</u>	Der Haupttext der E-Mail als HTML.
BodyPlainText	<u>String</u>	Der Haupttext der Mail als reiner Text. Programme, die HTML oder Enriched nicht unterstützen, werden die Nachricht als reinen Text unter Verwendung des Inhalts dieser Eigenschaft anzeigen.
CCAddress	<u>String</u>	Liefert eine komma-separierte Liste von <u>Strings</u> . Liefert die CC-Adressen für diese Mail.
FromAddress	<u>String</u>	Liefert einen <u>String</u> , der die Adresse enthält, von der die E-Mail kommt.
FromAddress	<u>String</u>	Setzt die From-Adresse für diese E-Mail-Nachricht.
Headers	<u>EmailHeaders</u>	Die Header für die E-Mail-Nachricht.
Source	<u>String</u>	Liefert einen <u>String</u> , der den rohen Quelltext der E-Mail-Nachricht enthält.
Subject	<u>String</u>	Liefert einen <u>String</u> . Liefert den Betreff (Subject) der E-Mail.
Subject	Subject als <u>String</u>	Legt den Betreff (Subject) der Mail fest.
ToAddress	<u>String</u>	Liefert eine komma-separierte Liste aus <u>Strings</u> . Liefert die Adressen für diese E-Mail.

Methoden

Name	Parameter	Beschreibung
AddBCCRecipient	Recipient als <u>String</u>	Fügt einen BCC-Empfänger der E-Mail hinzu.
AddCCRecipient	Recipient als <u>String</u>	Fügt einen CC-Empfänger der E-Mail hinzu.
AddRecipient	Recipient als <u>String</u>	Fügt einen Empfänger der E-Mail hinzu.

Beispiel

Folgendes Beispiel setzt die Werte von From Address, Subject, Body und Headers mit den Werten der Text-Eigenschaften von EditFields eines Fensters. Die EditFields namens fromAddressFld, subjectFld, bodyFld und htmlFld enthalten diese Informationen:

```
Dim mail as EmailMessage
mail = New EmailMessage
mail.fromAddress = fromAddressFld.text
mail.subject = subjectFld.text
mail.bodyPlainText = bodyFld.text
mail.bodyHTML = htmlFld.text
```

Siehe auch das Beispiel für die SMTPSocket-Klasse.

Siehe auch

[EmailHeaders](#), [EmailAttachment](#), [HTTPSocket](#), [HTTPSecureSocket](#), [POP3Socket](#), [POP3SecureSocket](#), [SMTPSocket](#), [SMTPSecureSocket](#), [TCPSocket](#), [SSLSocket](#), [SocketCore](#) Klassen.

EnableMenuItems Methode

Erzwingt einen Neuaufbau der Menüs und startet einen Event „EnableMenuItems“.

Syntax

EnableMenuItems

Hinweis

Meistens reicht die Automatik von REALbasic vollkommen aus, um eine Menüleiste korrekt darzustellen. In einigen Fällen greift diese Automatik allerdings nicht. Wenn Sie z.B. aus einem SelChange-Event eines [EditField](#) heraus alle Menüpunkte eines Menüs deaktivieren, sollte auch das gesamte Menü deaktiviert werden. Dies geschieht leider nicht, da der Anwender nicht in die Menüleiste geklickt hat. In solchen Fällen können Sie mit der **EnableMenuItems** Methode REALbasic auffordern, den Status aller Menüs zu überprüfen und sie gegebenenfalls neu zu zeichnen.

EncodeBase64 Funktion

Konvertiert einen String ins Base64-Format.

Syntax

Ergebnis = **EncodeBase64**(str) [*lineWrap*]

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Die Base64-Repräsentation von <i>str</i> .
<i>str</i>	String	Der String -Ausdruck, der ins Base64-Format konvertiert werden soll.
<i>lineWrap</i>	Integer	Optionaler Parameter, der die maximale Anzahl der Zeichen einer Zeile angibt. Standardmäßig ist dieser auf 76 gesetzt, daß der Zeilenlänge entspricht, die in Base64 kodierten MIME-Daten verwendet wird. Wenn Sie hier 0 angeben, wird EncodeBase64 keinen Zeilenumbruch einfügen.

Hinweise

Base64 wird für E-Mail-Anhänge verwendet. SMTP wurde auf Basis von ASCII entwickelt, daher geht es davon aus, daß ASCII-Zeichen unter 32 Steuerzeichen sind und es erlaubt nichts über ASCII 127. Mit Base64 können Sie beliebige Binärdaten in ein 64-Zeichen-Alphabet, zusammengesetzt aus den druckbaren ASCII-Zeichen, übersetzen. Aus drei Bytes, die man eingibt werden hierbei für die Ausgabe vier Bytes. Base64 wird außerdem bei Authentifizierung als eine Möglichkeit verwendet, einen Verschlüsselungs-Schlüssel oder einen Hash-Wert über einen Link, der Text erwartet, zu versenden.

Die [DecodeBase64](#)-Funktion führt die umgekehrte Operation aus, sie erwartet einen Base64-Ausdruck und konvertiert ihn in die Original-Daten.

Beispiel

Das folgende Beispiel kodiert UTF-8 Text für eine Übertragung ohne Zeilenumbrüche.

```
Dim encodedText as String
encodedText = EncodingBase64(utf8text, 0)
```

Siehe auch

[DecodeBase64](#) Funktion.

EncodeQuotedPrintable Function

Konvertiert einen [String](#) ins Quoted-Printable-Format.

Syntax

Ergebnis = **EncodeQuotedPrintable**(str)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Das Ergebnis der Konvertierung von <i>str</i> .
<i>str</i>	<u>String</u>	Der zu konvertierende String-Ausdruck.

Hinweise

EncodeQuotedPrintable konvertiert undruckbare Zeichen (das sind Steuerzeichen, Returns und Tabs) ins Hexadezimale. Es wurde für die Verarbeitung von E-Mails und Usenet-News entwickelt.

Die DecodeQuotedPrintable-Funktion führt die umgekehrte Operation durch.

Siehe auch

DecodeQuotedPrintable Funktion.

Encoding Funktion

Liefert das Text-Encoding des übergebenen Strings.

Syntax

Ergebnis = **Encoding**(str)

oder

Ergebnis = str.**Encoding**

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>TextEncoding</u>	Das Text-Encoding von <i>str</i> .
<i>str</i>	<u>String</u>	Der <u>String</u> , dessen <u>TextEncoding</u> zurückgeliefert werden wird.

Hinweise

Benutzen Sie die Encoding-Funktion, wenn Sie die Kodierung eines Strings ermitteln wollen.

Wenn die Kodierung des Strings unbekannt ist, liefert **Encoding** Nil zurück. Testen Sie, ob das TextEncoding-Objekt Nil ist oder fügen Sie einen Exception Block hinzu, wenn die Möglichkeit besteht, daß die Kodierung des Strings zur Laufzeit nicht bekannt ist.

Beispiel

Folgendes Beispiel liefert die Kodierung des Textes in einem EditField. Der Exception Block behandelt den Aufruf auf eine Eigenschaft des TextEncoding-Objekts, wenn die Kodierung nicht ermittelt werden kann.

```
Dim t as TextEncoding
t=Encoding(EditField1.text)
t=EditField1.Text.Encoding
MsgBox "Der Text-Encoding-Code ist "+Str(t.Base)
Exception err as NilObjectException
MsgBox "Das Text-Encoding ist nicht bekannt"
```

Siehe auch

TextEncoding, TextConverter Klassen, ConvertEncoding, DefineEncoding, GetTextEncoding Funktionen; Encodings Objekt.

Encodings Objekt

Liefert das festgelegte TextEncoding.

Syntax

Ergebnis = **Encodings.EncodingName**

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>TextEncoding</u>	Das Text-Encoding, das durch <i>EncodingName</i> festgelegt wird.

Teil	Typ	Beschreibung
<i>EncodingsName</i> <i>ORDER</i>	<u>String</u>	Der Name eines <u>TextEncoding</u> . Siehe Anhang A der gedruckten oder PDF-Version der Sprachreferenz, oder verwenden Sie bevorzugt die Autovervollständigen-Funktion des Editors, um alle möglichen Werte von <i>EndodingsName</i> anzuzeigen.
<i>GetFromCode</i>		Parameter ist <i>Code</i> als <u>Integer</u> und liefert ein <u>TextEncoding</u> -Objekt. Werte für <i>Code</i> sind die von <u>TextEncoding</u> verwendeten. Hiermit hat man eine bequeme Möglichkeit, ein <u>TextEncoding</u> in einer Datei zu speichern, es über ein Socket oder den seriellen Anschluß und so weiter zu versenden.

Hinweise

Das **Encodings**-Objekt macht es einfach, ein bestimmtes TextEncoding zu erhalten. Jedes Text-Encoding kann über das **Encodings**-Objekt erhalten werden. Einige der Nützlichsten sind: UTF8, UTF16, ASCII, MacRoman, MacJapanese, und WindowsLatin1. Verwenden Sie die Autovervollständigen-Funktion des Code-Editors, um eine komplette Liste anzuzeigen.

Beispiel

Benutzen Sie die Chr-Methode der TextEncoding-Klasse, um bestimmte Zeichen in einer bestimmten Kodierung zu erhalten. Sie verwenden das Encodings-Objekt, um zuerst die gewünschte Kodierung zu bekommen. Zum Beispiel:

```
Dim s as String
s=Encodings.UTF8.Chr(169)
```

Wenn Sie Strings verwenden, die außerhalb von REALbasic erstellt wurden, sollten Sie immer eine Kodierung angeben, damit REALbasic den String korrekt interpretieren kann. Verwenden Sie das Encodings-Objekt, um die Kodierung zu erhalten und weisen Sie diese dann der Encoding-Eigenschaft der TextInputStream-Klasse zu. Dieses Beispiel verwendet die MacCentralEurRoman Kodierung.

```
Dim f As FolderItem
Dim t As TextInputStream
f=GetOpenFolderItem("text")
If f <> Nil then
    t=f.OpenAsTextFile
    t.Encoding=Encodings.MacCentralEurRoman
    EditField1.text=t.ReadAll
    t.Close
End if
```

Sie können die Kodierung auf für den optionalen Parameter der Methoden Read, ReadLine, und ReadAll verwenden.

Siehe auch

TextEncoding Klasse, DefineEncoding, ConvertEncoding, Encoding Funktionen.

EndOfLine Klasse

Liefert ein End-Of-Line-Zeichen (Zeilenende).

Übergeordnete Klasse

Object

Syntax

result=EndOfLine

Teil	Typ	Beschreibung
result	<u>String</u>	Liefert ein Zeilenende-Zeichen der Plattform, auf der gerade kompiliert wird.

oder

result=EndOfLine.EndOfLineType

Teil	Typ	Beschreibung
result	<u>String</u>	Zeilenende-Zeichen vom angegebenen Typ <i>EndOfLineType</i> .

Teil	Typ	Beschreibung
EndOfLineType		Gibt an, welches Zeilenende-Zeichen benötigt wird. Zur Auswahl stehen: Macintosh Windows Unix

Eigenschaften

Name	Typ	Beschreibung
Macintosh	<u>String</u>	Ein Macintosh-Zeilenende, <u>Chr</u> (13).
Windows	<u>String</u>	Ein Windows-Zeilenende, <u>Chr</u> (13) + <u>Chr</u> (10).
Unix	<u>String</u>	Ein Unix-Zeilenende, <u>Chr</u> (10).

Hinweise

Wie die Syntax schon andeutet, kann man **EndOfLine** auf zwei Arten verwenden. Sie können eine der Eigenschaften benutzen oder Sie verwenden die Methoden. Wenn Sie EndOfLine direkt aufrufen, bekommen Sie das Zeilenende der Plattform, auf der Sie die Applikation kompiliert haben. Für Konsolen-Applikationen unter Mac OS X und Linux liefert EndOfLine ein Unix-Zeilenende.

Verwenden Sie die ReplaceLineEndings Funktion, um zwischen verschiedenen Zeilenenden zu konvertieren.

Einige Applikation unter Mac OS X benötigen ein Unix-Zeilenende, da Mac OS X auf BSD Unix basiert.

Beispiele

Das folgende Beispiel verwendet die **EndOfLine** Function, um einen Text in zwei Zeilen zu teilen, der in einer MsgBox angezeigt wird. Wenn Sie **EndOfLine** so benutzen, wird die zweite Zeile weniger hervorgehoben als die erste.

```
MsgBox "Hallo Welt"+EndOfLine+EndOfLine+"Wie geht es Dir?"
```

Das folgende Beispiel verwendet ein Unix-Zeilenende.

```
Dim cr as String
cr=EndOfLine.Unix
Editfield1.Text ="Hallo Welt"+cr+"Wie geht es Dir?"
```

Siehe auch

ReplaceLineEndings Funktion.

Es gibt keine Klasse mit diesem Namen Fehler

Sie haben einen nicht existierenden Klassennamen verwendet.

Beispiele

Es wurde ein nicht existierender Klassenname bei einem Dim-Befehl verwendet.

```
Dim Fred as Ehemann //'Ehemann' ist kein Datentyp
Dim f as Folder //müßte FolderItem heißen
Dim d as longdouble //so etwas gibt es nicht
```

Es wurde eine nicht existierende Klasse mit dem New-Operator verwendet:

```
Dim f as FolderItem
f=New Folder
```

Siehe auch

Dim Befehl; New Operator.

ExcelApplication Klasse

Wird verwendet, um Microsoft Excel zu automatisieren.

Übergeordnete Klasse

OLEObject

Siehe auch

[Office](#), [PowerPointApplication](#), [WordApplication](#), [OLEObject](#) Klassen.

Exception Block

Wird verwendet, um [RuntimeException](#) Errors (Laufzeitfehler) zu verwalten.

Syntax

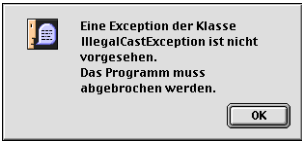
Exception [ErrorParameter] [**As** ErrorType]

Teil	Beschreibung
<i>ErrorParameter</i>	Optional, wird verwendet, um den Typ des Laufzeitfehlers festzulegen.
<i>ErrorType</i>	Optional, falls verwendet, muß er mit <i>ErrorParameter</i> verwendet werden. Wird verwendet, um einen bestimmten Typ eines Laufzeitfehlers abzufangen. Falls verwendet, wird der Exception Block nur diesen Typ von Laufzeitfehler verwalten.

Hinweise

Einer oder mehrere **Exception** Blocks können nach der letzten „regulären“ Code-Zeile eingefügt werden, um Laufzeitfehler abzufangen und zu verwalten, die irgendwo innerhalb der Methode auftreten können.

Wenn eine Runtime Exception in der erzeugten Applikation auftritt und nicht verwaltet wird, zeigt REALbasic eine eigene Laufzeitfehlermeldungsbox an und beendet die Applikation.



Exception Blocks bieten eine Möglichkeit, Fehler eleganter zu behandeln.

Exception Blocks erscheinen immer am Ende einer Methode (nicht dort, wo Sie meinen, daß der Fehler auftritt), da jede Zeile nach der Zeile mit **Exception** ein Teil des Exception Blocks ist. Im Code-Editor hat die Zeile mit **Exception** die gleiche Einrückungsebene wie die Zeile mit [Sub](#) oder [Function](#).

Sie können **Exception** alleine verwenden, wenn Sie jeden Exception-Typ im Exception Block verwalten wollen, wie unten gezeigt:

```
Sub...
.
.
Exception
  MsgBox "Irgendwas ganz schlimmes ist passiert aber ich weiß nicht, was."
End Sub
```

Das oben gezeigte Beispiel ist ausreichend, um zu verhindern, daß das Programm beendet wird, die Nachricht ist aber nicht sehr informativ, da man keinen Hinweis bekommt, welche Art von Exception aufgetreten ist.

Um den Typ des aufgetretenen Laufzeitfehlers festzulegen, müssen Sie *ErrorParameter* verwenden und in manchen Fällen seinen Typ testen. *ErrorParameter* kann einer der folgenden Fehler-Typen sein:

ErrorType	Beschreibung
<i>IllegalCastException</i>	Sie haben versucht, den Typ eines Objekts zu ändern und ihm eine Nachricht zu schicken, das seine wirkliche Klasse nicht annehmen kann.
<i>KeyNotFoundException</i>	Sie haben versucht, auf einen Eintrag in einem Dictionary mit einem Schlüssel zuzugreifen, der in dem Dictionary nicht vorhanden ist.
<i>NilObjectException</i>	Es wurde versucht, auf ein Objekt zuzugreifen, das nicht existiert.
<i>OutOfBoundsException</i>	Es wurde versucht, bei einem Objekt oder Datentyp einen Wert, ein Zeichen oder ein Element zu lesen oder zu schreiben, das sich außerhalb dessen Grenzen befindet, d.h. Sie haben versucht, auf ein Array-Element zuzugreifen, das nicht existiert.

ErrorType	Beschreibung
<i>OutOfMemoryException</i>	Tritt in bestimmten Fällen auf, wenn der Speicher ausgeht, z.B. nicht genug Speicher zur Verfügung steht, um eine Operation zu Ende zu führen.
<i>RegexException</i>	Sie haben in einem regulären Ausdruck eine falsche Syntax verwendet.
<i>RegexSearchPatternException</i>	Sie haben ein unkorrektes Suchmuster in einem regulären Ausdruck festgelegt.
<i>ShellNotRunningException</i>	Sie haben versucht, auf eine interaktive Shell zuzugreifen, als diese nicht lief oder haben versucht, den Kontext des Skriptes während es lief zu ändern.
<i>StackOverflowException</i>	Wenn eine Routine (Methode/Event Handler/Menu Handler) eine andere aufruft, wird Speicher benötigt, um zu der Stelle zurückfinden zu können, an der die Routine aufgerufen wurde und die Werte der lokalen Variablen müssen ebenfalls zwischengespeichert werden. Der Grund dafür ist (wenn die aufgerufene Routine beendet wurde), daß man zur vorigen Routine zurückkehren können muß und alle lokalen Variablen den gleichen Wert wie zuvor haben müssen. Der Speicher, der hierfür verwendet wird, nennt sich Stack (Stapel) (da Sie eine Routine auf der anderen stapeln). Wenn Ihrer Applikation der Stack-Speicher ausgeht, tritt eine <u>StackOverflowException</u> auf.
<i>TypeMismatchException</i>	Sie haben versucht, einem Objekt einen falschen Datentyp zuzuweisen. Tritt nur auf, wenn Sie ein Objekt verwenden, das REALbasic zur Compilierzeit nicht erkennen kann.
<i>UnsupportedFormatException</i>	Sie haben eine unkorrekte Syntax in einem Ausdruck verwendet, z.B. für die Spaltenbreite in einer <u>ListBox</u> , oder Sie haben versucht, ein nicht unterstütztes Format beim Laden oder Speichern von Bildern zu verwenden.

Wenn Sie die Office Automation Plugins installieren, fügen Sie außerdem die Typen WordException, ExcelException und PowerPointException hinzu. Andere optionale Plugins können ebenfalls ihre eigene Exception-Typen hinzufügen.

Eine Möglichkeit, ErrorParameter zu testen, ist mit einem If-Befehl im Exception Block:

```
Sub...
.
.
Exception err
  If err IsA TypeMismatchException then
    MsgBox "Sie haben versucht, den Typ eines Objekts zu ändern!"
  elseif err IsA NilObjectException then
    MsgBox "Sie haben versucht, auf ein Nil-Objekt zuzugreifen!"
  .
  .
  End if
End Sub
```

Statt daß Sie mehrere If-Befehle verwenden, können Sie mehrere Exception Blocks verwenden, von denen jeder einen anderen Typ von Laufzeitfehler behandelt.

```
Sub...
.
.
Exception err as TypeMismatchException
  MsgBox "Sie haben versucht, den Typ eines Objekts zu ändern!"
Exception err as NilObjectException
  MsgBox "Sie haben versucht, auf ein Nil-Objekt zuzugreifen!"
End Sub
```

Beispiele

Sehen Sie bei den Beispielen für die Fehler NilObjectException, IllegalCastException, TypeMismatchException, OutOfBoundsException und StackOverflowException nach.

Siehe auch

Function Befehl, Raise Befehl, NilObjectException, IllegalCastException, KeyNotFoundException, TypeMismatchException, OutOfBoundsException, OutOfMemoryException, RegexException, RegexSearchPatternException, StackOverflowException, ShellNotRunningException, UnsupportedFormatException Fehler; RuntimeException Klasse; Nil Ausdruck.

Exception Objekte müssen vom Typ RuntimeException sein Fehler

Sie haben eine Exception-Variable als etwas anderes als eine RuntimeException oder eine Unterklasse von RuntimeException deklariert.

Beispiel

Folgende Deklaration produziert diesen Fehler:

```
Exception error as Double
```

Exit Befehl

Der Exit-Befehl beendet die Ausführung einer Schleife und führt dazu, daß der der Schleife folgende Befehl ausgeführt wird, obwohl die in der Schleife angegebenen Bedingungen nicht erfüllt sind.

Syntax

Exit

Beispiel

Dieses Beispiel erhöht eine Zählervariable. Wenn der Anwender gleichzeitig die Befehlstaste und den Punkt, unter Macintosh, oder ESC unter Windows drückt, erhält die UserCancelled Funktion den Wert True und die Schleife wird verlassen. Andernfalls wird die Schleife ausgeführt, bis die Stop-Eigenschaft den Wert True annimmt:

```
Dim counter as Integer
Do
  counter=counter+1
  If UserCancelled then
    Exit
  End if
Loop Until Stop
```

Siehe auch

Do...Loop , For...Next , While...Wend, Goto Befehl.

Exp Funktion

Rechnet die Exponentialfunktion „e“ mit dem angegebenen Wert als Exponenten.

Syntax

Ergebnis=**Exp** (Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Das Ergebnis der Exponentialfunktion „e“ mit dem Exponenten <i>Wert</i> .
<i>Wert</i>	<u>Double</u>	Der Wert, mit dem Sie rechnen möchten.

Hinweis

Die **Exp** Funktion liefert das Ergebnis der Exponentialfunktion „e“ hoch des angegebenen Wertes zurück.

Beispiele

Dieses Beispiel verwendet die **Exp** Funktion, um e^{10} zu berechnen.

```
Dim d as double
d=Exp(10) //ergibt 22026.4657948
```

ExportPicture Funktion

Exportiert ein Objekt vom Typ Picture (Bild).

Syntax

result=**ExportPicture**(pic)

Teil	Typ	Beschreibung
<i>result</i>	<u>Boolean</u>	True, wenn das Picture erfolgreich gespeichert werden konnte; <u>False</u> , wenn das Bild nicht gespeichert wurde.
<i>pic</i>	<u>Picture</u>	Ein Picture.

Hinweis

Wenn **ExportPicture** aufgerufen wird, wird eine „Sichern unter“-Dialogbox angezeigt, damit der Benutzer das Bild in einem beliebigen Verzeichnis speichern kann. Der Benutzer kann außerdem in der Dialogbox ein bestimmtes Bildformat auswählen. Zur Auswahl stehen: BMP, PICT, PhotoShop, JPEG, PNG, SGI, TGA, TIFF und QuickTime-Image. Wenn der Benutzer ein Format wählt, fügt REALbasic die passende Endung an den Dateinamen an.

Die Formate PICT und BMP werden auf dem Macintosh beziehungsweise unter Windows unterstützt. Für alle anderen Formate wird QuickTime benötigt.

ExportPicture liefert True, wenn das Bild gespeichert wurde; Wenn der Benutzer auf Abbruch klickt, liefert ExportPicture False.

Beispiel

Der folgende Code erlaubt es dem Benutzer, ein Picture zu speichern.

```
Dim p as Picture
Dim r as Boolean
Dim f as FolderItem
f=GetFolderItem("IconView")
p=f.OpenAsPicture
r=ExportPicture(p)
If r then
    MsgBox "Das Bild wurde gesichert."
else
    MsgBox "Das Bild wurde nicht gesichert."
end
```

Siehe auch

Picture, FolderItem Klassen.

Extends Schlüsselwort

Wird in einer Methoden-Deklaration verwendet, um anzuzeigen, daß die Methode unter Verwendung der Syntax des Objektes aufgerufen werden muß, also so, als würde die Methode zu dem Objekt gehören.

Syntax

Extends Parameter As Object

Teil	Typ	Beschreibung
<i>Parameter</i>		Der Name des ersten Parameters.
Object	Beliebiger gültiger Objekttyp	Der Objekttyp, von dem die neue Methode aufgerufen wird.

Hinweise

Das **Extends**-Schlüsselwort erlaubt Ihnen, eine selbstdefinierte Methode so aufzurufen, als würde sie zu einem Objekt gehören. Sie verwenden das Extends-Schlüsselwort nur für den ersten Parameter in der Methoden-Deklaration. **Extends** zeigt an, daß dieser Parameter auf der linken Seite des Punkt-Operators (".") verwendet werden muß.

Sie können Standardwerte für einen Aufruf angeben, solange Sie dem „Extend“-Parameter selbst keinen Standardwert zuweisen.

Methoden, die auf diese Weise deklariert wurden, werden manchmal auch „Class-Extensions-Methoden“ genannt. Sie können **Extends** nur für Methoden in einem Modul verwenden.

Extends können keine anderen Methoden überladen, sondern lediglich eine Klasse erweitern. Erweiternde Methoden können nicht virtuell sein, da sie kein tatsächlicher Bestandteil der Klasse sind.

Beispiel

Um eine neue Methode, myNewFolderItemMethod als eine Methode der FolderItem-Klasse aufzurufen, verwenden Sie eine Deklaration wie diese:

```
Sub myNewFolderItemMethod(Extends f as FolderItem)
```

Sie können diese im Code dann so aufrufen:

```
Dim f as FolderItem  
f.myNewFolderItemMethod
```

Siehe auch

Extends kann nur auf den ersten Parameter angewendet werden Fehler.

Extends kann nur auf den ersten Parameter angewendet werden Fehler

Sie haben das Extends-Schlüsselwort mehr als einmal und/oder auf einen anderen als den ersten Parameter angewendet.

Beispiel

```
Sub myNewMethod(x as Integer, Extends f as FolderItem)
```

Sollte folgendermaßen umgeschrieben werden:

```
Sub myNewMethod(Extends f as FolderItem, x as Integer, )
```

Siehe auch

Extends Schlüsselwort.

ExtensionsFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Ordner „Systemerweiterungen“ des Systemordners zu bekommen. Unter Windows greift es auf das Verzeichnis WINDOWS\SYSTEM zu.

Syntax

Ergebnis=**ExtensionsFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein FolderItem, das die Objekte des Systemerweiterungen-Ordners enthält.

Hinweis

Verwenden Sie die **ExtensionsFolder**-Funktion immer dann, wenn Sie auf Objekte aus dem Systemerweiterungen-Ordner zugreifen müssen.

Die Funktion **ExtensionsFolder** bietet eine Möglichkeit, auf die Einträge im Ordner „Systemerweiterungen“ zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

Unter Linux liefert diese Funktion Nil zurück.

Beispiele

Bei der Beschreibung der Funktion AppleMenuFolder finden Sie ein passendes Beispiel.

Siehe auch

AppleMenuFolder, ApplicationSupportFolder, ControlPanelsFolder, ExtensionsFolder, FontsFolder, PreferencesFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

Externe Funktionen können Objekte nicht als Parameter verwenden Fehler

Dieser Fehler tritt auf, wenn Sie eine externe Funktion mit Declare deklarieren und versuchen festzulegen, daß einer der Parameter ein Objekt ist.

Wenn Sie an ein Declare einen MemoryBlock übergeben wollen, verwenden Sie stattdessen Ptr. Wenn Sie an Declare ein Window übergeben, verwenden Sie stattdessen WindowPtr.

Siehe auch

Declare Befehl, MemoryBlock Klasse.

Externe Funktionen können String-Datentypen nicht als Parameter verwenden Fehler

Tritt auf, wenn einer der Parameter Ihrer Declare-Befehle ein normaler REALbasic-String ist. Verwenden Sie stattdessen CString oder PString.

Siehe auch

Declare Befehl.

Falscher Typ Fehler

Tritt auf, wenn Sie versuchen, einen Parameter mit falschem Datentyp zu übergeben oder in einem Zuweisungsbefehl einen falschen Datentyp verwenden. Die zweite Zeile der Meldung sagt Ihnen, welchen Datentyp REALbasic erhalten hat und welchen es erwartete.

Beispiele

Es wurde versucht, einem String eine Integer-Variable zuzuweisen:

```
Dim n as Integer
n="Anton"
```

Die zweite Zeile der Meldung wird sein: „Erwartete Integer, erhielt String“. Hier wurde nun versucht, einen Wert einem Array zuzuweisen und nicht einem Element des Arrays.

```
Dim truth(3) as Boolean
truth=True //benötigt einen Index
```

Die zweite Zeile der Meldung wird sein: „Erwartete Boolean(), erhielt Boolean“.

False Schlüsselwort

Wird für Variablen vom Typ Boolean verwendet. False ist das Ergebnis eines Vergleichs, bei dem sich die beiden Objekte unterscheiden.

Syntax

Ausdruck=**False**

Ausdruck ist ein beliebiger, gültiger boole'sche Ausdruck.

Siehe auch

True Schlüsselwort, And, Or, Not Operatoren.

FigureShape Klasse

Wird zum Zeichnen von Vektorgrafik-Figuren verwendet, die aus Linien und Kurven zusammengesetzt sind. Diese sind Polygonen ähnlich, es kann jedoch eine „Seite“ auch gebogen sein. Eine Figur besteht aus einem Set von CurveShapes, wobei der Endpunkt einer Kurve automatisch mit dem Startpunkt der nächsten verbunden ist.

Übergeordnete Klasse

Object2D

Eigenschaften

Name	Typ	Beschreibung
<i>Count</i>	<u>Integer</u>	Die Anzahl der CurveShapes.
<i>Item</i>	<u>CurveShape</u>	Parameter ist Index als <u>Integer</u> . Ermittelt oder setzt die festgelegte Kurve.

Methoden

Name	Parameter	Beschreibung
<i>AddCubic</i>	X als Integer	Fügt der Figur eine kubische Kurve hinzu. Die Start- und Endpunkte sind x,y und x2,y2. Dies ist einfacher, als eine CurveShape anzufügen. CX, CY, CX2, und CY2 sind die Kontrollpunkte, Control X(i) und Control Y(i), der CurveShape .
	Y als Integer	
	X2 als Integer	
	Y2 als Integer	
	CX als Integer	
	CY als Integer	
	CX2 als Integer	
<i>AddLine</i>	Y2 als Integer	Fügt der Figur eine gerade Linie hinzu. X und Y sind die Start-Koordinaten; X2 und Y2 sind die End-Koordinaten.
	X als Integer	
	Y als Integer	
	X2 als Integer	
<i>AddQuad</i>	Y2 als Integer	Fügt eine quadratische Kurve der Figur hinzu. Die Start- und Endpunkte sind x,y und x2, y2. Dies ist einfacher, als eine CurveShape anzufügen. CX, CY, CX2, und CY2 sind die Kontrollpunkte Control X(i) und ControlY(i) der CurveShape .
	X als Integer	
	Y als Integer	
	X2 als Integer	
	Y2 als Integer	
	CX als Integer	
<i>Append</i>	CY als Integer	Fügt die übergebene CurveShape der Figur hinzu.
	Curve als CurveShape	
<i>Insert</i>	Index als Integer	Fügt die übergebene CurveShape an der durch Index festgelegten Position ein.
	Curve als CurveShape	
<i>Remove</i>	Index als Integer ODER	Entfernt eine Kurve, festgelegt entweder durch ihren Index oder durch Referenz.
	Curve als CurveShape	

Hinweise

Eine Figur wird durch Nacheinander-Zeichnen jeder Kurve, aus der sie besteht, geformt, wobei der Endpunkt der einen mit dem Startpunkt der nächsten verbunden wird. Die Figur ist degeneriert, wenn kein Bereich von ihr eingeschlossen wird. Dies ist z.B. der Fall bei einer Figur, die nur aus einer einzelnen Linie besteht. Das Erscheinungsbild einer degenerierten Figur ist undefiniert.

AddLine, AddCubic und AddQuad sind bequeme Methoden, die es einfacher machen, einer Figur Kurven hinzuzufügen. Sie können stattdessen auch Ihre eigenen CurveShapes erzeugen und diese mit den Methoden Append oder Insert hinzufügen.

Beispiel

Das folgende Beispiel zeichnet ein Dreieck. Dieser Code steht in dem Paint-Event eines [Window](#)- oder [Canvas](#)-Objekts.

```
Sub Paint (g as Graphics)
    Dim fx as New FigureShape
    fx.AddLine 0, 100, 50, 0
    fx.AddLine 50, 0, -50, 0
    fx.Border = 100 // opaque border
    fx.BorderColor = &cFF0000 // red border
    fx.FillColor = &cFFFF00 // yellow interior
    g.DrawObject fx, 100,100
End Sub
```

Siehe auch

[ArcShape](#), [CurveShape](#), [FolderItem](#), [Group2D](#), [Graphics](#), [Object2D](#), [OvalShape](#), [Picture](#), [PixmapShape](#), [RectShape](#), [RoundRectShape](#), [StringShape](#) Klassen.

FillColor Funktion

Liefert die momentan gewählte Hintergrundfarbe des Erscheinungsbild-Managers.

Syntax

Ergebnis=**FillColor**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Die für das Zeichnen von Hintergründen von Steuerelementen und Fenstern verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit Canvas Steuerelementen wichtig, um deren Aussehen mit denen des Erscheinungsbild-Managers gleich zu halten. Wenn Sie Objekte zeichnen, sollten Sie diese Farbe als Hintergrundfarbe definieren.

Dieser Wert kann vom Anwender geändert werden, weshalb Sie in jedem Zeichen-Event diesen Wert neu abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

DarkBevelColor, DarkTingeColor, FrameColor, LightBevelColor, LightTingeColor, HighlightColor, TextColor Funktionen; Color Datentyp.

Floor Funktion

Rundet einen Wert auf den nächsten Integer-Wert ab.

Syntax

Ergebnis=**Floor** (Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der gerundete Wert von <i>Wert</i> .
<i>Wert</i>	<u>Double</u>	Der Wert, den Sie abrunden lassen möchten.

Hinweis

Die Funktion **Floor** rundet einen Wert ab.

Beispiel

Dieses Beispiel rundet mit der Funktion **Floor** eine Zahl zum nächsten Integer ab.

```
Dim d as Double
d=Floor(1.234) //ergibt 1
```

Siehe auch

Ceil, Round Funktionen.

FolderItem Klasse

Objekte der Klasse **FolderItem** verweisen auf Dateien, Ordner oder Programme. Sie werden z.B. mit der Methode GetFolderItem erzeugt.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i><u>AbsolutePath</u></i>	<u>String</u>	Der komplette Pfad des FolderItem. Auf dem Macintosh endet der Pfad mit einem abschließenden Doppelpunkt; unter Windows gibt es keinen abschließenden Backslash. Wenn Sie auf ein Laufwerk unter Windows zugreifen, für das Sie keine Zugriffsrechte haben oder das nicht existiert, hat AbsolutePath kein abschließenden Backslash. Wenn „a:“ ein Floppylaufwerk ist, in dem keine Disk ist, ist bei „a:\“ eine Disk darin.
<i><u>Alias</u></i>	<u>Boolean</u>	Liefert <u>True</u> , wenn das Item ein Alias ist (nur auf dem Macintosh). Unter Windows liefert <i>Alias</i> immer <u>False</u> .

Name	Typ	Beschreibung
<i>Count</i>	<u>Integer</u>	Die Anzahl der Objekte in einem FolderItem (wenn es sich um ein Verzeichnis handelt).
<i>CreationDate</i>	<u>Date</u>	Das Erzeugungsdatum des FolderItem.
<i>DesktopFolder</i>	<u>FolderItem</u>	Liefert ein FolderItem, das auf den Schreibtisch-Ordner des Eltern-Volumes verweist. Unter Windows liefert es den Schreibtisch-Ordner des derzeitigen Anwenders.
<i>Directory</i>	<u>Boolean</u>	Das FolderItem ist ein Verzeichnis (Ordner).
<i>DisplayName</i>	<u>String</u>	Der Name des FolderItem, wie es vom Anwender gesehen wird. Normalerweise das gleiche wie die Name-Eigenschaft. Ab Mac OS X 10.1 kann es der Name ohne Suffix sein, je nachdem, ob der Anwender dies für das FolderItem ausgeblendet hat. Verwenden Sie unter Mac OS X statt Name eher DisplayName, wenn Sie den Namen des Objektes dem Anwender anzeigen.
<i>ExtensionVisible</i>	<u>Boolean</u>	Hiermit kann man ermitteln, ob die Anzeige des Suffix ein- oder ausgeschaltet ist und man kann dieses damit umschalten. Funktioniert derzeit nur ab Mac OS X 10.1. Beachten Sie, daß wenn Sie diese Einstellung ändern, der Finder oder andere Navigations-Dialoge diese Änderung nicht sofort anzeigen. Wenn Sie unter Mac OS X eine neue Datei erzeugen, setzt REALbasic das Flag „Suffix ausblenden“ passend zum aktuellen Wert dieser Eigenschaft.
<i>Exists</i>	<u>Boolean</u>	Ermittelt, ob die Eigenschaft <i>AbsolutePath</i> auf eine existierende Datei (oder Verzeichnis) verweist.
<i>IsReadable</i>	<u>Boolean</u>	True, wenn es möglich ist, das FolderItem auszulesen.
<i>IsWritable</i>	<u>Boolean</u>	True, wenn es möglich ist, in das FolderItem zu schreiben.
<i>LastErrorCode</i>	<u>Integer</u>	Enthält den Fehlercode der letzten unterstützten Operation mit dem FolderItem. Die unterstützten Operationen sind: - CopyFileTo - MoveFileTo - Delete - Methoden, die eine Datei oder einen Ordner erzeugen Folgende Fehlercodes können geliefert werden: 100: Ziel existiert nicht. Diesen Fehler bekommen Sie nur bei CopyFileTo und MoveFileTo. 101: Datei nicht gefunden. 102: Zugriff verweigert 103: Kein Speicher mehr 104: Datei wird gerade benutzt 105: Ungültiger Name Sie können feststellen, ob sich ein Fehler ereignet hat, indem Sie den LastErrorCode gegen eine der Fehler-Konstanten testen. LastErrorCode kann außerdem System-spezifische Fehlercodes liefern, falls der Fehler nicht einem der obigen entspricht.
<i>Length</i>	<u>Integer</u>	Die Größe der Data-Fork der Datei. Bei Ordnern ist die Größe Null.
<i>Locked</i>	<u>Boolean</u>	Das FolderItem ist geschützt oder befindet sich auf einem geschützten Volume.
<i>MacCreator</i>	<u>String</u>	Dies ist bei Dokumenten der 4-buchstabe Code, der die Applikation bestimmt, die gestartet wird und das FolderItem lädt, wenn dieses vom Schreibtisch aus geöffnet wird. Weisen Sie den 4-buchstabigen String dieser Eigenschaft zu. Wenn das FolderItem auf ein Verzeichnis unter MacOS verweist, liefert MacCreator „MACS“. Dabei wird die MacRoman-Kodierung verwendet.
<i>MacDirID</i>	<u>Integer</u>	Macintosh-Verzeichnis ID.

Name	Typ	Beschreibung
<i>MacType</i>	<u>String</u>	Der 4-buchstabile Code, der den Dateityp angibt. Zum Beispiel ist „APPL“ der Typ aller Applikationen. „TEXT“ ist der Dateityp der meisten Textdokumente. Weisen Sie den 4-buchstabigen String dieser Eigenschaft zu. Wenn das FolderItem auf ein Verzeichnis unter MacOS verweist, liefert MacType „fold“. Dabei wird die MacRoman-Kodierung verwendet.
<i>MacVRefNum</i>	<u>Integer</u>	Mac-Volume-Referenz-Nummer.
<i>ModificationDate</i>	<u>Date</u>	Das Änderungsdatum des FolderItem.
<i>Name</i>	<u>String</u>	Der Name des FolderItem. Wenn Sie <i>Name</i> ändern, ändern Sie den Datei- oder Ordnernamen. Unter Windows können Sie mit „Name“ nicht ein geöffnetes FolderItem umbenennen. Siehe Hinweise.
<i>Parent</i>	<u>FolderItem</u>	Das FolderItem-Objekt des Eltern-Ordners dieses Items. Es ist <u>Nil</u> , wenn sich das Item im Wurzelverzeichnis (Root) befindet.
<i>ResourceForkLength</i>	<u>Integer</u>	Die Anzahl an Bytes, die die Resource-Fork der Datei umfaßt. Zugriff auf den Resource-Fork wird nur auf dem Mac unterstützt.
<i>SharedTrashFolder</i>	<u>FolderItem</u>	Liefert ein FolderItem, das auf den Shared-Trash-Ordner (gemeinsamer Papierkorb) verweist, der zum Eltern-Volume gehört.
<i>Temporary Folder</i>	<u>FolderItem</u>	Liefert ein FolderItem, das auf den Temporary-Ordner verweist, der zum Eltern-Volume gehört.
<i>TrashFolder</i>	<u>FolderItem</u>	Liefert ein FolderItem, das auf den Trash-Ordner (Papierkorb) verweist, der zum Eltern-Volume gehört.
<i>Type</i>	<u>String</u>	Wenn das FolderItem kein Verzeichnis ist, enthält diese Eigenschaft den Dateityp. Ist das FolderItem ein Verzeichnis (Mac OS), liefert es einen passenden Dateityp, wenn Sie einen erzeugt haben.
<i>VirtualVolume</i>	<u>Virtual-Volume</u>	Wenn sich das FolderItem auf einem VirtualVolume befindet, liefert es das VirtualVolume, auf dem die Datei gespeichert ist. Wenn es eine echte Datei ist, liefert es Nil.
<i>Visible</i>	<u>Boolean</u>	True, wenn das FolderItem sichtbar ist, False, wenn es unsichtbar ist.

Methoden

Name	Parameter	Beschreibung
<i>AppendToTextFile</i>		Liefert einen TextOutputStream. Hängt Text an eine Textdatei an. Existiert die Datei nicht, so wird sie erzeugt.
<i>Child</i>	Name als <u>String</u>	Wenn dies ein Ordner ist, liefert Child ein FolderItem-Objekt für das Item in diesem Verzeichnis mit dem angegebenen Namen zurück. Aliase werden automatisch aufgelöst (Macintosh). Unter Windows löst <i>Child</i> Aliase nicht auf.
<i>CopyFileTo</i>	Objekt als <u>FolderItem</u>	Wenn <i>Objekt</i> ein Ordner ist, wird das FolderItem in <i>Objekt</i> kopiert. Wenn <i>Objekt</i> eine Datei ist und diese bereits existiert, wird das Kopieren abgebrochen. Sie müssen die schon vorhandene Datei dann zuerst löschen. Falls ein Fehler auftritt, enthält die LastErrorCode-Eigenschaft einen Fehlercode.
<i>CreateAsFolder</i>		Erzeugt einen Ordner an der in den Eigenschaften des FolderItems angegebenen Stelle.
<i>CreateBinaryFile</i>	Dateityp als <u>String</u>	Schreibt eine Binärdatei mit dem angegebenen Dateityp. Dateityp muß der Name eines Dateityps aus dem Dateitypendialog sein. Gibt einen <u>Binary-Stream</u> zurück.
<i>CreateMovie</i>		Erzeugt ein neues <u>EditableMovie</u> . Liefert ein EditableMovie-Objekt.

Name	Parameter	Beschreibung
<i>CreateResourceFork</i>	Dateityp als <u>String</u>	Erzeugt einen neuen Resourcefork. Falls die Datei bereits existiert, wird der vorhandene Resourcefork gelöscht, andernfalls eine neue Datei mit dem angegebenen Dateityp erzeugt. Dateityp ist der Name eines Dateityps, der im Dateitypen-Dialog erzeugt wurde und nicht ein 4-buchstabiger Macintosh Dateityp.
<i>CreateTextFile</i>	Datei als <u>FolderItem</u>	Erzeugt eine Textdatei an der in den Eigenschaften des FolderItems angegebenen Stelle. Gibt einen <u>TextOutputStream</u> zurück.
<i>CreateVirtualVolume</i>		Erzeugt und formatiert ein neues VirtualVolume an der durch FolderItem festgelegten Position.
<i>Delete</i>		Löscht die durch das FolderItem festgelegte Datei oder das Verzeichnis. Diese Methode entfernt die Datei oder das Verzeichnis unwiederbringlich von dem Volume, auf dem sich das FolderItem befand. Wenn Sie ein Verzeichnis löschen möchten, muß dieses leer sein. Falls ein Fehler auftritt, enthält die <i>LastErrorCode</i> -Eigenschaft einen Fehlercode.
<i>GetSaveInfo</i>	RelativeTo als <u>FolderItem</u> , Modus als <u>Integer</u>	Erlaubt das Speichern von FolderItem-Referenzen ohne sich auf den absoluten Pfad zu verlassen. Liefert einen <u>String</u> . Der zurückgelieferte String stellt einen relativen Pfad dar, der relativ zum in <i>RelativeTo</i> übergebenen Ordner ist. Werte für Modus sind: 0: Das FolderItem sollte in irgendeiner möglichen Weise es zu erzeugen geöffnet sein. Dies ist der Default-Wert. 1: Das FolderItem sollte nur mit Hilfe der relativen Pfadinformation geöffnet werden. 2: Das FolderItem sollte nur mit Hilfe der absoluten Pfadinformation geöffnet werden. Hinweis: Der zurückgelieferte String ist nicht dazu gedacht, menschenlesbar zu sein und jede Modifikation kann ihn nutzlos machen.
<i>GetRelative</i>	Index als <u>Integer</u>	Liefert ein FolderItem. Liefert ein FolderItem, das auf dem übergebenen String basiert. Wenn der String einen relativen Pfad darstellt, ist das aktuelle FolderItem als sein Referenzpunkt gedacht. Wenn der übergebene String ein absoluter Pfad ist, wird das aktuelle FolderItem beim Auflösen des Pfades ignoriert. GetRelative liefert nur <u>Nil</u> , wenn es in <i>SaveInfo</i> nicht genug Informationen gibt, um ein FolderItem zu konstruieren (z.B., wenn ein relativer Pfad verwendet wird, der dazu führt, daß das Parsen unter das Wurzelverzeichnis geht).
<i>Item</i>	Index als <u>Integer</u>	Wenn das FolderItem ein Verzeichnis ist, ist <i>Index</i> ein Element in einem 1-basierten Array der FolderItems in diesem Verzeichnis. Wenn das FolderItem ein Alias ist, wird dieses automatisch aufgelöst und gibt ein FolderItem der Originaldatei (bzw. Ordner oder Applikation) zurück
<i>Launch</i>	[Activate als Boolean]	Wenn das FolderItem eine Anwendung ist, so wird diese gestartet. Ist FolderItem ein Dokument, so wird das Dokument geöffnet (und ggfs. die dafür notwendige Anwendung gestartet). Der optionale Parameter <i>Activate</i> legt fest, ob die Applikation im Vorder- oder im Hintergrund gestartet werden soll. Der Default-Wert ist True. Wenn Sie False festlegen, versucht REALbasic, die Applikation im Hintergrund zu starten, das funktioniert aber nicht mit allen Applikationen.

Name	Parameter	Beschreibung
<i>MoveFileTo</i>	Destination as <u>FolderItem</u>	Verschiebt das FolderItem in den Pfad, der durch <i>Destination</i> bestimmt wird. Es wird auch dann verschoben und nicht kopiert, wenn sich die Quelldatei auf einem anderen Volume befindet. Falls ein Fehler auftritt, enthält die <i>LastErrorCode</i> -Eigenschaft einen Fehlercode.
<i>OpenAsBinaryFile</i>	ReadWrite als <u>Boolean</u>	Öffnet das FolderItem als Binärdatei. Wenn <i>ReadWrite</i> True ist, kann diese sowohl gelesen als auch beschrieben werden, andernfalls kann aus ihr nur gelesen werden. Gibt einen <i>BinaryStream</i> zurück.
<i>OpenAsMovie</i>		Öffnet das FolderItem als QuickTime-Film und liefert den Film als Movie-Objekt zurück. Kann der Film nicht gelesen werden, so wird <i>Nil</i> zurückgegeben.
<i>OpenAsPicture</i>		Öffnet das FolderItem als PICT und liefert das Bild als Image-Objekt zurück. Kann das Bild nicht gelesen werden, so wird <i>Nil</i> zurückgegeben. Falls QuickTime installiert ist, kann <i>OpenAsPicture</i> alle von QuickTime unterstützten Bilddateien öffnen (JPEG, GIF, etc.). QuickTime ist nicht notwendig, um JPEG-Dateien zu öffnen. Es wird die tatsächliche Auflösung verwendet anstatt 76dpi anzunehmen.
<i>OpenAsSound</i>		Öffnet das FolderItem als Mac OS 7-Sound und liefert den Ton als Sound-Objekt zurück. Unter Windows erkennt <i>OpenAsSound</i> alle Formate, die von QuickTime unterstützt werden. WAV-Dateien werden über DirectSound abgespielt. Dies ermöglicht individuelle Lautstärke und Panning für den einzelnen Ton. Sie können auch mehrere Töne gleichzeitig abspielen. Kann der Ton nicht gelesen werden, so wird <i>Nil</i> zurückgegeben.
<i>OpenAsTextFile</i>		Öffnet das FolderItem als Textdatei und liefert einen <u>TextInputStream</u> zurück.
<i>OpenAsVectorPicture</i>		Öffnet das FolderItem, um es als Vektor-Bild zu lesen. Es versucht sein Bestes, eine PICT-Datei (auf dem Macintosh) oder eine .emf-Datei (Windows) in ein aus Object2D-Objekten zusammengesetztes Objekt zu konvertieren. Die Original-Datei kann Elemente enthalten, für die es kein äquivalentes Object2D-Objekt gibt. REALbasic versucht sein Bestes, diese Objekte umzusetzen, es kann aber zu Informationsverlusten kommen, je nach Charakteristik der Original-Datei. PICTs unterstützen nicht-rotierte Rechtecke, Linien, Ellipsen, RoundRects, Polygone, Text, Pixmaps und Bögen. .emf-Dateien werden in der aktuellen Größe dargestellt. Meistens ist das riesig; Sie werden diese daher meist herunterskalieren, bevor Sie sie anzeigen (<i>pic1.objects.scale=scalingFactor</i>). Eine guter Wert für den Skalierfaktor ist unserer Meinung nach 0.045.
<i>OpenAsVirtualVolume</i>		Versucht, das FolderItem als ein VirtualVolume zu öffnen. Es liefert Nil, falls das FolderItem kein VirtualVolume ist. Es wird im Schreib-/Lese-Modus geöffnet, außer, es ist auf Betriebssystem-Ebene schreibgeschützt. Liefert ein <u>VirtualVolume</u> .
<i>OpenEditableMovie</i>		Öffnet das FolderItem als ein <u>EditableMovie</u> . Wenn der Film nicht gelesen werden kann, wird <i>Nil</i> zurückgegeben.
<i>OpenResourceFork</i>		Öffnet die Resource-Fork des FolderItems. Zugriff auf den Resource-Fork wird nur auf dem Macintosh unterstützt.
<i>OpenResourceMovie</i>	resID als <u>Integer</u>	Öffnet den Film mit <i>resID</i> als ein Movie. Wird nur verwendet, wenn der Film als Moov-Resource gespeichert ist. Verwenden Sie <i>OpenAsMovie</i> für QT 4.0-Dateien (und neuere).

Name	Parameter	Beschreibung
<i>OpenStyledEditField</i>	Field als <u>EditField</u>	Setzt Text mit Textstilen aus dem FolderItem in das angegebene EditField ein.
<i>SaveAsJPEG</i>	Picture als <u>Picture</u>	Speichert das Bild im JPEG-Format. Hängt bei Windows-Dateinamen „.jpg“. Benötigt Quicktime.
<i>SaveAsPicture</i>	Picture als <u>Picture</u> [Format als <u>Integer</u>]	Sichert das Dokument im PICT Format auf dem Macintosh oder im BMP-Format unter Windows und fügt „.bmp“ an den Dateinamen an. QuickTime wird nicht benötigt. Der optionale zweite Parameter legt das Format fest: Wenn man das passende Format festlegt, kann der Anwender Vektordaten im Vektorformat sichern ohne daß diese in eine Bitmap konvertiert werden. Bei den Hinweisen finden Sie eine Beschreibung der möglichen Werte für <i>Format</i> . Macintosh PICT-Dateien werden im „Typ2“-Format gespeichert, das die volle Auflösung der Bilder enthält.
<i>SaveStyledEditField</i>	Field als <u>EditField</u>	Sichert den Text mit Textstilen aus dem EditField in das FolderItem. Wenn Sie einen Dateityp „styledText“ oder „text“ definiert haben, werden dessen Creator/Type-Werte für die zu speichernde Datei verwendet.
<i>TrueChild</i>	Pfadname als <u>String</u>	Wenn Pfadname ein Verzeichnis ist, liefert <i>TrueChild</i> ein FolderItem-Objekt für das Verzeichnis mit dem übergebenen Namen. <i>TrueChild</i> gibt das aktuelle FolderItem zurück, auch wenn es sich um ein Alias handelt.
<i>Trueltem</i>	Index als <u>Integer</u>	Wenn das FolderItem ein Verzeichnis ist, ist <i>Index</i> ein Element eines 1-basiertes Array von FolderItems in diesem Verzeichnis. <i>Trueltem</i> gibt das aktuelle FolderItem zurück, auch wenn es sich um ein Alias handelt.

Hinweise

Verwenden Sie die Volume-Funktion, die Parent-Eigenschaft der **FolderItem**-Klasse und die Child-Methode der **FolderItem**-Klasse, um Pfadnamen zusammenzubauen. Die Volume-Funktion liefert eine Referenz auf jedes Volume, das auf dem Computer aktiviert ist. Geben Sie eine Nummer für das gewünschte Volume an. Null ist das Boot-Volume. Sie können die Anzahl der Volumes mit der Funktion VolumeCount ermitteln. Um ein **FolderItem** für Microsoft Word im Programme-Verzeichnis des Boot-Volumes zu erhalten, können Sie folgenden Code verwenden:

```
Dim f as FolderItem
f = Volume(0).Child("Applications").Child("Microsoft Word")
```

Die GetFolderItem-Funktion kann verwendet werden, um ein **FolderItem** für einen Eintrag im aktuellen Verzeichnis zu erhalten (das Verzeichnis, in dem sich die Applikation oder die IDE befindet, wenn diese aus der IDE gestartet wurde). Geben Sie einfach den Namen des Eintrags an. Das folgende Beispiel liefert ein **FolderItem** für das Verzeichnis „MeineVorlagen“ im aktuellen Verzeichnis:

```
Dim f as FolderItem
f = GetFolderItem("MeineVorlagen")
```

Wenn die Datei oder das Verzeichnis nicht existieren, ist die Exists-Eigenschaft des **FolderItem** auf False gesetzt.

Wenn Sie einen leeren String an GetFolderItem übergeben, wird ein **FolderItem** des Verzeichnisses zurückgegeben, in dem sich die Applikation befindet.

```
Dim f as FolderItem
f = GetFolderItem("")
```

Die Parent-Eigenschaft ermöglicht Ihnen das Navigieren in die nächsthöhere Ebene der Verzeichnishierarchie. Im folgenden Beispiel erhalten Sie ein FolderItem für das Verzeichnis, in dem sich die Applikation befindet:

```
Dim f as FolderItem
f = GetFolderItem("").Parent
```

Mac OS X

Mac OS X basiert auf BSD Unix, welches „/“ als Pfadtrenner verwendet. Da aber REALbasic und alle mit REALbasic erzeugten Programme Carbon-Applikationen unter Mac OS X sind, verwenden diese weiterhin „:“ als Separator. Einzige Ausnahme ist, wenn Sie auf Dateien mit Hilfe der Shell-Klasse zugreifen. In diesem Fall verwenden Sie den gleichen Separator wie Sie ihn in der Kommandozeile verwenden.

Konstruktoren

Wenn Sie ein FolderItem mit dem New-Befehl erzeugen, können Sie den vollständigen oder den relativen Pfad des neuen FolderItems als optionalen Parameter übergeben. Zum Beispiel

```
Dim f as FolderItem
f=New FolderItem("meinDok.txt")
```

legt den Namen des neuen FolderItems fest und dieses befindet sich im gleichen Ordner wie die REALbasic-Applikation (wenn Sie sich in der IDE befinden) bzw. im gleichen Ordner wie das erzeugte Programm.

Sie können auch ein FolderItem erzeugen, ohne irgendwelche Parameter zu übergeben. Dies funktioniert genauso wie wenn man einen leeren Textstring übergibt.

Bild-Formate

Die SaveAsPicture-Methode besitzt einen zweiten Parameter, der es erlaubt, das Format festzulegen, in welchem das Bild gespeichert werden soll. Formate im Bereich von 0 bis 99 sind für Meta-Formate reserviert. Meta-Formate werden verschiedenen konkreten Formaten zugeordnet, basierend auf dem Ziel, den zu speichernden Daten oder anderen Kriterien (die Zuordnung könnte sich in zukünftigen Versionen von REALbasic ändern).

- Die Nummern 100-199 sind für Formate reserviert, die auf mehreren Plattformen unterstützt werden (derzeit keine).
- 200-299 sind für Formate reserviert, die auf dem Macintosh unterstützt werden.
- 300-399 sind für Formate reserviert, die unter Windows unterstützt werden.
- x00-x49 sind Vektor-Formate, während x50-x99 Raster-Formate sind.

MetaFormate

Wert	Kompatibilität	Beschreibung
0	MostCompatible	Das am meisten verwendete Format der Plattform Mac = PICT Win32 = BMP
1	MostComplete	Das für Vektor-Infos am ehesten verwendete Format Mac = PICT Win32 = EMF)
2	Default	DefaultVector oder DefaultRaster, je nach Bilddaten Mac = PICT Win32 vector=EMF Win32 raster= BMP
3	DefaultVector	Das Standard-Vektorformat der Plattform Mac = PICT Win32 = EMF
4	DefaultRaster	Das Standard-Raster-Format der Plattform Mac = Raster PICT Win32 = BMP

Macintosh-Formate

Wert	Format	Beschreibung
100	PICT	Enthält einfache Vektor-Daten.
250	RasterPICT	Packt alle Vektor-Daten in Pixel.

Win32-Formate

Wert	Format	Beschreibung
300	WMF	Windows Metafile-Format (altes Vektor-Format).
301	EMF	Erweitertes Metafile-Format (neueres Vektor-Format).
350	BMP	Windows Bitmap-Format.

FolderItem Fehler-Konstanten

Um festzustellen, welcher Fehler in der LastErrorCode-Eigenschaft abgelegt ist, prüfen Sie diese gegen die folgenden Konstanten:

Fehler-Code	Konstante	Beschreibung
100	DestDoesNotExistError	Ziel existiert nicht. Diesen Fehler erhalten Sie nur bei CopyFileTo und MoveFileTo.
101	FileNotFound	Datei/Verzeichnis nicht gefunden.
102	AccessDenied	Zugriff verweigert.
103	NotEnoughMemory	Es ist kein Speicher verfügbar.
104	FileInUse	Die Datei ist in Benutzung.
105	InvalidName	Der Name ist ungültig.

Plattformübergreifende Formate

Derzeit sind keine plattformübergreifenden Formate implementiert.

Nicht erkannte Formate oder Formate, die auf der Zielplattform nicht unterstützt werden, führen zu einer UnsupportedFormatException. Die Message-Eigenschaft der Exception enthält zusätzliche Informationen dazu, was schief lief.

Aliase

Wenn ein **FolderItem** ein Alias auf ein anderes **Folderitem** ist, so wird das Alias beim Zugriff auf **FolderItem** aufgelöst, außer wenn Sie TrueChild und TrueItem (Aliase für Win32 werden bei Item und Child nicht aufgelöst) verwenden. Diese liefern das Objekt selbst, auch wenn es sich um ein Alias handelt. Verwenden Sie die Eigenschaft Alias, um festzustellen, ob das FolderItem ein Alias ist.

Weitere Informationen zu Dateitypen finden Sie im Kapitel „Mit Dateien arbeiten“ im Entwickler-Handbuch.

Filme

Die OpenResourceMovie-Methode erlaubt Ihnen, einen Film zu laden, der in einer MooV-Resource innerhalb einer Datei oder Applikation gespeichert ist. Vor QuickTime 4.0 wurde der aktuelle Film in einer MooV-Resource in einer QuickTime-Filmdatei gespeichert. Ab QuickTime 4.0 wird der Film im Data-Fork gespeichert.

Windows

Die Eigenschaft AbsolutePath liefert einen abschließenden Doppelpunkt auf dem Macintosh und einen abschließenden Backslash unter Windows.

Beispiele

Dieses Beispiel liest die Namen aller Objekte vom Schreibtisch des Start-Laufwerks ein und fügt diese in die ListBox Listbox1 ein:

```
Dim i,n as integer
Dim f as FolderItem
f=Volume(0).DesktopFolder
n=f.count
If n>0 then
  For i=1 to n
    ListBox1.addrow f.item(i).name
  Next
End if
```

Folgendes Beispiel verwendet MoveFileTo. Die Quelldatei wird gelöscht und in den Zielordner verschoben.

```
Dim f, g as FolderItem
g=GetFolderItem("HardDisk:Zielverzeichnis")
f=GetFolderItem("StartupVol:Release Notes")
if f <> nil and g<>Nil then
  f.MoveFileTo(g)
  MsgBox "Erfolg!"
else
  MsgBox "Dateien nicht gefunden"
end if
```

Das folgende Beispiel erzeugt eine Textdatei, ändert den Creator vom Default-Creator „R*ch“ in „txt“ und schreibt etwas Text in die Datei.

```
Dim f as FolderItem
Dim stream as TextOutputStream
f=GetSaveFolderItem("TEXT","Daily Planet Staff")
stream=f.CreateTextFile
```

```
f.MacCreator="ttxx"
Stream.WriteLine ("Perry White")
Stream.WriteLine ("Lois Lane")
Stream.WriteLine ("Jimmy Olsen")
Stream.Close
```

Folgendes Beispiel ruft eine „Datei öffnen“ Dialogbox auf, die den Anwender eine QuickTime-Datei auswählen läßt. Der QuickTime-Film wird dann in die Movie-Eigenschaft eines [MoviePlayer](#)-Steuerelements kopiert.

```
Dim f As FolderItem
f=GetOpenFolderItem("video/quicktime")
If f<>Nil then
    MoviePlayer1.movie=f.OpenAsMovie
End if
```

Folgendes Beispiel kopiert Dateien in einen bestimmten Ordner. Der folgende Code gehört in das Action-Event eines Buttons:

```
Dim origin, destination as FolderItem
origin=SelectFolder
if origin <> Nil then
    destination=SelectFolder
    if destination <> Nil then
        CopyFileOrFolderTo origin, destination
        msgBox "Kopiervorgang abgeschlossen!"
    end if
end if
```

Die CopyFileorFolder-Methode sieht folgendermaßen aus:

```
Sub CopyFileorFolder (source as FolderItem,destination as FolderItem)
    dim i as integer
    dim newFolder as FolderItem

    if source.directory then //Es ist ein Ordner
        newFolder=destination.child(source.name)
        newFolder.createAsFolder
        for i=1 to source.count //jeden Eintrag des Ordners abarbeiten
            if source.item(i).directory then
                //es ist ein Ordner
                CopyFileOrFolderTo source.item(i), newFolder //Rekursiver Aufruf dieser
                                                                    Routine mit Übergabe des Ordners
            else
                source.item(i).CopyFileTo newFolder
                //es ist eine Datei, also kopiere sie
            end if
        next
    else //es ist kein Ordner
        source.CopyFileTo destination
    end if
End Sub
```

Verwenden von GetRelative

```
Dim s as String
Dim f,g as FolderItem
f=new Folderitem
g=f.GetRelative(f.GetSaveInfo(Volume(0).Child("Dokumente"),0))
Statictext2.text=g.Absolutepath
```

Siehe auch

[GetFolderItem](#), [GetOpenFolderItem](#), [GetSaveFolderItem](#), [Volume](#), [VolumeCount](#) Funktionen; [BinaryStream](#), [TextInputStream](#), [TextOutputStream](#) Klassen.

FolderItemDialog Klasse

Dies ist die Basisklasse für die [OpenDialog](#), [SaveAsDialog](#), und [SelectFolderDialog](#) Klassen, welche es Ihnen ermöglicht, selbstdefinierte Öffnen-, Sichern- und „Ordner auswählen“-Dialoge zu erzeugen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
ActionButtonCaption	<u>String</u>	Text des Aktionsknopfes (z.B. Wähle, Sichern, Öffnen etc., je nach Zusammenhang). Nicht unbedingt der Default-Knopf des Dialogs.
CancelButtonCaption	<u>String</u>	Text des Abbrechen-Knopfs.
Filter	<u>String</u>	Einer oder mehrere Dateitypen, durch Semikolon getrennt, die zuvor im Dateitypen-Dialog festgelegt wurden. <i>Filter</i> legt fest, welche Dateien innerhalb von <i>InitialDirectory</i> sichtbar sind.
InitialDirectory	<u>FolderItem</u>	Vollständiger oder relativer Pfad des Verzeichnisses, dessen Inhalt gezeigt wird, wenn der Dialog das erste Mal erscheint. Filter legt fest, welche Dateien in diesem Verzeichnis sichtbar sind.
Left	<u>Integer</u>	Abstand (in Pixeln) des linken Randes des Dialogs zum linken Rand des Hauptbildschirms.
PromptText	<u>String</u>	Hilfetext, der innerhalb des Dialogs erscheint. Siehe Beschreibung bei <u>OpenDialog</u> , <u>SaveAsDialog</u> und <u>SelectFolderDialog</u> .
Result	<u>FolderItem</u>	Speichert das Ergebnis des Aufrufs von ShowModal() oder ShowModalWithin. Wenn der Anwender den Dialog bestätigt, enthält <i>Result</i> das <u>FolderItem</u> , das der Auswahl entspricht; andernfalls ist <i>Result</i> <u>Nil</u> .
SuggestedFileName	<u>String</u>	Defaultname der Datei; erscheint als Textvorgabe im Eingabe-Bereich für den Dateinamen.
Title	<u>String</u>	String, der im Titelfalken erscheint.
Top	<u>Integer</u>	Abstand (in Pixeln) der oberen Kante des Dialogs vom oberen Rand des Hauptbildschirms.

Methoden

Name	Parameter	Beschreibung
ShowModal()		Ruft den <u>FolderItemDialog</u> auf. Liefert ein <u>FolderItem</u> in <i>Result</i> , wenn der Anwender den Dialog bestätigt; andernfalls ist <i>Result</i> <u>Nil</u> .
ShowModalWithin	Parent als Window	Ruft unter Mac OS X den <u>FolderItemDialog</u> als ein Sheet-Fenster in dem Elternfenster auf, das durch <i>Parent</i> festgelegt wird. Wenn die Applikation nicht unter Mac OS X läuft, ist ShowModalWithin äquivalent zu ShowModal.

Hinweise

Macintosh-Versionen des **FolderItemDialog** benötigen die Navigation Services für fast alle Eigenschaften. Nicht alle Eigenschaften sind für alle drei Unterklassen von **FolderItemDialog** verfügbar und manche werden unter Windows nicht unterstützt. Die Beschreibungen für jede der drei Unterklassen haben ein Häkchen bei den Dialogen auf der jeweiligen Plattform (Mac OS X, frühere Mac OS Versionen mit Navigation Services, Windows 32).

Folgende Tabelle dokumentiert dies:

Eigenschaft	Macintosh			Windows		
	Open Dialog	SaveAs Dialog	Select Folder	Open Dialog	SaveAs Dialog	Select Folder
Top & Left	✓	✓	✓	✓	✓	✓
PromptText	✓	✓	✓			✓
Title	✓	✓	✓	✓	✓	✓
ActionButtonCaption	✓	✓	✓			✓
CancelButtonCaption	✓	✓	✓			
SuggestedFileName		✓		✓	✓	

Eigenschaft	Macintosh			Windows		
	Open Dialog	SaveAs Dialog	Select Folder	Open Dialog	SaveAs Dialog	Select Folder
Filter	✓			✓	✓	
InitialDirectory	✓	✓	✓	✓	✓	✓

Wenn die Navigation Services auf Rechnern mit Mac OS älter als X nicht installiert sind, ist nur eine sehr kleine Untermenge der Funktionalität verfügbar, wie obige Tabelle dokumentiert. Bei OpenFileDialog wird nur die Filter-Eigenschaft unterstützt; bei SaveAsDialog werden nur die Eigenschaften PromptText und SuggestedFileName unterstützt. Die Navigation Services werden automatisch bei den Systemen von 8.5 bis 9.1 installiert, können auf älteren Systemen aber auch von Hand installiert werden.

Beispiele

Sehen Sie bei den Beispielen für [OpenDialog](#), [SaveAsDialog](#) und [SelectFolderDialog](#) nach.

Siehe auch

[OpenDialog](#), [SaveAsDialog](#), [SelectFolderDialog](#) Klassen.

Font Funktion

Wird verwendet, um auf die Namen der auf dem Rechner des Anwenders installierten Schriften zugreifen zu können.

Syntax

Ergebnis=**Font**(index)

Teil	Art	Beschreibung
<i>Ergebnis</i>	String	Der Name der Schrift, deren Index angegeben wird. Das Ergebnis ist entsprechend kodiert.
<i>index</i>	Integer	Der Index der Schrift.

Hinweis

Auf Schriften wird in alphabetischer Reihenfolge zugegriffen, wobei die Schrift mit dem Index 0 die erste Schrift ist. Verwenden Sie die Funktion [FontCount](#), um die Anzahl der Schriften festzustellen.

Beispiele

Diese Funktion stellt fest, ob die mit Namen angegebene Schrift auf dem Rechner des Anwenders installiert ist:

```
Function FontAvailable(FontName As String) As Boolean
    Dim i,n as Integer
    n=FontCount-1
    For i=0 to n
        If Font(i)=FontName Then
            Return True
        End If
    Next
    Return False
End Function
```

Dieses Beispiel legt dynamisch ein Schrift-Menü im Menü-Array FontFontName an:

```
Dim m as MenuItem
Dim i,n as Integer
n= FontCount-1
FontFontName(0).Text=Font(0)
For i=1 to n
    m=New FontFontName
    m.Text=Font(i)
Next
```

Siehe auch

[FontCount](#), [FontsFolder](#) Funktionen.

FontCount Funktion

Stellt fest, wie viele Schriften auf dem Rechner des Anwenders installiert sind.

Syntax

Ergebnis=**FontCount**

Teil	Art	Name
<i>Ergebnis</i>	<u>Integer</u>	Die Anzahl installierter Schriften.

Hinweis

FontCount wird insbesondere gebraucht, wenn Sie eine Liste aller verfügbaren Schriften aufstellen oder feststellen möchten, ob eine bestimmte Schrift installiert ist.

Beispiele

Ein Beispiel finden Sie in der Beschreibung der Font-Funktion.

Siehe auch

FontCount, FontsFolder Funktionen.

FontsFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Ordner „Zeichensätze“ des Systemordners zu bekommen. Unter Windows bietet es Zugriff auf das Verzeichnis WINDOWS\FONTS.

Syntax

Ergebnis=**FontsFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein <u>FolderItem</u> , das den Zeichensätze-Ordner repräsentiert.

Hinweis

Verwenden Sie die Funktion **FontsFolder** immer dann, wenn Sie Zugriff auf den Ordner „Zeichensätze“ benötigen.

Die Funktion **FontsFolder** bietet eine Möglichkeit, auf den Zeichensätze-Ordner zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktioniert wird.

Wenn Sie feststellen möchten, welche Schriften der Anwender installiert hat, sollten Sie die Funktionen FontCount und Font verwenden.

Unter Linux liefert diese Funktion Nil.

Beispiele

Dieses Beispiel listet alle Einträge des Zeichensatz-Ordners in einer ListBox auf.

```
Dim i as Integer
Dim f as FolderItem
f=FontsFolder
For i=1 to f.count
    ListBox1.addrow f.item(i).name
Next
```

Siehe auch

AppleMenuFolder, ApplicationSupportFolder, ControlPanelsFolder, DesktopFolder, ExtensionsFolder, Font, FontCount, PreferencesFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

For Each...Next Befehl

Geht in einer Schleife in einem eindimensionalen Array alle Elemente durch.

Syntax

For Each Element In Array

[Befehle]

Next

Teil	Beschreibung
<i>Element</i>	Eine Variable vom gleichem Datentyp wie <i>Array</i> , die auf ein Element von <i>Array</i> verweist. Die Schleife arbeitet jeden Wert in <i>Array</i> ab.
<i>Array</i>	Ein eindimensionales Array vom Typ <u>Integer</u> , <u>Single</u> , <u>Double</u> , oder <u>String</u> .

Beispiele

Folgende Funktion addiert die Zahlen in dem übergebenen Array.

```
Function SumStuff(values() as Double) as Double
    Dim sum, element as Double
    For Each element In values
        sum=sum+element
    Next
    Return sum
```

Um die Funktion aufzurufen, übergeben Sie ein Array aus Doubles in einem Befehl wie folgendem:

```
s=SumStuff(MyNumbers)
```

wobei **s** als Double deklariert ist und **MyNumbers** ein Array aus Doubles ist.

Siehe auch

For...Next, Dim Befehl.

For...Next Befehl

Führt eine Reihe von Befehlen eine bestimmte Anzahl oft aus. Der For Each...Next-Befehl ist eine Variante, die einmal für jedes Element eines eindimensionalen Arrays ausgeführt wird.

Syntax

Schleifen haben zwei Syntax-Formen:

For counter = start To end [Step step]

[Anweisungen]

[Exit]

[Anweisungen]

Next

oder

For counter = start DownTo end [Step step]

[Anweisungen]

[Exit]

[Anweisungen]

Next

Teil	Beschreibung
<i>counter</i>	Eine lokale Variable, die bei jedem Durchlauf der Schleife erhöht wird. Sie kann ein <u>Integer</u> , <u>Single</u> oder <u>Double</u> sein.
<i>start</i>	Anfangswert von <i>counter</i> .
<i>end</i>	Schlußwert von <i>counter</i> .
<i>step</i>	Betrag, um den <i>counter</i> bei jedem Durchlauf der Schleife erhöht wird. <i>Step</i> hat den Wert 1, wenn kein anderer Wert angegeben wird.

Teil	Beschreibung
Anweisungen	Befehle, die <i>counter</i> -mal ausgeführt werden sollen (genauer: bis <i>counter</i> größer als <i>end</i> ist oder kleiner als <i>end</i> , falls man <i>DownTo</i> verwendet).

Hinweis

Wenn *start*, *end* oder *step* ein Ausdruck ist, der berechnet werden muß, führt die For-Schleife diese Berechnung jedesmal aus, wenn die Schleife ausgeführt wird, auch wenn der Ausdruck jedesmal das gleiche Ergebnis ergibt. Daher wird diese Schleife:

```
For i=0 to FontCount-1
.
.
Next
```

sehr viel langsamer sein als folgende Variante:

```
Dim nFonts as integer
nFonts=FontCount-1
For i=0 to nFonts
.
.
Next
```

Wie groß der Unterschied ist, hängt natürlich von der Geschwindigkeit des Rechners und der benötigten Zeit für die Berechnung des Ausdrucks ab.

Sie können den Zähler entweder erhöhen oder erniedrigen. Wenn *Step* positiv oder Null ist, so wird die Schleife so lange ausgeführt, bis *counter* größer als *end* ist.

Um den Zähler zu erniedrigen, verwenden Sie die zweite Syntax unter Verwendung von **DownTo** und einem positiven Wert von *Step*. *Counter* wird erhöht oder gesenkt, wenn der Befehl **Next** erreicht wird.

Counter muß eine lokale Variable und vom Typ Integer sein.

Der Befehl Exit kann verwendet werden, um eine Schleife vorzeitig zu unterbrechen und setzt die Anwendung beim Befehl fort, der **Next** folgt.

For...Next-Befehle können beliebig ineinander verschachtelt werden. Dennoch muß jeder **For...Next**-Befehl einen eigenen *counter* haben.

Wenn die For-Schleife läuft, übernimmt sie die Benutzerschnittstelle, um den Anwender daran zu hindern, mit Menüs oder Steuerelementen zu arbeiten. Normalerweise macht das nichts aus. Wenn jedoch die Schleife sehr lange läuft, sollten Sie deren Code in einen separaten Thread verschieben, damit sie im Hintergrund läuft.

Beispiele

Dieses Beispiel verwendet den Befehl **For...Next**, um die Werte aus einem Array zu überprüfen und solchen mit dem Wert „Heute“ den Wert „Morgen“ zuzuweisen.

```
Dim i.last As Integer
last=Ubound(aDays)
For i=1 to last
If aDays(i)="Heute" Then
aDays(i)="Morgen"
End If
Next
```

Folgender Code verwendet das Schlüsselwort *DownTo* um den *counter* zu erniedrigen:

```
Dim i as Integer
For i=5 DownTo 1 Step 1
Beep
Next
```

Siehe auch

Do...Loop, While...Wend, For...Each Befehle.

Format Funktion

Formatiert eine Zahl anhand von Parametern und liefert das formatierte Ergebnis als String zurück. Die Format-Funktion ist ähnlich der Art und Weise, in der in Tabellenkalkulations-Anwendungen Zahlen formatiert ausgegeben werden können.

Syntax

Ergebnis = **Format**(Zahl, formatSpec)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die formatierte Zahl.
<i>Zahl</i>	<u>Double</u>	Die zu formatierende Zahl.
<i>formatSpec</i>	<u>String</u>	Definiert das Format, das auf die Zahl angewendet werden soll.

Hinweis

FormatSpec ist ein String aus einem oder mehreren besonderen Zeichen, die bestimmen, wie eine Zahl formatiert wird:

Zeichen	Beschreibung
#	Platzhalter, der ein Zeichen der Zahl ausgibt, falls an der Stelle ein Zeichen vorhanden ist.
0	Platzhalter, der ein Zeichen der Zahl ausgibt, falls an der Stelle ein Zeichen vorhanden ist. Ist kein Zeichen vorhanden, so wird stattdessen 0 (null) ausgegeben.
.	Platzhalter für die Position des Dezimalpunktes.
,	Platzhalter, der angibt, ob die Zahl mit Tausender-Punkten formatiert werden soll.
%	Die mit 100 multiplizierte Zahl.
(	Gibt eine offene Klammer aus.
)	Gibt eine geschlossene Klammer aus.
+	Gibt links von der Zahl das Vorzeichen der Zahl aus.
-	Gibt ein negatives Vorzeichen links von der Zahl aus, falls die Zahl negativ ist. Auf positive Zahlen hat dies keine Auswirkung.
<i>E oder e</i>	Zeigt die Zahl in wissenschaftlicher Notation an.
<i>\character</i>	Zeigt das Zeichen, das dem Backslash (\) folgt.

Der absolute Wert der Zahl wird angezeigt. Sie müssen die Zeichen „+“ oder „-“ verwenden, wenn das passende Vorzeichen ausgegeben werden soll.

Bitte beachten Sie, daß die tatsächlich ausgegebenen Zeichen zusätzlich noch anhand der Einstellungen des „Zahlen“-Kontrollfeldes formatiert werden.

FormatSpec kann aus bis zu drei Formatierungszuweisungen bestehen, die durch jeweils ein Semikolon unterteilt werden. Das erste Format wird zur Formatierung positiver Werte verwendet, das zweite Format für negative Werte. Das dritte Format wird verwendet, um den Wert 0 (Null) zu formatieren.

Beispiele

Dies sind mehrere Beispiele zur Formatierung mit Hilfe der Sonderzeichen zur Formatierung.

Format	Zahl	Ausgegebener String
#.#	1.784	1.78
#.0000	1.3	1.3000
0000	5	0005
##%	0.25	25%
###,###.##	145678.5	145,678.5
##.#e+	145678.5	146e+5
-.##	-3.7	-3.7
+#.##	3.7	+3.7

Format	Zahl	Ausgegebener String
#.##;(##.##);\N\U\I	3.7	3.7
#.##;(##.##);\N\U\I	-3.7	(3.7)
#.##;(##.##);\N\U\I	0	Null

Das folgende Beispiel formatiert die Zahl 3560.3 als \$3,560.30:

```
Dim s as String
s=Format(3560.3, "$###,##0.00")
```

FrameColor Funktion

Liefert die momentan gewählte Rahmenfarbe des Erscheinungsbild-Managers zurück.

Syntax

Ergebnis=FrameColor

Teil	Art	Beschreibung
Ergebnis	Color	Die zum Zeichen des Rahmens für ein <u>Control</u> -Steuerelement verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit Canvas Steuerelementen wichtig, um deren Aussehen mit denen des Erscheinungsbild-Managers gleich zu halten. Wenn Sie Objekte zeichnen, sollten Sie diese Farbe als Rahmenfarbe definieren.

ListBoxen verwenden z.B. diese Farbe, um den dunklen Rahmen um die ListBox herum zu zeichnen

Dieser Wert kann vom Anwender geändert werden, weshalb Sie in jedem Zeichen-Event diesen Wert neu abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

DarkBevelColor, DarkTingeColor, FillColor, HighlightColor, LightBevelColor, LightTingeColor, TextColor Funktionen; Color Datentyp.

Für die Deklaration von Bibliotheken können nur String-Konstanten verwendet werden Fehler

Tritt auf, wenn Sie eine externe Funktion unter Verwendung einer Konstanten deklariert haben, die Konstante aber entweder nicht definiert ist oder sie etwas anderes als einen String enthält.

Function Befehl

Deklariert Namen, Parameter, Rückgabewert und Quelltext einer Funktion (Methode).

Syntax

Function name(((parameterList))) As Typ

[local variable declarations]

[Anweisungen]

[Return]

[Anweisungen]

[exception handlers]

[Finally]

End Function

Teil	Beschreibung
name	Benötigt. Dies ist der Name der Funktion (Methode) und folgt den üblichen Konventionen, nach denen auch Variablen benannt werden.

Teil	Beschreibung
<i>parameterList</i>	Optional. Dies ist eine Liste von Werten, die die Parameter repräsentieren, die beim Funktionsaufruf an die Funktion übergeben werden. Mehrere Parameter werden durch Kommata getrennt.
<i>Typ</i>	Der Datentyp des Rückgabewerts

Hinweis

Der Befehl **Function** wird verwendet, um eigene Methoden zu definieren, die genau wie z.B. die eingebauten Funktionen Abs, Len, etc. auf der rechten Seite eines Ausdrucks stehen können. Methoden werden immer mit einem Objekt verknüpft. Jeder ausführbare Programmcode muß von einem Sub- oder einem **Function**-Befehl umschlossen werden. Eine **Function** unterscheidet sich von einer Sub-Routine dadurch, daß eine **Function** auf der rechten Seite eines Ausdrucks stehen kann, nicht aber eine Sub-Routine. Eine **Function** kann nicht innerhalb einer anderen **Function** oder einer Sub-Routine definiert werden. Eine **Function** führt jede Zeile des Programmcodes von oben nach unten hin aus. Wenn die letzte Zeile des Programmcodes einer **Function** abgearbeitet wurde, gibt diese die Kontrolle über den Programmablauf wieder an die Stelle im Programmcode ab, die die **Function** aufgerufen hat.

Eine **Function** wird aufgerufen, indem ihr Name gefolgt von den in Klammern eingeschlossenen Parametern angegeben wird.

Alle in einer **Function** verwendeten Variablen müssen deklariert werden, bevor sie in einem Aufruf verwendet werden können. Variablen und Parameter, die in einer **Function** verwendet werden, sind lokal. Auf Eigenschaften kann aus einer beliebigen **Function** oder einer Sub-Routine zugegriffen werden. Lokale Variablen sind Variablen, die bei jedem Start einer **Function** neu angelegt und beim Ende der **Function** zerstört werden. Als Folge kann auf sie nur mit den Befehlen innerhalb der **Function** zugegriffen werden. Sie werden durch den Befehl Dim aus einer Sub-Routine oder einer **Function** heraus angelegt. Variablen, die zu einem Modul gehören, bilden eine Ausnahme: diese würden dadurch zu globalen Variablen gemacht werden.

Der Befehl **Return** kann verwendet werden, um sofort die Kontrolle über den Programmablauf an den Befehl zurückzugeben, der die **Function** aufgerufen hat. Mit **Return** kann auch ein Rückgabewert an diesen Befehl übermittelt werden. Wird der Befehl **Return** nicht aufgerufen, so wird dem aufrufenden Befehl der Wert Null in dem Datenformat übermittelt, in dem die **Function** aufgerufen wurde.

Exception-Handler sind Befehle, die Fehler verarbeiten. In der Beschreibung der RuntimeException Klasse und Exception Block finden Sie weitere Informationen.

Wenn Ihre Methode keinen Rückgabewert benötigt, sollten Sie diese als Sub-Routine deklarieren. Eine Sub-Routine ist eine Methode, die keinen Wert zurückliefert. Bei der Beschreibung des Befehls Sub finden Sie weitere Informationen.

Die Variable, die eine Funktion zurückgeben kann, ist entweder ein einzelnes Element oder ein Array. Möchten Sie nur einen Wert zurückgeben, spezifizieren Sie dessen Typ im Rückgabetypp-Feld. Möchten Sie ein Array zurückgeben, fügen Sie dem Typ noch leere Klammern an. Wenn Sie zum Beispiel ein Array von Integern zurückgeben wollen, geben Sie „Integer()“ als Rückgabetypp an, anstatt „Integer“.

Wenn Sie verschiedene Werte zurückliefern müssen und dies nicht in Form eines Arrays, können Sie das Schlüsselwort ByRef verwenden, wenn Sie die Parameter der Routine definieren. Wenn Sie ByRef verwenden, können Sie die Ergebnisse über die Parameter zurückgeben.

Eine **Function**-Methode kann sich selbst aufrufen, was eine Rekursion hervorrufen kann. Eine zu tief gehende Rekursion kann zu Stack-Overflow-Fehlern führen.

Manchmal muß eine Funktion einige Aufräumarbeiten vornehmen, je nachdem ob sie normal beendet wurde oder ob sie verfrüht durch eine Exception verlassen werden mußte. Der optionale „Finally“-Block am Ende der Funktion ist für diesen Zweck gedacht. Code in diesem Block wird ausgeführt, auch wenn eine Exception auftrat, egal ob die Exception abgefangen und bearbeitet wurde oder nicht.

Beispiele

Dieses Beispiel ist eine **Function**, die aus der angegebenen Länge und Breite die Fläche berechnet:

```
Function Flaeche(laenge as Double, hoehe as Double) As Double
    Dim dieFlaeche As Double
    dieFlaeche=laenge*hoehe
    Return dieFlaeche
End Function
```

Dieses Beispiel ist eine vereinfachte Version der obigen **Function** (ohne die zusätzliche lokale Variable):

```
Function Flaeche(laenge as Double, hoehe as Double) As Double
```

```
Return laenge*hoehe
End Function
```

Dieses Beispiel zeigt, wie die oben aufgezeigte Funktion aufgerufen und der Rückgabewert einer Variable zugewiesen werden kann:

```
Dim a As Double
a=Flaeche(10,10) //gibt 100 zurück
```

Siehe auch

Exit, Return, Break, Raise Befehle, RuntimeException Klasse, Sub Befehl.

GameInputElement Klasse

Verwaltet ein Element eines GameInputDevice.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Name</i>	<u>String</u>	Ein lesbarer Name des Eingabe-Elements.
<i>Device</i>	<u>GameInputDevice</u>	Das Eingabegerät des Elements.
<i>Value</i>	<u>Integer</u>	Der momentane Wert des Elements.

Hinweise

Ein **GameInputElement** ist ein Steuerelement eines Eingabegeräts, das für Benutzereingaben bei Spielen verwendet wird (oder auch Tastatur und Maus). Zum Beispiel ist der Feuerknopf eines Joysticks ein Element, sowie ein Mausknopf oder eine Taste auf der Tastatur. Die Element-Methode der GameInputDevice Klasse gibt Ihnen Zugriff auf die Elemente eines spezifischen Geräts. Sie können die WaitForElement-Methode der GameInputManager-Klasse verwenden, wenn Sie auf eine Eingabe über dieses Element warten wollen.

Beispiele

Lesen Sie hierzu die Beispiele für den GameInputManager.

Siehe auch

GameInputManager, GameInputDevice Klassen.

GameInputManager Klasse

Verwaltet alle Spiele-Eingabegeräte eines Computers.

Übergeordnete Klasse

Object

Methoden

Name	Parameter	Beschreibung
DeviceCount		Liefert ein <u>Integer</u> mit der Anzahl der Eingabegeräte.
Device	Index als <u>Integer</u>	Liefert das Gerät mit dem angegebenen Index. Rückgabetyt ist <u>GameInputDevice</u> .
WaitForElement	timeout als <u>Single</u>	Liefert ein <u>GameInputElement</u> , das die Benutzereingabe erhalten hat. Wartet auf alle Elemente aller Eingabegeräte. <i>Timeout</i> ist in Sekunden.

Hinweise

Verwenden Sie die **GameInputManager**-, GameInputDevice- und GameInputElement-Klassen, um Eingabegeräte, wie Joysticks, zu verwalten. Jedes dieser Geräte ist ein GameInputDevice und jedes der Elemente, wie Knöpfe und Joystick-Achsen, sind vom Typ GameInputElement.

Die Systemanforderungen für die jeweilige Plattform sind wie folgt:

- **Windows:** DirectX 8 (oder neuer) muß installiert sein, um Spiel-Eingabegeräte(auch Tastatur) verwenden zu können.
- **Mac OS 8-9:** Es werden nur die Tastatur und die Mausknöpfe unterstützt.
- **Mac OS X:** Tastatur und Mausknöpfe werden, wie unter Mac OS 8-9, unterstützt. Zusätzlich benötigen Sie das HID.Bundle im Verzeichnis der Applikation, wenn Sie weitere Eingabegeräte (zum Beispiel Joysticks) verwenden wollen.

Um Spiel-Eingabegeräte zu unterstützen, erzeugen Sie eine einzige Instanz der **GameInputManager**-Klasse. Sie können nun die Device- und DeviceCount-Eigenschaft der **GameInputManager** Klasse verwenden, um eine Liste der Geräte zu bekommen. Jedes dieser Geräte ist vom Typ GameInputDevice und hat eines oder mehrere Elemente vom Typ GameInputElement, wie zum Beispiel Feuerknöpfe, Mäusräder und so weiter. Sie können eine Liste der Eingabegeräte über die Name-Eigenschaft der GameInputDevice-Klasse und deren Elemente erzeugen.

Benutzen Sie die **WaitForElement**-Methode, um Eingaben von den Elementen zu verarbeiten.

Beispiele

Das folgende Beispiel ermittelt, welches Element vom Benutzer als Feuerknopf verwendet wird. Es wird vorausgesetzt, daß **mManager** eine globale Eigenschaft vom Typ **GameInputManager** und **mFireButton** ein GameInputElement ist.

```
If mManager = Nil then
    mManager = New GameInputManager
End if
mFireButton = mManager.WaitForInput(3)
If mFireButton <> Nil then
    MsgBox "Sie verwenden "+mFireButton.Name+" als Feuerknopf."
End if
```

Dieses Beispiel verarbeitet Eingaben über den Feuerknopf:

```
If mFireButton <> Nil and mFireButton.Value <> 0 then
    //hier Aktion auslösen
End if
```

Das erste Beispiel zeigt, wie der Benutzer die Eingabegeräte konfigurieren kann. Im zweiten Beispiel sieht man, wie man auf die Benutzereingabe eines GameInputElements reagiert. Beachten Sie, daß Sie im zweiten Beispiel den aktuellen Status abholen, statt einen älteren Wert zu erhalten.

Das folgende Beispiel stellt alle Eingabegeräte in einem PopupMenu dar. Auch hier wird eine globale Eigenschaft **mManager** vom Typ **GameInputManager** vorausgesetzt.

```
Dim gCount,i as Integer
If mManager = Nil then
    mManager = New GameInputManager
end if
gCount = mManager.DeviceCount
For i = 0 to gCount-1
    PopupMenu1.AddRow mManager.device(i).Name
Next
```

Dieses Beispiel lädt die Bezeichnungen der Elemente in ein PopupMenu mit Namen ElementPop. Dieser Code wird im Change-Event des PopupMenüs mit den Eingabegeräten abgearbeitet.

```
Dim i, Maxi As Integer
Dim device As GameInputDevice

ElementPop.DeleteAllRows
Device = mManager.Device(Me.ListIndex) //ausgewähltes Gerät im Geräte-Popup
If Device <> Nil then
    Maxi = Device.ElementCount
    For i = 0 to Maxi-1
        ElementPop.AddRow Device.Element(i).Name
    next
end if

mElement = Nil
```

Siehe auch

GameInputDevice, GameInputElement Klassen.

GetAppleEventTarget Funktion

Wird verwendet, um eine Dialogbox anzuzeigen, die dem Anwender erlaubt, die Zielapplikation auf seinem oder einem Netzwerk-rechner auszuwählen. Die Dialogbox zeigt auf der linken Seite die Rechner und auf der rechten Seite die Applikationen. Die Applika-tionen-Liste kann über den Parameter *ListLabel* mit einer Überschrift versehen werden.

Syntax

result=**GetAppleEventTarget** (Prompt,ListLabel)

Teil	Typ	Beschreibung
<i>Result</i>	<u>AppleEventTarget</u>	Referenz auf die Zielapplikation.
<i>Prompt</i>	<u>String</u>	Der Text, der als Überschrift der Dialogbox erscheint.
<i>ListLabel</i>	<u>String</u>	Bezeichnung der Liste der Applikationen in der Dialogbox.

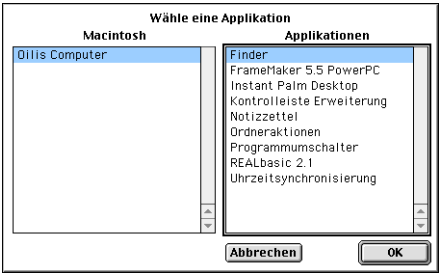
Hinweise

Diese Funktion arbeitet nur unter Mac OS Classic. Insbesondere arbeitet sie nicht unter Carbon oder Mac OS X.

Beispiel

Folgendes Beispiel zeigt die darunter gezeigte „Wähle eine Applikation“-Dialogbox an:

```
Dim AE as AppleEventTarget  
AE=GetAppleEventTarget("Wähle eine Applikation","Applikationen")
```



GetFolderItem Funktion

Wird verwendet, um auf ein Objekt in einem Ordner (Verzeichnis) zuzugreifen.

Syntax

Ergebnis=**GetFolderItem**(Dateiname)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Folderitem</u>	Repräsentiert die geöffnete Datei.
<i>Dateiname</i>	string	Leerer String oder Dateiname.

Hinweise

Die **GetFolderItem**-Funktion erzeugt ein FolderItem-Objekt für einen bestimmten Verzeichniseintrag (Applikation, Datei, Verzeich-nis). **GetFolderItem** kann ein Dateiname übergeben werden oder ein leerer String. Wenn Sie Zugriff über einen absoluten Pfad benötigen, benutzen Sie bitte die Volume-Funktion und die Child-Methode der FolderItem-Klasse.

Wenn Sie eine existierende Datei über **GetFolderItem** öffnen wollen, sollten Sie die Exists-Eigenschaft des FolderItems prüfen, um sicher zu gehen, daß diese Datei existiert.

Wenn Sie einen leeren String übergeben, liefert **GetFolderItem** ein FolderItem für das Verzeichnis, in dem sich die Applikation befindet. Haben Sie das REALbasic-Projekt noch nicht gespeichert, wird das REALbasic-Verzeichnis verwendet.

Für Mac OS X Mach-O Applikationen liefert **GetFolderItem** das Verzeichnis, in dem das Applikation-Bundle liegt, statt des Verzeich-nisses, in dem die ausführbare Datei liegt.

GetFolderItem löst Verknüpfungen (Aliase/Links) automatisch auf. Wenn Sie dies nicht wünschen, verwenden Sie stattdessen GetTrueFolderItem.

Beispiele

Dieses Beispiel stellt den Namen des Verzeichnisses dar, in dem sich das REALbasic-Projekt befindet.

```
Dim currentFolder as FolderItem
currentFolder=GetFolderItem("")
MsgBox currentFolder.name
```

In diesem Beispiel wird die Child-Methode verwendet, um ein Eintrag im aktuellen Verzeichnis zu benutzen:

```
Dim f as FolderItem
f=GetFolderItem("").Child("Project Templates")
```

Hier wird die Parent-Eigenschaft verwendet, um das übergeordnete Verzeichnis des Verzeichnisses zu bekommen, in dem die Applikation liegt:

```
Dim f as FolderItem
f=GetFolderItem("").Parent
```

Im folgenden Beispiel wird eine TIFF-Datei geöffnet, die im gleichen Verzeichnis liegt wie die Applikation und verwendet diese als Hintergrund für ein Canvas Steuerelement:

```
Dim f as FolderItem
f=GetFolderItem("Zippy.tif")
If f.exists then
    Canvas1.backdrop=f.OpenAsPicture
end if
```

Um über einen absoluten Pfad zuzugreifen, konstruieren Sie diesen über die Volume-Funktion und die Child-Methode der FolderItem Klasse:

```
Dim f as FolderItem
f=Volume(0).Child("Documents").Child("Schedule")
```

Siehe auch

FolderItem, FolderItemDialog Klassen, GetTrueFolderItem Funktion, RuntimeException Klasse, NilObjectException Fehler, Nil Objekt.

GetFontTextEncoding Funktion

Ermittelt die Textkodierung, die verwendet wird, wenn man Text mit diesem Font anzeigt. Die meisten Fonts haben die gleiche Kodierung, es gibt aber Ausnahmen wie „Symbol“.

Syntax

result=**GetFontTextEncoding**(FontName)

Teil	Typ	Beschreibung
<i>result</i>	<u>TextEncoding</u>	Textkodierung von <i>FontName</i>
<i>FontName</i>	<u>String</u>	Name des Fonts. Es liefert das Default-Encoding des Systems, wenn Sie einen der Metafont-Namen (z.B. „System“, „SmallSystem“, „System.UTF8“, etc.) übergeben.

Beispiel

Folgendes Beispiel liefert die Textkodierung für den Symbol-Font: Base=33 (MacSymbol), Variant=0, Internet Name=X-MAC-SYMBOL.

```
Dim t as TextEncoding
t=GetFontTextEncoding("Symbol")
```

Siehe auch

TextEncoding Klasse.

GetIndexedObjectDescriptor Funktion

Liefert einen [AppleEventObjectSpecifier](#) zurück, der in einem anderen [AppleEventObjectSpecifier](#) liegt. Der [AppleEventObjectSpecifier](#) wird über Klasse und Index identifiziert.

Syntax

Ergebnis=**GetIndexedObjectDescriptor**(Desired Class, Object, index)

Teil	Art	Beschreibung
<i>Ergebnis</i>	AppleEventObjectSpecifier	Das gefundene Objekt.
<i>DesiredClass</i>	String	Gibt die Klasse des AppleEvent -Objektes an, nach dem Sie suchen.
<i>Object</i>	AppleEventObjectSpecifier	Das Quellobjekt. Übergeben Sie Nil , um in der Anwendungsebene zu suchen.
<i>Index</i>	Integer	Der Index-Wert des Objektes (mit 1 beginnend), auf das Sie zugreifen möchten.

Siehe auch

[AppleEvent](#) Klasse, [GetNamedObjectDescriptor](#), [GetOrdinalObjectDescriptor](#), [GetPropertyObjectDescriptor](#), [GetRangeObjectDescriptor](#), [GetStringComparisonObjectDescriptor](#), [GetTestObjectDescriptor](#) Funktionen.

GetInternetTextEncoding Funktion

Wenn Sie Mails oder Web-Seiten decodieren wollen, sollten Sie diese Funktion verwenden und nicht [GetTextEncoding](#). Diese Funktion ist das Gegenteil von [TextEncoding](#).InternetName. Sie liefert das Encoding, das mit einem Standard-Namen für das Encoding verbunden ist, wie es im Internet verwendet wurde.

Syntax

result=**GetInternetTextEncoding**(InternetEncoding)

Teil	Typ	Beschreibung
<i>result</i>	TextEncoding	Textkodierung von <i>InternetEncoding</i>
<i>InternetEncoding</i>	String	Internet-Text.

Siehe auch

[GetTextEncoding](#), [TextEncoding](#) Funktion.

GetNamedObjectDescriptor Funktion

Liefert einen [AppleEventObjectSpecifier](#), der in einem anderen [AppleEventObjectSpecifier](#) liegt. Der [AppleEventObjectSpecifier](#) wird über Klasse und Name identifiziert.

Syntax

Ergebnis=**GetNamedObjectDescriptor**(DesiredClass, Object, Name)

Teil	Art	Beschreibung
<i>Ergebnis</i>	AppleEventObjectSpecifier	Das gefundene Objekt.
<i>DesiredClass</i>	String	Gibt die Klasse des AppleEvent -Objektes an, nach dem Sie suchen.
<i>Object</i>	AppleEventObjectSpecifier	Das Quellobjekt. Übergeben Sie Nil , um in der Anwendungsebene zu suchen.
<i>Name</i>	String	Der Index-Wert des Objektes (mit 1 beginnend), auf das Sie zugreifen möchten.

Siehe auch

[AppleEvent](#) Klasse, [GetIndexedObjectDescriptor](#), [GetOrdinalObjectDescriptor](#), [GetPropertyObjectDescriptor](#), [GetRangeObjectDescriptor](#), [GetStringComparisonObjectDescriptor](#), [GetTestObjectDescriptor](#) Funktionen.

GetOpenFolderItem Funktion

Wird verwendet, um auf ein vom Anwender über die Datei-Öffnen-Dialogbox ausgewähltes Dokument oder eine ausgewählte Anwendung zuzugreifen zu können.

Syntax

Ergebnis = **GetOpenFolderItem**(filter)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Repräsentiert die vom Anwender geöffnete Datei. Wählt der Benutzer den Abbruch-Button, so hat <i>Ergebnis</i> den Wert <u>Nil</u> .
<i>filter</i>	<u>String</u>	Durch Semikolon (;) getrennte Listeneinträge der Dateitypen, die der Anwender öffnen kann. Die Dateitypen müssen im Dateitypen-Dialog der REALbasic-Entwicklungsumgebung definiert sein.

Hinweis

Die Funktion **GetOpenFolderItem** zeigt die Standard-Datei-Öffnen-Dialogbox an. Wenn der Anwender Mac OS 8.5 oder neuer installiert hat, erscheint automatisch die neue „Datei öffnen“-Navigation-Services-Dialogbox als beweglicher modaler Dialog:



Der Parameter *filter* wird verwendet, um die zur Auswahl stehenden Dateien auf eine oder mehrere der Dateien zu reduzieren, die der REALbasic-Programmierer im Dateitypen-Dialog auswählen kann. Der *filter* ist eine durch Semikolon getrennte Liste von Dateitypen. Wenn der Anwender beispielsweise beim Aufruf von **GetOpenFolderItem** nur Textdateien und PostScript-Dokumente öffnen können soll, fügen Sie der Liste der Dateitypen in ihrer Anwendung die zwei passenden Dateitypen hinzu. Wenn Sie die eingebauten Dateityp-Vorlagen nutzen, fügen Sie einfach die Dateitypen „application/postscript“ und „application/text“ hinzu.

Beachten Sie, daß Windows-Extensions automatisch ausgefüllt werden, wenn Sie die Auswahl über das Dateitypen-Popup vornehmen.



Sie müssen dann der **GetOpenFolderItem**-Funktion „application/text; application/postscript“ als *filter* übergeben.

Nur Dateien, die zu einem der in *filter* angegebenen Dateitypen passen, werden in der Datei-Öffnen-Dialogbox angezeigt. Möchten Sie alle Dateien in der Datei-Öffnen-Dialogbox anzeigen, so müssen Sie im Dateitypen-Dialog einen Dateityp mit dem Wert „????“ als Macintosh-Typ eintragen. Der eingebaute Dateityp „special/any“ legt fest, daß ein beliebiger Dateityp akzeptiert wird.

Die Funktion **GetOpenFolderItem** liefert ein FolderItem zurück, das die vom Anwender ausgewählte Datei repräsentiert. Sie können dieses FolderItem verwenden, um verschiedene Informationen über eine Datei wie etwa den Namen oder ihren vollständigen Zugriffspfad zu erhalten. Bei der Beschreibung des Objekts FolderItem finden Sie weitere Informationen.

Wenn der Anwender in der Datei-Öffnen-Dialogbox den Abbruch-Button betätigt, hat das FolderItem den Wert Nil. Vergleichen Sie daher das FolderItem mit dem Wert Nil, bevor Sie mit dem FolderItem arbeiten. Der Zugriff auf ein FolderItem mit dem Wert Nil wird einen Fehler hervorrufen.

Beispiele

Dieses Beispiel zeigt eine Datei-Öffnen-Dialogbox an, in dem der Anwender einen QuickTime-Film auswählen kann. Der QuickTime-Film wird dann in die Movie-Eigenschaft eines MoviePlayer Steuerelements kopiert.

```
Dim f As FolderItem
f=GetOpenFolderItem("video/quicktime")
If f<>Nil then
    MoviePlayer1.movie=f.OpenAsMovie
End if
```

Siehe auch

GetSaveFolderItem, SelectFolder Funktionen; FolderItemDialog Klasse.

GetOrdinalObjectDescriptor Funktion

Liefert einen AppleEventObjectSpecifier zurück, der in einem anderen AppleEventObjectSpecifier liegt. Der AppleEventObjectSpecifier wird über Klasse und Ordnungsschlüssel (Ordinal Key) identifiziert.

Syntax

Ergebnis=GetOrdinalObjectDescriptor(DesiredClass, Object, OrdinalKey)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>AppleEventObjectSpecifier</u>	Das gefundene Objekt.
<i>DesiredClass</i>	<u>String</u>	Gibt die Klasse des <u>AppleEvent</u> -Objektes an, nach dem Sie suchen.
<i>Object</i>	<u>AppleEventObjectSpecifier</u>	Das Quellobjekt. Übergeben Sie <u>Nil</u> , um in der Anwendungsebene zu suchen.
<i>OrdinalKey</i>	<u>String</u>	Der bei der Suche zu beachtende Bereich.

Hinweis

Die Ordnungsschlüssel sind:

Schlüssel	Beschreibung
<i>firs</i>	Erster
<i>last</i>	Letzter
<i>midd</i>	Mittlerer
<i>any</i>	Beliebiger
<i>all</i>	Alle

Ordnungsschlüssel sind vier Zeichen lang. Den Schlüssel „any“ and „all“ folgt jeweils ein Leerzeichen.

Beispiele

Dieses Beispiel fragt den Finder nach der Anzahl momentan laufender Prozesse:

```
Dim a as AppleEvent
Dim i, count as Integer
a = NewAppleEvent("core", "cnte", "MACS")
a.MacTypeParam("koc1") = "prcs"
a.ObjectSpecifierParam("----") = GetOrdinalObjectDescriptor("prcs", nil, "all ")
If a.Send then
    count = Val(a.ReplyString)
End if
```

Siehe auch

AppleEvent Klasse; GetIndexedObjectDescriptor, GetNamedObjectDescriptor, GetPropertyObjectDescriptor, GetRangeObjectDescriptor, GetStringComparisonObjectDescriptor, GetTestObjectDescriptor Funktionen.

GetPropertyObjectDescriptor Funktion

Liefert einen [AppleEventObjectSpecifier](#), der auf eine Eigenschaft des übergebenen [AppleEventObjectSpecifier](#) verweist.

Syntax

Ergebnis=**GetPropertyObjectDescriptor**(Object, Name)

Teil	Art	Beschreibung
<i>Ergebnis</i>	AppleEventObjectSpecifier	Ein Objekt, das auf die Name-Eigenschaft eines Objekts verweist.
<i>Object</i>	AppleEventObjectSpecifier	Das Quellobjekt. Übergeben Sie Nil , um in der Anwendungsebene zu suchen.
<i>Name</i>	String	Der Name der Eigenschaft des Objektes, die zurückgeliefert werden soll.

Siehe auch

[AppleEvent](#) Klasse; [GetIndexedObjectDescriptor](#); [GetNamedObjectDescriptor](#); [GetOrdinalObjectDescriptor](#); [GetRangeObjectDescriptor](#); [GetStringComparisonObjectDescriptor](#); [GetTestObjectDescriptor](#) Funktionen.

GetQTCrossFadeEffect Funktion

Erzeugt einen cross-fade-Effekt. Verwenden Sie [QTEffectSequence](#) um eine Effekt-Sequenz für den cross-fade-Effekt zu erzeugen.

Syntax

result=**GetQTCrossFadeEffect**

Teil	Typ	Beschreibung
<i>result</i>	QTEffect	Cross-fade-Effekt

Beispiel

Dieses Beispiel erzeugt einen QuickTime-Movie aus zwei Bildern (p1 und p2) unter Verwendung des cross-fade-Effekts.

```
Dim theEffect as QTEffect
Dim sequence as QTEffectSequence
theEffect=GetQTCrossFadeEffect
sequence=New QTEffectSequence(theEffect,p1,p2,96)
```

Siehe auch

[QTTrack](#), [QTVideoTrack](#), [QTEffect](#), [QTEffectSequence](#) Klassen; [GetQTSMPTEEffect](#) Funktion.

GetQTGraphicsExporter Funktion

Erzeugt ein Exporter-Objekt, das ein bestimmtes Grafik-Format verwendet.

Syntax

Ergebnis=**GetQTGraphicsExporter**(Format)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	QTGraphicsExporter	Exporter-Objekt.
<i>Format</i>	String	Das zum Speichern der Grafik verwendete Format BMPf - BMP-Dateiformat PNGf - PNG-Dateiformat PNTG - MacPaint 8BPS - Photoshop .SGI - SGI TPIC - Targa JPEG - Jpeg

Beispiel

Siehe Beispiel unter [QTGraphicsExporter](#).

Siehe auch

[QTGraphicsExporter](#) Klasse.

GetQTSMPTEffect Funktion

Liefert einen SMPTE-Effekt.

Syntax

Ergebnis = **GetQTSMPTEffect**(ID)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>QTEffect</u>	Der gewählte SMPTE-Effekt.
<i>ID</i>	<u>Integer</u>	ID des SMPTE-Effekts (siehe folgende Tabelle)

ID	Typ	Effekt
1	Wipe	Slide horizontal
2	Wipe	Slide vertical
3	Wipe	Top left
4	Wipe	Top right
5	Wipe	Bottom right
6	Wipe	Bottom left
7	Wipe	Four corner
8	Wipe	Four box
21	Wipe	Barn vertical
22	Wipe	Barn horizontal
23	Wipe	Top center
24	Wipe	Right center
25	Wipe	Bottom center
26	Wipe	Left center
41	Wipe	Diagonal left down
42	Wipe	Diagonal right down
43	Wipe	Vertical bow tie
44	Wipe	Horizontal bow tie
45	Wipe	Diagonal left out
46	Wipe	Diagonal right out
47	Wipe	Diagonal cross
48	Wipe	Diagonal box
61	Wipe	Filled V
62	Wipe	Filled V right
63	Wipe	Filled V bottom
64	Wipe	Filled V left
65	Wipe	Hollow V
66	Wipe	Hollow V right
67	Wipe	Hollow V bottom
68	Wipe	Hollow V left

ID	Typ	Effekt
71	Wipe	Vertical zig zag
72	Wipe	Horizontal zig zag
73	Wipe	Vertical barn zig zag
74	Wipe	Horizontal barn zig zag
501	Wipe	Random wipe — einer der 34 SMPTE Wipe-Effekte wird per Zufall ausgewählt
101	Iris	Rectangle
102	Iris	Diamond
103	Iris	Triangle
104	Iris	Triangle right
105	Iris	Triangle upside down
106	Iris	Triangle left
107	Iris	Arrowhead
108	Iris	Arrowhead right
109	Iris	Arrowhead upside down
110	Iris	Arrowhead left
111	Iris	Pentagon
112	Iris	Pentagon upside down
113	Iris	Hexagon
114	Iris	Hexagon side
119	Iris	Circle
120	Iris	Oval
121	Iris	Oval side
122	Iris	Cat eye
123	Iris	Cat eye slide
124	Iris	Round rect
125	Iris	Round rect side
127	Iris	4 point star
128	Iris	5 point star
129	Iris	6 point star
130	Iris	Heart
131	Iris	Keyhole
502	Iris	Random iris — einer der 26 SMPTE Iris-Effekte wird per Zufall ausgewählt
201	Radial	Rotating top
202	Radial	Rotating right
203	Radial	Rotating bottom
204	Radial	Rotating left
205	Radial	Rotating top bottom
206	Radial	Rotating left right
207	Radial	Rotating quadrant
211	Radial	Top to bottom 180°
212	Radial	Right to left 180°
213	Radial	Top to bottom 90°

ID	Typ	Effekt
214	Radial	Right to left 90°
221	Radial	Top 180°
222	Radial	Right 180°
223	Radial	Bottom 180°
224	Radial	Left 180°
225	Radial	Counter rotating top bottom
226	Radial	Counter rotating left right
227	Radial	Double rotating top bottom
228	Radial	Double rotating left right
231	Radial	V Open top
232	Radial	V Open right
233	Radial	V Open bottom
234	Radial	V Open left
235	Radial	V Open top bottom
236	Radial	V Open left right
241	Radial	Rotating top left
242	Radial	Rotating bottom left
243	Radial	Rotating bottom right
244	Radial	Rotating top right
245	Radial	Rotating top left bottom right
246	Radial	Rotating bottom left top right
251	Radial	Rotating top left right
252	Radial	Rotating left top bottom
253	Radial	Rotating bottom left right
254	Radial	Rotating right top bottom
261	Radial	Rotating double center right
262	Radial	Rotating double center top
263	Radial	Rotating double center top bottom
264	Radial	Rotating double center left right
503	Radial	Random radial — einer der 39 SMPTE Radial-Effekte wird per Zufall ausgewählt
301	Matrix	Horizontal matrix
302	Matrix	Vertical matrix
303	Matrix	Top left diagonal matrix
304	Matrix	Top right diagonal matrix
305	Matrix	Bottom right diagonal matrix
306	Matrix	Bottom left diagonal matrix
310	Matrix	Clockwise top left matrix
311	Matrix	Clockwise top right matrix
312	Matrix	Clockwise bottom right matrix
313	Matrix	Clockwise bottom left matrix
314	Matrix	Counterclockwise top left matrix
315	Matrix	Counterclockwise top right matrix

ID	Typ	Effekt
316	Matrix	Counterclockwise bottom right matrix
317	Matrix	Counterclockwise bottom left matrix
320	Matrix	Vertical start top matrix
321	Matrix	Vertical start bottom matrix
322	Matrix	Vertical start top opposite matrix
323	Matrix	Vertical start left matrix
324	Matrix	Horizontal start left matrix
325	Matrix	Horizontal start right matrix
326	Matrix	Horizontal start left opposite
327	Matrix	Horizontal start right opposite matrix
328	Matrix	Double diagonal top right matrix
329	Matrix	Double diagonal bottom right matrix
340	Matrix	Double spiral top matrix
341	Matrix	Double spiral bottom matrix
342	Matrix	Double spiral left matrix
343	Matrix	Double spiral right matrix
344	Matrix	Quad spiral vertical matrix
345	Matrix	Quad spiral horizontal matrix
350	Matrix	Vertical waterfall left matrix
351	Matrix	Vertical waterfall right matrix
352	Matrix	Horizontal waterfall left matrix
353	Matrix	Horizontal waterfall right matrix
504	Matrix	Random matrix — einer der 34 SMPTE Matrix-Effekte wird per Zufall ausgewählt
409		Random effect — einer der 133 SMPTE Effekte wird per Zufall ausgewählt

Beispiel

Siehe Beispiel bei [QVideoTrack](#).

Siehe auch

[GetQTCrossFadeEffect](#) Funktion; [QTEffect](#), [QTEffectSequence](#), [QTTrack](#), [QVideoTrack](#) Klassen.

GetRangeObjectDescriptor Funktion

Liefert einen [AppleEventObjectSpecifier](#), der einen Bereich von Objekten enthält, die im übergebenen [AppleEventObjectSpecifier](#) enthalten sind.

Syntax

Ergebnis=**GetRangeObjectDescriptor**(DesiredClass, Object, RangeStart, RangeEnd)

Teil	Art	Beschreibung
<i>Ergebnis</i>	AppleEventObjectSpecifier	Das Objekt, das den Bereich der angegebenen Objekte enthält.
<i>DesiredClass</i>	String	Die Klasse des AppleEvent -Objektes, nach dem Sie suchen.
<i>Object</i>	AppleEventObjectSpecifier	Das Quellobjekt. Übergeben Sie Nil , um in der Anwendungsebene zu suchen.
<i>RangeStart</i>	AppleEventObjectSpecifier	Gibt den Anfang des Suchbereichs an.
<i>RangeEnd</i>	AppleEventObjectSpecifier	Gibt das Ende des Suchbereichs an.

Siehe auch

[AppleEvent](#) Klasse; [GetIndexedObjectDescriptor](#); [GetNamedObjectDescriptor](#); [GetOrdinalObjectDescriptor](#); [GetPropertyObjectDescriptor](#); [GetStringComparisonObjectDescriptor](#); [GetTestObjectDescriptor](#) Funktionen.

GetSaveFolderItem Funktion

Wird verwendet, um dem Anwender die Sichern-Unter-Dialogbox zu zeigen.

Syntax

Ergebnis=**GetSaveFolderItem**(filter, default file name)

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Repräsentiert die Datei, die der Anwender anlegen möchte.
<i>filter</i>	String	Der Dateityp, der dieser Datei zugewiesen werden soll, so wie er in der Dateitypen-Dialogbox definiert wurde.
<i>default file name</i>	String	Der in der Dialogbox vorgegebene Dateiname.

Hinweis

Die Funktion **GetSaveFolderItem** zeigt dem Benutzer die Standard-Sichern-Unter-Dialogbox, damit dieser einen Ort und einen Namen für eine zu sichernde Datei festlegen kann. Wenn der Anwender Mac OS 8.5 oder neuer installiert hat, erscheint automatisch die neue „Datei sichern“-Navigation-Services-Dialogbox.



Die Funktion **GetSaveFolderItem** legt diese Datei nicht an, sie liefert lediglich ein [FolderItem](#), das die potentielle Datei repräsentiert. Um die Datei dann auch wirklich anzulegen, müssen Sie die Methoden [CreateBinaryFile](#) oder [CreateTextfile](#) mit diesem [FolderItem](#) aufrufen.

Der *filter* sollte entweder ein leerer String sein oder der Name eines Dateityps, wie er in der Dateitypen-Dialogbox definiert wurde, so wie „image/gif“, „image/jpeg“ oder „application/rtf“.

Unter Mac OS X erscheint eine Checkbox mit „Suffix ausblenden“ im „Datei sichern“-Dialog. Die `ExtensionVisible`-Eigenschaft des zurückgegebenen FolderItems wird auf die vom Anwender gewählte Einstellung gesetzt.

Beispiele

Dieses Beispiel zeigte die Sichern-Unter-Dialogbox. Danach wird eine Textdatei angelegt und die Text-Eigenschaften dreier EditFields werden in diese Datei geschrieben. Schließlich wird die Datei geschlossen.

```
Dim file As FolderItem
Dim fileStream as TextOutputStream
file=GetSaveFolderItem("", "Mein Info")
if file<>Nil then
  fileStream=file.CreateTextFile
  fileStream.WriteLine namefeld.text
  fileStream.WriteLine adressefeld.text
  fileStream.WriteLine telefonfeld.text
  fileStream.Close
end if
```

Siehe auch

[GetOpenFolderItem](#), [SelectFolder](#) Funktionen; [FolderItemDialog](#), [FolderItem](#) Klassen.

GetStringComparisonObjectDescriptor Funktion

Liefert den [AppleEventObjectSpecifier](#), der auf den angegebenen String-Vergleich paßt.

Syntax

Ergebnis=**GetStringComparisonObjectDescriptor**(ComparisonKey, ComparisonForm, Field, Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	AppleEventObjectSpecifier	Das gefundene Objekt.
<i>ComparisonKey</i>	String	Die Art des Vergleichs. Dieser String muß vier Zeichen lang sein.
<i>ComparisonForm</i>	String	Die Form des Vergleichs. „prop“ gibt beispielsweise an, daß der Vergleich eine Eigenschaft (property) prüfen wird.
<i>Field</i>	String	Das Feld, auf dem der Vergleich angewendet wird. Der Parameter „ComparisonForm“ gibt an, wie die Werte des Feldes interpretiert werden sollen.
<i>Wert</i>	String	Der Vergleichswert, gegen den der Vergleich geprüft wird.

Hinweis

Gültige Werte für den Vergleichsschlüssel (comparison keys) sind:

Vergleichsschlüssel	Beschreibung
=	gleich
>	größer als
> =	größer oder gleich
<	kleiner
< =	kleiner oder gleich

Vergleichsschlüssel müssen vier Zeichen lang sein. Füllen Sie den Vergleichsschlüssel nach den Zeichen soweit mit Leerzeichen auf, daß der übergebene String vier Zeichen lang ist.

Siehe auch

[AppleEvent](#) Klasse; [GetIndexedObjectDescriptor](#); [GetNamedObjectDescriptor](#); [GetOrdinalObjectDescriptor](#); [GetPropertyObjectDescriptor](#); [GetRangeObjectDescriptor](#); [GetTestObjectDescriptor](#) Funktionen.

GetTemporaryFolderItem

Die Funktion **GetTemporaryFolderItem** erzeugt ein [FolderItem](#)-Objekt im aktuellen Temporary Folder.

Syntax

Ergebnis=**GetTemporaryFolderItem**

Teil	Typ	Beschreibung
<i>Ergebnis</i>	FolderItem	FolderItem -Objekt, das auf ein Dokument im Temporary Folder verweist.

Beispiel

Folgender Code erzeugt ein [FolderItem](#) im aktiven Temporary Folder und zeigt seinen absoluten Pfadnamen an.

```
Dim f as FolderItem
f=GetTemporaryFolderItem
MsgBox f.absolutePath
```

Siehe auch

[FolderItem](#) Klasse; [GetFolderItem](#); [TemporaryFolder](#) Funktionen.

GetTestObjectDescriptor Funktion

Liefert einen [AppleEventObjectSpecifier](#), der zur Erledigung eines Tests angelegt wurde.

Syntax

Ergebnis=**GetTestObjectDescriptor**(DesiredClass, Object, Comparison)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>AppleEventObjectSpecifier</u>	Das angelegte Objekt.
<i>DesiredClass</i>	<u>String</u>	Die Klasse des <u>AppleEvent</u> -Objektes, nach dem Sie suchen.
<i>Object</i>	<u>AppleEventObjectSpecifier</u>	Das Quellobjekt. Übergeben Sie <u>Nil</u> , um in der Anwendungsebene zu suchen.
<i>Comparison</i>	<u>AppleEventObjectSpecifier</u>	Das Objekt, das den auszuführenden Test beschreibt.

Hinweis

Zur Zeit ist [GetStringComparisonObjectDescriptor](#) die einzige Funktion, die einen [AppleEventObjectSpecifier](#) zurückliefert, der einen Test beschreibt. Möglicherweise wird sich dies ändern und REALbasic in dieser Hinsicht erweitert werden.

Siehe auch

[AppleEvent](#) Klasse; [GetIndexedObjectDescriptor](#); [GetNamedObjectDescriptor](#); [GetOrdinalObjectDescriptor](#); [GetPropertyObjectDescriptor](#); [GetRangeObjectDescriptor](#); [GetStringComparisonObjectDescriptor](#) Funktionen.

GetTextConverter Funktion

Erzeugt ein [TextConverter](#)-Objekt, um zwischen zwei Kodierungen konvertieren zu können. Es funktioniert auch, wenn die Eingabe- und Ausgabe-Kodierung die Gleiche ist.

Syntax

result=**GetTextConverter**(inputEncoding, outputEncoding)

Teil	Typ	Beschreibung
<i>result</i>	<u>TextConverter</u>	Ein Objekt vom Typ <u>TextConverter</u> , das für eine Konvertierung von <i>InputEncoding</i> nach <i>OutputEncoding</i> verwendet werden kann.
<i>inputEncoding</i>	<u>TextEncoding</u>	Ursprüngliche Textkodierung.
<i>outputEncoding</i>	<u>TextEncoding</u>	Gewünschte Textkodierung.

Hinweise

In REALbasic kann jeder String intern zusätzlich zu den Bytes, die den eigentlichen Text enthalten, ein Record von dessen Kodierung besitzen. In vielen Fällen ist die Kodierung unbekannt, aber jeder Text, der von einem [TextConverter](#) geliefert wird, hat eine bekannte Kodierung und wird somit bei [EditFields](#) und [Graphics.DrawString](#) passend behandelt. Das ist der Grund, warum es manchmal sinnvoll sein kann, einen [TextConverter](#) zu verwenden, obwohl die Kodierung von Input und Output identisch ist; es bietet die Möglichkeit, sicher zu gehen, daß REALbasic weiß, welche Kodierung ein String repräsentiert.

Beispiel

Folgendes Beispiel konvertiert den Text in einem [EditField](#):

```
Dim c as TextConverter  
c=GetTextConverter(GetTextEncoding(&h500), getTextEncoding(0))  
EditField2.text=c.convert(EditField1.text)
```

Siehe auch

[ConvertEncoding](#); [DefineEncoding](#); [GetInternetTextEncoding](#); [GetTextEncoding](#); [GetFontTextEncoding](#); [Encoding](#) Funktionen; [TextConverter](#); [TextEncoding](#) Klassen.

GetTextEncoding Funktion

Liefert ein [TextEncoding](#) Objekt basierend auf benutzerdefinierten Parametern.

Syntax

result=**GetTextEncoding** (Base, [Variant, Format])

Teil	Typ	Beschreibung
<i>result</i>	<u>TextEncoding</u>	TextEncoding-Objekt.
<i>Base</i>	<u>Integer</u>	Kodierungstyp.
<i>Variant</i>	<u>Integer</u>	Spezifische Abwandlung von <i>Base</i>
<i>Format</i>	<u>Integer</u>	Format — wird nur von Unicode verwendet, um zu bestimmen, welches Unicode-Format Sie verwenden wollen.

Hinweis

Wegen der Codes für *Base* und *Variant* siehe Anhang A der gedruckten oder PDF-Version der Sprachreferenz. Die Codes für *Format* sind wie folgt:

Name	Code
<i>Default Format</i>	0
<i>Unicode16BitFormat</i>	0
<i>UnicodeUTF7Format</i>	1
<i>UnicodeUTF8Format</i>	2
<i>Unicode32BitFormat</i>	3

Beispiel

Siehe Beispiel für GetTextConverter.

Siehe auch

ConvertEncoding, DefineEncoding, Encoding, GetTextConverter, GetFontTextEncoding, GetInternetTextEncoding Funktionen; Encodings Objekt; TextEncoding, TextConverter Klassen.

GetTrueFolderItem Funktion

Liefert ein FolderItem, das den echten Pfad (ohne Auflösung von Aliasen) einer Datei oder eines Ordners liefert.

Syntax

result=**GetTrueFolderItem**(Name)

Teil	Typ	Beschreibung
<i>result</i>	<u>FolderItem</u>	<u>FolderItem</u> , das auf <i>Name</i> verweist.
<i>Name</i>	<u>String</u>	Pfadname der Datei oder des Ordners.

Hinweis

Wenn das Objekt, auf das *Name* verweist, nicht existiert, liefert **GetTrueFolderItem** Nil.

Beispiel

Folgendes Beispiel vergleicht GetTrueFolderItem mit GetFolderItem. Während beide Funktionen den gleichen absoluten Pfad zu einem Alias bekommen, verweist f auf das Alias, während g auf die eigentliche Applikation verweist.

```
dim f,g as FolderItem
f=GetTrueFolderItem("Hard Disk:Photoshop")
g=GetFolderItem("Hard Disk:Photoshop")
EditField1.text=f.Name //Das Alias auf die Applikation
EditField2.text=g.Name //Die Applikation
```

Siehe auch

GetFolderItem Funktion; FolderItem Klasse.

GetUniqueIDObjectDescriptor Funktion

Liefert eine eindeutige AppleEvent-Object ID.

Syntax

result=**GetUniqueIDObjectDescriptor**(DesiredClass, Object, ID)

Teil	Typ	Beschreibung
<i>result</i>	<u>AppleEventObjectSpecifier</u>	AppleEvent-Object Specifier
<i>DesiredClass</i>	<u>String</u>	AppleEvent-Klasse
<i>Object</i>	<u>AppleEventObjectSpecifier</u>	AppleEvent-Objekt
<i>ID</i>	<u>String</u>	ID

Siehe auch

AppleEventObjectSpecifier Klasse.

GOTO Befehl

Springt zu einem Label.

Syntax

GOTO label

Teil	Typ	Beschreibung
<i>label</i>	<u>String</u>	Label, an das gesprungen werden soll.

Hinweis

Der GOTO-Befehl ist nur aus Kompatibilitätsgründen zu anderen BASIC-Implementierungen eingebaut. Früher wurde GOTO häufig in unstrukturiertem BASIC für den Programmablauf verwendet. Zu dieser Zeit bestand ein Programm aus einem einzigen Code-Segment und der Programmablauf wurde durch zahlreiche GOTO-Befehle gesteuert, die von einem Code-Bereich in einen anderen sprangen. Nach gewisser Zeit waren solche Programme nur schwer zu warten, da die zugrundeliegende Logik nur schwer zu verfolgen war – vor allem, wenn der ursprüngliche Programmierer das Projekt verlassen hatte. Die Schwierigkeit, solche unstrukturierten Programme zu warten, führte zur Entwicklung event-gesteuerter, objektorientierter Programmierungsumgebungen wie REALbasic.

Beispiel

Folgendes Beispiel zeigt eine Warnmeldung, wenn die Checkbox 1 angekreuzt ist.

```
If checkbox1.value then
  GOTO myCode
End If

//Return wird eingesetzt, damit RB den Label-Ausdruck
//nicht sowieso ausführt.
Return
myCode: //Label, auf das gesprungen werden soll.
MsgBox "Unstrukturierte Programmierung ist schlecht!"
```

Siehe auch

Exit Befehl.

GOTO Ziel nicht gefunden Fehler

Sie haben einen Marker mit einem GOTO Befehl benutzt, der sich nicht in dieser Methode befindet. Aus diesem Grund konnte der Sprung nicht durchgeführt werden.

Beispiel

Das Beispiel des GOTO Schlüsselworts wurde derart modifiziert, das nun ein Marker angesprungen werden soll, der nicht in dieser Methode existiert.


```

If checkbox1.value then
GOTO myCode
End If

//Return wird eingesetzt, damit RB den Label-Ausdruck
//nicht sowieso ausführt.

Return
myLabel: //paßt nicht auf Label in GOTO
MsgBox "Unstrukturierte Programmierung ist schlecht!"

```

Siehe auch

GOTO Befehl.

Graphics Klasse

Die Objekte der Graphics Klasse werden zum Zeichnen von Text, Linien, Rechtecken, Ovalen und Bildern verwendet. Normalerweise verwenden Sie ein Graphics-Objekt als Antwort auf ein Paint-Event, aber Sie können auch direkt zeichnen, indem Sie die Graphics-Eigenschaft eines Canvas Steuerelementes verwenden. Sie können alternativ dazu auch Vektorgrafiken durch das Verwenden von Unterklassen der Object2D-Klasse erzeugen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Bold</i>	<u>Boolean</u>	Ist dies <u>True</u> , so erscheint über die <u>DrawString</u> -Methode ausgegebener Text fett.
<i>Copies</i>	<u>Integer</u>	Der Wert im „Kopien“-Feld des Druckdialoges. Diese Eigenschaft ist nur dann wichtig, wenn das Graphics-Objekt durch die Funktion <u>OpenPrinterDialog</u> angelegt wurde.
<i>FirstPage</i>	<u>Integer</u>	Der Wert im „von Seite ...“-Feld des Druckdialoges. Diese Eigenschaft ist nur dann wichtig, wenn das Graphics-Objekt durch die Funktion <u>OpenPrinterDialog</u> angelegt wurde.
<i>ForeColor</i>	<u>Color</u>	Die momentan für das Graphics-Objekt ausgewählte Farbe. Diese Farbe wird für die verschiedenen Zeichenmethoden verwendet.
<i>Height</i>	<u>Integer</u>	Die Höhe des Elternobjekts in Pixeln, normalerweise ein <u>Canvas</u> -Steuerelement oder ein <u>Fenster</u> . Wenn sich zum Beispiel ein Graphics-Klassenobjekt in einem <u>Canvas</u> -Steuerelement befindet, liefert <i>Height</i> die Höhe des Steuerelements.
<i>Italic</i>	<u>Boolean</u>	Ist dies <u>True</u> , so erscheint über die DrawString-Methode ausgegebener Text kursiv.
<i>LastPage</i>	<u>Integer</u>	Der Wert im „bis Seite ...“-Feld des Druckdialoges. Diese Eigenschaft ist nur dann wichtig, wenn das Graphics-Objekt durch die Funktion <u>OpenPrinterDialog</u> angelegt wurde.
<i>PenHeight</i>	<u>Integer</u>	Die Linienhöhe (in Pixeln), die zum Zeichnen von Linien, Ovalen oder Rechtecken verwendet wird.
<i>PenWidth</i>	<u>Integer</u>	Die Linienbreite (in Pixeln), die zum Zeichnen von Linien, Ovalen oder Rechtecken verwendet wird.
<i>Pixel</i>	X als <u>Integer</u> Y als <u>Integer</u>	Wird verwendet, um die <u>Farbe</u> eines einzelnen Pixels abzufragen oder einzustellen.
<i>TextAscent</i>	<u>Integer</u>	Liefert die Neigung einer Textzeile zurück, die in der aktuellen Schriftart ausgegeben werden würde.
<i>TextFont</i>	<u>String</u>	Wird verwendet, um die aktuelle Schriftart abzufragen oder einzustellen.
<i>TextHeight</i>	<u>Integer</u>	Liefert die Höhe einer Textzeile zurück, die in der aktuellen Schriftart ausgegeben werden würde.
<i>TextSize</i>	<u>Integer</u>	Wird verwendet, um die Schriftgröße der aktuellen Schrift abzufragen oder einzustellen.
<i>Underline</i>	<u>Boolean</u>	Ist dies <u>True</u> , so erscheint über die DrawString-Methode ausgegebener Text unterstrichen.

Name	Typ	Beschreibung
<i>UseOldRenderer</i>	<u>Boolean</u>	Ist dies <u>True</u> , so wird das Rendering unter Mac OS X über QuickDraw und nicht über Quartz erledigt. Auch das Zeichnen von Text wird unter Mac OS X über QuickDraw erledigt. Diese Eigenschaft kann zur Laufzeit geändert werden. Die Voreinstellung für ein neues Objekt ist <u>False</u> . Diese Eigenschaft kann nützlich sein, wenn Sie erreichen wollen, daß eine Grafik unter Mac OS 9 und Mac OS X identisch aussieht. Die QuickDraw-Rendering-Engine ist generell schneller als Quartz, Quartz liefert aber das „native“ Mac OS X-Aussehen. Benötigen Sie schnelleres Zeichnen von Text und können auf das Anti-Aliasing verzichten, sollten Sie die Verwendung von UseOldRenderer in Betracht ziehen.
Width	<u>Integer</u>	Die Breite des Elternobjekts in Pixeln, normalerweise ein <u>Canvas</u> -Steuerelement oder ein <u>Fenster</u> . Wenn sich zum Beispiel ein Graphics-Klassenobjekt in einem <u>Canvas</u> -Steuerelement befindet, liefert <i>Width</i> die Breite des Steuerelements.

Methoden

Name	Parameter	Beschreibung
<i>ClearRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u>	Leert das in den Parametern angegebene Rechteck, indem es dieses mit der Hintergrundfarbe des übergeordneten Fensters füllt.
<i>DrawCautionIcon</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Zeichnet an den angegebenen Koordinaten das Warn-Icon.
<i>DrawLine</i>	X1 als <u>Integer</u> , Y1 als <u>Integer</u> , X2 als <u>Integer</u> , Y2 als <u>Integer</u>	Zeichnet in der aktuellen Farbe eine Linie von X1, Y1 nach X2, Y2.
<i>DrawObject</i>	Objekt als Object2D, [X als <u>Integer</u> , Y als <u>Integer</u>]	Zeichnet das übergebene Object2D-Objekt. Die optionalen Parameter X und Y sind der Abstand von der linken oberen Ecke des Graphics -Objekts.
<i>DrawOval</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u>	Zeichnet die Konturen eines Ovals in der aktuellen Farbe.
<i>DrawPicture</i>	Bild als <u>Picture</u> , X als <u>Integer</u> , Y als <u>Integer</u> [,DestWidth als <u>Integer</u> , DestHeight als <u>Integer</u> , SourceX als <u>Integer</u> , SourceY als <u>Integer</u> , SourceWidth als <u>Integer</u> , SourceHeight als <u>Integer</u>]	Zeichnet das Bild an der angegebenen Position. Die optionalen (mit eckigen Klammern umrahmten) Parameter werden verwendet, um einen Ausschnitt des Bildes zu kopieren oder das Bild zu skalieren. <i>DestWidth</i> und <i>DestHeight</i> werden verwendet, um das Bild beim Zeichnen zu skalieren und sind mit Breite und Höhe des Bildes vorbelegt. <i>SourceX</i> und <i>SourceY</i> sind mit 0 vorbelegt und werden verwendet, um die obere linke Ecke des zu kopierenden Bildausschnitts festzulegen. <i>SourceWidth</i> und <i>SourceHeight</i> sind mit Breite und Höhe des Bildes vorbelegt und werden verwendet, um den Bereich des zu kopierenden Bildausschnitts festzulegen.
<i>DrawPolygon</i>	Points() als <u>Integer</u>	Zeichnet ein Polygon mit den Werten aus dem übergebenen Array. Ungerade Felder des Arrays sind X-Koordinaten, gerade Felder des Arrays sind Y-Koordinaten.
<i>DrawNotelIcon</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Zeichnet an den angegebenen Koordinaten das Sprechblasen-Icon.
<i>DrawRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u>	Zeichnet die Konturen eines Rechtecks in der aktuellen Farbe. X und Y sind die Koordinaten der linken oberen Ecke.
<i>DrawRoundRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u> , OvalWidth als <u>Integer</u> , OvalHeight als <u>Integer</u>	Zeichnet die Konturen eines abgerundeten Rechtecks in der aktuellen Farbe.
<i>DrawStopIcon</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Zeichnet an den angegebenen Koordinaten das Stop-Icon.

Name	Parameter	Beschreibung
<i>DrawString</i>	Text als <u>String</u> , X als <u>Integer</u> , Y als <u>Integer</u> , [WrapWidth als <u>Integer</u>]	Zeichnet den Text an der angegebenen Position. Der Parameter Y legt die untere Seite des Textes fest. Der optionale Parameter WrapWidth gibt an, ab welcher Breite (in Pixeln) der Text umbrochen werden soll.
<i>FillOval</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u>	Zeichnet ein mit der aktuellen Farbe gefülltes Oval.
<i>FillPolygon</i>	Points() als <u>Integer</u>	Zeichnet ein Polygon mit den Werten aus dem übergebenen Array. Ungerade Felder des Arrays sind X-Koordinaten, gerade Felder des Arrays sind Y-Koordinaten. Das Polygon wird mit der aktuellen <u>Vordergrundfarbe</u> gefüllt.
<i>FillRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u>	Zeichnet ein mit der aktuellen Farbe gefülltes Rechteck.
<i>FillRoundRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Breite als <u>Integer</u> , Höhe als <u>Integer</u> , OvalWidth als <u>Integer</u> , OvalHeight als <u>Integer</u>	Zeichnet ein mit der aktuellen <u>Vordergrundfarbe</u> gefülltes, abgerundetes Rechteck.
<i>NextPage</i>		Wenn ein Graphics-Objekt gedruckt wird, schließt diese Methode die aktuelle Seite ab und sorgt dafür, daß eine neue Seite angelegt wird.
<i>StringHeight</i>	Text als <u>String</u> , WrapWidth als <u>Integer</u>	Liefert die Höhe des Textes (anhand aktueller Schriftart und -Größe) und ein WrapWidth in Pixeln. Der WrapWidth-Parameter legt die Breite (in Pixeln) fest, ab der der Text umbrochen werden soll.
<i>StringWidth</i>	Text als <u>String</u>	Liefert die Breite des Textes (anhand aktueller Schriftart und -Größe) in Pixeln.

Hinweise

Normalerweise verwendet REALbasic die Quartz-Grafik-Engine unter Mac OS X, wenn Sie Methoden der **Graphics**-Klasse wie z.B. DrawLine, DrawRect, DrawOval, und so weiter aufrufen. Die Quartz-Grafik-Engine verwendet Anti-Aliasing, damit Linien weicher aussehen. Der Nachteil ist, daß es langsamer ist, als eine Linie mit der älteren Apple-Technologie zu zeichnen. Bei den meisten Applikationen ist die Quartz-Engine schnell genug und liefert schönere Ergebnisse unter Mac OS X. Wenn Sie jedoch mehr Geschwindigkeit benötigen, können Sie die Zeichen-Methoden unter Mac OS X beschleunigen, indem Sie die Quartz-Grafik-Engine ausschalten und stattdessen die QuickDraw-Engine verwenden. Dies erreichen Sie, indem Sie die UseOldRenderer-Eigenschaft der **Graphics**-Klasse auf True setzen, bevor Sie mit dem Zeichnen beginnen.

Beispiele

Dieses Beispiel verwendet den Event-Handler „Paint“ eines Canvas Steuerelementes, um den Text „The quick brown fox“ in fett-kursiver Helvetica, 18 Punkt mit 50 Pixeln Abstand von der Oberkante und 10 Pixeln Abstand von der linken Seite des Steuerelements zu zeichnen.

```
Sub Paint(g As Graphics)
    g.Bold=True
    g.Italic=True
    g.TextFont="Helvetica"
    g.TextSize=18
    g.DrawString "The quick brown fox", 10, 50
End Sub
```

Dieses Beispiel weist einer Variablen die Farbe eines bestimmten Pixels aus einem Canvas Steuerelement zu.

```
Dim c as Color
c=Canvas1.Graphics.Pixel(10,10)
```

Dieses Beispiel weist einem bestimmten Pixel eines Canvas Steuerelements einen RGB-Farbwert zu.

```
Canvas1.Graphics.Pixel(10,10)=RGB(100,105,225)
```

Dieses Beispiel zeichnet ein Dreieck in ein Canvas Steuerelement:

```
Dim Points(6) as Integer
```

```

Points(1)=10 //X von Punkt 1
Points(2)=10 //Y von Punkt 1
Points(3)=75 //X von Punkt 2
Points(4)=30 //Y von Punkt 2
Points(5)=10 //X von Punkt 3
Points(6)=125 //Y von Punkt 3
Canvas1.Graphics.ForeColor=RGB(100,200,255)
Canvas1.Graphics.FillPolygon Points

```

Folgendes Beispiel geht davon aus, daß Sie eine Resource-Datei in das Projekt-Fenster importiert haben, die eine PICT-Resource mit ID 129 beinhaltet. Folgende Code-Zeile im Open-Event des Canvas-Steuerelements weist das Bild der Backdrop-Eigenschaft des Steuerelements zu:

```
me.backdrop=app.resourceFork.getpicture(129)
```

Hinweis: Zugriff auf den Resource-Fork wird nur auf dem Macintosh unterstützt. Prüfen Sie den Wert der Konstante TargetMacOS, bevor Sie versuchen, auf den Resourcefork zuzugreifen.

Hinweis: Jede Resource-Datei, die Sie dem Projekt-Fenster hinzufügen, muß „Resources“ heißen.

Siehe auch

Canvas Steuerelement.

Group2D Klasse

Wird zum Gruppieren von Object2D-Objekten verwendet.

Übergeordnete Klasse

Object2D

Eigenschaften

Name	Typ	Beschreibung
<i>Count</i>	<u>Integer</u>	Die Anzahl von <u>Object2D</u> -Objekten, die die Gruppe enthält.
<i>Item</i>	<u>Object2D</u>	Parameter ist <i>Index</i> als <u>Integer</u> . Liefert das durch <i>Index</i> festgelegte <u>Object2D</u> -Objekt.

Methoden

Name	Parameter	Beschreibung
<i>Append</i>	Object als <u>Object2D</u>	Fügt das übergebene Objekt hinzu.
<i>Insert</i>	Index als <u>Integer</u> Object als <u>Object2D</u>	Fügt das übergebene Objekt an der festgelegten Position ein.
<i>Remove</i> ODER	Index als <u>Integer</u>	Entfernt ein durch seinen Index festgelegtes Objekt.
<i>Remove</i>	Object als <u>Object2D</u>	Entfernt ein durch seine Referenz festgelegtes Objekt.

Hinweise

Verwenden Sie die Objects-Eigenschaft der Picture-Klasse, um ein **Group2D**-Objekt mit einem Bild zu verknüpfen und die DrawObject-Methode der Graphics-Klasse, um das Objekt zu zeichnen.

Ein **Group2D** ist ein Container für Object2D-Figuren (inklusive anderer **Group2Ds**). Wenn ein **Group2D** rotiert, verschoben oder skaliert wird, wird sein gesamter Inhalt passend aktualisiert. Das bedeutet, daß alle Objekte das gleiche Koordinatensystem verwenden, Sie können aber eine gesamte Gruppe von Objekten schnell transformieren, indem Sie einfach deren Gruppe transformieren.

Beispiel

Diese Methode fügt dem **Group2D** Objekt ein PixmapShape und ein StringShape hinzu und stellt dieses im Fenster dar. Das Bild „h1“ wurde in das Projekt-Fenster gedraggt.

```

Dim px as PixmapShape
Dim s as StringShape
Dim d as Group2D
d=new group2d
px=new PixmapShape(h1)

```

```
d.append px
s=New StringShape
s.y=70
s.Text="Das nenne ich ein ECHTES Auto!"
s.TextFont="Helvetica"
s.Bold=true
d.append s
graphics.drawobject d,100,100
```

Siehe auch

[ArcShape](#), [CurveShape](#), [FigureShape](#), [FolderItem](#), [Graphics](#), [OvalShape](#), [Picture](#), [PixmapShape](#), [RectShape](#), [RoundRectShape](#), [StringShape](#) Klassen.

Group3D Klasse

Wird zum Gruppieren von Object3D-Objekten verwendet.

Übergeordnete Klasse

[Object3D](#)

Methoden

Name	Parameter	Beschreibung
<i>Count</i>		Anzahl von Object3D -Objekten im gruppierten Objekt. Gibt ein Integer zurück.
<i>Item</i>	Index als Integer	Das Objekt <i>Item</i> im gruppierten Objekt. Das erste <i>Item</i> hat den Index 0. Gibt ein Object3D -Objekt zurück.
<i>Append</i>	obj als Object3D	Fügt das Objekt <i>obj</i> dem gruppierten Objekt hinzu.
<i>Remove</i>	Index als Integer ODER obj als Object3D	Entfernt ein Objekt aus der Gruppe. Entfernt das Objekt <i>Index</i> oder das durch <i>obj</i> festgelegte Objekt. Beachten Sie, daß Sie, da die Remove-Methode überladen wird, das Argument immer in Klammern einschließen müssen.
<i>Update</i>		Aktualisiert die Position und Ausrichtung des Group3D-Objekts.

Hinweise

Die Group3D-Klasse wird zum Gruppieren von Objekten verwendet, ist aber auch ein Object3D-Objekt, das Ihnen die Möglichkeit bietet, alle Positionierungs- und Ausrichtungs-Funktionen die es bei einem Object3D gibt, zu verwenden. Wenn Sie einer Gruppe ein Objekt hinzufügen, bewegt es sich und rotiert mit der Gruppe, was jedoch etwas aufwendig ist, so daß Sie Gruppen nur sehr sparsam auf diese Weise einsetzen sollten. Die Position und Ausrichtung der gruppierten Objekte wird aktualisiert, um zur Gruppe zu passen, wenn Sie *Update* aufrufen oder wenn die Gruppe in einer Ansicht gezeichnet wird. Sie können ein Objekt aus einer Gruppe entweder über seinen Index oder über die Objekt-Referenz selbst entfernen.

Siehe auch

[Vector3D](#), [Quaternion](#), [Object3D](#), [Light3D](#), [Bounds3D](#) Klassen; [RB3DSpace](#) Steuerelement.

GroupBox Steuerelement



Zeichnet einen Gruppierungskasten für Radiobuttons. Ein Zweck ist die Kennzeichnung von zusammengehörigen Radiobuttons. Nur einer der von der GroupBox umschlossenen [Radiobuttons](#) kann ausgewählt sein; REALbasic deselektiert automatisch alle anderen Radiobuttons in der GroupBox, wenn der Anwender einen selektiert. Radiobuttons müssen von einer GroupBox komplett umrandet sein, damit diese als eine Radiobutton-Gruppe behandelt wird.

Ein GroupBox-Steuerelement kann auch für die Organisation von anderen Interface-Elementen verwendet werden, hat aber keinen Einfluß auf deren Verhalten.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>Bold</i>	<u>Boolean</u>	Wenn True, so wird die Überschrift fett dargestellt.
<i>Caption</i>	<u>String</u>	Die Überschrift der GroupBox.
<i>Italic</i>	<u>Boolean</u>	Wenn True, so wird die Überschrift kursiv dargestellt.
<i>TextFont</i>	<u>String</u>	Name der für die Überschrift verwendeten Schrift.
<i>TextSize</i>	<u>Integer</u>	Schriftgröße der Überschrift.
<i>Underline</i>	<u>Boolean</u>	Wenn True, so wird die Überschrift unterstrichen.

Events

Name	Parameter	Beschreibung
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der GroupBox an der Position x,y gedrückt. Gibt ein Boolean zurück. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen. Die Koordinaten x und y sind lokal zum Steuerelement. Der Punkt 0,0 ist die linke obere Ecke des gesamten Steuerelements (links von der Beschriftung und über der Box, die gezeichnet wurde).
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der GroupBox gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der GroupBox an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.

Hinweise

Wenn die Caption-Eigenschaft ein Und-Zeichen enthält, wird dieses nur angezeigt, wenn ein weiteres Und-Zeichen folgt. Dies ist so, damit sich Macintosh- und Windows-Programme gleich verhalten.

Einige Versionen vom Mac OS (8.0, 8.1 und OS X Public Beta) haben einen Fehler, der dazu führt, daß eine Groupbox alle Steuerelemente dahinter verdeckt. Dies können Sie umgehen, indem Sie mit dem Befehl „Format/nach hinten bewegen“ jede Groupbox hinter die Steuerelemente befördern, die verdeckt werden.

Beispiel

Siehe RadioButton Steuerelement.

Siehe auch

RadioButton Steuerelement.

GuessJapaneseEncoding Funktion

Errät die verwendete Textkodierung für einen Textausschnitt eines japanischen Textes (2022_JP, EUC, S-JIS, oder normales US_ASCII). Wenn dieser Funktion ein String übergeben wird, dessen Kodierung REALbasic bereits kennt, liefert es diese Kodierung. Wenn die Kodierung unbekannt ist, geht es davon aus, daß es eine japanische Schrift ist und versucht zu erkennen, um welche es sich handelt.

Syntax

result=GuessJapaneseEncoding(text)

Teil	Typ	Beschreibung
<i>result</i>	<u>TextEncoding</u>	Die vermutete Textkodierung

Teil	Typ	Beschreibung
<i>text</i>	<u>String</u>	Der Text, dessen Kodierung erraten werden soll

Siehe auch

ConvertEncoding, DefineEncoding, Encoding Funktionen; TextEncoding Klasse, Encodings Objekt.

Hex Funktion

Liefert die hexadezimale Schreibweise der angegebenen Zahl im übergebenen String.

Syntax

Ergebnis = **Hex**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die übergebene Zahl in hexadezimaler Schreibweise.
<i>Wert</i>	<u>Integer</u>	Die in hexadezimaler Schreibweise auszugebende Zahl.

Hinweis

Wenn der Wert keine ganze Zahl ist, wird der Nachkommaanteil nicht beachtet.

Sie können Binär-, Hexadezimal- oder Oktal-Zahlen festlegen, indem Sie der Zahl den &-Operator und den Buchstaben für die Zahlen-Basis voranstellen. Der Buchstabe „b“ legt die Zahl als Binär-, „h“ als Hexadezimal- und „o“ als Oktal-Zahl fest.

Notiz zur Kompatibilität mit Visual Basic: Visual Basic rundet in diesem Fall den Wert auf die nächste ganze Zahl. Möglicherweise wird das Verhalten der Hex-Funktion in einer zukünftigen Version geändert.

Beispiele

Dies sind Beispiele verschiedener Zahlen, die in hexadezimaler Schreibweise ausgegeben werden:

```
Dim hexVersion As String
hexVersion=Hex(5) //liefert "5"
hexVersion=Hex(75) //liefert "4B"
hexVersion=Hex(256) //liefert "100"
```

Siehe auch

& Operator; Bin und Oct Funktionen.

HighlightColor Funktion

Liefert die Farbe, die der Anwender als Hervorhebungsfarbe eingestellt hat.

Syntax

Ergebnis = **HighlightColor**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Die eingestellte Hervorhebungsfarbe.

Hinweis

Verwendet der Anwender Mac OS 8 – 9.2 oder neuer, so wird die Hervorhebungsfarbe über das Kontrollfeld „Erscheinungsbild“ eingestellt. Unter Mac OS X wird die Hervorhebungsfarbe im Kontrollfeld Allgemein bei den Systemeinstellungen vorgenommen.

Siehe auch

DarkBevelColor, DarkTingeColor, FillColor, FrameColor, LightBevelColor, TextColor Funktionen; Color Datentyp.

HSV Funktion

Liefert ein Color-Objekt basierend auf dem HSV-Farbmodell (hue, saturation, value – Farbtön, Sättigung, Helligkeit).

Syntax

result=**HSV**(hue, saturation, value)

Teil	Typ	Beschreibung
<i>result</i>	<u>Color</u>	Ein Objekt, das die Farbe, basierend auf den Werten <i>bue</i> , <i>saturation</i> und <i>value</i> repräsentiert.
<i>bue</i>	<u>Double</u>	Der Farbtton der Farbe.
<i>saturation</i>	<u>Double</u>	Die Sättigung der Farbe.
<i>value</i>	<u>Double</u>	Die Helligkeit der Farbe.

Hinweis

Die HSV-Funktion liefert ein Color-Objekt basierend auf den übergebenen Werten von Hue, Saturation und Value- Diese Werte werden als Doubles übergeben und haben einen Wert zwischen 0 und 1.

Farben können ebenso mit dem RGB- oder CMY-Modell erzeugt werden.

Beispiele

Dieses Beispiel verwendet die **HSV** Funktion, um der FillColor-Eigenschaft eines Rectangle Steuerelements eine Farbe zuzuweisen.

```
Rectangle1.FillColor=HSV(.8,.5,.75)
```

Siehe auch

Color Datentyp, RGB, CMY, SelectColor Funktionen; &c-Operator.

HTTPSocket Klasse

Wird zum Senden und Empfangen von Daten über das HTTP-Protokoll verwendet.

Übergeordnete Klasse

TCPSocket

Eigenschaften

Keine

Methoden

Name	Parameter	Beschreibung
<i>ClearRequestHeaders</i>		Löscht alle gesetzten Request Headers. Weitere Informationen finden Sie in der Beschreibung der SetRequestHeaders-Methode.
<i>EncodeFormData</i>	Form als <u>Dictionary</u>	Liefert einen <u>String</u> . Übernimmt die Schlüssel/Wert-Paare aus dem angegebenen Dictionary und formatiert diese in einen URL-codierten String um. Das Ergebnis kann dann in der URL verwendet werden, um Daten an ein Formular mit einer GET-Anweisung zu senden oder als Inhalt für ein POST.
<i>Get</i>	URL als <u>String</u> [File als <u>FolderItem</u>]	Fordert den Inhalt der angegebenen URL an. Das PageReceived-Event wird ausgeführt, wenn die Daten empfangen wurden. Wenn ein <u>FolderItem</u> übergeben wurde, werden die Daten in diese Datei heruntergeladen. Das DownloadComplete-Event wird ausgeführt, wenn die Antwort-Daten empfangen wurden.
<i>Post</i>	URL als <u>String</u> [File als <u>FolderItem</u>]	Sendet einen POST-Befehl an den Web-Server. Das PageReceived-Event wird ausgeführt, wenn die Antwort-Daten empfangen wurden. Wenn ein <u>FolderItem</u> übergeben wurde, werden die Daten in diese Datei heruntergeladen. Das DownloadComplete-Event wird ausgeführt, wenn die Antwort-Daten empfangen wurden.

Name	Parameter	Beschreibung
<i>SetFormData</i>	Form als <u>Dictionary</u>	Übernimmt die Schlüssel/Wert-Paare aus dem angegebenen Dictionary und formatiert diese in einen URL-codierten String um. Diese Methode stellt außerdem POST ein. Nachdem man diese Methode verwendet hat, kann das Formular mit der POST-Anweisung an den Server gesendet werden.
<i>SetPostContent</i>	Content als <u>String</u> , ContentType als <u>String</u>	Legt die Inhaltsdaten und den Inhaltstyp fest, die für einen POST-Befehl an den Server gesendet werden sollen.
<i>SetRequestHeader</i>	Name als <u>String</u> , Value als <u>String</u>	Legt den Wert für einen Request Header fest. Request Header werden zusammen mit der Aktion (GET, POST, etc.) an den Server gesendet. Sie können Daten enthalten, die der Server verwenden kann, um Ihre Anforderung zu verarbeiten. Einige Beispiele für gebräuchliche Request Headers sind Dinge wie zurückverweisende Seiten, Klient-Name, OS CPU und so weiter.

Events

Name	Parameter	Beschreibung
<i>Connected</i>		Wird ausgeführt, wenn die Verbindung mit dem Web-Server hergestellt ist.
<i>DownloadComplete</i>	URL als <u>String</u> , HTTPStatus als <u>Integer</u> , Headers als <u>Dictionary</u> , File als <u>FolderItem</u>	Wird ausgeführt, wenn ein Download komplett ist, nachdem die DownloadURL-Methode verwendet wurde. URL enthält die URL, die heruntergeladen wurde. File ist das FolderItem, in das die Daten heruntergeladen wurden und Headers ist ein Dictionary, aus den HTTP-Headern, die vom Server geliefert wurden. Der HTTPStatus-Code wird diesem Event ebenfalls übergeben. Diese Codes werden für Meldungen wie „Page not found“ (error 404), usw. verwendet.
<i>Error</i>	Code als <u>Integer</u>	Wird ausgeführt, wenn ein Fehler auftritt. Alle Standard-TCPsocket-Fehler-Codes werden in Code übergeben. Ein zusätzlicher Fehler-Code ist: 1 - Download-Datei konnte nicht erzeugt werden.
<i>HeadersReceived</i>	Headers als <u>Dictionary</u>	Wird ausgeführt, wenn die Header von Server empfangen wurden.
<i>PageReceived</i>	URL als <u>String</u> , HTTPStatus als <u>Integer</u> , Headers als <u>Dictionary</u>	Wird ausgeführt, wenn eine neue Seite vom Server empfangen wurde. Es liefert die URL, die angefordert wurde und die HTTP-Header vom Web-Server. Der HTTPStatus-Code wird diesem Event ebenfalls übergeben. Diese Codes werden für Meldungen wie „Page not found“ (error 404), usw. verwendet.
<i>ReceiveProgress</i>	BytesReceived als <u>Integer</u> , TotalBytes als <u>Integer</u>	Wird während des Download-Prozesses periodisch aufgerufen. Es liefert die Anzahl aktuell empfangener Bytes und die Gesamtzahl zu empfangender Bytes (sofern verfügbar).
<i>SendProgress</i>	BytesSent als <u>Integer</u> , BytesLeft als <u>Integer</u>	Wird periodisch während des Upload-Prozesses ausgeführt. Es liefert die Abzahl der versendeten und die Anzahl der noch zu versendenden Bytes.

Hinweise

Die HTTPSocket-Klasse verwendet man dazu, Daten über das HTTP-Protokoll zu senden und zu empfangen. Da es sich hierbei um eine Unterklasse der TCPsocket-Klasse handelt, sind ebenso alle Standard-Socket-APIs verfügbar. Um eine gesicherte Kommunikation über HTTP aufzubauen, sollten Sie stattdessen die HTTPSecureSocket-Klasse verwenden. Sie ist so weit identisch, mit Ausnahme, daß sie statt von TCPsocket von SSLsocket abgeleitet ist. Hiermit werden alle Möglichkeiten für verschlüsselte Kommunikation für HTTPSecureSocket-Objekte verfügbar gemacht.

Wenn Sie den optionalen Parameter Timeout der Methoden Get, GetHeaders oder Post verwenden, wird die Seite synchron übertragen, beziehungsweise empfangen. Wenn Sie den Timeout auf Null setzen, warten Sie unendlich oder bis ein Fehler auftritt.

Um den synchronen Modus zu verwenden, geben Sie einfach den optionalen Parameter an, wie in dem Beispiel gezeigt:

```
Dim http as New HTTPSocket
```

`MsgBox http.Get("www.realsoftware.com",30)`

Wenn Sie den synchronen Modus verwenden, können Sie die Yield-Eigenschaft auf True setzen, wenn Sie neben dem Warten auch andere Events verarbeiten lassen wollen. Fügen Sie einfach diese Zeile hinzu,

`http.Yield=True`

um Hintergrundprozesse zu erlauben, während der Socket wartet.

Beispiele

Folgendes Beispiel liefert die angegebene URL:

```
Dim socket1 as HTTPSocket
socket1=New HTTPSocket
socket1.get "http://www.application-systems.de/"
```

Folgendes Beispiel sendet ein einfaches Formular:

```
Dim form as Dictionary
Dim socket1 as HTTPSocket
socket1=New HTTPSocket

// Formular-Objekt erzeugen und ausfüllen
form = New Dictionary
form.value("Vorname") = "Peter"
form.value("Nachname") = "Braun"

// Der Socket für das POSTen des Formulars einrichten
socket1.setFormData form
socket1.post "http://www.myformlocation.com/form.php"
```

Siehe auch

SocketCore, TCPSocket, HTTPSecureSocket Klassen.

HTTPSecureSocket Klasse

Wird für sichere Übertragungen über das HTTP-Protokoll unter Verwendung des SSL- oder TLS-Protokoll verwendet.

Übergeordnete Klasse

SSLSocket

Eigenschaften

Keine

Methoden

Name	Parameter	Beschreibung
<i>ClearRequestHeaders</i>		Löscht alle gesetzten Request Headers. Weitere Informationen finden Sie in der Beschreibung der SetRequestHeaders-Methode.
<i>EncodeFormData</i>	Form als <u>Dictionary</u>	Liefert einen <u>String</u> . Übernimmt die Schlüssel/Wert-Paare aus dem angegebenen Dictionary und formatiert diese in einen URL-codierten String um. Das Ergebnis kann dann in der URL verwendet werden, um Daten an ein Formular mit einer GET-Anweisung zu senden oder als Inhalt für ein POST.
<i>Get</i>	URL als <u>String</u> [File als <u>FolderItem</u>]	Fordert den Inhalt der angegebenen URL an. Das PageReceived-Event wird ausgeführt, wenn die Daten empfangen wurden. Wenn ein <u>FolderItem</u> übergeben wurde, werden die Daten in diese Datei heruntergeladen. Das Download-Complete-Event wird ausgeführt, wenn die Antwort-Daten empfangen wurden.
<i>Post</i>	URL als <u>String</u> [File als <u>FolderItem</u>]	Sendet einen POST-Befehl an den Web-Server. Das PageReceived-Event wird ausgeführt, wenn die Antwort-Daten empfangen wurden. Wenn ein <u>FolderItem</u> übergeben wurde, werden die Daten in diese Datei heruntergeladen. Das DownloadComplete-Event wird ausgeführt, wenn die Antwort-Daten empfangen wurden.

Name	Parameter	Beschreibung
<i>SetFormData</i>	Form als <u>Dictionary</u>	Übernimmt die Schlüssel/Wert-Paare aus dem angegebenen Dictionary und formatiert diese in einen URL-codierten String um. Diese Methode stellt außerdem POST ein. Nachdem man diese Methode verwendet hat, kann das Formular mit der POST-Anweisung an den Server gesendet werden.
<i>SetPostContent</i>	Content als <u>String</u> , ContentType als <u>String</u>	Legt die Inhaltsdaten und den Inhaltstyp fest, die für einen POST-Befehl an den Server gesendet werden sollen.
<i>SetRequestHeader</i>	Name als <u>String</u> , Value als <u>String</u>	Legt den Wert für einen Request Header fest. Request Header werden zusammen mit der Aktion (GET, POST, etc.) an den Server gesendet. Sie können Daten enthalten, die der Server verwenden kann, um Ihre Anforderung zu verarbeiten. Einige Beispiele für gebräuchliche Request Headers sind Dinge wie zurückverweisende Seiten, Klient-Name, OS CPU und so weiter.

Events

Name	Parameter	Beschreibung
<i>Connected</i>		Wird ausgeführt, wenn die Verbindung mit dem Web-Server hergestellt ist.
<i>DownloadComplete</i>	URL als <u>String</u> , HTTPStatus als <u>Integer</u> , Headers als <u>Dictionary</u> , File als <u>FolderItem</u>	Wird ausgeführt, wenn ein Download komplett ist, nachdem die Get-Methode verwendet wurde. URL enthält die URL, die heruntergeladen wurde. File ist das FolderItem, in das die Daten heruntergeladen wurden und Headers ist ein Dictionary, aus den HTTP-Headern, die vom Server geliefert wurden. Der HTTPStatus-Code wird diesem Event ebenfalls übergeben. Diese Codes werden für Meldungen wie „Page not found“ (error 404), usw. verwendet.
<i>Error</i>	Code als <u>Integer</u>	Wird ausgeführt, wenn ein Fehler auftritt. Alle Standard-TCPsocket-Fehler-Codes werden in Code übergeben. Ein zusätzlicher Fehler-Codes ist: 1 - Download-Datei konnte nicht erzeugt werden.
<i>PageReceived</i>	URL als <u>String</u> , HTTPStatus als <u>Integer</u> , Headers als <u>Dictionary</u> , Content als <u>String</u>	Wird ausgeführt, wenn eine neue Seite vom Server empfangen wurde. Es liefert die URL, die angefordert wurde und die HTTP-Header vom Web-Server. Der HTTPStatus-Code wird diesem Event ebenfalls übergeben. Diese Codes werden für Meldungen wie „Page not found“ (error 404), usw. verwendet.
<i>ReceiveProgress</i>	BytesReceived als <u>Integer</u> , TotalBytes als <u>Integer</u>	Wird während des Download-Prozesses periodisch aufgerufen. Es liefert die Anzahl aktuell empfangener Bytes und die Gesamtzahl zu empfangender Bytes (sofern verfügbar).

Hinweise

Verwenden Sie die **HTTPSecureSocket**-Klasse, um gesicherte Kommunikation über das HTTP-Protokoll durchzuführen. Sie ist identisch zur HTTPSocket-Klasse, mit Ausnahme, daß sie von der SSLSocket-Klasse an Stelle der TCPSocket-Klasse abgeleitet ist. Das bedeutet, daß Sie die Secure-Eigenschaft von SSLSocket verwenden können, um zu HTTPS zu wechseln. Wenn Sie Secure auf True setzen, ändert sich der Default-Port von 80 auf 443.

Beispiel

Folgendes Beispiel sendet ein einfaches Formular:

```

Dim form as Dictionary
Dim socket1 as HTTPSecureSocket
Socket1.Secure=True

// Formular-Objekt erzeugen und ausfüllen
form = New Dictionary
form.value("Vorname") = "Peter"
form.value("Nachname") = "Braun"

// Der Socket für das POSTen des Formulars einrichten
socket1.setFormData form

```

```
socket1.post "https://www.myformlocation.com/form.php"
```

Siehe auch

[SocketCore](#), [TCPSocket](#), [SSLSocket](#), [HTTPSocket](#) Klassen.

If...Then...Else Befehl

Führt in Abhängigkeit von einem boole'schen Ausdruck eine Reihe von Befehlen aus.

Syntax

If Bedingung **Then**

Anweisungen

[**Elseif** Bedingung-n **Then**

elseifAnweisungen]...

[**Else**

elseAnweisungen]

End [**If**]

ODER

If Bedingung **Then** Anweisung [**Else**] [Anweisung]

Teil	Beschreibung
<i>Bedingung</i>	Benötigt. Ein <u>boole'scher</u> , numerischer oder <u>String</u> -Ausdruck, der sich zu <u>True</u> oder <u>False</u> auflöst.
<i>Anweisungen</i>	Optional. Einer oder mehrere Befehle, die ausgeführt werden, falls <i>Bedingung</i> <u>True</u> ist.
<i>Bedingung-n</i>	Optional. Weitere Bedingungen wie <i>Bedingung</i> .
<i>elseifAnweisungen</i>	Optional. Ein oder mehrere Befehle, die ausgeführt werden, wenn die damit verbundene <i>Bedingung-n</i> <u>True</u> ist.
<i>elseAnweisungen</i>	Optional. Einer oder mehrere Befehle, die ausgeführt werden, falls keine zuvorige <i>Bedingung</i> oder <i>Bedingung-n</i> <u>True</u> ist.

Hinweise

Wenn ein **If**-Befehl ausgeführt wird, wird die *Bedingung* geprüft. Wenn diese Bedingung True ist, werden die damit verbundenen (dem **Then** folgende) Befehle ausgeführt. Ist die Bedingung False und folgt ein **Else**-Befehl, so werden dessen Befehle ausgeführt. Ist die Bedingung False und gibt es keinen **Else**-Befehl oder steht davor ein **Elseif**-Befehl, so wird die Bedingung des **Elseif**-Befehls geprüft. Nachdem die **Then**, **Elseif** oder **Else** folgenden Befehle ausgeführt wurden, wird die Anweisung in der dem **End If** folgenden Zeile ausgeführt.

Ein **If**-Befehl kann in einer Zeile geschrieben werden, vorausgesetzt, der Code, der den **Then**- und **Else**-Befehlen folgt, paßt in die gleiche Zeile. Wenn Sie diese Syntax verwenden, müssen Sie den **End If**-Befehl weglassen. Folgende Beispiele sind z.B. gültig:

```
If error=123 Then MsgBox "Ein Fehler ist aufgetreten."
```

```
If error=123 Then MsgBox "Ein Fehler ist aufgetreten." Else MsgBox "Egal!"
```

Folgendes ist nicht erlaubt, da der **Then**-Fall zwei Zeilen benötigt:

```
If error=123 Then Beep MsgBox "Ein Fehler ist aufgetreten."
```

Verwenden Sie die Syntax aus dem nächsten Beispiel.

Beispiele

Dieses Beispiel zeigt einen **If**-Befehl.

```
If error=-123 Then
    Beep
    MsgBox "Ups! Ein Fehler trat auf."
End If
```

Dieses Beispiel zeigt einen **If**-Befehl, der **Elseif**- und **Else**-Befehle enthält.

```
Dim theZahl As Integer
```

```

Dim digits As Integer
theZahl=33
If theZahl<10 Then
    digits=1
ElseIf theZahl<100 Then
    digits=2
Else
    digits=3
End If

```

Siehe auch

Select Case Befehl.

IllegalCastException Laufzeitfehler

Wird durch den Versuch verursacht, den Typ eines Objekts zu ändern und sendet ihm dann eine Nachricht, daß sein Original-Typ es nicht verarbeiten kann.

Übergeordnete Klasse

RuntimeException Klasse.

Hinweise

Ein IllegalCastException Fehler tritt auf, wenn Sie versuchen, den Typ eines Objektes in einen anderen zu ändern und eine Methode aufrufen oder auf eine Eigenschaft zugreifen, die es nur beim Original-Typ gibt. Wenn Sie z.B. einen BevelButton in einen PushButton ändern (wie im Beispiel unten) und dann die Push-Methode des PushButtons aufrufen, wird ein IllegalCastException Fehler auftreten, da ein BevelButton kein PushButton ist und daher keine Push-Methode besitzt.

Beispiel

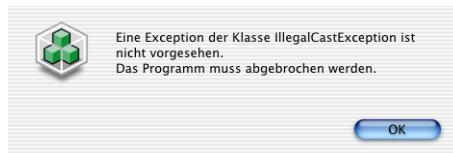
Dieses Beispiel versucht, einen BevelButton in einen PushButton zu ändern und dann die Push-Methode der PushButton-Klasse aufzurufen. Dieses Beispiel setzt voraus, daß es einen BevelButton namens „BevelButton1“ im gleichen Fenster gibt, in dem der Code aufgerufen wird:

```

Dim c as Control
c = BevelButton1
PushButton(c).Push

```

Wenn Sie diesen Code in der IDE testen, wird die Ausführung bei der letzten Zeile stoppen und einen IllegalCastException Fehler in Ihrem Code-Editor anzeigen. Wenn Sie den Code in einer eigenständigen Applikation testen, wird die Applikation eine Fehler-Meldung anzeigen und sich beenden, wenn der Anwender auf OK klickt:



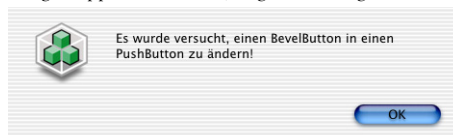
Sie können den Fehler durch Hinzufügen eines Exception Blocks in Ihren Code verarbeiten:

```

Dim c as Control
c = BevelButton1
PushButton(c).Push
Exception err
If err isA IllegalCastException then
    MsgBox "Es wurde versucht, einen BevelButton in einen PushButton zu ändern!"
end if

```

Wenn Sie diesen Code in einer eigenständigen Applikation testen, zeigt die Messagebox eine etwas informativere Meldung an.



Der Hauptunterschied ist, daß wenn der Anwender diesen Dialog bestätigt, die Anwendung nicht beendet wird.

Und die Moral von der Geschicht: Versuchen Sie, Exception Fehler abzufangen.

Siehe auch

[RuntimeException](#) Klasse, [OutOfBoundsException](#) , [NilObjectException](#), [StackOverFlowException](#), [TypeMismatchException](#) Fehler, [Function](#), [Raise](#) Ausdruck, [Nil](#) Schlüsselwort, [Exception](#) Block.

ImageWell Steuerelement



Fügt einem Fenster ein ImageWell hinzu, welches ein PICT-Bild auf einem Macintosh oder ein BMP-Bild unter Windows anzeigen kann. Falls QuickTime auf dem Rechner des Anwenders installiert ist, können die Bildformate, die von QuickTime zusätzlich unterstützt werden, verwendet werden.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Eigenschaften

Name	Typ	Beschreibung
<i>Image</i>	Picture	Das im ImageWell angezeigte Bild. Das Bild muß auf einem Macintosh ein PICT sein.

Beispiele

Die folgende Code-Zeile im Open-Event eines ImageWells zeigt ein Bild eines Autos in diesem ImageWell an. (Das Bild muß zuvor in das Projekt-Fenster gedraggt worden sein.)

```
me.image=Porsche
```

Folgendes Beispiel implementiert Drag&Drop zwischen zwei ImageWells. Eine PICT-Datei kann vom Schreibtisch auf eines der ImageWells gedraggt werden und eines der ImageWells Bilder kann auf den Schreibtisch gedraggt werden, um einen Clip zu erzeugen.

Im Open-Event des ImageWells bewirken die beiden Befehle dem Steuerelement, daß es sowohl ein gedraggtes Bild als auch eine gedraggte Datei vom Typ PICT akzeptieren soll (der Dateityp „image/x-pict“), die zuvor in der Dateitypen-Dialogbox definiert wurde.

```
Me.AcceptPictureDrop
Me.AcceptFileDrop("image/x-pict")
```

Der DropObject-Event-Handler sind so aus:

```
If Obj.PictureAvailable then
  Me.Image=Obj.Picture
elseif Obj.FolderItemAvailable then
  Me.Image=Obj.FolderItem.OpenAsPicture
End if
```

Der MouseDown-Event-Handler sieht so aus:

```
Dim d as DragItem
d=NewDragItem(Me.left,Me.top,Me.width,Me.height)
d.picture=Me.Image
d.Drag //Dragging erlauben
```

Siehe auch

[RectControl](#) Klasse.

IndexOf Methode

Wird verwendet, um einen Wert in einem Array zu ermitteln. Liefert den Index des Elements, das den angegebenen Wert enthält.

Syntax

Ergebnis=**array.IndexOf**(TargetValue,[StartingIndex])

Teil	Beschreibung
Ergebnis	Der Index des Elements, das den angegebenen Wert enthält. Wurde kein Element gefunden, ist dies -1.
<i>array</i>	Das Array, das Sie durchsuchen wollen.
<i>TargetValue</i>	Der Wert, den Sie in dem Array finden möchten. Es muß vom gleichen Datentyp sein wie das Array.
StartingIndex	Optional. Das Array wird, beginnend mit dem angegebenen StartingIndex, durchsucht. Wenn Sie einen Index außerhalb der Arraygrenzen angeben, wird eine <u>OutOfBoundsException</u> ausgelöst.

Beispiele

Dieses Beispiel verwendet eine ComboBox, die mit den Namen der fünf Arbeitstage einer Woche gefüllt ist. Der ausgewählte Tag ist der Text-Eigenschaft zugewiesen. Es wird der Index des ausgewählten Elements ausgegeben.

```
Dim days() as String
days() = Array("Montag", "Dienstag", "Mittwoch", "Donnerstag", "Freitag")
Dim dayNumber as Integer
dayNumber = days.IndexOf(ComboBox1.Text)
If dayNumber >= 0 then
    MsgBox "Sie haben Tag Nummer " + Str(dayNumber)+" eingegeben"
else
    MsgBox "Sie haben keinen Namen eines Wochentags eingegeben."
end if
```

Siehe auch

Dim Befehl; Array, Split, Join, Ubound Funktionen; Append, Redim, Remove, Pop, IndexOf, Insert, Shuffle, Sort Methoden; ParamArray Schlüsselwort.

Input Methode

Konsolen-Applikationen können damit eine Zeile vom Terminal einlesen (stdin).

Syntax

Ergebnis=**Input**

Teil	Typ	Beschreibung
Ergebnis	<u>String</u>	Die Zeile, die über das Terminal eingegeben wurde. Das Zeilenende ist in diesem String nicht enthalten.

Hinweise

Input ist äquivalent zu einem Aufruf der ReadLine-Methode der StandardInputStream-Klasse.

Siehe auch

StandardInputStream, StandardOutputStream, ConsoleApplication Klassen; Print, StdIn Methoden; TargetHasGUI Konstante.

InvalidParentException Fehler

Übergeordnete Klasse

RuntimeException

Hinweise

Sie haben versucht, das Elternobjekt eines Steuerelements über die Parent-Eigenschaft der Control-Klasse zu ermitteln, das Elternobjekt befindet sich aber in einem anderen Fenster. Das Eltern-Steuerelement muß sich im gleichen Fenster wie das Steuerelement befinden oder, wenn das Elternobjekt kein Steuerelement ist, muß das Fenster, das das Steuerelement enthält, das Elternobjekt sein.

Siehe auch

Parent-Eigenschaft der Control-Klasse; Exception Block; RuntimeException Klasse.

Insert Methode

Fügt einem Array ein neues Element hinzu.

Syntax

`array.Insert index, Wert`

Teil	Art	Beschreibung
<i>array</i>	Array beliebiger Art	Das Array, dem der Wert hinzugefügt werden soll.
<i>index</i>	<u>Integer</u>	Die Position im <i>Array</i> , an der das neue Element eingefügt werden soll.
<i>Wert</i>	Gleiche Art wie das Array	Der Wert, der dem neuen Element zugewiesen werden soll.

Hinweis

Alle Objekte unter und inklusive *index* werden um eine Position nach unten verschoben, um für das neue Element Platz zu schaffen.

Die Methode **Insert** funktioniert nur mit eindimensionalen Arrays.

Beispiele

Dieses Beispiel fügt in das Array `aNames` ein neues Element an Position 5 ein und weist ihm den String „Bill“ zu.

```
aNames.Insert 5, "Bill"
```

Siehe auch

[Append](#) Methode; [IndexOf](#) Methode; [Dim](#), [Redim](#) Befehl; [Remove](#), [Sort](#) Methode; [UBound](#) Funktion.

InStr Funktion

Liefert die Position des ersten Auftreten eines Strings in einem anderen String zurück. Das erste Zeichen trägt die Nummer 1.

Syntax

`Ergebnis=InStr([start,] quelle, suche)`

ODER

`Ergebnis=StringVariable.InStr([start,] suche)`

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Position, an der <i>suche</i> in <i>quelle</i> gefunden wurde.
<i>start</i>	<u>Integer</u>	Optional: Position, ab der der Quell-String durchsucht wird. Wird nichts angegeben, so wird ab Position 1 (Anfang) gesucht.
<i>quelle</i>	<u>String</u>	Benötigt. Der zu durchsuchende String
<i>suche</i>	<u>String</u>	Benötigt. Der String, nach dem gesucht werden soll.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweise

Wenn der String *suche* nicht im String *quelle* gefunden werden kann, wird 0 (Null) zurückgeliefert. Wenn *suche* leer ist, wird *start* zurückgegeben. Das bedeutet, `InStr("Dies", "")` gibt 1 zurück und `InStr(3, "Dies", "")` gibt 3 zurück. **InStr** ist case-insensitiv und ignoriert die Groß-/Kleinschreibung der Strings. Selbst bei Zeichen mit Akzent oder nicht-römischem Alphabet.

Wenn Sie die Byte-Position des Strings *suche* innerhalb des Strings *quelle* suchen oder die Suche case-sensitiv sein soll, sollten Sie die Funktion [InStrB](#) verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **InStr**, um einen String in einem anderen String zu finden.

```
Dim first As Integer
first = InStr("This is a test", "t")
//liefert 1, da case-insensitiv
first = InStr("This is a test", "is")
//liefert 3
```



```
first = InStr(4, "This is a test", "is")
//liefert 6
first = InStr("This is a test", "tester")
//liefert 0
```

Siehe auch

[Asc](#), [Chr](#), [Left](#), [Len](#), [Mid](#), [Right](#), [StrComp](#) Funktionen.

InStrB Funktion

Liefert die Byte-Position des ersten Auftretens eines Strings in einem anderen String. Das erste Zeichen trägt die Nummer 1.

Syntax

Ergebnis=**InStrB**([start,] quelle, suche)

ODER

Ergebnis=StringVariable.**InStrB**([start], suche)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Byte-Position von <i>suche</i> in <i>quelle</i> .
<i>start</i>	<u>Integer</u>	Optional: Position, ab der der Quell- <u>String</u> durchsucht wird. Wird nichts angegeben, so wird ab Position 1 gesucht.
<i>quelle</i>	<u>String</u>	Benötigt. Der zu durchsuchende <u>String</u> .
<i>suche</i>	<u>String</u>	Benötigt. Der <u>String</u> , nach dem gesucht werden soll.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweise

Wenn der String *suche* nicht im String *quelle* gefunden werden kann, wird 0 (Null) zurückgeliefert. **InStrB** ist case-sensitiv; es behandelt *quelle* als eine Reihe von nackten Bytes. Es sollte anstelle von **InStr** verwendet werden, wenn es sich bei dem String um binäre Daten handelt oder wenn das Programm unter einem Ein-Byte-Zeichensatzsystem (z.B. US-System) läuft und Sie Case-Sensitivität benötigen.

Wenn Sie die Zeichenposition des Strings *suche* im String *quelle* suchen, sollten Sie die Funktion [InStr](#) verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **InStrB**, um einen String in einem anderen String zu finden.

```
Dim first As Integer
first = InStrB("This is a test", "T")
//liefert 1
first = InStrB("This is a test", "t")
//liefert 11
first = InStrB("This is a test", "is")
//liefert 3
first = InStrB(4, "This is a test", "is")
//liefert 6
first = InStrB("This is a test", "tester")
//liefert 0
first = InStrB("This Is a test", "Is")
//liefert 6
```

Siehe auch

[AscB](#), [ChrB](#), [InStr](#), [LeftB](#), [LenB](#), [MidB](#), [RightB](#), [StrComp](#) Funktionen.

Integer Datentyp

Ein **Integer** ist ein wesentlicher Datentyp in REALbasic. Ein Integer ist eine ganze Zahl zwischen $\pm 2,147,483,648$. In anderen Programmiersprachen ist der Integer von REALbasic ein Long Integer. Da Integer Zahlen sind, können Sie damit mathematische Berechnungen durchführen. Ein **Integer**-Wert belegt 4 Byte im Speicher.

Siehe auch

[Single](#), [Double](#), [String](#), [Boolean](#), [Color](#) Datentypen; [+](#), [-](#), [*](#), [/](#), [<](#), [<=](#), [=](#), [>=](#), [>](#), [<>](#), [\](#), [Mod](#), [Str](#), [Val](#), [Vartype](#), [IsNumeric](#) Funktionen; [Dim](#) Ausdruck.

InternetHeaders Klasse

Wird zur Verarbeitung von HTTP- und E-Mail-Headern verwendet.

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
AppendHeader	Name als String , Value als String	Hängt einen neuen Header an.
Count		Liefert die Anzahl der Header als Integer .
Delete	Name als String ODER Name als String , Index als Integer	Entfernt den angegebenen Header.
DeleteAllHeaders		Entfernt alle Header.
InternetHeaders		
Name	Index als Integer	Liefert einen String mit dem Namen der angegebenen Headers.
SetHeader	Name als String , Value als String	Setzt den Namen und den Wert eines Headers.
Source		Liefert einen String , der den reinen Quelltext des Headers wiedergibt.
Value	Name als String ODER Name als String , Index als Integer ODER Index als Integer	Liefert einen String der den Wert des angegebenen Headers enthält.

Siehe auch

[EmailHeaders](#), [EmailMessage](#), [EmailAttachment](#), [HTTPSocket](#), [POP3Socket](#), [SMTPSocket](#) Klassen.

IPCSocket Klasse

Wird für Interprozesskommunikation unter Mac OS X, Linux und Windows verwendet.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>IsConnected</i>	Boolean	Liefert True zurück wenn eine Verbindung besteht.
<i>LastErrorCode</i>	Integer	Der Fehler-Code des zuletzt aufgetretenen Fehlers. Um zu ermitteln, welcher Fehler aufgetreten ist, prüfen Sie diesen Wert gegen die Fehler-Konstanten, wie im Beispiel gezeigt.

Name	Typ	Beschreibung
Path	<u>String</u>	Ein Pfad im Unix-Pfad-Format zu einem Verzeichnis, in dem Sie Schreibzugriff haben, gefolgt von einem Namen für Ihren Socket. Der Name und der Pfad sind frei wählbar aber für maximale Kompatibilität sollten Sie etwas in der Art /tmp/meinProgramm oder /var/tmp/meinProgramm verwenden.

Methoden

Name	Parameter	Beschreibung
Close		Schließt die Verbindung. Rufen Sie diese Methode auf, wenn eine Sitzung beenden wollen.
Connect		Versucht eine Verbindung zum angegebenen Pfad aufzubauen.
Listen		Warten auf eine Verbindung.
Poll		Fragt den Socket nach Daten ab (Original: polls the socket for)
Read	Bytes als <u>Integer</u> , [Encoding als <u>TextEncoding</u>]	Liest die Anzahl Bytes, die mit der Größe des internen Puffers spezifiziert ist. Der optionale Parameter Encoding gibt Ihnen die Möglichkeit eine Kodierung dieser Daten festzulegen. Verwenden Sie das <u>Encodings</u> -Objekt um eine Kodierung anzugeben.
ReadAll	[Encoding als <u>TextEncoding</u>]	Liest alle verfügbaren Daten aus dem internen Puffer. Der optionale Parameter Encoding gibt Ihnen die Möglichkeit eine Kodierung dieser Daten festzulegen. Verwenden Sie das <u>Encodings</u> -Objekt um eine Kodierung anzugeben.
Write	Text als <u>String</u>	Schreibt den angegebenen Text auf den Socket, der dann verschickt wird.

Events

Name	Parameter	Beschreibung
Connected		Wird ausgelöst, wenn eine Verbindung aufgebaut wurde.
DataAvailable		Wird ausgelöst, wenn Daten zum Lesen verfügbar sind.
Error		Ein Fehler ist aufgetreten. Bitte prüfen Sie den Wert der LastErrorCode-Eigenschaft gegen die Fehler-Konstanten, um den genauen Fehler zu ermitteln. Wenn zum Beispiel: if <code>IPCSocket1.LastErrorCode = IPCSocket1.LostConnection</code> dann wurde die Verbindung aus irgendeinem Grund geschlossen oder die Gegenstelle wurde heruntergefahren.

Klassenkonstanten

Diese Konstanten beschreiben, welcher Fehler aufgetreten ist. Um den Fehler zu ermitteln, vergleichen Sie die LastErrorCode-Eigenschaft mit diesen Konstanten. Diese Konstanten entsprechen den Fehler-Codes der SocketCore-Klasse.

Fehler-Code	Klassenkonstante
100	OpenDriverError
102	LostConnection
103	NameResolutionError
105	AddressInUseError
106	InvalidStateError
107	InvalidPortError
108	OutOfMemoryError

Hinweise

Um Interprozesskommunikation zu verwenden, müssen Sie einen Pfad für den Socket angeben, mit dem Ihre Applikation kommunizieren soll. Dieser Pfad wird im Unix-Format angegeben und gibt ein Verzeichnis an, in dem Sie Schreibzugriff haben. Typischerweise etwa derart:

```
/tmp/meinProgrammName
```

oder

```
/var/tmp/meinProgrammName.
```

Nun können Sie versuchen, die Applikation über diesen Socket zu erreichen. Läuft eine weitere Instanz Ihres Programms auf dem Computer, so wird eine Verbindung zu diesem aufgenommen. Sollte keine weitere Instanz vorhanden sein, bekommen Sie einen Fehler 103, der in der `LastErrorCode`-Eigenschaft gesetzt wird. In dem Beispiel wird der Socket dann auf eine neue Verbindung lauschen, um sich mit der nächsten Instanz der Applikation zu verbinden.

Dieses Beispiel stellt sicher, daß Ihre Applikation mit der zweiten Instanz der Applikation in Verbindung tritt, sobald diese gestartet wird. Sie können dies sehen, wenn Sie eine `MsgBox` im `Connected`-Event anzeigen lassen.

Wenn die Verbindung aufgebaut ist, verhält sich die **IPCsocket** wie ein normaler TCPsocket. Möchten Sie die Verbindung schließen, rufen Sie einfach die `Close`-Methode auf.

Die **IPCsocket**-Klasse implementiert das „Readable“- und „Writable“-Klassen-Interface. Überschreiben Sie diese Methoden, so müssen die Methoden wie hier aufgeführt deklariert werden.

Ergebnis=Read(count, [enc])

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der Text, der zurückgegeben wird.
count	<u>Integer</u>	Anzahl der Bytes, die gelesen werden sollen.
enc	<u>TextEncoding</u>	Optionaler Parameter, der die Kodierung der gelesenen Daten angibt.

Das Readable Klassen-Interface hat zwei Methoden.

Name	Parameter	Beschreibung
EOF		Liefert <u>True</u> wenn das Ende des Streams erreicht ist. Liefert ebenfalls <u>True</u> , wenn der Socket geschlossen ist und keine weiteren Daten im internen Puffer vorhanden sind.
ReadError		Liefert <u>True</u> wenn ein Fehler aufgetreten ist.

Write(data)

Teil	Typ	Beschreibung
<i>data</i>	<u>String</u>	Der Text, der geschrieben werden soll.

Das Writable Klassen-Interface hat die zwei folgenden Methoden:

Name	Parameter	Beschreibung
Flush		Leert den Inhalt des internen Puffers und schreibt die verbleibenden Daten auf das Volume. Für Sockets und „SerialStreams“ wird <code>Poll</code> ein letztes mal aufgerufen und der Puffer gelöscht.
WriteError		Liefert <u>True</u> wenn ein Fehler aufgetreten ist.

Für weitere Informationen über Klassen-Interfaces lesen Sie bitte das Kapitel „Klassen-Interfaces“ im Benutzerhandbuch durch.

Beispiel

Das folgende Beispiel öffnet eine Verbindung zu einer weiteren Instanz der Applikation. Dieser Code befindet sich im `Open`-Event des Fensters. Der **IPCsocket** wurde über den Menüpunkt „Steuerelement hinzufügen/Beliebig/IPCsocket“ dem Fenster hinzugefügt.

```
' Pick the path that the IPCsocket should use.
IPCsocket1.Path = "/tmp/myAwesomeApplication"

' Let's see if we can connect to an application already
' on this path. If we can, then cool.
IPCsocket1.Connect
While True
' Poll the socket
IPCsocket1.Poll

' Check to see if we're connected, or if
' there is an error
If IPCsocket1.IsConnected or IPCsocket1.LastErrorCode <> 0 then
Exit
end

' Update the user interface
App.DoEvents
wend
```

```

' Check to see why we exited the loop.
If IPCSocket1.LastErrorCode = IPCSocket1.NameResolutionError then
' It's because no one else was listening, so let's
' just start listening.
IPCSocket1.Listen
End

```

Siehe auch

[SocketCore](#), [TCPSocket](#) Klassen.

Is Operator

Vergleicht zwei Objekt-Referenzen, um festzustellen, ob diese auf das gleiche Objekt verweisen.

Syntax

Ergebnis=Objekt1 Is Objekt2

Teil	Typ	Beschreibung
<i>Ergebnis</i>	Boolean	Beliebiger Container, der einen Boolean -Wert erwartet.
<i>Objekt1</i>	Object	Beliebiger Objekt-Name.
<i>Objekt2</i>	Object	Ein anderer Objekt-Name.

Hinweise

Der **Is**-Operator liefert [True](#), wenn Objekt1 und Objekt2 derzeit auf das gleiche Objekt verweisen. Es prüft die Identität, keinen Inhalt, so daß es nicht durch Vorhandensein eines Vergleichsoperators beeinflusst wird. Verwenden Sie es auf die gleiche Art wie den „=“-Operator.

Beispiel

Im folgenden Beispiel liefert der **Is**-Operator [True](#).

```

Dim w1,w2 as Window
w1=Window1
w2=Window1
If w1 Is w2 then
//mache hier irgendwas
Else
//mache hier irgendwas anderes
End if

```

Siehe auch

[IsA](#) Operator.

IsA Operator

Wird verwendet, um die Klasse einer bestimmten Objekt-Referenz festzustellen.

Syntax

Ergebnis=object IsA object class

Teil	Art	Beschreibung
<i>Ergebnis</i>	Boolean	Ein beliebiger Aufruf, der einen Boole'schen Ausdruck erwartet.
<i>object</i>	Object	Ein beliebiger Objektname.
<i>object class</i>	Object class	Eine beliebige Objektklasse oder Klassen-Interface.

Hinweise

Wenn *object* zur Klasse *object class* gehört, liefert **IsA** [True](#); falls nicht, liefert es [False](#). Wenn *object* Nil ist, liefert **IsA** immer [False](#). **IsA** kann dann nicht verwendet werden.

Im folgenden Code liefert **IsA** [False](#):

```

Dim t as TCPSocket
If t IsA TCPSocket then //liefert False
//mache hier irgendwas
End if

```

Sie müssen die lokale Variable mit Hilfe des New-Operators instantiieren, so daß sie nicht Nil ist:

```

Dim t as TCPSocket
t=New TCPSocket
If t IsA TCPSocket then //liefert True
//mache hier irgendwas
End if

```

Der **IsA**-Operator liefert sowohl für die eigene Klasse des Objekts, als auch für die übergeordnete Klasse, von der die Klasse abgeleitet wurde, True, bis hin zur Object-Klasse. **IsA** meldet, daß alle Klassen letztendlich von Object abgeleitet werden.

Erzeugen Sie zum Beispiel eine Unterklasse von EditField namens SecureEditField, die Kopieren und Ausschneiden im Bearbeiten-Menü deaktiviert. Wenn EditField1 eine Instanz von SecureEditField ist, liefern folgende beiden Tests True:

```

EditField1 IsA EditField
EditField1 IsA SecureEditField

```

Der IsA-Operator wird zum *casten* von Objekten verwendet. Mit dem IsA-Operator testen Sie, ob ein Objekt einer spezifischen Unterklasse entspricht und, falls das so ist, casten Sie diesen Typ um damit etwas bestimmtes zu unternehmen.

Hier folgt ein Beispiel, das eine For Loop Schleife einsetzt, um alle Steuerelemente eines Fensters zu überprüfen, ob es sich jeweils um ein EditField handelt. Wenn das der Fall ist, wird das Steuerelement gecastet, es wird der Name und der Wert seiner Text-Eigenschaft ermittelt und der Inhalt von EditField einem Feld in einer Datenbank-Tabelle namens *fieldname* zugewiesen.

```

Dim r as DatabaseRecord
Dim fieldname, fieldContents as String
:
For i = 1 to Self.ControlCount //Anzahl Steuerelemente im Fenster
If Self.control(i) IsA EditField then
fieldname = EditField(control(i)).DataField //caste es
//Die Text-Eigenschaft, die dem Inhalt dieses Feldes zugewiesen ist
fieldContents = EditField(control(i)).text
r.column(fieldname)= fieldContents
end if
next

```

Wie Sie sehen können, ist der Code universell, da er auf kein spezifisches EditField oder eine bestimmte Datenbank zurückgreift.

Ein zweites Beispiel zum Casten verwendet die Window-Funktion, um das oberste Fenster zu ermitteln und dann IsA, um festzustellen, um was für ein Fenster es sich handelt (das bedeutet Fenster1, Fenster2) und castet es dann, um auf etwas bestimmtes in diesem Fenster zuzugreifen.

```

If Window(0) IsA Window1 then
Window1(Window(0)).Pushbutton1.enabled = True
End if

```

Sie können den IsA-Operator in Verbindung mit einem Klassen-Interface verwenden. Wenn eine Klasse ein bestimmtes Interface implementiert, liefert IsA True zurück. Wenn Sie zum Beispiel ein Klassen-Interface meinInterface erzeugen und eine Steuerelement der Klasse StaticText, EditField oder Canvas implementiert diese, wird IsA True zurückgeben.

Auf diesem Weg können Sie generischen Code schreiben, der alle Steuerelemente eines Fensters prüft, ob Sie ein bestimmtes Klassen-Interface unterstützen.

Wenn Sie beispielsweise ein Klassen-Interface geschrieben haben, das den Zeichensatz eines Steuerelements ändert; zum Beispiel über die Methode UpdateTheFont, so können Sie Code wie diesen verwenden, um den Zeichensatz für alle Steuerelemente zu setzen, die dies unterstützen. Das folgende Beispiel eines Action-Event-Handlers eines PushButtons prüft alle Elemente des Fensters, ob sie die Methode UpdateTheFont unterstützen und dementsprechend das Klassen-Interface implementiert haben.

```

Dim i as Integer
For i=0 to Self.ControlCount
If Self.Control(i) IsA FontInterfaceManager then
FontInterfaceManager(Self.Control(i)).UpdateTheFont
End If
Next

```

Beispiel

Dieses Beispiel verwendet den Operator **IsA** um herauszufinden, ob **p** ein PushButton ist. Falls dem so ist, wird der Caption-Eigenschaft der Wert „OK“ zugewiesen.

```
If p IsA PushButton Then
    p.caption="OK"
End if
```

Siehe auch

Lesen Sie auch das Kapitel „Wiederverwendbare Objekte durch Klassen“ im Entwicklerhandbuch durch.

IsCMMClick Funktion

Liefert True, wenn das Event MouseDown auftritt, während die Control-Taste (auf dem Macintosh) gedrückt wurde (der Anwender möchte ein Kontextmenü für das Objekt bekommen, auf das er geklickt hat) oder unter allen anderen Plattformen, wenn die rechte Maustaste gedrückt wurde. Dies funktioniert nur in Fenstern und Steuerelementen, die einen MouseDown-Eventhandler besitzen. Auf dem Macintosh wird auch True zurückgeliefert, wenn zusätzlich zur linken Maustaste auch die Control-Taste gedrückt wurde.

Syntax

Ergebnis = IsCMMClick

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Boolean</u>	<u>True</u> falls die Control-Taste während des Mausclicks gedrückt wurde und <u>False</u> , falls das nicht der Fall war. Unter Windows liefert die Funktion <u>True</u> , wenn die rechte Maustaste in einem Fenster oder einem Steuerelement gedrückt wurde, das einen MouseDown-Event-Handler besitzt.

Hinweis

Diese Funktion wird verwendet um festzustellen, ob der Anwender ein mögliches Kontextmenü für das angeklickte Objekt sehen möchte. Zur Nutzung von Kontextmenüs sollten Sie die Beschreibung des ContextualMenu Steuerelements lesen.

Beispiele

Dieses Beispiel prüft, ob der Anwender die Control-Taste beim Mausclick gedrückt hat und zeigt dementsprechend ein ContextualMenu (Kontextmenü).

```
If IsCMMClick Then
    ContextualMenu1.Open // Zeigt das Kontextmenü
End If
```

Siehe auch

ContextualMenu Steuerelement.

IsNumeric Funktion

Ermittelt, ob ein Wert numerisch ist.

Syntax

Ergebnis = IsNumeric(arg)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>Boolean</u>	<u>True</u> wenn arg ein numerischer Wert ist, andernfalls <u>False</u>
<i>arg</i>		Die Variable, deren Wert überprüft werden soll.

Hinweise

Wenn der übergebene Wert ein String ist, der eine Zahl enthält, so liefert **IsNumeric** ebenfalls True. Zum Beispiel liefert

```
IsNumeric("3.1416")
```

True zurück.

Eine Anwendung kann sein, um festzustellen, ob der Benutzer einen String eingegeben hat, der in eine Zahl konvertiert werden kann. Zum Beispiel

```
IsNumeric( "1234.65e+10" ) //liefert True
IsNumeric( "567e-10.5" ) //liefert False
```

Siehe auch

[Variant](#), [Single](#), [Integer](#), [Double](#), [String](#), [Color](#), [Boolean](#) Datentypen.

Join Funktion

Fügt einem String einen Wert an, indem es die Elemente eines [String](#)-Arrays aneinanderhängt.

Syntax

Ergebnis=Join(sourceArray, [delimiter])

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Ein String, der alle Elemente des Quell-Arrays enthält, optional durch <i>delimiter</i> getrennt.
<i>sourceArray</i>	String-Array	Quell-Array der Elemente, die zusammengefügt werden.
<i>delimiter</i>	String	Optionales Trennzeichen, das zwischen die einzelnen Elemente gesetzt wird. Voreingestellt ist ein Leerzeichen.

Hinweise

Join verarbeitet ein [String](#)-Array und hängt dessen Elemente aneinander, um einen einzelnen [String](#) zu bilden. Sie können optional ein Trennzeichen angeben, das zwischen den Elementen eingefügt wird. Geben Sie kein Trennzeichen an, wird ein einzelnes Leerzeichen verwendet.

Die [Split](#)-Funktion führt das genaue Gegenteil durch. Es wandelt einen [String](#) in ein Array um, bei dem der String jeweils am Trennzeichen geteilt wird.

Beispiel

Dieses Beispiel fügt ein Array mit drei Elementen zu „Anthony,Aardvark,Accountant“ zusammen.

```
Dim myArray(2) as String
Dim s as String
myArray=Array("Anthony", "Aardvark", "Accountant")
//Komma wird als Trennzeichen festgelegt
s=Join(myArray, ",") //erzeugt "Anthony,Aardvark,Accountant"
```

Siehe auch

[Dim](#) Befehl; [Array](#), [Split](#), [Ubound](#), [NthField](#) Funktionen; [Append](#), [Insert](#), [Redim](#), [Remove](#), [Pop](#), [IndexOf](#), [Insert](#), [Shuffle](#), [Sort](#) Methoden; [ParamArray](#) Schlüsselwort.

Kann dieser Eigenschaft keinen Wert zuweisen Fehler

Sie haben versucht einer konstanten Eigenschaft oder einer Eigenschaft, die den Werte-Typ nicht unterstützt, einen Wert zuzuweisen.

Beispiel

Einer Konstanten einen Wert zuweisen:

```
Const SerialPort=1080
SerialPort=1040
```

Keyboard Objekt

Ein Objekt, das den Status der Tastatur wiedergibt.

Eigenschaften

Name	Art	Beschreibung
<i>AsyncCommandKey</i>	Boolean	True , wenn die Befehlstaste (Macintosh) oder die Alt-Taste (Windows und Linux) gedrückt ist.
<i>AsyncControlKey</i>	Boolean	True , wenn die Controltaste gedrückt ist.

Name	Art	Beschreibung
<i>AsyncKeyDown</i>	<u>Boolean</u>	<u>True</u> , wenn die Taste mit angegebenem Keycode gedrückt ist.
<i>AsyncOptionKey</i>	<u>Boolean</u>	<u>True</u> , wenn die Optionstaste gedrückt ist.
<i>AsyncShiftKey</i>	<u>Boolean</u>	<u>True</u> , wenn die Shift-Taste gedrückt ist.
<i>CommandKey</i>	<u>Boolean</u>	<u>True</u> , wenn die Befehlstaste beim Start der Methode oder des Event-Handlers gedrückt war. Unter Windows ist die „Windows“-Taste die Befehlstaste.
<i>ControlKey</i>	<u>Boolean</u>	<u>True</u> , wenn die Controltaste beim Start der Methode oder des Event-Handlers gedrückt war.
<i>OptionKey</i>	<u>Boolean</u>	<u>True</u> , wenn die Optionstaste beim Start der Methode oder des Event-Handlers gedrückt war.
<i>ShiftKey</i>	<u>Boolean</u>	<u>True</u> , wenn die Shift-Taste beim Start der Methode oder des Event-Handlers gedrückt war.

Events

Keine.

Methoden

Keine.

Hinweis

Das **Keyboard**-Objekt wird verwendet, um festzustellen, ob bestimmte Tasten gedrückt wurden. Sie sollten die Eigenschaften *CommandKey*, *ControlKey*, *OptionKey* und *ShiftKey* verwenden, wenn Sie einfach nur wissen möchten, ob eine dieser Sondertasten gedrückt war, als der aktuelle Event-Handler oder die aktuelle Methode gestartet wurde. Wenn Sie (wie z.B. in einer Schleife) mehr als einmal den Status dieser (oder anderer) Tasten abfragen möchten, sollten Sie die Eigenschaften *AsyncCommandKey*, *AsyncControlKey*, *AsyncKeyDown*, *AsyncOptionKey* und *AsyncShiftKey* verwenden. Diese Eigenschaften werden während des Programmablaufs ständig auf aktuellen Stand gebracht.

Windows

Sie können reine Win32-Tastatur-Codes an *AsyncKeyDown* übergeben. Addieren Sie 1 zu dem reinen keyCode und setzen das in der High-Word des Integers, den Sie übergeben. Zum Beispiel:

Keyboard.AsyncKeyDown((rawKeyCode + 1) * &hFFFF)

Der „normale“ Weg, einen bestimmten keyCode zu ermitteln, der durch das erste Beispiel im Beispiele-Abschnitt gezeigt wird, funktioniert auch unter Windows.



Beispiele

Folgendes Beispiel testet, ob die Taste für den Buchstaben „A“ gedrückt wurde:

```
If Keyboard.AsyncKeyDown(&h00) then
    //Mache mit dieser Taste hier irgendwas
end if
```

Folgendes Beispiel erlaubt nur numerische Eingaben in ein EditField:

```
Function Keydown(Key as String) as Boolean
    If (key="0" or key="1" or key="2" or key="3" or key="4" or key="5" or key="6" or key="7" or
        key="8" or key="9") then
        Return False
    Else
        Return True
    End if
End Function
```

Dieses Beispiel zeigt das Tastaturkürzel für einen Button, wenn die Befehlstaste gedrückt gehalten wird. Der Code im Event-Handler „Action“ eines Timers prüft die Eigenschaft *CommandKey* zweimal pro Sekunde:

```
If Keyboard.CommandKey Then
    PushButton1.Caption="Speichern "+Chr(17)+"S"
Else
    PushButton1.Caption="Speichern"
End if
```

Folgendes Beispiel überwacht die Pfeiltasten und stellt fest, wenn eine Pfeiltaste gedrückt wurde. Plazieren Sie dies in das EnableMenuItems-Event der Application-Klasse, damit es auch funktioniert, wenn kein Fenster geöffnet ist.

```
If Keyboard.AsyncKeyDown(123) then
    //hier irgendwas machen...
end if
If Keyboard.AsyncKeyDown(124) then
    //hier irgendwas machen...
end if
If Keyboard.AsyncKeyDown(125) then
    //hier irgendwas machen...
end if
If Keyboard.AsyncKeyDown(126) then
    //hier irgendwas machen...
end if
```

KeyChain Klasse

Ermöglicht Ihnen den Zugriff auf die Schlüsselbunde von Mac OS Classic und Mac OS X für Ihre Applikationen. Die **KeyChain**-Klasse liefert keinen Zugriff auf die Internet-Paßwörter.

Übergeordnete Klasse

Object

Constructor

Name	Parameter	Beschreibung
<i>KeyChain</i>	Index als <u>Integer</u>	Liefert Ihnen eine Referenz auf einen beliebigen Schlüsselbund (KeyChain), den das Schlüsselbund-Programm kennt. Der Index-Wert sollte sich im Bereich von 0 bis <u>System.KeyChainCount</u> befinden.

Eigenschaften

Name	Typ	Beschreibung
<i>Handle</i>	<u>Integer</u>	Liefert ein Handle auf das KeyChain . Nur nützlich, wenn Sie auf die Schlüsselbunde von REALbasic mit <u>Declare</u> -Befehlen verweisen wollen.

Methoden

Name	Parameter	Beschreibung
<i>AddPassword</i>	kci als <u>KeyChainItem</u> Password als <u>String</u>	Fügt das Passwort dem Keychain hinzu und verbindet es mit einem <u>KeyChainItem</u> . Falls dies scheitert, wird eine <u>KeyChainException</u> ausgelöst.
<i>FindPassword</i>	<u>ByRef</u> kci als <u>KeyChainItem</u>	Liefert einen <u>String</u> . Versucht, das Passwort für ein Objekt im festgelegten Schlüsselbund zu finden, das auf <i>kci</i> paßt. Am Ende eines erfolgreichen Aufrufs zeigt <i>kci</i> auf das gefundene KeyChain-Objekt. Falls dies scheitert, wird eine <u>KeyChainException</u> ausgelöst.
<i>Lock</i>	LockAll als <u>Boolean</u>	Schützt den Schlüsselbund. Falls LockAll <u>True</u> ist, werden alle Schlüsselbunde auf dem System geschützt. Hinweis von Apple: Es gibt normalerweise keinen Grund für eine Applikation, jemals Lock aufzurufen. So lange Ihre Applikation direkt auf die Anforderung eines Anwenders reagiert, einen Schlüsselbund zu schützen, wird empfohlen, daß Sie den Schlüsselbund ungeschützt lassen, damit der Anwender ihn in einer anderen Applikation nicht wieder freigeben muß. Falls dies scheitert, wird eine <u>KeyChainException</u> ausgelöst.
<i>Unlock</i>	Password as <u>String</u>	Gibt den Schlüsselbund frei, wenn das Passwort ein leerer String ("") ist. Das System zeigt den Standard-Freigeben-Dialog an und der übergebene Schlüsselbund erscheint als der ausgewählte Eintrag im Popup-Menü dieses Dialogs. HINWEIS: Aufgrund der Implementation von Apple benötigt das System bis zu einem Neustart menschliche Intervention für diesen Schlüsselbund, wenn Sie einmal ein falsches Passwort für einen bestimmten Schlüsselbund übergeben haben. Falls dies scheitert, wird eine <u>KeyChainException</u> ausgelöst.

Hinweise

Der *Schlüsselbund* (*keychain*) ist eine systemweite Einrichtung in Mac OS 8.5 und neuer (inklusive Mac OS X), um Zugangspasswörter für Applikationen zu speichern. Indem Sie sich die eingebaute Schlüsselbund-Einrichtung zu Nutze machen, müssen Anwender deren Passwörter nicht mehr eintippen, wenn deren Schlüsselbund freigegeben ist. Sie sollten den Anwender immer fragen, bevor Sie etwas in den Schlüsselbund speichern.

Eine äquivalente Technologie zum Mac OS-Schlüsselbund existiert derzeit nicht auf der Windows-Plattform, daher wird KeyChain unter Windows nicht unterstützt.

Beispiele

Folgendes Beispiel fügt ein KeyChainItem einer Applikation hinzu und weist ein Passwort zu:

```

Dim NewItem as KeyChainItem
If System.KeyChainCount > 0 then
    NewItem = New KeyChainItem
    'Namen der Applikation angeben
    NewItem.ServiceName = "MyApplication"

    'Erzeugt einen neuen Schlüsselbund-Eintrag für die Applikation und weist das Passwort zu
    System.KeyChain.AddPassword NewItem, "GeheimesPaßwort"
Else
    Beep
    MsgBox "Sie haben keinen Schlüsselbund."
End if

Exception err as KeyChainException
MsgBox "Kann Eintrag: " + err.Message + " nicht hinzufügen."

```

Folgendes Beispiel ermittelt das Passwort und zeigt es in einer MessageBox an.

```

Dim ItemToFind as KeyChainItem
Dim password As String

```

```
ItemToFind = New KeyChainItem

'Gibt den Namen der Applikation an, dessen Schlüsselbund-Eintrag man finden möchte
ItemToFind.ServiceName = "MyApplication"

'Ermittle Paßwort der Applikation aus dem System-Schlüsselbund
password = System.KeyChain.FindPassword(ItemToFind)
MsgBox "Das Paßwort für diesen Eintrag ist: " + password

Exception err as KeyChainException
MsgBox "Kann Eintrag: " + err.Message + " nicht finden."
```

Siehe auch

[KeyChainItem](#) Klasse; [KeyChainException](#) Fehler; [System](#) Objekt.

KeyChainException Fehler

Eine Methode der [KeyChain](#)- oder [KeyChainItem](#)-Klasse schlug fehl. Um das Problem näher zu untersuchen, sollten Sie sich den von **KeyChainException** zurückgelieferten Fehlercode genauer ansehen.

Übergeordnete Klasse

[RuntimeException](#)

Eigenschaften

Name	Typ	Beschreibung
<i>ErrorNumber</i>	Integer	Kennzeichnet den Typ der Fehlers, der aufgetreten ist.

Hinweise

Folgende Tabelle listet die Werte von *ErrorNumber* und die zugehörigen Texte, die in der Message-Eigenschaft zurückgeliefert werden, auf:

Fehlernummer	Mitteilung
-128	User Cancelled.
-25291	Key Chain not available.
-25292	Key Chain read only.
-25293	Key Chain Authorization failed.
-25294	No such Key Chain.
-25295	Invalid Key Chain.
-25296	Duplicate Key Chain.
-25797	Key Chain Duplicate Callback.
-25298	Key Chain Invalid Callback.
-25299	Key Chain Duplicate Item.
-25300	Key Chain Item Not Found.
-25301	Key Chain Buffer Too Small.
-25302	Key Chain Data Too Large.
-25303	Key Chain No Such Attribute.
-25304	Key Chain Invalid Item Reference.
-25305	Key Chain Invalid Search Reference.
-25306	Key Chain No Such Class.
-25307	No Default Key Chain.
-25308	Key Chain Interaction Not Allowed.
-25309	Key Chain Read Only Attribute.

Fehlernummer	Mitteilung
-25310	Wrong Key Chain Version.
-25311	Key Chain Key Size Not Allowed.
-25312	Key Chain No Storage Module.
-25313	Key Chain No Certificate Module.
-25314	Key Chain No Policy Module.
-25315	Key Chain Interaction Required.
-25316	Key Chain Data Not Available.
-25317	Key Chain Data Not Modifiable.
-25318	Key Chain Create Chain Failed.

Beispiele

Folgendes Beispiel zeigt eine MessageBox, falls Sie z.B. versuchen, für die gleiche Applikation mehr als ein KeyChainItem zu erzeugen:

```
Dim newItem as KeyChainItem
If System.KeyChainCount > 0 then

    newItem = New KeyChainItem
    'Namen der Applikation angeben
    newItem.ServiceName = "MyApplication"

    'Erzeugt einen neuen Schlüsselbund-Eintrag für die Applikation und weist das Paßwort zu
    System.KeyChain.AddPassword newItem, "GeheimesPaßwort"
Else
    Beep
    MsgBox "Sie haben keinen Schlüsselbund."
End if

Exception err as KeyChainException
MsgBox err.Message + ". Fehler-Code: " + Str(err.errorNumber)
```

Folgendes Beispiel verwendet einen Exception-Block, um eine MessageBox anzuzeigen, falls die Applikation, die durch ServiceName festgelegt wird, kein KeyChainItem hat.

```
Dim itemToFind as KeyChainItem
Dim password As String

itemToFind = New KeyChainItem

'Legt den Namen der Applikation fest, dessen Schlüsselbund-Eintrag Sie finden möchten.
itemToFind.ServiceName = "MyApplication"

'ermittle das Paßwort der Applikation aus dem System-Schlüsselbund
password = System.KeyChain.FindPassword(itemToFind)
MsgBox "Das Paßwort für diesen Eintrag ist: " + password

Exception err as KeyChainException
MsgBox "Kann Eintrag: " + err.Message + " nicht finden."
```

Siehe auch

KeyChain, KeyChainItem, RuntimeException Klassen; System Objekt.

KeyChainItem Klasse

Verweist auf einen KeyChain-Eintrag.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>AccountName</i>	<u>String</u>	Enthält den Namen des Accounts (benötigt für das Hinzufügen, kann <u>Nil</u> zum Suchen sein).
<i>Comment</i>	<u>String</u>	Ein String, den der Endanwender bearbeiten kann, der Kommentare für diesen Schlüsselbund-Eintrag enthält.
<i>Description</i>	<u>String</u>	String, der für den Endanwender sichtbar ist, der diesen Schlüsselbund-Eintrag beschreibt.
<i>Handle</i>	<u>Integer</u>	Enthält die KeyChainItem -Referenz, die für Toolbox-Aufrufe verwendet werden kann.
<i>Label</i>	<u>String</u>	Ein String, den der Endanwender bearbeiten kann, der das Label für diesen Schlüsselbund-Eintrag enthält.
<i>ServiceName</i>	<u>String</u>	Enthält den Namen des Services (benötigt für das Hinzufügen, kann <u>Nil</u> zum Suchen sein). Um einer Applikation ein Passwort hinzuzufügen, setzen Sie <i>ServiceName</i> auf den Namen der Applikation.

Methoden

Name	Parameter	Beschreibung
Delete		Löscht den KeyChainItem . Sie können Passwörter in einem <u>KeyChain</u> ändern, indem Sie den Original-Eintrag mit dieser Methode löschen und einen anderen Eintrag hinzufügen.

Hinweise

REALbasic-KeyChainItems können nur auf Passwörter von Applikationen zugreifen, nicht auf Internet-Passwörter.

Beispiele

Folgendes Beispiel fügt ein KeyChainItem einer Applikation hinzu und weist ein Passwort zu:

```

Dim NewItem as KeyChainItem
If System.KeyChainCount > 0 then

    NewItem = New KeyChainItem
    'Legt den Namen der Applikation fest
    NewItem.ServiceName = "MyApplication"

    'Erzeugt einen neuen Schlüsselbund-Eintrag für die Applikation und weist ein Passwort zu
    System.KeyChain.AddPassword NewItem, "SecretPassword"
Else
    Beep
    MsgBox "Sie haben keinen Schlüsselbund."
End if

Exception err as KeyChainException
MsgBox "Kann Eintrag: " + err.Message + " nicht hinzufügen."
```

Folgendes Beispiel ermittelt das Passwort, das in vorigem Beispiel gesetzt wurde und zeigt es in einer MessageBox an

```

Dim ItemToFind as KeyChainItem
Dim password As String

ItemToFind = New KeyChainItem

'Legt den Namen der Applikation fest, deren Schlüsselbund-Eintrag Sie finden möchten
ItemToFind.ServiceName = "MyApplication"

'Ermittle das Passwort der Applikation aus dem System-Schlüsselbund
password = System.KeyChain.FindPassword(ItemToFind)
MsgBox "Das Passwort für diesen Eintrag ist: " + password

Exception err as KeyChainException
MsgBox "Kann Eintrag: " + err.Message + " nicht finden."
```

Siehe auch

KeyChain Klasse, KeyChainException Fehler; System Objekt.

KeyNotFoundException Fehler

Sie haben versucht, auf ein Objekt in einem Dictionary über einen Schlüssel zuzugreifen, den es im Dictionary nicht gibt.

Übergeordnete Klasse

RuntimeException

Beispiel

Die folgende For-Schleife produziert einen **KeyNotFoundException** Fehler, wenn der Zähler der Schleife die 2 erreicht. Der Wert von 2 – der ein Variant ist und kein Integer – wurde gerade entfernt.

```
Dim d as Dictionary
Dim i as Integer
d=New Dictionary
d.Value(1)="ID"
d.Value(2)="Lois Lane"
d.Value(3)="Reporter"
d.Value(4)=84000
d.Remove(2)
For i=1 to d.Count //scheitert, wenn i=2
    ListBox1.Addrow d.value(i)
Next
```

Um den Fehler abzufangen, fügen Sie an das Ende der Methode einen Exception Block wie diesen hinzu:

```
Exception err as KeyNotFoundException
MsgBox "Sie haben versucht, auf ein nicht existierendes Objekt zuzugreifen!"
```

Siehe auch

Dictionary, RuntimeException Klassen, Exception Block.

Left Funktion

Liefert die ersten *anzahl* Zeichen eines Strings (von links beginnend) zurück.

Syntax

Ergebnis=**Left**(quelle, anzahl)

ODER

Ergebnis=StringVariable.**Left**(anzahl)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die ersten <i>anzahl</i> Zeichen aus <i>quelle</i> .
<i>quelle</i>	<u>String</u>	Der String, aus dem die Zeichen geholt werden.
<i>anzahl</i>	<u>Integer</u>	Die Anzahl an Zeichen, die Sie aus <i>quelle</i> erhalten möchten. Ist <i>anzahl</i> größer als die Anzahl an Zeichen im String <i>quelle</i> , so werden alle Zeichen zurückgeliefert.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **Left** liefert links beginnend *anzahl* Zeichen des Strings *quelle* zurück (daher auch der Name dieser Funktion).

Wenn Sie Bytes anstatt von Zeichen einlesen möchten, sollten Sie die Funktion LeftB verwenden.

Beispiel

Dieses Beispiel liefert die ersten fünf Zeichen in einem String.

```
Dim s As String
s=Left("Hallo Welt", 5) //liefert "Hallo"
```

Siehe auch

Asc, Chr, InStr, Len, Mid, Right Funktionen.

LeftB Funktion

Liefert die ersten *Anzahl* Bytes eines Strings (von links beginnend) zurück.

Syntax

Ergebnis=**LeftB**(Quelle, Anzahl)

ODER

Ergebnis=**StringVariable.LeftB**(Anzahl)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die ersten <i>Anzahl</i> Bytes von <i>Quelle</i> .
<i>Quelle</i>	<u>String</u>	Der String, aus dem die Zeichen geholt werden.
<i>Anzahl</i>	<u>Integer</u>	Die Anzahl an Bytes, die Sie aus <i>Quelle</i> erhalten möchten. Ist <i>Anzahl</i> größer als die Anzahl an Bytes im String <i>Quelle</i> , so werden alle Bytes zurückgeliefert.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **LeftB** liefert *Anzahl* Bytes von der linken Seite des Strings *Quelle* beginnend zurück (daher auch der Name dieser Funktion).

Wenn Sie Zeichen anstatt von Bytes einlesen möchten, sollten Sie die Funktion Left verwenden.

Beispiel

Dieses Beispiel verwendet die Funktion **LeftB**, um die ersten fünf Bytes aus einem String zurückzuliefern.

```
Dim s As String
s=LeftB("Hallo Welt", 5) //liefert "Hallo"
```

Siehe auch

AscB, ChrB, InStrB, Left, LenB, MidB, RightB Funktionen.

Len Funktion

Liefert die Anzahl der Zeichen in einem String.

Syntax

Ergebnis=**Len**(String)

ODER

Ergebnis=**StringVariable.Len**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Anzahl von Zeichen in <i>string</i> .
<i>String</i>	<u>String</u>	Ein beliebiger, gültiger String.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Wenn Sie die Anzahl von Bytes im String benötigen, sollten Sie die Funktion LenB verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **Len** um die Anzahl von Zeichen im String festzustellen.

```
Dim n As Integer
n=Len("Hallo Welt") //liefert 10
```

Siehe auch

Asc, Chr, InStr, Len, Mid, Right Funktionen.

LenB Funktion

Liefert die Anzahl von Bytes in einem String.

Syntax

Ergebnis=**LenB**(String)

ODER

Ergebnis=**StringVariable.LenB**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Anzahl von Bytes in <i>string</i> .
<i>String</i>	<u>String</u>	Ein beliebiger, gültiger String.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweise

LenB behandelt *string* als eine Reihe von Bytes und nicht Buchstaben. Es sollte verwendet werden, wenn *string* aus binären Daten besteht. Wenn Sie die Anzahl von Zeichen und nicht die Anzahl von Bytes im String wissen möchten, sollten Sie die Funktion Len verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **LenB** um die Anzahl von Bytes im String festzustellen.

```
Dim n As Integer
n=LenB("Hallo Welt") //liefert 10
```

Siehe auch

AscB, ChrB, InStrB, Len, MidB, RightB Funktionen.

Light3D Klasse

Erlaubt Ihnen, Ihre eigene Beleuchtung für RB3DSpace-Welten zu definieren.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Attenuation</i>	<u>Integer</u>	Legt fest, wie die Helligkeit eines Punkt-Lichtes mit der Distanz abnimmt. 0: Überhaupt keine Abschwächung. 1: Helligkeit nimmt mit der Entfernung linear ab. 2: Helligkeit nimmt mit der Entfernung quadratisch ab. Der Default-Wert ist 1 (Linear).
<i>Brightness</i>	<u>Integer</u>	Helligkeit als Prozentwert ausgedrückt von 0 (ganz aus) bis zu 100 (die Voreinstellung). Werte größer als 100 sind gültig, wenn Sie zusätzliche Helligkeit benötigen.
<i>Direction</i>	<u>Vector3D</u>	Falls <u>Nil</u> , erhalten Sie ein Punkt-Licht, das in alle Richtungen strahlt. Wenn es nicht Nil ist, ist es die Richtung für ein Flutlicht. Es kann zwischen Punkt- und Flutlicht gewechselt werden, so lange das Licht einem RB3DSpace angehängt ist.
<i>LightColor</i>	<u>Color</u>	Die Farbe des Lichtes. Voreinstellung ist weiß.
<i>Position</i>	<u>Vector3D</u>	Falls <u>Nil</u> , erhalten Sie ein Flutlicht. Wenn es nicht Nil ist, ist es die Position eines Punkt-Lichtes.

Hinweise

Es ist nicht erlaubt, ein Licht zu verwenden, bei dem Position und Direction entweder beide Nil oder beide nicht Nil sind.

Falls der Wert von Attenuation 1 oder 2 ist, nimmt der Effekt des Lichtes mit der Entfernung ab; daher müssen Sie evtl. Brightness erheblich vergrößern oder Ihre Objekte kleiner machen und näher am Licht postieren.

Siehe auch

[RB3DSpace](#), [Vector3D](#), [Bounds3D](#) Klassen.

LightBevelColor Funktion

Liefert die momentan gewählte Farbe des Erscheinungsbild-Managers zurück, die zum Zeichnen heller Linien bei Unterteilungslinien und GroupBoxen verwendet wird.

Syntax

Ergebnis=LightBevelColor

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Die zum Zeichnen der hellen Linien in Unterteilungslinien und GroupBoxen verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit [Canvas](#) Steuerelementen wichtig, um deren Aussehen mit den Einstellungen des Erscheinungsbild-Managers in Einklang zu bringen. Wenn Sie Steuerelemente wie Unterteilungslinien oder GroupBox-Objekte zeichnen, sollten Sie diese Farbe zum Zeichnen heller Bereiche (üblicherweise die linke und die obere Seite des Objekts) verwenden.

Dieser Wert kann vom Anwender geändert werden, weshalb Sie in jedem Zeichen-Event diesen Wert neu abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

[DarkBevelColor](#), [DarkTingeColor](#), [FillColor](#), [FrameColor](#), [HighlightColor](#), [LightTingeColor](#), [TextColor](#) Funktionen; [Color](#) Datentyp.

LightTingeColor Funktion

Liefert die momentan gewählte Farbe des Erscheinungsbild-Managers zurück, die zum Zeichnen heller Linien an linker und oberer Seite in Hervorhebungsrahmen verwendet wird.

Syntax

Ergebnis=LightTingeColor

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Color</u>	Die zum Zeichnen der hellen Linien innerhalb eines Rahmens verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit [Canvas](#) Steuerelementen wichtig, um deren Aussehen mit den Einstellungen des Erscheinungsbild-Managers in Einklang zu bringen. Wenn Sie hervorgehobene Objekte zeichnen, sollten Sie diese Farbe zum Zeichnen heller Bereiche (üblicherweise die linke und die obere Seite des Objekts) verwenden. In einem [Checkbox](#) Steuerelement wird diese Farbe verwendet, um den hervorgehobenen Eindruck hervorzurufen.

Dieser Wert kann vom Anwender geändert werden, weshalb Sie in jedem Zeichen-Event diesen Wert neu abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

[DarkBevelColor](#), [DarkTingeColor](#), [FillColor](#), [FrameColor](#), [LightBevelColor](#), [TextColor](#) Funktionen; [Color](#) Datentyp.

Line Steuerelement



Zeichnet eine Linie.

Übergeordnete Klasse

[RectControl](#)

Da dies ein Steuerelement ([RectControl](#)) ist, informieren Sie sich bei der [RectControl](#)-Klasse über die anderen Eigenschaften und Events, die alle [RectControl](#)-Objekte gemeinsam haben.

Eigenschaften

Name	Art	Beschreibung
<i>BorderWidth</i>	<u>Integer</u>	Die Linienbreite in Pixeln.
<i>LineColor</i>	<u>Color</u>	Die Linienfarbe.
Name	<u>String</u>	Der Name des Objekts.
Super	Object Class	Die Objektklasse, auf der das Objekt basiert.
<i>Visible</i>	<u>Boolean</u>	Wenn die Linie beim Öffnen eines Fensters sichtbar sein soll, <u>True</u> .
<i>X1</i>	<u>Integer</u>	Der Punkt auf der X-Achse, an dem die Linie beginnt.
<i>X2</i>	<u>Integer</u>	Der Punkt auf der X-Achse, an dem die Linie endet.
<i>Y1</i>	<u>Integer</u>	Der Punkt auf der Y-Achse, an dem die Linie beginnt.
<i>Y2</i>	<u>Integer</u>	Der Punkt auf der Y-Achse, an dem die Linie endet.

Methoden

Name	Parameter	Beschreibung
<i>MouseDown</i> (function)	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der Linie an der Position x,y gedrückt. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen.
<i>MouseDrag</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der Linie gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der Linie an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.

Siehe auch

Control Klasse.

ListBox Steuerelement



Das scrollbare ListBox-Steuerelement wird verwendet, um eine oder mehrere Spalten mit Informationen anzuzeigen. Sie können eine Checkbox und/oder ein Bild in eine Zeile hinzufügen. Sie können den Font-Stil auf Zellbasis einstellen und die Ausrichtung der Spalten steuern. Mit ein wenig Programmierung können Sie hierarchische Listen erzeugen, um z.B. die Listendarstellung vom Finder nachzubilden.

Die Steuerelemente-Palette enthält Icons für ein- und mehrspaltige ListBoxes. Sie können nach Verwenden eines der beiden Icons die Anzahl der Spalten durch Setzen der ColumnCount-Eigenschaft ändern. Der einzige Unterschied ist, daß Sie bei dem Werkzeug der mehrspaltigen ListBox gleich mit einer zweispaltigen Box beginnen.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>ActiveCell</i>		EditField, das die ListBox für seine Operationen in einer editierbaren Zelle verwendet. Sie können diese Eigenschaft verwenden, um den Text dieser ListBox-Zelle zu setzen oder zu ermitteln (selText), die Selektion zu setzen oder andere Eigenschaften des EditFields der ListBox zu ändern.
<i>Bold</i>	<u>Boolean</u>	Alle Listeneinträge werden fett dargestellt.

Name	Art	Beschreibung
<i>CellAlignment</i>	<u>Integer</u>	Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Die erste Zelle ist 0,0. Richtet die angegebene Zelle aus. Werte sind: 0 - Default-Ausrichtung 1 - Linksbündig 2 - Zentriert 3 - Rechtsbündig 4 - Auf Dezimalpunkt. Bei Dezimalpunkt wird dieser mit der rechten Kante der Zelle ausgerichtet. Sie müssen <code>CellAlignmentOffset</code> oder <code>ColumnAlignmentOffset</code> verwenden, um die Daten in die Zelle hineinzuschieben.
<i>CellAlignmentOffset</i>	<u>Integer</u>	Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Der Wert ist der Abstand in Pixeln von der rechten Kante der Zelle. Überschreibt <code>ColumnAlignmentOffset</code> dieser Zelle.
<i>CellBorderBottom</i>	<u>Integer</u>	Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Setzt den unteren Rahmen der Zelle, die durch <i>Zeile</i>, <i>Spalte</i> festgelegt wird, auf einen bestimmten Stil. Mögliche Werte sind: 0 - Default 1 - Keiner 2 - Dünn gepunktet 3 - Dünn durchgehend 4 - Dick durchgehend 5 - Doppelt dünn durchgehend
<i>CellBorderLeft</i>	<u>Integer</u>	Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Setzt den linken Rahmen der Zelle, die durch <i>Zeile</i>, <i>Spalte</i> festgelegt wird, auf einen bestimmten Stil. Mögliche Werte sind: 0 - Default 1 - Keiner 2 - Dünn gepunktet 3 - Dünn durchgehend 4 - Dick durchgehend 5 - Doppelt dünn durchgehend
<i>CellBorderRigth</i>	<u>Integer</u>	Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Setzt den rechten Rahmen der Zelle, die durch <i>Zeile</i>, <i>Spalte</i> festgelegt wird, auf einen bestimmten Stil. Mögliche Werte sind: 0 - Default 1 - Keiner 2 - Dünn gepunktet 3 - Dünn durchgehend 4 - Dick durchgehend 5 - Doppelt dünn durchgehend
<i>CellBorderTop</i>	<u>Integer</u>	Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Setzt den oberen Rahmen der Zelle, die durch <i>Zeile</i>, <i>Spalte</i> festgelegt wird, auf einen bestimmten Stil. Mögliche Werte sind: 0 - Default 1 - Keiner 2 - Dünn gepunktet 3 - Dünn durchgehend 4 - Dick durchgehend 5 - Doppelt dünn durchgehend
<i>CellCheck</i>	<u>Boolean</u>	Die Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Die erste Zelle ist 0,0. Wenn man <code>CellCheck</code> auf True setzt, ist die Checkbox angekreuzt.

Name	Art	Beschreibung
<i>CellType</i>	<u>Integer</u>	Die Parameter sind <i>Zeile</i>, <i>Spalte</i> (Integer). Die erste Zelle ist 0,0. Die Werte sind: 0 - Default 1 - Normal 2 - Fügt Checkbox hinzu 3 - Editierbar Der Wert von CellType überschreibt ColumnType. Wenn ColumnType z.B. 2 ist, aber eine Zelle in dieser Spalte den CellType 0 hat, wird die Zelle normal dargestellt.
<i>Column</i>	<u>ListColumn</u>	Der Parameter ist <i>SpaltenNummer</i> als Integer. Die erste Spalte hat die Nummer Null. Verwenden Sie -1, um auf alle Spalten zu verweisen. Erlaubt Ihnen, auf die ListColumn-Eigenschaften der angegebenen Spalte zuzugreifen.
<i>ColumnAlignment</i>	<u>Integer</u>	Parameter ist die Spaltennummer; die erste Spalte hat die Nummer Null. Richtet die angegebene Spalte aus. Die Werte sind: 0 - Default-Ausrichtung 1 - Linksbündig 2 - Zentriert 3 - Rechtsbündig 4 - Auf Dezimalpunkt. Bei Dezimalpunkt wird dieser mit der rechten Kante der Spalte ausgerichtet. Sie müssen ColumnAlignmentOffset verwenden, um die Daten in die Zelle hineinzuschieben.
<i>ColumnAlignmentOffset</i>	<u>Integer</u>	Parameter ist die Spaltennummer; die erste Spalte hat die Nummer Null. Der Wert ist der Abstand in Pixeln von der rechten Kante der Zelle. Ein negativer Wert bewegt den Dezimaltrenner nach links, in den Körper der Spalte. Siehe Beispiel bei den Hinweisen im Unterabschnitt „Dezimal-Ausrichtung“.
<i>ColumnCount</i>	<u>Integer</u>	Die Anzahl der Spalten in der ListBox. Die Maximalzahl an sichtbaren Spalten ist 64 (Spalte 0 bis 63).
<i>ColumnSortDirection</i>	<u>Integer</u>	Parameter ist die Spaltennummer (Integer); die erste Spalte hat die Nummer Null. Wird verwendet, um die Sortierrichtung zu ermitteln oder zu setzen, je nach folgenden Werten: -1 - Absteigend 0 - Nicht sortieren 1 - Aufsteigend (Voreinstellung) ColumnSortDirection sortiert nicht direkt die Zeilen; es legt nur die Sortierreihenfolge fest, die verwendet wird, wenn die Sort-Methode aufgerufen wird. Die Sortierrichtung kann auch festgelegt werden, wenn es keine Kopfzeile in der ListBox gibt. Wenn Sie ColumnSortDirection auf „Nicht sortieren“ stellen, kann der Anwender die Spalte nicht durch Klicken in die Kopfzeile sortieren. Bei „Nicht sortieren“ werden die Events SortColumns und CompareRows nicht aufgerufen.
<i>ColumnType</i>	<u>Integer</u>	Parameter ist die Spaltennummer; die erste Spalte hat die Nummer Null. Richtet die angegebene Spalte aus. Die Werte sind: 0 - Default 1 - Normal 2 - Füge CheckBox hinzu 3 - Innerhalb editierbar. (Aufklappdreiecke funktionieren nicht in hierarchischen ListBoxen, wenn „Innerhalb editierbar“ selektiert ist).

Name	Art	Beschreibung
<i>ColumnWidths</i>	<u>String</u>	Eine Liste von komma-getrennten Werten, wobei jeder Wert die Breite einer Spalte definiert. Ein Wert kann hierbei entweder ein absoluter Wert (in Pixeln), eine Prozentangabe oder eine relative Länge, ausgedrückt mit <i>i</i>*, wobei <i>i</i> ein Integer ist, sein. Wenn Sie eine Prozentangabe verwenden, können Sie Nicht-Integer-Werte verwenden, um Bruchteile von Prozentwerten wie z.B. 43,52% festzulegen. Der Prozentwert kann größer als 100% sein. Wenn Sie Pixel verwenden, wächst die letzte Spalte nicht auf die Größe des Rests der ListBox. Ohne eine <i>ColumnWidth</i>-Angabe werden die Spaltenköpfe gleichmäßig aufgeteilt. Wenn weniger Spaltenbreiten festgelegt sind als es Spalten gibt, bekommen die restlichen Spalten die Größe der zuletzt festgelegten Spalte. Ein Element mit der Länge von „3*“ erhält den dreifachen Platz eines Elements mit der Länge „1*“. Der Wert „*“ ist äquivalent zu „1*“ und kann verwendet werden, wenn man den verbleibenden Platz ausfüllen möchte. Sie können eine Mischung aus Pixeln, Prozentwerten und relativen Längen verwenden. Spaltenbreiten, die über Pixel festgelegt werden, haben garantiert die festgelegte Breite. Spaltenbreiten, die durch einen Prozentwert festgelegt werden, haben garantiert diesen Prozentsatz der sichtbaren Breite der ListBox. Die Spaltenbreiten die durch das „*“ festgelegt wurden, teilen sich die verbleibende Breite proportional. Wenn Sie z.B. vier Spalten mit der Festlegung 20,20%,*,2* haben, erhält die erste Spalte 20 Pixel, die zweite Spalte 20% der gesamten Breite und die letzten beiden teilen sich die restliche Breite im Verhältnis 1:2, wobei die letzte Spalte evtl. vorhandene Teil-Pixel bekommt. Nicht erkannte oder nicht unterstützte Ausdrücke (wie z.B. „2@“) für die Spaltenbreite rufen eine <code>UnsupportedFormatException</code> hervor. Die Meldung dieser Exception beschreibt auf englisch, was falsch ist. Wenn die Größe einer Spalte geändert wird, ändert sich der Wert des Ausdrucks. Wenn Sie die ListBox in der Größe ändern, berechnen die prozentualen und relativen Längen deren aktuelle Breite neu. Falls Spaltenköpfe verwendet werden, wird, falls die ListBox breiter als die Summe der Breiten der verwendeten Spaltenköpfe ist, am rechten Rand der Spaltenköpfe ein zusätzlicher, nicht genutzter Raum hinzugefügt. Dieser Spaltenkopf reagiert nicht, wenn man auf ihn klickt.
<i>DataField</i>	<u>String</u>	Nur relevant, wenn die ListBox mit einem <u>DataControl</u> verbunden ist, um den Inhalt von Feldern einer Datenbank-Tabelle anzuzeigen. Auf den Name eines Feldes in der Tabelle wird über das <u>DataControl</u> verwiesen.
<i>DataSource</i>	<u>DataControl</u>	Das <u>DataControl</u> , das auf eine Tabelle in einer Datenbank verweist, deren Felder mit der ListBox angezeigt werden. Verwenden Sie die <i>DataField</i> -Eigenschaft, um es mit dem in der ListBox anzuzeigenden Feld zu verbinden. In der IDE erscheint ein Pop-up-Menü mit den Feldnamen. Wenn es im Code eingestellt wird, erwartet es einen String, z.B. <code>ListBox1.DataSource = DataControl1.Name</code>
<i>DefaultRowHeight</i>	<u>Integer</u>	Legt die Höhe jeder neuen Zeile in der ListBox in Pixeln fest. Damit REALbasic die Höhe ausgehend von der aktuellen Fontgröße automatisch einstellt, müssen Sie <i>DefaultRowHeight</i> auf -1 setzen.
<i>EnableDrag</i>	<u>Boolean</u>	Erlaubt das Ziehen (draggen) von Zeilen.
<i>EnableDragReorder</i>	<u>Boolean</u>	Falls True, können Sie die Zeilen innerhalb der ListBox durch Draggen umsortieren. Eine Einfüge-Linie erscheint beim Draggen innerhalb der ListBox, um Ihnen anzuzeigen, wo die Zeile eingefügt wird, wenn Sie die Maustaste loslassen. Automatische Umsortierung wird bei hierarchischen ListBoxen nicht unterstützt.

Name	Art	Beschreibung
<i>Expanded</i>	<u>Integer</u>	Wird verwendet, um den Status „ausgeklappt“ der angegebenen Zeile auszulesen oder zu setzen. Die Zeile muß vorher über die AddFolder-Methode hinzugefügt worden sein.
<i>GridLinesHorizontal</i>	<u>Integer</u>	Zeichnet horizontale Linien zwischen den Zeilen in einem der fünf Stile: 0 - Voreinstellung 1 - Keine 2 - Dünn gepunktet 3 - Dünn durchgehend 4 - Dick durchgehend 5 - Doppelt dünn durchgehend
<i>GridLinesVertical</i>	<u>Integer</u>	Zeichnet vertikale Linien zwischen den Zeilen in einem der fünf Stile: 0 - Voreinstellung 1 - Keine 2 - Dünn gepunktet 3 - Dünn durchgehend 4 - Dick durchgehend 5 - Doppelt dünn durchgehend
<i>HasHeading</i>	<u>Boolean</u>	Wenn True, wird über der Liste eine Zeile mit Spaltenüberschriften dargestellt. Der Anwender kann die Liste durch Klicken auf eine Spaltenüberschrift nach dieser Spalte sortieren.
<i>Heading</i>	<u>String-Array</u>	Null-basiertes Array von Spaltenköpfen. Kopfzeilen erscheinen nur, wenn HasHeading True ist. Wenn Sie Werte sowohl Heading als auch <u>InitialValue</u> zuweisen, wird die erste Zeile von InitialValue als die erste Datenzeile interpretiert; andernfalls wird sie als Spaltenkopf und die zweite Zeile von InitialValue als die erste Datenzeile verwendet. Sie können die Spaltenköpfe in einer mehrspaltigen ListBox einstellen, indem Sie Heading(-1) den Text der Spaltenköpfe, getrennt durch das Tab-Zeichen, zuweisen, z.B. me.heading(-1) = "Vorname" + Chr(9) + "Nachname". Sie müssen ein Leerzeichen verwenden, wenn Sie einen leeren Spaltenkopf möchten; bei einem Leerstring werden die Default-Spaltenköpfe verwendet. ListBox1.heading(-1) = "" stellt alle Spaltenköpfe auf deren Defaultwert. ListBox1.heading(5) = " " leert den Spaltenkopf von Spalte 5.
<i>HeadingIndex</i>	<u>Integer</u>	Erlaubt, die Spalte zu ermitteln bzw. zu setzen, nach der die ListBox sortiert werden soll. Die erste Spalte hat die Nummer Null. Um eine ListBox zu sortieren, stellen Sie HeadingIndex auf die Nummer der Spalte, nach der Sie sortieren möchten. Dies kann im Eigenschaften-Fenster oder im Programmcode geschehen. Man kann dies sogar machen, wenn die Spalten keinen Spaltenkopf hat. Wenn es im Eigenschaften-Fenster gemacht wird, wird die gewählte Spalte im Fenster-Editor schattiert. Setzen Sie HeadingIndex auf -1, wenn Sie nicht wollen, daß die ListBox beim Klick in den Spaltenkopf sortiert wird. Wenn die ListBox unsortiert ist, liefert HeadingIndex den Wert -1.
<i>Hierarchical</i>	<u>Boolean</u>	Wendet eine der Finder-Listendarstellung ähnelnde Darstellung an und erlaubt über die AddFolder-Methode eine hierarchische Liste mit Hilfe von Aufklappdreiecken (disclosure triangles). Unter Windows werden statt der Dreiecke Plus- und Minus-Zeichen verwendet.

Name	Art	Beschreibung
InitialValue	<u>String</u>	Eine Liste mit vorbelegten Einträgen, die jeweils durch ein Carriage Return voneinander getrennt sind. Wenn die ListBox mehr als eine Spalte hat (<code>ColumnCount > 1</code>), trennen Sie die Spalteneinträge mit Tabs in jeder Zeile. Wenn <code>HasHeading True</code> ist, wird davon ausgegangen, daß die erste Zeile von <code>InitialValue</code> zum Spaltenkopf wird – bis das <code>Heading</code> -Array ebenfalls gefüllt wird. Diese Eigenschaft kann nicht ausgelesen werden, da die Liste aus dem Speicher entfernt wird, sobald das Steuerelement aufgebaut wurde.
<i>Italic</i>	<u>Boolean</u>	Alle Listeneinträge werden kursiv dargestellt.
LastIndex	<u>Integer</u>	Der Index-Wert der zuletzt über die <code>AddRow</code> - oder <code>AddFolder</code> -Methode hinzugefügten Zeile.
<i>List</i>	Array als <u>String</u>	Die Liste der Listeneinträge. Sie bekommt einen Parameter, den Zeilenindex. Sie können dies verwenden, um den Inhalt dieser Zelle, oder, bei mehrspaltigen Listboxen, der ersten Spalte zu setzen oder zu ermitteln.
ListCount	<u>Integer</u>	Die Anzahl der Listeneinträge
<i>ListIndex</i>	<u>Integer</u>	Das ausgewählte Objekt im Listenindex.
<i>ScrollBarHorizontal</i>	<u>Boolean</u>	Falls <code>True</code> , wird ein horizontaler Scrollbalken der ListBox hinzugefügt. Die Position des Greifers (Thumb) wird durch <code>ScrollPositionX</code> definiert. Wird verwendet, um die ListBox ohne separates <code>ScrollBar</code> -Steuerelement zu scrollen.
<i>ScrollBarVertical</i>	<u>Boolean</u>	Falls <code>True</code> , wird ein vertikaler Scrollbalken der ListBox hinzugefügt. Die Position des Greifers (Thumb) wird durch <code>ScrollPosition</code> definiert.
<i>ScrollPosition</i>	<u>Integer</u>	Null-basierter Index der obersten sichtbaren Zeile in der ListBox. Lesen Sie <code>ScrollPosition</code> aus, um die oberste sichtbare Zeile zu ermitteln; schreiben Sie in <code>ScrollPosition</code> , um die ListBox zu scrollen. Wenn der Greifer (Thumb) bis zum unteren Ende gescrollt ist, kann <code>ScrollPosition</code> nicht weiter vergrößert werden.
<i>ScrollPositionX</i>	<u>Integer</u>	Null-basierter Index der horizontalen Position der ListBox. Wird mit einem <code>ScrollBar</code> -Steuerelement verwendet, um in der ListBox horizontal zu scrollen. Siehe Abschnitt <i>Hinweise</i> .
SelCount	<u>Integer</u>	Die Zahl ausgewählter Zeilen.
<i>Selected</i>	Zeile als <u>Integer</u>	Zeigt an, ob die übergebene Zeile ausgewählt (hervorgehoben) ist oder nicht. Diese Eigenschaft kann verwendet werden, um festzulegen, ob die Zeile selektiert ist und um die Zeile zu selektieren.
SelectionType	<u>Integer</u>	Gibt die Art der erlaubten Auswahl an. 0 = Auswahl einzelner Zeilen, 1 = Auswahl mehrerer Zeilen.
SortedColumn	<u>Integer</u>	Zeigt während der Laufzeit an, nach welcher Spalte sortiert wird.
<i>Text</i>	<u>String</u>	Der Text des momentan ausgewählten Listeneintrags.
<i>TextFont</i>	<u>String</u>	Der Name der zur Anzeige verwendeten Schrift.
<i>TextSize</i>	<u>Integer</u>	Die Größe der zur Anzeige verwendeten Schrift.
<i>Underline</i>	<u>Boolean</u>	Alle Listeneinträge werden unterstrichen dargestellt.
UseFocusRing	<u>Boolean</u>	Wenn <code>True</code> , erscheint das Objekt, falls es den Fokus erhält, mit einem Rahmen um seinen Rand; wenn <code>False</code> , ändert sich das Aussehen des Objekts nicht, wenn es den Fokus erhält.

Events

Name	Parameter	Beschreibung
<i>CellAction</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Falls eine Zelle editierbar ist, tritt ein CellAction-Event auf, sobald der Benutzer diese Zelle editiert. „Editieren“ ist definiert als Verlassen der Zelle, nachdem man in sie geklickt hat. Das Klicken in eine CheckBox in einer CheckBox-Zelle wird auch als „Editieren“ gewertet. Der Benutzer muß nicht unbedingt den Inhalt ändern.
<i>CellBackgroundPaint</i>	g als <u>Graphics</u> , Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Der Parameter g ist ein Graphics-Objekt, das sich auf den Inhalt der Zelle <i>Zeile</i> , <i>Spalte</i> bezieht. 0,0 ist die linke obere Ecke der Zelle. Liefert ein Boolean. True bedeutet, der Anwender hat sich um das Zeichnen des Hintergrundes gekümmert und mit dem Hintergrund wird nichts weiter gemacht. In diesem Fall ist der Anwender für das gesamte Hervorheben verantwortlich. False bedeutet, daß der Anwender möchte, daß REALbasic beim Zeichnen des Hintergrundes helfen soll. REALbasic versucht, die Zeile oder Spalte passend hervorzuheben (je nach Plattform – z.B. Win32, Mac OS 9, Mac OS X – und dem hierarchischen Stil). Es kann dazu kommen, daß schon etwas gezeichnet wurde, bevor CellBackgroundPaint ausgeführt wurde. Es ist also nicht sicher anzunehmen, der Hintergrund wäre weiß. Wenn Sie einen sauberen Hintergrund benötigen, reinigen Sie ihn selbst und liefern True zurück.
<i>CellClick</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u> , X als <u>Integer</u> , Y als <u>Integer</u>	Der Anwender hat in die Zelle <i>Zeile</i> , <i>Spalte</i> geklickt. Die Parameter X, Y sind die Koordinaten des Mausclicks relativ zur linken oberen Ecke der Zelle, in die geklickt wurde. X und Y sind auf der gleichen Referenz-Skala wie die Koordinaten, die von der Graphics-Eigenschaft des CellBackgroundPaint-Events verwendet werden. Liefert ein Boolean. Wenn es True liefert, bedeutet dies, daß das Event nicht weiter bearbeitet wird (das bedeutet, editierbare Zellen werden nicht editierbar sein und eine ListBox-Selektion wird sich nicht ändern).
<i>CellGotFocus</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Der Anwender hat eine editierbare Zelle einer ListBox selektiert. Die Parameter <i>Zeile</i> und <i>Spalte</i> legen fest, welche Zelle gerade den Fokus bekommen hat.
<i>CellKeyDown</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u> , Taste als <u>String</u>	Der Anwender hat eine Taste gedrückt, während eine Zelle in der ListBox editiert wird. Diese Zelle ist durch die Parameter <i>Zeile</i> und <i>Spalte</i> festgelegt. <i>Taste</i> ist die Taste, die der Anwender gedrückt hat. Liefert ein Boolean. Die Rückgabe von True verhindert, daß der Text automatisch geändert wird und unterdrückt das CellTextChanged-Event.
<i>CellLostFocus</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Die Zelle <i>Zeile</i> und <i>Spalte</i> hat den Fokus gerade verloren. Der Anwender kann eine andere Zelle angeklickt oder Return oder esc gedrückt haben.
<i>CellTextChanged</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Tritt nach dem KeyDown-Event auf, wenn das KeyDown-Event False liefert. Das Event übergibt <i>Zeile</i> und <i>Spalte</i> der Zelle, die editiert wurde.

Name	Parameter	Beschreibung
<i>CellTextPaint</i>	g als <u>Graphics</u> , x als <u>Integer</u> , y als <u>Integer</u> , Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Der Parameter g ist ein Graphics-Objekt, das sich auf den Text-Zeichenbereich der Zelle <i>Zeile</i>, <i>Spalte</i> bezieht. Es bezieht sich nicht unbedingt auf den gesamten Zelleninhalt, z. B., wenn Sie ein Zeilen-Bild in der Zelle verwenden. Die Parameter x und y sind die Koordinaten der vermutlich besten Position, um den Text zu zeichnen, basierend auf den aktuellen Werten von ColumnAlignment oder CellAlignment und auch Font, Fontgröße und Font-Stil. Liefert ein Boolean. Die Zeichenfolge der Zelle ist wie folgt, mit dem Hintergrund zuerst: Hintergrund Aufklappdreiecke CheckBox RowPicture Text Border (Rahmen) Auch wenn der Rahmen als letztes gezeichnet wird, ist es nicht empfehlenswert, den Status des Rahmens im CellTextPaint-Event zu ändern, da der Bereich durch die Größe des Rahmens festgelegt wird, bevor die Zelle gezeichnet wurde, so daß ungezeichnete Bereich übrig bleiben könnten oder andere Bereiche, die Sie gezeichnet haben, überdeckt werden könnten. True bedeutet, daß der Anwender das Zeichnen des Textes selbst übernimmt und nichts weiteres mit dem Text gemacht werden soll. In diesem Fall ist der Anwender für das Hervorheben von Text selbst verantwortlich. Text-Hervorhebung gibt es momentan nur beim hierarchischen ListBox-Stil. False bedeutet, daß der Anwender möchte, daß REALbasic den Text zeichnen soll. REALbasic versucht, den Text passend hervorzuheben (passend zur Plattform und dem hierarchischen Stil).
<i>Change</i>		Die Auswahl wurde geändert.
<i>CollapseRow</i>	Zeile als <u>Integer</u>	Der Anwender hat auf das Aufklappdreieck (disclosure triangle) der angegebenen Zeile geklickt.
<i>CompareRows</i>	Zeile1 als <u>Integer</u> , Zeile2 als <u>Integer</u> , Spalte als <u>Integer</u> , ByRef Ergebnis als <u>Integer</u>	Liefert ein Boolean, Das CompareRows-Event ist nützlich für das Sortieren einer Spalte einer ListBox in einer Art und Weise, die nicht von vorneherein angeboten wird. Der vor-eingestellte Mechanismus sortiert alphabetisch. Wenn Sie dieses Event verwenden, wird es während einer ListBox-Sortierung aufgerufen, z.B. wenn der Anwender in den Bereich der Kopfzeile klickt. Parameter: Zeile1: Zeilennummer der einen Zeile, die verglichen werden soll. Zeile2: Zeilennummer der anderen Zeile, die verglichen werden soll. Spalte: Nummer der zu sortierenden Spalte. Ergebnis: 0 – wenn Objekte in Zeile1 und Zeile2 in angegebener Spalte gleich sind. 1 – Inhalte von Zeile1 > Inhalte von Zeile2. -1 – Inhalte von Zeile2 > Inhalte von Zeile1. Geben Sie <u>True</u> zurück, wenn der gelieferte Ergebnis-Parameter für die Sortierung passend ist. Geben Sie <u>False</u> zurück, wenn Sie wollen, daß REALbasic die standardmäßige alphabetische Sortierung für die Spalte verwendet.
<i>DoubleClick</i>		Der Anwender hat einen Doppelklick auf einem Objekt gemacht.

Name	Parameter	Beschreibung
<i>DragReorder-Row</i>	NewPosition als <u>Integer</u>	Tritt auf, wenn eine Zeile durch Draggen eine Umsortierung hervorgerufen hat. Der Parameter <i>NewPosition</i> zeigt die Zeile an, an der die gedraggte Zeile eingefügt werden muß. Liefert ein Boolean. Wird True zurückgegeben, wird das Umsortieren verhindert. Automatisches Umsortieren durch Draggen wird bei hierarchischen ListBoxen nicht unterstützt. Das DragReorderEvent wird zwar ausgeführt, der Anwender ist jedoch verantwortlich für die Umsortierung der Zeilen. Es ist das gleiche Verhalten wie beim Zurückgeben von True im DragReorderRows-Event.
<i>DragRow</i>	Drag als <u>DragItem</u> , Zeile als <u>Integer</u>	Der Anwender zieht (draggt) eine Zeile. Das Objekt dazu ist das automatisch dazu angelegte DragItem. Weisen Sie dem DragItem die Eigenschaften zu, die der Anwender draggen soll. <i>Zeile</i> ist die Zeile der ListBox, die gedraggt wird. Sie müssen in diesem Event-Handler <u>True</u> zurückliefern, damit das Draggen stattfinden kann.
<i>ExpandRow</i>	Zeile als <u>Integer</u>	Der Anwender hat auf das Aufklapppdreieck (disclosure triangle) der angegebenen Zeile geklickt.
<i>GotFocus</i>		Die ListBox hat den Fokus, d.h. der Benutzer hat die ListBox selektiert.
<i>Header-Pressed</i>	Spalte als <u>Integer</u>	Liefert ein Boolean. Wird ausgeführt, nachdem ein ListBox-Spaltenkopf angeklickt/gedrückt wurde. Dieses Event erlaubt es, festzulegen, ob Sie wollen, daß REALbasic die Spalte sortiert. Wenn Sie True zurückgeben, sortiert REALbasic die Spalte nicht und es aktualisiert auch nicht die SortedColumn-Eigenschaft.
<i>KeyDown</i>	Taste als <u>String</u>	Der Anwender hat die angegebene Taste gedrückt, während der Eingabebereich die ListBox war. Liefert ein Boolean. Das Liefern von True bedeutet, daß mit <i>Taste</i> nichts weiter gemacht werden soll.
<i>LostFocus</i>		Die ListBox hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Liefert ein Boolean. Die Maustaste wurde innerhalb der Region der Listbox an der gelieferten Stelle x,y gedrückt. Wenn True im MouseDown-Event zurückgegeben wird, werden die Events MouseDrag und MouseUp ausgelöst und nicht der voreingestellte ListBox-Klick-Prozeß.
<i>MouseDrag</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Listbox-Steuerelements gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region der Listbox an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht <u>True</u> im MouseDown-Event liefern.
<i>SortColumn</i>	Spalte als <u>Integer</u>	Der Benutzer hat auf den Spaltenkopf geklickt, um diese Spalte zu sortieren. Geben Sie <u>True</u> zurück, wenn Sie nicht wollen, daß die ListBox sortiert wird.

Methoden

Name	Parameter	Beschreibung
<i>AddFolder</i>	Objekt als <u>String</u>	Hängt das Objekt in eine neue Zeile ans Ende der Liste an und fügt ebenso ein Aufklapppdreieck ein, wenn die Hierarchical-Eigenschaft True ist. Bei mehrspaltigen ListBoxen wird das Objekt immer an die Spalte 0 gebunden. Im Falle von hierarchischen ListBoxen fügt AddFolder <i>Objekt</i> an die Unterobjekte der aufgeklappten Zeile an, wenn es im ExpandRow-Event aufgerufen wird.
<i>AddRow</i>	Objekt als <u>String</u>	Hängt das Objekt in einer neuen Zeile ans Ende der Liste an. Bei mehrspaltigen ListBoxen wird das Objekt immer an die Spalte 0 gebunden. Im Falle von hierarchischen ListBoxen fügt AddRow <i>Objekt</i> an die Unterobjekte der aufgeklappten Zeile an, wenn es im ExpandRow-Event aufgerufen wird.

Name	Parameter	Beschreibung
<i>Cell</i>	ZeilenNummer als <u>Integer</u> , SpaltenNummer als <u>Integer</u>	Wird verwendet, um auf die angegebene Zelle lesend oder schreibend zugreifen zu können.
<i>CellBold</i>	ZeilenNummer als <u>Integer</u> , SpaltenNummer als <u>Integer</u>	Wird verwendet, um den Text in der angegebenen Zelle fett darzustellen oder den Textstil „fett“ zu entfernen. Ist <u>True</u> , um Text fett darzustellen oder <u>False</u> , um den Text nicht fett darzustellen.
<i>CellItalic</i>	ZeilenNummer als <u>Integer</u> , SpaltenNummer als <u>Integer</u>	Wird verwendet, um den Text in der angegebenen Zelle kursiv darzustellen oder den Textstil „kursiv“ zu entfernen. Ist <u>True</u> , um Text kursiv darzustellen oder <u>False</u> , um den Text nicht kursiv darzustellen.
<i>CellUnderline</i>	ZeilenNummer als <u>Integer</u> , SpaltenNummer als <u>Integer</u>	Wird verwendet, um den Text in der angegebenen Zelle unterstrichen darzustellen oder den Textstil „unterstrichen“ zu entfernen. Ist <u>True</u> , um Text unterstrichen darzustellen oder <u>False</u> , um den Text nicht unterstrichen darzustellen.
<i>ColumnValueProvider</i>	Spalte als <u>Integer</u>	Wird zum Programmieren von eigenen Verknüpfungen verwendet. Liefert <u>StringProvider</u> . Das Rückgabeobjekt beinhaltet ebenso DataNotifier und DataAvailableProvider.
<i>DeleteAllRows</i>		Entfernt alle Zeilen der Liste.
<i>EditCell</i>	Zeile als <u>Integer</u> , Spalte als <u>Integer</u>	Scrollt die Zelle <i>Zeile</i> , <i>Spalte</i> in den sichtbaren Bereich (falls nötig) und macht die Zelle vorrübergehend editierbar. Es setzt den Fokus innerhalb der ListBox auf die Zelle <i>Zeile</i> , <i>Spalte</i> . Siehe auch CellGotFocus-Event. Die editierbare Zelle besitzt einen Fokus-Rahmen.
<i>InsertFolder</i>	ZeilenNummer als <u>Integer</u> , Objekt als <u>String</u>	Erzeugt einen neuen Ordner an der Position <i>ZeilenNummer</i> (und verschiebt die vorhandenen Zeilen nach unten). Bei mehrspaltigen ListBoxen ist <i>Objekt</i> immer der Spalte Null zugeordnet.
<i>InsertRow</i>	ZeilenNummer als <u>Integer</u> , Objekt als <u>String</u>	Legt eine neue Zeile an der Position <i>ZeilenNummer</i> an (wodurch bereits existierende Zeilen nach unten rutschen). Bei mehrspaltigen ListBoxen wird das Objekt immer an die Spalte 0 gebunden.
<i>PressHeader</i>	Spalte als <u>Integer</u>	Führt dazu, daß der ListBox-Spaltenknopf gedrückt wird und führt dadurch dazu, daß ein HeaderPressed-Event auftritt. Der Aufruf dieser Methode aktualisiert nicht die Sortierrichtung.
<i>RemoveRow</i>	ZeilenNummer als <u>Integer</u>	Entfernt die angegebene Zeile.
<i>RowPicture</i>	ZeilenNummer als <u>Integer</u>	Fügt das Bild der Zeile, die durch <i>ZeilenNummer</i> festgelegt wird, hinzu.
<i>SetFocus</i>		Setzt den Eingabebereich auf die ListBox und sorgt so dafür, daß alle Tastendrucke (Events „KeyDown“) an die ListBox geschickt werden.
<i>Sort</i>		Sortiert die Zeilen basierend auf den aktuellen HeadingIndex und ColumnSortDirection. Obwohl HeadingIndex dies auch macht, ist diese Methode zu empfehlen, da HeadingIndex in späteren Versionen von REALbasic nicht mehr sortieren wird.

Hinweise

Auf Objekte in einspaltigen ListBoxen kann über die List-Eigenschaft zugegriffen werden. Die List-Eigenschaft ist ein Array. Da das erste Element eines Arrays mit dem Index 0 angesprochen wird, hat die erste Zeile einer ListBox die Zeilennummer 0 (Null).

Mehrspaltige ListBoxen

Sie können mehrspaltige ListBoxen anlegen, indem Sie die Eigenschaft ColumnCount verändern. Die erste Spalte einer mehrspaltigen ListBox ist die Spalte 0 (null). Dies bedeutet, daß die ColumnCount-Eigenschaft immer um eins größer ist als die Nummer der letzten Spalte. Die maximale Anzahl sichtbarer Spalten ist 64 (Spalten 0 bis 63). Sie sollten ColumnCount auf die Anzahl der anzuzeigenden Spalten setzen. Wenn Sie Daten in unsichtbare Spalten einfügen möchten, setzen Sie die Spaltenbreite auf Null.

Sie können die Eigenschaft InitialValue auch für die vorgelegte Liste mehrspaltiger ListBoxes verwenden, indem Sie Spaltenwerte mit Tabs und die Zeilen mit Zeilenenden (Carriage Return) trennen.

Die Spaltenbreite mehrspaltiger ListBoxes läßt sich mit der Eigenschaft ColumnWidth verändern, indem Sie eine Liste mit durch Kommata getrennten Werten angeben. Die Breiten können in Pixeln oder in prozentualer Schreibweise (relativ zur Breite der ListBox) angegeben werden. Wenn Sie zu wenige Spaltenbreiten angeben, werden die verbleibenden Spalten gleichmäßig über den verbleibenden Platz verteilt. Wenn Sie mehr Spaltenbreiten angeben, als Spalten in Ihrer ListBox vorhanden sind, werden die überflüssigen Werte ignoriert. Wenn die Summe aller Spaltenbreiten größer ist als die Breite der ListBox, werden die verbleibenden Spalten beschnitten.

Mit der Cell-Methode können Sie in einer mehrspaltigen ListBox gezielt auf einzelne Zellen zugreifen.

Dezimale Ausrichtung

Wenn Sie in einer Zelle oder Spalte eine Dezimale Ausrichtung verwenden, müssen Sie beachten, daß REALbasic den Dezimalpunkt mit der rechten Kante der Zelle oder Spalte ausrichtet. Sie müssen eine negative Zahl an CellAlignmentOffset oder ColumnAlignmentOffset übergeben, um Platz für die Zahlen rechts vom Dezimalpunkt zu schaffen. Der korrekte zu übergebende Wert hängt von der Anzahl der Stellen rechts vom Dezimalpunkt in der Spalte oder Zelle ab.

First	Last	Num 1	Num2
Rollin	Hand	1.32	45.▲
James	Phelps	.78	0.▲

Ohne ColumnAlignmentOffset

First	Last	Num 1	Num2
Rollin	Hand	1.32	45.75▲
James	Phelps	.785	0.38▲

**ColumnAlignmentOffset(2)=-32
ColumnAlignmentOffset(3)=-18**

Hierarchische Listboxen

Das Erzeugen hierarchischer ListBoxes ist aufwendiger als eine zweispaltige ListBox, da Sie das Verstecken und Anzeigen der verschaltelten Daten verwalten müssen. Ein einfacher Weg hierzu ist das Zuweisen einer Unterliste an eine „versteckte“ Spalte in der ListBox und die Anzeige der Daten umzuschalten, wenn der Anwender einen Doppelklick auf das Eltern-Element ausführt.

Sie erzeugen eine Zeile mit einem Ausklapdreieck mittels der AddFolder-Methode (anstelle der AddRow-Methode) und setzen dann die Hierarchical-Eigenschaft auf True. Sehen Sie im Beispiel für eine einfache hierarchische ListBox mit einer Ebene nach.

Spalten in der Breite ändern

Es gibt zwei „Modi“, die Spalten in der Größe zu ändern. Es gibt keine formale Modus-Eigenschaft. Vielmehr ist es so, daß der „Modus“ sich daraus ergibt, ob jede Spaltenbreite als ein absoluter Wert festgelegt ist. Wenn Sie alle Spalten entweder in Pixeln oder als einen Prozentwert festlegen, verwenden Sie den zweiten Modus. Wenn Sie den Stern verwenden oder eine Spaltenbreite leer lassen, verwenden Sie den ersten Modus.

- Eine Änderung an einer Spaltenbreite beeinflusst die Breite einer anderen Spalte.
Wenn Spalte i breiter wird, wird Spalte i+1 um den gleichen Wert schmaler. Dieser Modus ist dann geeignet, wenn Sie eine ListBox ohne horizontalen Scrollbalken verwenden. Sie aktivieren diesen Modus, wenn Sie mindestens eine Spaltenbreite haben, die leer ist oder mittels Stern festgelegt ist (z.B. "", " ", "*", oder "4*").
HINWEIS: Sie können die rechte Kante der letzten Spalte in diesem Modus nicht verschieben. Um die letzte Spalte in der Größe zu ändern, müssen Sie die vorige Spalte in der Größe ändern.
- Jede Spaltenbreite ist unabhängig und kann alleine breiter oder schmaler werden.
Sie sind verantwortlich, wenn der Anwender dies macht und Sie müssen einen horizontalen Scrollbalken anbieten, damit der Anwender auch an jeden Spaltenkopf herankommt, der nach rechts aus dem sichtbaren Bereich gerutscht ist. Sie aktivieren diesen Modus, indem Sie sicherstellen, daß jede Spaltenbreite als eine absolute Pixelbreite oder über einen Prozentwert (z.B. "20", oder "35%") festgelegt ist. Wenn Sie einen Stern verwenden oder eine Spaltenbreite leer lassen, verwenden Sie automatisch den ersten Modus.

Sie können zwischen Modus 1 und 2 während der Laufzeit umschalten, indem Sie die gleichen Kriterien verwenden wie oben.

Die `ColumnWidths`-Eigenschaft (Die einzige, die in der IDE eingestellt werden kann), ist äquivalent mit der Verkettung aller `ColumnWidthExpressions`.

`ColumnWidthExpressions` sind Strings und sie können mehrere verschiedene Arten von Breitenberechnung repräsentieren: Absolute Pixel (z.B. "45"), Prozentwerte (z.B. "22.3%") und Stern-Breiten (oder leere) (z.B. " ", "4*").

`ColumnWidthExpressions` behalten Ihren Typ auch wenn eine Spalte in der Breite geändert wurde. Das bedeutet, daß Sie:

- Wenn Sie ein Fenster, mit dem eine `ListBox` verbunden (locked) ist, sich in der Größe ändert, wird diese wachsen oder schrumpfen. Die Spalten wachsen oder schrumpfen auch, wenn die Ausdrücke `*`-basiert (außer Sie verwenden "`0*`") sind oder auf Prozentwerten (0%) basieren. Wenn Sie möchten, daß sie ihre Breite beibehalten, müssen Sie `ColumnWidthExpression` als einen absoluten Pixel-Wert ausdrücken.
- Wenn Sie eine Spalte intern durch Draggen in der Breite ändern, wird diese ihren Prozent- oder Stern-Wert neu berechnen. Dadurch können Sie z.B. mit einer zweispaltigen `ListBox` ohne festgelegter Breite beginnen (jede Spalte verwendet den halben Platz). Danach können Sie eine Spalte so vergrößern, daß sie 3/4 des Raumes einnimmt, danach die `ListBox` vergrößern und nun vergrößert sich die Breite beider Spalten so, daß deren Breitenverhältnis weiterhin 3/4 zu 1/4 beträgt.

Wenn Sie den Pixelwert einer Spalte ändern, ändert das nicht deren fundamentalen Typ, ändert jedoch den Wert dieses Typs.

Wenn Sie Spalten erzeugen möchten, die in der Größe nicht geändert werden können, ändern Sie die `UserResizable`-Eigenschaft jeder dieser Spalten. Wenn Sie `Modus 1` verwenden, müssen Sie die `UserResizable`-Eigenschaften sowohl für diese Spalte als auch für die Spalte links davon ändern.

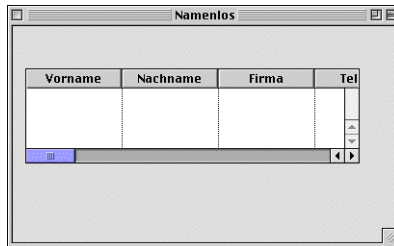
Datenbanken

`ListBoxen` werden oft verwendet, um die Ergebnisse von Datenbankabfragen darzustellen. Eine `ListBox` kann mit einem `DatabaseQuery`-Steuerelement verbunden werden, so daß sie die Ergebnisse der übergebenen SQL-Abfrage automatisch anzeigt. Siehe im Beispiel von Objektverknüpfungen im Kapitel „Datenbankprogrammierung mit REALbasic“ im Entwicklerhandbuch nach. Andernfalls kann eine `ListBox` mit den Ergebnissen einer Abfrage durch Programmierung gefüllt werden. Siehe dazu das Beispiel für `DatabaseField`.

Horizontales Scrollen in einer ListBox

Die `ScrollBarHorizontal`-Eigenschaft fügt einer `ListBox` automatisch einen horizontalen Scrollbalken hinzu. Die `ScrollPositionX`-Eigenschaft kann verwendet werden, um die Position des Thumbs zu ermitteln oder die `ListBox` horizontal zu scrollen. Dies ist der einfachste Weg, um eine `ListBox` zu scrollen.

Sie können außerdem die `ScrollPositionX`-Eigenschaft in Verbindung mit einem `ScrollBar`-Steuerelement verwenden, um den Inhalt der `ListBox` horizontal zu scrollen.



Die `ListBox` hat neun Spalten, acht davon haben eine Breite von 100 Pixeln und eine hat 50 Pixel.

Das `ScrollBar`-Steuerelement besitzt in seinem `Open-Event-Handler` folgenden Code:

```
me.maximum=500
me.minimum=0
me.lineStep=50
```

Die Werte für `Maximum` und `lineStep` wurden so gewählt, daß sie zur Gesamtbreite der Spalten der `ListBox` passen. Ihr `ValueChanged-Event-Handler` besitzt folgende Code-Zeile:

```
ListBox1.scrollTopX=me.value
```

Vorname	Nachname	Firma	Tel
Fred	Feinbein	ASH	0622
Peter	Schlönzke		040
Claudia	Mustermann		030

Geburtstag	Straße	PLZ	Ort
11.5.1966	Pleikartsförsterh.	69126	Heidelberg
5.3.1955	Am Berg 4	2234	Hamburg
22.3.1981	Aufdem Holzweg	12345	Berlin

Benutzerdefinierte Scroll-Steuerelemente

Wenn Sie Miniatur-Scrollbalken möchten oder Scrollbalken, die Platz für eine Statusleiste oder Knöpfe lassen, setzen Sie ScrollBarVertical und/oder ScrollBarHorizontal auf False und verwenden Sie separate ScrollBar-Steuerelemente in Verbindung beliebiger anderer Steuerelemente, die Sie überall dorthin plazieren können, wo normalerweise der Standard-Scrollbalken erscheinen würde. Verwenden Sie die ScrollPosition und ScrollPositionX-Eigenschaften, um das Scrolling zu steuern, so wie es im vorangegangenen Abschnitt beschrieben ist.

Horizontale und vertikale Trennlinien

Folgende Bildschirmkopien zeigen die vier Arten der horizontalen und vertikalen Trennlinien, die mit den Eigenschaften GridLinesHorizontal und GridLinesVertical und den Methoden CellBorderTop, CellBorderLeft, CellBorderRight und CellborderBottom gezeichnet werden können. Die Voreinstellung ist „Keine“.

Name	Telefon	E-Mail
Oliver	8337772	oliver@web.de
Peter	222333	peter@msn.org
Fridolin	1234567	Frido@lin.net
David	984432	David@toll.com

Dünn gepunktet

Name	Telefon	E-Mail
Oliver	8337772	oliver@web.de
Peter	222333	peter@msn.org
Fridolin	1234567	Frido@lin.net
David	984432	David@toll.com

Dünn durchgehend

Name	Telefon	E-Mail
Oliver	8337772	oliver@web.de
Peter	222333	peter@msn.org
Fridolin	1234567	Frido@lin.net
David	984432	David@toll.com

Dick durchgehend

Name	Telefon	E-Mail
Oliver	8337772	oliver@web.de
Peter	222333	peter@msn.org
Fridolin	1234567	Frido@lin.net
David	984432	David@toll.com

Doppelt dünn durchgehend

Mit den Methoden CellBorderTop, ...Left, ...Bottom und ...Right können Sie diese Trennlinien selektierten Zellen oder sogar Zellenrahmen zuweisen. In dieser ListBox zum Beispiel wird eine Zelle, die eine Telefonnummer beinhaltet, mit dickem durchgehendem Rahmen hervorgehoben, während der Rest der Listbox dünne gepunktete Linien verwendet.

Name	Telefon	E-Mail
Oliver	8337772	oliver@web.de
Peter	222333	peter@msn.org
Fridolin	1234567	Frido@lin.net
David	984432	David@toll.com

Beispiele

ListBox1 wird eine Zeile hinzugefügt:

```
ListBox1.addRow "Oktober"
```

In ListBox1 wird in Zeile 1 eine Zeile hinzugefügt:

```
ListBox1.insertRow 1, "Oktober"
```

Stellt alle ListBox-Einträge fett und unterstrichen dar:

```
ListBox1.bold=True
ListBox1.underline=True
```

Kopiert den fünften Eintrag aus ListBox1 in eine Variable:

```
e=ListBox1.list(4)
```

Fügt ListBox1 eine Spalte hinzu und stellt die Breite für die erste Spalte (Spalte 0) auf 50 Pixel, für die zweite Spalte auf 65 Pixel ein:

```
ListBox1.columnCount=2
ListBox1.columnWidths "50,65"
```

Fügt ListBox1 zwei Spalten hinzu und stellt die Spaltenbreiten auf 60%, 20% und 20% zur ListBox-Breite ein:

```
ListBox1.columnCount=3
ListBox1.columnWidths "60%,20%,20%"
```

Wenn ListBox1 100 Pixel breit ist und drei Spalten hat, wird bei folgendem Befehl die dritte Spalte nur 10 Pixel statt der geforderten 20 Pixel breit sein:

```
ListBox1.columnWidths "60,30,20"
```

Wenn ListBox1 100 Pixel breit ist und drei Spalten hat, wird beim folgenden Befehl die dritte Spalte nicht dargestellt:

```
ListBox1.columnWidths "60,40,20"
```

Kopiert die Zelle aus der fünften Zeile und der dritten Spalte aus ListBox1 in eine Variable:

```
e=ListBox1.cell(4,2)
```

Weist der Zelle aus der fünften Zeile und der dritten Spalte von ListBox1 einen Wert zu:

```
ListBox1.cell(4,2)="Fred"
```

Stellt die Zelle in der fünften Zeile und der dritten Spalte von ListBox1 fett und kursiv dar:

```
ListBox1.cellBold(4,2)=True
ListBox1.cellItalic(4,2)=True
```

Setzt das Bild für die fünfte Zeile von ListBox1 auf „MeineBildDatei“:

```
ListBox1.RowPicture(4)=MeineBildDatei
```

Definiert einen Event-Handler „DragRow“, damit der Anwender einen Wert aus einer ListBox ziehen (draggen) kann:

```
Function DragRow(Drag as DragItem, Row as Integer) as Boolean
    Drag.Text=ListBox1.List(Row)
    Return True
End Function
```

Summiert die numerischen Werte der ausgewählten Zeilen:

```
Dim i, total as Integer
For i=0 to ListBox1.ListCount-1
    If ListBox1.Selected(i) then
        total=total+Val(ListBox1.List(i))
    End if
Next
```

Dieses Beispiel klappt die erste Zeile von ListBox1 aus (wenn sie eingeklappt war) oder klappt diese ein (wenn sie ausgeklappt war).

Die Zeile muß über die AddFolder-Methode hinzugefügt worden sein:

```
ListBox1.Expanded(1)=Not ListBox1.Expanded(1)
```

Folgendes Beispiel versieht eine dreispaltige ListBox mit Kopfzeilen:

```
ListBox1.heading(0)="ID"
ListBox1.heading(1)="JobTitle"
ListBox1.heading(2)="Name"
```

Folgendes Beispiel sortiert die ListBox nach der ersten Spalte und zeigt die Nummer der sortierten Spalte in einem EditField:

```
ListBox1.headingindex=0
EditField1.text=Str(ListBox1.headingindex)
```

Dieses Beispiel sortiert die erste Spalte in einer ListBox und zeigt die Nummer der sortierten Spalte in einem EditField an:

```
ListBox1.headingindex=0
EditField1.text=Str(ListBox1.headingindex)
```

Dieses Beispiel richtet eine ListBox mit vier sichtbaren Spalten und einer versteckten ein. Die Spalte Null ist versteckt:

```
Me.columncount=5
Me.columnwidths="0,25%,25%,25%,25%"
Me.heading(0)="ID"
Me.heading(1)="Vorname"
Me.heading(2)="Nachname"
Me.heading(3)="Telefon"
Me.heading(4)="PLZ"
```

Folgende Code-Zeile zeigt den Wert der versteckten Spalte in der gewählten Zeile an:

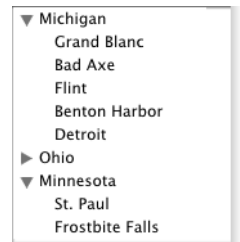
```
MsgBox ListBox1.Cell(ListBox1.ListIndex,0)
```


Hierarchische Listboxen

Folgendes Beispiel erzeugt eine hierarchische ListBox mit einer Ebene, die das Aussehen und Verhalten der Listenanzeige des Finders nachbildet:



Eingeklappte Staaten



Aufgeklappte Staaten

Folgender Code, der in den Open-Event-Handler der ListBox gehört, erzeugt folgende Hierarchie: Der String s1 enthält die Eltern-Ebene und sub1 enthält die Elemente, die mit jedem der Elemente von s1 verschachtelt sind. Es ist eine Liste aus mit Kommas getrennten Listen, wobei jede Liste durch Semikolons getrennt ist. Die Elemente von sub1 sind anfangs versteckt, da sie in einer versteckten Spalte gespeichert werden.

```
Dim i, u as integer
Dim s1, sub1 as string
Me.ColumnWidths="150"
s1="Michigan,Ohio,Minnesota"
sub1="Grand Blanc,Bad Axe,Flint,Benton Harbor,Detroit;Cleveland,Columbus,Akron, Pleasantville;St.
Paul,Frostbite Falls"
u=CountFields(s1,",")
For i=1 to u
    If NthField(sub1,";",i)<> "" then
        me.addFolder ""
        me.cell(i-1,1)=NthField(sub1,";",i)
    end if
    me.cell(i-1,0)=NthField(s1,"",i)
Next
me.ColumnCount=1
```

Beachten Sie, daß anstelle der AddRow die AddFolder-Methode verwendet wird, um die Namen der Staaten hinzuzufügen. Beachten Sie auch, daß die Hierarchical-Eigenschaft der ListBox auf True gesetzt werden muß, damit die Aufklapppreise angezeigt werden.

Folgende Code-Zeile im DoubleClick-Event schaltet die Zeile, auf die ein Doppelklick ausgeführt wird, in den aufgeklappten Zustand um:

```
Me.expanded(Me.listindex)=Not Me.expanded(Me.listindex)
```

Folgender Code im ExpandRow-Event-Handler wird ausgeführt, wenn der Anwender einen Doppelklick auf ein zugeklapptes Element ausführt:

```
Dim s1 as string
Dim i,u as integer
s1=me.cell(row,1)
u=CountFields(s1,",")
For i=1 to u
    Me.addRow ""
    Me.cell(me.lastIndex,0)=NthField(s1,"",i)
Next
```

Es erzeugt die Zeilen der Unterliste jedes Mal, wenn der Anwender einen Doppelklick auf einen zugeklappten Staatennamen ausführt.

Folgender Code gehört in den CollapseRow-Event-Handler:

```
Dim i,u,NSubRows as integer
NSubRows=CountFields(me.cell(row,1),",")
u=row+1
For i=row+NSubRows downto u
    Me.removeRow i
Next
```

Er entfernt die Zeilen, die durch den ExpandRow-Event-Handler erzeugt wurden.

Drag und Drop zwischen Listboxen

Folgendes Beispiel erlaubt dem Anwender, eine oder mehrere Zeilen von ListBox1 in ListBox2 zu draggen. Die gedragten Zeilen werden an das Ende der Liste angefügt.

Die EnableDrag-Eigenschaft von ListBox1 ist auf True gesetzt, damit Elemente in diese Liste gezogen werden dürfen und die SelectionType-Eigenschaft ist auf 1 gesetzt (damit man mehrere Zeilen selektieren kann). Sein DragRow-Event-Handler sieht folgendermaßen aus:

```
Function DragRow (Drag as DragItem, Row as Integer)
    Dim nRows, i as Integer
    nRows=Me.ListCount-1
    For i=0 to nRows
        If Me.Selected(i) then
            Drag.AddItem(0,0,20,4)
            Drag.Text=Me.List(i) //Hole nächsten
        End if
    Next
    Return True //Erlaube Dragging
End Function
```

Es verwendet die Methode AddItem von DragItem, um ein zusätzliches Objekt zu DragItem jeder selektierten Zeile hinzuzufügen. Der DropObject-Event-Handler geht dann durch alle Objekte durch, um alle gedragten Zeilen zu bekommen.

ListBox2 besitzt folgende Code-Zeile im Open-Event-Handler. Es erlaubt das Empfangen von gedragtem Text.

```
Me.AcceptTextDrop
```

Sein DropObject-Event-Handler prüft, ob das gedragte Objekt Text ist; wenn das der Fall ist, fügt es am Ende der Liste eine Zeile an und weist die Text-Eigenschaft des gedragten Objekts der neuen Zeile zu. Es geht durch alle Objekte in DragItem durch bis NextItem False liefert.

```
Sub DropObject(obj as DragItem)
    Do
        If Obj.TextAvailable then
            Me.AddRow(Obj.Text)
        End if
    Loop until Not obj.NextItem
End Sub
```

Sie können auch von ListBox1 auf den Schreibtisch draggen, um einen Text-Clip zu erhalten oder in eine anderen Anwendung, die Drag&Drop von Text akzeptiert, z.B. BBEdit.

Colors

Dieses Beispiel, das in das CellBackgroundPaint-Event gehört, weist den Zeilen in einer ListBox unterschiedliche Farben zu:

```
If row Mod 2=0 then
    g.ForeColor=RGB(158,238,255)
else
    g.ForeColor=RGB(172,255,145)
end if
g.FillRect 0,0,g.width,g.height
```

Hinweise: Das CellBackgroundPaint-Event übergibt die Parameter g (Graphics) und die Zeilen- und Spalten-Nummer (Integer). Sie können eine Farbe zuweisen, indem Sie diese im Farben-Fenster erzeugen und diese Farbe an die Texteingügemarke in den Zeilen draggen, in denen eine Farbe der ForeColor-Eigenschaft zugewiesen wird.

Folgende Zeile im CellTextPaint-Event weist REALbasic an, den Text im vorigen Beispiel rot zu zeichnen:

```
g.ForeColor=RGB(255,0,0)
```

Dem CellTextPaint-Event werden die Koordinaten der gewünschten Startposition für das Zeichnen des Textes mit den Parametern x und y übergeben. Sie können diese in einem Aufruf der DrawString-Eigenschaft verwenden, um festzulegen, daß der String in einer bestimmten Zelle gezeichnet werden soll:

```
If row=4 and column=1 then
    g.ForeColor=RGB(255,0,0)
    g.DrawString "Zahlung überfällig!",x,y
end if
Return True
```

Sortierungen

Um eine `ListBox` zu sortieren, legen Sie die Spalte in der `SortColumn`-Eigenschaft fest, nach der sortiert werden soll. Zusätzlich können Sie die Reihenfolge festlegen, indem Sie die Eigenschaft `ColumnSortDirection` setzen und dann die `Sort`-Methode aufrufen.

Das folgende Beispiel sortiert eine `ListBox`, absteigend nach der ersten Spalte.

```
ListBox1.ColumnSortDirection(0)=-1 //erste Spalte, absteigende Sortierung
ListBox1.SortColumn=0 //erste Spalte ist die Sortier-Spalte
ListBox1.Sort
```

Sie können auch eine Spalte auf Basis der aktuellen `ColumnSortDirection` sortieren, indem Sie die `PressHeader`-Methode aufrufen. Damit wird ein Klicken in den Spaltenkopf der angegebenen Spalte simuliert.

Selbstdefinierte Sortierung

Folgendes Beispiel nutzt das `CompareRows`-Event, um Zahlen-Spalten zu sortieren. Wenn Sie die Standard-Routine verwenden, werden die Zahlen nicht in numerischer Reihenfolge sortiert, das bedeutet, 2 wäre größer als 100 und kleiner als 200.

```
Function CompareRows(row1 as Integer, row2 as Integer, column as Integer, ByRef result as Integer)
as Boolean
    If Val(Me.Cell(row1,column))> Val(Me.cell(row2,column)) then
        result=1
    else
        result=-1
    End if
    Return True //weist RB an, die benutzerdefinierte Routine zu verwenden
End Function
```

Mit diesem Code am richtigen Ort wird eine korrekte numerische Sortierung vorgenommen, wenn der Anwender in die Kopfzeile klickt. Sorgen Sie dafür, daß die benutzerdefinierte Routine nur numerische Spalten beeinflusst.

Um Datums-Einträge zu sortieren, speichern Sie die `TotalSeconds`-Eigenschaft des Datums in einer Spalte der Breite Null und sortieren Sie die Zeilen auf Basis dieser Spalte.

Siehe auch

[DatabaseQuery](#), [EditField](#) Steuerelemente; [RecordSet](#), [DatabaseField](#), [ListColumn](#) Klassen.

ListColumn Klasse

Wird verwendet, um auf eine bestimmte Spalte in einer `ListBox` über deren `Column`-Eigenschaft zu verweisen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>MaxWidthActual</i>	Integer	Die maximale Breite der Spalte (in Pixeln).
<i>MaxWidthExpression</i>	String	Die maximale Breite der Spalte als ein String-Ausdruck, der Leerzeichen und das Prozentzeichen beinhalten kann.
<i>MinWidthActual</i>	Integer	Die minimale Breite der Spalte (in Pixeln).
<i>MinWidthExpression</i>	String	Die minimale Breite der Spalte als ein String-Ausdruck, der Leerzeichen und das Prozentzeichen beinhalten kann.
<i>UserResizable</i>	Boolean	True , falls die Spalte in der Größe änderbar ist; Voreinstellung ist False .
<i>WidthActual</i>	Integer	Die Breite der Spalte in Pixeln.

Name	Typ	Beschreibung
<i>WidthExpression</i>	<u>String</u>	Spaltenbreite als ein String-Ausdruck, der das Prozentzeichen beinhalten kann, einen Dezimalpunkt oder Leerzeichen. Die Breite kann als ein Wert in Pixeln, ein Prozentwert oder eine reative Länge festgelegt werden. Um die absolute Breite einer Spalte zu erhalten, die über einen Prozentwert oder eine relative Länge festgelegt wurde, kann man WidthActual aufrufen. Verwenden Sie * und %, wird die Breite mit der Breite der Liste variieren, sobald diese sich ändert.

Hinweise

Verwenden Sie die UserResizable-Eigenschaft, um diese Eigenschaft zu setzen, z.B.

```
Listbox1.Column(1).UserResizable=False
```

Um Spaltenbreiten zu setzen, können Sie eine Mischung aus Pixel- und Prozentangaben und relativen Längen verwenden. Spaltenbreiten in Pixeln haben garantiert die festgelegte Breite, bis der Anwender die ListBox oder die Spalten in der Breite verändert. Spaltenbreiten, die als Prozentwert festgelegt wurden, haben garantiert diesen Prozentsatz der sichtbaren Breite der ListBox. Die Spaltenbreiten, die mit „*“ festgelegt wurden, teilen sich die verbleibende Breite proportional. Wenn eine Spalte in der Breite geändert wird, wird der Wert dieses Ausdrucks verändert. Wenn Sie eine ListBox in der Größe verändern, werden sowohl die Prozent- als auch die Relativ-Längen neu berechnet. Es gibt zwei Größenänderungs-„Modi“; weitere Informationen finden Sie im Abschnitt „Spalten in der Breite ändern“ beim ListBox-Steuerelement.

Wenn es z.B. drei Spalten mit folgender Festlegung gibt:

```
Listbox1.Column(0).widthExpression = "3*"
Listbox1.Column(1).widthExpression = "*"
Listbox1.Column(2).widthExpression = "10"
```

wird Column(2) ein fester Raum zugewiesen. Die übrige Breite wird in vier Segmente aufgeteilt und im Verhältnis 3 zu 1 zwischen Columns 0 und 1 aufgeteilt. Das bedeutet, daß Column(0) die Breite von ListBox minus 10 mal 3/4 bekommt, Column(1) bekommt die Breite von ListBox minus 10 mal 1/4, und Column(2) bekommt 10 Pixel. Wenn sich die Größe von ListBox ändert (z.B. wenn das Fenster in der Größe geändert wird) ändern Column(0) und Column(1) ihre Breite, um diese Größenänderung zu reflektieren, wohingegen Column(2) unverändert bleibt.

Kopfzeilen-Endstücke werden für nicht verwendeten zusätzlichen Raum hinzugefügt, falls Kopfzeilen verwendet werden. Kopfzeilen-Endstücke machen nichts, wenn man darauf klickt.

Wenn der an WidthExpression, MinWidthExpression oder MaxWidthExpression übergebene String nicht in eine Breite oder prozentuelle Breite ausgewertet werden kann, tritt eine UnsupportedFormatException auf. Die Message-Eigenschaft diese Exception beschreibt in Englisch, was schief.

Beispiel

Folgende Zeile im Open-Event einer ListBox macht alle Spalten in der ListBox in der Größe änderbar:

```
Me.Column(-1).UserResizable=True
```

Folgende Zeile setzt die Maximal-Breite der ersten Spalte in der ListBox auf 75% der Gesamtbreite der ListBox.

```
Listbox1.Column(0).MaxWidthExpression="75%"
```

Siehe auch

ListBox Klasse.

ListSelectionNotificationReceiver Interface Klasse

Wird verwendet, um benutzerdefinierte Objektverknüpfungen zu programmieren. Beispiele dazu finden Sie im Entwicklerhandbuch.

Methoden

Name	Parameter	Beschreibung
<i>SelectionChanging</i>		Die Selektion ändert sich gerade.
<i>SelectionChanged</i>		Die Selektion hat sich geändert.

ListSelectionNotifier Interface Klasse

Wird verwendet, um benutzerdefinierte Objektverknüpfungen zu programmieren.

Methoden

Name	Parameter	Beschreibung
<i>addListSelectionNotificationReceiver</i>	receiver als ListSelection-NotificationReceiver	Fügt einer benutzerdefinierten Interface-Klasse ein Receiver-Objekt hinzu.
<i>removeListSelectionNotificationReceiver</i>	receiver als ListSelection-NotificationReceiver	Entfernt ein Receiver-Objekt aus einer benutzerdefinierten Interface-Klasse.

LittleArrows Steuerelement



Plaziert ein Paar vertikaler Pfeile auf einem Fenster. Sie können zum Beispiel ein LittleArrow-Steuerelement verwenden, um den Wert eines anderen Objekts zu erhöhen oder zu erniedrigen.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Events

Name	Parameter	Beschreibung
<i>Down</i>		Der Benutzer hat auf den Nach-unten-Pfeil geklickt.
<i>Up</i>		Der Benutzer hat auf den Nach-oben-Pfeil geklickt.

Siehe auch

[RectControl](#) Klasse.

Log Funktion

Liefert den natürlichen Logarithmus (Logarithmus der Basis 10) des angegebenen Wertes.

Syntax

Ergebnis = **Log** (Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	Double	Der Logarithmus von <i>Wert</i> .
<i>Wert</i>	Double	Der Wert, dessen Logarithmus Sie erfahren möchten.

Beispiele

Dieses Beispiel der Funktion **Log** liefert den Logarithmus einer Zahl.

```
Dim d As Double
d=Log(10) //liefert 2.3025851
```

Logische Operationen benötigen boole'sche Werte Fehler

Sie verwendeten eine Variable, deren Typ nicht [Boolean](#) war oder einen Ausdruck, der nicht in ein [Boolean](#) resultierte. REALbasic erwartete ein [Boolean](#).

Siehe auch

[Boolean](#) Datentyp

Lowercase Funktion

Wandelt alle Zeichen eines Strings in kleingeschriebene Zeichen um.

Syntax

Ergebnis=**Lowercase**(Wert)

ODER

Ergebnis=**StringVariable.Lowercase**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Eine Kopie des Eingabe-Strings, in dem alle Zeichen kleingeschrieben wurden.
<i>Wert</i>	<u>String</u>	Der Eingabestring.
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ <u>String</u> .

Hinweis

Liefert alle alphabetischen Zeichen kleingeschrieben zurück.

Beispiele

Dieses Beispiel wandelt die Strings in Kleinschreibung um.

```
Dim s As String
s=Lowercase("dEr Schnelle fUCHS") //liefert "der schnelle fuchs"
s=Lowercase("DIE 5 LAHMEN HUNDE") //liefert "die 5 lahmen hunde"
```

Siehe auch

Uppercase, Titlecase Funktionen.

LTrim Funktion

Liefert den übergebenen String ohne führende Leerzeichen zurück.

Syntax

Ergebnis=**LTrim**(sourceString)

ODER

Ergebnis=**StringVariable.LTrim**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der SourceString ohne führende Leerzeichen.
<i>SourceString</i>	<u>String</u>	Der String, der um führende Leerzeichen bereinigt werden soll.
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ <u>String</u> .

Beispiele

Dieses Beispiel entfernt die führenden Leerzeichen (von der linken Seite) des übergebenen Strings:

```
Dim s as String
s=LTrim("  Hallo Welt ")
//liefert „Hallo Welt “
```

Siehe auch

RTrim, Trim, Asc, Chr, InStr, Len, Left, Mid, Right Funktionen.

Max Funktion

Liefert die größere von zwei angegebenen Zahlen.

Syntax

Ergebnis=**Max**(Wert1, Wert2)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Die größere Zahl von <i>Wert1</i> und <i>Wert2</i> .
<i>Wert1</i>	<u>Double</u>	Die erste Zahl für den Vergleich
<i>Wert2</i>	<u>Double</u>	Die zweite Zahl für den Vergleich

Hinweis

Die Funktion **Max** vergleicht zwei Zahlen und liefert die größere von beiden zurück.

Beispiele

Dieses Beispiel verwendet die Funktion **Max**, um den größeren der beiden Werte zurückzuliefern.

```
Dim d As Double
d=Max(3.01, 4.05) //liefert 4.05
d=Max(3.012, 3.011) //liefert 3.012
```

MD5 Funktion

Liefert den MD5 Message-Digest-Wert (Kryptographische Prüfsumme) eines Strings.

Syntax

Ergebnis=**MD5**(String)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die MD5 Message-Digest-Form des übergebenen <u>String</u> .
<i>String</i>	<u>String</u>	Der Eingabe-String, der analysiert werden soll.

Hinweise

Verwenden Sie die MD5-Funktion, um die Roh-Daten zu verarbeiten. Eine vollständige Dokumentation zum MD5 Message-Digest-Format finden Sie in der RFC 1321.

Siehe auch

MD5Digest Klasse.

MD5Digest Klasse

Verarbeitet einen String und liefert die Message-Digest-Form des Input-Strings.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Value</i>	<u>String</u>	Enthält die aktuelle Message-Digest.

Methoden

Name	Parameter	Beschreibung
<i>Process</i>	Data als <u>String</u>	Data ist der Text, der verarbeitet werden soll.
<i>Clear</i>		Setzt das MD5Digest-Objekt zurück, damit Sie mit einem neuen Daten-Strom beginnen können.

Hinweise

Die **MD5Digest**-Klasse erlaubt es Ihnen, einen String in Segmenten zu verarbeiten. Übergeben Sie jedes String-Segment an die *Process*-Methode. Die *Value*-Eigenschaft enthält die aktuelle Message-Digest und die *Clear*-Methode löscht das **MD5Digest**-Objekt, damit man die Verarbeitung wiederholen kann.

Der MD5-Message-Digest-Algorithmus akzeptiert eine Message beliebiger Länge und produziert einen 128-Bit „Fingerprint“ oder *Message-Digest* des Input-Strings. Der MD5-Algorithmus ist nützlich in Applikationen mit Digitalen Signaturen, in denen eine große Datei auf eine sichere Art und Weise verarbeitet werden muß, bevor sie mit einem sicheren Schlüssel unter einem System wie RSA verschlüsselt wird. Eine vollständige Dokumentation finden Sie in der RFC 1321.

Siehe auch

[MD5 Funktion](#).

MDIWindow Klasse

Wird zum Konfigurieren des MDI Fensters im Multiple Document Interface verwendet (nur unter Windows).

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>Handle</i>	Integer	Liefert ein Handle auf das MDI-Fenster. Jedes sichtbare Win32-Steuerelement sollte ein Handle besitzen.
<i>Height</i>	Integer	Die Höhe des MDI-Fensters.
<i>Left</i>	Integer	Der Abstand (in Pixeln) zwischen der linken Kante des Bildschirms und der linken Kante des MDI-Fensters.
<i>MinHeight</i>	Integer	Die minimale Höhe auf die das MDI-Fenster verkleinert werden kann.
<i>MinWidth</i>	Integer	Die minimale Breite auf die das MDI-Fenster verkleinert werden kann.
<i>MaxHeight</i>	Integer	Die maximale Höhe auf die das MDI-Fenster vergrößert werden kann.
<i>MaxWidth</i>	Integer	Die maximale Breite auf die das MDI-Fenster vergrößert werden kann.
<i>Title</i>	String	Der MDI-Fenster-Titel.
<i>Top</i>	Integer	Der Abstand (in Pixeln) zwischen der oberen Kante des Bildschirms und der oberen Kante des MDI-Fensters.
<i>Visible</i>	Boolean	Falls True, ist das MDI-Fenster sichtbar, wenn es geöffnet wird.
<i>Width</i>	Integer	Die Breite des MDI-Fensters.

Events

Name	Parameter	Beschreibung
<i>Moved</i>		Tritt auf, wenn das MDI-Fenster sich bewegt.
<i>Open</i>		Tritt auf, wenn das MDI-Fenster erzeugt wurde.
<i>Resized</i>		Tritt auf, wenn das MDI-Fenster in der Größe verändert wurde.

Methoden

Name	Parameter	Beschreibung
<i>Maximize</i>		Bringt das MDI-Fenster auf volle Größe
<i>Minimize</i>		Bringt das MDI-Fenster auf minimale Größe.
<i>Restore</i>		Stellt die vorige Größe wieder her.

Hinweise

In der Dialogbox zum Erzeugen der Applikation haben Sie die Möglichkeit, die MDI-Option für die Windows-Version Ihrer Applikation zu setzen. Wenn Sie diese wählen, sind alle Fenster Ihrer Applikation in einem „Eltern“-Fenster eingeschlossen, das MDI-Fenster genannt wird. Wenn Sie die MDI-Option nicht anwählen, erscheint das Fenster Ihrer Applikation auf dem Schreibtisch, wie bei einer Macintosh-Applikation. Diese Klasse erlaubt es, bestimmte Eigenschaften und Verhaltensweisen des MDI-Fensters einzustellen.

Beispiel

Die MDIWindow-Klasse ist direkt von der Application-Klasse über das App-Objekt erreichbar. Folgender Code im Open-Event der Applikation stellt einige Eigenschaften des MDI-Fensters ein:

```
App.MDIWindow.Title="REAL Scheduler"  
App.MDIWindow.MinHeight=225  
App.MDIWindow.Height=72
```

Siehe auch

[Application](#) Klasse; [App](#) Objekt.

Me Schlüsselwort

Dies ist ein Verweis (Referenz) auf das Steuerelement des Event-Handlers, der gerade ausgeführt wird. Außerhalb eines Event-Handlers verweist Me auf das aktuelle Objekt ([Self](#)).

Hinweis

Wird **Me** in einem Event-Handler oder einer Methode eines Formular-Steuerelements aufgerufen, liefert es immer einen Verweis auf das Steuerelement, in dessen Event-Handler oder Methode **Me** aufgerufen wurde. **Me** unterscheidet sich nur innerhalb eines Event-Handlers von [Self](#).

Für den Event-Handler „Action“ von PushButton1 bedeutet dies folgendes:

PushButton1.Enabled=[False](#) und **Me.Enabled**=[False](#) bewirken das gleiche. Wenn **Me** innerhalb der Methode einer Klasse aufgerufen wird, ist **Me** ein Verweis auf die Instanz der momentan verwendeten Klasse. Das **Me**-Schlüsselwort unterscheidet sich von [Self](#) insofern, als daß die Funktion [Self](#) immer auf das übergeordnete Objekt und nicht auf das Objekt selbst verweist.

Für Code für ein Steuerelement in einem Fenster verweist „[Self](#)“ auf das Fenster und „Me“ auf das Steuerelement.

Ein großer Vorteil des **Me**-Schlüsselworts zeigt sich, wenn Sie auf das Steuerelement in einem seiner Event-Handler verweisen und den Namen des Steuerelements ändern. In diesem Fall müssen Sie Ihren Code in den Event-Handlern des Steuerelements nicht anpassen. Außerdem können Sie den Code in ein anderes Steuerelement des gleichen Typs kopieren und er wird ohne Änderung funktionieren.

Beispiel

Siehe Beispiel bei [Hierarchical ListBox](#).

Siehe auch

[Self](#) Schlüsselwort.

Me kann nicht in einer Methode oder einem Modul verwendet werden Fehler

Sie können [Self](#) oder [Me](#) nicht in einer Methode oder Funktion verwenden, da es dort kein Eltern-Fenster oder Eltern-Steuerelement gibt. [Self](#) kann nur verwendet werden, wenn es ein Eltern-Fenster für die Methode gibt und [Me kann nur im Event-Handler eines Steuerelements verwendet werden](#).

Siehe auch

[Self](#), [Me](#) Schlüsselwort.

MemoryBlock Klasse

Ein MemoryBlock-Objekt alloziert eine Sequenz von Bytes im Speicher und manipuliert diese Bytes direkt. MemoryBlock-Objekte können an Einstiegspunkte des CFM (Code Fragment Manager) übergeben werden, wo dann die Objekte zu Pointern (Ptr) umdekliert werden.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Parameter	Beschreibung
<i>BooleanValue</i>	Offset als <u>Integer</u>	Ermittelt (0= <u>False</u> ; ≠ 0 ist <u>True</u>) oder setzt (0= <u>False</u> ; 1= <u>True</u>) einen Ein-Byte-Boolean-Wert. Liefert ein Boolean.
<i>Byte</i>	Offset als <u>Integer</u>	Liefert ein Integer.
<i>ColorValue</i>	Offset als <u>Integer</u> , Bits als <u>Integer</u>	Ermittelt oder setzt ein <u>Color</u> in einem von drei Formaten, festgelegt durch <i>Bits</i> . 32: 32-Bit-Farbe (High-Byte nicht benutzt) 24: 24-Bit-Farbe. 16: 16-Bit-Farbe (5 Bit pro Farbe; High-Byte nicht benutzt) Liefert ein Color.
<i>CString</i>	Offset als <u>Integer</u>	Liefert einen <u>String</u> . Das letzte Byte ist \$00.
<i>DoubleValue</i>	Offset als <u>Integer</u>	Liefert oder setzt einen <u>Double</u> -Wert.
<i>LittleEndian</i>		Hiermit können Sie die Bytereihenfolge eines Memoryblocks festlegen. Die Voreinstellung ist die Bytereihenfolge der Plattform, für die der Code kompiliert wurde. Es handelt sich um ein Boolean.
<i>Long</i>	Offset als <u>Integer</u>	Liefert ein <u>Integer</u> mit der Länge vier Bytes.
<i>PString</i>	Offset als <u>Integer</u>	Liefert einen String mit bis zu 255 Zeichen. Das erste Byte ist die Länge des Strings.
<i>Ptr</i>	Offset als <u>Integer</u>	Liefert einen MemoryBlock .
<i>Short</i>	Offset als <u>Integer</u>	Liefert einen vorzeichenbehafteten 16-Bit-Integer.
<i>SingleValue</i>	Offset als <u>Integer</u>	Liefert eine 4-Byte-Repräsentation einer Fließkommazahl mit einfacher Präzision als ein Integer.
<i>StringValue</i>	Offset als <u>Integer</u> , Length als <u>Integer</u>	Ermittelt und setzt einen String mit der angegebenen Größe; kann Nullen enthalten und beginnt nicht mit einem Längen-Byte. Wenn man es setzt und <i>Length</i> größer als die Länge des Strings ist, wird es mit Nullen aufgefüllt.
<i>Size</i>		Liefert die Größe des MemoryBlock in Bytes. Wenn Sie dieser Eigenschaft einen Wert zuweisen, ändert REALbasic die Größe des MemoryBlocks und erhält dabei so viel vorhandene Daten wie möglich. Liefert ein Integer.
<i>UShort</i>	Offset als <u>Integer</u>	Liefert ein nicht vorzeichenbehafteten 16-Bit- <u>Integer</u> .

Methoden

Name	Parameter	Beschreibung
LeftB	Länge als <u>Integer</u>	Liefert einen neuen MemoryBlock der angegebenen Länge zurück. Die Byte-Reihenfolge (endianness) entspricht der des Quell- MemoryBlock .
MidB	Offset als <u>Integer</u> , [Länge als <u>Integer</u>]	Liefert einen neuen MemoryBlock , beginnend ab Offset und mit der angegebenen Länge zurück. Dieser hat die gleiche Byte-Reihenfolge wie der Quell- MemoryBlock .
RightB	Länge als <u>Integer</u>	Liefert einen neuen MemoryBlock der angegebenen Länge. Die Byte-Reihenfolge (endianness) entspricht der des Quell- MemoryBlock .

Hinweise

MemoryBlocks können immer dann verwendet werden, wenn Sie einen Container für beliebige binäre Daten benötigen.

MemoryBlocks werden verwendet, wenn Sie auf Einstiegspunkte von Shared Libraries zugreifen müssen und wenn über die Declare-Funktion OS-Aufrufe machen. Weitere Informationen zu Shared Libraries finden Sie in Apples „Inside Macintosh“ Reihe.

Der Inhalt eines MemoryBlocks kann im Debugger angezeigt werden.

Beispiele

In diesem Beispiel wird eine Gestalt Funktion mit Integer-Parametern deklariert, weshalb Ptr ein Integer zurückliefert:

```
Function Gestalt(setting As String) As Integer
    Dim m as MemoryBlock
    Dim Ergebnis as MemoryBlock
    Dim err as Integer
```

```

m = NewMemoryBlock(10)
Ergebnis = NewMemoryBlock(4)
m.CString(0) = setting

err = InterfaceLib.Gestalt(m.long(0), Ergebnis)

Return Ergebnis.long(0)
End Function

```

Folgendes Beispiel leist eine Real-Zahl in einen MemoryBlock ein und zeigt sie dann an:

```

Dim m as MemoryBlock
Dim d as single
m=NewMemoryBlock(4)
m.singlevalue(0)=123.456
d=m.singlevalue(0)
MsgBox Str(d)

```

Folgendes Beispiel speichert einen String in einen MemoryBlock und zeigt ihn an:

```

Dim m as MemoryBlock
m=NewMemoryBlock(13)
m.byte(0)=12
m.byte(1)=72
m.byte(2)=101
m.byte(3)=108
m.byte(4)=108
m.byte(5)=111
m.byte(6)=32
m.byte(7)=87
m.byte(8)=111
m.byte(9)=114
m.byte(10)=108
m.byte(11)=100
m.byte(12)=33
MsgBox m.pstring(0)

```

Um den String mittels cString auszulesen, fügen Sie vor dem Aufruf das Ende-Byte an, also:

```

m.byte(13)=0
MsgBox m.cString(1)

```

Siehe auch

[NewMemoryBlock](#) Funktion.

Menuitem Klasse

Ein individueller Menüeintrag, der über den Menü-Editor oder als Default dem Bearbeiten-Menü hinzugefügt wurde. Der Beenden-Menüeintrag ist ein besonderer Fall, er wird von der [QuitMenuItem](#)-Klasse abgeleitet.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>AutoEnable</i>	Boolean	Wenn dies auf True gesetzt ist, ist der Menüeintrag als Default aktiviert, so lange das Application-Objekt oder vorderste Fenster einen Menü-Handler für das MenuItem hat. Man muß keinen Code in den EnableMenuItems-Event-Handler schreiben, um den Menüeintrag explizit zu aktivieren. AutoEnable ist in der Voreinstellung True . Wenn Sie dynamische Menüs erzeugen wollen und eine neue Klasse von MenuItem ableiten, so ist auch in dieser abgeleiteten Klasse die Voreinstellung von AutoEnabled True .
<i>BalloonHelp</i>	Message als String	Nachrichtentext, der erscheinen soll, wenn die Sprechblasenhilfe eingeschaltet ist und der Anwender den Mauszeiger über diesen Menüeintrag bewegt, während der Menüeintrag auswählbar ist.

Name	Art	Beschreibung
Bold	<u>Boolean</u>	Gibt an, ob ein Menüeintrag fett dargestellt werden soll.
<i>Checked</i>	<u>Boolean</u>	Gibt an, ob ein Menüeintrag ausgewählt (angekreuzt) dargestellt werden soll.
<i>CommandKey</i>	<u>String</u>	Das Macintosh-Tastaturkürzel für diesen Menüeintrag. Tastaturkürzel für Windows werden durch Verwenden von Konstanten gesetzt. Sehen Sie im Entwicklerhandbuch unter „Verwendung von Konstanten, um Menüs und Menüobjekten Tastaturkürzel hinzuzufügen“.
<i>DisabledBalloonHelp</i>	Message als <u>String</u>	Nachrichtentext, der erscheinen soll, wenn die Sprechblasenhilfe eingeschaltet ist und der Anwender den Mauspfel über diesen Menüeintrag bewegt, während der Menüeintrag nicht auswählbar ist.
<i>Enabled</i>	<u>Boolean</u>	Gibt an, ob dieser Menüeintrag auswählbar (aktiviert) oder nicht auswählbar (deaktiviert) ist. Diese Eigenschaft kann nur aus einem Event-Handler „enableMenuItems“ heraus geändert werden. Sie können einen Menüeintrag entweder aktivieren, indem Sie der Enabled-Eigenschaft True zuweisen oder indem Sie die Enable-Methode aufrufen.
<i>Index</i>	<u>Integer</u>	Die Nummer des Menüeintrags, wenn er Teil eines Arrays ist.
Italic	<u>Boolean</u>	Gibt an, ob ein Menüeintrag kursiv dargestellt werden soll.
Name	<u>String</u>	Der Name des Menüeintrags.
Submenu	<u>Boolean</u>	Zeigt an, daß der Menüeintrag ein Submenü ist.
<i>Tag</i>	<u>Variant</u>	Ein „versteckter“ Textstring, der mit dem Menüeintrag assoziiert ist. Auf den Tag kann man über Code zugreifen, wenn der Anwender den Menüeintrag anwählt, aber anders als bei der Text-Eigenschaft wird es im Menü nicht angezeigt. Es verhält sich wie die RowTag-Eigenschaft eines PopupMenu-Steuerelements.
<i>Text</i>	<u>String</u>	Der Text des Menüeintrags
Underline	<u>Boolean</u>	Gibt an, ob ein Menüeintrag unterstrichen dargestellt werden soll.
<i>Visible</i>	<u>Boolean</u>	Gibt an, ob ein Menüeintrag sichtbar ist. Diese Eigenschaft wird unter Windows nicht unterstützt.

Methoden

Name	Parameter	Beschreibung
<i>Append</i>	newChild als MenuItem	Fügt das übergebene MenuItem dem Menü hinzu.
<i>Child</i>	Name als <u>String</u>	Sucht nach Menüeinträgen über Name oder Text und liefert ein MenuItem .
<i>Close</i>		Entfernt dynamisch angelegte Menüeinträge.
<i>Count</i>		Liefert die Anzahl der Menüeinträge als Integer. Bei einem Menüeintrag liefert es die Anzahl der Submenü-Einträge., wenn es welche gibt. Wenn es keine gibt, liefert es Null.
<i>Enable</i>		Setzt die Enabled-Eigenschaft des Menüeintrags auf True. Rufen Sie Enable nur innerhalb eines EnableMenuItems-Event-Handler auf.
<i>Insert</i>	Index als Integer newChild als MenuItem	Fügt <i>newChild</i> als ein MenuItem an der durch <i>Index</i> festgelegten Position ein.
<i>Item</i>	Index als Integer	Item liefert das n. ^{te} Objekt (nullbasiert) also ein MenuItem. Es wird eine <u>OutOfBoundsException</u> ausgelöst, wenn der angegebene Index außerhalb des Wertebereichs liegt.
<i>Remove</i>	Index als Integer ODER Child als MenuItem	Entfernt den angegebenen MenuItem entweder über seine Position (Index) oder über einen Verweis.

Hinweise

MenuItem-Objekte werden verwendet, um auf die Eigenschaften von Menüeinträgen zugreifen zu können. Sie können den Text des Menüeintrags über die Text-Eigenschaft ändern. Sie können herausfinden, ob ein Menüeintrag mit einem Häkchen versehen ist oder das Häkchen in einem Menüeintrag mit der Checked-Eigenschaft setzen oder löschen. Sie können Menüeinträge mit der Enabled-Eigenschaft aktivieren (auswählbar machen) oder sie deaktivieren. Die Enabled-Eigenschaft kann nur aus einem Event-Handler „EnableMenuItems“ heraus geändert werden. Ändern Sie die Eigenschaft von anderen Codestellen aus, so hat dies keinerlei Wirkung. Zwei weitere Klassen verwalten spezialisierte Menüeinträge: **QuitMenuItem** ist für das Verwalten des Menüs **Ablage/Beenden** einer erzeugten Applikation; es wird automatisch aktiviert und ruft automatisch die **Quit**-Methode auf. Die **PrefsMenuItem**-Klasse ist dazu da, den Voreinstellungen-Menüeintrag zu verwalten. Unter Mac OS X befindet sich dieser Menüeintrag im Applikations-Menü, unter anderen Betriebssystemen existiert dieses Menü jedoch nicht. Ein von der **PrefsMenuItem**-Klasse abgeleiteter Menüeintrag erscheint automatisch im Applikations-Menü unter Mac OS X; unter anderen Betriebssystemen erscheint er da, wo Sie ihn im Menü-Editor positioniert haben.

MenuItem-Objekte können im laufenden Betrieb mit dem Operator **New** angelegt werden. Bei der Beschreibung des Operators **New** finden Sie weitere Informationen.

Unter Macintosh können Sie nicht mehr als 255 Menüs erzeugen.

Beispiele

Das folgende Beispiel ändert den Text des BearbeitenEinfügen-Menüpunkts in „Spezielles Einfügen...“:

```
BearbeitenEinfügen.text="Spezielles Einfügen..."
```

Hier wird ein neuer Menüeintrag unterhalb des Eintrags „Spezielles Einfügen...“ erzeugt.

```
Dim BearbeitenEinfügenSpezial as New MenuItem
BearbeitenEinfügenSpezial.text="Spezielles Einfügen..."
Editmenu.Insert(5,BearbeitenEinfügenSpezial)
```

Im folgenden Beispiel wird ein „Alles auswählen“-Eintrag an das Menü angefügt.

```
Dim AllesAuswählen as New MenuItem
Dim c as New MenuItem
AllesAuswählen.text="Alles auswählen..."
BearbeitenMenu.Append(AllesAuswählen)
```

Das folgende Beispiel ermittelt das **MenuItem** für den „Ausschneiden“-Menüeintrag im Bearbeiten-Menü.

```
Dim c as New MenuItem
c=BearbeitenMenu.child("BearbeitenAusschneiden")
Msgbox c.text
```

Das folgende Beispiel ermittelt das **MenuItem** für den „Ausschneiden“-Menüeintrag im Bearbeiten-Menü über die Position:

```
Dim c as New MenuItem
c=BearbeitenMenu.item(2)
Msgbox c.text
```

Hier wird der vierte dynamisch erzeugte Menüeintrag aus einem MenuItem-Array namens WindowItem entfernt:

```
WindowItem(3).Close
```

Hier wird der Tag-Eigenschaft eines Menüeintrags ein Wert zugewiesen:

```
SearchFind.Tag="UserSearch"
```

Siehe auch

[PrefsMenuItem](#), [QuitMenuItem](#) Klassen; [EnableMenuItems](#) Funktion; [New](#) Operator.

MessageDialog Klasse

Hiermit können Dialogboxen erzeugt werden. Ähnlich der [MsgBox](#)-Funktion, die MessageDialog-Klasse ist jedoch viel flexibler.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
ActionButton	<u>MessageDialogButton</u>	Der Button für die Standardaktion. Die Visible-Eigenschaft ist <u>True</u> in der Voreinstellung und die Beschriftung ist „OK“ oder ein Äquivalent in der jeweiligen Sprache. Um sicher zu gehen, sollten Sie die Beschriftung explizit setzen.
AlternateActionButton	<u>MessageDialogButton</u>	Der Button für eine alternative Aktion. Die Visible-Eigenschaft ist auf <u>False</u> vor-eingestellt. Es gibt keine Standardbeschriftung, deshalb müssen Sie eine Beschriftung angeben, wenn Sie eine alternative Aktion anbieten.
CancelButton	<u>MessageDialogButton</u>	Der optionale Abbrechen-Button. Die Visible-Eigenschaft ist auf <u>False</u> voreingestellt. Die Standardbeschriftung ist „Abbrechen“ oder ein Äquivalent in der jeweiligen Sprache. Um sicher zu gehen, sollten Sie die Beschriftung explizit setzen. Wenn dieser Button gedrückt wird, wird die Cancel-Eigenschaft auf <u>True</u> gesetzt.
Explanation	<u>String</u>	Der erklärende Text, der kleiner als die eigentliche Meldung dargestellt wird (Macintosh). Verwenden Sie diesen Text, um eine genauere Erklärung der Situation oder der Konsequenzen abzugeben. Der Hinweis, daß eine Aktion nicht rückgängig gemacht werden kann, wäre hier angebracht.
Icon	<u>Integer</u>	Eine Ziffer, die den Typ des Icons angibt, das angezeigt werden soll. -1 - Kein Icon 0 - Hinweise-Icon (das Applikations-Icon unter Mac OS X) 1 - Achtung-Dreieck (unter Mac OS X das Applikations-Icon mit einem Warn-dreieck darin) 2 - Stopp-Icon (unter Mac OS X das Applikations-Icon mit einem Stopp-Zei-chen darin) 3 - Frage-Icon (auf dem Macintosh ist dies dasselbe wie 0) Ist der Wert außerhalb dieser Grenzen, wird eine <u>OutOfBoundsException</u> ausgelöst.
Message	<u>String</u>	Die Meldung, die angezeigt werden soll. Geben Sie hier einen kurzen Hinweis auf den Fehler, die Meldung, beziehungsweise der Situation oder Auswahl an. Oft wird diese Meldung als Frage formuliert. Auf dem Macintosh wird dieser Text durch Fettschrift hervorgehoben. Verwenden Sie die Explanation-Eigen-schaft um weitere Informationen anzuzeigen.
Title	<u>String</u>	Text, der in der Titelzeile angezeigt wird (nur Windows und Linux).

Methoden

Name	Parameter	Beschreibung
ShowModal		Zeigt den MessageDialog an. Liefert einen <u>MessageDialogButton</u> , der Aufschluß gibt, wel-cher Button gedrückt wurde. Vergleichen Sie diesen mit den Buttons des Dialogs.

Hinweise

Ein MessageDialog kann bis zu drei Buttons, ein Icon und ein Haupt- und Begleittext enthalten. Unter Windows und Linux ist zusätz-lich eine Titelzeile möglich. In der Voreinstellung ist nur der Standard-Aktionsknopf sichtbar. Möchten Sie die anderen Buttons ver-wenden, setzen Sie bitte deren Visible-Eigenschaft auf True.

Nachdem der Benutzer einen Knopf gedrückt hat, kehrt die ShowModal-Methode zurück und liefert den MessageDialogButton, der gedrückt wurde. Diesen müssen Sie mit den drei möglichen Buttons des Dialogs vergleichen. Beachten Sie das Beispiel.

Beispiel

Das folgende Beispiel erzeugt einen Speichern-Dialog, ohne ein Fenster dafür zu erzeugen.

```
Dim d as New MessageDialog //deklariere das MessageDialog-Objekt
Dim b as MessageDialogButton //Um das Ergebnis zu verwalten
d.icon=1 //zeigt das Warn-Icon
```

```
d.ActionButton.Caption="Speichern"
d.CancelButton.Visible=True //Zeigt den Abbrechen-Knopf
d.AlternateActionButton.Visible=True //zeigt den "Nicht Speichern"-Knopf an
d.AlternateActionButton.Caption="Nicht speichern"
d.Message="Möchten Sie vor dem Schließen die Änderungen am Dokument speichern?"
d.Explanation="Wenn sie nicht speichern, werden Ihre Änderungen verloren gehen."
b=d.ShowModal //den Dialog anzeigen
Select Case b //ermitteln, welcher Knopf gedrückt wurde.
Case d.ActionButton
//Anwender hat Speichern angeklickt
Case d.AlternateActionButton
//Anwender hat Nicht speichern angeklickt
Case d.CancelButton
//Anwender hat Abbrechen angeklickt
End select
```

Siehe auch

[MsgBox](#) Funktion, [MessageDialogButton](#), [Window](#) Klassen.

MessageDialogButton Klasse

Ein Knopf in einem [MessageDialog](#). Die drei möglichen Knöpfe sind: [ActionButton](#), [CancelButton](#) und [AlternateActionButton](#).

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
Cancel	Boolean	Setzen Sie dies auf True , damit der Button auf ESC oder, auf dem Macintosh, auf Kommando-. Kombination reagiert. Nur ein Knopf kann ein Cancel-Button sein. Einige Systeme erlauben es nicht, daß ein Knopf der Standard-Knopf und Abbrechen-Knopf ist.
Caption	String	Die Beschriftung des Knopfs. Setzen Sie einem Zeichen ein „&“ voran, um ein Tastaturkürzel zu erzeugen. Zum Beispiel erzeugt „&Nicht Speichern“ ein Tastaturkürzel für „N“. Auf dem Macintosh wird dazu die Kombination Kommando + Tastaturkürzel gedrückt.
Default	Boolean	Setzen Sie dies auf True , um diesen Knopf zu dem Standardknopf des MessageDialogs zu machen. Der Standardknopf reagiert auf das Drücken von Return. Voreingestellt ist diese Eigenschaft des ActionButtons auf True . Nur ein Button kann der Standard-Knopf sein. Einige Betriebssysteme erlauben es nicht, daß ein Knopf ein Standard-Knopf und Abbrechen-Knopf ist.
Visible	Boolean	Setzen Sie dies auf True , damit der Knopf sichtbar wird. Nur der Standardknopf ist per Voreinstellung sichtbar. Es hat dementsprechend keinen Effekt, diese Eigenschaft dort zu setzen.

Hinweise

Standardmäßig wird nur der [ActionButton](#) dargestellt. Möchten Sie weitere Buttons anzeigen, müssen Sie deren [Visible](#)-Eigenschaft auf [True](#) setzen.

Beispiele

Beachten Sie die Beispiele des [MessageDialogs](#).

Siehe auch

[MsgBox](#) Funktion, [MessageDialog](#), [Window](#) Klassen.

Microseconds Funktion

Liefert die Anzahl von Mikrosekunden (der 1.000.000.te Teil einer Sekunde), die vergangen sind, seit der Rechner des Anwenders eingeschaltet wurde.

Syntax

Ergebnis=**Microseconds**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Die Anzahl an Mikrosekunden, seitdem der Rechner des Anwenders eingeschaltet wurde. Unter Windows ist das Ergebnis nur Millisekunden-genau.

Beispiele

Dieses Beispiel zeigt in einer MessageBox an, seit wie vielen Minuten der Rechner läuft.

```
Dim minutes As Integer
minutes=Microseconds/1000000/60
Msgbox "Ihr Computer ist seit "+Str(minutes)+" Minuten eingeschaltet."
```

Mid Funktion

Liefert einen Teilstring aus einem String. Das erste Zeichen trägt die Nummer 1.

Syntax

Ergebnis=**Mid**(quelle, start [,laenge])

ODER

Ergebnis=**StringVariable.Mid**(start [,laenge])

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der Teilstring aus <i>quelle</i> , der bei <i>start</i> beginnt und <i>laenge</i> Zeichen lang ist (oder alle restlichen Zeichen, falls <i>laenge</i> nicht angegeben wurde).
<i>quelle</i>	<u>String</u>	Der String, aus dem der Teilstring stammen soll.
<i>start</i>	<u>Integer</u>	Die Anfangsposition des Teilstrings. Ist <i>start</i> größer als die Anzahl von Zeichen in <i>quelle</i> , wird ein leerer String zurückgegeben.
<i>laenge</i>	<u>Integer</u>	Optional. Die Anzahl von Zeichen, die der Teilstring aus <i>quelle</i> lang sein soll. Fehlt diese Angabe, so werden alle Zeichen zwischen <i>start</i> und Ende des Strings <i>quelle</i> zurückgeliefert. Wenn die Summe aus <i>laenge</i> und <i>start</i> größer ist als die Gesamtlänge von <i>quelle</i> , so werden alle Zeichen ab <i>start</i> zurückgeliefert.
StringVariable	String	Beliebige Variable vom Typ <u>String</u> .

Hinweise

Um die Anzahl von Zeichen eines Strings zu erfahren, können Sie die Funktion Len verwenden.

Wenn Sie den Teilstring in einer bestimmten Länge von Bytes statt Zeichen benötigen, sollten Sie die Funktion MidB verwenden.

Die Mid-Funktion funktioniert korrekt mit internationalem Text.

Beispiele

Diese Beispiele verwenden die **Mid**-Funktion, um Teilstrings aus einem String zu bilden.

```
Dim s As String
s = Mid("Dies ist ein Test", 6) //liefert "ist ein Test"
s = Mid("Dies ist ein Test", 14, 4) //liefert "Test"
```

Dieses Beispiel konvertiert den Text-String in EditField1 ins hexadezimale und schreibt das Ergebnis in EditField2:

```
Dim i as Integer
EditField2.text=""
For i=1 to LenB(EditField1.text)
    EditField2.text=EditField2.text+"\x"+Hex(Asc(Mid(EditField1.text,i,1)))
Next
```

Siehe auch

Asc, Chr, InStrLeft, Len, Right Funktionen.

MidB Funktion

Liefert einen Teilstring aus einem String. Das erste Zeichen trägt die Nummer 1.

Syntax

Ergebnis=**MidB**(source, start [,laenge])

ODER

Ergebnis=**StringVariable.MidB**(start [,laenge])

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der Teil von <i>quelle</i> , der bei <i>start</i> beginnt und <i>laenge</i> Bytes lang ist (oder alle restlichen Bytes, falls <i>laenge</i> nicht angegeben wurde).
<i>quelle</i>	<u>String</u>	Benötigt. Der String, von dem Bytes zurückgeliefert werden sollen.
<i>start</i>	<u>Integer</u>	Benötigt. Die Position des ersten Bytes, das zurückgeliefert werden soll. Ist <i>start</i> größer als die Anzahl von Bytes in <i>quelle</i> , wird ein leerer String zurückgegeben.
<i>laenge</i>	<u>Integer</u>	Optional. Die Anzahl von Bytes, die aus Quelle zurückgeliefert werden sollen. Fehlt diese Angabe, so werden alle Bytes zwischen <i>start</i> und Ende des Strings <i>quelle</i> zurückgeliefert. Wenn die Summe aus <i>laenge</i> und <i>start</i> größer ist als die Gesamtlänge von <i>quelle</i> , so werden alle Zeichen ab <i>start</i> zurückgeliefert.
StringVariable	String	Beliebige Variable vom Typ <u>String</u> .

Hinweise

MidB behandelt *string* als eine Reihe von Bytes und nicht Buchstaben. Es sollte verwendet werden, wenn *string* aus binären Daten besteht. Wenn Sie Buchstaben statt Bytes aus dem String benötigen, sollten Sie die Funktion Mid verwenden.

Um die Anzahl von Bytes eines Strings zu erfahren, können Sie die Funktion LenB verwenden.

Beispiele

Diese Beispiele verwenden die **MidB**-Funktion, um Abschnitte aus einem String zu liefern.

```
Dim s As String
s = MidB("Dies ist ein Test", 6)      //liefert "ist ein Test"
s = MidB("Dies ist ein Test" 14, 4) //liefert "Test"
```

Siehe auch

AscB, ChrB, InStrB, LeftB, LenB, Mid, RightB Funktionen.

Min Funktion

Liefert die kleinere von zwei angegebenen Zahlen.

Syntax

Ergebnis=**Min**(Wert1, Wert2)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Die kleinere Zahl von <i>Wert1</i> und <i>Wert2</i> .
<i>Wert1</i>	<u>Double</u>	Die erste Zahl für den Vergleich.
<i>Wert2</i>	<u>Double</u>	Die zweite Zahl für den Vergleich.

Hinweis

Die Funktion **Min** vergleicht zwei Zahlen und liefert die kleinere von beiden zurück.

Beispiele

Dieses Beispiel verwendet die Funktion **Min**, um den kleineren der beiden Werte zurückzuliefern.

```
Dim d As Double
d=Min(3.01, 4.05) //liefert 3.01
d=Min(3.012, 3.011) //liefert 3.011
```

Mit #if können nur Konstanten vom Typ Boolean verwendet werden Fehler

Nur die boole'schen Konstanten [TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetWin32](#), [DebugBuild](#), [TargetHasGUI](#), [RBVersion](#), [TargetLinux](#) und [RBVersionString](#) können in dem `#if`-Befehl verwendet werden.

Beispiel

Eine „reguläre“ boole'sche Variable kann nicht wie hier gezeigt verwendet werden.

```
Dim b as Boolean
b=True
#if b then
  Beep
#endif
```

Siehe auch

[TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetWin32](#), [DebugBuild](#), [Target68K](#), [TargetLinux](#), [RBVersion](#), [RBVersionString](#) Konstanten; [#If](#) Ausdruck.

Mod Operator

Liefert den nicht-teilbaren Rest nach Teilung zweier Zahlen.

Syntax

Ergebnis= Zahl1 **Mod** Zahl2

Teil	Beschreibung
<i>Ergebnis</i>	Der Rest (als Integer) von Zahl1 geteilt durch Zahl2.
<i>Zahl1</i>	Eine beliebige Zahl
<i>Zahl2</i>	Eine beliebige Zahl

Der Operator **Mod** teilt Zahl1 durch Zahl2 und liefert den nicht-teilbaren Rest als Ergebnis. Wenn Zahl2 Null ist, liefert **Mod** Zahl1.

Der Operator **Mod** arbeitet mit Integern. Wenn entweder Zahl1 oder Zahl2 Nicht-Integer sind, werden sie zuvor zu Integern durch Abschneiden der Nachkommastellen gemacht. Zum Beispiel:

```
5 Mod 2 ergibt 1
5 Mod 1.99999 ergibt 0
```

Beispiele

Diese Beispiele verwenden den Operator Mod, um zwei Zahlen zu teilen und den nicht-teilbaren Rest zurückzuliefern.

```
Dim r As Integer
r=10 Mod 3 //ergibt 1
r=2 Mod 4 //ergibt 2
r=9.3 Mod 2.75 //ergibt 1
r=4.5 Mod 1 //ergibt 0
r=25 Mod 5 //ergibt 0
```

Siehe auch

[\](#), [\](#) Operatoren.

MouseCursor Klasse

Wird verwendet, um das Aussehen des Mauszeigers festzulegen.

Übergeordnete Klasse

[Object](#)

Hinweise

Schauen Sie bei den MouseCursor-Eigenschaften der Klassen [Application](#), [Window](#) und [Control](#) nach. Die Beziehung zwischen diesen ist folgendermaßen:

Die [MouseCursor](#)-Eigenschaft eines Steuerelements legt die Form des Mauszeigers fest, wenn dieser den Bereich eines Steuerelements betritt, jedoch nur, wenn die [MouseCursor](#)-Eigenschaften der [Application](#)- und [Window](#)-Klasse [Nil](#) ist. Ähnlich dazu legt die [MouseCursor](#)-Eigenschaft der [Window](#)-Klasse die Form des Mauszeigers fest, wenn er den Bereich des Fensters betritt, nur wenn die [MouseCursor](#)-Eigenschaft der [Application](#)-Klasse [Nil](#) ist. Wenn die [MouseCursor](#)-Eigenschaft der [Application](#)-Klasse nicht [Nil](#) ist, dann steuert es die Form des Zeigers und die Werte der [MouseCursor](#)-Eigenschaften eines Fensters oder Steuerelements werden ignoriert.

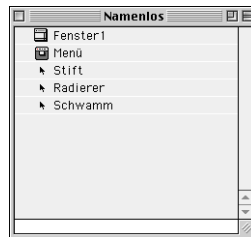
Ein [MouseCursor](#) kann auf drei Arten zugewiesen werden:

- Sie können die Namen [WatchCursor](#), [IBeamCursor](#) oder [ArrowCursor](#) verwenden.
- Sie können eine [Resource](#)-Datei in das Projektfenster ziehen, die eine [CURS](#)-Resource enthält und die [ResourceFork.GetCursor](#)-Methode verwenden.
- Sie können den [ResourceFork](#) einer Datei öffnen und eine [CURS](#)- oder [crsr](#)-Resource mit der [ResourceFork.GetCursor](#)-Methode verwenden.

Benutzerdefinierte Maus-Cursor

Sie können benutzerdefinierte Maus-Cursor in Ihrem Programm verwenden, wenn Sie folgende Technik verwenden:

Erzeugen Sie eine separate [CURS](#)-Resource, die nur einen Cursor für jeden benutzerdefinierten Cursor enthält, den Sie verwenden wollen. Geben Sie dieser [Resource](#)-Datei einen unterschiedlichen Namen und ziehen Sie diese ins Projekt-Fenster. Diese speziellen [Resource](#)-Dateien werden im Projekt-Fenster mit einem Pfeil-Icon angezeigt.



Sie können dann auf jeden benutzerdefinierten Cursor über den Namen zugreifen, z.B.:

```
Self.MouseCursor=Stift
```

Siehe auch

[MouseCursor](#)-Eigenschaft der Klassen [Application](#), [Window](#) und [Control](#).

Movie Klasse

Ein Verweis auf einen QuickTime-Film. Objekte der [Movie](#) Klasse haben keine Eigenschaften, Methoden oder Events.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>BaseMovieHeight</i>	Integer	Tatsächliche Höhe des Filmes in Pixeln.
<i>BaseMovieWidth</i>	Integer	Tatsächliche Breite des Filmes in Pixeln
<i>Handle</i>	Integer	Handle auf den Film, hauptsächlich für Toolbox-Aufrufe verwendet, die ein Handle auf den Film.
<i>MovieHandle</i>	Integer	Handle auf den Film. Bietet die gleiche Funktionalität wie <i>Handle</i> , wird aber vielleicht in künftigen Versionen von REALbasic nicht mehr unterstützt. Veraltet, verwenden Sie stattdessen <i>Handle</i> .
<i>MovieHeight</i>	Integer	Höhe des QuickTime-Filmes in Pixeln.
<i>MovieWidth</i>	Integer	Breite des QuickTime-Filmes in Pixeln.

Hinweis

Auf QuickTime-Filme kann über die Eigenschaft „Movie“ des MoviePlayer Steuerelements oder mit der OpenAsMovie-Methode des FolderItem-Objekts zugegriffen werden.

Beispiele

Folgendes Beispiel öffnet ein QuickTime-Movie auf der Festplatte, weist dieses der Movie-Eigenschaft eines MoviePlayer-Objekts zu und zeigt seine Höhe und Breite in EditFields an.

```
Dim f as FolderItem
Dim m as Movie
f=GetFolderItem("Logo.mov")
m=f.OpenAsMovie
moviePlayer1.border=true
moviePlayer1.movie=m
moviePlayer1.controller=2
EditField2.text=str(m.movieheight)
EditField3.text=str(m.moviewidth)
```

Siehe auch

OpenURLMovie Funktion; EditableMovie, QTTrack Klassen; MoviePlayer Steuerelement.

MoviePlayer Control



Zeigt einen Bereich an, in dem QuickTime-Filme abgespielt werden.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>AutoResize</i>	<u>Boolean</u>	Wenn True, ändert das MoviePlayer-Steuerelement seine Größe, so daß es der Größe des Films entspricht.
<i>Border</i>	<u>Boolean</u>	Zeichnet um den Abspielbereich einen Rahmen.
<i>Controller</i>	<u>Integer</u>	0=keine Steuerelemente, 1=Abspielknopf, 2=vollständige Steuerelemente.
<i>ControllerHeight</i>	<u>Integer</u>	Höhe des Steuerfelds (in Pixeln).
<i>ControllerWidth</i>	<u>Integer</u>	Breite des Steuerfelds (in Pixeln).
<i>EditingEnabled</i>	<u>Boolean</u>	Falls True, ist es möglich, den Film zu bearbeiten. Die Methoden eines MoviePlayers zum Bearbeiten eines Filmes funktionieren, wenn EditingEnabled True ist.
<i>HasStep</i>	<u>Boolean</u>	Wenn True, besitzt das Steuerelement Pfeile für Vor und Zurück rechts vom Schieberegler. Wenn dies im Code gesetzt wird, wirkt sich HasStep das <i>nächste</i> Mal aus, wenn dem Movie-Player ein Film zugewiesen wird.
<i>Looping</i>	<u>Boolean</u>	Spielt den Film erneut, sobald er das Ende erreicht hat.
<i>Movie</i>	<u>Movie</u>	Der abzuspielende QuickTime-Film.
<i>Palindrome</i>	<u>Boolean</u>	Spielt den Film rückwärts, sobald er das Ende erreicht hat. Die Funktion „Looping“ muß hierzu eingeschaltet sein.
<i>PlaySelection</i>	<u>Boolean</u>	Falls eine Auswahl vorgenommen wurde (mittels SelStart und SelLength) und diese Eigenschaft auf True gesetzt ist, wird nur die Auswahl abgespielt. Falls es False ist, wird es normal abgespielt.
<i>Position</i>	<u>Double</u>	Anzahl der Sekunden nach Beginn des Movies.

Name	Art	Beschreibung
<i>QTMovieController</i>	<u>Integer</u>	Liefert einen Pointer, der auf das MoviePlayer-Steuerelement verweist. Zur Verwendung für Toolbox-Programmierer.
<i>QTVRNode</i>	<u>Integer</u>	Der Knoten, der derzeit in der Steueranzeige angezeigt wird.
<i>QTVRNodeCount</i>	<u>Integer</u>	Die Anzahl der Knoten des Filmes des MoviePlayers.
<i>QTVRPan</i>	<u>Double</u>	Der Kameraschwenk-Winkel des QTRV-Films des MoviePlayers in Grad.
<i>QTVRPanMax</i>	<u>Double</u>	Der maximale Kameraschwenk-Winkel des QTRV-Films des MoviePlayers in Grad.
<i>QTVRPanMin</i>	<u>Double</u>	Der minimale Kameraschwenk-Winkel des QTRV-Films des MoviePlayers in Grad.
<i>QTVRPanTiltSpeed</i>	<u>Integer</u>	Die Geschwindigkeit der Kamer-Kipp-Bewegung (Winkel in Grad) des QTRV-Films des MoviePlayers.
<i>QTVRTilt</i>	<u>Double</u>	Der Kamera-Kipp-Winkel (Winkel in Grad) des QTRV-Films des MoviePlayers.
<i>QTVRTiltMax</i>	<u>Double</u>	Der maximale Kamera-Kipp-Winkel (Winkel in Grad) des QTRV-Films des MoviePlayers.
<i>QTVRTiltMin</i>	<u>Double</u>	Der minimale Kamera-Kipp-Winkel (Winkel in Grad) des QTRV-Films des MoviePlayers.
<i>QTVRZoom</i>	<u>Double</u>	Das gewünschte vertikale Blickfeld für den festgelegten Film. Der Wert ist durch das maximale Blickfeld des Filmes begrenzt. Werte, die außerhalb dieser Begrenzung liegen, werden auf das Maximum abgeschnitten. Schwenk- und Kipp-Winkel-Werte werden ebenso abgeschnitten, falls es, wenn mit dem aktuellen Blickfeld kombiniert, dazu führen würde, daß ein Bild außerhalb der aktuellen Begrenzung liegen würde.
<i>QTVRZoomMax</i>	<u>Double</u>	Der maximale Blickfeld-Zoom.
<i>QTVRZoomMin</i>	<u>Double</u>	Der minimale Blickfeld-Zoom.
<i>QTVRZoomSpeed</i>	<u>Integer</u>	Die Rate, mit der Sie zoomen können.
<i>Rate</i>	<u>Double</u>	Stellt die Abspielrate ein. Folgende Werte haben z.B. folgenden Effekt: 0 – Pause/gestoppt 0.5 – Halbe normale Abspielrate 1 – Normale Abspielrate 2 – Doppelte normale Abspielrate -1 – Rückwärts normale Abspielrate Da Rate ein Double ist, funktionieren auch andere Werte, einschließlich Werten außerhalb dieses Bereichs.
<i>SelLength</i>	<u>Double</u>	Ermittelt oder setzt die Länge der Auswahl (in Sekunden).
<i>SelStart</i>	<u>Double</u>	Ermittelt oder setzt die Startposition der Auswahl (in Sekunden).
<i>Speaker</i>	<u>Boolean</u>	Zeigt den QuickTime-Lautstärkeregler.
<i>Volume</i>	<u>Integer</u>	Ermittelt oder setzt die Lautstärke des Films. Der Bereich reicht von 0 bis 256.

Events

Name	Beschreibung
<i>ControllerSizeChanged</i>	Die Größe des Steuerfelds hat sich geändert. Wenn ein Movie gestartet wird, wird dieses Event ausgelöst und erlaubt es, die Eigenschaften ControllerWidth/Height auszulesen und dann die Ausmaße des MoviePlayers oder des Fensters wunschgemäß zu verändern. Das Event tritt ebenfalls auf, wenn das Movie „von sich aus“ die Größe ändert (nur QT4).
<i>Play</i>	Der aktuelle Film wird gleich starten.
<i>Stop</i>	Der aktuelle Film wird gleich angehalten.

Methoden

Name	Parameter	Beschreibung
<i>Clear</i>		Löscht die aktuelle Auswahl des Films. Setzt die EditingEnabled-Eigenschaft auf True, um es dem Anwender zu ermöglichen, eine Auswahl mit den Steuerelementen des MoviePlayers vorzunehmen. Wenn der Film kein EditableMovie ist, können Änderungen nicht gespeichert werden.
<i>Copy</i>		Kopiert die aktuelle Auswahl des Films in die Zwischenablage. Setzt die EditingEnabled-Eigenschaft auf True, um es dem Anwender zu ermöglichen, eine Auswahl mit den Steuerelementen des MoviePlayers vorzunehmen. Wenn der Film kein EditableMovie ist, können Änderungen nicht gespeichert werden.
<i>Cut</i>		Schneidet die aktuelle Auswahl des Films aus und plaziert diese in der Zwischenablage. Setzt die EditingEnabled-Eigenschaft auf True, um es dem Anwender zu ermöglichen, eine Auswahl mit den Steuerelementen des MoviePlayers vorzunehmen. Wenn der Film kein EditableMovie ist, können Änderungen nicht gespeichert werden.
<i>Paste</i>		Fügt den Inhalt der Zwischenablage an der aktuellen Auswahl des Films ein. Setzt die EditingEnabled-Eigenschaft auf True, um es dem Anwender zu ermöglichen, eine Auswahl mit den Steuerelementen des MoviePlayers vorzunehmen. Wenn der Film kein EditableMovie ist, können Änderungen nicht gespeichert werden.
<i>Play</i>		Spielt den aktuellen Film.
<i>QTVRHotSpotCount</i>	nodeID als Integer	Liefert ein Integer, die Anzahl der QTVR-Hot-Spots in dem Film mit dem in nodeID übergebenen Knoten.
<i>QTVRHotSpotID</i>	Knoten als Integer, index als Integer	Liefert ein Integer, die ID eines bestimmten Hot-Spots, basierend auf den als Parametern übergebenen Knoten-Nummer und Hot-Spot-Index.
<i>QTVRNodeTypeObject</i>	node als Integer	Legt fest, ob der aktuelle Film ein QTVR-Objekt enthält. Liefert ein Boolean; True, wenn der Film ein QTVR-Objekt enthält.
<i>QTVRNodeTypePanorama</i>	node als Integer	Legt fest, ob der aktuelle Film ein QTVR-Panorama enthält. Liefert ein Boolean; True, wenn der Film ein QTVR-Panorama enthält.
<i>QTVRToggleHotSpotNames</i>		Schaltet um, ob Hot-Spot-Namen bei den Film-Steuerelementen angezeigt werden.
<i>QTVRTriggerHotSpot</i>	hotSpotID als Integer	Navigiert zum angegebenen Hot-Spot im QTVR-Film des MoviePlayers.
<i>Stop</i>		Unterbricht den aktuellen Film.
<i>Trim</i>		Behält die aktuelle Auswahl bei (festgelegt durch SelStart und SelLength) und entfernt alle anderen Medien. Setzt die EditingEnabled-Eigenschaft auf True, um es dem Anwender zu ermöglichen, eine Auswahl mit den Steuerelementen des MoviePlayers vorzunehmen. Wenn der Film kein EditableMovie ist, können Änderungen nicht gespeichert werden.
<i>Undo</i>		Macht die letzte Aktion rückgängig. Dies ist eigentlich ein Undo/Redo-Umschalten. Setzt die EditingEnabled-Eigenschaft auf True, um es dem Anwender zu ermöglichen, eine Auswahl mit den Steuerelementen des MoviePlayers vorzunehmen. Wenn der Film kein EditableMovie ist, können Änderungen nicht gespeichert werden.

Hinweis

Die Controller-Eigenschaft schreibt QuickTime vor, wie und welche QuickTime-Steuerelemente angezeigt werden sollen. Übergeben Sie 0 (null), so gibt es keine Steuerelemente für den Anwender. Übergeben Sie 1, so wird ein Film-Icon oder ein Abspielknopf in der linken unteren Ecke des Abspielbereichs dargestellt. Wenn dieser Abspielknopf vom Anwender angeklickt wird, verschwindet er und die üblichen QuickTime-Steuerelemente erscheinen am unteren Rand des Abspielbereichs. Übergeben Sie 2, so erscheinen die üblichen QuickTime-Steuerelemente.

Beispiele

Dieses Beispiel trägt den QuickTime-Film „MacintoshJr“ als den abzuspielenden Film ein.

```
MoviePlayer1.movie=MacintoshJr
```

Dieses Beispiel lädt einen QuickTime-Film „MeinFilm“ aus dem aktuellen Ordner in das Steuerelement MoviePlayer1 und spielt ihn ab.

```
Dim f As FolderItem
f=New FolderItem
f=GetFolderItem("MeinFilm")
MoviePlayer1.Speaker=True
MoviePlayer1.HasStep=False
MoviePlayer1.movie=f.OpenAsMovie
MoviePlayer1.play
```

Wenn Sie das Movie in Ihr Projektfenster gedraggt haben, können Sie es mit einer einzigen Zeile Code der Movie-Eigenschaft zuweisen:

```
MoviePlayer1.movie=MyMovie
```

Wenn Sie Objektverknüpfungen verwenden, können Sie PushButtons hinzufügen, die den Film abspielen oder stoppen. Schauen Sie hierzu im Abschnitt über den MoviePlayer im Entwicklerhandbuch nach.

Siehe auch

[OpenURLMovie](#) Funktion; [Movie](#), [EditableMovie](#) Klassen.

MsgBox Funktion

Zeigt eine Dialogbox mit dem angegebenen [String](#) an. Mit Ausnahme einfacher Nachrichten sollten Sie die [MessageDialog](#) Klasse verwenden.

Syntax

Ergebnis=**MsgBox**(message,[buttons],[title])

Teil	Typ	Beschreibung
<i>Ergebnis</i>	Integer	Gibt an, welcher Knopf gedrückt wurde. Wenn Sie keine optionalen Parameter verwenden, brauchen Sie das Ergebnis, wie unter REALbasic vor Version 5.5, nicht zu beachten. 1 – OK gedrückt 2 – Abbrechen gedrückt 3 – Verlassen gedrückt 4 – Nochmal versuchen gedrückt 5 – Ignorieren gedrückt 6 – Ja gedrückt 7 – Nein gedrückt
<i>message</i>	String	Dieser Parameter ist die Meldung, die mit diesem Dialog angezeigt wird. Ist im String ein Zeilenumbruch enthalten, wird auf dem Macintosh der nachstehende Text kleiner dargestellt als die erste Zeile.
<i>buttons</i>	Integer	Optionaler Code, der angibt, welches Icon und welche Knöpfe dargestellt werden sollen. Sie müssen das Ergebnis verarbeiten, wenn Sie diesen Parameter angeben.
<i>title</i>	String	Optionaler Text, der in der Titelzeile angezeigt wird (nur Windows) Titel werden unter Macintosh nicht angezeigt. Sie müssen das Ergebnis verarbeiten, wenn Sie einen Titel angeben, da Sie ebenfalls den buttons-Parameter angeben müssen.

Hinweise

Mit REALbasic 5.5 kann eine Nachricht über die **MsgBox**-Funktion oder über die [MessageDialog](#)-Klasse angezeigt werden. Die **MsgBox** Funktion wird für einfache informative Meldungen empfohlen. Für andere Situationen sollten Sie die [MessageDialog](#)-Klasse verwenden. Diese bietet Ihnen die Möglichkeit, bis zu drei Knöpfe mit freier Beschriftung darzustellen. **MsgBox** bietet sich auch an, wenn Sie VisualBasic-Programme nach REALbasic portieren wollen.

Der Dialog, der von **MsgBox** geöffnet wird, hat eine Titelzeile. Unter Windows hat der Dialog zusätzlich ein Schließen-Knopf in der Titelzeile. Der Dialog kann über beide geschlossen werden. Auf dem Macintosh ist dieser Dialog von fester Größe, unter Windows richtet sich die Größe nach dem längsten Absatz. Auf dem Macintosh wird der Absatz umbrochen.

Mehrere Absätze können im **message**-Parameter enthalten sein, wenn diese mittels der EndOfLine Funktion voneinander getrennt sind.

Der Buttons-Parameter

Buttons ist ein optionaler Parameter, der Ihnen die Möglichkeit gibt, festzulegen, welche Knöpfe und welches Icon angezeigt werden soll. Der Wert von Buttons ist die Summe der Werte, die den einzelnen Buttons zugewiesen ist plus dem Icon und dem Standardknopf. Es gibt drei Gruppen von Knöpfen, aus denen Sie wählen können.

Die erste Gruppe: Nummern und Typen der Knöpfe:

Wert	Beschreibung
0	Nur OK-Button anzeigen
1	OK- und Abbrechen-Button anzeigen
2	Abbrechen-, Nochmal versuchen- und Ignorieren-Knöpfe anzeigen.
3	Ja-, Nein- und Abbrechen-Knöpfe anzeigen.
4	Ja- und Nein-Knöpfe anzeigen.
5	Nochmal versuchen- und Abbrechen-Knöpfe anzeigen.

Die zweite Gruppe spezifiziert das anzuzeigende Icon

Wert	Beschreibung
0	Kein Icon
16	Stoppzeichen-Icon
32	Frage-Icon
48	Achtung-Dreieck
64	Hinweis-Icon

Die dritte Gruppe entscheidet über den Standardknopf. Die Beschreibung Erster, Zweiter und Dritter meinen nicht unbedingt die angezeigte Reihenfolge, sondern die Reihenfolge in der Tabelle für Gruppe 1.

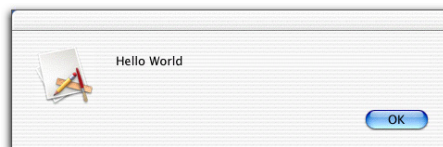
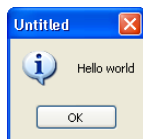
Wert	Beschreibung
0	Erster Knopf der Gruppe 1 ist Standard.
256	Zweiter Knopf der Gruppe 1 ist Standard.
512	Dritter Knopf der Gruppe 1 ist Standard.
768	Kein Knopf ist der Standard-Knopf.

Da der Wert des Button-Parameters der Summe der Werte der Tabellen entspricht, können Sie einfach eine Auswahl treffen und diese Werte zusammenzählen. Wenn Sie zum Beispiel einen „Abbrechen-“, „Nochmal versuchen“- und „Ignorieren“-Knopf anzeigen wollen, rechnen Sie $2 + 48 + 256 = 306$. Dies zeigt, warum Sie die **MessageDialog**-Klasse für Dialoge verwenden sollten. Dort können Sie die Beschriftung für jeden einzelnen Knopf setzen und festlegen, welcher Knopf dargestellt werden soll.

Beispiele

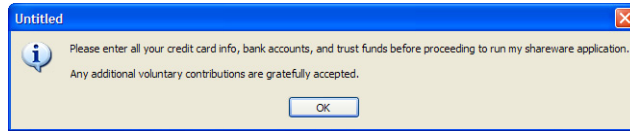
Dieses Beispiel zeigt einen Hinweis-Dialog.

MsgBox "Hello World."



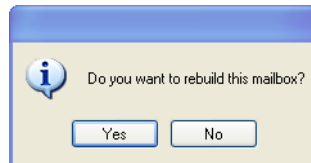
Dies zeigt einen mehrzeiligen Hinweis-Dialog. Der Text nach den beiden EndOfLine erscheint auf dem Macintosh in einem kleineren Zeichensatz. Beachten Sie, daß der Text in einer Zeile im Code-Editor eingegeben wird.

MsgBox "Please enter all your credit card info, bank accounts, and trust funds before proceeding to run my shareware application."+EndOfLine+EndOfLine+"Any additional voluntary contributions are gratefully accepted."



Dieses Beispiel zeigt einen Dialog mit „Ja“- und „Nein“-Knöpfen.

```
Dim n as Integer
n=MsgBox("Do you want to rebuild this mailbox",36)
If n=6 then
    //user pressed Yes
elseif n=7 then
    //user pressed No
end if
```



Siehe auch

EndOfLine Funktion, MessageDialog, MessageDialogButton Klasse.

MySQLDatabase Klasse

Wird zum Öffnen von MySQL-Datenbanken verwendet.

Eigenschaften

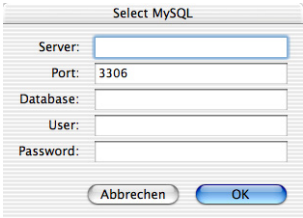
Name	Typ	Beschreibung
<i>Port</i>	<u>Integer</u>	Der Port, den die MySQL-Datenbank für die Verbindung verwenden wird. Der Default-Wert ist 3306.
<i>Timeout</i>	<u>Integer</u>	Der Timeout-Wert für die Verbindung (Sekunden). Der Default-Wert ist 15.

Methoden

Name	Parameter	Beschreibung
<i>GetAffectedRows</i>		Liefert als ein <u>Integer</u> die Anzahl von Datensätzen, die vom letzten SQLExecute-Befehl geändert wurden.
<i>GetInsertID</i>		Liefert als ein <u>Integer</u> den Wert des AUTOINCREMENT-Feldes für den zuletzt eingefügten Datensatz.
<i>LogPackets</i>	Location als <u>FolderItem</u>	Eine Debugging-Funktion, die alle Netzwerk-Pakete zwischen dem Treiber und dem Server in die durch <i>Location</i> festgelegte Datei protokolliert.

Hinweise

Um diese Klasse verwenden zu können, müssen Sie das MySQL-Datenbank-Plugin in Ihrem Plugins-Ordner installiert haben. Wenn das Plugin installiert ist, können Sie eine MySQL-Datenbank auch in der REALbasic-IDE über den Menüpunkt **Ablage/Datenquelle hinzufügen/Select MySQL Database** öffnen. Dieser Befehl zeigt eine Dialogbox an, die nach den Server-, Port-, Datenbank-, Anwendername- und Paßwort-Eigenschaften fragt.



Die Sammlung aller Datenbank-Plugins befindet sich auf der REALbasic-CD, Sie finden die aktuellsten Versionen aber auch immer auf der REAL Software Website, www.realsoftware.com.

Beispiel

Dieses Beispiel stellt eine Verbindung zu einer entfernten Datenbank her.

```
Dim db as MySQLDatabase
db=New MySQLDatabase
db.host="192.168.1.172"
db.port=3306
db.databaseName="MyOwnmySQLdatabase"
db.userName="Mary"
db.Password="Elton"
If db.Connect then
    //weiter mit Datenbank-Operationen
else
    MsgBox "Konnte keine Verbindung herstellen!"
end if
```

Siehe auch

[Database](#) Klasse.

NetworkInterface Objekt

Wird verwendet, um Informationen über eine Netzwerkkarte zu erhalten.

Eigenschaften

Name	Typ	Beschreibung
<i>IPAddress</i>	<u>String</u>	Die IP-Adresse des Computers.
<i>MacAddress</i>	<u>String</u>	Die MAC-Adresse der Netzwerkkarte.
<i>SubnetMask</i>	<u>String</u>	Die Subnet-Maske des Computers.

Hinweise

Benutzen Sie die GetNetworkInterface-Methode des [System](#)-Objekts, um ein **NetworkInterface**-Objekt zu instanzieren. REALbasic 5.5 (und neuer) unterstützen mehrere Netzwerkkarten unter Mac OS X, Windows und Linux. Damit können Sie Applikationen schreiben, die verschiedene Netzwerkkarten im Computer verwenden. Zum Beispiel eine Applikation, die sogenanntes „Tunneling“ durchführt, was mit vorherigen Versionen von REALbasic nicht möglich wäre. Um herauszufinden, welche Netzwerkkarten im Computer vorhanden sind, verwenden Sie die Methode GetNetworkInterface des [System](#)-Objekts. Sie können das erhaltene Interface-Objekt der NetworkInterface-Eigenschaft der [SocketCore](#)-Klasse zuweisen. Damit wird der Socket an diese Netzwerkkarte gebunden.

Beispiel

Das folgende Beispiel zeigt die IP-Adresse des Computers in einem [StaticText](#)-Feld an.

```
Dim ni as NetworkInterface
```

```
ni=System.GetNetworkInterface
StaticText1.Text=ni.IPAddress
```

Siehe auch

System-Objekt, SocketCore, TCPSocket-Klassen.

New Operator

Hiermit werden neue Instanzen eines Objekts zur Laufzeit erzeugt.

Syntax

Der **New**-Operator bietet verschiedene Syntaxen:

Ein Fenster-Objekt instanzieren:

Objektreferenz = New ObjektName

Teil	Typ	Beschreibung
<i>Objektreferenz</i>	beliebiger Objekttyp	Erforderlich; neue Variable vom Typ ObjektName.
<i>ObjektName</i>	Objektname	Erforderlich. Beliebiger Name eines Objekts im aktuellen Fenster.

Instanzieren aller anderen Typen von Objekten:

Objektreferenz = New ObjectKlasse

Teil	Typ	Beschreibung
<i>Objektreferenz</i>	beliebiger Objekttyp	Erforderlich; Neue Variable vom Typ Objektklasse.
<i>Objektklasse</i>	Bezeichnung.	Erforderlich; Bezeichnung der Objektklasse. Darf nicht mit aktueller Steuerelementklasse übereinstimmen.

Instanzieren aller anderen Typen von Objekten:

Dim ObjectReferenz As New ObjectKlasse

Teil	Typ	Beschreibung
<i>Objektreferenz</i>	beliebiger Objekttyp	Erforderlich; Neue Variable vom Typ Objektklasse.
<i>ObjectKlasse</i>	Bezeichnung	Erforderlich; Bezeichnung der Objektklasse. Darf nicht mit aktueller Steuerelementklasse übereinstimmen.

Hinweise

Der **New**-Operator kann als Zusatz zum Dim Befehl benutzt werden, wenn Sie ein neues Objekt erzeugen. Normalerweise wären dazu zwei Zeilen notwendig, die damit zusammengefaßt werden können. Der Dim-Befehl, der das Objekt erzeugt und der **New**-Operator, der das Objekt instanziiert.

Neue Instanzen von Steuerelementen oder Menueinträgen erzeugen

Der **New**-Operator kann nur neue Instanzen von Steuerelementen erzeugen, die schon als Steuerelemente im Fenster, oder als Menueinträge in einem Menü vorhanden sind. Der Objektname verhält sich dabei wie eine Schablone für die neue Instanz des Steuerelements oder Menueintrags. Der **New**-Operator funktioniert am besten, wenn das Steuerelement oder der Menueintrag in einem Array liegt – man kann sonst das alte und das neue Steuerelement nicht unterscheiden. Die Variable für den Objektverweis muß vom gleichen Typ wie das Objekt selbst sein.

Anlegen neuer Instanzen anderer Objekttypen

Der **New**-Operator kann verwendet werden, um eine neue Instanz eines Objekts einer beliebigen Objektart anzulegen (bitte beachten Sie den Hinweise zu den Steuerelementen). Der Objektverweis muß als passende Objektklasse angelegt werden.

Wenn die Visible-Eigenschaft auf True gesetzt ist, wird die neue Instanz in der laufenden Applikation sichtbar erscheinen.

Beispiele

Dieses Beispiel erzeugt eine neue Instanz eines PushButtons namens „PushButton1“ und setzt die Beschriftung auf „OK“.

```
Dim pb As PushButton
pb=New pushbutton1
pb.caption="OK"
```

Hier wird eine neue Instanz eines Window-Objekts namens „AboutBox“ erzeugt und angezeigt.

```
Dim w As aboutbox
w=New aboutbox
```

Erzeugen eines MenuItem mit der Bezeichnung „WindowItem1“.

```
Dim m As MenuItem
m=New WindowItem1
```

Eine neue Instanz einer MessageDialog-Box.

```
Dim d as New MessageDialog
```

Es ist äquivalent zu den folgenden zwei Zeilen:

```
Dim d as MessageDialog
d=New MessageDialog
```

Siehe auch

Nil, NewPicture-Funktionen.

NewAppleEvent Funktion

Legt ein neues Objekt der AppleEvent-Klasse an.

Syntax

Ergebnis=**NewAppleEvent**(eventClass, eventID, creatorCode)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>AppleEvent</u> Objekt	Das neue AppleEvent-Objekt.
<i>EventClass</i>	<u>String</u>	Der vierstellige Code für die <u>AppleEvent</u> -Klasse.
<i>EventID</i>	<u>String</u>	Die vierstellige ID des AppleEvents.
<i>creatorCode</i>	<u>String</u>	Der vierstellige Creator-Code der Anwendung, an die das AppleEvent geschickt werden wird.

Hinweis

Verwenden Sie diese Funktion, um ein AppleEvent anzulegen, der Daten an eine andere Anwendung schicken soll. EventClass und EventID definieren zusammen ein bestimmtes AppleEvent. EventClass arbeitet als Kategorie, um logisch zusammengehörige Events zu gruppieren. Obwohl es viele Standard-AppleEvents (und sogar einige Pflicht-Events) gibt, verwenden viele Anwendungen eigene AppleEvents, um bestimmte Aktionen auszuführen, die nur auf diese Anwendung zutreffen. Fragen Sie den Autor oder lesen Sie in der Dokumentation zur Anwendung nach, welche Informationen zu anwendungsspezifischen AppleEvents verfügbar sind. Wenn Sie einen Leerstring als Creator-Code übergeben, wird das Event an die eigene Applikation gesendet.

Wenn Sie das AppleEvent angelegt und die notwendigen Parameter mit Daten ausgefüllt haben, können Sie das AppleEvent mit der AppleEvent-Methode „Send“ an die Anwendung verschicken.

Beispiele

Bei der Beschreibung der AppleEvent-Klasse finden Sie ein Beispiel, wie man ein AppleEvent anlegt und verschickt.

Siehe auch

AppleEvent Klasse.

NewAppleEventTarget Funktion

Wird verwendet, um die Ziellapplikation zu bestimmen, die das Apple-Event erhalten wird.

Syntax

Ergebnis=**NewAppleEventTarget**(Name, Computer, Zone, PortType)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>AppleEventTarget</u>	Das Objekt, das das Apple Event Target repräsentiert
<i>Name</i>	<u>String</u>	Name der Ziellapplikation.

Teil	Typ	Beschreibung
<i>Computer</i>	<u>String</u>	Der Computer, auf dem die Applikation installiert ist.
<i>Zone</i>	<u>String</u>	AppleTalk-Zone.
<i>PortType</i>	<u>String</u>	Vierbuchstabiger Creator-Code, der die Zielapplikation festlegt.

Beispiel

Dieses Beispiel bestimmt als Ziel das Programm Mailsmith, das auf dem Rechner namens Daystar läuft.

```
Dim apptarget as AppleEventTarget
apptarget=NewAppleEventTarget("Mailsmith 1.1","Daystar","*", "BLT0")
```

Siehe auch

AppleEvent Klasse.

NewDragItem Funktion

Legt das Drag-Rechteck (basierend auf den übergebenen Koordinaten) fest, das erscheint, wenn der Anwender von diesem Steuerelement oder Fenster draggt. Liefert ein neues DragItem-Objekt.

Syntax

Ergebnis=**NewDragItem**(Left, Top, Width, Height)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>DragItem</u>	Objekt, das das gedraggte Objekt repräsentiert.
<i>Left</i>	<u>Integer</u>	Linke Koordinate des Drag-Rechtecks.
<i>Top</i>	<u>Integer</u>	Obere Koordinate des Drag-Rechtecks.
<i>Width</i>	<u>Integer</u>	Breite des Drag-Rechtecks.
<i>Height</i>	<u>Integer</u>	Höhe des Drag-Rechtecks.

Hinweise

Das **NewDragItem** wird in einem Event-Handler des Steuerelements verwendet, von dem der Anwender draggen wird. Der Mouse-Down-Event-Handler ist ein geeigneter Ort. Ihr Code verwendet **NewDragItem**, um das DragItem zu erzeugen und die Größe der Drag-Region festzulegen. Je nach Datentyp, der gedraggt wird (Sie können mehrere DragItems zulassen) verwenden Sie die Eigenschaften FolderItem, Picture, MacData oder Text des DragItems, um das gedraggte Objekt (oder die gedraggten Objekte) zu speichern.

Wenn Sie von einem RectControl aus draggen, können Sie ebenso die NewDragItem-Methode der RectControl-Klasse verwenden, um ein DragItem zu erzeugen.

Beispiele

Folgender Code imMouseDown-Event-Handler eines ImageWell erzeugt ein DragItem und erlaubt das Draggen:

```
Dim d as DragItem
d=NewDragItem(Me.left,Me.top,Me.width,Me.height)
d.picture=Me.image
d.Drag //Allow the drag
```

Folgender Code imOpen-Event eines anderen ImageWells erlaubt das Empfangen eines gedraggten Bildes:

```
Me.AcceptPictureDrop
```

Folgender DropObject-Event-Handler weist das Bild, das sich im DragItem befindet, der BackDrop-Eigenschaft des Steuerelements zu. Das führt dazu, daß es angezeigt wird.

```
Sub DropObject(obj as DragItem)
Me.image=obj.Picture
End Sub
```

Siehe auch

DragItem Klasse.

NewMemoryBlock Funktion

Liefert ein neues Objekt der [MemoryBlock](#)-Klasse in der angegebenen Größe zurück. Es wird verwendet, um auf jeder Plattform mit purem Speicher umzugehen und kann verwendet werden, um Daten über einen Socket oder einen seriellen Port zu übermitteln. Sie kann mit Declared-Funktionen und Pointer-Parametern als Einstiegspunkte von PowerPC-Shared-Libraries verwendet werden.

Syntax

Ergebnis=**NewMemoryBlock**(size)

Teil	Art	Beschreibung
<i>Ergebnis</i>	MemoryBlock	Der neueMemoryBlock in der Größe <i>size</i> .
<i>Size</i>	Integer	Die Größe des MemoryBlocks in Bytes.

Hinweise

Verwenden Sie die Funktion **NewMemoryBlock**, um einen leeren [MemoryBlock](#) anzulegen, der dazu dient, an den Einstiegspunkt einer PowerPC-Shared-Library Daten zu übergeben oder Daten von dort wieder zurückzuliefern.

Wenn der Applikation nicht mehr genug Speicher zur Verfügung steht, um den MemoryBlock anzulegen, wird Nil geliefert. Überprüfen Sie den Rückgabewert daher auf jeden Fall auf Nil, bevor Sie fortfahren.

Beispiel

Folgendes Beispiel erzeugt einen 384 Byte großen MemoryBlock, dessen Bytes von 0 bis 383 durchnummeriert sind:

```
Dim m as MemoryBlock
m=NewMemoryBlock(384)
If m=Nil then
    Beep
    MsgBox "Nicht genug Speicher zum Fortfahren"
else
    // normal weiter
end if
```

Siehe auch

[MemoryBlock](#) Klasse, [Nil](#) Operator.

NewPicture Funktion

Liefert ein Picture-Objekt in der angegebenen Größe und Farbtiefe zurück. Diese Funktion wird zum Off-screen-Zeichnen und zur direkten Manipulation von Pixeln verwendet, indem es die Eigenschaft RGBSurface der [Picture](#)-Klasse verwendet.

Syntax

Ergebnis=**NewPicture**(Breite, Hoehe, Tiefe)

Teil	Art	Beschreibung
<i>Ergebnis</i>	Picture	Ein neues Picture-Objekt in der angegebenen Größe und Farbtiefe.
<i>Breite</i>	Integer	Die Breite des neuen Picture-Objektes (in Pixeln)
<i>Hoehe</i>	Integer	Die Höhe des neuen Picture-Objektes (in Pixeln)
<i>Tiefe</i>	Integer	Die Farbtiefe des neuen Picture-Objektes. Mögliche Werte sind hierbei z.B. 0, 1, 2, 4, 8, 16 oder 32. Eine Tiefe von Null legt fest, daß das Bild ohne Pixelbelegung erzeugt wird; diese Option ist für das Erzeugen von Vektorgrafiken, z.B. mit der Unterklasse der Object2D-Klasse.

Hinweise

Verwenden Sie die Funktion **NewPicture**, um ein leeres Picture-Objekt anzulegen. Dieses Objekt können Sie für Grafiken verwenden, die Sie nicht sofort auf den Bildschirm zeichnen wollen. Diese Arbeitsweise wird allgemein als „Off-Screen-Zeichnen“ bezeichnet. Bilder, die mit NewPicture mit einer Farbtiefe von 16 oder 32 erzeugt wurden, können auf Pixelebene mit der RGBSurface-Eigenschaft der [Picture](#)-Klasse manipuliert werden. Dies ist schneller als eine gleiche Operation mit den Methoden der [Graphics](#)-Klasse.

Wenn der Applikation nicht mehr genug Speicher zur Verfügung steht, um ein neues Bild zu erzeugen, wird Nil zurückgeliefert. Um das festzustellen, müssen Sie direkt nach dem Aufruf von **NewPicture** das Picture-Objekt auf Nil überprüfen. Dies wird im Beispiel im folgenden Abschnitt gezeigt.

Alternativ dazu können Sie anstelle von **NewPicture** ein neues Bild mit dem **New**-Operator erzeugen. Die Parameter sind in beiden Fällen die gleichen (Breite, Höhe, Tiefe), wenn Sie jedoch **New** verwenden, tritt eine OutOfMemoryException auf, wenn nicht genug Speicher zum Erzeugen frei ist.

Beispiel

Das folgende Beispiel erzeugt eine Animation in einem Offscreen-Bereich und stellt sie dann in einem Canvas-Steuerelement dar, wenn die Animation komplett ist. Dies sorgt für eine nahezu flimmerfreie Animation.

```
Dim x,y as integer
Dim buffer as Picture
'Erzeugt einen Puffer, um die Grafik aufzunehmen
'so lange sie zusammengebaut wird. Verwenden Sie
'die Width,- Height-, und Depth-Eigenschaft des
'Canvas auf dem Hauptschirm
Buffer=NewPicture(cvsImage.Graphics.Width,cvsImage.Graphics.Height,Screen(0).Depth)
If Buffer<>Nil then
    'Setzt den Startwert von x und einen zufälligen
    'für y für die Raumschiffposition
    x=0
    y=Rnd*(cvsImage.Graphics.width-50)

    'Schleife, bis die x-Position des Schiffs größer
    'als die Breite des Canvas ist.

    Do
        'Zeichne den Hintergrund in den Puffer
        'buffer.Graphics.DrawPicture Space,0,0
        'Zeichne das Raumschiff in den Puffer
        'buffer.Graphics.DrawPicture Ship,x,y
        'Erhöhe den Wert der Schiffposition
        'Hinweis: Es wird das Inkrement erhöht, da diese
        ' Methode etwas länger dauert.
        x=x+2
        'Das Canvas mit dem Pufferbild updaten
        cvsImage.Graphics.DrawPicture buffer,0,0
        loop until x>cvsImage.Graphics.Width
    else 'buffer ist Nil
        MsgBox "Nicht genug Speicher zum Fortfahren"
    end if
```

Folgender Code erzeugt die Animation direkt im Canvas-Steuerelement, was zu einem Flackern beim Bewegen des Objekts führt:

```
Dim x,y as integer
'Setzt den Startwert von x und einen zufälligen
'für y für die Raumschiffposition
x=0
y=Rnd*(cvsImage.Graphics.width-50)

'Schleife, bis die x-Position des Schiffs größer
'als die Breite des Canvas ist.

Do
    'Zeichne den Hintergrund
    cvsImage.Graphics.DrawPicture Space,0,0
    'Zeichne das Raumschiff
    cvsImage.Graphics.DrawPicture Ship,x,y
    'Erhöhe den Wert der Schiffposition
    x=x+1
    loop until x>cvsImage.Graphics.Width
```

Verwenden Sie möglichst immer **NewPicture**, um flüssige Animationen zu erzeugen.

Siehe auch

Picture, Graphics Klassen; Canvas Steuerelement; Nil Operator.

NewREALdatabase Funktion

Erzeugt eine neue REAL-Datenbank. Diese Funktion ist obsolet. Bitte verwenden Sie stattdessen die Methode CreateDatabaseFile der [REALdatabase](#)-Klasse.

Siehe auch

[REALdatabase](#)-Klasse.

Nil Schlüsselwort

Wird verwendet, um festzustellen, ob ein Objekt Nil (nicht definiert) ist.

Syntax

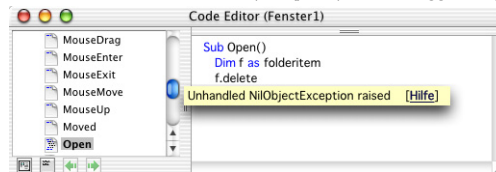
expression=**Nil**

Hinweis

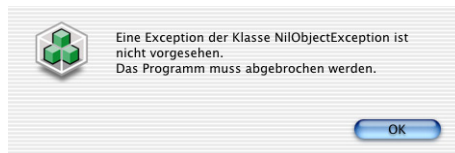
Nil bedeutet „kein Wert“. Einige Funktionen liefern unter bestimmten Voraussetzungen **Nil** zurück. So gibt [GetOpenFolderItem](#) **Nil** zurück, wenn der Anwender in der Datei-Öffnen-Dialogbox den Abbruch-Button angeklickt hat.

Funktionen wie TrashFolder und ControlPanelsFolder liefern Nil zurück, wenn sie unter Windows aufgerufen werden.

Wenn Sie versuchen, auf ein Objekt zuzugreifen, das den Wert Nil hat, wird ein [NilObjectException](#) Fehler erzeugt. Wenn Sie erlauben, daß der Fehler auftritt, wird die Ausführung gestoppt. Wenn Sie das Programm in der IDE testen, können Sie unter dem Menüpunkt „Debug/Break bei Exceptions“ einstellen, ob bei einer Ausnahme (Exception) der Debugger aufgerufen werden soll.



Wenn der Code in einer erzeugten Anwendung ausgeführt wird, wird dem Anwender eine Fehlermeldung gezeigt, bevor das Programm abgebrochen wird.



Um dies zu verhindern, sollten Sie immer auf Nil-Werte testen, entweder durch einen [If](#)-Befehl oder durch Einbinden eines [Exception](#) Blocks am Ende der Methode.

Beispiel

Dieses Beispiel verwendet die **Nil**-Schlüsselwort um festzustellen, ob der Anwender beim Aufruf der Funktion [GetOpenFolderItem](#) den Abbruch-Button angeklickt hat.

```
Dim f as FolderItem
f=GetOpenFolderItem("image/x-pict")
If f <> Nil Then
    //Der Anwender hat OK angeklickt, also hier weitermachen
End If
```

Siehe auch

[RuntimeException](#) Klasse, [Exception](#) Block, [NilObjectException](#) Fehler.

NilObjectException Laufzeitfehler

Tritt auf, wenn Code versucht, auf ein Objekt zuzugreifen, das nicht existiert.

Übergeordnete Klasse

RuntimeException

Hinweise

`NilObjectExceptions` treten auf, wenn Sie eine Eigenschaft oder Variable verwenden, um auf ein Objekt zuzugreifen, das nicht existiert. Wenn der `GetFolderItem`-Funktion z.B. ein ungültiger Pfadname übergeben wird, gibt sie `Nil` zurück. Bestimmte Funktionen, die dazu vorgesehen sind, `FolderItems` auf Verzeichnisse zurückzuliefern, die auf dem Macintosh existieren, geben `Nil` zurück, wenn Sie unter Windows laufen. `TrashFolder` und `ControlPanelsFolder` liefern unter Windows z.B. `Nil`. Wenn Ihr Code versucht, auf eine Eigenschaft eines `NilFolderItem` zuzugreifen oder hineinzuschreiben, tritt ein **`NilObjectException`**-Fehler auf.

Eine Funktion liefert ebenfalls `Nil`, wenn nicht genug Speicher frei ist, um ein Objekt zu laden. Wenn Sie z.B. versuchen, ein sehr großes Bild in den Speicher zu laden, Ihr Programm aber nicht mehr genug Speicher frei hat, um das Bild zu laden, würde die Funktion, die zum Laden des Bildes verwendet wird, `Nil` zurückliefern.

Direkt nachdem Sie eine Funktion aufgerufen haben, die ein Objekt zurückliefert (wie z.B. `GetFolderItem`), sollten Sie überprüfen, ob das Objekt `Nil` ist oder einen `Exception` Block an das Ende der Methode anfügen, das eine `NilObjectException` behandelt. Jedes Funktionsresultat, das kein `String`, `Integer`, `Single`, `Double`, `Color` oder `Boolean` ist, ist ein Objekt und sollte überprüft werden.

Beispiele

Dieses Beispiel versucht eine PICT-Datei mittels `GetFolderItem` zu öffnen und zeigt diese in einem `ImageWell` an. Wenn der an `GetFolderItem` übergebene Pfadname ungültig ist, wird `Nil` zurückgeliefert. Die **`NilObjectException`** wird im `Exception` Block behandelt, welcher sich nach der letzten „regulären“ Zeile der Methode befindet.

Wenn der Pfadname gültig ist, das Objekt aber nicht existiert, liefert `GetFolderItem` ein gültiges `FolderItem` zurück, dessen `AbsolutePath`-Eigenschaft mit dem Pfadnamen identisch ist. Sie sollten die `Exists`-Eigenschaft des `FolderItem` testen, um festzustellen, ob die Datei existiert, bevor Sie irgendetwas mit dem `FolderItem` machen.

Dieser Code prüft demnach zwei verschiedene Probleme – ungültiger Pfadname und gültiger Pfad auf eine nicht existierende Datei.

```
Sub Action ()
    Dim f as FolderItem
    f = GetFolderItem("HD:Images:Logo")
    If Not f.exists then
        Beep
        MsgBox "Die Datei "+f.absolutePath+" existiert nicht!"
    else //Datei existiert
        If f.MacType="PICT" then
            ImageWell1.image=f.OpenAsPicture
        end if
    end if
    Exception err as NilObjectException
        MsgBox "Ungültiger Pfadname!"
End Sub
```

Folgendes Beispiel erzeugt ein neues, leeres Bild. Wenn nicht genug Speicher zum Erzeugen des Bildes übrig ist, wird `Nil` von der Funktion `NewPicture` zurückgeliefert. Dieser Beispiel-Code überprüft das und informiert den Anwender:

```
Dim p as Picture
p = NewPicture(100,100,32)
If p = Nil then
    Beep
    MsgBox "Es ist nicht genug Speicher vorhanden, um das Bild zu erzeugen."
else
    //Mache irgendetwas mit dem Bild hier
end if
```

Siehe auch

`RuntimeException` Klasse; `OutOfBoundsException`, `IllegalCastException`, `StackOverflowException`, `TypeMismatchException` Fehler; `Function`, `Raise` Befehl; `Nil` Schlüsselwort, `Exception` Block.

Not Operator

Wird verwendet, um eine logische Negierung eines boole'schen Ausdruck durchzuführen.

Syntax

Ergebnis = **Not** expression

Teil	Beschreibung
<i>Ergebnis</i>	Ein <u>boole'scher</u> Wert.
<i>expression</i>	Ein gültiger <u>boole'scher</u> Ausdruck

Hinweis

Not liefert das Gegenteil dessen, was der boole'sche Wert *expression* ergibt. Ergibt *expression* True, so liefert **Not** False. Ergibt *expression* False, so liefert **Not** True.

Siehe auch

And, Or Operator.

NotePlayer Steuerelement



Spielt über die Systemerweiterung „QuickTime Musikinstrumente“ einen Ton.

Übergeordnete Klasse

Control

Da dies ein Steuerelement (Control) ist, informieren Sie sich bei der Control-Klasse über die anderen Eigenschaften und Events, die alle Control-Objekte gemeinsam haben.

Eigenschaften

Name	Art	Beschreibung
<i>Index</i>	<u>Integer</u>	Die Nummer des Controls, wenn es Teil eines Control-Arrays ist.
<i>Instrument</i>	<u>Integer</u>	Die Nummer des Musikinstruments, das zum Spielen der Note verwendet werden soll (siehe Liste unten).
<i>Left</i>	<u>Integer</u>	Der linke Abstand des Controls in lokalen Koordinaten (relativ zum Fenster).
<i>Name</i>	<u>String</u>	Der Name des Objekts.
<i>Top</i>	<u>Integer</u>	Der obere Abstand des Controls in lokalen Koordinaten (relativ zum Fenster).

Methoden

Name	Parameter	Beschreibung
<i>PlayNote</i>	Tonhöhe (Pitch) als <u>Integer</u> , Velocity als <u>Integer</u>	Spielt den über Tonhöhe und Velocity definierten Ton.

Hinweis

Beim Aufruf der Methode PlayNote repräsentiert Pitch die Tonhöhe, wobei ein mittleres C dem Wert 60 entspricht. Eine Änderung des Wertes um 1 erhöht bzw. senkt die Tonhöhe um einen halben Schritt.

Sie können den Wert von Pitch auch als eine Zahl zwischen 256 und 32767 ausdrücken, also $\text{pitch} * 256$. Verwenden Sie dies, um halbe Schritte festzulegen. Wenn Sie z.B. einen Pitch von 60.5 wollen, legen Sie $60.5 * 256 = 15488$ als Wert fest.

Der Parameter „Velocity“ gibt auf einer Skala von 0 bis 127 an, wie stark die Taste angeschlagen wurde. 127 ist hierbei ein besonders harter Anschlag, bei 0 ist die Taste nicht mehr angeschlagen. Wenn Sie die Methode PlayNote aufrufen, wird der Ton solange spielen, bis Sie ihn mit einer Velocity von 0 erneut anspielen.

Der NotePlayer ist unter Linux nicht verfügbar.

Instrumente

Instrument	Wert	Instrument	Wert
<i>Acoustic Grand Piano</i>	1	<i>Soprano Sax</i>	65
<i>Bright Acoustic Piano</i>	2	<i>Alto Sax</i>	66

Instrument	Wert	Instrument	Wert
<i>Electric Grand Piano</i>	3	<i>Tenor Sax</i>	67
<i>Honky-tonk Piano</i>	4	<i>Baritone Sax</i>	68
<i>Rhodes Piano</i>	5	<i>Oboe</i>	69
<i>Chorused Piano</i>	6	<i>English Horn</i>	70
<i>Harpsichord</i>	7	<i>Bassoon</i>	71
<i>Clavinet</i>	8	<i>Clarinet</i>	72
<i>Celesta</i>	9	<i>Piccolo</i>	73
<i>Glockenspiel</i>	10	<i>Flute</i>	74
<i>Music Box</i>	11	<i>Recorder</i>	75
<i>Vibraphone</i>	12	<i>Pan Flute</i>	76
<i>Marimba</i>	13	<i>Bottle blow</i>	77
<i>Xylophone</i>	14	<i>Shakubachi</i>	78
<i>Tubular bells</i>	15	<i>Whistle</i>	79
<i>Dulcimer</i>	16	<i>Ocarina</i>	80
<i>Draw Organ</i>	17	<i>Square Lead</i>	81
<i>Percussive Organ</i>	18	<i>Saw Lead</i>	82
<i>Rock Organ</i>	19	<i>Calliope</i>	83
<i>Church Organ</i>	20	<i>Chiffer</i>	84
<i>Reed Organ</i>	21	<i>Synth Lead 5</i>	85
<i>Accordion</i>	22	<i>Synth Lead 6</i>	86
<i>Harmonica</i>	23	<i>Synth Lead 7</i>	87
<i>Tango Accordion</i>	24	<i>Synth Lead 8</i>	88
<i>Acoustic Nylon Guitar</i>	25	<i>Synth Pad 1</i>	89
<i>Acoustic Steel Guitar</i>	26	<i>Synth Pad 2</i>	90
<i>Electric Jazz Guitar</i>	27	<i>Synth Pad 3</i>	91
<i>Electric clean Guitar</i>	28	<i>Synth Pad 4</i>	92
<i>Electric Guitar muted</i>	29	<i>Synth Pad 5</i>	93
<i>Overdriven Guitar</i>	30	<i>Synth Pad 6</i>	94
<i>Distortion Guitar</i>	31	<i>Synth Pad 7</i>	95
<i>Guitar Harmonics</i>	32	<i>Synth Pad 8</i>	96
<i>Wood Bass</i>	33	<i>Ice Rain</i>	97
<i>Electric Bass Fingered</i>	34	<i>Soundtracks</i>	98
<i>Electric Bass Picked</i>	35	<i>Crystal</i>	99
<i>Fretless Bass</i>	36	<i>Atmosphere</i>	100
<i>Slap Bass 1</i>	37	<i>Bright</i>	101
<i>Slap Bass 2</i>	38	<i>Goblin</i>	102
<i>Synth Bass 1</i>	39	<i>Echoes</i>	103
<i>Synth Bass 2</i>	40	<i>Space</i>	104
<i>Violin</i>	41	<i>Sitar</i>	105
<i>Viola</i>	42	<i>Banjo</i>	106
<i>Cello</i>	43	<i>Shamisen</i>	107
<i>Contrabass</i>	44	<i>Koto</i>	108

Instrument	Wert	Instrument	Wert
<i>Tremolo Strings</i>	45	<i>Kalimba</i>	109
<i>Pizzicato Strings</i>	46	<i>Bagpipe</i>	110
<i>Orchestral Harp</i>	47	<i>Fiddle</i>	111
<i>Timpani</i>	48	<i>Sbanai</i>	112
<i>Acoustic String Ensemble 1</i>	49	<i>Tinkle bell</i>	113
<i>Acoustic String Ensemble 2</i>	50	<i>Agogo</i>	114
<i>Synth Strings 1</i>	51	<i>Steel Drums</i>	115
<i>Synth Strings 2</i>	52	<i>Woodblock</i>	116
<i>Aab Choir</i>	53	<i>Taiko Drum</i>	117
<i>Oob Choir</i>	54	<i>Melodic Tom</i>	118
<i>Synvox</i>	55	<i>Synth Tom</i>	119
<i>Orchestra Hit</i>	56	<i>Reverse Cymbal</i>	120
<i>Trumpet</i>	57	<i>Guitar Fret Noise</i>	121
<i>Trombone</i>	58	<i>Breath Noise</i>	122
<i>Tuba</i>	59	<i>Seashore</i>	123
<i>Muted Trumpet</i>	60	<i>Bird Tiveet</i>	124
<i>French Horn</i>	61	<i>Telephone Ring</i>	125
<i>Brass Section</i>	62	<i>Helicopter</i>	126
<i>Synth Brass 1</i>	63	<i>Applause</i>	127
<i>Synth Brass 2</i>	64	<i>Gunsbot Table</i>	128
		<i>Drum Kit</i>	16385

Beispiele

Folgendes spielt eine Note auf dem „Church Organ“ (Kirchenorgel) Instrument.

```
NotePlayer1.Instrument=20
NotePlayer1.PlayNote(60,60)
```

Folgende Code-Zeile stoppt die Note:

```
NotePlayer1.PlayNote(60,0)
```

Siehe auch

[Control](#) Klasse.

NthField Funktion

Liefert ein Feld aus einer Datenzeile. Das erste Feld tragt die Nummer 1.

Syntax

Ergebnis=NthField(quelle, separator, feldnummer)

ODER

Ergebnis=StringVariable.NthField(separator, feldnummer)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der gewunschte Wert des Feldes.
<i>quelle</i>	<u>String</u>	Die Datenfelder.
<i>separator</i>	<u>String</u>	Das Zeichen, das die Datenfelder voneinander trennt.
<i>feldnummer</i>	<u>Integer</u>	Die Spaltennummer des gewunschten Datenfeldes.
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **NthField** liefert den Wert aus *quelle*, der vor dem *feldnummer*-ten Auftreten des Trennungszeichens *separator* in *quelle* steht.

Wenn *feldnummer* außerhalb des Bereichs liegt, wird ein leerer String zurückgeliefert. NthField ist nicht case-sensitiv.

Beispiele

Dieses Beispiel liefert „Schmidt“ zurück.

```
Dim field As String
field=NthField("Frank*Schmidt*11.02.69*512 555 4323*Männlich", "*", 2)
```

Verwenden Sie folgende Syntax:

```
Dim s,field As String
s="Frank*Schmidt*11.02.69*512 555 4323*Männlich"
field=s.NthField("*",2)
MsgBox field
```

Schauen Sie auch beim Beispiel nach, wie man ein PopupMenu erzeugt.

Siehe auch

TextInputStream Beispiel, CountFields Funktion.

Object Klasse

Object ist die Basisklasse aller anderen Klassen. Ein Objekt kann einer Variable vom Typ Object zugewiesen werden. Die Object-Klasse hat keine Eigenschaften, Methoden oder Events.

Object2D Klasse

Die Basis-Klasse eines jeden Objekts, das in einer Vektorgrafik-Umgebung gezeichnet werden kann.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Border</i>	<u>Double</u>	Durchsichtigkeit der Randes, von 0 (transparent) bis zu 100 (undurchsichtig). Voreinstellung ist 0.
<i>BorderColor</i>	<u>Color</u>	Farbe des Randes.
<i>BorderWidth</i>	<u>Double</u>	Breite des Randes in Punkten. Voreinstellung ist 1.
<i>Fill</i>	<u>Double</u>	Durchsichtigkeit des Inneren, von 0 (transparent) bis 100 (undurchsichtig). Voreinstellung ist 100.
<i>FillColor</i>	<u>Color</u>	Farbe des Inneren der Figur.
<i>Rotation</i>	<u>Double</u>	Rotation im Uhrzeigersinn, im Bogenmaß, um den Punkt X, Y.
<i>Scale</i>	<u>Double</u>	Skalierungs-Faktor relativ zur Originalgröße des Objekts.
<i>X</i>	<u>Double</u>	Horizontale Position des Zentrums oder des Hauptankerpunktes.
<i>Y</i>	<u>Double</u>	Vertikale Position (von oben nach unten) des Zentrums oder des Hauptankerpunktes.

Hinweise

Object2D ist die Basis-Klasse für eine Gruppe von Unterklassen, die es Ihnen erlaubt, Vektorgrafiken (im Gegensatz zu Pixelgrafiken) zu erzeugen. Die Unterklassen sind in folgender Tabelle aufgeführt:

Klasse	Beschreibung
<i>ArcShape</i>	Zeichnet einen Kreisbogen.

Klasse	Beschreibung
<i>CurveShape</i>	Zeichnet eine gerade Linie oder Kurve unter Verwendung eines oder mehrerer „Kontrollpunkte“.
<i>FigureShape</i>	Zeichnet Polygone, die (optional) gebogene Seiten haben können.
<i>OvalShape</i>	Zeichnet Kreise und Ovale.
<i>PixmapShape</i>	Importiert eine Bitmap-Grafik in das Bild.
<i>RectShape</i>	Zeichnet ein Quadrat oder Rechteck.
<i>RoundRectShape</i>	Zeichnet ein Quadrat oder Rechteck mit abgerundeten Ecken (als Unterklasse von <u>RectShape</u>).
<i>StringShape</i>	Zeichnet Text-Strings in einem bestimmten Font, mit einer bestimmten Fontgröße und Stil.

Diese Basis-Figuren können mit der Group2D-Klasse in einer Hierarchie organisiert werden.

Die Picture-Klasse besitzt eine neue Eigenschaft Objects, die es erlaubt, in ein Bild eine Vektorgrafik-Zeichnung (ein Group2D-Objekt) einzubinden und die Graphics-Klasse besitzt eine neue Methode, DrawObject, die es erlaubt, ein Group2D-Objekt innerhalb des Graphics-Objektes zu zeichnen. Außerdem besitzt die SaveAsPicture-Method der FolderItem-Klasse nun einen optionalen Parameter, der es erlaubt, ein Vektorgrafik-Bild in einem Format zu speichern, bei dem die Vektor-Informationen erhalten bleiben. Die OpenAsVectorPicture-Methode der FolderItem-Klasse versucht, ein vorhandenes Bild als eine Vektorgrafik zu öffnen und wandelt den Inhalt des Bildes in Object2D-Zeichenobjekte um.

Beispiele

Siehe Beispiele bei ArcShape, CurveShape, FigureShape, OvalShape, PixmapShape, RectShape, RoundRectShape und StringShape.

Siehe auch

ArcShape, CurveShape, FigureShape, FolderItem, Group2D, Graphics, OvalShape, Picture, PixmapShape, RectShape, RoundRectShape, StringShape Klassen.

Object3D Klasse

Der Basistyp für 3D-Objekte.

Eigenschaften

Name	Typ	Beschreibung
<i>Bounds</i>	<u>Bounds3D</u>	Speichert den abgrenzenden Raum für das Object3D . Der Defaultwert ist <u>Nil</u> . Wenn Sie hier ein <u>Bounds3D</u> zuweisen, wird es automatisch aktualisiert, wenn das Objekt gezeichnet wird oder wenn Sie <u>UpdateBounds</u> aufrufen.
<i>NullShader</i>	<u>Boolean</u>	Falls dies auf <u>True</u> gesetzt wird, wird das Objekt ohne Rücksicht auf externe Beleuchtung gezeichnet, wie wenn Sie <u>AmbientLight=100</u> und <u>FloodLight=0</u> gesetzt hätten. Das ist bei Objekten nützlich, die aussehen sollen, als würden Sie von Innen leuchten, wie bei Lampen und Explosionen oder bei Hintergründen von Sprites, bei denen die Beleuchtung als Teil der Textur vorberechnet ist.
<i>Orientation</i>	<u>Quaternion</u>	Winkel des Objekts im 3D-Raum. Kann auch durch die <u>Pitch</u> -, <u>Roll</u> - und <u>Yaw</u> -Methoden kontrolliert werden.
<i>Position</i>	<u>Vector3D</u>	Die Position (das sind die x, y, z Koordinaten) des Objektes im 3D-Raum.
<i>RenderBack-Faces</i>	<u>Boolean</u>	Erlaubt Ihnen auf einer von Objekt-zu-Objekt-Basis zu entscheiden, ob Polygone, die von der Kamera weggerichtet sind, gezeichnet werden sollen. Die Voreinstellung ist <u>False</u> , das bedeutet, daß solche Polygone übersprungen werden, damit das Zeichnen schneller geht.
<i>Scale</i>	<u>Double</u>	Legt die Größe des Modells im 3D-Raum fest; ein <u>Scale</u> von 1.0 bedeutet, daß es mit der gleichen Größe wie bei der Erzeugung gezeichnet wird.
<i>Shape</i>	<u>Integer</u>	Wählt aus, welche von verschiedenen Formen derzeit gezeichnet wird. Die erste hinzugefügte Form hat die Nummer 0. Durch Setzen von <u>Shape=0</u> wählen Sie diese aus und so weiter.

Name	Typ	Beschreibung
<i>Visible</i>	<u>Boolean</u>	<u>True</u> , wenn sichtbar.

Methoden

Name	Parameter	Beschreibung
<i>AddShapeFromFile</i>	Datei als <u>FolderItem</u>	Lädt eine Modell-Datei und fügt sie in einem Schritt dem Objekt hinzu. Wenn die Datei nicht existiert oder keine gültigen Modelldaten enthält, wird keine Figur hinzugefügt.
<i>AddShapeFromHandle</i>	Daten als <u>Integer</u>	Fügt eine geometrisches Gebilde, die mittels Declare-Befehlen erzeugt wurde, hinzu.
<i>AddShapeFromString</i>	Daten als <u>String</u>	Lädt ein Modell aus 3DMF-Daten in einem String.
<i>AddShapePicture</i>	Bild als <u>Picture</u> , <i>scale</i> als <u>Double</u>	Erzeugt ein flaches, rechteckiges Modell aus einem REALbasic-Bild, dessen weiße Pixel transparent erscheinen. Ein <i>scale</i> von 1.0 bedeutet, daß jedes Pixel des Bildes einer Einheit im 3D-Raum entspricht.
<i>AddShapePictureWithMask</i>	Bild als <u>Picture</u> , Maske als <u>Picture</u> , <i>scale</i> als <u>Double</u>	Wie AddShapePicture, jedoch legt ein zweites Bild als Maske das durchsichtige Erscheinen fest.
<i>Clone</i>		Liefert ein Object3D. Klont das Objekt und gibt ein anderes Objekt vom Typ Object3D zurück. Der Klon hat die gleiche Geometrie wie das Original. Liefert ein Object3D .
<i>ComputeBounds</i>	Typ als <u>Integer</u>	Liefert ein Bounds3D begrenzenden Raum vom Typ <i>Typ</i> . Wird normalerweise zum Erzeugen des begrenzenden Raums eines Objektes verwendet, um diesen der Bound-Eigenschaft zuzuweisen.
<i>GetShapeHandle</i>	Index als <u>Integer</u>	Liefert (als Integer) das TQ3GeometryObject des angegebenen Gebildes.
<i>MoveForward</i>	Abstand als <u>Double</u>	Bewegt das Objekt in die Richtung, in die es zeigt.
<i>ObjectToWorldTransform</i>	pt als Vector3D	Liefert ein Vector3D. Erwartet einen Vektor, der einen Punkt relativ zu dem Objekt festlegt und liefert die Koordinaten dieses Punktes in der 3D-Welt unter Berücksichtigung der aktuellen Skalierung, Ausrichtung und Position des Objekts.
<i>Pitch</i>	Winkel als <u>Double</u>	Rotiert das Objekt in der vertikalen Ebene.
<i>Roll</i>	Winkel als <u>Double</u>	Rotiert das Objekt mit dem oder gegen den Uhrzeigersinn.
<i>UpdateBounds</i>		Aktualisiert das <u>Bounds3D</u> -Objekt, das dem Objekt über dessen Bounds-Eigenschaft angehängt wurde. Im Falle einer <u>Group3D</u> aktualisiert dies die Begrenzung jedes Objektes in dieser Gruppe. Dies wird mit der <u>RB3DSpace</u> .objects-Gruppe automatisch gemacht, wenn Sie <u>RB3DSpace</u> .Update aufrufen.
<i>WorldToObjectTransform</i>	pt als Vector3D	Erwartet einen Vektor, der einen Punkt in der 3D-Welt festlegt und liefert die Koordinaten dieses Punktes relativ zu dem Objekt, unter Berücksichtigung der aktuellen Skalierung, Ausrichtung und Position des Objekts. Liefert ein Vector3D.
<i>Yaw</i>	Winkel als <u>Double</u>	Rotiert das Objekt in der horizontalen Ebene.

Hinweise

Dies ist der dreidimensionale Basis-Objekttyp. Ein Object3D besitzt eine Position, eine Ausrichtung und es kann mehrere geometrische Modelle besitzen. Verwenden Sie AddShapeFromString, um aus einem String mit 3DMF-Daten ein Modell zu laden oder verwenden Sie AddShapePicture, um ein flaches, rechteckiges Modell aus einem REALbasic-Bild zu erzeugen, dessen weiße Pixel transparent erscheinen. Der *scale*-Parameter legt die Größe des Modells im 3D-Raum fest; wenn *scale* 1 ist, entspricht jedes Pixel des Bildes einer Einheit im 3D-Raum. Sie können seine Position und Ausrichtung direkt durch Modifikation seiner Eigenschaften ändern oder Sie können die Methoden Roll, Pitch und Yaw zum Rotieren des Objekts verwenden und die MoveForward-Methode, um es in die Richtung, in die es weist, zu bewegen (anfangs ist dies die +Z-Richtung).

Sie können die Skalierung durch Modifizieren der *Scale*-Eigenschaft ändern; der Defaultwert ist 1.0. Wenn Sie mehr als eine Kopie eines Objektes im Spiel benötigen (was ja meist der Fall ist), können Sie die Clone-Methode verwenden, die ein neues Objekt liefert, das die gleiche Geometrie wie das Original teilt. Da sie sich die Geometrie teilen, ist dies viel effektiver als wenn man separate Duplikate dieses Modells laden würde.

Siehe auch

[RB3DSpace](#) Steuerelement, [Vector3D](#), [Group3D](#), [Quaternion](#), [Bounds3D](#), [Light3D](#) Klasse.

Oct Funktion

Liefert die oktale Schreibweise einer Zahl als [String](#) zurück.

Syntax

Ergebnis=**Oct**(Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	String	Der umgewandelte Wert.
<i>Wert</i>	Integer	Der umzuwandelnde Wert.

Hinweise

Ist *Wert* keine ganze Zahl, so werden die Nachkommastellen abgeschnitten.

Notiz zur Kompatibilität mit Visual Basic: Visual Basic rundet den Wert zur nächsten ganzen Zahl. Möglicherweise wird das Verhalten der Oct-Funktion in einer zukünftigen Version daraufhin angepaßt.

Sie können Binär-, Hexadezimal- oder Oktal-Zahlen festlegen, indem Sie der Zahl den &-Operator und den Buchstaben für die Zahlen-Basis voranstellen. Der Buchstabe „b“ legt die Zahl als Binär-, „h“ als Hexadezimal-, „o“ als Oktal-Zahl fest.

Beispiele

Hier sind einige Beispiele verschiedener Zahlen, die in die oktale Schreibweise umgewandelt wurden:

```
Dim OctVersion As String
OctVersion=Oct(5) //liefert "5"
OctVersion=Oct(75) //liefert "113"
OctVersion=Oct(256) //liefert "400"
```

Siehe auch

[&](#) Operator, [Bin](#) und [Hex](#) Funktionen.

ODBCDatabase Klasse

Wird zum Öffnen von ODBC-Datenbanken verwendet. Verwenden Sie die **ODBCDatabase**-Klasse anstelle von [OpenODBCDatabase](#).

Übergeordnete Klasse

[Database](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Attribute</i>	Integer	Optionales Attribut. Siehe Dokumentation von ODBC bei SQLSetConnectAttr. Wenn Sie <i>Attribute</i> nicht verwenden wollen, übergeben Sie Null.
<i>AttributeString</i>	String	Optionales Attribut. Siehe Dokumentation von ODBC bei SQLSetConnectAttr. Wenn Sie <i>Attribute</i> nicht verwenden wollen, übergeben Sie einen leeren String.
<i>DataSource</i>	String	Der Connection-String, der verwendet wird, um eine Verbindung zu einer ODBC-Datenbank herzustellen.

Hinweise

Um diese Klasse verwenden zu können, müssen Sie das passende Datenbank-Plugin im Plugins-Ordner installiert haben.

Die REALbasic-CD enthält alle Datenbanken-Plugins und Sie finden die aktuellen Versionen auch immer auf der REAL Software Web-Site auf www.realsoftware.com.

Die **ODBCDatabase**-Klasse benötigt zusätzlich einen Treiber eines Dritt-Herstellers. REALbasic unterstützt die Treiber von Open-Link (www.openlinksw.com) und Merant/Metro Technologies ODBC Manager. OpenLink unterstützt Mac OS 8/9 und Mac OS X, während Merant momentan nur Mac OS X unterstützt. Ein Treiber-Manager muß ebenso installiert werden wie der Datenbanktreiber selbst. Hinweise auf Updates dieser Treiber finden Sie auf den Internetseiten von REAL Software, unter <http://www.realsoftware.com>.

Beispiel

In diesem Beispiel wird ein Öffnen-Dialog angezeigt, in dem Sie eine ODBC-Datenbank auswählen können.

```
Dim db as ODBCDatabase
db=New ODBCDatabase
db.DataSource=""
If db.Connect then
    //weiter mit Datenbank-Operationen
else
    MsgBox "Die Verbindung konnte nicht hergestellt werden."
end if
```

Siehe auch

[Database](#), [RecordSet](#), [DatabaseField](#), [DatabaseRecord](#)-Klassen.

Office Klasse

Die Klasse, die die Befehle beinhaltet, die Sie für die Office Automation benötigen.

Übergeordnete Klasse

[Object](#)

Hinweise

Office Automatisierung ist unter Mac OS „Classic“ nicht verfügbar.

Da unter Mac OS X die OLE Bibliotheken nicht systemweit zugänglich sind, müssen Ihre Applikationen nach dem Compilieren in das Verzeichnis **/Applications/Microsoft Office X/Office/** abgelegt werden. Wenn Sie die Applikationen debuggen wollen, müssen Sie die REALbasic IDE ebenfalls dorthin kopieren. Wir arbeiten daran, daß Ihre Applikationen auch außerhalb des Office-Verzeichnisses arbeiten können.

Office Automatisierung in REALbasic vs. VB

Wir gehen hier von der Applikations-Klasse aus. Wenn Sie VBA-Code in Excel, PowerPoint oder Word schreiben, existiert implizit eine Instanz der Applikation-Klasse. Zum Beispiel:

```
Dim pres as Presentation
Dim slide1 as Slide

Set pres = Presentations.Add
' Das oben ist das gleiche wie:
' Set pres = Application.Presentations.Add

Set slide1 = pres.Slides.Add(1, ppLayoutText)
```

In PowerPoint funktioniert dieser Code ohne Probleme. In Word oder Excel würde dies aber Fehler erzeugen, da diese nicht wissen, was ein Presentation-Objekt ist. Hier ist der REALbasic-Code:

```
Dim PowerPoint as new PowerPointApplication
Dim pres as OfficeObject
Dim slide1 as OfficeObject

pres = PowerPoint.Presentations.Add
slide1 = pres.Slides.Add(1, Office.ppLayoutText)
```

Es gibt hier tatsächlich nur zwei Unterschiede. Der erste ist, daß alle Objekte nun vom Typ **OfficeObject** sind. Sprich, es ist egal, ob es sich um eine Präsentation oder ein Slide handelt, Sie verwenden für diese Objekte den Typ **OfficeObject**. Alle konstanten Werte sind in den Office-Modulen enthalten.

Parameter über Namen angeben

REALbasic unterstützt die Parameterübergabe über Namen nicht, aber Sie können dies mit ein wenig Arbeit dennoch erreichen. Dafür müssen Sie vorher verstehen, wie man mit OLE-Objekten arbeitet. Hier ist ein Beispiel, das Sie nachbauen können. Wir erzeugen nun ein Suchen-und-Ersetzen-Makro in Word und übersetzen dieses in REALbasic.

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Suche dies"
    .Replacement.Text = "ersetze mit"
    .Wrap = wdFindContinue
    .Format = false
    .MatchCase = false
    .MatchWholeWord = false
    .MatchWildcards = false
    .MatchSoundsLike = false
    .MatchAllWordForms = false
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

So sieht es in REALbasic aus.

```
Dim word as New WordApplication
Dim find as OfficeObject

find = word.Selection.Find

find.ClearFormatting
find.Replacement.ClearFormatting
find.text = "Suche dies"
find.Replacement.Text = "ersetze mit"
find.Wrap = Office.wdFindContinue
find.Format = False
find.MatchCase = false
find.MatchWholeWord = false
find.MatchWildcards = false
find.MatchSoundsLike = false
find.MatchAllWordForms = false

// Nun der interessante Teil

Dim replaceParam as New OLEParameter

replaceParam.Value = Office.wdReplaceAll
// passend zur Doku von Find.Execute ist der Replace-Parameter der eflte
replaceParam.Position = 11

find.Execute replaceParam
```

Das ist es. Ganz offensichtlich ist der schwierigste Teil das Herausfinden der korrekten Position des Names-Parameters (named parameter). Dies ist das einzige Mal, daß Sie VBA starten und in dem Objekt-Browser schauen müssen.

Problemlösungen

Hier ist die Prozedur, um festzustellen, ob die OLE-Bibliotheken installiert sind.

Eine Möglichkeit ist es, den Visual Basic Editor zu starten und die Automatisierung mit VBA zu probieren.

Hier sind die Schritte, um PowerPoint aus Word heraus zu automatisieren:

- Word starten.
- Visual Basic Editor in Word starten.
- Ein Benutzerformular einfügen.
- Einen CommandButton in das Formular einfügen.
- Den Event des Buttons bearbeiten und diesen Code einfügen:

```
Dim obj as Object
Set obj = CreateObject("PowerPoint.Application")
obj.Activate
```

- Das Programm starten und den Button drücken.

Wenn PowerPoint startet und Sie keine Fehler bekommen, sind die OLE-Bibliotheken korrekt installiert.

Fehlerbehandlung

Fehler werden über [OLEExceptions](#) angezeigt, deshalb müssen Sie diese Ausnahmen (Exceptions) verwenden, wenn Sie Fehler überprüfen wollen. Hier wird das letzte Kommando ausgegeben, das einen Fehler verursachte. Inklusive zusätzlicher Informationen über das [OLEExceptions](#)-Objekt.

```
Dim word as New WordApplication
word.ShowClipboard
Exception err as OLEException
Msgbox err.message
```

Siehe auch

[WordApplication](#), [ExcelApplication](#), [PowerPointApplication](#), [OLEObject](#)-Klassen.



OLEContainer Steuerelement

Der allgemeine Container, um ActiveX-Steuerelemente (nur Windows) einzubetten.

Übergeordnete Klasse

[RectControl](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Content</i>	OLEObject	Das mit dem ActiveX-Steuerelement verbundene OLEObject. Hiermit automatisieren Sie das ActiveX-Steuerelement.
<i>ProgramID</i>	String	Die Program-ID des ActiveX-Steuerelements, wie es in der Registry gespeichert ist. Es kann ebenso die GUID sein (eingeschlossen in geschweiften Klammern).

Methoden

Name	Parameter	Beschreibung
<i>Create</i>		Erzeugt ein ActiveX-Steuerelement. Create verwendet die ProgramID-Eigenschaft, um das ActiveX-Steuerelement zu erzeugen. Liefert ein Boolean : True falls erfolgreich, False wenn nicht erfolgreich.
<i>Destroy</i>		Zerstört das derzeit eingebettete ActiveX-Steuerelement, damit Sie Ihren OLEContainer wieder verwenden können.
<i>ShowPropertyPages</i>		Zeigt die Eigenschaften-Seiten des ActiveX-Steuerelements, wenn es welche gibt. Nicht alle ActiveX-Steuerelemente unterstützen Eigenschaften-Seiten.

Events

Name	Parameters	Beschreibung
<i>EventTriggered</i>	NameOfEvent als String Parameters() als array of Variants	Tritt auf, wenn das ActiveX-Steuerelement ein Event vom Anwender empfängt. Der Event-Name wird als erster Parameter übergeben und legt fest, welches Event auftrat. Der zweite Parameter enthält die Parameter für dieses Event, übergeben als ein Array von Variants. Der Anwender sollte das UBound der Parameter-Liste überprüfen.
<i>ShowObject</i>		Tritt auf, wenn das ActiveX-Objekt bereit ist, angezeigt zu werden.

Siehe auch

[OLEObject](#) Klasse.

OLEException Fehler

Ein OLE-bezogener Laufzeitfehler ist aufgetreten.

Übergeordnete Klasse

RuntimeException

OLEObject Klasse

Wird zum Automatisieren von COM-Servern verwendet. Benutzen Sie die WordApplication-, ExcelApplication- und PowerPointApplication- Klassen, um Microsofts Office-Applikationen zu automatisieren.

Übergeordnete Klasse

Object

Constructors

Name	Parameter	Beschreibung
<i>OLEObject</i>	ProgramID als <u>String</u>	<i>ProgramID</i> ist die Programm-ID des COM-Servers, wie es in der Registry gespeichert ist. Es kann auch die Class-ID sein (in geschweiften Klammern). Dieser Kontruktor versucht, eine vorhandene Instanz des COM-Servers zu finden, wenn dieser läuft. Andernfalls erzeugt es eine neue Instanz.
<i>OLEObject</i>	ProgramID als <u>String</u> NewInstance als <u>Boolean</u>	Der ProgramID-Parameter ist der gleiche wie oben. Der zweite Parameter legt fest, ob eine neue Instanz des COM-Servers erzeugt werden soll (<u>True</u>) oder ob versucht werden soll, eine schon vorhandene zu verwenden, wenn diese bereits läuft (<u>False</u>).

Methoden

Name	Parameter	Beschreibung
<i>Invoke</i>	NameOfFunction als <u>String</u>	Ruft eine Methode des COM-Servers auf. Liefert ein <u>Variant</u> .
<i>Invoke</i>	NameOfFunction als <u>String</u> Parameters() als array of <u>Variants</u>	Ruft eine InvMethode des COM-Servers auf und übergibt das Array aus Parametern dieser Methode. Stellen Sie sicher, daß das Array die korrekte Dimension aufweist, da dies die Anzahl an Parametern festlegt, die an die Methode übergeben werden. Der erste Parameter beginnt bei 1.
<i>Property</i>	PropertyName als <u>String</u> [Zuweisung durch Wert]	Wird zum Setzen oder Ermitteln einer Eigenschaft des Objekts verwendet. Der Parameter ist der Name der Eigenschaft, der ein neuer Wert zugewiesen werden soll. Ist der optionale Parameter „Zuweisung durch Wert“ angegeben, wird die Zuweisung durch den Wert bestimmt.

Hinweise

In der Voreinstellung werden die Eigenschaften eines **OLEObject**-Objekts mit Werten gefüllt. Wird ein Fehler festgestellt, wird versucht die eingetragenen Werte als Referenz zu verwenden, falls es sich um ein Objekt handelt. Wenn der optionale Parameter *ByValue* auf True gesetzt ist, wird einer Eigenschaft ein Wert zugewiesen. Andernfalls wird eine Referenz zugewiesen. In VisualBasic wird eine Zuweisung einer Referenz durch den Set-Befehl durchgeführt, aber da REALbasic dies nicht unterstützt, müssen Sie den *ByValue*-Parameter verwenden, wenn Sie wissen, daß Sie eine Referenz zuweisen wollen.

Beispiel

Folgendes Beispiel steuert den Internet Explorer:

```
Dim obj as OLEObject
Dim v as Variant
Dim params(1) as Variant

// InternetExplorer.Application ist die Programm-ID.
// Die Class-ID ist {0002DF01-0000-0000-C000-000000000046}
// RB akzeptiert beide Formate
```

```
//obj = New OLEObject("{0002DF01-0000-0000-C000-000000000046}", True)
obj = New OLEObject("InternetExplorer.Application", True)
obj.property("Visible") = True
params(1) = "http://www.realsoftware.com/"
//v = obj.invoke("Navigate", params)
//v = obj.invoke("this_will_throw_an_exception")
```

```
Exception err as OLEException
Msgbox err.message
```

Siehe auch

[OLEContainer](#), [Office](#), [WordApplication](#), [ExcelApplication](#), [PowerPointApplication](#) Klassen; [OLEException](#) Fehler.

OLEParameter Klasse

Wird verwendet, um einen OLE-Parameter anzugeben.

Eigenschaften

Name	Typ	Beschreibung
PassByRef	Boolean	Wenn True , wird der Parameter als Referenz angegeben. Voreingestellt ist False .
Position	Integer	Erlaubt Ihnen die Position (1-basierter Index) des Parameters anzugeben, den Sie als Referenz übergeben wollen. Dies ist ähnlich eines Namens-Parameters (named parameter). Wenn Sie die Position des Parameters in der Funktion kennen, können Sie ein OLEParameter-Objekt erzeugen und diese Eigenschaft entsprechend setzen, um denselben Effekt zu erzielen. Voreingestellt ist 0, was keine Position bedeutet.
Type	Integer	Gibt den Typ des Parameters an.
Value	Variant	Der Wert des Parameters.

Siehe auch

[OLEContainer](#), [OLEObject](#)-Klassen.

Open4DDatabasebyADSP Funktion

Öffnet eine 4D-Datenbank (4D Server) über ADSP unter Mac OS „Classic“. Diese Funktion ist obsolet. Bitte verwenden Sie stattdessen die [ADSP4DServer](#)-Klasse. Sie können auch die [TCPIP4DServer](#)-Klasse benutzen, wenn Sie eine 4D Server Datenbank verwenden wollen.

Siehe auch

[ADSP4DServer](#), [TCPIP4DServer](#) Klassen.

Open4DDatabasebyTCPIP Funktion

Öffnet eine 4D-Datenbank (4D Server) über TCP/IP. Diese Funktion ist obsolet. Sie wurde durch die [TCPIP4DServer](#)-Klasse ersetzt.

Siehe auch

[Database](#), [RecordSet](#), [TCPIP4DServer](#) Klassen.

OpenBaseDatabase Klasse

Wird zum Öffnen einer OpenBase-Datenbank verwendet. Bitte verwenden Sie diese Klasse anstelle von [OpenOpenBaseDatabase](#).

Übergeordnete Klasse

[Database](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Encoding</i>	<u>Integer</u>	Dieser Kodier-Wert ist der gleiche Wert, den Sie für <u>GetTextEncoding</u> verwenden würden. Zum Beispiel ist &H201 ISOLatin1, &H202 ist ISOLatin2, und so weiter. Dies wird verwendet, um von Ihrer festgelegten Kodierung in die OpenBase-Datenbank-Kodierung zu konvertieren und umgekehrt. Dies ist aus Kompatibilitätsgründen zur Version 4.0.2 vorhanden und ab Version 4.5 überflüssig.

Hinweise

Um diese Klasse verwenden zu können, müssen Sie das passende Datenbank-Plugin im Plugins-Ordner installiert haben.

Die REALbasic-CD enthält alle Datenbanken-Plugins und Sie finden die aktuellen Versionen auch immer auf der REAL Software Web-Site auf www.realsoftware.com.

Wenn Sie die RecordSet-Klasse verwenden, um Einträge zu bearbeiten oder zu aktualisieren, müssen Sie die „_rowid“-Spalte der Tabelle miteinbeziehen. Tun Sie dies nicht, wird OpenBase den Eintrag nicht finden um ihn bearbeiten zu können.

Achtung: Wenn Sie Daten in die Objektfelder einfügen, kann dies die Datenbank beschädigen. Um dies zu umgehen, fügen Sie den Eintrag ohne die Objektfelder ein und benutzen dann die Update-Methode, um Daten in die Objektfelder einzufügen.

Beispiel

Folgender Code öffnet die OpenBase-Datenbank namens „pubs“. Die Host-Eigenschaft kann ebenso eine IP-Adresse sein.

```
Dim obdb as OpenBaseDatabase
Obdb = New OpenBaseDatabase
Obdb.host = "opendb.mydomain.com"
Obdb.database = "pubs"
If obdb.connect() then
  MsgBox "Gut"
Else
  MsgBox "Schlecht"
end if
```

Siehe auch

Database Klasse.

OpenCSVCursor Funktion

Öffnet eine Textdatei als ein RecordSet. In der Datei müssen Felder mit Kommas und Datensätze mit Returns getrennt sein. Textfelder müssen in doppelte Gänsefüßchen (""") eingeschlossen sein. Kommas sind innerhalb des Strings in Gänsefüßchen verboten. Ein gültiger Datensatz sieht folgendermaßen aus:

```
22,"Daniel","Düsentrieb","Erfinder"
```

Syntax

result=**OpenCSVCursor**(file)

Teil	Typ	Beschreibung
<i>result</i>	<u>RecordSet</u>	RecordSet, das die Zeilen und Spalten der Textdatei enthält.
<i>file</i>	<u>FolderItem</u>	<u>FolderItem</u> , das auf die Textdatei verweist.

Hinweis

OpenCSVCursor benötigt das CSV Database-Plugin. Kopieren Sie das passende Plugin in den Plugins-Ordner. Die REALbasic-CD enthält alle Datenbanken-Plugins und Sie finden die aktuellen Versionen auch immer auf der REAL Software Web-Site auf www.realsoftware.com.

Beispiel

Folgendes Beispiel öffnet die angegebene Textdatei:

```
Dim db as RecordSet
db=OpenCSVCursor(GetFolderItem("table1.txt"))
if db = Nil then
  Beep
  MsgBox "Die Datenbank konnte nicht geöffnet werden"
```

```

else
    //Weiter mit Datenbankoperationen
end if

```

Siehe auch

[Database](#), [RecordSet](#), [DatabaseField](#) Klassen; [NewREALDatabase](#), [Open4DDatabaseByTCPIP](#), [Open4DDatabaseByADSP](#), [OpenDBFCursor](#), [OpenDTFDatabase](#), [OpenODBCDatabase](#), [OpenOracleDatabase](#), [OpenOpenBaseDatabase](#), [OpenREALDatabase](#), [Select4DDatabaseByADSP](#), [Select4DDatabaseByTCPIP](#), [SelectODBCDatabase](#) Funktionen.

OpenDBFCursor Funktion

Öffnet eine Datei im xBase-Format (also eine dbf-Datei von dBase) als [RecordSet](#).

Syntax

result=OpenDBFCursor(file)

Teil	Typ	Beschreibung
result	RecordSet	RecordSet , das die Zeilen und Spalten der .dbf-Datei enthält.
file	FolderItem	FolderItem , das auf die .dbf-Datei verweist.

Beispiel

```

Dim db as RecordSet
db=OpenDBFCursor(GetFolderItem("phone.dbf"))
If db = Nil then
    Beep
    MsgBox "Die Datenbank konnte nicht geöffnet werden."
else
    //Weiter mit Datenbankoperationen
end if

```

Hinweis

OpenDBFCursor benötigt das DBF Database-Plugin. Kopieren Sie das passende Plugin in den Plugins-Ordner. Die REALbasic-CD enthält alle Datenbanken-Plugins und Sie finden die aktuellen Versionen auch immer auf der REAL Software Web-Site auf www.real-software.com.

Siehe auch

[Database](#), [RecordSet](#), [DatabaseField](#), [DatabaseRecord](#) Klassen; [OpenDTFDatabase](#), [OpenCSVCursor](#), [OpenOracleDatabase](#), Funktionen.

OpenDialog Klasse

Erzeugt eine benutzerdefinierbare „Datei öffnen“-Dialogbox. Die nicht-benutzerdefinierbare Version dieses Dialogs wird mit der Funktion [GetOpenFolderItem](#) angezeigt.

Übergeordnete Klasse

[FolderItem](#) Dialog

Hinweise

Durch die Eigenschaften der FolderItemDialog-Klasse können Sie folgende Aspekte einer „Datei öffnen“-Dialogbox selbst definieren:

- Position (Eigenschaften Left und Top)
- Ausgangsverzeichnis (Eigenschaft Initial Directory)
- Gültige Dateitypen, die angezeigt werden sollen (Eigenschaft Filter)
- Beschriftung der Bestätigen- und Abbrechen-Knöpfe (Eigenschaften ActionButtonCaption und CancelButtonCaption)
- Text, der im Titelfeld des Dialogs erscheint (Eigenschaft Title)
- Text, der innerhalb des Dialogs erscheint (Eigenschaft PromptText)

Beispiel

Folgender Code öffnet eine benutzerdefinierbare „Datei öffnen“-Dialogbox und zeigt das Online-Handbücher-Verzeichnis im Datei-Browser an. Nur MIF-Dateien werden angezeigt. Der Dateityp, der an den Filter-Parameter übergeben wird, wurde zuvor im Dateitypen-Dialog definiert.

```
Dim dlg as OpenFileDialog
Dim f as FolderItem
dlg=New OpenFileDialog
dlg.InitialDirectory=GetFolderItem("HD:FrameMaker6.0:OnlineManuals")
dlg.Title="Wähle eine MIF-Datei"
dlg.Filter="application/x-mif"
f=dlg.ShowModal()
If f <> Nil then
    //Zeigt den vollständigen Pfad zu den Dateien in einem EditField an
    resultEdit.text = dlg.Result.AbsolutePath
Else
    resultEdit.text = "Benutzer-Abbruch"
End if
```

Siehe auch

[FolderItemDialog](#), [SelectFolderDialog](#), [SaveAsDialog](#), [FolderItem](#) Klassen; [GetOpenFolderItem](#) Funktion.

OpenDTFDatabase Funktion

Öffnet eine dtF-Datenbank.

Syntax

result=OpenDTFDatabase (dbFile,blobFile,username,password)

Teil	Typ	Beschreibung
result	Database	Datenbank-Objekt, das auf die dtF-Datenbank verweist.
dbFile	FolderItem	Alle Felder mit fester Länge in der dtF-Datei.
blobFile	FolderItem	Alle BLOB-Felder in der dtF-Datei.
username	String	Benutzername für die dtF-Datenbank.
password	String	Paßwort für die dtF-Datenbank.

Hinweise

OpenDTFDatabase benötigt das dtfSQL Database-Plugin. Kopieren Sie das passende Plugin in den Plugins-Ordner. Die REALbasic-CD enthält alle Datenbanken-Plugins und Sie finden die aktuellen Versionen auch immer auf der REAL Software Web-Site auf www.realsoftware.com.

Diese Funktion benötigt die Dtf_PPC Shared Library, die im Systemerweiterungen-Ordner installiert sein muß.

Beispiel

```
Dim db as Database
Dim textdata as folderitem
Dim blobdata as FolderItem
textdata=GetFolderItem("HD:mydb")
blobdata=GetFolderItem("HD:myblobs")
db=openDTFDatabase(textdata, blobdata,"Nancy","Pavlov")
If db = Nil then
    Beep
    MsgBox "Die Datenbank konnte nicht geöffnet werden."
else
    //Weiter mit Datenbankoperationen
end if
```

Siehe auch

[Database](#), [RecordSet](#), [DatabaseRecord](#), [DatabaseField](#) Klassen; [OpenDBFCursor](#), [OpenOracleDatabase](#) Funktionen.

OpenODBCDatabase Funktion

Öffnet eine ODBC-Datenbank. Diese Funktion ist obsolet. Verwenden Sie stattdessen die [ODBCDatabase](#)-Klasse.

Siehe auch

[ODBCDatabase](#) Klasse.

OpenOpenBaseDatabase Funktion

Öffnet eine OpenBase Datenbank. Diese Funktion ist obsolet. Verwenden Sie stattdessen die [OpenBaseDatabase](#) Klasse.

Siehe auch

[OpenBaseDatabase](#) Klasse.

OpenOracleDatabase Funktion

Öffnet eine Oracle-Datenbank.

Syntax

Ergebnis=**OpenOracleDatabase**(host,username,password)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	Database	Datenbank-Objekt, das auf eine Oracle-Datenbank verweist.
<i>host</i>	String	Datenbank-Host. Kann eine Kombination aus Benutzernamen und Host sein, wie im Beispiel gezeigt.
<i>username</i>	String	Benutzername für die Datenbank. Der Benutzername kann den Namen des Hosts enthalten.
<i>password</i>	String	Das Datenbank-Passwort.

Hinweise

OpenOracleDatabase benötigt das Oracle (7.1) Database-Plugin. Kopieren Sie das passende Plugin in den Plugins-Ordner. Die REALbasic-CD enthält alle Datenbanken-Plugins und Sie finden die aktuellen Versionen auch immer auf der REAL Software Web-Site auf <http://www.realsoftware.com>.

Mit REALbasic Version 5.5 unterstützt die OpenOracleDatabase auch Oracle 8i oder 9i unter Mac OS X.

Um OpenOracleDatabase benutzen zu können, müssen die OCI Shared Libraries im Systemerweiterungen-Ordner installiert sein. Um auf Datenbanken auf Remote-Rechnern zugreifen zu können, muß außerdem SQL*Net installiert sein.

Beispiel

```
Dim db as Database
db=openOracleDatabase("Allison@nickelodeon","rugRat")
If db = Nil then
    Beep
    MsgBox "Die Datenbank konnte nicht geöffnet werden."
else
    //Weiter mit Datenbankoperationen
end if
```

Siehe auch

[Database](#), [RecordSet](#), [DatabaseField](#), [DatabaseRecord](#), [OracleDatabase](#) Klassen.

OpenPostgreSQLDatabase Funktion

Öffnet eine Postgres-Datenbank. Diese Funktion ist obsolet. Verwenden Sie stattdessen die [PostgreSQLDatabase](#) Klasse.

Siehe auch

[PostgreSQLDatabase](#) Klasse.

OpenPrinter Funktion

Liefert ein Graphics-Objekt, das auf dem momentan ausgewählten Drucker gedruckt werden kann.

Syntax

Ergebnis=**OpenPrinter**([PageSetup])

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Graphics</u>	Ein <u>Graphics</u> -Objekt, das die zu druckende Seite repräsentiert.
<i>PageSetup</i>	<u>PrinterSetup</u>	Optional. Ein <u>PrinterSetup</u> -Objekt, dessen Eigenschaften (z.B. Seitenausrichtung oder Skalierung) zum Druck verwendet werden.

Hinweis

Die Funktion **OpenPrinter** liefert ein Graphics-Objekt zurück. Die verschiedenen Zeichenroutinen können verwendet werden, um in dieses Graphics-Objekt zu zeichnen, woraufhin diese Daten zum Ausdruck an den Drucker geschickt werden.

Beispiele

Dieses Beispiel druckt „Hallo Welt“ auf dem ausgewählten Drucker.

```
Dim g As Graphics
g = OpenPrinter()
g.DrawString "Hallo Welt", 100, 100
```

Dieses Beispiel verwendet zusätzlich die in der Variable „ps“ gespeicherten PageSetup-Eigenschaften für den Ausdruck:

```
Dim g As Graphics
g = OpenPrinter(ps)
g.DrawString "Hallo Welt", 100, 100
```

Siehe auch

Graphics Klasse, OpenPrinterDialog Funktion, PrinterSetup Klasse.

OpenPrinterDialog Funktion

Zeigt die Druck-Dialogbox und liefert ein Graphics-Objekt zurück, damit auf den aktuell ausgewählten Drucker gedruckt werden kann.

Syntax

Ergebnis=**OpenPrinterDialog**([PageSetup], [window])

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Graphics</u>	Ein <u>Graphics</u> -Objekt, das die zu druckende Seite repräsentiert.
<i>PageSetup</i>	<u>PrinterSetup</u>	Optional. Ein <u>PrinterSetup</u> -Objekt, dessen Eigenschaften (z.B. Seitenausrichtung oder Skalierung) zum Druck verwendet werden.
<i>window</i>	<u>Window</u>	Optional. Ein Sheet-Fenster, das Sie als das Seite einrichten-Sheet-Fenster an Stelle des eingebauten Seite einrichten-Dialogs verwenden können.

Hinweis

Die Funktion **OpenPrinterDialog** liefert ein Graphics-Objekt zurück. Die verschiedenen Zeichenroutinen können verwendet werden, um in dieses Graphics-Objekt zu zeichnen, woraufhin diese Daten dann zum Ausdruck an den Drucker geschickt werden. Sie können die Werte in den Feldern „Kopien“, „von Seite ...“ und „bis Seite ...“ mit Hilfe der Eigenschaften Copies, FirstPage und LastPage des Graphics-Objektes auslesen, das von der Funktion **OpenPrinterDialog** zurückgeliefert wird.

Klickt der Anwender auf den Abbruch-Button, so liefert die Funktion **OpenPrinterDialog** Nil zurück.

Da beide Parameter optional sind, kann es passieren, daß Sie den zweiten, aber nicht den ersten Parameter übergeben wollen. Wenn Sie nur einen Parameter übergeben, geht REALbasic davon aus, daß es sich um den ersten Parameter handelt. Wenn Sie nur den zweiten Parameter übergeben wollen, übergeben Sie als ersten Parameter Nil, z.B.:

```
Dim g As Graphics
g = OpenPrinterDialog(Nil, myWindow)
```

Beispiele

Dieses Beispiel druckt „Hallo Welt“ auf dem ausgewählten Drucker.

```
Dim g As Graphics
g = OpenPrinterDialog()
If g<> Nil then
    g.DrawString "Hallo Welt", 100, 100
End if
```

Dieses Beispiel verwendet zusätzlich die in der Variable „ps“ gespeicherten PageSetup-Eigenschaften für den Ausdruck:

```
Dim g As Graphics
g = OpenPrinterDialog(ps)
g.DrawString "Hallo Welt", 100, 100
```

Siehe auch

[Graphics](#), [PrinterSetup](#), [StyledTextPrinter](#) Klassen; [OpenPrinter](#) Funktion.

OpenREALdatabase Funktion

Öffnet eine REAL Datenbank. Diese Funktion ist obsolet. Verwenden Sie stattdessen die [REALdatabase](#)-Klasse.

Siehe auch

[REALdatabase](#) Klasse.

OpenURLMovie Funktion

Öffnet ein Movie an der angegebenen URL.

Syntax

result=**OpenURLMovie**(URL)

Teil	Typ	Beschreibung
<i>result</i>	Movie	Movie-Objekt.
<i>URL</i>	String	URL, an der sich das QuickTime-Movie befindet.

Hinweis

Wenn der Anwender QuickTime 4 (oder neuer) installiert hat, unterstützt OpenURLMovie auch Video-Streaming.

Beispiel

Folgendes Beispiel lädt ein Video-Streaming und weist es der Movie-Eigenschaft eines [MoviePlayer](#)-Steuerelements zu:

```
MoviePlayer1.movie=openURLMovie("http://www.apple.com/trailers/miramax/frida.html")
```

Hinweis: Es kann sein, daß Apple Computer, Inc. diesen Film-Trailer irgendwann entfernt. Er ist hier nur aufgeführt, um die Syntax der Funktion zu illustrieren.

Siehe auch

[MoviePlayer](#) Steuerelement, [Movie](#) Klasse.

Operator_ Schlüsselwort

Erlaubt Vergleiche und arithmetische Operationen bei selbstdefinierten Klassen auszuführen.

Syntax

result=Operator_operation(ob)

Teil	Typ	Beschreibung
result	Type	Der Typ der selbstdefinierten Klasse, der die arithmetische Operation hinzugefügt wurde oder eine Zahl oder ein String, der das Ergebnis eines Vergleichs darstellt, wenn Compare als operation verwendet wurde.
ob	Type	Der Typ der selbstdefinierten Klasse, der der Vergleich oder der die arithmetische Operation hinzugefügt wurde.
operation	Literal String	Gewünschter Vergleich oder arithmetische Operation.

Hinweise

Das **Operator_**-Schlüsselwort repräsentiert ein Set aus reservierten Wörtern, die es Ihnen erlauben, die gebräuchlichen arithmetischen Operationen und Vergleichs-Operatoren mit selbstdefinierten Klassen zu implementieren. Die unterstützten Operatoren und Schlüsselwörter sind im Folgenden aufgeführt:

Operator	Schlüsselwort
+	Operator_Add
	Operator_AddRight
-	Operator_Subtract
	Operator_SubtractRight
*	Operator_Multiply
	Operator_MultiplyRight
/	Operator_Divide
	Operator_DivideRight
\	Operator_IntegerDivide
	Operator_IntegerDivideRight
Mod	Operator_Modulo
	Operator_ModuloRight
=, <, >, <=, >=	Operator_Compare
(negation)	Operator_Negate
(convert)	Operator_Convert

Um zum Beispiel in der Lage sein zu können, zwei Instanzen einer benutzerdefinierten Klasse mit dem „+“-Operator zu addieren, müssen Sie der Klasse eine Funktion namens „**Operator_Add**“ hinzufügen. Der Additions-Operator, +, ist für diese Klasse so lange nicht definiert, bis sie dies getan haben.

Hier ist ein Beispiel. Angenommen, Sie haben ein QueryResult-Objekt, das eine Liste gefundener Datensätze gespeichert hat und Sie wollen diese mit einem anderen QueryResult kombinieren, indem Sie einen eigenen Additions-Operator definieren. Definieren Sie das Operator_Add-Schlüsselwort mit einer Funktion etwa so:

```
Function Operator_Add( rval As QueryResult ) As QueryResult
    Dim result As QueryResult
    Dim item As FoundRecord
    result = New QueryResult
    For Each item In FoundRecords
        result.FoundRecords.Append item
    Next
    For Each item In rval.FoundRecords
        result.FoundRecords.Append item
    Next
    Return result
End Function
```

Nachdem Sie diese Definition eingebaut haben, wird diese Funktion aufgerufen, wenn Sie zwei QueryResult-Objekte addieren.

Alle diese Operator-Methoden definieren Self als den linken Operanden und übergeben den rechten Operanden als einen Parameter. Manchmal ist das nicht ausreichend; wenn Sie eine Vector-Klasse haben und eine Skalar-Multiplikation erlauben möchten, könnten Sie eine `Operator_Multiply`-Methode wie diese definieren:

```
Function Operator_Multiply(opB As Double) As Vector
```

Danach können Sie einen Ausdruck folgendermaßen schreiben:

```
vectorA = VectorB * 2.5
```

Sie können aber nicht folgenden Ausdruck verwenden:

```
vectorA = 2.5 * VectorB
```

Für diese Situationen gibt es ein zusätzliches Set von Operatoren. Sie sind mit den normalen arithmetischen Operatoren identisch, kehren aber die Reihenfolge um: Self ist nun der rechte Operand und der linke Operand wird als Parameter übergeben.

```
Operator_AddRight(opA As Type) As Type
Operator_SubtractRight(opA As Type) As Type
Operator_MultiplyRight(opA As Type) As Type
Operator_DivideRight(opA As Type) As Type
Operator_IntegerDivide(opA As Type) As Type
Operator_ModuloRight(opA As Type) As Type
```

Die normalen Methoden werden immer bevorzugt; REALbasic verwendet die Rechts-Versionen nur, wenn es keine gültige Links-Version gibt.

Es gibt zwei Arten von Konvertierungs-Operatoren. Der `Convert To`-Operator besitzt keine Parameter und liefert einen Wert:

```
Operator_Convert() As Type
```

Das erlaubt es Ihnen, das Objekt überall dort zu verwenden, wo der Rückgabewert erlaubt ist. Sie können diesen Operator mit so vielen verschiedenen Rückgabebetypen überladen wie Sie wollen, so lange es immer klar ist, welche Konvertierung beabsichtigt ist. Sie können z.B. nicht gleichzeitig in einen Integer und ein Double konvertieren, da beides in der gleichen Situation möglich wäre.

Der andere Konvertierungs-Operator ist ein „konvertiere von“. Das ist wie eine Vereinigung aus dem C++ Copy-Constructor und dem Zuweisungs-Operator. Er besitzt einen Parameter und liefert keinen Wert zurück:

```
Operator_Convert(val As Type)
```

Im „konvertiere von“-Beispiel wurde eine vorhandene Objekt-Instanz in einen anderen Typ konvertiert. Mit dem `From`-Operator wird eine neue Instanz erzeugt und die passende Konvertierung wird aufgerufen, um sie zu initialisieren. Wenn eine Objekt-Instanz durch Konvertierung erzeugt wurde, wird kein Constructor aufgerufen – der Konvertierungs-Operator übernimmt ihren Platz.

Wenn in einer bestimmten Situation beide Konvertierungsarten auftauchen, bekommt die „von“-Konvertierung von REALbasic Priorität.

Wenn Sie ein Objekt in einem Ausdruck verwenden und dieser keinen passenden Operator definiert, gibt der Compiler einen Undefinierten Operator-Fehler aus.

Or Operator

Wird verwendet, eine logische Vergleich zwischen zwei boole'schen Ausdrücken herzustellen.

Syntax

Ergebnis = Ausdruck1 Or Ausdruck2

Teil	Beschreibung
<i>Ergebnis</i>	Ein <u>boole'scher</u> Wert.
<i>Ausdruck1</i>	Ein gültiger <u>boole'scher</u> Ausdruck.
<i>Ausdruck2</i>	Ein gültiger <u>boole'scher</u> Ausdruck.

Hinweis

Wenn einer der beiden Ausdrücke True ist, ist *Ergebnis* True. Sind beide Ausdrücke False, so ist *Ergebnis* False.

Beispiele

Dieses Beispiel führt mit dem Operator **Or** einen logischen Vergleich durch.

```
Dim a As Boolean
```

```

Dim b As Boolean
Dim c As Boolean
Dim d As Boolean
a=True
b=True
d=a Or b //liefert True
d=d Or c //liefert False

```

Siehe auch

And, Not Operator.

OracleDatabase Klassen

Verwenden Sie diese Klasse mit dem Oracle-Plugin für Oracle 8i oder 9i (Mac OS X).

Übergeordnete Klasse

Database

Eigenschaften

Name	Typ	Beschreibung
Debug	<u>Integer</u>	Setzen Sie dies auf 1, um Debuggen zu aktivieren.

Beispiel

```

Dim db as OracleDatabase
'open a database connection
db = OpenOracleDatabase("alexis","scott","tiger")
If db.Connect then
//weiter mit Datenbank-Operationen
end if

```

Siehe auch

Database, RecordSet Klassen; OpenOracleDatabase Funktion.

OutOfBoundsException Fehler

Tritt auf, wenn der Code versucht, auf ein Array-Element zuzugreifen, das nicht existiert.

Übergeordnete Klasse

RuntimeException Klasse.

Hinweise

Ein OutOfBoundsException Fehler tritt auf, wenn Ihr Code einen Array-Index verwendet, der außerhalb des Bereichs liegt. Das tritt z.B. auf, wenn Sie vergessen, daß Arrays in REALbasic nullbasiert sind und Sie einen Fehler machen, wenn Sie auf das letzte Element zuzugreifen versuchen.

Beispiele

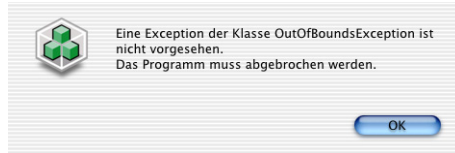
Dieses Beispiel ist dazu gedacht, die Fonts, die auf dem Rechner des Anwenders installiert sind, in einer ListBox darzustellen. Es erzeugt eine OutOfBoundsException, da die Schleife von 0 bis zu nFonts-1 und nicht von 1 bis nFonts gehen sollte. Wenn die Schleife bei nFonts ankommt, ist der Wert von i außerhalb des Bereiches und ein Fehler tritt auf.

```

Dim i, nFonts as Integer
nFonts=FontCount
For i=1 to nFonts //Tun Sie das nicht!
ListBox1.addrow(font(i))
Next

```

Wenn Sie diesen Code in einer erzeugten Applikation ausführen lassen, werden Sie eine Fehlermeldung sehen, wenn i den Wert nFonts erreicht:



Die Applikation wird beendet, wenn der Anwender diesen Dialog bestätigt. Sie können die Exception durch Hinzufügen von folgendem Code behandeln:

```
Dim i, nFonts as Integer
nFonts=FontCount
For i=1 to nFonts
    ListBox1.addrow(font(i))
Next
Exception err
If err IsA OutOfBoundsException then
    MsgBox "Der Wert: "+Str(i)+" liegt außerhalb des Bereichs!"
end if
```

Wenn der Code in der erzeugten Applikation ausgeführt wird, sieht der Anwender folgende Dialogbox:



Die Applikation wird nicht beendet, wenn der Anwender die Dialogbox bestätigt.

Siehe auch

[RuntimeException](#) Klasse, [NilObjectException](#), [IllegalCastException](#), [StackOverflowException](#), [TypeMismatchException](#) Fehler, [Function](#), [Raise](#) Befehl, [Nil](#) Schlüsselwort, [Exception](#) Block.

OutOfMemoryException Fehler

Wird in bestimmten Fällen aufgerufen, wenn der Speicher ausgeht.

Übergeordnete Klasse

[RuntimeException](#) Klasse

Hinweise

Eine **OutOfMemoryException** wird ausgelöst, wenn das Betriebssystem eine Operation nicht ausführen konnte, da nicht genügend Speicher zur Verfügung stand. Es ist schwer zu sagen, wann dies auftritt, da alle von REALbasic unterstützten Betriebssysteme dynamische Speicher-Allozierung verwenden. Mit Ausnahme von Mac OS „Classic“. Eine Möglichkeit wäre, wenn Sie versuchen, eine sehr große Grafik in den Speicher zu laden.

Beispiel

Dieses Beispiel verwendet den Catch-Befehl in einem Open-Event-Handler eines Fensters, um eine OutOfMemoryException abzufangen und zu behandeln, wenn ein importiertes GIF-Bild gezeichnet werden soll. Die Variable myPicture ist eine globale Eigenschaft. Das Bild „Logo“ wurde in das Projektfenster gezogen.

```
Sub Open
    myPicture=New Picture(Logo.width,Logo.height,32)
    myPicture.Graphics.DrawPicture Logo,0,0

    Catch err as OutOfMemoryException
        MsgBox "Nicht genug Speicher um das Bild zu zeichnen!"
        Quit
    End Sub
```

Siehe auch

[RuntimeException](#) Klasse, [NilObjectException](#) Fehler, [IllegalCastException](#) Fehler, [StackOverflowException](#) Fehler, [TypeMismatchException](#) Fehler, [Function](#) Befehl, [Raise](#) Befehl, [Nil](#) Schlüsselwort, [Exception](#) Block.

Oval Steuerelement



Zeichnet ein Oval.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Eigenschaften

Name	Art	Beschreibung
<i>BorderColor</i>	Color	Die Rahmenfarbe des Ovals.
<i>BorderWidth</i>	Integer	Die Rahmenbreite des Ovals
<i>FillColor</i>	Color	Die innere Füllfarbe des Ovals

Events

Name	Parameter	Beschreibung
<i>MouseDown</i>	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des Ovals an der Position x,y gedrückt. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen.
<i>MouseDrag</i>	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des Ovals gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des Ovals an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.

Siehe auch

[RectControl](#) Klasse.

OvalShape Klasse

Wird zum Zeichnen von (zweidimensionalen) Ellipsen in einer Vektorgrafik-Umgebung verwendet.

Übergeordnete Klasse

[RectShape](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Segments</i>	Integer	Die Anzahl der zu verwendenden Polygon-Seiten, wenn eine Annäherung benötigt wird. Null bedeutet, daß es dem Renderer überlassen wird.

Beispiel

Platzieren Sie folgende Methode in den MouseDown-Event-Handler eines Fensters. Sie zeichnet ein Oval im Fenster, wenn der Anwender die Maustaste drückt.

```
Dim o as OvalShape
o = New OvalShape
o.width = 60
o.height = 30
o.FillColor = RGB(127, 127, 255)
Graphics.DrawObject o, x, y
```


Siehe auch

[ArcShape](#), [CurveShape](#), [FigureShape](#), [FolderItem](#), [Group2D](#), [Graphics](#), [Picture](#), [PixmapShape](#), [RectShape](#), [RoundRectShape](#), [StringShape](#) Klassen.



PagePanel Steuerelement

Ähnlich dem [TabPanel](#)-Steuerelement, beinhaltet aber keine Tabs (oder irgend einen anderen sichtbaren Anzeiger), um zwischen den unterschiedlichen Seiten zu wechseln. Das [TabPanel](#) ist als Unterklasse vom **PagePanel** abgeleitet.

Übergeordnete Klasse

[RectControl](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Value</i>	Integer	Die Nummer der aktuell gewählten Seite. Die erste Seite hat die Nummer Null. Um die Seitenzahl zu ändern, die gerade angezeigt wird, setzen Sie Value auf einen anderen Wert. Dies funktioniert ebenso mit dem angezeigten Panel eines TabPanel -Steuerelements, das eine Unterklasse von PagePanel ist.

Events

Name	Parameter	Beschreibung
<i>Change</i>		Die oberste Seite hat sich geändert.

Hinweise

Das **PagePanel**-Steuerelement erlaubt Ihnen, Seiten Steuerelemente hinzuzufügen, so wie Sie Steuerelemente den einzelnen Tabs in einem [TabPanel](#) hinzufügen. Der Unterschied ist, daß das **PagePanel**-Steuerelement selbst in Ihrer erzeugten Applikation nicht sichtbar ist – nur die Steuerelemente auf der aktuell gewählten Seite sind sichtbar. Sie können die dargestellte Seite durch Ändern des Wertes der Value-Eigenschaft wechseln. Sie können PagePanels innerhalb von PagePanels integrieren.

In der IDE können Sie wie bei einem [TabPanel](#) Dinge hinzufügen, einfügen, löschen, umsortieren und von Seite zu Seite navigieren. Wenn Sie ein Interface unter Verwendung des **PagePanel** erstellen, können Sie die Vorzüge der Steuerelemente-Hierarchie ausnutzen, wie dies im Abschnitt beim Control-Object beschrieben ist.

Siehe auch

[TabPanel](#) Steuerelement; [Control](#) Objekt.

Paragraph Klasse

Ein Absatz vom Typ [StyledText](#), der Stiltext enthält.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Alignment</i>	Integer	Textausrichtung: 0 – linksbündig 1 – zentriert 2 – rechtsbündig
<i>EndPos</i>	Integer	Endposition.
<i>Length</i>	Integer	Absatzlänge.

Name	Typ	Beschreibung
StartPos	Integer	Startposition

Hinweise

Verwenden Sie die **Paragraph**-Klasse, um auf einen Absatz innerhalb eines Stiltexts vom Typ [StyledText](#) zu verweisen. Ein Absatz wird definiert als Text zwischen zwei Absatztrennzeichen. Ein Trennzeichen kann die [EndOfLine](#)-Funktion oder das korrekte Trennzeichen für die jeweilige Plattform sein, auf der die Applikation läuft. Ein Reihe von Leerzeilen gilt hier nicht als Absatz.

Sie können mit der ParagraphCount-Methode der [StyledText](#) Klasse die Anzahl der Absätze eines [StyledText](#) Objekts ermitteln und über die Paragraph-Methode der [StyledText](#) Klasse auf einzelne Absätze referenzieren.

Ein Absatz hat nur ein Stilattribut, die Ausrichtung (Alignment). Es kann mit Hilfe der SetParagraphAlignment-Methode der [StyledText](#) Klasse gesetzt werden.

Beispiel

Das folgende Beispiel setzt die Ausrichtung auf „Mitte“. Dann ermittelt es die Startposition und die Länge des Absatzes mit Hilfe der **Paragraph**-Klasse und setzt einige Stilattribute für den gesamten Absatz.

```
Dim Text as String
Dim st,ln as Integer

//define four paragraphs in Text
Text = "This is the text that we are going to save " & _
      +"into our file from the EditField." & EndOfLine & _
      +"Isn't that interesting?" & EndOfLine & _
      +"Man, I sure do love using RB to take care of my projects." & EndOfLine & _
      +"That's because the RS Engineering staff is just so awesome!"
EditField1.StyledText.Text = text //four paragraphs in Text

//center the second paragraph
EditField1.StyledText.SetParagraphAlignment(1,1)

//set the second paragraph in Helvetica, 18 pt bold, red
//first get its character positions...
st=EditField1.StyledText.Paragraph(1).StartPos-1
ln=EditField1.StyledText.Paragraph(1).Length

//next apply attributes...
EditField1.StyledText.Bold(st,ln)=True
EditField1.StyledText.Font(st,ln)="Helvetica"
EditField1.StyledText.Size(st,ln)=18
EditField1.StyledText.TextColor(st,ln)=&cFF0000 //red
```

Siehe auch

[StyledText](#), [StyleRun](#), [Range](#), [EditField](#) Klasse.

ParamArray Schlüsselwort

Wird in einem [Sub](#)- oder [Function](#)-Ausdruck verwendet, um festzulegen, daß eine Vielzahl an Parametern vom angegebenen Typ übergeben werden können.

Syntax

ParamArray [parameterName As DataType]

Teil	Beschreibung
parameterName	Name des Parameter, der für eine unbestimmte Anzahl von Werten verwendet wird.
DataType	Datentyp des Parameters. Ein beliebiger gültiger Datentyp.

Hinweise

Mit dem **ParamArray**-Schlüsselwort können Sie eine unbestimmte Anzahl von Werten eines spezifischen Typs an eine Funktion oder Methode übergeben, ohne ein Array zu verwenden. Sie können beim Aufruf einer Funktion oder Methode, die mit einem ParamArray-Parameter deklariert ist, statt eines Arrays eine Komma-separierte Liste von Werten angeben.

Normalerweise würden Sie einen Parameter an eine Methode übergeben, die mit einem ParamArray definiert ist, aber dieser Parameter ist optional.

Beispiel

Die folgende Funktion nimmt eine Liste von Zahlen an, die über ein ParamArray übergeben wird.

```
Function AddNumbers(ParamArray Nums as Integer) As Integer
    Dim i, Total as Integer
    For Each i in Nums
        total=total+i
    Next
    Return Total
End Function
```

Ein Aufruf sieht folgendermaßen aus:

```
Dim n as Integer
n=AddNumbers(5,10,20)
```

Beliebig viele Zahlen können an AddNumbers übergeben werden.

Dieser Ansatz entspricht der Benutzung eines Arrays. Nehmen wir folgende Funktionen an:

```
Function AddArray(Nums() as Integer) As Integer
    Dim i, total as Integer
    For Each i in Nums
        total=total+i
    Next
    Return total
End Function
```

Sie können das Array wie folgt initialisieren und dann die Funktion aufrufen.

```
Dim myArray(-1) as Integer //Deklariere Array ohne die Größe festzulegen
Dim n as Integer
myArray=Array(5,10,20) //erzeuge 3 Elemente und weise Werte zu
n=AddArray(myArray)
```

Siehe auch

Array Funktion, Dim, Function, Sub Befehl.

Parameter und Standardwert müssen vom selben Typ sein

Sie haben einen Wert mit einem inkorrekten Datentyp als einen Default-Wert verwendet. Wenn Sie in der **Neue Methode**-Dialogbox eine Methode erzeugen, können Sie wahlweise einen Default-Wert für jeden Parameter festlegen. Natürlich muß der Datentyp des Default-Wertes zu dem Datentyp des Parameters passen.

Beispiel

Schauen Sie sich folgende Parameter-Deklaration an:

```
s as String = 15
```

Da s als String deklariert ist, muß sein Default-Wert ein String sein. 15 ist aber keiner.

Parameter sind nicht zu dieser Funktion kompatibel

Sie haben einer Funktion Parameter übergeben, die den falschen Datentyp haben.

Beispiel

Die RGB-Funktion erwartet Integer-Werte:

```
Dim c as Color
c=RGB ("rot","grün","blau")
```

ParseDate Funktion

Wertet ein als String übergebenes Datum aus und erzeugt ein Date-Objekt.

Syntax

Ergebnis = **ParseDate**(Text, ByRef ParsedDate)

Name	Art	Beschreibung
<i>Ergebnis</i>	<u>Boolean</u>	Liefert <u>True</u> , wenn <i>Text</i> erfolgreich ausgewertet werden konnte.
<i>Text</i>	<u>String</u>	Der <u>Datums</u> -String, der ausgewertet werden soll.
<i>ParsedDate</i>	<u>Date</u>	Nach erfolgreichem Aufruf enthält dies das ausgewertete Datum aus ein <u>Date</u> -Objekt..

Hinweise

Der *Text*-Parameter muß ein String sein, der ein gültiges Datum darstellt. Sie können den Slash, das Minus, den Punkt oder ein Leerzeichen verwenden, um Tag, Monat und Jahr zu trennen. Folgendes sind gültige Datum-Strings: 7/12/1341, 7-12-1341, 7.12.1341, 7Dez1341, 7.Dez.1341, und 7 Dez 1341.

Wenn Sie nur die letzten beiden Ziffern eines Jahres verwenden, wird vom aktuellen Jahrhundert ausgegangen.

ParseDate wertet keinen Zeit-Wert eines Datums aus, falls vorhanden.

Beispiel

Folgender Code zeigt das ausgewertete Datum in einer Message-Box im eingestellten Datumsformat. Sie können jede Date-Eigenschaft verwenden, um die entsprechenden Werte zu liefern, z.B. Day, Month und Year enthalten Tag, Monat und Jahr des übergebenen Datums.

```
Dim theDate as Date
Dim theTrueBool as Boolean
theDate=New Date
thetruebool=ParseDate("7Dez1341",theDate)
If theTrueBool then
    MsgBox thedate.AbbreviatedDate
else
    MsgBox "Falsches Datums-Format."
end if
```

Siehe auch

Date Klasse, Database Klassen Beispiel.

Picture Klasse

Ein Picture-Objekt enthält ein Bild.

Übergeordnete Klasse

Object

Eigenschaften

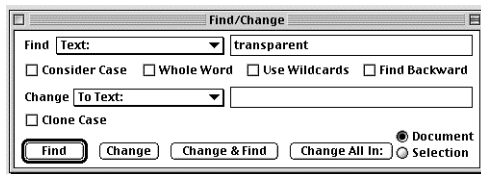
Name	Art	Beschreibung
<i>Depth</i>	<u>Integer</u>	Die Farbtiefe (Bytes) des Bildes. Eine Tiefe von Null zeigt an, daß das Bild überhaupt keine Bitmap hat, sondern für Vektorgrafiken ist.
<i>ImageCount</i>	<u>Integer</u>	Anzahl der indizierten Bilder im Picture (nur ab QuickTime 4).
<i>IndexedImage</i>	<u>Picture</u>	Liefert das i-te Bild im Picture zurück (nur QuickTime 4 und neuer). Der Parameter ist Index als Integer.
<i>Graphics</i>	<u>Graphics</u>	Das Bild.
<i>Height</i>	<u>Integer</u>	Die Höhe des Bildes (in Pixeln).
<i>HorizontalResolution</i>	<u>Integer</u>	Die horizontale Auflösung des Bildes. Wird eingestellt, wenn Sie die OpenAsPicture-Methode der FolderItem-Klasse aufrufen und wird geschrieben, wenn Sie die SaveAsPicture-Methode (der FolderItem-Klasse) aufrufen. Wenn Sie ein Bild erzeugen, ist die voreingestellte Auflösung 72 (Pixel).
<i>Mask</i>	<u>Picture</u>	Erlaubt Ihnen, auf eine Maske zuzugreifen, die die Transparenz des Bildes steuert.
<i>Objects</i>	<u>Group2D</u>	Ein Set von Vektorgrafiken, die mit dem Bild verbunden sind (optional).

Name	Art	Beschreibung
<i>RGBSurface</i>	<u>RGBSurface</u>	Wird nur für Bilder verwendet, die mit <u>NewPicture</u> und einer depth von 32 erzeugt wurden, und liefert sonst nil zurück. Liefert ein <u>RGBSurface</u> Objekt.
<i>Transparent</i>	<u>Integer</u>	Steuert die Transparenz des weißen „Layers“ des Bildes. 0 - Nicht transparent. 1 - Weiß ist transparent.
<i>VerticalResolution</i>	<u>Integer</u>	Die vertikale Auflösung des Bildes. Wird eingestellt, wenn Sie die OpenAsPicture-Methode der FolderItem-Klasse aufrufen und wird geschrieben, wenn Sie die SaveAsPicture-Methode (der FolderItem-Klasse) aufrufen. Wenn Sie ein Bild erzeugen, ist die voreingestellte Auflösung 72 (Pixel).
<i>Width</i>	<u>Integer</u>	Die Breite des Bildes (in Pixeln).

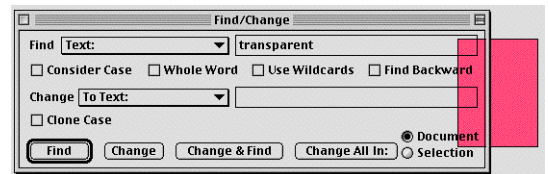
Hinweise

Ein **Picture**-Objekt wird erzeugt, indem man ein Bild in das Projekt-Fenster einfügt, indem man die Methoden OpenAsPicture oder OpenAsVectorPicture eines FolderItem-Objektes aufruft, indem man die NewPicture-Funktion aufruft, oder durch Verwendung des New-Operators. Wenn Sie den New-Operator verwenden, müssen Sie die gleichen Parameter wie bei der Verwendung von NewPicture übergeben: Width als Integer, Height als Integer und Depth als Integer. Width und Height sind Pixel und die Werte von Depth können 0, 1, 2, 4, 8, 16, oder 32 sein. Wenn Sie den New-Operator verwenden und der Speicher ausgeht, tritt ein OutOfMemoryException-Fehler auf. Wenn Sie eine Tiefe von Null festlegen, erzeugt REALbasic ein Bild ohne jegliche Pixelbelegung aber mit vorinstallierter Objects-Eigenschaft. Verwenden Sie diese Option, um Vektorgrafik-Bilder mit dem Set der Unterklassen von Object2D zu erzeugen. **Picture**-Objekte besitzen keine Methoden oder Events.

Die Transparent-Eigenschaft funktioniert nur, wenn sich der erwünschte Effekt dadurch erzielen lässt, dass man die Farbe Weiß transparent macht.



Transparent=0



Transparent=1

Andernfalls müssen Sie mit der Mask-Eigenschaft arbeiten. Wenn Sie z.B. ein Bild namens MyPicture und eine benutzerdefinierte Maske myMask haben, beides im Projektfenster enthalten, wird folgender Code:

```
myPicture.Mask.Graphics.DrawPicture myMask 0,0
```

die benutzerdefinierte Maske der Mask-Eigenschaft von myPicture zuweisen.

Siehe auch

NewPicture Funktion, RGBSurface Objekt, Graphics-Klasse.

PixmapShape Klasse

Wird zum Plazieren eines Bildes in einer Vektorgrafik-Umgebung verwendet.

Übergeordnete Klasse

Object2D

Eigenschaften

Name	Typ	Beschreibung
<i>Image</i>	<u>Picture</u>	Das zu verwendende Quellbild.
<i>SourceHeight</i>	<u>Integer</u>	Höhe (in Pixeln) des Quellbildes von <i>SourceTop</i> , <i>SourceLeft</i> aus. Wird für die Festlegung einer zu verwendenden Unterregion des Bildes verwendet.

Name	Typ	Beschreibung
<i>SourceLeft</i>	<u>Integer</u>	Anzahl der Pixel von der linken Kante des Quellbildes. Wird für die Festlegung einer Unterregion des Bildes zum Zeichnen verwendet.
<i>SourceTop</i>	<u>Integer</u>	Anzahl der Pixel von der oberen Kante des Quellbildes. Wird für die Festlegung einer Unterregion des Bildes zum Zeichnen verwendet.
<i>SourceWidth</i>	<u>Integer</u>	Breite (in Pixeln) von SourceLeft des zu zeichnenden Bildes aus. Wird für die Festlegung einer zu verwendenden Unterregion des Bildes verwendet.

Constructor

Name	Parameter	Beschreibung
<i>PixmapShape</i>	Image als <u>Picture</u>	Initialisiert das Objekt auf das gesamte Quellbild.

Beispiel

Folgende Methode im Action-Event eines PushButtons zeichnet ein rotes Oval im Eltern-Fenster.

```
Dim p as Picture
Dim px as PixmapShape

p=New Picture(240,200,32)
p.graphics.foreColor = &CFF0000
p.graphics.fillOval 0,0,240,200

px=New PixmapShape(p)
px.rotation = 45/57.2958 //45 Grad im Bogenmaß

Graphics.drawObject px,150,150
```

Siehe auch

[ArcShape](#), [CurveShape](#), [FigureShape](#), [FolderItem](#), [Group2D](#), [Graphics](#), [Object2D](#), [OvalShape](#), [Picture](#), [RectShape](#), [RoundRectShape](#), [StringShape](#) Klassen.

Placard Steuerelement



Wird verwendet, um ein Fenster mit einem Placard zu versehen. Ein Placard ist ein Steuerelement, das als Informationsanzeige oder zum Ausfüllen des Hintergrunds von Steuerbereichen verwendet werden kann. Placards haben drei Zustände: normal, gedrückt und disabled.

Die vermutlich bekannteste Anwendung eines Placard-Steuerelements ist als kleines Informationsfeld, das meist am unteren Rand eines Fensters links vom horizontalen Scrollbalken platziert wird.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Eigenschaften

Name	Typ	Beschreibung
<i>Enabled</i>	<u>Boolean</u>	Wenn <u>True</u> , wird ein schattierter Rahmen um das Objekt gezeichnet; wenn <u>False</u> , wird der Rahmen als Linie gezeichnet und das Placard ist disabled.
<i>Value</i>	<u>Boolean</u>	Wenn Enabled <u>True</u> ist, schattiert Value den inneren Bereich des Placard (Gedrücktzustand). Wenn Enabled <u>True</u> und Value <u>False</u> ist, wird der innere Bereich unschattiert gezeichnet (Normalzustand).

Events

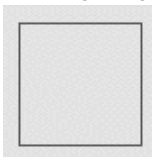
Name	Parameter	Beschreibung
<i>MouseDown</i>	X als Integer , Y als Integer	Der Mausknopf wurde innerhalb des Fensters an den übergebenen lokalen Koordinaten x,y gedrückt.
<i>MouseUp</i>	X als Integer , Y als Integer	Der Mausknopf wurde innerhalb des Fensters an den übergebenen lokalen Koordinaten x,y losgelassen.
<i>MouseDown</i>	X als Integer , Y als Integer	Der Mausknopf wurde innerhalb des Fensters gedrückt und an die lokalen Koordinaten, die in x,y übergeben wurden, gedraggt. Dieses Event wird nicht auftreten, bis Sie True im MouseDown Event zurückliefern.

Hinweis

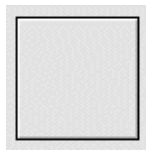
Unter Mac OS X können Sie einige Darstellungsprobleme mit Placards lösen, indem Sie die Composite-Eigenschaft des Eltern-Fensters aktivieren. Ein Beispiel finden Sie bei den Hinweisen zur [Window](#)-Klasse.

Beispiele

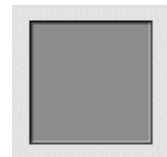
Folgende Abbildungen zeigen ein Placard-Steuerelement in seinen verschiedenen Zuständen:



Disabled



Value=False



Value=True

Siehe auch

[RectControl](#) Klasse.

Pop Methode

Entfernt das letzte Element aus einem Array und liefert dessen Wert zurück.

Syntax

Ergebnis=[Array](#).Pop

Teil	Beschreibung
<i>Ergebnis</i>	Das Element von <i>Array</i> , das entfernt wurde.
<i>Array</i>	Benötigt. Das Array, von welchem der letzte Wert ge„popt“ wird.

Hinweise

Die [Append](#)- und **Pop**-Methoden können zusammen verwendet werden, um ein Array wie einen Stack zu behandeln. Die [Append](#)-Methode pusht ein neues Element auf das Array/den Stack und die Pop-Methode entfernt das letzte Element aus dem Array/vom Stack und liefert dessen Wert zurück.

Beispiel

Dies ist eine Verzeichnis-Durchforsten-Routine, die ohne Rekursion eine Tiefensuche vornimmt:

```
Dim ItemsToInspect(0) As FolderItem
Dim item As FolderItem, ctr As Integer
ItemsToInspect(0) = directory
While Ubound(itemsToInspect) >= 0
    item = itemsToInspect.Pop
    If item.Directory Then
        For ctr = 1 To item.Count
            itemsToInspect.Append item
        Next
    End If
Wend
```

Siehe auch

[Dim](#) Befehl; [Array](#), [Split](#), [Join](#), [Ubound](#) Funktionen; [Append](#), [Insert](#), [Redim](#), [Remove](#), [IndexOf](#), [Insert](#), [Shuffle](#), [Sort](#) Methoden; [ParamArray](#) Schlüsselwort.

POP3SecureSocket Klasse

Wird zum Abrufen von Nachrichten auf einem POP3-Server mittels SSL- oder TLS-Verschlüsselung verwendet.

Übergeordnete Klasse

[SSLSocket](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Username</i>	String	Der Anwendername, der für die Authentifizierung verwendet wird, wenn man eine Verbindung zum Mail-Server herstellt.
<i>Password</i>	String	Das Sicherheits-Paßwort das verwendet wird, wenn man eine Verbindung zum Mail-Server herstellt.
<i>EncryptPassword</i>	Boolean	Falls True , wird das Paßwort verschlüsselt, bevor es an den Mail-Server geschickt wird.

Methoden

Name	Parameter	Beschreibung
<i>CheckServerConnection</i>		Sendet einen „NOOP“-Befehl an den Mail-Server. Das ist einfach ein Befehl, der den Server bittet, zu antworten. Das kann nützlich sein, um zu überprüfen, ob der Mail-Server immer noch antwortet und teilt dem Mail-Server außerdem mit, daß Sie nach wie vor verbunden sind, falls für eine längere Zeit keine Aktivitäten vorhanden waren.
<i>Connect</i>		Stellt mit dem Mail-Server eine Verbindung her und loggt sich mit Username und Password ein.
<i>CountMessages</i>		Fragt den Server nach der Anzahl der Nachrichten in der Mailbox.
<i>DeleteMessage</i>	Index als Integer	Fordert den Mail-Server auf, die angegebene Nachricht zu löschen.
<i>DisconnectFromServer</i>		Trennt die Verbindung zum Mail-Server. Dies sendet einen „QUIT“-Befehl an den Mail-Server und wartet darauf, die Verbindung zu schließen.
<i>ListMessages</i>	[Index als Integer]	Fordert eine Nachrichten-Liste an. Die Liste besteht aus dem Nachrichten-Index und der Größe der Nachricht. Wenn kein Index übergeben wurde, holt es die gesamte Liste vom Server. Wenn ein bestimmter Index übergeben wurde, holt es nur die Nachricht <i>index</i> und die Größe der Nachricht.
<i>RetrieveLines</i>	Index als Integer LineCount als Integer	Liefert die angegebene Anzahl Zeilen einer Nachricht. Der Mail-Server liefert die ersten <i>LineCount</i> Zeilen, die in der Nachricht, die Sie über den Parameter <i>Index</i> festlegen, vorhanden sind. Wenn <i>LineCount</i> Null ist, dann liefert der Mail-Server nur die Header der Nachricht.
<i>RetrieveMessage</i>	Index als Integer	Liest die komplette durch <i>Index</i> festgelegte Nachricht.
<i>RollbackServer</i>		Setzt den Mail-Server in den Zustand zurück, in dem er war, als Sie sich einloggten. Dies kann dazu verwendet werden, um versehentliches Löschen rückgängig zu machen. Die Änderungen werden nicht gespeichert, bevor die Verbindung geschlossen wird. RollbackServer macht Änderungen rückgängig, die noch nicht gespeichert wurden.
<i>SendServerCommand</i>	Command als String	Sendet den durch <i>Command</i> festgelegten Befehl an den Mail-Server. Das ist hilfreich, wenn Sie einen Befehl senden müssen, den REALbasic noch nicht unterstützt.

Events

Name	Parameter	Beschreibung
<i>ConnectionEstablished</i>		Tritt auf, wenn eine Verbindung hergestellt wurde.
<i>Disconnected</i>		Tritt auf, wenn eine Verbindung getrennt wurde.
<i>ListReceived</i>	List als <u>String</u>	Wird ausgeführt, wenn <i>ListMessages</i> aufgerufen wurde. <i>List</i> enthält die Nachrichten-Liste.
<i>LoginSuccessful</i>		Wird ausgeführt, wenn der Login-Prozeß, der durch Aufruf der Connect-Methode initiiert wurde, vollendet ist.
<i>MessageCount</i>	Count als <u>Integer</u>	Wird ausgeführt, wenn der Mail-Server auf einen CountMessages-Aufruf antwortet und enthält die Anzahl der Nachrichten in der Mailbox.
<i>MessageDeleted</i>	Index als <u>Integer</u>	Wird ausgeführt, wenn der Mail-Server auf einen DeleteMessage-Aufruf antwortet und enthält die Index-Nummer der gelöschten Nachricht.
<i>MessageReceived</i>	Index als <u>Integer</u> , Message als <u>EmailMessage</u>	Wird ausgeführt, wenn eine Nachricht vom Mail-Server empfangen wurde, als Antwort auf einen Aufruf von RetrieveMessage. Index enthält die Index-Nummer der erhaltenen Nachricht und der Nachrichten-Inhalt befindet sich in Message.
<i>RollbackSuccessful</i>		Wird als Antwort auf einen Aufruf von RollbackServer ausgeführt und zeigt an, daß der Status der Mailbox zurückgesetzt wurde.
<i>ServerAvailable</i>		Wird ausgeführt, wenn der Mail-Server auf einen Aufruf von CheckServer-Connection geantwortet hat und zeigt an, daß der Mail-Server auf den Aufruf geantwortet hat.
<i>ServerError</i>	ErrorCode als <u>Integer</u> , ErrorMessage als <u>String</u> , MessageID als <u>Integer</u>	Wird ausgeführt, wenn ein Protokoll-bezogener Fehler auftritt. Die Fehler-codes sind folgende: 0 - Unbekannte Fehlermeldung 1 - Paßwort falsch 2 - Anwendername falsch 3 - Nachricht konnte nicht gelöscht werden 4 - Nachrichtenliste konnte nicht abgerufen werden 5 - Empfang der Zeilen ist fehlgeschlagen 6 - Empfang der Nachricht ist fehlgeschlagen
<i>TopLinesReceived</i>	Index als <u>Integer</u> , data als <u>EmailMessage</u>	Wird als Antwort auf einen Aufruf von RetrieveLines ausgeführt. Index enthält die Index-Nummer der bestimmten zu empfangenden Nachricht und data enthält die angeforderten Zeilen der Nachricht.

Hinweise

Die **POP3SecureSocket**-Klasse ist identisch zur **POP3Socket**-Klasse. Der Unterschied ist, daß diese von der **SSLSocket**-Klasse an Stelle von der **TCPSocket**-Klasse abgeleitet ist. Dadurch können Sie gesicherte E-Mails versenden, indem Sie die Secure-Eigenschaft der **SSLSocket**-Klasse setzen.

Beispiel

Folgender Code im MessageReceived-Eventhandler platziert den Haupttext einer E-Mail-Nachricht in ein EditField.

```
Sub MessageReceived(ID as Integer, email as emailMessage)
    Dim s as String
    s=email.bodyHTML
    If s="" then
        s=email.bodyPlainText
    end
    EditField1.text=s
End Sub
```

Siehe auch

SMTPSocket, SMTPSecureSocket, POP3Socket, EmailMessage, HTTPSocket, TCPSocket, SSLSocket, SocketCore Klassen.

POP3Socket Klasse

Wird zum Empfangen und Verwalten von Nachrichten auf einem POP3-Mail-Server verwendet.

Übergeordnete Klasse

TCPSocket

Eigenschaften

Name	Typ	Beschreibung
<i>Username</i>	<u>String</u>	Der Anwendername, der für die Authentifizierung beim Verbinden mit dem Mail-Server verwendet wird.
<i>Password</i>	<u>String</u>	Das Sicherheits-Paßwort, das beim Verbinden mit dem Mail-Server verwendet wird.
<i>EncryptPassword</i>	<u>Boolean</u>	Falls <u>True</u> , wird das Paßwort verschlüsselt, bevor es an den Mail-Server gesendet wird.

Methoden

Name	Parameter	Beschreibung
<i>CheckServerConnection</i>		Sendet einen „NOOP“-Befehl an den Mail-Server. Das ist einfach ein Befehl, der den Server bittet, zu antworten. Das kann nützlich sein, um zu überprüfen, ob der Mail-Server immer noch antwortet und teilt dem Mail-Server außerdem mit, daß Sie nach wie vor verbunden sind, falls für eine längere Zeit keine Aktivitäten vorhanden waren.
<i>Connect</i>		Stellt mit dem Mail-Server eine Verbindung her und loggt sich mit Username und Password ein.
<i>CountMessages</i>		Fragt den Server nach der Anzahl der Nachrichten in der Mailbox.
<i>DeleteMessage</i>	Index als <u>Integer</u>	Fordert den Mail-Server auf, die angegebene Nachricht zu löschen.
<i>DisconnectFromServer</i>		Trennt die Verbindung zum Mail-Server. Dies sendet einen „QUIT“-Befehl an den Mail-Server und wartet darauf, die Verbindung zu schließen.
<i>ListMessages</i>	[Index als <u>Integer</u>]	Fordert eine Nachrichten-Liste an. Die Liste besteht aus dem Nachrichten-Index und der Größe der Nachricht. Wenn kein Index übergeben wurde, holt es die gesamte Liste vom Server. Wenn ein bestimmter Index übergeben wurde, holt es nur die Nachricht <i>index</i> und die Größe der Nachricht.
<i>RetrieveLines</i>	Index als <u>Integer</u> , LineCount als <u>Integer</u>	Liefert die angegebene Anzahl Zeilen einer Nachricht. Der Mail-Server liefert die ersten <i>LineCount</i> Zeilen, die in der Nachricht, die Sie über den Parameter <i>Index</i> festlegen, vorhanden sind. Wenn <i>LineCount</i> Null ist, dann liefert der Mail-Server nur die Header der Nachricht.
<i>RetrieveMessage</i>	Index als <u>Integer</u>	Liest die komplette durch <i>Index</i> festgelegte Nachricht.
<i>RollbackServer</i>		Setzt den Mail-Server in den Zustand zurück, in dem er war, als Sie sich einloggten. Dies kann dazu verwendet werden, um versehentliches Löschen rückgängig zu machen. Die Änderungen werden nicht gespeichert, bevor die Verbindung geschlossen wird. RollbackServer macht Änderungen rückgängig, die noch nicht gespeichert wurden.
<i>SendServerCommand</i>	Command als <u>String</u>	Sendet den durch <i>Command</i> festgelegten Befehl an den Mail-Server. Das ist hilfreich, wenn Sie einen Befehl senden müssen, den REALbasic noch nicht unterstützt.

Events

Name	Parameter	Beschreibung
<i>ConnectionEstablished</i>		Tritt auf, wenn eine Verbindung hergestellt wurde.

Name	Parameter	Beschreibung
<i>Disconnected</i>		Tritt auf, wenn eine Verbindung getrennt wurde.
<i>ListReceived</i>	List als <u>String</u>	Wird ausgeführt, wenn <i>ListMessages</i> aufgerufen wurde. <i>List</i> enthält die Nachrichten-Liste.
<i>LoginSuccessful</i>		Wird ausgeführt, wenn der Login-Prozeß, der durch Aufruf der Connect-Methode initiiert wurde, vollendet ist.
<i>MessageCount</i>	Count als <u>Integer</u>	Wird ausgeführt, wenn der Mail-Server auf einen CountMessages-Aufruf antwortet und enthält die Anzahl der Nachrichten in der Mailbox.
<i>MessageDeleted</i>	Index als <u>Integer</u>	Wird ausgeführt, wenn der Mail-Server auf einen DeleteMessage-Aufruf antwortet und enthält die Index-Nummer der gelöschten Nachricht.
<i>MessageReceived</i>	Index als <u>Integer</u> , Message als <u>EmailMessage</u>	Wird ausgeführt, wenn eine Nachricht vom Mail-Server empfangen wurde, als Antwort auf einen Aufruf von RetrieveMessage. Index enthält die Index-Nummer der erhaltenen Nachricht und der Nachrichten-Inhalt befindet sich in Message.
<i>RollbackSuccessful</i>		Wird als Antwort auf einen Aufruf von RollbackServer ausgeführt und zeigt an, daß der Status der Mailbox zurückgesetzt wurde.
<i>ServerAvailable</i>		Wird ausgeführt, wenn der Mail-Server auf einen Aufruf von CheckServerConnection geantwortet hat und zeigt an, daß der Mail-Server auf den Aufruf geantwortet hat.
<i>ServerCommandReply</i>	Command als <u>String</u> , Data als <u>String</u>	Wird als Antwort auf einen Aufruf von SendServerCommand ausgeführt und enthält die Antwort des Mail-Servers auf den übergebenen Befehl.
<i>ServerError</i>	ErrorCode als <u>Integer</u> , ErrorMessage als <u>String</u> , MessageID als <u>Integer</u>	Wird ausgeführt, wenn ein Protokoll-bezogener Fehler auftritt. Die Fehlercodes sind folgende: 0 - Unbekannte Fehlermeldung 1 - Paßwort falsch 2 - Anwendername falsch 3 - Nachricht konnte nicht gelöscht werden 4 - Nachrichtenliste konnte nicht abgerufen werden 5 - Empfang der Zeilen ist fehlgeschlagen 6 - Empfang der Nachricht ist fehlgeschlagen
<i>TopLinesReceived</i>	Index als <u>Integer</u> , data als <u>EmailMessage</u>	Wird als Antwort auf einen Aufruf von RetrieveLines ausgeführt. Index enthält die Index-Nummer der bestimmten zu empfangenden Nachricht und data enthält die angeforderten Zeilen der Nachricht.

Siehe auch

SMTPSocket, EmailMessage, HTTPSocket, TCPSocket, SocketCore Klassen.

PopupArrow Steuerelement



Wird verwendet, um einem Fenster einen Popup-Pfeil hinzuzufügen.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Typ	Beschreibung
<i>Facing</i>	<u>Integer</u>	Ausrichtung und Größe des PopupArrow 0 – Ost 1 – West 2 – Nord 3 – Süd 4 – Ost, klein 5 – West, klein 6 – Nord, klein 7 – Süd, klein

Beispiel

Die folgende Zeile Code ändert die Größe und Ausrichtung eines PopupArrow:

```
popuparrow1.facing=3
```

Siehe auch

RectControl Klasse.

PopupMenu Steuerelement



Ein Steuerelement, das eine Liste von Einträgen repräsentiert. Der Benutzer kann in dieser Liste einen Eintrag auswählen. Das ComboBox-Steuerelement ist ähnlich, der Anwender kann aber zusätzlich den Text eingeben, statt nur den Eintrag auszuwählen.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>Bold</i>	<u>Boolean</u>	Stellt alle Listeneinträge fett dar.
<i>DataField</i>	<u>String</u>	Nur relevant, wenn das PopupMenu in Verbindung mit einem <u>DataControl</u> verwendet wird, um den Inhalt von Feldern einer Datenbanktabelle anzuzeigen. Der Name eines Feldes in der Tabelle, auf die das <u>DataControl</u> verweist.
<i>DataSource</i>	<u>DataControl</u>	Das <u>DataControl</u> , das auf eine Tabelle in einer Datenbank verweist, dessen Felder dargestellt werden. Verwenden Sie die DataField-Eigenschaft, um jedes anzuzeigende Feld mit einem PopupMenu zu verbinden. In der IDE erscheint ein Popup-Menü mit den Feldnamen. Wenn Sie es im Code einstellen, erwartet es einen <u>String</u> . Weisen Sie ihn der Name-Eigenschaft eines DataControls zu. Hinweis: Datenbankfelder können ebenso durch <u>EditFields</u> , <u>ListBoxes</u> und <u>StaticText</u> Steuerelemente angezeigt werden.
<i>InitialValue</i>	<u>String</u>	Eine Liste mit vorbelegten Einträgen, die jeweils durch ein carriage return voneinander getrennt sind. Diese Eigenschaft kann nicht ausgelesen werden, da die Liste aus dem Speicher entfernt wird, sobald das Steuerelement aufgebaut wurde. Sie können einen Vorgabewert für das Steuerelement eingeben, das dann im Popup erscheint, indem Sie diesen Wert bei <i>InitialValue</i> eingeben und den Vorgabewert mittels <i>ListIndex</i> einstellen.
<i>Italic</i>	<u>Boolean</u>	Stellt alle Listeneinträge kursiv dar.
<i>List</i>	Array als <u>String</u>	Die Liste der Listeneinträge
<i>ListCount</i>	<u>Integer</u>	Die Anzahl der Listeneinträge

Name	Art	Beschreibung
<i>ListIndex</i>	<u>Integer</u>	Wird verwendet, um die Nummer des selektierten Objektes zu bekommen oder zu setzen. Das erste Objekt hat die Nummer 0. Ein Wert von -1 bedeutet „keine Selektion“. Kann über das Eigenschaften-Fenster eingestellt werden, um den Defaultwert des Popup-Menüs festzulegen. Wenn Sie die <u>InitialValue</u> -Eigenschaft verwenden, um das PopupMenu-Steuerelement mit Werten zu füllen, können Sie durch einen Wert bei <u>ListIndex</u> festlegen, daß dieser Default-Eintrag in der IDE angezeigt wird.
Text	<u>String</u>	Der Text des ausgewählten Eintrags
<i>TextFont</i>	<u>String</u>	Name der Schrift, die zur Darstellung verwendet wird.
<i>TextSize</i>	<u>Integer</u>	Größe der Schrift, die zur Darstellung verwendet wird.
<i>Underline</i>	Boolean	Stellt alle Listeneinträge unterstrichen dar.

Events

Name	Parameter	Beschreibung
<i>Change</i>		Die Auswahl hat sich geändert.
<i>GotFocus</i>		(Nur Win32) Das Steuerelement hat den Fokus bekommen und besitzt einen Selektionsrahmen um die Beschriftung.
<i>LostFocus</i>		(Nur Win32) Das Steuerelement hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y gedrückt. Wenn Sie <i>MouseDown</i> auswerten möchten, liefern Sie <u>True</u> zurück.
<i>MouseUp</i>	x als <u>Integer</u> y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht <u>True</u> im <i>MouseDown</i> -Event liefern.

Methoden

Name	Parameter	Beschreibung
<i>AddRow</i>	Objekt als <u>String</u>	Hängt das Objekt als neue Zeile ans Ende der Liste an.
<i>AddSeparator</i>		Fügt eine Unterteilungslinie ins Menü ein.
<i>DeleteAllRows</i>		Löscht alle Zeilen der Liste
<i>InsertRow</i>	RowNummer als <u>Integer</u> , Item als <u>String</u>	Legt eine neue Zeile an der Position RowNummer an (wodurch die existierenden Listeneinträge um eine Position nach unten rutschen).
<i>RemoveRow</i>	RowNummer als <u>Integer</u>	Löscht die angegebene Zeile.
<i>RowTag</i>	Row als <u>Integer</u>	Ein „versteckter“ Bezeichner, der mit dem Eintrag <i>Row</i> verknüpft ist. Im Entwicklerhandbuch finden Sie im Kapitel „Datenbankprogrammierung mit REALbasic“ ein Beispiel dafür, wie <i>RowTag</i> verwendet wird, um eine ID für jeden Menüpunkt des Popup-Menüs zu speichern. Liefert ein <u>Variant</u> . Wenn Sie den Wert von <i>RowTag</i> mit einem anderen Wert vergleichen möchten, müssen Sie den Wert von <i>RowTag</i> erst in diesen anderen Datentyp konvertieren.

Beispiele

Dieser Code im Open-Event-Handler eines Popup-Objektes erzeugt ein Popup-Menü und setzt den Vorgabewert auf den aktuellen Monat:

```
Dim s as string
Dim i,last as integer
Dim d as Date
d=New Date
s="Januar,Februar,März,April,Mai,Jun, Juli,August,September,Oktober,November, Dezember"
last=CountFields(s,",")
For i=1 to last
    me.addRow NthField(s,",",i)
Next
me.ListIndex=d.Month-1
```

Überprüfen Sie den Wert von *ListIndex* im *Change*-Event-Handler, um zu ermitteln, welcher Eintrag selektiert wurde.

Dieses Beispiel fügt einen Eintrag zu PopupMenu1 hinzu.

```
PopupMenu1.addrow "Oktober"
```

Folgendes Beispiel füllt den RowTag-Bezeichner mit einer Reihe von Zahlen:

```
Dim i,nItems as Integer
nItems=Me.ListCount-1
For i=0 to nitems
    Me.RowTag(i)=i
Next
```

Da RowTag ein Variant ist, müssen Sie es erst in ein Integer konvertieren, wenn Sie es mit einem anderen Integer vergleichen wollen.

Dies kann mit einer einfachen Anweisung gemacht werden:

```
Dim RecID as Integer
RecID=PopupMenu1.Rowtag(1)
```

Vergleichen Sie danach RecID mit dem anderen Integer.

Dieses Beispiel öffnet ein neues Fenster, wenn ein Listeneintrag ausgewählt wird.

```
Sub Change()
Dim w As ListEditorWindow
If PopupMenu1.text="Edit List..." Then
    w=New ListEditorWindow
End If
End Sub
```

Ändert die Auswahl in einem Popup-Menü:

```
PopupMenu1.listIndex=3
```

Anzeigen des RowTag des gewählten Menüeintrags:

```
EditField1.text=popupmenu1.rowtag(popupMenu1.listindex)
```

Siehe auch

[BevelButton](#), [ContextualMenu](#), [ComboBox](#) Steuerelemente; [RectControl](#) Klasse.

PostgreSQLDatabase Klasse

Wird zum Öffnen einer PostgreSQL-Datenbank verwendet. Bitte verwenden Sie diese Klasse anstelle von [OpenPostgreSQLDatabase](#).

Übergeordnete Klasse

[Database](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Port</i>	Integer	Der Port der PostgreSQL-Datenbank, zu der eine Verbindung aufgebaut werden soll. PostgreSQLs Standardport ist 5432.

Hinweise

Die **PostgreSQLDatabase**-Klasse benötigt das PostgreSQL Datenbank-Plugin. Legen Sie das Plugin in REALbasics Plugin-Verzeichnis. Die REALbasic-CD enthält alle Datenbank-Plugins und Sie finden die jeweils aktuellste Version auf den Internetseiten von REAL Software unter <http://www.realsoftware.com>.

Ein Feld vom Typ Float wird von postgresQL als sogenanntes „Double“ (8 Bytes) implementiert

Die IsPrimary-Spalte des Feldschemas für eine PostgreSQL-Datenbank liefert immer [False](#) zurück.

PostgreSQL Datum: Die Feldtypen Date, Time und TimeStamp werden nun unterstützt. Verwenden Sie die [DateColumn](#) und [DateValue](#)-Methoden für eine Datums-Repräsentation dieser Daten oder verwenden Sie [StringValue](#), um eine für Menschen lesbare Form zu erhalten.

Beispiel

Diese Beispiel öffnet eine existierende PostgreSQL Datenbank:

```
Dim db as PostgreSQLDatabase
db=New PostgreSQLDatabase
```

```

db.host="192.168.1.172"
db.port=5432
db.DatabaseName="myDatabase"
db.Username="Charlie"
db.Password="mashie"
If db.connect then
    //weiter mit Datenbank-Operationen
else
    MsgBox "Die Verbindung konnte nicht hergestellt werden."
end if

```

Siehe auch

[Database](#), [RecordSet](#) Klassen.

Pow Funktion

Liefert die Potenz eines Wertes zurück.

Syntax

Ergebnis = **Pow** (Wert, power)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Wert potenziert mit <i>power</i> .
<i>Wert</i>	<u>Double</u>	Der mit <i>power</i> zu potenzierende Wert.
<i>power</i>	<u>Double</u>	Die Basiszahl, mit der <i>Wert</i> potenziert wird.

Beispiele

Dieses Beispiel verwendet die Funktion **Pow**, um die Potenz aus vier und sieben zu berechnen.

```

Dim d As Double
d=Pow(4,7) //liefert 4 hoch 7=16384

```

PowerPointApplication Klasse

Wird für die Automatisierung von Microsoft PowerPoint verwendet.

Übergeordnete Klasse

[OLEObject](#)

Siehe auch

[Office](#), [ExcelApplication](#), [WordApplication](#), [OLEObject](#) Klassen.

PreferencesFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Ordner „Preferences“ im Systemordner zu bekommen. Unter Windows verweist es, wenn vorhanden, auf das Verzeichnis „C:\Dokumente und Einstellungen\Anwendername\Lokale Einstellungen\Anwendungsdaten\“. Unter Windows 98 greift es auf das Windows-Verzeichnis zu. Unter Linux wird das Home-Verzeichnis des aktuellen Benutzers verwendet.

Syntax

Ergebnis = **PreferencesFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein <u>FolderItem</u> , das den Preferences-Ordner oder das Anwendungsdaten-Verzeichnis repräsentiert.

Hinweis

Verwenden Sie die Funktion **PreferencesFolder** immer dann, wenn Sie Zugriff auf den Preferences-Ordner benötigen.

Wenn Ihre Anwendung Benutzereinstellungen vorsieht, ist es sinnvoll, diese auch in einer Datei im Preferences-Ordner zu sichern. Dies erlaubt dem Anwender, die Anwendung neu zu installieren, ohne die Voreinstellungen zu verlieren.

Die Funktion **PreferencesFolder** bietet eine Möglichkeit, auf die Einträge im Preferences-Ordner zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

Beispiele

Bei der Beschreibung der Funktion AppleMenuFolder finden Sie ein passendes Beispiel.

Siehe auch

AppleMenuFolder, ApplicationSupportFolder, ControlPanelsFolder, DesktopFolder, ExtensionsFolder, FontsFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

PrefsMenuItem Klasse

Ein Menüeintrag, der von der **PrefsMenuItem**-Klasse abgeleitet wird und automatisch in das Applikations-Menü unter Mac OS X verschoben wird (oder in das REALbasic-Menü, wenn man das Programm in der IDE startet). Unter anderen Betriebssystemen bleibt das Menü dort, wo man es im Menü-Editor positioniert hat.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>BalloonHelp</i>	Message als <u>String</u>	Nachrichtentext, der erscheinen soll, wenn die Sprechblasenhilfe eingeschaltet ist und der Anwender den Mauspfel über diesen Menüeintrag bewegt, während der Menüeintrag auswählbar ist.
Bold	<u>Boolean</u>	Gibt an, ob ein Menüeintrag fett dargestellt werden soll.
<i>Checked</i>	<u>Boolean</u>	Gibt an, ob ein Menüeintrag ausgewählt (angekreuzt) dargestellt werden soll.
<i>CommandKey</i>	<u>String</u>	Das Macintosh-Tastaturkürzel für diesen Menüeintrag. Tastaturkürzel für Windows werden durch Verwenden von Konstanten gesetzt. Sehen Sie im Entwicklerhandbuch unter „Verwendung von Konstanten, um Menüs und Menüobjekten Tastaturkürzel hinzuzufügen“.
<i>DisabledBalloonHelp</i>	Message als <u>String</u>	Nachrichtentext, der erscheinen soll, wenn die Sprechblasenhilfe eingeschaltet ist und der Anwender den Mauspfel über diesen Menüeintrag bewegt, während der Menüeintrag nicht auswählbar ist.
<i>Enabled</i>	<u>Boolean</u>	Gibt an, ob dieser Menüeintrag auswählbar (aktiviert) oder nicht auswählbar (deaktiviert) ist. Diese Eigenschaft kann nur aus einem Event-Handler „enableMenuItem“ heraus geändert werden. Sie können einen Menüeintrag entweder aktivieren, indem Sie der Enabled-Eigenschaft True zuweisen oder indem Sie die Enable-Methode aufrufen.
Italic	<u>Boolean</u>	Gibt an, ob ein Menüeintrag kursiv dargestellt werden soll.
Name	<u>String</u>	Der Name des Menüeintrags.
Submenu	<u>Boolean</u>	Zeigt an, daß der Menüeintrag ein Submenü ist.
<i>Tag</i>	<u>String</u>	Ein „versteckter“ Textstring, der mit dem Menüeintrag assoziiert ist. Auf den Tag kann man über Code zugreifen, wenn der Anwender den Menüeintrag anwählt, aber anders als bei der Text-Eigenschaft wird es im Menü nicht angezeigt. Es verhält sich wie die RowTag-Eigenschaft eines PopupMenu-Steuerelements.
<i>Text</i>	<u>String</u>	Der Text des Menüeintrags
Underline	<u>Boolean</u>	Gibt an, ob ein Menüeintrag unterstrichen dargestellt werden soll.
<i>Visible</i>	<u>Boolean</u>	Gibt an, ob ein Menüeintrag sichtbar ist.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Entfernt dynamisch angelegte Menüeinträge.
<i>Enable</i>		Setzt die Enabled-Eigenschaft des Menüeintrags auf True. Rufen Sie Enable nur innerhalb eines EnableMenuItems-Event-Handler auf.

Hinweise

Diese Klasse ist für das Erzeugen des Einstellungen-Menüeintrags für Ihre Applikation gedacht. Erzeugen Sie den Menüeintrag in dem Menü, in dem er in der Windows- und Classic-Version Ihres Programmes erscheinen soll und verwenden Sie dann das Eigenschaften-Fenster, um die übergeordnete Klasse in **PrefsMenuItem** zu ändern. Ansonsten konfigurieren Sie den Menüeintrag ganz normal. In der Mac OS X Version Ihres Programms wird der Menüeintrag dann automatisch in das Applikations-Menü verschoben.

Gemäß den Apple User Interface Guidelines wird als Tastaturkürzel für den **Einstellungen**-Menüeintrag einer Applikation **Apfel-**(Apfel-Komma) empfohlen.

Siehe auch

[MenuItem](#), [QuitMenuItem](#) Klassen.

Print Methode

Wird für Ausgaben im Terminal von [ConsoleApplications](#) verwendet.

Syntax

Print (data)

Teil	Typ	Beschreibung
data	String	Der Text, der ausgegeben werden soll. <i>Print</i> hängt einen Zeilenumbruch an den angegebenen Text an.

Hinweise

Die **Print**-Methode entspricht dem Aufruf der [WriteLine](#)-Methode der [StandardOutputStream](#)-Klasse.

Siehe auch

[StandardInputStream](#), [StandardOutputStream](#), [ConsoleApplication](#) Klassen; [TargetHasGUI](#) Konstante.

PrinterSetup Klasse

Wird verwendet, um die Einstellungen zur Ausgabeseite abzufragen oder neu einzustellen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>Height</i>	Integer	Die Höhe des bedruckbaren Bereichs in Pixeln. Height ergibt sich aus PageHeight minus den Rändern.
<i>HorizontalResolution</i>	Integer	Die horizontale Ausgabeauflösung in Punkten pro Zoll (dpi) – normalerweise ein Drucker.
<i>Left</i>	Integer	Der linke Ursprung des bedruckbaren Bereichs. Sollte immer 0 (null) sein.

Name	Art	Beschreibung
<i>MaxHorizontalResolution</i>	<u>Integer</u>	Setzen Sie diesen Wert auf die maximale horizontale Auflösung, mit der Sie drucken möchten. Der Defaultwert ist 72. Sie können ihn auf die maximale Druckerauflösung einstellen, die Sie verarbeiten können oder auf -1 für die höchstmögliche Auflösung des Ausgabege- räts. Nachdem Sie PageSetupDialog, OpenPrinter oder OpenPrinterDialog aufgerufen haben, wird die HorizontalResolution-Eigenschaft automatisch auf die höchste vom Druckertreiber unterstützte Auflösung innerhalb Ihrer Vorgaben gestellt. Es wird dann in dieser Auflösung gedruckt. Wirkt sich nicht auf vorhandenen Code aus.
<i>MaxVerticalResolution</i>	<u>Integer</u>	Setzen Sie diesen Wert auf die maximale vertikale Auflösung, mit der Sie drucken möchten. Der Defaultwert ist 72. Sie können ihn auf die maximale Druckerauflösung einstellen, die Sie verarbeiten können oder auf -1 für die höchstmögliche Auflösung des Ausgabege- räts. Nachdem Sie PageSetupDialog, OpenPrinter oder OpenPrinterDialog aufgerufen haben, wird die VerticalResolution-Eigenschaft automatisch auf die höchste vom Druckertreiber unterstützte Auflösung innerhalb Ihrer Vorgaben gestellt. Es wird dann in dieser Auflösung gedruckt. Wirkt sich nicht auf vorhandenen Code aus.
PageHeight	<u>Integer</u>	Die absolute Höhe der Seite in Pixeln.
PageLeft	<u>Integer</u>	Der Abstand (in Pixeln) des linken Randes vom linken Rand der tatsächlichen Seite.
PageTop	<u>Integer</u>	Der Abstand (in Pixeln) des oberen Randes zum oberen Rand der tatsächlichen Seite.
PageWidth	<u>Integer</u>	Die Breite (in Pixeln) der tatsächlichen Seite.
<i>SetupString</i>	<u>String</u>	Ein Wert, der alle anderen Eigenschaften repräsentiert. Klickt der Anwender in der „Seite einrichten“-Dialogbox auf „OK“, so wird dieser Wert kopiert. Sie können den Wert speichern, um die Benutzereinstellungen zu sichern. Weisen Sie dieser Eigenschaft einen der gespeicherten Werte zu, so werden dadurch alle Benutzereinstellungen der „Seite einrichten“-Dialogbox wieder hergestellt.
Top	<u>Integer</u>	Der obere Ursprung des bedruckbaren Bereichs. Sollte immer 0 (null) sein.
VerticalResolution	<u>Integer</u>	Die vertikale Ausgabeauflösung in Punkten pro Zoll (dpi).
Width	<u>Integer</u>	Die Breite des bedruckbaren Bereichs in Pixeln. Die Breite ergibt sich aus PageWidth minus den Rändern.

Methoden

Name	Parameter	Beschreibung
<i>PageSetupDialog</i>	[window] als <u>Window</u>	Zeigt die „Seite einrichten“-Standard-Dialogbox an. Wenn Sie vor dem Aufruf von PageSetupDialog der Eigenschaft SetupString benutzerdefinierte Einstellungen zugewiesen haben, werden diese in die „Seite einrichten“-Dialogbox übernommen. Nachdem der Anwender den OK-Button angeklickt hat, werden alle Eigenschaften entsprechend den vom Nutzer vorgenommenen Einstellungen angepaßt. Diese Funktion liefert <u>True</u> , wenn der Anwender „OK“ anklickt und <u>False</u> , wenn der Anwender den Dialog abbricht.

Hinweise

Indem Sie den Funktionen OpenPrinter oder OpenPrinterDialog ein PrinterSetup-Objekt übergeben, erreichen Sie, daß der Drucker die Eigenschaften dieses Objekts für den Ausdruck verwendet. Wenn der Anwender beispielsweise im Feld „Skalierung“ den Wert „200%“ einträgt, wird der Drucker automatisch mit diesem Faktor ausdrucken.

Der „Seite einrichten“-Dialog ist unter Linux nicht verfügbar. Hier wird stattdessen False zurückgegeben und kein Dialog angezeigt.

MaxHorizontalResolution und MaxVerticalResolution

Diese Eigenschaften erlauben es Ihnen, in höheren Auflösungen als 72 dpi zu drucken. Normalerweise müssen Sie die zu druckenden Texte/Grafiken skalieren (und vielleicht die Größe der Steuerelemente) um das gewünschte Resultat zu erzielen. Wenn Sie z.B. Styled Text in einem EditField mittels DrawBlock drucken, werden Sie angemessene Änderungen an den Fontgrößen vornehmen müssen,

um einen WYSIWYG-Ausgabe zu erhalten. Eine Verdoppelung der Auflösung benötigt eine Verdoppelung der Fontgröße; andernfalls erhalten Sie Text, der nur halb so groß wie der Bildschirmfont ist.

Beispiele

Dieses Beispiel ruft die „Seite einrichten“-Dialogbox auf und sichert die vom Benutzer ausgewählten Einstellungen in einer Variable:

```
Dim settings as String
Dim PageSetup as PrinterSetup
PageSetup=New PrinterSetup
If PageSetup.PageSetupDialog Then
    settings=PageSetup.SetupString
End If
```

Dieses Beispiel stellt die Werte aus der Variable „settings“ wieder her und ruft daraufhin die „Seite einrichten“-Dialogbox mit diesen Einstellungen auf:

```
Dim PageSetup as PrinterSetup
PageSetup=New PrinterSetup
PageSetup.SetupString=settings
If PageSetup.PageSetupDialog Then
    settings=PageSetup.SetupString
End If
```

Dieses Beispiel ruft die „Seite einrichten“-Dialogbox auf, übergibt die Einstellungen der Funktion [OpenPrinterDialog](#) und druckt einen Beispieltext aus:

```
Dim g as Graphics
Dim p as PrinterSetup
p=New PrinterSetup
If p.PageSetupDialog then
    g=OpenPrinterDialog(p)
    if g<> Nil then
        g.DrawString "Hallo Welt", 50,50
    End if
End if
```

Dieses Beispiel ruft die Seitenformat-Dialogbox auf und zeigt danach die Seitengröße, den druckbaren Bereich und die Ränder in StaticText-Steuerelementen an. Die Ergebnisse hängen natürlich vom Seitenformat ab, das der Anwender einstellt. Da PageLeft und PageTop die horizontalen und vertikalen Ränder gemessen vom druckbaren Bereich sind und nicht von den Kanten der Seite, sind diese negativ.

```
Dim settings as String
Dim p as PrinterSetup
p=New PrinterSetup
If p.PageSetupDialog Then
    settings=p.SetupString
End If
staticText1.text="PageLeft="+str(p.pageLeft)
staticText2.text="PageTop="+str(p.pagetop)
staticText3.text="PageHeight="+str(p.pageheight)
staticText4.text="PageWidth="+str(p.pagewidth)
staticText5.text="Height="+str(p.Height)
staticText6.text="Width="+str(p.width)
staticText7.text="Computed height="+str(p.height-2*p.pagetop)
staticText8.text="Computed width="+str(p.width-2*p.pageleft)
```

Siehe auch

[Graphics](#) Klasse; [OpenPrinter](#); [OpenPrinterDialog](#) Funktionen, [StyledTextPrinter](#) Klasse.

ProgressBar Steuerelement



Zeigt einen Fortschrittsbalken (ProgressBar) an.

Übergeordnete Klasse

[RectControl](#)

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>Maximum</i>	<u>Integer</u>	Wert von Value, wenn der ProgressBar ganz am rechten Rand angekommen ist. Der Maximalwert von <i>Maximum</i> ist 65536.
<i>Value</i>	<u>Integer</u>	Die gegenwärtige Position der Anzeige.

Events

Name	Parameter	Beschreibung
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der ProgressBar an der Position x,y gedrückt. Liefert ein Boolean. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen.
<i>MouseDownDrag</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der ProgressBar gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseDownUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb der ProgressBar an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.

Hinweis

Ist vorher nicht genau zu bestimmen, wie lange ein Prozeß dauern wird, macht man von einem geringelten Streifen Gebrauch. Warum dieser geringelte Streifen im Englischen „Barber Pole“ genannt wird, erfahren Sie im Entwicklerleitfaden.



Beispiel

Definiert den Maximalwert des ProgressBar:
`ProgressBar1.maximum=350`

Siehe auch

RectControl Klasse.

PushButton Steuerelement



Der Pushbutton.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>Bold</i>	<u>Boolean</u>	Stellt den Buttontext fett dar.
<i>Cancel</i>	<u>Boolean</u>	Wenn <u>True</u> , lösen Escape-Taste oder die Tastenkombination Befehlstaste-Punkt diesen Button aus.
<i>Caption</i>	<u>String</u>	Der Buttontext.
<i>Default</i>	<u>Boolean</u>	Wenn <u>True</u> , so wird der Button als Defaultbutton gekennzeichnet und sowohl Return- als auch Enter-Taste lösen diesen Button aus.

Name	Art	Beschreibung
<i>Italic</i>	<u>Boolean</u>	Stellt den Buttontext kursiv dar.
<i>TextFont</i>	<u>String</u>	Name der für den Buttontext zu verwendenden Schrift.
<i>TextSize</i>	<u>Integer</u>	Größe der für den Buttontext zu verwendenden Schrift.
<i>Underline</i>	<u>Boolean</u>	Stellt den Buttontext unterstrichen dar.

Methoden

Name	Beschreibung
<i>Push</i>	Bewirkt, daß der Knopf sichtbar gedrückt wird, so wie wenn das Action-Event ausgelöst wird. Rufen Sie Push auf, wenn der Anwender die entsprechende Taste auf der Tastatur gedrückt hat, um ihm eine passende visuelle Rückmeldung zu liefern.
<i>SetFocus</i>	Setzt den Fokus auf den PushButton (Windows und Linux).

Events

Name	Parameter	Beschreibung
<i>Action</i>		Der Button wurde ausgelöst.
<i>GotFocus</i>		(Windows und Linux) Das Steuerelement hat den Fokus bekommen und besitzt einen Selektionsrahmen um die Beschriftung.
<i>LostFocus</i>		(Windows und Linux) Das Steuerelement hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y gedrückt. Wenn Sie MouseDown auswerten möchten, liefern Sie <u>True</u> zurück. Das Action-Event wird nicht ausgeführt und der Status des Objektes ändert sich nicht.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht <u>True</u> im MouseDown-Event liefern. Der Rückgabewert wird ignoriert.

Hinweis

Wenn die Caption-Eigenschaft ein Und-Zeichen enthält, wird dieses nur angezeigt, wenn ein weiteres Und-Zeichen folgt. Dies ist so, damit sich Macintosh- und Windows-Programme gleich verhalten.

Die Caption-Eigenschaft des PushButtons kann eine Beschriftung anzeigen, die eine Kodierung verwendet, die nicht zum Regionalcode der Applikation paßt (oder, wenn das Programm in der IDE läuft, nicht zum Regionalcode der IDE). Zum Beispiel kann eine englische Applikation eine japanische Beschriftung einstellen, sofern sie zuerst den TextFont des Knopfes auf Osaka setzt.

Siehe auch

BevelButton Steuerelement, RectControl Klasse.

QuickTime Objekt

Bietet Zugriff auf bestimmte Eigenschaften der QuickTime-Software, die auf dem System des Anwenders installiert ist. Zugriff hat man nur über das System-Objekt.

Eigenschaften

Name	Typ	Beschreibung
Installed	<u>Boolean</u>	Liefert <u>True</u> , falls QuickTime auf dem Rechner des Anwenders installiert ist.

Beispiel

Dieses Beispiel stellt fest, ob QuickTime installiert ist.

```
If System.QuickTime.Installed then
  Beep
  MsgBox "QuickTime ist installiert!"
End if
```

Siehe auchSystem Objekt**QT3DAudio Klasse**

Ein Container zum Speichern von 3D-Audio-Einstellungen. Die Eigenschaften dieser Klasse erlauben es Ihnen, solche Einstellungen zu sichern.

Übergeordnete KlasseObject**Eigenschaften**

Name	Typ	Beschreibung
<i>ConeAngleCos</i>	<u>Double</u>	Kosinus (angle/2) des Dämpfungskegels in Grad. Der Bereich reicht von Null bis 360.
<i>ConeAttenuation</i>	<u>Double</u>	Wert der Dämpfung (in Dezibel) außerhalb des Kegels.
<i>CPULoad</i>	<u>Boolean</u>	CPU-Auslastung gegen Qualität: Null bedeutet beste Qualität.
<i>Humidity</i>	<u>Double</u>	Die Feuchtigkeit (Prozent), wenn Luft das Medium ist. Der Bereich reicht von Null bis 100.
<i>LocationAzimuth</i>	<u>Double</u>	Der Wert reicht von -1.57 bis 1.57.
<i>LocationDistance</i>	<u>Double</u>	Die Einheit ist Meter und der Bereich reicht von 1.0 bis 10.0.
<i>LocationElevation</i>	<u>Double</u>	Die Einheit ist Meter und der Bereich reicht von -3.14 bis 3.14.
<i>LocationListenerVelocity</i>	<u>Double</u>	Die Einheit ist Meter/Sekunde. Der Minimalwert ist Null.
<i>LocationProjectionAngle</i>	<u>Double</u>	Die Einheit ist Grad und der Bereich reicht von Null bis 180.
<i>LocationSourceVelocity</i>	<u>Double</u>	Die Einheit ist Meter/Sekunde. Der Minimalwert ist Null.
<i>Medium</i>	<u>Integer</u>	Medium für die Sound-Ausbreitung: 0 — Luft, 1 — Wasser.
<i>ReferenceDistance</i>	<u>Double</u>	Nominaler Abstand für Aufnahme. Die Einheit ist Meter und der Minimal-Abstand ist Null.
<i>ReverbAttenuation</i>	<u>Double</u>	Nachhall-Modell: Mix-Level. Die Einheit ist Dezibel.
<i>RoomReflectivity</i>	<u>Double</u>	Nachhall-Modell: Reflexionsabschwächung. Die Einheit ist Dezibel.
<i>RoomSize</i>	<u>Double</u>	Nachhall-Modell: Abstand zwischen den Wänden. Die Einheit ist Meter und der Minimalwert ist Null.
<i>Source.Mode</i>	<u>Integer</u>	Art der anzuwendenden Filterung. 0 — ungefiltert, 1 — lokal 2 — Umgebung, 3 — Binaural.

Hinweise

Die QT3DAudio-Klasse ist dazu gedacht, Tonspuren räumlichen Klang (Spezialeffekte) hinzuzufügen. Fügen Sie einem QTSoundTrack das QT3DAudio-Objekt mit der QTSoundTrack.Set3D-Methode hinzu.

Die Eigenschaften dieser Klasse erlauben es, QuickTime-Parameter einzustellen, die unter folgender URL dokumentiert sind:

<http://developer.apple.com/techpubs/quicktime/qtdevdocs/APIREF/SOURCESIV/ssplocalizationdata.htm>

Siehe auchEditableMovie, QTSoundTrack Klassen.

QTEffect Klasse

Ein QuickTime-Effekt, erzeugt mit [GetQTSMPTEEffect](#) oder [GETQTCrossFadeEffect](#) unter Verwendung der [QTEffectSequence](#) Instanz. Siehe Beispiel für die [QTEffectSequence](#) Klasse. Die QTEffect Klasse hat keine Eigenschaften, Methoden oder Events.

QTEffectSequence Klasse

Baut ein QuickTime-Movie aus zwei Bildern unter Verwendung eines QuickTime-Effekts zusammen.

Verwenden Sie QTEffectSequence als Constructor, um ein QTEffectSequence-Objekt zu erzeugen.

Übergeordnete Klasse

[Object](#)

Syntax

Ergebnis = [New](#) **QTEffectSequence**(effect,image1,image2,frames)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	QTEffectSequence	QTEffectSequence-Objekt, erzeugt aus den Werten der übergebenen Parameter.
<i>effect</i>	QTEffect	QT-Effekt, der zum Erzeugen der Sequenz verwendet wird.
<i>image1</i>	Picture	Ausgangsbild der Sequenz.
<i>image2</i>	Picture	Endbild der Sequenz.
<i>frames</i>	Integer	Anzahl der Frames in der Sequenz.

Hinweis

Verwenden Sie [GetQTSMPTEEffect](#) oder [GETQTCrossFadeEffect](#), um den Wert von *effect* zu erhalten.

Eigenschaften

Eigenschaft	Typ	Beschreibung
<i>Image</i>	Picture	Ein Bild, das einem einzelnen Frame der QT-Sequenz entspricht.
<i>Frame</i>	Integer	Frame der QT-Sequenz.

Beispiel

Dieses Beispiel erzeugt ein QuickTime-Movie aus zwei Bildern p1 und p2 unter Verwendung des cross-fade-Effekts.

```
Dim sequence as QTEffectSequence
Dim theEffect as QTEffect
theEffect=GetQTCrossFadeEffect
sequence=New QTEffectSequence (theEffect,p1,p2,96)
```

Siehe auch

[EditableMovie](#), [QTTrack](#), [QTEffect](#), [QTEffectSequence](#) Klassen; [GetQTCrossFadeEffect](#), [GetQTSMPTEEffect](#) Funktion.

QTGraphicsExporter Klasse

Wird verwendet, um Grafiken zu exportieren.

Übergeordnete Klasse

[Object](#)

Eigenschaften

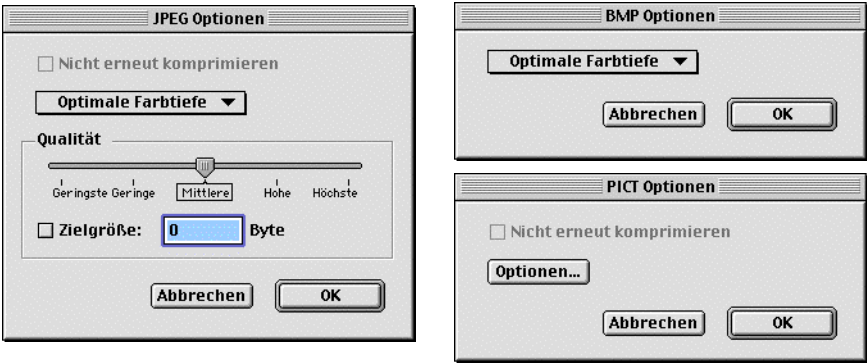
Eigenschaft	Typ	Beschreibung
<i>CompressionQuality</i>	<u>Integer</u>	0 = Minimalqualität 256 = Niedrige Qualität 512 = Normale Qualität 768 = Hohe Qualität 1023 = Maximalqualität 1024 = Verlustlos
<i>DefaultExtension</i>	<u>String</u>	Vorgabe-Dateiendung.
<i>DesiredTargetDataSize</i>	<u>Integer</u>	Ziel-Datengröße in Bytes.
<i>HasSettingsDialog</i>	<u>Boolean</u>	True, wenn das Exporter-Objekt einen Einstellungsdialog besitzt, der von RequestSettings angezeigt werden kann.
<i>OutPutFileCreator</i>	<u>String</u>	Macintosh-Datei-Creator
<i>OutputFileType</i>	<u>String</u>	Macintosh-Datei-Type
<i>RequestSettings</i>	<u>Boolean</u>	Zeigt eine Dialogbox. Siehe Hinweise unten.
<i>SettingsDescription</i>	<u>String</u>	Beschreibung der Einstellungen.

Methoden

Name	Parameter	Beschreibung
<i>SavePicture</i>	Ort als <u>FolderItem</u> , Bild als <u>Picture</u>	Speichert das Bild. Liefert ein <u>Boolean</u> .

Hinweise

Die RequestSettings Dialogbox bietet verschiedene Möglichkeiten für verschiedene Dateitypen. Hier ein paar Beispiele für RequestSettings-Dialogboxen:



Beispiel

Folgendes Beispiel exportiert eine Grafik namens „lois“, die zuvor ins Projektfenster gezogen wurde. Sie können entweder die Eigenschaften des Exporter-Objekts angeben (wie gezeigt) oder eine RequestSettings-Dialogbox anzeigen.

```
Dim f as FolderItem
Dim exporter as QTGraphicsExporter
Dim mybool as Boolean
f=GetFolderItem("Macintosh-HD")
exporter=GetQTGraphicsExporter("JPEG")
//mybool=exporter.RequestSettings
exporter.DesiredTargetDataSize=36
exporter.CompressionQuality=512
exporter.OutputFileType="JPEG"
exporter.OutputFileCreator="ogle"
mybool=exporter.SavePicture(f,lois)
```


Siehe auch

[GetQTGraphicsExporter](#) Funktion.

QTSoundTrack Klasse

Ermöglicht es, einem [EditableMovie](#) Sound hinzuzufügen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>Balance</i>	Integer	Ermittelt oder setzt die Links-Rechts-Balance des QTSoundTrack . Der Bereich ist ± 128 .
<i>Bass</i>	Integer	Ermittelt oder setzt das Bass-Level des QTSoundTrack . Der Bereich ist ± 255 .
<i>Compressor</i>	String	Ermittelt die Kompressionsart, die vom QTSoundTrack verwendet wurde. Der Rückgabewert ist ein Mac OS API OSType. Z.B. "raw " oder "MAC3". Verwenden Sie diese Eigenschaft, um QTSoundTracks zu vergleichen, um zu vermeiden, daß verschiedene Kompressions-Codecs im gleichen QTSoundTrack kombiniert werden, wenn Sie Sound ein- oder anfügen.
<i>SampleRate</i>	Integer	Hiermit ermitteln Sie die Sampling-Rate des QTSoundTrack . Verwenden Sie diese Eigenschaft, um QTSoundTracks zu vergleichen, um zu vermeiden, daß verschiedene Sampling-Raten im gleichen QTSoundTrack kombiniert werden, wenn Sie Sound ein- oder anfügen.
<i>SampleSize</i>	Integer	Ermittelt die Sample-Größe (Bit-Tiefe) des QTSoundTrack . Verwenden Sie diese Eigenschaft, um QTSoundTracks zu vergleichen, um zu vermeiden, daß verschiedene Sampling-Größen im gleichen QTSoundTrack kombiniert werden, wenn Sie Sound ein- oder anfügen.
<i>Stereo</i>	Boolean	Ermittelt, ob der QTSoundTrack in Stereo vorliegt. True bedeutet Stereo. Verwenden Sie diese Eigenschaft, um QTSoundTracks zu vergleichen.
<i>Treble</i>	Integer	Wird zum Ermitteln oder Setzen der Höhen des QTSoundTrack verwendet. Der Bereich ist ± 255 .
<i>Volume</i>	Integer	Wird zum Ermitteln oder Setzen der Lautstärke des QTSoundTrack verwendet. Der Bereich reicht von 0 bis 512.

Methoden

Name	Parameter	Beschreibung
<i>AppendSoundTrackSegment</i>	SourceTrack als QTSoundTrack , SourcePosition als Double , SourceDuration als Double , CopyMedia als Boolean , ShowProgress als Boolean	Fügt ein Segment aus einem anderen QTSoundTrack an das Ende des aufrufenden QTSoundTrack an. <i>SourcePosition</i> (in Sekunden) ist der Anfang des Segements in SourceTrack, der an den aufrufenden QTSoundTrack angehängt wird. <i>SourceDuration</i> (in Sekunden) ist die Länge des Segments in <i>SourceTrack</i> die an das aufrufende QTSoundTrack angehängt wird. Falls <i>CopyMedia</i> True ist, wird der Inhalt in <i>SourceTrack</i> an den Zieltrack kopiert; andernfalls wird eine Referenz auf SourceTrack verwendet. Falls <i>ShowProgress</i> True ist und die Operation lange dauert, zeigt QuickTime eine modale Dialogbox mit einem Fortschrittsbalken an, während die Operation voranschreitet.

Name	Parameter	Beschreibung
<i>Get3D</i>		Erhält eine Referenz auf das <u>QT3DAudio</u> -Objekt in dem Film. Liefert ein <u>QT3DAudio</u> -Objekt.
<i>InsertEmptySoundTrackSegment</i>	ThePosition als <u>Double</u> , TheDuration als <u>Double</u>	Fügt leeren Raum in ein QTSoundTrack ein, das bereits Inhalt besitzt. Die Parameter <i>ThePosition</i> und <i>TheDuration</i> legen in Sekunden den Startpunkt und die Dauer des leeren einzufügenden Segments fest. Diese Methode hat keinen Effekt bei einem QTSoundTrack das keinen Inhalt besitzt.
<i>InsertSoundTrackSegment</i>	SourceTrack als QTSoundTrack , SourcePosition als <u>Double</u> , SourceDuration als <u>Double</u> , destinationPosition als <u>Double</u> , copyMedia als <u>Boolean</u> , ShowProgress als <u>Boolean</u>	Fügt ein Segment aus einem anderen QTSoundTrack in den aufrufenden QTSoundTrack ein. SourcePosition ist die Startposition (in Sekunden) des einzufügenden Segments in SourceTrack; SourceDuration ist die Länge (in Sekunden) des Segments; DestinationPosition ist der Startpunkt (in Sekunden) im aufrufenden QTSoundTrack, an dem <i>SourceTrack</i> eingefügt wird. Falls <i>CopyMedia</i> <u>True</u> ist, wird der Inhalt von <i>SourceTrack</i> in den Zieltrack kopiert; andernfalls wird eine Referenz auf SourceTrack verwendet. Falls <i>ShowProgress</i> <u>True</u> ist und die Operation lange dauert, zeigt QuickTime eine modale Dialogbox mit einem Fortschrittsbalken an, während die Operation voranschreitet.
<i>Set3D</i>	3DAudio als <u>QT3DAudio</u>	Stellt die Audio-Lokalisierung eines QTSoundTrack ein.

QTTrack Klasse

Ein QuickTime-Track, erzeugt aus einem EditableMovie.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
Duration	<u>Double</u>	Dauer des Tracks in Sekunden.
Enabled	<u>Boolean</u>	True, falls Enabled. Ein Track, der disabled ist, wird nicht angezeigt (falls Video) oder ist nicht zu hören (falls Audio).
MediaType	<u>String</u>	4-buchstabiger Medien-Typ (soun, vide).
TimeDuration	<u>Integer</u>	Dauer des Films in TimeScale-Einheiten.
TimeScale	<u>Integer</u>	Definiert die Zeitbasis des Tracks in 1/TimeScale-Einheiten pro Sekunde.
TrackID	<u>Integer</u>	ID des Tracks.

Methoden

Name	Parameter	Beschreibung
DataSize	StartTime als <u>Integer</u> DurationTime als <u>Integer</u>	Liefert einen Integer, die Anzahl der Bytes im angegebenen Segment, festgelegt durch StartTime und DurationTime.
<i>Delete</i>		Löscht den QTTrack .
<i>DeleteTrackSegment</i>	StartTime als <u>Double</u> Duration als <u>Double</u>	Löscht ein Segment aus einem QTTrack . <i>StartTime</i> ist der Anfang des Segments, das gelöscht werden soll (in Sekunden) und <i>Duration</i> ist die Länge (in Sekunden) des Segments.

Name	Parameter	Beschreibung
<i>FindFirstSample</i>		Sucht das erste Sample. Im Falle eines Videotracks ein Frame.
<i>FindNextSample</i>		Sucht nächstes Sample (Frame im Falle eines Video-Tracks).
<i>FindFirstKeyFrame</i>		Sucht erstes Key-Frame (Basisbild, aus dem die Differenzen errechnet wurden).
<i>FindNextKeyFrame</i>		Sucht nächstes Key-Frame.

Hinweise

Der *TimeScale*-Parameter legt die Anzahl von Zeiteinheiten fest, die jede Sekunde durchlaufen. Zum Beispiel setzt ein *TimeScale*-Wert von 60 die Zeitbasis auf sechzigstel Sekunden. Der Parameter *TimeDuration* wird in *TimeScale*-Einheiten angegeben.

Siehe auch

[EditableMovie](#), [QTVideoTrack](#) Klassen.

QTUserData Klasse

Dient zum Ermitteln und Setzen von QuickTime-Anwenderdaten, wie Copyright-Hinweise, Credits und andere.

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
<i>AddUserData</i>	udType als String Wert als String	Fügt ein User-Data-Objekt vom Typ <i>udType</i> der User-Daten-Liste hinzu.
<i>GetUserData</i>	udType als String Index als Integer ByRef Wert als String	Holt User-Daten vom Typ <i>udType</i> an der durch Index übergebenen Position. Nach erfolgreichem Aufruf sind die User-Daten in <i>Wert</i> gespeichert. Liefert ein Boolean .
<i>GetUserDataText</i>	udType als String Index als Integer ByRef Wert als String Region als Integer	Holt User-Daten vom Typ <i>udType</i> für <i>Region</i> . Nach erfolgreichem Aufruf sind die User-Daten in <i>Wert</i> gespeichert. Liefert ein Boolean .
<i>SetUserDataText</i>	udType als String Index als Integer Value als String Region als Integer	Fügt ein User-Data-Objekt an der durch <i>Index</i> angegebenen Position vom Typ <i>udType</i> für <i>Region</i> hinzu.
<i>UserDataCount</i>	udType als String	Anzahl von Objekten in <i>udType</i> . Liefert einen Integer .

Hinweise

Ein QuickTime-Film kann durch den *udType*-Parameter bezeichnete User-Daten-Listen enthalten. *UserData* enthält zusätzliche Informationen über den Film, inklusive Copyright-Informationen etc. Die *QTUserData*-Klasse wird verwendet, um die User-Daten-Listen zu verwalten. *UserData* ist eine Eigenschaft von [EditableMovie](#).

Der *udType*-Parameter ist ein vierbuchstabiger Code für den Typ von User-Daten. Apples derzeit definierten Daten-Typen zeigt folgende Tabelle:

udType	Beschreibung
©nam	Name des Films.
©cpy	Copyright.
©day	Datum, an dem der Film erzeugt wurde.
©dir	Name des Regisseurs des Films.
©ed1 bis ©ed9	Änderungsdaten und Beschreibungen.
©fmt	Hinweise auf Filmformat.

udType	Beschreibung
©inf	Informationen über den Film.
©prd	Produzent des Films.
©prf	Name des Darstellers.
©req	Besondere Hard- und Softwarevoraussetzungen.
©src	Credits für die, die Filmmaterial zur Verfügung gestellt haben.
©wrt	Name des Drehbuchautors.

Apple hat alle kleinbuchstabigen Datentyp-Werte reserviert; Sie können jedoch User-Daten-Werte unter Verwendung von Großbuchstaben erzeugen. Apple empfiehlt, Type-Werte zu verwenden, die mit einem © beginnen, um User-Daten-Objekte festzulegen, die Text-Daten speichern.

Es ist möglich, mehr als ein User-Daten-Objekt in einer User-Daten-Lsite zu erzeugen. Daher wird auf jedes User-Daten-Objekt über den Index-Parameter verwiesen, wobei das erste Objekt als 1 numeriert ist. Bitte schauen Sie im [Beispiel](#) für mehrfache Objekte innerhalb eines Daten-Typs nach.

Sie können ebenso einen alternativen Text für ein vorhandenes User-Daten-Objekt anbieten, z.B., um mehrere Sprachen zu unterstützen. Die alternative Version wird durch den Wert des *Region*-Parameters angezeigt. Er repräsentiert einen Länder-Code. Das Beispiel zeigt englische und französische Titel für einen Film, indem es den *Region*-Parameter verwendet.

Beispiele

Folgender Code fügt verschiedene Darsteller und einen Regisseur dem Film hinzu und zeigt dann die Namen der Darsteller in einer [ListBox](#) und den Regisseur in einem [EditField](#). Es fügt außerdem englische und französische Titel dem Film hinzu.

```
Dim mybool as Boolean
Dim myval as String
Dim i, myint as Integer
Dim f as FolderItem
Dim m as EditableMovie
f=GetFolderItem("FrmAmngTheDead.Mov")
If f <> Nil then
    m=f.OpenEditableMovie
    //Darsteller
    m.UserData.AddUserData("@prf","Kim Novak")
    m.UserData.AddUserData("@prf","James Stewart")
    m.UserData.AddUserData("@prf","Barbara Bel Geddes")
    m.UserData.AddUserData("@prf","Tom Helmore")
    //Anzahl der Darsteller
    MyInt=m.UserData.UserDataCount("@prf")
    //Der Regisseur...
    m.UserData.AddUserData("@dir","Alfred Hitchcock")
    //English and French titles
    m.UserData.SetUserDataText("@nam",1,"Vertigo",1)
    m.UserData.SetUserDataText("@nam",1,"D'Entre les Morts",2)
    //zeige Namen des Regisseurs
    mybool=m.UserData.GetUserData("@dir",1,myval)
    EditField1.text=myval
    //zeige Darsteller in ListBox an
    For i=1 to MyInt
        mybool=m.UserData.GetUserData("@prf",i,myval) ListBox1.AddRow(myval)
    Next
end if
```

Siehe auch

[EditableMovie](#) Klasse.

QTVideoTrack Klasse

Ein QuickTime-Videotrack.

Übergeordnete Klasse

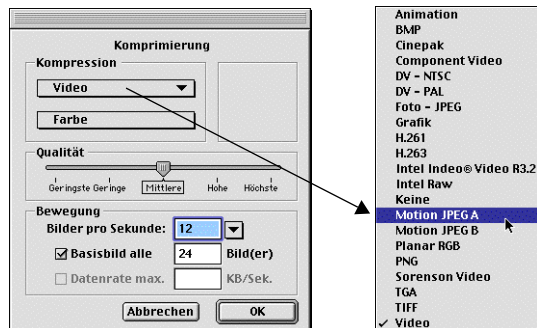
[QTTrack](#)

Methoden

Name	Parameter	Beschreibung
<i>AppendPicture</i>	Pic als <u>Picture</u>	Bild, das an einen Videotrack angehängt werden soll. Wird verwendet, um ein Movie Frame für Frame zu erzeugen.
<i>AppendVideoTrackSegment</i>	SourceTrack als QTVideoTrack , SourcePosition als Double, SourceDuration als Double, CopyMedia als Boolean, ShowProgress als Boolean	Fügt ein Segment aus einem SourceTrack in den aufrufenden QTVideoTrack ein. <i>SourcePosition</i> ist der Startpunkt oder das Segment (in Sekunden) in <i>SourceTrack</i> und <i>SourceDuration</i> ist die Länge (in Sekunden) des Segments. Falls <i>CopyMedia</i> True ist, wird der Inhalt in SourceTrack in den Zieltrack kopiert; andernfalls wird eine Referenz auf <i>SourceTrack</i> verwendet. Falls <i>ShowProgress</i> <u>True</u> ist und die Operation lange dauert, zeigt QuickTime eine modale Dialogbox mit einem Fortschrittsbalken an, während die Operation voranschreitet.
<i>SelectCompressionSettings</i>		Ruft die Standard-QT-Kompressionseinstellungen-Dialogbox auf. Liefert ein <u>Boolean</u> , das <u>True</u> ist, wenn der Anwender auf OK geklickt hat.

Hinweis

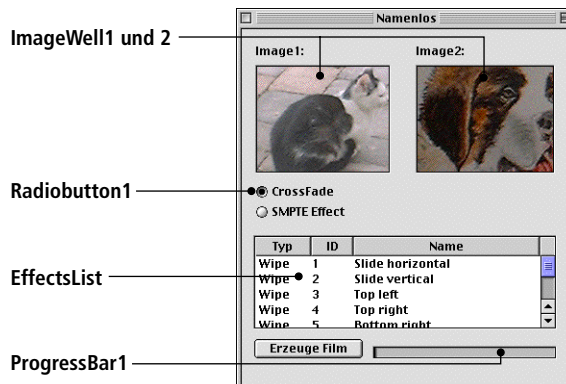
Die SelectCompressionSettings-Methode zeigt folgende Dialogbox:



Bei einigen Video-Formaten ist noch ein Optionen-Knopf unter dem Qualität-Slider vorhanden.

Beispiel

Folgendes Beispiel erzeugt einen QuickTime-Film aus zwei Bildern und einem vom Anwender gewählten Überblendeffekt. Das Interface für dieses Beispiel sieht folgendermaßen aus:



```
Dim sequence as QTEffectSequence
Dim i,effectID as Integer
Dim theEffect as QTEffect
Dim track as QTVideoTrack
Dim m as EditableMovie
Dim f as FolderItem
Dim mybool as Boolean
f=GetFolderItem("Movie")
m=f.CreateMovie //erzeuge leeren Film
If radiobutton1.value then //ermittle gewählten Effekt
    theEffect=GetQTCrossFadeEffect
else
    theEffect=GetQTSMPTEffect(val(EffectsList.cell(EffectsList.ListIndex,1)))
end if
sequence=New QTEffectSequence (theEffect,ImageWell1.Image, ImageWell2.Image,96)
track=m.NewVideoTrack(ImageWell1.Image.width, ImageWell1.Image.height, 32)
mybool=track.SelectCompressionSettings
For i=1 to 96
    sequence.frame=i
    track.appendpicture sequence.image
    progressBar1.value=i
Next
f.launch //starte Movieplayer
```

Siehe auch

FolderItem, EditableMovie, QTTrack, QTEffect, QTEffectSequence Klassen; GetCrossFadeEffect, GetQTSMPTEffect Funktionen.

Quaternion Klasse

Ein 3D-Objekt, das die Ausrichtung oder Rotation in einer RB3dSpace-Welt repräsentiert.

Eigenschaften

Name	Art	Beschreibung
W	<u>Double</u>	Die W-Komponente des Quaternion.
X	<u>Double</u>	Die X-Komponente des Quaternion.
Y	<u>Double</u>	Die Y-Komponente des Quaternion.
Z	<u>Double</u>	Die Z-Komponente des Quaternion.

Methoden

Name	Parameter	Beschreibung
<i>Inverse</i>		Liefert das Gegenteil des Quaternion als Quaternion zurück.
<i>Invert</i>		Dreht das Quaternion um, so daß es in die entgegengesetzte Richtung weist.
<i>MultiplyBy</i>	quat als Quaternion	Multipliziert dieses Quaternion mit einem anderen Quaternion und aktualisiert dieses Quaternion. Das gleiche wie die Time-Methode, außer daß es genau dieses Quaternion aktualisiert und nicht das Resultat als Quaternion liefert.
<i>Normalize</i>		Skaliert das Quaternion auf die Standardgröße.
<i>SetBetween</i>	quat1 als Quaternion, quat2 als Quaternion, position als <u>Double</u>	Setzt dieses Quaternion auf eine Kombination von <i>quat1</i> und <i>quat2</i> .
<i>SetRotateAbout Axis</i>	AchseX als <u>Double</u> , AchseY als <u>Double</u> , AchseZ als <u>Double</u> , Winkel als <u>Double</u>	x, y und z-Koordinaten und Winkel (im Bogenmaß) des Objekts. Wird verwendet, um das Objekt in einem bestimmten Winkel um die Achsen zu rotieren.
<i>Times</i>	quat als Quaternion	Multipliziert <i>quat</i> mit einem anderen Quaternion und liefert das Produkt als ein Quaternion zurück.

Name	Parameter	Beschreibung
<i>Transform</i>	Vektor als Vector3D	Liefert das Ergebnis einer Rotation <i>Vektor</i> durch dieses Quaternion als Vector3D zurück.

Hinweise

Ein **Quaternion** ist eine kompakte, effiziente Art, Ausrichtung oder Rotation im dreidimensionalen Raum zu beschreiben. Es vollbringt die gleichen Aufgaben für die Ausrichtung und Rotation wie Vektoren dies für Position und Bewegung machen. Jede mögliche Rotation kann als ein Winkel um eine bestimmte Achse beschrieben werden. Ein **Quaternion** ist im Wesentlichen eine Achse plus ein Winkel, obwohl die interne Repräsentation aus mathematischen Gründen geringfügig davon abweicht.

So wie ein Vektor zur Repräsentation entweder einer Position oder Verschiebung verwendet werden kann, kann ein **Quaternion** zur Repräsentation einer Ausrichtung oder einer Rotation verwendet werden. Um absolute Ausrichtung zu repräsentieren, legen Sie einfach das Quaternion als eine Rotation relativ zu einer Null- oder Standard-Ausrichtung fest.

Das Schöne an Quaternions ist, daß diese auf einfache Weise kombiniert werden können. Wenn Sie Quaternion A mit Quaternion B multiplizieren ist das Ergebnis ein Quaternion namens C. Eine Rotation C positioniert das Objekt in die gleiche Ausrichtung wie wenn man die Rotation A gefolgt von der Rotation B ausführen würde. Das Multiplizieren von Quaternions ist exakt das gleiche wie das Zusammensetzen von Rotationen.

Damit ein Quaternion eine bestimmte Rotation repräsentiert, verwenden Sie die SetRotateAbout Axis-Methode und übergeben dieser die Achse und den Winkel (im Bogenmaß), im dem rotiert werden soll. Wenn Sie z.B. ein Quaternion möchten, das 90 Grad um die X-Achse rotiert, verwenden Sie

```
q.SetRotateAboutAxis 1,0,0, Pi/2
```

Einen Vektor rotieren: Verwenden Sie die Transform-Methode eines **Quaternion**. Geben Sie diesem einen Vektor und als Ergebnis erhalten Sie einen neuen Vektor, rotiert um den Ursprung (0,0,0).

Zwischen zwei Quaternions interpolieren: Dies wird verwendet, wenn Sie das Quaternion als die Repräsentation einer Ausrichtung ansehen. Sie haben q1, welches das Objekt in einer Ausrichtung und q2, welches das Objekt in einer anderen Ausrichtung repräsentiert. Nun möchten Sie das Objekt glatt zwischen q1 und q2 rotieren. Setzen Sie dazu die Ausrichtung des Objekts auf etwas zwischen q1 und q2, indem Sie die SetBetween-Methode einsetzen. Der letzte Parameter ist die Position zwischen diesen beiden, wobei 0,0 q1 entspricht und 1,0 q2. 0,5 wäre exakt die Position zwischen beiden.

Eine gängige Anwendung ist das Aktualisieren der Ausrichtung eines Objektes durch eine Rotation. Wenn man z.B. annimmt, Object3D hätte keine Yaw-Methode eingebaut, könnte man das gleiche folgendermaßen selbst erledigen:

```
Dim yawQuat as Quaternion
yawQuat = New Quaternion
yawQuat.SetRotateAboutAxis 0,1,0, yawAngle // "yaw" -rotiert um die Y-Achse
obj.orientation = yawQuat.Times(obj.orientation)
```

Beachten Sie, daß beim Multiplizieren von Quaternions die Reihenfolge eine Rolle spielt. A.Times(B) ist nicht das gleiche wie B.Times(A); dies ist bei Rotationen im dreidimensionalen Raum so.

Beachten Sie außerdem, daß Quaternions auch eine Funktion namens MultiplyBy besitzen; **Q1=Q1.Times(Q2)** kann effektiver als **Q1.MultiplyBy Q2** ausgedrückt werden. Dies hilft im obigen Beispiel jedoch nicht, da **Q2=Q1.Times(Q2)** benötigt wird, um die korrekte Reihenfolge zu bekommen. Da mit der aktuellen Ausrichtung des Objektes gestartet werden soll, muß dieses rechts erscheinen.

Interpolation

Verwenden Sie die SetBetween-Methode, um zwischen zwei Quaternions mit einem Interpolationsfaktor zu interpolieren. 0.0 paßt exakt auf das erste Quaternion, 1.0 exakt auf das zweite und alle Werte dazwischen zwischen diese beiden Position:

```
Q.SetBetween Q1, Q2, 0.25 // setzt Q auf 25% von Q1 plus 75% von Q2
```

Dies ist eine sehr nützliche Technik. Mit dieser ist es möglich, sanfte Animationen zwischen der Ausrichtung, die Sie haben und der, die Sie wollen zu produzieren. Wenn Sie z.B. zwischen zwei Positionen und Ausrichtungen (vordefiniert oder errechnet) eines Objektes in 11 Schritten interpolieren möchten, können Sie dies folgendermaßen erreichen:

```
For i = 0 to 10
    obj.position.x = position1.x * i/10 + position2.x * (1 - i/10)
    obj.position.y = position1.y * i/10 + position2.y * (1 - i/10)
    obj.position.z = position1.z * i/10 + position2.z * (1 - i/10)
    obj.orientation.SetBetween orientation1, orientation2, i/10
```

```
Rb3DSpace1.Update
next
```

Siehe auch

[Vector3D](#), [Object3D](#), [Group3D](#) Klassen; [RB3DSpace](#) Steuerelement.

Quit Methode

Beendet die Anwendung.

Syntax

Quit(*/status/*)

Teil	Typ	Beschreibung
status	Integer	Optionaler Parameter. Wenn Sie Quit aufrufen, wird Ihre Applikation beendet und der Wert status zurückgegeben. Dieser Wert wird für Konsolen-Applikationen verwendet.

Hinweis

Wenn Fenster offen sind, ruft die Methode **Quit** für jedes Fenster den Event-Handler CancelClose auf. Wenn der Event-Handler CancelClose [False](#) zurückliefert (Standardeinstellung), wird der Event-Handler Close für das Fenster aufgerufen. Wenn ein beliebiger Event-Handler CancelClose [True](#) zurückliefert, beendet REALbasic den Versand von CancelClose- und Close-Events und die Anwendung wird nicht beendet.

Wenn keine Fenster geöffnet sind oder kein Event-Handler CancelClose [True](#) zurückliefert und eine Unterklasse auf Basis des Objektes [Application](#) existiert, wird deren Close-Event aufgerufen.

Beispiele

Dieses Beispiel beendet die Anwendung.

Quit

QuitMenuItemem Klasse

Der Beenden-Menü-Eintrag ist von der QuitMenuItemem-Klasse abgeleitet.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>BallonHelp</i>	Nachricht als String	Nachricht, die erscheint, wenn die Ballon-Hilfe aktiv ist und der Anwender sich mit der Maus über dem Menüeintrag befindet, während dieser aktiviert ist.
<i>Bold</i>	Boolean	Menüeintrag ist fett.
<i>Checked</i>	Boolean	Menüeintrag hat ein Häkchen.
<i>CommandKey</i>	Boolean	Der Macintosh-Tastatur-Kurzbefehl für den Menüeintrag. Tastatur-Kürzel für Windows werden mittels Konstanten festgelegt. Siehe Abschnitt „Verwendung von Konstanten, um Menüs und Menüeinträgen Windows-Tastatur-Shortcuts hinzuzufügen“ im Entwicklerhandbuch.
<i>DisableBallonHelp</i>	Nachricht als String	Nachricht, die erscheint, wenn die Ballon-Hilfe aktiv ist und der Anwender sich mit der Maus über dem Menüeintrag befindet, während dieser nicht aktiviert ist.
<i>Enabled</i>	Boolean	Legt fest, ob ein Menüeintrag aktiviert ist oder nicht. Menüeinträge, die auf dieser Klasse basieren, sind in der Voreinstellung aktiviert. Sie können einen Menüeintrag auch deaktivieren, indem Sie der Enabled-Eigenschaft den Wert False zuweisen.
<i>Italic</i>	Boolean	Menüeintrag ist kursiv.
<i>Name</i>	String	Der Name des Menüeintrags.

Name	Art	Beschreibung
Submenu	<u>Boolean</u>	Zeigt an, daß der Menüeintrag ein Untermenü besitzt.
<i>Tag</i>	<u>String</u>	Ein „versteckter“ Textstring, der mit diesem Menüeintrag verknüpft ist. Auf das Tag kann über Code zugegriffen werden, wenn der Anwender den Menüeintrag anwählt, im Gegensatz zur Text-Eigenschaft wird es nicht im Menü dargestellt.
<i>Text</i>	<u>String</u>	Der Text des Menüeintrags.
Underline	<u>Boolean</u>	Menüeintrag ist unterstrichen.
<i>Visible</i>	<u>Boolean</u>	Menüeintrag ist sichtbar.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Entfernt dynamisch erzeugte Menüeinträge.
<i>Enable</i>		Setzt die Enabled-Eigenschaft des Menüeintrags auf <u>True</u> . Rufen Sie Enable nur innerhalb des EnableMenuItems-Event-Handler auf.

Hinweise

Der Beenden-Menüeintrag, der voreingestellt jeder Applikation hinzugefügt wird, ist eine Instanz der **QuitMenuItem**-Klasse. Er unterscheidet sich von Menüeinträgen, die von der MenuItem-Klasse abgeleitet wurden darin, daß er als Voreinstellung aktiviert ist und automatisch die Quit-Methode aufruft, um die Applikation zu beenden. Es ist daher nicht nötig, diesen Menüeintrag über Code zu aktivieren oder seinen Handler zu schreiben.

Beispiele

Sie müssen normalerweise keine zusätzliche Instanz der QuitMenuItem-Klasse erzeugen, aber Sie können seine Eigenschaften verwenden, um das Standard-Aussehen oder Verhalten des Beenden-Menüeintrags zu modifizieren.

```
FileQuit.BalloonHelp="Beendet die Applikation"
```

Siehe auch

EnableMenuItems Funktion, New Operator; MenuItem Klasse.

RadioButton Steuerelement



Das Standard-Radiobutton-Steuerelement. Wenn Sie eine Gruppe von Radiobuttons durch ein GroupBox-Steuerlement einschließen, wird REALbasic automatische die übrigen Radiobuttons deselektieren, wenn der Anwender einen Radiobutton in der Gruppe anklickt.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>Bold</i>	<u>Boolean</u>	Stellt den Buttontext fett dar.
<i>Caption</i>	<u>String</u>	Der Buttontext.
<i>Italic</i>	<u>Boolean</u>	Stellt den Buttontext kursiv dar.
<i>TextFont</i>	<u>String</u>	Name der für den Buttontext zu verwendenden Schrift.
<i>TextSize</i>	<u>Integer</u>	Größe der für den Buttontext zu verwendenden Schrift.
<i>Underline</i>	<u>Boolean</u>	Stellt den Buttontext unterstrichen dar.
<i>Value</i>	<u>Boolean</u>	Wert des Radiobuttons, wenn das Fenster geöffnet wird, in dem sich der Radiobutton befindet.

Events

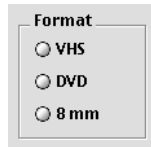
Name	Parameter	Beschreibung
<i>Action</i>		Der Button wurde angeklickt.
<i>GotFocus</i>		(Nur Win32) Das Steuerelement hat den Fokus bekommen und besitzt einen Selektionsrahmen um die Beschriftung.
<i>LostFocus</i>		(Nur Win32) Das Steuerelement hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y gedrückt. Wenn Sie MouseDown auswerten möchten, liefern Sie <u>True</u> zurück. Das Action-Event wird nicht ausgeführt und der Status des Objektes ändert sich nicht.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Steuerelements an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht <u>True</u> im MouseDown-Event liefern. Der Rückgabewert wird ignoriert.

Hinweis

Wenn die Caption-Eigenschaft ein Und-Zeichen enthält, wird dieses nur angezeigt, wenn ein weiteres Und-Zeichen folgt. Dies ist so, damit sich Macintosh- und Windows-Programme gleich verhalten.

Beispiel

Folgendes Interface verwendet eine GroupBox und drei Radiobuttons:



Folgender Code wird im Action-Event-Handler von RadioButton1 verwendet:

```
If radiobutton1.value then
  EditField1.text="VHS"
end if
```

Ähnlicher Code wird im Action-Event-Handler für RadioButton 2 und 3 verwendet.

Siehe auch

GroupBox Steuerelement; RectControl Klasse.

Raise Befehl

Der Befehl Raise wird verwendet, um eine Exception auszulösen.

Syntax

Raise expression

Expression ist ein Objekt, das eine Unterklasse von RuntimeException ist. *Expression* ist das Objekt, das an jeden Exception-Handler übergeben wird.

Beispiel

```
Dim f as FolderItem
f.delete
exception err as NilObjectException
Raise err
```

Siehe auch

Function Befehl, RuntimeException Klasse.

Random Klasse

Wird zum Erzeugen von Zufallszahlen verwendet.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Seed</i>	<u>Integer</u>	Ermittelt oder setzt Seed, das vom Zufallszahlengenerator verwendet wird.

Methoden

Name	Parameter	Beschreibung
<i>InRange</i>	minR als <u>Integer</u> maxR als <u>Integer</u>	Liefert eine Zufallszahl als ein <u>Integer</u> im Bereich von <i>minR</i> bis <i>maxR</i> .
<i>LessThan</i>	Range als <u>Integer</u>	Liefert eine Zufallszahl als ein <u>Integer</u> im Bereich größer oder gleich 0 und kleiner als <i>Range</i> -1.
<i>Number</i>		Liefert eine Zufallszahl, als ein <u>Double</u> im Bereich größer oder gleich 0 und kleiner als 1.

Hinweise

Die **Random**-Klasse ersetzt die Rnd-Funktion. Bitte schreiben Sie Ihren Code so um, daß er die **Random**-Klasse verwendet.

Sie sollten nur ein Random-Objekt in einer Applikation erzeugen. Wenn Sie immer dann ein Random-Objekt erzeugen, wenn Sie eine neue Zufallszahl benötigen, ist dies sehr ineffizient und kann zu schwer auffindbaren Fehlern führen.

Beispiel

Folgendes Beispiel erzeugt eine Zufalls-Integerzahl im Bereich von 0 bis 1.000.

```
Dim r as Random
r=New Random
Static txt1.Text=Str(r.InRange(0,1000))
```

Siehe auch

Rnd Funktion.

Range Klasse

Wird verwendet, um die Startposition, Länge und Endposition eines StyleRun-Objekts zu ermitteln.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>StartPos</i>	<u>Integer</u>	Die Startposition des <u>StyleRun</u> -Objekts.
<i>EndPos</i>	<u>Integer</u>	Die Endposition des <u>StyleRun</u> -Objekts.
<i>Length</i>	<u>Integer</u>	Die Länge des <u>StyleRun</u> -Objekts.

Hinweise

Ein StyleRun ist eine Reihe von Zeichen, die, innerhalb eines StyledText-Objekts, dieselben Stilattribute besitzen. Man kann sich einen Block von Stil-Text als eine Aneinanderreihung von StyleRuns-Objekten vorstellen. Verwenden Sie die **Range**-Klasse, um auf die Position eines StyleRun-Objekts in einem Stil-Text zuzugreifen.

Beispiel

Dieses Beispiel ermittelt in einer Schleife die Positionen aller vorhandenen StyleRuns-Objekte eines StyledText-Objekts und zeigt deren Texte und Startpositionen in einer ListBox an.

```
Dim count, i as Integer
count=EditField1.StyledText.StyleRunCount //ermittle die Anzahl von StyleRuns
For i=0 to count-1 //gehe durch alle durch
  ListBox1.Addrow Str(EditField1.StyledText.StyleRunRange(i).StartPos)
  ListBox1.Cell(i,1)=Str(EditField1.StyledText.StyleRunRange(i).Length)
  ListBox1.Cell(i,2)=EditField1.StyledText.StyleRun(i).Text
```

Next

Siehe auch

[StyledText](#), [Paragraph](#), [StyleRun](#), [EditField](#) Klassen.

RB3DSpace Steuerelement



Wird verwendet, um Objekte im 3D-Raum anzuzeigen und zu manipulieren.

Eigenschaften

Name	Art	Beschreibung
<i>AmbientColor</i>	Color	Die Farbe des eingebauten Umgebungslichtes.
<i>AmbientLight</i>	Integer	Helligkeit des Umgebungslicht. Reicht von 0 bis 100.
<i>Background</i>	Object3D	Hintergrundobjekt oder gruppiertes Objekt (Group3D).
<i>Camera</i>	Object3D	Kamera-Position und Ausrichtung.
<i>DebugCube</i>	Boolean	Falls <u>True</u> , werden die X-, Y-, und Z-Achsen im Hintergrund des RB3DSpace gezeichnet, um die Orientierung zu erleichtern. Nicht verfügbar in REALbasic Version 5.0.
<i>FieldOfView</i>	Integer	Ansichtswinkel (Grad).
<i>FloodColor</i>	Color	Die Farbe des Lichtes des eingebauten Flutlichtes. Die Default-Farbe ist weiß.
<i>FloodDirection</i>	Vector3D	Legt die Richtung des eingebauten Flutlichtes fest. Falls Nil, erhalten Sie ein Punkt-Licht, das in alle Richtungen strahlt.
<i>FloodLight</i>	Integer	Helligkeit des Scheinwerfers. Reicht von 0 bis 100.
<i>FogStart</i>	Integer	Falls FogVisible <u>True</u> ist, gibt dies an, wie weit entfernt von der Kamera der Nebel beginnen soll. Der Nebel wird von diesem Punkt (oder der Hither-Ebene) an dicker, je nach dem, was weiter weg ist, bis zur Yon-Distanz, wo er komplett undurchsichtig ist. Die Voreinstellung ist Null. Die Nebelfarbe ist die gleiche wie SkyColor .
<i>FogVisible</i>	Boolean	Falls <u>True</u> , wird der Nebel passend zu den FogStart -Parametern gezeichnet. Voreinstellung ist False.
<i>Hither</i>	Integer	Nahe Clipping-Ebene.
<i>LightCount</i>	Integer	Die Anzahl der Lichtquellen in der Szene.
<i>Objects</i>	Group3D	Zu rendernde Objekte.
<i>Hither</i>	Integer	Nahe Clipping-Ebene.
<i>ViewHandle</i>	Integer	Erlaubt Ihnen, auf das TQ3ViewObject , das mit dem Steuerelement verbunden ist, zuzugreifen, um es mit Declarations zu Quesa/QD3D zu verwenden. Bedenken Sie dabei, daß das zurückgegebene Objekt nicht geschützt ist, so daß Sie es weder entfernen sollten noch versuchen sollten, es zu benutzen, nachdem das Steuerelement geschlossen wurde.
<i>WireFrame</i>	Boolean	Wenn <u>True</u> , wird es als Drahtgitter gerendert.
<i>Yon</i>	Integer	Entfernte Clipping-Ebene.

Methoden

Name	Parameter	Beschreibung
<i>AddLight</i>	Light als Light3D	Fügt der Szene eine Lichtquelle hinzu.
<i>FindObject</i>	x als Integer y als Integer	Findet das Objekt bei x,y (Pixel) innerhalb des RB3DSpace. Liefert <u>Nil</u> , wenn der Punkt kein Objekt enthält. Liefert sonst das an diesem Punkt gezeichnete Objekt als ein Object3D .
<i>FindPoint</i>	x als Integer y als Integer	Findet den Punkt bei x,y (Pixel) innerhalb des RB3DSpace. Liefert <u>Nil</u> , wenn x,y keinen Punkt enthält. Liefert sonst ein Vector3D .

Name	Parameter	Beschreibung
<i>GetPicture</i>	[Width als <u>Integer</u> , Height als <u>Integer</u> , Depth als <u>Integer</u>]	Rendert die aktuelle Szene in ein neu erzeugtes <u>Picture</u> und liefert dieses zurück. Wenn die Parameter weggelassen werden, werden die aktuelle Breite und Höhe des Steuerelements und 32 als Farbtiefe verwendet.
<i>Light</i>	Index als <u>Integer</u>	Ermittelt oder setzt eine Lichtquelle.
<i>RemoveLight</i>	Index als <u>Integer</u> ODER Light als <u>Light3D</u>	Entfernt die angegebene Lichtquelle aus der Szene. Sie können die zu entfernende Lichtquelle entweder über ihren Index oder über die Referenz auf diese entfernen.
<i>Update</i>		Aktualisiert den 3D-Raum ohne zu löschen oder ein Neuzeichnen auszulösen.

Hinweise

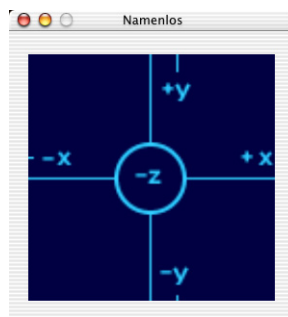
Systemvoraussetzungen

Die 3D-Klassen in REALbasic benötigen entweder Quesa oder QuickDraw 3D, um funktionieren zu können. Letzteres ist eine Library von Apple Computer und Teil von QuickTime, wogegen Quesa ein Open-Source-Projekt ist, das ungefähr das gleiche erledigt. Welches von beiden Sie verwenden, hängt auch vom Betriebssystem ab, das Sie verwenden.

Unter Linux wird OpenGL benötigt.

Unter dem Classic Mac OS ist QuickDraw 3D normalerweise installiert, so daß Sie keine zusätzlichen Bibliotheken installieren müssen. Sie können stattdessen Quesa Classic installieren, wenn Sie das bevorzugen.

Unter OS X oder Carbon müssen Sie Quesa (und OpenGL) installieren. Wenn Quesa installiert ist, sehen Sie die X-, Y-, und Z-Achsen, wenn Sie die DebugCube-Eigenschaft auf True setzen, wie im folgenden zu sehen:



Unter Windows haben Sie die Wahl. QuickDraw 3D ist Teil von QuickTime 5 für Windows, so daß Sie dies oder Quesa (und OpenGL) verwenden können. Das RB3DSpace-Steuerelement ist so geschrieben, daß es Quesa erwartet und da Windows Libraries aufgrund der Dateinamen sucht, müssen Sie, um QuickDraw 3D zu verwenden, QD3D.DLL in QUESA.DLL umbenennen (wir raten aber davon ab, Quesa läuft unter Windows schneller).

Wenn die benötigten Libraries nicht installiert sind, sind die Objekt-Eigenschaften eines jeden RB3DSpace-Objektes Nil. Wenn die passenden Libraries vorhanden sind, wird diese Eigenschaft automatisch zu einem Group3D initialisiert (welches nicht Nil ist). Somit sollten Sie in das Open-Event des RB3DSpace-Steuerelements eine Überprüfung wie folgende einbauen:

```
If Me.objects = Nil then
    MsgBox "Sie müssen QuickDraw 3D oder Quesa installieren!"
Return
End if
```

Beachten Sie, daß dies nur unter Mac OS funktioniert. Wenn unter Windows die QUESA.DLL nicht gefunden werden kann, startet Ihr Programm nicht einmal. Dies ist nur ein Seiteneffekt davon, wie Shared Libraries unter Windows funktionieren.

Hintergrund

Das RB3DSpace-Steuerelement ist das Steuerelement, in dem das 3D-Rendering vorgenommen wird. Die Achsen des 3D-Raums werden angezeigt, wenn Sie die DebugCube-Eigenschaft auf True setzen. Die Z-Achse ist die „Tiefen“-Dimension und die „Länge“ dieser Achse wird durch die Hither- und Yon-Eigenschaften festgelegt. Die Hither- und Yon-Eigenschaften legen die nahe und die entfernte Clipping-Ebene (die Z-Achse) fest; Sie sollten Hither und Yon so nah wie möglich zueinander wählen, um den Z-Buffer Ihrer Hardware bestmöglich auszunutzen. FieldOfView wird in Grad angegeben und legt fest, wie weit der Blickwinkel der Kamera ist; verkleinern Sie den Wert, um hineinzuzoomen oder vergrößern Sie ihn, um herauszuzoomen.

Beachten Sie, daß die Kamera als ein Objekt definiert ist; sie benötigt keinerlei Geometrie, Sie können so aber die Kamera exakt so wie jedes andere Objekt bewegen. Der Hintergrund ist als ein Object3D definiert, Sie können jedoch, wenn Sie wollen, ihm ein Group3D-Objekt zuweisen.

Alle anderen Objekte in der Szene werden in den Objekt-Eigenschaften gespeichert, wobei es sich hierbei um ein Group3D-Objekt handelt, damit Sie einfach Dinge hinzufügen können (oder über Objekten in der Szene wiederholen, etc.).

Wenn Sie das Objekt oder den Punkt im 3D-Raum finden müssen, der zu einer bestimmten Pixelposition im Sichtbereich gehört, können Sie FindObject und FindPoint verwenden (diese liefern Nil, wenn der angeklickte Punkt kein Objekt enthält).

Eine Stelle im 3D-Raum festlegen

Jede Stelle im 3D-Raum kann durch drei Koordinaten festgelegt werden: X, Y und Z. Wenn man sich die Welt als eine endlose flache Ebene vorstellt, kann man einen willkürlichen Punkt als den Ursprung wählen. Dann kann man jede Stelle als eine Ost/West-Entfernung, eine Nord/Süd-Entfernung und eine Höhe/Tiefe festlegen.

Das willkürliche Zentrum des virtuellen Universums wird „Ursprung“ genannt. Die drei Koordinaten in 3D-Grafiken werden nach Konvention X, Y und Z genannt und der Ursprung ist der Punkt 0,0,0. Die X-Achse ist die Linie durch den Ursprung, wobei Y=0 und Z=0 sind; gleichermaßen ist die XY-Ebene die unendliche Ebene durch den Ursprung, wobei Z=0 ist.

Die Standard-Interpretation der Achsen ist, daß die XZ-Ebene der Boden ist (oder parallel zum Boden) und die Koordinate Y die Höhe ist. In REALbasic sind die drei Ebenen so angeordnet, als wenn man in die -Z-Richtung blickt, wobei +Y nach oben zeigt und +X rechts ist (das ist auch die Voreinstellung der Kamera). Wenn man nun in diese Richtung blickt, führt eine Vergrößerung der X-Koordinate der Position eines Objektes, daß es sich von Ihrer Sicht aus nach rechts bewegt. Eine Vergrößerung von Z führt es näher an Sie heran und eine Verkleinerung bewegt das Objekt weiter weg.

Die FieldOfView-Eigenschaft

FieldOfView ist der Winkel in Grad, der im Bereich des RB3DSpace-Steuerelements gezeigt wird. Ein normaler Mensch hat einen Blickwinkel von 180 Grad, der Monitor selbst liefert jedoch bei normaler Betrachtung etwa 10 Grad. Wenn Sie versuchen, ein großes Blickfeld auf eine Anzeige von nur 10 Grad zu komprimieren, erzeugen Sie einen „Fischauge“-Effekt, der sehr unnatürlich wirkt. Andererseits ist es so, wenn Sie nur 10 Grad Blickfeld in einer typischen 3D-Umgebung verwenden, daß sich der Anwender so fühlt, als hätte er eine Art Tunnelblick und wird ständig mit der Kamera umherschwenken. Die Wahl von FieldOfView muß also ein Kompromiß sein in diesem Konflikt.

Man möchte einen möglichst großen Ausschnitt der Szene sehen, dieser soll aber so realistisch wie möglich aussehen.

Ein typischer FieldOfView-Wert in vielen 3D-Spielen liegt zwischen 40 und 50 Grad. Sie können jedoch auch eine Zoom-Funktion (Teleskop-Ansicht) anbieten, indem Sie vorübergehend FieldOfView verringern – ein kleineres FieldOfView führt dazu, daß ein kleinerer Teil der Welt im gleichen Bereich auf den Bildschirm gezeigt wird, was dazu führt, daß alles in diesem Teil größer aussieht.

Außerdem erlaubt Ihnen die Teleskop-Ansicht durch einen kleinen FieldOfView, die Perspektive bei Bedarf zu reduzieren (und nahezu zu eliminieren). Positionieren Sie die Kamera weit weg von der Szene und verwenden Sie einen kleinen FieldOfView; das entspricht einem Blick aus dem Himmel auf den Boden und unter diesen Umständen gibt es kaum perspektivische Verzerrungen.

Die Hither- und Yon-Eigenschaften

Das sind die Standard-3D-Bezeichnungen für die „nahe-ferne“ Z-Achse. Es wird nichts gezeichnet, was sich näher an der Kamera befindet als die Hither-Entfernung; und wenn sich ein Objekt mitten durch Hither erstreckt, wird es einfach in zwei Hälften geschnitten. Alles, was näher als Hither ist, wird aus der Szene ausgeblendet (also der „nahen Clipping-Ebene“). Genauemaßen wird nichts gezeichnet, was sich weiter von der Kamera als *Yon* befindet.

Wenn das alles wäre, könnte man einfach Hither auf Null und Yon auf einen sehr großen Wert setzen, so daß nichts abgeschnitten wird. Das ist in Wirklichkeit aber eine schlechte Idee. Der Grund hierfür ist, daß Grafikkarten eine begrenzte Auflösung im „Tiefen-Puffer“ haben, der dazu verwendet wird, zu ermitteln, was sich vor welchem Objekt befindet. Das Verhältnis Yon/Hither muß genau

durch die Anzahl von Bits im Tiefen-Puffer der Karte geteilt werden. Wenn dieses Verhältnis sehr groß ist, wird die Tiefenauflösung unpassend und man sieht viele Arten von Artefakten – Objekte auf seltsame Weise vermischt, weit entfernte Objekte werden vor näheren gezeichnet usw. Daher ist es sehr wichtig für die Qualität des Renderings, daß Sie das Verhältnis von Yon/Hither so klein wie möglich belassen. Vergrößern Sie Hither und verkleinern Sie Yon so weit wie möglich in Ihrer Applikation.

Ein Objekt ausrichten

Es gibt unzählige Ausrichtungen, die alle auf einen bestimmten interessanten Punkt (Point of Interest oder auch POI) zeigen. Überprüfen Sie dies selbst, indem Sie eine echte Kamera mit Sucher nehmen (oder sich vorstellen). Schauen Sie durch den Sucher auf ein bestimmtes Objekt, z.B. einen Baum auf der Straße. Rotieren Sie nun, ohne den Baum aus der Sicht zu verlieren, die Kamera um ihre Achsen. Es gibt eine unzählige Zahl unterschiedlicher Ausrichtungen, in der Sie die Kamera anhalten können, auch wenn alle auf den Baum zeigen.

Das erste, was man machen muß, ist das Festlegen einiger zusätzlicher Einschränkungen bei der Ausrichtung. Eine übliche ist, daß das Objekt so „aufrecht“ wie möglich bleibt. Es gibt viele Möglichkeiten, das zu erreichen, hier ist eine mögliche davon:

```
Sub PointObjectAt(obj As Object3D, POI As Vector3D)
    // Läßt 'obj' auf den interessanten Punkt 'POI' zeigen,
    // indem zuerst ein Yaw (aus der voreingestellten Position) ausgeführt wird und
    // dann ein Pitch. Die vorige Ausrichtung des Objektes ist nicht betroffen.
    Dim yawAngle As Double
    Dim pitchAngle As Double
    Dim v1, v2 As Vector3D
    // Finde den nötigen yaw-Winkel, um in die richtige Richtung zu kommen
    // (also Y ignorieren, um zu erreichen, daß es in die richtige XZ-Richtung zeigt).
    // Die Atan2-Funktion liefert eine einfache Möglichkeit, diesen zu finden.
    yawAngle = Atan2(POI.x-obj.position.x, POI.z-obj.position.z)
    // Setze die Ausrichtung des Objekts auf ein Yaw mit diesem Wert.
    obj.orientation.SetRotateAboutAxis 0,1,0, yawAngle
    // Finde nun den Pitch-Winkel, der für Y benötigt wird.
    // Wir verwenden ein Skalarprodukt, um den Winkel zwischen einem Vektor in der
    // Richtung, in die das Objekt nun zeigt und einem Vektor, der direkt
    // auf den POI zeigt, zu bekommen.
    v1 = New Vector3D
    v1.z = 1.0
    v1 = obj.orientation.Transform(v1)
    v2 = New Vector3D
    v2.x = POI.x - obj.position.x
    v2.y = POI.y - obj.position.y
    v2.z = POI.z - obj.position.z
    v2.Normalize
    pitchAngle = Acos( Min(Max(v1.Dot(v2),-1.0),1.0) )
    If v2.y > 0 then
        pitchAngle = -pitchAngle
    end if
    // Diesen Pitch zuweisen.
    obj.Pitch pitchAngle
End Sub
```

Ein Objekt um einen POI herumkreisen lassen

Dies ist eine weitere Aufgabe, die auf viele Wege gelöst werden kann. Zunächst sollte man sich genau bewußt machen, was man machen will. Wenn das Objekt im „Orbit“ eine Kamera ist, die man um einen interessanten Punkt POI bewegen möchte, gibt es verschiedene Möglichkeiten:

- Bewegen Sie die Kamera auf der Oberfläche eines imaginären Y-Achsen-basierten Zylinder, der das Objekt umgibt. Links/Rechts-Bewegungen bewegen die Kamera um den Zylinder herum und Hoch/Runter-Bewegungen bewegen die Kamera an der Seite des Zylinders hoch und runter. Beachten Sie, daß sich die Kamera hierbei vom POI entfernt, wenn Sie sie hoch oder runter bewegen. Das bedeutet aber auch, daß die Links/Rechts-Hoch/Runter-Bewegungen gut festgelegt und offensichtlich sind.
- Bewegen Sie die Kamera auf der Oberfläche einer imaginären Kugel um das Objekt herum. Das Kamera-Objekt hat eine bestimmte Richtung, in die es zeigt (auf das Zentrum) und Links/Rechts-Hoch/Runter-Bewegungen bewegen die Kamera auf der Oberfläche der Kugel relativ zu dieser Richtung. Dies führt zu einem konstanten Abstand zum POI, die Steuerelemente können jedoch durcheinanderkommen, da das sich bewegende Objekt leicht verkehrt-herum oder seitlich enden kann.

- Bewegen Sie die Kamera auf der Oberfläche einer imaginären Kugel um das Objekt herum. In diesem Fall lassen Sie die Links/Rechts-Steuerelemente die Longituden-Position (Längenangabe) und die Hoch/Runter-Steuerelemente die Latituden-Position (Breitenachse) der Kamera steuern. Dies führt zu einem konstanten Abstand zum POI und ist sehr einfach zu steuern da (wie im Falle des Zylinders) die Links/Rechts-Hoch/Runter-Richtungen immer gut festgelegt sind.

In all diesen Fällen kann die Position des Objektes durch zwei Parameter festgelegt werden. Links/Rechts-Steuerelemente erhöhen oder erniedrigen einen der Parameter und die Hoch/Runter-Steuerelemente den anderen. Der einzige Unterschied zwischen diesen ist, wie diese Parameter in 3D-Koordinaten konvertiert werden (und daraus folgend, was die Parameter bedeuten). In allen Fällen können Sie einen dritten Parameter, den Radius, hinzufügen, um die Größe des Orbits zu verändern.

Hier ist der Code für die erste Option:

```
Function OrbitCylinder(POI As Vector3D, radius As Double, angle As Double, altitude As Double) As Vector3D
// Berechnet eine Position auf einem virtuellen, Y-basierenden Zylinder, zentriert
// auf POI. Mit gegebenem Radius, Winkel und Höhe.
Dim out As Vector3D
out = New Vector3D
out.x = POI.x + radius * Cos(angle)
out.z = POI.z + radius * Sin(angle)
out.y = POI.y + altitude
Return out
End Function
```

Beispiel

Dieses Beispiel zeigt, wie man eine 3DMF-Datei in ein RB3DSpace lädt. Dieser Code gehört in das Open-Event. Beachten Sie, daß "3DMF" im Dateitypen-Dialog mit dem Type "3DMF" definiert werden sollte.

```
Dim f As FolderItem
Dim modelCode As String
Dim obj As Object3D
f = GetFolderItem("Penguin.3DMF")
If f <> Nil then
// holt den 3DMF-Code für das Modell
modelCode = f.OpenAsTextFile.ReadAll
// Erzeuge ein Object
obj = New Object3D
// gibt dem Objekt eine Gestalt, festgelegt durch das 3DMF
obj.AddShapeFromString modelCode
// fügt das Object diesem Rb3DSpace hinzu
Me.objects.Append obj
end if
// Bewege die Kamera, damit man das Objekt sehen kann
Me.camera.position.Z = 500
```

Die letzte Zeile bewegt die Kamera weg vom Objekt, so daß es sichtbar wird. Voreingestellt befindet sich die Kamera an Position 0,0,0 (auch bekannt als der „Ursprung“ im 3D-Raum). Voreingestellt werden alle Objekte an der gleichen Stelle geladen. Das bedeutet, daß sich die Kamera in Ihrem Objekt befindet, wo Sie sie normalerweise nicht wollen. Beheben Sie dies, indem Sie die Kamera vom Ursprung weg bewegen. Wie weit Sie sie bewegen müssen, hängt von der Größe des Objekts ab.

Siehe auch

[Object3D](#), [Group3D](#), [Vector3D](#), [Quaternion](#), [Bounds3D](#), [Light3D](#) Klassen.

RBScript Steuerelement



Wird zum Ausführen von REALbasic-Code innerhalb einer laufenden (compilierten) Anwendung verwendet.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>Context</i>	<u>Object</u>	Objekt (z.B. Fenster oder Klasse) das für das RBScript verfügbar gemacht wurde. Sie können zum Beispiel eine selbstdefinierte Klasse erzeugen und diese der Context-Eigenschaft zuweisen. Das Skript hat dann Zugriff auf die Methoden und Eigenschaften dieser Klasse. Die Methoden werden zu globalen Methoden und die Eigenschaften werden globale Eigenschaften.
<i>CurrentLineNumber</i>	<u>Integer</u>	Die Zeile, die der Interpreter ausführt.
<i>EncodingFont</i>	<u>String</u>	Nur auf Doppel-Byte-Zeichen-Systemen nützlich. Dieser Font wird benutzt, um festzulegen, welches Skript-System verwendet wird, um Strings in Funktionen wie Len und Mid zu interpretieren. Um beispielsweise die Interpretation von Strings als japanisches Skript fest einzustellen, müssen Sie diese Eigenschaft auf <i>Osaka</i> stellen. Wenn es leer gelassen wird, wird das System-Skript verwendet.
<i>Source</i>	<u>String</u>	Der Source-Code, den der Interpreter ausführen wird.
<i>State</i>	<u>Integer</u>	Status des Skript-Interpreters: 0=Bereit 1=Läuft gerade 2=Fertig 3=Abgebrochen

Methoden

Name	Parameter	Beschreibung
<i>PartialRun</i>	Millisekunde als <u>Integer</u>	Führt den Code <i>Source</i> aus, bis er fertig ist oder <i>Millisekunde</i> verstrichen ist.
<i>Precompile()</i>		Läßt den Parser sofort laufen und wartet nicht auf den nächsten Run-Aufruf.
<i>Reset()</i>		Stellt den Interpreter auf Null zurück.
<i>Run()</i>		Läßt den Code <i>Source</i> laufen, bis er fertig ist.

Events

Name	Parameter	Beschreibung
<i>CompilerError</i>	line als <u>Integer</u> ; errorNumber als <u>Integer</u> ; errorMsg als <u>String</u>	Ein Compilerfehler ist aufgetreten. Dies kann an einem Syntaxfehler im Code liegen. Beachten Sie hierzu die Fehlernummern unten bei den Hinweisen.
<i>Input</i>	prompt als <u>String</u>	Das Skript verlangt eine Eingabe vom End-Anwender.
<i>Print</i>	msg als <u>String</u>	Der Interpreter gibt das Ergebnis des Skripts in <i>msg</i> zurück.
<i>RunTimeError</i>	line als <u>Integer</u> ; error als <u>RunTimeException</u>	Ein Laufzeitfehler ist aufgetreten.

Hinweise

Die RBScript-Sprache ist eine Implementation der REALbasic-Sprache, die es dem Anwender erlaubt, REALbasic-Code innerhalb einer compilierten Anwendung zu schreiben und auszuführen. Skripte werden in Maschinensprache compiliert und laufen nicht durch einen Interpreter wie in REALbasic-Versionen vor der 4.5.

Da RBScript eine Klasse ist, verwenden Sie es, indem Sie eine Instanz dieser Klasse entweder durch Code erzeugen oder das RBScript-Werkzeug aus der Steuerelemente-Palette in ein Fenster draggen. Der einfachste Weg, RBScript zu verwenden ist, den Code der Source-Eigenschaft des RBScript-Objekts zuzuweisen und die Run-Methode aufzurufen.

Um Informationen an ein RBScript zu übergeben, wenn es läuft, können Sie die Input-Funktion verwenden. Diese Funktion ruft das Input-Event des RBScript-Objekts auf, wo Sie die gewünschten Information zurückliefern können, die von der Input-Funktion in Ihrem RBScript-Code zurückgeliefert wurden. Im folgenden Beispiel wird das Ergebnis der Input-Funktion einer Variablen zugewiesen:

```
Dim Years, Days as Integer
Years = Val(Input(""))
Days = Years * 365
```

Die Input-Funktion übernimmt einen String, der verwendet werden kann, um eine Eingabemöglichkeit anzubieten, falls Sie eine Dialogbox anbieten möchten, in die der Anwender Informationen eingeben kann. Da die Input-Funktion einen String zurückliefert und der Wert als Integer gespeichert werden soll, wird die Val-Funktion verwendet, um den String in ein Integer zu konvertieren. In diesem Fall, wenn die Anzahl der Jahre in ein EditField namens EditField1 eingegeben wird, würde das Input-Event des RBScript-Objekts etwa so aussehen:

```
Function Input(prompt as String) as String
    Return Editfield1.text
End Function
```

Wenn die Run-Methode des RBScript-Objekts aufgerufen wird, wird der Code kompiliert und dann ausgeführt. Sobald die Input-Funktion aufgerufen wurde, wird das Input-Event des RBScript-Objekts ausgeführt und der Inhalt der Text-Eigenschaft des EditFields wird zurückgeliefert und der Years-Variablen zugewiesen. Danach wird der Days-Wert errechnet.

Informationen von einem RBScript bekommen

Sie erhalten Daten von einem RBScript, indem Sie die Print-Methode verwenden. Diese Methode übernimmt einen String und übergibt ihn an das Print-Event des RBScript-Objekts. Hier ist ein Beispiel, das so verändert wurde, daß es die Print-Funktion verwendet:

```
Dim Years, Days as Integer
Years = Val(Input(""))
Days = Years * 365
Print Str(days)
```

Sie können auf den Wert, der an die Print-Methode übergeben wird, durch das Print-Event zugreifen. Wenn Sie zum Beispiel den Wert der Text-Eigenschaft eines StaticText-Objekts zuweisen wollen, würde der Code für das Print-Event des RBScript-Objekts etwa so aussehen:

```
Sub Print(msg as String)
    StaticText1.text = msg
End Sub
```

Fehlerbehandlung im Ihrem Code

Wenn ein Fehler auftritt, wenn das RBScript Ihren Code kompiliert, wird das CompilerError-Event des RBScript-Objekts aufgerufen und liefert die passende Fehler-Information, so daß Sie entscheiden können, wie der Fehler behandelt werden soll. Wenn der Fehler auftritt, während der Code am Laufen ist, wird das RuntimeError-Event des RBScript-Objekts aufgerufen und liefert die passende Fehler-Information. Sie können dann entscheiden, wie auf den Fehler reagiert werden soll.

Variablen und Konstanten

Alle numerischen und alphanumerischen Operatoren werden unterstützt:

+, -, *, /, \, Mod, <=, =, >, <=>, >=>, <>

Logische Operatoren: And, Not, Or.

Alle Formen von Kommentaren werden unterstützt: ', // und REM.

Daten-Typen

RBScript unterstützt die folgenden Daten-Typen:

- Integer
- Single
- Double
- Boolean
- String
- Color
- Object
- Variant

Arrays können jeden dieser Typen verwenden.

Kontrollstrukturen

Alle Kontrollstrukturen, die in dieser *Sprachreferenz* festgelegt sind, werden unterstützt. Dies beinhaltet:

- Function ... End Funktion
- Sub ... End Sub
- For ... Next
- Do ... Loop
- If ... Then ... End If
- Select Case ... End Select
- While...Wend
- Return Befehl

Klassen

Sie können eine Klasse mit der Struktur **Class ... End Class** erzeugen. Eine neue Klasse kann Eigenschaften, Methoden und Events besitzen.

```
Class NewClass
  Dim i as Integer //Eigenschaften-Definitionen..
  Dim c as Color
  Sub myMethod (x as Integer, y as Integer)
    //Methoden-Definition hierhin
  End Sub
//Klassen-Definition
End Class
```

Ein Klasse kann von einer anderen Klasse mit der „Inherits“-Deklaration abgeleitet werden.

Module

Sie können **Module** über die Module-Struktur erzeugen. Zum Beispiel:

```
Module Foo
  Dim bar As Integer
  Sub Baz ()
    bar = 42
  End Sub
End Module
```

Module funktionieren ähnlich wie REALbasic-Module. Der Unterschied ist, daß Eigenschaften immer geschützt und Konstanten nicht erlaubt sind. Sie können den Effekt einer öffentlichen Eigenschaft oder Konstante einfach dadurch bekommen, indem Sie den Dim- oder Const-Befehl außerhalb des Moduls plazieren.

TypeCasting

TypeCasting wird unterstützt. Verwenden Sie den gewünschten Typen-Namen als eine Funktion, deren einziges Argument die Objekt-Referenz ist, die Sie casten wollen. Wenn dies schief geht, wird ein `IllegalExceptionError` ausgelöst.

„Super“-Schlüsselwort

Das neue Schlüsselwort „Super“ erlaubt es Ihnen, überschriebene Methoden aufzurufen, ohne festlegen zu müssen von welcher Klasse die Methode kommt. Dies funktioniert auch bei Konstruktoren. Sie können überschriebene Methoden einer übergeordneten Klasse aufrufen, indem Sie die REALbasic-Syntax *Klassen-Name.Methoden-Name()* verwenden.

Vererbung (Inheritance)

Klassenvererbung ist über die **Inherits**-Deklaration implementiert.

```
Class MyNewSubClass
  Inherits NewClass
  //weiter mit dem Code hier....
End Class
```

Events

Klassen können neue Events mit der **New Event** Deklaration deklarieren.

```
Class myClass
  New Event myEvent (myParameter as DataType) as DataType
End Class
```

Methoden können solche Events unter Verwendung der **Handles Event** Deklaration bearbeiten. Zum Beispiel:

```
Class myClass
  New Event myEvent (myParameter as DataType) as DataType
```

```
End Class
Class mySubClass
  Inherits myClass
  Function myEvent (myParameter as DataType) Handles Event
    //weiter mit der Funktion hier
  End Function
End Class
```

Interfaces

RBScript unterstützt nun Interfaces, die durch die **Interface ... End Interface** Struktur deklariert werden. Interface-Methoden werden mit einer Sub- oder Function-Zeile deklariert, so wie diese innerhalb einer Klasse deklariert werden würden, sie haben jedoch keinen Inhalt und brauchen keine **End Sub** oder **End Function** Zeile:

```
Interface MovableItem
  Sub Move( x As Integer, y As Integer)
    Function LocationX() As Integer
    Function LocationY() As Integer
  End Interface
```

Hinweis: Funktionsdeklarationen ohne Parameter benötigen eine leere Klammer, wie oben zu sehen ist.

Eine Klasse kann eine Deklaration vornehmen, daß sie ein Interface mit dem **Implements**-Schlüsselwort implementiert:

```
Class Box
  Implements MovableItem
End Class
```

Klassen können beliebig viele Interfaces implementieren, sie müssen aber alle im Interface aufgeführten Methoden anbieten. Interfaces können in allen Situationen wie in REALbasic verwendet werden.

Ein- und Ausgabe

Funktion	Beschreibung
Input(Prompt als <u>String</u>)	Bekommt die Eingabe vom Anwender. Diese Funktion löst das Input-Event aus.
Print(<u>String</u>)	Sendet die Ausgabe an das implizierte Ausgabegerät. Diese Funktion löst das Print-Event aus.

Funktionen

Standard-Library-Funktionen werden wie folgt unterstützt. Diese Funktionen arbeiten wie in dieser Sprachreferenz angegeben, außer der Kommentar sagt etwas anderes. String-Funktionen können mit der Syntax der String-Methoden, wie in REALbasic, aufgerufen werden.

Funktion	Kommentar
<u>Abs(Double)</u> As <u>Double</u>	
<u>Acos(Double)</u> As <u>Double</u>	
<u>Asc(String)</u> As <u>Integer</u>	
<u>AscB(String)</u> As <u>Integer</u>	
<u>Asin(Double)</u> As <u>Double</u>	
<u>Atan(Double)</u> As <u>Double</u>	
<u>Atan2(Double, Double)</u> As <u>Double</u>	
<u>BitwiseAnd(Integer, Integer)</u> As <u>Integer</u>	
<u>BitwiseOr(Integer, Integer)</u> As <u>Integer</u>	
<u>BitwiseXor(Integer, Integer)</u> As <u>Integer</u>	
<u>ByRef</u> Schlüsselwort	
<u>ByVal</u> Schlüsselwort	
<u>CDBl(String)</u> As <u>Double</u>	In jedem Fall zu Val() identisch.
<u>Ceil(Double)</u> As <u>Double</u>	
<u>Chr(Double)</u> As <u>String</u>	
<u>ChrB(Double)</u> As <u>String</u>	
<u>CMY</u>	

Funktion	Kommentar
<u>Color</u>	
<u>Const</u> Name = Wert	
<u>Cos</u> (Double) As Double	
<u>CountFields</u> (String, String) As Integer	
<u>CStr</u> (Double) as String	
<u>Dim</u>	Erlaubt es eher, mehrere Variablen von verschiedenem Typ zu deklarieren, als nur mehrere Variablen des gleichen Typs. Leere Arrays können durch Angabe von -1 Elementen erzeugt werden.
<u>Do...Loop</u>	
<u>Exit</u>	
<u>Exp</u> (Double) As Double	
<u>False</u>	
<u>Floor</u> (Double) As Double	
<u>For...Next</u>	Erlaubt Fließkomma-Schleifenzähler und Schrittwerte.
<u>Format</u> (Double, String) As String	
<u>Function</u>	Funktionen können aufgerufen werden, bevor Sie definiert sind.
<u>Goto</u> Schlüsselwort	
<u>Hex</u> (Integer) As String	
<u>HSV</u>	
<u>If...Then...Else</u>	
<u>InStr</u> (Integer, String, String) As Integer	
<u>InStrB</u> (Integer, String, String) As Integer	
<u>IsA</u> Operator	Kann mit einem Objekt verwendet werden; liefert True bei jedem nicht-null Objekt.
<u>Left</u> (String, Integer) As String	
<u>LeftB</u> (String, Integer) As String	
<u>Len</u> (String) As Integer	
<u>LenB</u> (String) As Integer	
<u>Log</u> (Double) As Double	
<u>Lowercase</u> (String) As String	
<u>LTrim</u> (String) As String	
<u>Max</u> (Double, Double) As Double	
<u>Me</u>	
<u>Microseconds</u> As Double	
<u>Mid</u> (String, Integer, Integer) As String	
<u>MidB</u> (String, Integer, Integer) As String	
<u>Min</u> (Double, Double) As Double	
<u>Nil</u>	
<u>NthField</u> (String, String, Integer) As String	
<u>Oct</u> (Integer) As String	
<u>Pow</u> (Double, Double) As Double	
<u>Redim</u>	
<u>Rem</u>	

Funktion	Kommentar
<u>Replace</u> (String, String, String) As String	
<u>ReplaceB</u> (String, String, String) As String	
<u>ReplaceAll</u> (String, String, String) As String	
<u>ReplaceAllB</u> (String, String, String) As String	
<u>RGB</u>	
<u>Right</u> (String, Integer) As String	Arbeitet wie <u>RightB</u> .
<u>RightB</u> (String, Integer) As String	
<u>Rnd</u> As Double	
<u>Round</u> (Double) As Double	
<u>RTrip</u> (String) As String	
<u>Select Case</u>	
<u>Self</u>	
<u>Sin</u> (Double) As Double	
<u>Sqrt</u> (Double) As Double	
<u>Str</u> (Double) As String	
<u>StrComp</u> (String, String, Integer) As Integer	
<u>Sub</u>	
<u>Tan</u> (Double) As Double	
<u>Ticks</u> as Integer	
<u>Titlecase</u> (String) As String	
<u>Trim</u> (String) As String	
<u>True</u>	
<u>Ubound</u> (Array) As Integer	
<u>Uppercase</u> (String) As String	
<u>Val</u> (String) As Double	Wissenschaftliches Format wird nicht erkannt.
<u>While...Wend</u>	

Compilerfehlernummern

Compilerfehlernummern, die in **errorNumber** zurückgeliefert werden, sind hier aufgelistet:

Fehlernummer	Beschreibung
1	Syntax-Fehler.
2	Falscher Typ.
3	Select Case unterstützt diese Ausdrucks-Typ nicht.
4	Der Compiler ist nicht implementiert (überholt).
5	Der Stack des Parsers ist übergelaufen.
6	Zu viele Parameter für diese Funktion.
7	Nicht genug Parameter für diese Funktion.
8	Die Anzahl der übergebenen Parameter für diesen Funktionsaufruf stimmt nicht.
9	Parameter sind inkompatibel mit dieser Version.
10	Zuweisung eines inkompatiblen Datentyps.
11	Nicht definierter Identifier.
12	Nicht definierter Operator.
13	Logische Operationen benötigen boole'sche Operanden.

Fehlernummer	Beschreibung
14	Array-Grenzen müssen Integer sein.
15	Eine Nicht-Funktion kann nicht aufgerufen werden.
16	Ein Element kann nicht von etwas angefordert werden, das kein Array ist.
17	Nicht genug Indizes für die Dimension dieses Arrays.
18	Zu viele Indizes für die Dimension dieses Arrays.
19	Sie können einem Wert nicht auf einmal ein ganzes Array zuweisen.
20	Ein ganzes Array kann nicht auf einmal in einem Ausdruck verwendet werden.
21	Ein Ausdruck kann nicht als ein ByRef-Parameter übergeben werden.
22	Doppelter Identifier.
23	Der Back-End-Code-Generator versagte.
24	Zweideutiger Aufruf einer überladenen Methode.
25	Mehrfache Vererbung ist nicht erlaubt.
26	Kann keine Instance von einem Interface erzeugen.
27	Kann eine Klasse nicht implementieren als wäre es ein Interface.
28	Vererbung funktioniert nur bei Klassen.
29	Diese Klasse implementiert das angegebene Interface nicht vollständig.
30	Event-Handler können sich nicht außerhalb einer Klasse befinden.
31	Das Ergebnis eines Funktionsaufrufs darf nicht ignoriert werden.
32	Das „Self“-Schlüsselwort darf nicht außerhalb einer Klasse verwendet werden.
33	Das „Me“-Schlüsselwort darf nicht außerhalb einer Klasse verwendet werden.
34	Kann aus einer „Sub“-Routine keinen Wert zurückliefern.
35	Hier wird ein Exception-Objekt benötigt.
36-39	Überholt.
40	Destruktoren dürfen keine Parameter haben.
41	Das „Super“-Schlüsselwort darf nicht außerhalb einer Klasse verwendet werden.
42	Das „Super“-Schlüsselwort darf nicht in einer Klasse verwendet werden, die keine Eltern hat.

RbScriptAlreadyRunningException Fehler

Tritt auf, wenn der Anwender versucht, den Quelltext eines [RbScripts](#) zu modifizieren, während dieses läuft oder wenn Sie versuchen, das Context-Objekt zu ändern, während das Skript läuft. Dies darf man nicht machen.

Übergeordnete Klasse

[RuntimeException](#).

Siehe auch

[RbScript](#) Klasse, [Exception Block](#).

RBVersion Konstante

Liefert die größere und kleinere Versionsnummer von REALbasic.

Syntax

Ergebnis=RBVersion

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Konstante, die die große und kleine Versionsnummer angibt.

Beispiel

Folgende Zeile zeigt die Version des laufenden REALbasic an:

```
MsgBox "Es läuft gerade die Version "+Str(RBVersion)+" von REALbasic!"
```



Hinweis

Wenn Sie Ihre Applikation erzeugen, können Sie Versions-Informationen über Ihre Applikation in der Dialogbox „Compiler-Einstellungen“ eingeben. Diese Informationen werden in der „vers“-Resource der Applikation gespeichert. Sehen Sie im Entwicklerhandbuch im Kapitel „Stand-Alone-Programme erzeugen“ nach.

Sie können die RBVersion-Konstante in einem #If Ausdruck verwenden, um festzustellen, ob eine bestimmte REALbasic-Version benutzt wird und entsprechend speziellen Code dafür compilieren.

Beispiel

Das folgende Beispiel enthält Code, der nur für REALbasic ab Version 5 vorgesehen ist.

```
#If RBVersion>5
//hier 5.0-Code einsetzen
#endif
```

Siehe auch

[TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetWin32](#), [DebugBuild](#), [RBVersionString](#) Konstanten; [#If](#) Ausdruck.

RBVersionString Konstante

Liefert die Version von REALbasic als [String](#).

Syntax

Ergebnis=**RBVersion**

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Version von REALbasic, die gerade läuft.

Hinweis

Wenn Sie Ihre Applikation erzeugen, können Sie Versions-Informationen über Ihre Applikation in der Dialogbox „Applikation erzeugen“ eingeben. Diese Informationen werden in der „vers“-Resource der Applikation gespeichert. Sehen Sie im Entwicklerhandbuch im Kapitel „Stand-Alone-Programme erzeugen“ nach.

Beispiel

Folgende Zeile zeigt die Version des laufenden REALbasic an:

```
MsgBox "Es läuft gerade die Version "+Rbversionstring+" von REALbasic!"
```



Siehe auch

[TargetMacOS](#), [Target68K](#), [TargetCarbon](#), [TargetPPC](#), [TargetWin32](#), [DebugBuild](#), [RBVersionString](#) Konstanten.

REALdatabase Klasse

Die **REALdatabase** Klasse bietet Zugriff auf eine REALdatenbank. Sie ersetzt die ursprüngliche REALdatenbank-Datenquelle, die bis REALbasic Version 5.0 ausgeliefert wurde. Eine Datenbank, in dem nun älteren Format, kann weiterhin über die Klasse [REALdatabaseOldFormat](#) verwendet werden.

Die Standard- und Professional-Variante von REALbasic bieten volle Unterstützung der REALdatenbank-Engine. Plugins für andere Datenbank-Engines werden nur mit der Professional-Variante von REALbasic unterstützt.

Benutzen Sie die **REALdatabase** Klasse, um eine REALdatenbank aus Ihrem Programm heraus zu öffnen. Sie können zudem ein REAL-Datenbank über den Menüpunkt „Ablage/Datenquelle hinzufügen“ erzeugen.

Übergeordnete Klasse

Database Klasse.

Eigenschaften

Name	Typ	Beschreibung
DatabaseFile	<u>FolderItem</u>	Ein <u>FolderItem</u> der REALdatenbank-Datei.

Methoden

Name	Parameter	Beschreibung
CreateDataBaseFile		Erzeugt eine neue REALdatenbank. Verwendet die DatabaseFile-Eigenschaft als <u>FolderItem</u> für das Erzeugen der Datei. Liefert <u>True</u> , wenn die Datenbank erzeugt wurde, andernfalls <u>False</u> .
GetSchemaData		Liefert einen <u>String</u> , der das aktuelle Datenbank-Schema enthält. GetSchemaData schlägt fehl, wenn momentan eine Transaktion ausgeführt wird.
Last_RowID	tablename als <u>String</u>	Liefert ein <u>Integer</u> mit der letzten <u>_rowID</u> der angegebenen Tabelle.
ReplaceSchemaData	NewSchema als <u>String</u>	Ersetzt das aktuelle Datenbank-Schema mit dem angegebenen. ReplaceSchemaData schlägt fehl, wenn momentan eine Transaktion ausgeführt wird. Achtung: Diese Änderung kann nicht rückgängig gemacht werden. Verwenden Sie sie also mit Vorsicht!

Hinweis

REALdatenbank-Tabellen können hinzugefügt, verworfen oder in unterschiedlichster Weise modifiziert werden (Spalten oder Indizes können hinzugefügt oder verworfen werden), ohne existierende Daten zu verlieren oder versehentlich zu duplizieren.

Die neue **REALdatabase** unterstützt eine Untermenge von SQL/92 und SQL/99 (Details weiter unten), einschließlich Abfragen, die ein Self-Join oder Aggregatfunktionen beinhalten. Mit Ausnahme kleiner Erweiterungen ist die Syntax der REALdatenbank-Engine voll SQL-konform. Es werden sogar die Standard-SQL-Fehler-Codes zurückgegeben.

Jede neue REALdatenbank-Tabelle besitzt eine spezielle ID-Spalte, die „_rowid“ heißt. Diese enthält eine eindeutige ID für eine Zeile. Die ID wird automatisch fortgeführt und kann als komfortables Join-Field für relationale Datenbanken dienen. Ein Aufruf eines SQL-Selects liefert ein dynamisches RecordSet; Sie können sich vorwärts, rückwärts und an das Ende beziehungsweise den Anfang bewegen.

Die **REALdatabase**-Engine unterstützt Transaktionen für Änderungen am Datenbank-Schema und den Daten. Eine Transaktion wird automatisch initiiert, wenn Sie Änderungen an der Datenbank vornehmen und müssen über den Aufruf der Commit- oder Roll-Back-Methode der Database-Klasse abgeschlossen werden.

Rückgabewerte

Die **REALdatabase**-Engine setzt die Error-Eigenschaft der Database-Klasse und schreibt Rückgabewerte in die ErrorCode-Eigenschaft und die entsprechende Nachricht in der ErrorMessage-Eigenschaft. Wenn die Error-Eigenschaft auf False gesetzt ist, ist der Error-Code 0 und die Nachricht „Success“. Ist die Error-Eigenschaft True, können Sie den entsprechenden Fehler-Code und -Nachricht entnehmen. SQL-Fehler-Codes werden in eckigen Klammern der Nachricht angehängt.

SQL Fehler-Code	Nachricht
02000	No records found.
01004	Warning: String truncated on right.
01S09	Warning: Invalid SQL keyword.
07000	Error: Dynamic SQL error.
42S01	Error: Table already exists.

SQL Fehler-Code	Nachricht
42S02	Error: Table not found.
42S11	Error: Index already exists.
42S12	Error: Index not found.
42S21	Error: Column already exists.
42S22	Error: Column not found.
21S01	Error: Insert value list does not match column list.
22000	Error: Data exception.
22023	Error: Invalid parameter.
25000	Error: Invalid transaction state.
0A000	Error: Unsupported SQL feature.
72000	Error: Internal SQL processor error.

Datentypen

Die **REALdatabase**-Engine unterstützt folgende Datentypen:

Typ	Code	Kommentar
Integer	3	32-bit signed integer.
VarChar	5	Text mit bis zu 2 hoch 31 Bytes ^a .
Double	7	64-bit Fließkommezahl.
Date	8	Datum im Format, JJJJ-MM-TT. Die SQLDate-Eigenschaft der <u>Date</u> Klasse unterstützt dieses Format.
Time	9	Zeit im Format HH:MM:SS.
TimeStamp	10	Zeitstempel im Format JJJJ-MMM-TT HH:MM:SS. Die SQLDateTime-Eigenschaft der <u>Date</u> Klasse unterstützt das Format.
Boolean	12	Boole'scher Wert; TRUE oder FALSE ^(b) .
Binary	14	Binärdaten von bis zu 2 hoch 31 Bytes
String	18	Text von bis zu 2 hoch 31 Bytes ^(a) .

a. Textfelder einer REALdatenbank speichern zusätzlich Kodier-Informationen zum Text. Wenn Sie Einträge einfügen, wird die Kodierung mit dem Feld gespeichert. Wenn Sie das Feld abrufen, wird die gespeicherte Kodierung für diesen Text verwendet.

b. Boolean ist kein SQL-Datentyp und TRUE und FALSE sind keine SQL-Schlüsselwörter. Diese gehören zu REALbasics Erweiterungen zu SQL.

Eine REALdatenbank erzeugen

Sie können eine REALdatenbank in der IDE oder über den Menüpunkt „Ablage/Datenquelle hinzufügen“ erzeugen. REALbasic wird dann einen Speichern-Dialog anzeigen, in dem Sie der Datenbank einen Namen geben und festlegen können, wo diese gespeichert werden soll.

Klicken Sie OK, wird REALbasic die Datenbank dem Projekt hinzufügen. Den Datenbank-Eintrag können Sie doppelklicken und Tabellen und Felder hinzufügen. Wenn Sie Daten hinzugefügt haben, können Sie über den Tabellen-Dialog Ihre Daten einsehen. Für weitere Informationen beachten Sie das Kapitel Datenbanken im Benutzerhandbuch.

Keine Tabelle kann größer als 2 GB werden. Wenn die Datenbank in einem VirtualVolume abgelegt wird, darf die Gesamtgröße der Datenbank 2 GB nicht übersteigen. Row IDs werden automatisch erhöht und niemals wiederverwendet. Das bedeutet, daß Sie mehr als 2 hoch 32 (über 4 Milliarden) Einträge in einer Tabelle speichern können.

Lesen Sie das Kapitel „Datenbanken mit REALbasic erzeugen“ im Benutzerhandbuch, wenn Sie mehr über die Fähigkeiten einer REALdatenbank erfahren wollen.

Die Methoden CreateDatabaseFile und Connect sollten statt NewREALdatabase und OpenREALdatabase verwendet werden, um REALdatenbanken zu erzeugen oder zu öffnen. OpenREALdatabase wird in zukünftigen Versionen von REALbasic nicht länger unterstützt.

Beispiele

Das folgende Beispiel erzeugt eine REALdatenbank:

```
Dim db as REALdatabase
Dim f as FolderItem
f=New FolderItem("mydb")
db=New REALdatabase
db.databaseFile=f
If db.CreateDatabaseFile then
    //weiter mit Datenbank-Operationen...
else
    MsgBox "Datenbank wurde nicht erzeugt!"
end if
```

Hier wird eine existierende Datenbank geöffnet:

```
Dim dbFile as FolderItem
Dim db as REALdatabase
db=New REALdatabase
dbFile = GetFolderItem("Pubs")
db.DatabaseFile=dbFile
If db.Connect() then
    //weiter mit Datenbank-Operationen...
else
    Beep
    MsgBox "Die Datenbank konnte nicht geöffnet werden."
end if
```

Das Einfügen eines Eintrags in eine Tabelle:

```
Dim dbFile as FolderItem
Dim db as REALdatabase
db=New REALdatabase
Dim rs As New Recordset
dbFile=New FolderItem("Employees")
db.databaseFile=dbFile
if db.error then
    MsgBox db.errorMessage
else
    db.sqlexecute ("Insert into Employees (Name,Job,YearJoined) Values "_
        +"('Dr.Strangelove','Advisor',1962)")
    If db.error then
        MsgBox db.errorMessage
    else
        db.Commit
    end if
end if
```

Siehe auch

[OpenCSVCursor](#), [OpenDBFCursor](#), [OpenDTFDatabase](#), [OpenOracleDatabase](#) Funktionen; [Database](#), [RecordSet](#) Klassen.

REALdatabaseOldFormat Klasse

Die **REALdatabaseOldFormat** Klasse ist obsolet. Sie unterstützt die ursprüngliche REALdatenbank, die vor REALbasic in der Version 5.5 verwendet wurde. Mit Version 5.5 wird diese Technologie durch eine neue Implementierung der REALdatenbank ersetzt. Diese Klasse verwendet die alte Implementation und ist nur aus Kompatibilitätsgründen vorhanden.

Siehe auch

[REALdatabase](#) Klasse.

RecordSet Klasse

Ein **RecordSet** ist eine Gruppe von [Database](#)-Records (Datensätzen). Verwenden Sie die Methoden und Eigenschaften der **RecordSet**-Klasse, um durch diesen Satz von Datensätzen zu navigieren und einzelne Datensätze zu bearbeiten und zu aktualisieren. **RecordSet** bieten die gleiche Funktionalität wie [DatabaseCursor](#) in den ersten Versionen von REALbasic. Funktionen jedoch, die in diesen Versionen einen [DatabaseCursor](#) lieferten, liefern nun ein **RecordSet**.

DatabaseCursor ist nur aus Gründen der Abwärtskompatibilität in REALbasic noch vorhanden.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
BOF	<u>Boolean</u>	Anfang des Sets von Datensätzen.
EOF	<u>Boolean</u>	Ende des Sets von Datensätzen.
FieldCount	<u>Integer</u>	Anzahl der Felder im RecordSet
RecordCount	<u>Integer</u>	Anzahl der Datensätze im RecordSet . Derzeit unterstützt in OpenBase, MySQL, PostgreSQL, und 4D Plugins. Bei Datenbanken, die diese Funktion nicht unterstützen, liefert RecordCount -1.

Methoden

Name	Parameter	Beschreibung
Close		Schließt einen offenen RecordSet . Schließen Sie die Applikation, wird REALbasic das RecordSet implizit schließen.
DeleteRecord		Löscht den Eintrag, auf den das RecordSet verweist.
Edit		Rufen Sie diese Methode auf, bevor Sie Änderungen an dem Eintrag vornehmen.
Field	Name als <u>String</u>	Liefert den Inhalt des Felds der Zeile, auf die der Eintrag verweist. Rückgabetyt ist <u>DatabaseField</u> .
IdxField	Index als <u>Integer</u>	1-basiertes Array. Liefert das n ^{te} Feld im RecordSet . Rückgabetyt ist <u>DatabaseField</u> .
MoveFirst		Verschiebt den Zeiger auf den ersten Eintrag im RecordSet . Dies wird nur von REALdatabase-Datenquellen unterstützt.
MoveLast		Verschiebt den Zeiger auf den letzten Eintrag im RecordSet . Dies wird nur von REALdatabase-Datenquellen unterstützt.
MoveNext		Verschiebt den Zeiger auf den nächsten Eintrag im RecordSet . Dies wird von allen Datenquellen unterstützt.
MovePrevious		Verschiebt den Zeiger auf den vorherigen Eintrag im RecordSet . Dies wird nur von REALdatabase-Datenquellen unterstützt.
Update		Aktualisiert das RecordSet , damit es die Änderungen des Eintrags wiedergibt, auf das es verweist.

Hinweise

Wenn Sie REALbasic als Frontend in einer Mehrbenutzerumgebung verwenden, sollten Sie folgendes beachten: Die Edit-Methode wird versuchen, den aktuellen Datensatz für andere Anwender zu sperren. Wenn ein Anwender versucht, auf einen gesperrten Datensatz zuzugreifen, liefert die Error-Eigenschaft der Database-Klasse True.

Die MoveNext-Methode entspermt den aktuellen Datensatz, wenn Sie diesen zuvor mit der Edit-Methode gesperrt hatten. Es ruft außerdem die Close-Methode der Klasse auf und die Close-Methode der Database-Klasse.

Wenn Sie OpenBase als Datenquelle verwenden und die **RecordSet**-Klasse mit Edit und Update zum Aktualisieren eines Datensatzes verwenden, müssen Sie die “_rowid”-Spalte der Tabelle hinzufügen. Wenn Sie dies nicht machen, ist OpenBase nicht in der Lage, den Datensatz zu finden, um ihn zu aktualisieren.

Beispiele

Folgendes Beispiel ändert ein Feld und aktualisiert die Tabelle.

```
Dim db as Database
dim rs as RecordSet
.
.
rs = db.SQLSelect("select * from employees where ID=01")
rs.Edit
rs.IdxField(2).setString ("VP")
```

```
rs.Update
db.Commit
```

Die folgende Methode stellt ein **RecordSet** in einer Listbox dar. Es benutzt die Namen- und StringValue-Eigenschaft der DatabaseField-Klasse, um die Kopfzeilen und Daten darzustellen.

```
Sub DisplayRecordSet (rs as RecordSet)
    Dim i as Integer
    Dim v as String
    Listbox1.DeleteAllRows
    Listbox1.Columncount = rs.FieldCount
    Listbox1.AddRow rs.IdxFeld(1).Name
    Listbox1.CellBold(Listbox1.LastIndex, 0) = True
    For i = 2 to rs.FieldCount
        Listbox1.Cell(Listbox1.LastIndex, i - 1) = rs.IdxFeld(i).Name
        Listbox1.CellBold(Listbox1.LastIndex, i - 1) = True
    Next
    While Not rs.eof
        v = rs.IdxFeld(1).StringValue //erste Spalte anzeigen
        Listbox1.AddRow v
        For i = 2 to rs.fieldcount
            Listbox1.Cell(Listbox1.LastIndex, i - 1) = rs.IdxFeld(i).StringValue
        next
        rs.MoveNext
    Wend
End Sub
```

Siehe auch

Database, DatabaseField, DatabaseRecord, ODBCDatabase ADSP4DServer, TCPIP4DServer, PostgreSQLDatabase, OpenBaseDatabase Klassen.

Rectangle Steuerelement



Zeichnet ein Rechteck.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
<i>BorderWidth</i>	<u>Integer</u>	Die Breite des Rechteck-Rahmens in Pixeln.
<i>BottomRightColor</i>	<u>Color</u>	Die Farbe der unteren und rechten Seite des Rechteck-Rahmens.
<i>FillColor</i>	<u>Color</u>	Die Füllfarbe des Rechtecks.
<i>LeftTopColor</i>	<u>Color</u>	Die Farbe der linken und oberen Seite des Rechteck-Rahmens.

Events

Name	Parameter	Beschreibung
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Rectangle an der gelieferten Stelle x,y gedrückt. Wenn Sie MouseDown auswerten möchten, liefern Sie <u>True</u> zurück.
<i>MouseDrag</i>	x als <u>Integer</u> , y als <u>Integer</u>	Der Mausknopf wurde innerhalb des Rectangle -Steuerelements gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u>	Die Maustaste wurde innerhalb der Region des Rectangle an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht <u>True</u> im MouseDown-Event liefern.

Beispiel

Sie können einem Rechteck durch die globalen Funktionen `LightBevelColor`, `DarkBevelColor` und `FillColor` ein eingedrücktes oder ein erhöhtes Aussehen geben.

```
//Eingedrücktes Erscheinungsbild
Me.BorderWidth=2
Me.TopLeftColor=DarkBevelColor
Me.BottomRightColor=LightBevelColor
Me.FillColor=FillColor

//Erhöhtes Erscheinungsbild
Me.BorderWidth=2
Me.TopLeftColor=LightBevelColor
Me.BottomRightColor=DarkBevelColor
Me.FillColor=FillColor
```

Siehe auch

[RoundRectangle](#) Steuerelement; [RectControl](#) Klasse.

RectControl Klasse

RectControl ist die Basisklasse für die meisten anderen Steuerelement-Klassen. Dies bedeutet, daß alle Steuerelemente die Eigenschaften des **RectControl** übernehmen. **RectControls** können nicht direkt angelegt oder verändert werden, stattdessen können Sie Unterklassen von **RectControl** (wie z.B. [PushButtons](#), [EditFields](#), [ListBoxen](#), ...) anlegen oder verändern.

Übergeordnete Klasse

[Control](#)

Da dies ein Steuerelement ([Control](#)) ist, schauen Sie bei der [Control](#)-Klasse wegen der anderen Eigenschaften und Events nach, die alle [Control](#)-Objekte gemeinsam haben.

Eigenschaften

Name	Art	Beschreibung
Active	Boolean	Zeigt an, ob das RectControl aktiv ist. Active ist False , wenn das Fenster des RectControls nicht im Vordergrund ist. Wenn ein Fenster deaktiviert ist, sind seine Steuerelemente automatisch deaktiviert, außer AutoDeactivate ist auf False gesetzt.
<i>AutoDeactivate</i>	Boolean	Legt fest, ob das Steuerelement deaktiviert werden soll (unter Mac OS), wenn das Eltern-Fenster deaktiviert ist. Voreinstellung ist True.
<i>Balloon Help</i>	String	Ballon-Hilfe-Text für das Steuerelement in Mac OS Classic und Windows oder das erweiterte Help Tag in Mac OS X.
<i>Parent</i>	Control	Wird zum Setzen und Ermitteln des Eltern-Steuerelements oder -Fenster des Steuerelements verwendet. Liefert Nil, falls das Elter das Fenster ist. Weisen Sie Nil zu, um das Fenster zum Elter des Steuerelements zu machen, auch wenn es sich innerhalb anderer Steuerelemente befindet. Das Verhalten des Kindes des Steuerelements in der IDE (siehe Steuerelemente-Hierarchie bei den Hinweisen weiter unten) reflektiert den Zustand des Elter. Wenn sich das Eltern-Steuerelement aus irgendwelchen Gründen in einem anderen Fenster befindet, tritt eine InvalidParentException auf.
<i>DisableBalloonHelp</i>	Nachricht als String	Nachricht, die erscheint, wenn die Ballon-Hilfe aktiv ist und der Anwender sich mit der Maus über dem Menüeintrag befindet, während dieser nicht aktiviert ist.
<i>Enabled</i>	Boolean	Entscheidet, ob das Steuerelement aktiviert sein soll, wenn das Fenster geöffnet wird, in dem es sich befindet. Ein deaktiviertes Steuerelement kann nicht angeklickt werden und nicht den Fokus bekommen.
<i>HelpTag</i>	String	Text der Hilfe-Nachricht, die als Windows-Tip oder Carbon/Mac OS X Help Tag angezeigt wird. Es funktioniert in der Carbon-Version der REALbasic-IDE und in alle Carbon-Applikationen. Der Tip wird angezeigt, wenn der Anwender den Mauszeiger auf das Steuerelement platziert und ihn dort läßt.

Name	Art	Beschreibung
<i>Height</i>	<u>Integer</u>	Die Höhe des Steuerelements in Pixeln.
<i>Left</i>	<u>Integer</u>	Die linke Seite des Steuerelements (die Koordinaten sind relativ zum Fenster)
<i>LockBottom</i>	<u>Boolean</u>	Entscheidet, ob die untere Seite des Steuerelements immer im gleichen Abstand zum unteren Fensterrand dargestellt werden soll.
<i>LockLeft</i>	<u>Boolean</u>	Entscheidet, ob die linke Seite des Steuerelements immer im gleichen Abstand zum linken Fensterrand dargestellt werden soll. LockLeft ist unwirksam, wenn nicht auch LockRight <u>True</u> ist.
<i>LockRight</i>	<u>Boolean</u>	Entscheidet, ob die rechte Seite des Steuerelements immer im gleichen Abstand zum rechten Fensterrand dargestellt werden soll.
<i>LockTop</i>	<u>Boolean</u>	Entscheidet, ob die obere Seite des Steuerelements immer im gleichen Abstand zum oberen Fensterrand dargestellt werden soll. LockTop ist unwirksam, wenn nicht auch LockBottom <u>True</u> ist.
<i>MouseCursor</i>	<u>MouseCursor</u>	Der Mauszeiger, der dargestellt wird, solange sich der Mauszeiger im Steuerelement befindet und die MouseCursor-Eigenschaft der Application- und der Window-Klasse Nil ist. Wenn die <u>Application</u> -Klasse MouseCursor nicht Nil ist, oder die MouseCursor-Eigenschaft der Window-Klasse nicht Nil ist, werden die MouseCursor-Eigenschaften aller Steuerelemente ignoriert. Mögliche Namen sind: WatchCursor, IBeamCursor und ArrowCursor. MouseCursors können ebenso aus Resource-Dateien zugewiesen werden, die dem Projekt hinzugefügt wurden.
<i>Top</i>	<u>Integer</u>	Die obere Seite des Steuerelements (die Koordinaten sind relativ zum Fenster)
<i>Visible</i>	<u>Boolean</u>	Entscheidet, ob das Steuerelement beim Öffnen des Fensters angezeigt wird.
<i>Width</i>	<u>Integer</u>	Die Breite des Steuerelements in Pixeln.
<i>Window</i>	<u>Window</u>	Das Eltern-Fenster des RectControls.

Events

Name	Parameter	Beschreibung
<i>DropObject</i>	Obj als <u>DragItem</u>	Das durch Obj vertretene Objekt wurde auf das Steuerelement gezogen.
<i>MouseEnter</i>		Der Mauszeiger hat den Bereich des Steuerelements betreten.
<i>MouseExit</i>		Der Mauszeiger hat den Bereich des Steuerelements verlassen.
<i>MouseMove</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Der Mauszeiger hat sich innerhalb des Steuerelements auf diese Koordinaten bewegt.

Methoden

Name	Parameter	Beschreibung
<i>AcceptFileDrop</i>	FileType als <u>String</u>	Erlaubt das Darauf-Ziehen (Droppen) von Dokumenten mit dem Dateityp <i>FileType</i> auf dieses Steuerelement. <i>FileType</i> muß ein Dateityp sein, den Sie in der Dateitypen-Dialogbox definiert haben.
<i>AcceptMacData</i>	Type als <u>String</u>	Erlaubt das Ziehen von Daten des angegebenen Typs auf dieses Steuerelement. <i>Type</i> ist ein vierbuchstabiger Resource-Code, z.B. „snd“ oder „TEXT“.
<i>AcceptPictureDrop</i>		Erlaubt das Ziehen von Bildern auf dieses Steuerelement.
<i>AcceptTextDrop</i>		Erlaubt das Ziehen von Text auf dieses Steuerelement.
<i>NewDragItem</i>	Left als <u>Integer</u> , Top als <u>Integer</u> , Width als <u>Integer</u> , Height als <u>Integer</u>	Legt das Drag-Rechteck fest (basierend auf den übergebenen Koordinaten) das erscheinen wird, wenn der Anwender etwas von dem Steuerelement draggt. Liefert ein neues <u>DragItem</u> -Objekt. Left und Top sind die Koordinaten der linken, oberen Ecke des Drag-Rechtecks und Width und Height sind die Größe des Drag-Rechtecks. Sehen Sie bei der globalen <u>NewDragItem</u> -Funktion nach, welche eine äquivalente Funktionalität bietet, genau so wie das Draggen auf ein Fenster.

Name	Parameter	Beschreibung
<i>Refresh</i>	[LöschHintergrund als <u>Boolean</u>]	Zeichnet den gesamten Inhalt des Steuerelements neu. Wenn der Parameter LöschHintergrund <u>True</u> ist, wird der Hintergrund erst gelöscht, bevor das Steuerelement neu gezeichnet wird. Dies ist die Standardeinstellung.
<i>RefreshRect</i>	X als <u>Integer</u> , Y als <u>Integer</u> , Width als <u>Integer</u> , Height als <u>Integer</u> , [LöschHintergrund als <u>Boolean</u>]	Zeichnet nur die angegebenen Bereiche des Steuerelements neu. Wenn der Parameter LöschHintergrund <u>True</u> ist, wird der Hintergrund erst gelöscht, bevor das Steuerelement neu gezeichnet wird. Dies ist die Standardeinstellung.
<i>SetFocus</i>		Setzt, falls machbar, den Fokus auf das Steuerelement. KeyDown-Events werden an das Steuerelement weitergeleitet. Bei <u>EditFields</u> wird die Einfügemarke in dieses Steuerelement bewegt. Wenn das Steuerelement den Fokus auf der Plattform, auf der die Applikation läuft, nicht bekommen kann, passiert bei SetFocus nichts. Auf dem Macintosh können nur <u>EditFields</u> und <u>ListBoxes</u> den Fokus bekommen. Unter Windows bekommen <u>EditFields</u> , <u>PushButtons</u> , <u>CheckBoxes</u> , <u>ListBoxes</u> und <u>Sliders</u> den Fokus. Jedoch nur <u>EditFields</u> und <u>ListBoxes</u> besitzen GetFocus und LostFocus Events. Die SetFocus-Methode der <u>Window</u> -Klasse oder die <u>ClearFocus</u> -Methode kann verwendet werden, um den Fokus vom Steuerelement, das gerade den Fokus hat, zu entfernen, so daß kein Steuerelement mehr den Fokus hat.

Die Steuerelemente-Hierarchie

In manchen Fällen erzeugen Sie Teile Ihres Interfaces, indem Sie einige Steuerelemente innerhalb eines anderen Steuerelements platzieren. Zum Beispiel verwenden Sie das GroupBox-Steuerelement, um andere Steuerelemente zu organisieren, normalerweise RadioButtons oder mögliche CheckBoxen. Die Steuerelemente TabPanel und PagePanel sind ebenso dazu gedacht, andere Steuerelemente zu beinhalten. Ein gutes Beispiel ist die „Suchen und Ersetzen“-Box aus dem REALbasic Tutorial. In diesem Fall befinden sich die Steuerelemente StaticText, EditField und PushButton innerhalb eines TabPanels.

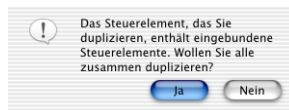
Das Steuerelement, das die anderen beinhaltet, wird Eltern-Steuerelement genannt und die Steuerelemente, die sich komplett innerhalb dessen Grenzen befinden, sind die Kinder-Steuerelemente. Dies funktioniert automatisch, wenn Sie die Steuerelemente in dieser Reihenfolge erzeugen: Eltern-Kind. Erzeugen Sie also zuerst das Eltern-Steuerelement, das die anderen beinhaltet, erzeugen Sie dann die Kinder-Steuerelemente oder duplizieren Sie vorhandene Kinder-Steuerelemente und halten Sie diese innerhalb des Eltern-Steuerelements.

Sie können die Parent-Eigenschaft der **RectControl**-Klasse verwenden, um das Elternobjekt eines vorhandenen Steuerelements entweder zu ermitteln oder festzulegen. Wenn Sie ein Kind vor seinem Elternobjekt erzeugen, können Sie sein Elternobjekt über die Parent-Eigenschaft festlegen.

Falls ein Steuerelement nicht vollständig von einem anderen umschlossen wird, wird es nicht automatisch zu einem Kind; es überlappt das andere Steuerelement einfach nur.

Wenn Sie ein Kind aus seinem Elternobjekt herausverschieben, ist es nicht mehr länger Kind dieses Steuerelements. Wenn Sie es komplett in ein anderes Steuerelement verschieben, wird es Kind dieses Steuerelements.

Wenn Sie ein Elternobjekt kopieren, können Sie dessen Kinder ebenfalls kopieren. Wenn Sie die Kopieraktion ausführen, zeigt REALbasic eine Warnbox an, in welcher Sie die Wahl haben, alle Kinder-Steuerelemente zu kopieren.



Sie können mehr als eine Verschachtelungstiefe erzeugen. Zum Beispiel können Sie eine GroupBox in ein TabPanel platzieren und dann verschiedene RadioButtons in diese GroupBox platzieren. In diesem Fall ist die GroupBox das Elternobjekt des RadioButtons und das TabPanel ist das Elternobjekt der GroupBox. Damit das automatisch funktioniert, müssen Sie diese in der Reihenfolge erzeugen, die ihrer Hierarchie entspricht: Zuerst erzeugen Sie das TabPanel, dann die GroupBox und dann die RadioButtons.

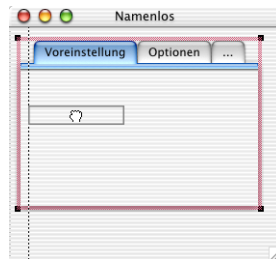
Hinweis: Natürlich muß sich das Elternobjekt im gleichen Fenster wie die Kinder-Steuerelemente befinden. Wenn Sie versuchen, das Elternobjekt eines Steuerelementes zu bekommen, dessen Elternobjekt sich in einem anderen Fenster befindet, erhalten Sie einen InvalidParentException Fehler.

Wenn ein Steuerelement sich nicht komplett in einem anderen Steuerelement befindet, ist sein Elternobjekt sein Fenster. Da die Parent-Eigenschaft eines Steuerelements ein **Control** ist, wird die Parent-Eigenschaft Nil – denn ein Window ist nicht von der **Control**-Klasse abgeleitet. Bevor Sie versuchen, auf die Parent-Eigenschaft eines Steuerelements zuzugreifen, sollten Sie zuerst testen, ob sie Nil ist. Wenn Sie das nicht machen, erhalten Sie einen NilObjectException Fehler, wenn das Elternobjekt des Steuerelements das Eltern-Fenster ist.

Wenn sich ein Steuerelement innerhalb eines anderen befindet, Sie aber nicht wollen, daß es sich wie ein Kind davon verhält, setzen Sie die Parent-Eigenschaft auf Nil. Das zerstört die normale Eltern-Kind-Beziehung, die normalerweise durch die Steuerelemente-Hierarchie aufrecht erhalten wird.

Steuerelemente-Hierarchie Besonderheiten

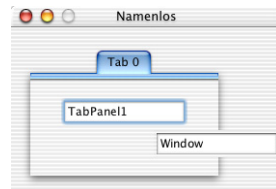
Um automatisch die Vorzüge der Steuerelemente-Hierarchie genießen zu können, erzeugen Sie das Eltern-Steuerelement zuerst und fügen dann die Kinder-Steuerelemente hinzu. Wenn Sie auf diese Weise ein Kind einem Elternobjekt hinzufügen, erscheint ein Rahmen um das Elternobjekt. Dies signalisiert Ihnen optisch, daß Sie tatsächlich dabei sind, ein Kind einem Elternobjekt hinzuzufügen. In folgender Abbildung wird zum Beispiel ein StaticText als ein Kind einem TabPanel hinzugefügt.



Ein Rahmen umgibt das Elternobjekt auch immer, wenn Sie ein vorhandenes Kinder-Steuerelement auswählen. Sie können den Rahmen deaktivieren, indem Sie die entsprechende Voreinstellung unter **REALbasic/Einstellungen** unter **Fenster-Editor** deaktivieren. Wenn Sie diese Funktion aktiviert lassen, können Sie die Farbe des Rahmens im **Farben-Voreinstellungen-Dialog** einstellen.

Wenn Sie ein Kinder-Steuerelement duplizieren und das Duplikat im Elternobjekt belassen, ist es automatisch ein Kind. Wenn Sie das Duplikat jedoch aus dem Elternobjekt heraus verschieben, ist es kein Kind mehr. Ein Steuerelement muß sich komplett innerhalb des Elternobjekts befinden, damit es als ein Kind anerkannt wird.

Im folgenden Beispiel zeigen beide EditFields die Identität ihrer Eltern an. Das untere wurde aus dem oberen dupliziert, wurde jedoch aus dem TabPanel heraus verschoben. Es ist kein Kind mehr.



Wenn Sie ein Kind aus dem Zuständigkeitsbereich seines Elternobjekts verschieben, wird der Rahmen um das Elternobjekt ausgeschaltet, um anzuzeigen, daß sich das Kind nicht mehr unter elterlicher Kontrolle befindet.

Wenn Sie das untere EditField zurück in das TabPanel verschieben, wird es wieder ein Kind des TabPanel.

Das Verschieben eines Eltern-Steuerelements bewegt seine Kinder ebenso. Im folgenden Beispiel wurde das oben gezeigte TabPanel nach unten verschoben. Es nimmt das obere EditField mit sich, läßt das untere aber zurück.



- Das Löschen eines Eltern-Steuerelements löscht alle Kinder-Steuerelemente.
- Das Verstecken eines Eltern-Steuerelements versteckt alle Kinder, erhält aber den zuvor bestehenden Sichtbarkeits-Status aller Kinder bei.
- Das Anzeigen eines Elternobjekts zeigt nur die Kinder an, deren Sichtbarkeit eingestellt ist.
- Das Deaktivieren des Elternobjekts deaktiviert alle seine Kinder, erhält aber den zuvor bestehenden Sichtbarkeits-Status aller Kinder bei. In der IDE deaktiviert das Deaktivieren eines Containers alle Kinder-Steuerelemente, aktualisiert aber nicht die Enabled-Eigenschaft.

Hier ist eine deaktivierte GroupBox in der IDE, in welcher zwei RadioButtons und zwei EditFields Kinder sind. Wenn die GroupBox deaktiviert ist, sind alle vier Kinder deaktiviert. Da alle Kinder deaktiviert sind, wenn das Elternobjekt deaktiviert ist, kann man in der Applikation die EditFields nicht ansteuern.



Das Aktivieren eines Eltern-Steuerelements aktiviert nur die Kinder-Steuerelemente, deren Enabled-Eigenschaft gesetzt ist. Im folgenden Beispiel ist nur bei den Kinder-Steuerelementen der unteren Reihe die Enabled-Eigenschaft gesetzt. Wenn die GroupBox aktiviert ist, bleiben die Steuerelemente der oberen Reihe deaktiviert.



Das Deaktivieren der GroupBox deaktiviert wie erwartet alle Kinder-Steuerelemente.

Die Composite-Eigenschaft der Window-Klasse

Unter Mac OS X gibt es ein Darstellungsproblem, das bestimmte Arten von **RectControls** betrifft, die in Metal- oder Drawer-Fenstern oder auf anderen Steuerelementen in einem Dokument-Fenster platziert wurden. Nicht-rechteckige Steuerelemente wie PushButtons oder RadioButtons stellen ihren rechteckigen Umriss als einen weißen Hintergrund dar. Um das Problem zu lösen, können Sie die Composite-Eigenschaft des Eltern-Fensters einschalten.

Siehe auch

Control Klasse.

RectShape Klasse

Zeichnet ein (zweidimensionales) Rechteck in einer Vektorgrafikumgebung.

Übergeordnete Klasse

Object2D

Eigenschaften

Name	Typ	Beschreibung
<i>Height</i>	<u>Double</u>	Die Höhe des Rechtecks.
<i>Width</i>	<u>Double</u>	Die Breite des Rechtecks.

Methoden

Name	Parameter	Beschreibung
<i>Contains</i>	X als <u>Double</u> , Y als <u>Double</u>	Liefert ein <u>Boolean</u> . Liefert <u>True</u> falls das Rechteck den durch X, Y festgelegten Punkt enthält.

Beispiel

Folgender Code im Paint-Event eines Fensters zeichnet ein diamantenförmiges Rechteck:

```
Dim r as RectShape
r=New RectShape
r.width=75
r.height=75
r.border=100
r.borderColor=RGB(0,0,0) //schwarz
r.fillcolor=RGB(0,127,127) // türkis
r.borderWidth=2.5
r.rotation=-.78
g.DrawObject r,100,100
```

Siehe auch

ArcShape, CurveShape, FigureShape, FolderItem, Group2D, Graphics, OvalShape, Picture, PixmapShape, RoundRectShape, StringShape Klassen.

ReDim Methode

Paßt die Größe des Arrays oder der Variablen an.

Syntax

ReDim *arrayName*(*neueGroesse1*,[*neueGroesseN*])

Teil	Art	Beschreibung
<i>arrayName</i>	Variablenname	Der Name des vorher deklarierten Arrays.
<i>neueGroesse1</i>	<u>Integer</u>	Die neue Größe des Arrays. Im Falle eines mehr-dimensionalen Arrays ist dies die Größe der ersten Dimension des Arrays.
<i>neueGroesseN</i>	<u>Integer</u>	Die neue Größe für die n-te Dimension des Arrays.

Hinweise

Die Methode **ReDim** wird verwendet, um die mögliche Anzahl an Elementen für ein Array zu verkleinern oder zu vergrößern. Arrays sind null-basiert (ihre Elemente werden ab 0 gezählt), weshalb Sie beim Anpassen einen Wert angeben müssen, der um eins kleiner ist als die Anzahl der Elemente, die Sie tatsächlich benötigen.

Die Maximalzahl an Dimensionen in einem mehrdimensionalen Array ist 20.

Beispiele

Dieses Beispiel verkleinert das Array Namen auf 11 Elemente.

```
Redim Namen(10)
```

Dieses Beispiel fügt dem Array Namen Platz für 10 weitere Elemente hinzu.

```
Redim Namen(UBound(Namen)+10)
```

Dieses Beispiel verkleinert das Array Leute auf 11 Elemente für die erste und 6 Elemente für die zweite Dimension.

```
Redim Leute(10,5)
```

Siehe auch

Dim Befehl; Array, Split, Join, Ubound Funktionen; Append, Insert, Remove, Pop, IndexOf, Insert, Shuffle, Sort Methoden; ParamArray Schlüsselwort.

RegEx Klasse

Wird für Suche- und Ersetze-Operationen mit Hilfe von regulären Ausdrücken (perl) verwendet.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Options</i>	<u>RegexOptions</u>	Diese Optionen sind verschiedene Zustände, die Sie für die Engine für die regulären Ausdrücke setzen können. Näheres siehe <u>RegexOptions</u> -Klasse.
<i>ReplacementPattern</i>	<u>String</u>	Dies ist der Ersetze-String, der über die bei regulären Ausdrücken übliche Standard-Notation '^1' oder '\$1' Verweise auf bereits gefundene Substrings beinhalten kann. Dieses Muster wird entweder in der Replace-Eigenschaft verwendet oder an die <u>RegexMatch</u> -Klasse übergeben, wenn Search erfolgt ist und wird später mit Replace verwendet, wenn keine Parameter festgelegt werden.
<i>SearchPattern</i>	<u>String</u>	Dies ist das Muster, nach dem Sie momentan suchen.
<i>SearchStartPosition</i>	<u>Integer</u>	Zeichen-Position, an der Sie die Suche starten wollen, wenn der optionale Parameter TargetString zu <i>Replace</i> nicht festgelegt wurde. Beachten Sie, daß, wenn Sie ihn setzen, er nur verwendet wird, wenn Sie keinen TargetString festlegen, da das Setzen eines neuen TargetString den Wert zurücksetzt.

Methoden

Name	Parameter	Beschreibung
<i>Replace</i>	Optional: TargetString als <u>String</u> ; SearchStartPosition als <u>Integer</u>	Findet <i>SearchPattern</i> in <i>Target</i> . Ersetzt den Inhalt von <i>SearchPattern</i> mit <i>ReplacementPattern</i> . Liefert den resultierenden String. Replace kann den links angegebenen optionalen Parameter erhalten. Liefert einen <u>String</u> zurück.
<i>Search</i>	TargetString als <u>String</u> ; SearchStartPosition als <u>Integer</u>	Findet <i>SearchPattern</i> in <i>TargetString</i> und liefert, falls erfolgreich, ein <u>RegexMatch</u> . Das <u>RegexMatch</u> merkt sich ReplacementPattern, das zum Zeitpunkt der Suche festgelegt wurde.

Hinweise

Dieser Abschnitt beschreibt die Syntax der regulären Ausdrücke in REALbasic.

Muster (Pattern)	Beschreibung
.	Findet jedes Zeichen außer Zeilenvorschub (newline).
[a-z0-9]	Findet jedes einzelne Zeichen im Set
[^ a-z0-9]	Findet jedes einzelne Zeichen, das nicht im Set ist
\d	Findet eine Zahl. Das gleiche wie [0-9]
\D	Findet eine Nicht-Zahl. Das gleiche wie [^ 0-9]
\w	Findet ein alphanumerisches (Wort-)Zeichen — [a-zA-Z0-9_]
\W	Findet ein nicht-alphanumerisches (Wort-)Zeichen [^ a-zA-Z0-9_]
\s	Findet ein Steuerzeichen (Leerzeichen, Tab, Zeilenvorschub, etc.)
\S	Findet ein Nicht-Steuerzeichen
\n	Findet einen Zeilenvorschub

Muster (Pattern)	Beschreibung
\r	Findet ein Return
\t	Findet ein Tab
\f	Findet einen Seitenvorschub (formfeed)
\b	Findet ein Backspace
\0	Findet ein Null-Zeichen
\000	Findet auch ein Null-Zeichen (siehe nächste Zeile)
\nnn	Findet ein ASCII-Zeichen mit diesem oktalen Wert
\xnn	Findet ein ASCII-Zeichen mit diesem hexadezimalen Wert
\cX	Findet ein ASCII Kontroll-Zeichen
\metachar	Findet ein Meta-Zeichen (z.B., \, .,)
(abc)	Wird zum Erzeugen von Unterausdrücken verwendet. Merkt sich das Ergebnis für spätere Rückverweise. Referenziert durch die Ersetzen-Patterns, die \1, \2 etc. verwenden.
\1, \2, ...	Findet, was als erstes (zweites und so weiter) paßt.
x?	Findet 0 oder 1 x, wobei x eines von oben ist.
x*	Findet 0 oder mehr x.
x+	Findet 1 oder mehr x.
x{m,n}	Muß mindestens m, aber darf nicht häufiger als n vorkommen.
abc	Findet alles von a, b und c in dieser Reihenfolge.
a b c	Findet eines von a, b oder c.
\b	Findet eine Wortgrenze (nur außerhalb von []).
\B	Findet eine nicht-Wortgrenze.
^	Wahr am Anfang einer Zeile oder eines Strings.
\$	Wahr am Ende einer Zeile oder eines Strings.

Ersetzen-Pattern

Folgende Ausdrücke können nur der Ersetzen-Pattern zugewiesen werden:

\$`	Wird mit dem gesamten Zielstring vor der Übereinstimmung ersetzt.
\$&	Der gesamte Übereinstimmungsbereich; ist identisch zu \0 und \$0.
\$'	Wird mit dem gesamten Zielstring ersetzt, hinter dem übereinstimmenden Text.
\$0-\$50	\$0-\$50 wird zu nichts ausgewertet, wenn der Unterausdruck, der zur Nummer gehört, nicht existiert.
\0-\50	
\xnn	Wird durch den Buchstaben ersetzt, der durch nn in Hexadezimal repräsentiert wird, z.B. ist \xAA das TM .
\nnn	Wird durch den Buchstaben ersetzt, der durch nn in Oktal repräsentiert wird
\cX	Wird durch den Buchstaben ersetzt, der die Control-Version von X ist, z.B. ist \cP das DLE, Data Line Escape.

Doppel-Byte-Systeme

Wenn Sie auf einem Doppel-Byte-System wie dem Japanischen arbeiten, kann RegEx nicht direkt mit den Zeichen direkt arbeiten. Sie sollten zuerst allen Doppel-Byte-Text in UTF8 mit den in REALbasic eingebauten Textkonvertier-Funktionen konvertieren. Ein Beispiel finden Sie bei der TextConverter-Klasse.

Jeder Text, der von RegEx bearbeitet werden soll, sollte konvertiert werden. Dies schließt SearchPattern, ReplacementPattern und TargetString mit ein. Das Ergebnis der Suche oder des Suchen und Ersetzens ist ein UTF8-String, so daß Sie diesen mit den Textkonvertier-Funktionen gegebenenfalls wieder zurück in die Originalform konvertieren müssen. Sowohl die Suchen Funktion als auch das Suchen und Ersetzen funktionieren auf dem Macintosh als auch unter Windows, vorausgesetzt, daß diese Konvertierung vorgenommen wird.

Beispiele für Reguläre Ausdrücke

Die Grundidee bei Regulären Ausdrücken ist, daß Sie in der Lage sind, Text, der auf eine Auswahl von festzulegenden Bedingungen paßt, zu suchen und zu ersetzen. Es erweitert das normale Suchen und Ersetzen um Suchmuster (Search Patterns).

Wildcards

Einige Spezialzeichen werden zum Finden einer Klasse von Zeichen verwendet.

Wildcard	Findet
.	Beliebiges Zeichen außer einem Return.

Wenn Sie den „.“ als Suchmuster verwenden, selektieren Sie das erste Zeichen im Zielstring und, wenn Sie die Suche wiederholen, finden Sie jedes weitere Zeichen außer den Return-Zeichen.

Folgende Wildcards finden Stellen in einer Zeile:

Wildcard	Findet	Beispiel
^	Beginn einer Zeile (sofern nicht in einer Character-Klasse verwendet)	^ Telefon: Findet Zeilen, die mit „Telefon“ beginnen.
\$	Ende einer Zeile (sofern nicht in einer Character-Klasse verwendet)	\$: Findet das letzte Zeichen in der aktuellen Zeile.

Character-Klassen

Das Konzept einer Character-Klasse erlaubt Ihnen, eine Gruppe oder ein Set von Zeichen festzulegen. Sie können wählen, ob diese Character-Klasse gefunden oder ignoriert werden sollen. Die Character-Klasse wird durch eckige Klammern ([]) um die Zeichen, die sie beinhalten soll, eingeschlossen. Um festzulegen, daß Sie die Gruppe von Zeichen in der Character-Klasse ausschließen möchten, verwenden Sie das Caret-Zeichen (^) direkt hinter der öffnenden Klammer. Hier sind ein paar Beispiele:

Character-Klasse	Findet
[aeiou]	Jedes der Zeichen a, e, i, o, u
[^ aeiou]	Jedes Zeichen außer a, e, i, o, u
[a-e]	Jedes Zeichen im Bereich a-e (inklusive)
[a-zA-Z0-9]	Jedes alphanumerische Zeichen. Hinweis: Case-Sensitivität wird durch die CaseSensitive-Eigenschaft der RegexOptions-Klasse gesteuert.
[.]	Findet ein [.
[.])	Findet ein]. Um eine schließende Klammer zu finden, platzieren Sie diese direkt hinter die öffnende Klammer.
[a-e ^]	Findet Zeichen im Bereich a-e oder das Caret-Zeichen. Um das Caret-Zeichen zu finden, platzieren Sie es irgendwo, jedoch nicht als erstes Zeichen nach der öffnenden Klammer.
[a-c-]	Findet Zeichen im Bereich a-c oder das - Zeichen. Um ein - zu finden, platzieren Sie es an den Anfang oder das Ende einer Gruppe.

Nichtdruckbare Zeichen

Sie können die folgenden Spezial-Zeichen verwenden, um nach Zeilenvorschüben und anderen nichtdruckbaren Zeichen zu suchen:

Spezial-Zeichen	Findet
\r	Return.
\n	Zeilenvorschub (newline).
\t	Tab.
\f	Seitenvorschub
\xNN	Hexadezimaleres Zeichen NN (z.B., \x0D für Return).

Andere Spezial-Zeichen

Spezial-Zeichen	Findet
\s	Jedes Leerraum-Zeichen (Leerzeichen, Tab, Return, Zeilenvorschub, Seitenvorschub).
\S	Jedes Nicht-Leerraum-Zeichen.
\w	Jedes „Wort“-Zeichen (a-z, 0-9 und _).
\W	Jedes Nicht-Wort-Zeichen (alle Zeichen, die nicht zu \w gehören).
\d	Jede Ziffer [0-9]
\D	Jede Nicht-Ziffer.

Wiederholungszeichen

Wiederholungszeichen sind Modifier, die das Wiederholen eines bestimmten Musters erlauben.

Wiederholungs-Zeichen	Findet	Beispiel
*	Null oder mehr Zeichen.	d* findet keine Zeichen oder ein oder mehr aufeinanderfolgende „d“s. .* findet eine gesamte Textzeile, schließt das Return-Zeichen jedoch nicht ein.
+	Ein oder mehr Zeichen	d+ findet ein oder mehr aufeinanderfolgende „d“s. [0-9]+ findet einen String mit einer oder mehr aufeinanderfolgenden Ziffern, so wie „90404“, „1938“, das „32“ in „Win32“ etc.
?	Null oder ein Zeichen	d? findet kein Zeichen oder ein „d“.

Bitte beachten Sie, daß * und ? auch kein Vorkommen des Suchmusters finden, so daß sie immer erfolgreich sind aber nicht unbedingt Text selektieren. Sie können sie verwenden, um ein optionales Zeichen festzulegen, wie im Beispiel im folgenden Abschnitt.

„Gier“

REALbasic unterstützt das „?“ als ein „gieriges“ Zeichen für ein Untermuster in einem Regulären Ausdruck. Als Voreinstellung wird die Gier über die Greedy-Eigenschaft in der RegexOptions-Klasse gesteuert, kann aber durch Verwenden von „?“ überschrieben werden. Sie können ein „?“ direkt nach einem * oder + plazieren, um die Gier-Einstellung umzustellen. Das bedeutet, falls Greedy True ist, führt die Verwendung von ? nach einem * oder + dazu, daß die mögliche Minimalanzahl gefunden wird.

Zielstring	Gier (Greedy)	Regulärer Ausdruck	Ergebnis
aaaa	True	(a+?)(a+)	\$1=a, \$2=aaa
aaaa	False	(a+?)(a+)	\$1=aaa, \$2=a

Erweiterte Form der Klammerung

REALbasic unterstützt auch die erweiterte Form der Klammerung, wie sie in Perl verwendet wird. Z.B.:

(?#text)	comment
(?:muster)	Gruppieren, ohne Rückwärtsreferenzen zu erzeugen.
(?=muster)	Positive Lookahead-Zusicherung der Länge Null. Z.B. erkennt \w+(?=t) ein Wort, dem ein Tabulator folgt, ohne den Tabulator mit in \$\$ anzulegen.
(?!muster)	Negative Lookahead-Zusicherung der Länge Null. Z.B. erkennt foo(?!bar)/ jedes Auftreten von „foo“, dem nicht „bar“ folgt.
(?<=muster)	Positive Lookbehind-Zusicherung der Länge Null. Z.B. erkennt (?<=t)\w+ ein Wort, das auf einen Tabulator folgt, ohne den Tabulator mit in \$\$ einzubeziehen. Das Look-behind-Muster muß eine feste Länge haben.
(?<!muster)	Negative Lookbehind-Zusicherung der Länge null. Das Beispiel (?<!bar)foo findet jedes Auftreten von „foo“, das nicht hinter „bar“ steht. Das Lookbehind-Muster muß eine feste Länge haben.

Teilmuster

Sie können in Suchmustern Klammern verwenden, um Teile der gefundenen Zeichenkette zu isolieren. Dies ist dann sinnvoll, wenn Sie sich in der Ersetzungszeichenkette auf Teile der gefundenen Zeichenkette beziehen wollen. Beispielsweise wäre das der Fall, wenn Sie nur einen Teil der gefundenen Zeichenkette ersetzen oder zusätzlichen Text an der Fundstelle einsetzen wollen.

Beispiel: Wenn Sie jedes Datum finden wollen, auf das die Zeichen „v.Chr.“ folgen, können Sie das Muster „\d+\sv\Chr.“ verwenden. Dieses paßt auf eine Zahl beliebiger Länge, der ein Leerzeichen und dann die Zeichen „n.Chr.“ folgen. Damit finden Sie Datumsangaben der Form 33 v.Chr., 1742 v.Chr. etc. Wenn Sie nun das Jahr stehen lassen und die Buchstaben durch etwas anderes ersetzen wollen, müssen Sie Klammern verwenden. Das Suchmuster „(\d+)\s(v\Chr\.)“ leistet genau das.

In der Ersetzung können Sie sich mit der Variablen \1 auf das Jahr und der Variablen \2 auf die Buchstaben beziehen.

Wenn Sie „(\d+)\s(v.Chr. | n.Chr. | v.C. | n.C.)“ verwenden, enthält \2 die gefundenen Buchstaben.

Muster kombinieren

Reguläre Ausdrücke sind deshalb so leistungsfähig, weil man die elementaren Muster für komplexe Suchvorgänge kombinieren kann. Hier folgen einige Beispiele:

Muster	Findet
<code>\\$?[0-9,]+\.\#*</code>	Findet Dollarbeträge mit optionalem Dollarzeichen.
<code>\d+\sn\Chr\.</code>	Findet eine oder mehrere Ziffern, gefolgt von einem Leerzeichen und „n.Chr.“.

Der Alternierungsoperator

Der Alternierungsoperator „|“ gestattet Ihnen, mehrere Suchmuster anzugeben, von denen nur eines passen muss. Setzen sie ihn dazu einfach zwischen zwei Suchmuster, von denen eines zutreffen soll, damit die Suche erfolgreich verläuft.

In einem Suchmuster können Sie mehrere Alternierungsoperatoren verwenden.

Muster	Findet
<code>\ser\s \ssie\s</code>	„er“ oder „sie“
<code>Katze Hund Ratte</code>	„Katze“, „Hund“, oder „Ratte“
<code>([0-9,]+\sv\Chr\.) ([0-9,]+\sn\Chr\.) or [0-9,]+\s((v\Chr\.) (n\Chr\.))</code>	Jahreszahlen der Form „ <i>jahr</i> v.Chr. oder n.Chr.“, z.B. „2,175 v.Chr.“ oder „215 n.Chr.“

Suchen und ersetzen

Spezielle Muster repräsentieren die gefundenen Zeichenketten. Mit Hilfe dieser Muster können Sie dem gefundenen Text andere Texte voranstellen oder anhängen.

Muster	Beschreibung
<code>\$&</code>	Enthält den vollständigen gefundenen Text. Wenn „\d\d\d\d\sv\Chr.“ den Text „1541 v.Chr.“ findet, erzeugt das Ersetzungsmuster „das Jahr \$&“ den Text „das Jahr 1541 v.Chr.“, da \$& die Zeichenkette „1541 v.Chr.“ enthält.
<code>\1, \2, etc.</code>	Enthält die gefundenen Teilmuster, die durch Verwendung von Klammern im Suchmuster definiert wurden. Das Suchmuster „(\d+)\s(v\Chr\. n\Chr\. v\C\. n\D\.)“ sucht eine beliebige Anzahl von Ziffern, denen ein Leerzeichen und entweder der Text „v.Chr.“, „n.Chr.“, „v.C.“ oder „n.C.“ folgt. Die Variable \1 enthält die gefundene Jahreszahl, während \2 den darauf folgenden Text liefert.

Danksagungen

REALbasic verwendet eine modifizierte Version des PCRE-Library-Pakets, welches Open Software ist, die von Philip Hazel geschrieben wurde und dessen Urheberrecht bei der University of Cambridge, England, liegt.

Der Source-Code dieser Library ist hier verfügbar:

<ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

Beispiele

Das folgende Action-Event des PushButtons erlaubt Ihnen, den Text in EditField1 mit Hilfe des Suchmusters, das in EditField2 eingegeben werden kann, zu durchsuchen und das Ergebnis der Suche in einem StaticText anzuzeigen.

```
Dim rg as RegEx
Dim myMatch as RegExMatch
rg=new RegEx
```



```

rg.SearchPattern=EditField2.text
myMatch=rg.search(EditField1.text)
if myMatch <> Nil then
    StaticText1.text=myMatch.SubExpressionString(0)
else
    StaticText1.text="Text nicht gefunden!"
End if
exception err as RegExException
MsgBox err.message

```

Siehe auch

RegexMatch, RegexOptions Klassen; RegExException Fehler.

RegExException Error

Die RegEx-Engine hat einen Laufzeitfehler in Ihrem Code gefunden. Derzeit ist der einzige Typ von Laufzeitfehler eine RegexSearchPatternException, künftige REALbasic-Versionen werden aber voraussichtlich auch andere Fehlerarten abfangen.

Übergeordnete Klasse

RuntimeException

Siehe auch

RuntimeException Klasse; RegexSearchPatternException Fehler.

RegexMatch Klasse

Wird zum Extrahieren des gefundenen Strings verwendet, wenn man eine Suche mit regulären Ausdrücken ausführt.

Übergeordnete Klasse

Object

Eigenschaften

Name	Parameter	Beschreibung
<i>SubExpressionCount</i>		Anzahl der verfügbaren SubExpressions der gerade erfolgten Suche. Die regulären Ausdrücke von REALbasic unterstützen sowohl \number als auch \$number für SubExpressions. SubExpressions erlauben das Ersetzen von Teilen des Musters (Pattern). Liefert ein Integer.
<i>SubExpressionString</i>	matchNumber als <u>Integer</u>	Liefert SubExpression als einen String für die angegebene matchNumber. 0 liefert den gesamten MatchString (das implizierte 0te subExpression) und 1 ist die erste echte subExpression. Liefert einen String.
<i>SubExpressionStartB</i>	matchNumber als <u>Integer</u>	Liefert die Start-Byte-Position von der durch matchNumber festgelegten subExpression. Liefert einen Integer
<i>Replace</i>	ReplacementPattern als <u>String</u> (Optional)	Ersetzt das gefundene Ergebnis in der Weise, die durch <i>ReplacementPattern</i> festgelegt wird. Wenn kein <i>ReplacementPattern</i> festgelegt ist, verwendet es das ReplacementPattern, das im <u>RegEx</u> -Objekt zur Zeit der Suche festgelegt wurde. Liefert einen String.

Siehe auch

RegEx Klasse.

RegexOptions Klasse

Wird verwendet, um die Optionen für eine Suche mit regulären Ausdrücken festzulegen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>CaseSensitive</i>	<u>Boolean</u>	Legt fest, ob Groß-/Kleinschreibung beim Suchen eines Strings berücksichtigt werden soll. Voreinstellung ist <u>False</u> .
<i>DotMatchAll</i>	<u>Boolean</u>	Normalerweise findet der Punkt alles außer einem Zeilenvorschub. Diese Option erlaubt ihm, auch Zeilenvorschübe zu finden. Voreinstellung ist <u>False</u> .
<i>Greedy</i>	<u>Boolean</u>	Greedy (Gier) bedeutet, daß alles vom ersten bis zum Ende des letzten Trennzeichens durchsucht wird. Wenn Sie z.B. den folgenden Text mit Bold-Tags in HTML finden wollen: The quick brown fox jumped und Sie dieses Muster verwenden: .+ erhalten Sie bei der Suche "quick brown fox", was nicht das ist, was Sie eigentlich wollten. Daher können Sie <i>Greedy</i> ausschalten oder diese Syntax verwenden: .+? und Sie werden "quick" finden (was genau das ist, was Sie wollten.) Voreinstellung ist <u>True</u> .
<i>LineEndType</i>	<u>Integer</u>	Dies wirkt sich auf die aktuelle Session der regulären Ausdrücke aus. Es hat keinen Effekt auf SearchPatterns, wenn <i>TreatTargetAsOneLine</i> <u>True</u> ist. Ändert die Art, wie „\n“ für ReplacementPatterns expandiert wird 0 = beliebiges Zeilenende (Mac oder Win32 oder Unix) 1 = Default für die Plattform (wenn auf einem Mac, dann 2) (wenn unter Win32, dann 3) 2 = Mac ASCII 13 oder \r 3 = Win32 ASCII 10 oder \n 4 = Unix ASCII 10 oder \n Voreinstellung ist 0.
<i>MatchEmpty</i>	<u>Boolean</u>	Legt fest, ob es Mustern erlaubt ist, leere Strings zu finden. Voreinstellung ist <u>True</u> .
<i>ReplaceAllMatches</i>	<u>Boolean</u>	Legt fest, ob jedes Auftreten des Musters ersetzt werden soll. Voreinstellung ist <u>False</u> .
<i>StringBeginIsLineBegin</i>	<u>Boolean</u>	Legt fest, ob der Anfang eines Strings als der Anfang einer Zeile zählt. Voreinstellung ist <u>True</u> .
<i>StringEndIsLineEnd</i>	<u>Boolean</u>	Legt fest, ob das Ende eines Strings als Ende einer Zeile zählt. Voreinstellung ist <u>True</u> .
<i>TreatTargetAsOneLine</i>	<u>Boolean</u>	Ignoriert interne Zeilenumbrüche zum Zwecke des Findens von '^' und '\$'. Voreinstellung ist <u>False</u> .

Siehe auch

Regex, RegexMatch Klassen.

RegexSearchPatternException Error

Sie haben ein ungültiges Suchmuster für reguläre Ausdrücke verwendet

Übergeordnete Klasse

RegexException Fehler.

Beispiel

Folgender Exception-Block der an das Ende einer Methode gesetzt wird, die eine Suche mit Regulären Ausdrücken macht, fängt ungültige Suchmuster ab und gibt die Message-Eigenschaft der `RegexSearchPatternException` aus. Sie können auch `RegexException` verwenden, da `RegexSearchPatternException` eine Unterklasse von `RegexException` ist.

```
Exception err as RegexSearchPatternException
MsgBox err.message
```

Siehe auch

[RuntimeException](#) Klasse.

RegistryAccessErrorException Fehler

Tritt auf, wenn Sie versuchen, die `RegistryItem`-Klasse ohne geeignete Zugriffsprivilegien zu verwenden oder wenn Sie versuchen sie auf einem Macintosh OS zu verwenden. Registry-Einträge sind ein Feature, das es nur unter Windows gibt.

Übergeordnete Klasse

[RuntimeException](#)

Siehe auch

[RegistryItem](#) Klasse, [Exception](#) Block.

RegistryItem Klasse

Erzeugt und verwaltet Registry-Einträge unter Windows. Kann nicht auf einem Macintosh verwendet werden.

Übergeordnete Klasse

[Object](#)

Constructors

Name	Parameter	Beschreibung
<i>RegistryItem</i>	Path als String	Erzeugt den Registry-Eintrag aus dem übergebenen Pfad. Liefert ein neues RegistryItem .
<i>RegistryItem</i>	Copy als RegistryItem	Dupliziert das übergebene RegistryItem . Liefert ein neues RegistryItem .

Eigenschaften

Name	Typ	Beschreibung
FolderCount	Integer	Die Anzahl der Kind-Ordner, die im RegistryItem enthalten sind.
KeyCount	Integer	Die Anzahl der Schlüssel, die im RegistryItem enthalten sind.
Parent	RegistryItem	Das Elternobjekt des RegistryItem. Hinweis: Sie können nicht das Elternobjekt einer Untereinheit bekommen. Jede Untereinheit ist als Wurzel gedacht.
Path	String	Der komplette Pfad auf das RegistryItem .

Methoden

Name	Parameter	Beschreibung
Child	Name als String	Liefert das genannte Kind als ein RegistryItem . Liefert ein RegistryItem .
DefaultValue		Liefert den Default-Wert des RegistryItem als ein Variant .
Item	Index als Integer	Liefert als ein RegistryItem das zu <i>Index</i> gehörende RegistryItem . Falls <i>Index</i> größer als <i>KeyCount</i> ist, tritt eine OutOfBoundsException auf.
Name	Index als Integer	Liefert als ein String den Namen des RegistryItem das zu <i>Index</i> gehört. Falls <i>Index</i> größer als <i>KeyCount</i> ist, tritt eine OutOfBoundsException auf.

Name	Parameter	Beschreibung
Value	Name als <u>String</u>	Liefert den RegistryItem -Wert als ein <u>Variant</u> . Index verweist auf einen Schlüssel, der im RegistryItem enthalten ist. Falls <i>Index</i> größer als KeyCount ist, tritt eine <u>OutOfBoundsException</u> auf.

Hinweise

Die Value-Methode liefert gültige Informationen für erweiterbare Strings, binär und Mehrfach-Strings. Im Falle eines erweiterbaren Strings (expandable string) wird REALbasic den Strings automatisch erweitern und als String zurückgeben.

Im Falle eines Mehrfach-Strings wird ein String zurückgegeben, der die einzelnen Strings mit einem Chr(0) voneinander trennt. Verwenden Sie die Split Funktion, um die einzelnen Strings zu erhalten. Zum Beispiel,

```
part = Split(regValue, Chr(0))
```

wobei *part* ein String-Array ist.

Im Falle eines binären Strings liefert REALbasic einen MemoryBlock. Sie können in dem Moment keinen erweiterbaren oder Mehrfach-String zuweisen. Um einen binären String zuzuweisen, verwenden Sie bitte einen MemoryBlock.

Siehe auch

RegistryAccessErrorException Fehler.

Rem Befehl

Wird verwendet, um dem Code Kommentare beizufügen.

Syntax

Rem Text-String.

Hinweis

REALbasic ignoriert alle Kommentare, die Sie mit dem Befehl Rem eingeben. Kommentare werden auch nicht in die fertige Anwendung übernommen.

Sie können auch ein ' (Apostroph) oder zwei Schrägstriche (//) verwenden, um ihren Code zu kommentieren. Kommentare können in einer separaten Zeile oder auch auf der gleichen Zeile wie Ihr ausführbarer Code stehen, solange sie rechts vom ausführbaren Code stehen. Gültige Kommentare erscheinen im Code-Editor in rot.

Das Tastaturkürzel Befehl-' auf der Tastatur schaltet die selektierte Code-Zeile zwischen Kommentar und „Lebendig“-Status hin und her.

Siehe auch

//'-Operatoren.

Beispiel

Folgender Code verwendet Kommentare, um die Ergebnisse von boole'schen Vergleichen zu dokumentieren:

```
Dim a, b, c, d As Boolean
a=True
b=False
d=a Or b Rem Ergibt True
d=b And c Rem Ergibt False
```

Remove Methode

Entfernt ein Element aus einem Array.

Syntax

array.**Remove** index

Teil	Art	Beschreibung
<i>array</i>	beliebiger Datentyp	Das Array, aus dem Sie Daten entfernen möchten.
<i>index</i>	<u>Integer</u>	Das zu entfernende Element

Hinweis

Die Methode **Remove** entfernt das Element *index* aus *array*.

Die Methode **Remove** arbeitet nur mit eindimensionalen Arrays.

Beispiel

Dieses Beispiel entfernt das Element 4 (das fünfte Element) aus dem Array Namen:

```
Namen.Remove 4
```

Siehe auch

[Dim](#) Befehl; [Array](#), [Split](#), [Join](#), [Ubound](#) Funktionen; [Append](#), [Insert](#), [Redim](#), [Pop](#), [IndexOf](#), [Insert](#), [Shuffle](#), [Sort](#) Methoden; [ParamArray](#) Schlüsselwort.

Replace Funktion

Ersetzt das erste Vorkommen eines Strings mit einem anderen String.

Syntax

Ergebnis=**Replace**(QuellString, altString, neuString)

ODER

Ergebnis=**StringVariable.Replace**(altString, neuString)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Eine Kopie von <i>QuellString</i> , in der das erste Vorkommen von <i>altString</i> durch <i>neuString</i> ersetzt wurde.
<i>QuellString</i>	<u>String</u>	Der Original-String.
<i>altString</i>	<u>String</u>	Die zu ersetzenden Zeichen.
<i>neuString</i>	<u>String</u>	Die ersetzenden Zeichen.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweise

Ersetzt das erste Vorkommen von *altString* im *QuellString* mit *neuString*. **Replace** ist case-insensitiv.

Wenn *neuString* ein leerer String (""), entfernt die Funktion **Replace** das erste Vorkommen von *altString* im *QuellString*.

Ist *altString* ein leerer String (""), so liefert **Replace** eine unveränderte Kopie von *QuellString* zurück.

Beispiele

Dies sind Beispiele, die Ergebnisse der **Replace**-Funktion zeigen:

```
Dim Ergebnis As String
Ergebnis=Replace("Der schnelle Fuchs","Fuchs","Hase") //liefert "Der schnelle Hase"
Ergebnis=Replace("Der schnelle Fuchs","F","L") //liefert "Der schnelle Luchs"
Ergebnis=Replace("Der schnelle Fuchs","schnelle","") //liefert "Der Fuchs"
```

Unter Verwendung der zweiten Syntax:

```
Dim result,s As String
s="Der schnelle Fuchs"
result=s.Replace("Fuchs","Hase") //liefert "Der schnelle Hase"
MsgBox result
```

Siehe auch

[ReplaceAll](#), [ReplaceAllB](#), [ReplaceB](#) Funktionen.

ReplaceAll Funktion

Ersetzt alle Vorkommen eines Strings mit einem anderen String.

Syntax

Ergebnis=**ReplaceAll**(QuellString, altString, neuString)

ODER

Ergebnis=**StringVariable.ReplaceAll**(altString, neuString)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Eine Kopie von <i>QuellString</i> , in der alle Vorkommen von <i>altString</i> durch <i>neuString</i> ersetzt wurden.
<i>QuellString</i>	<u>String</u>	Der Original-String.
<i>altString</i>	<u>String</u>	Die zu ersetzenden Zeichen.
<i>neuString</i>	<u>String</u>	Die ersetzenden Zeichen.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **ReplaceAll** ersetzt alle Vorkommen von *altString* in *QuellString* durch *neuString*. **ReplaceAll** ist case-insensitiv. Wenn *neuString* ein leerer String ("") ist, entfernt die Funktion **ReplaceAll** jedes Vorkommen von *altString* im *QuellString*. Ist *altString* in leerer String (""), so liefert die Funktion **ReplaceAll** eine unveränderte Kopie von *QuellString* zurück.

Beispiele

Dies sind Beispiele, die Ergebnisse der Funktion **ReplaceAll** zeigen:

```
Dim Ergebnis As String
Ergebnis=ReplaceAll("xyxyxy","x","z") //liefert "zyzyzy"
Ergebnis=ReplaceAll("Der schnelle Fuchs"," ","") //liefert "DerschnelleFuchs"
```

Siehe auch

Replace, ReplaceAllB, ReplaceB Funktionen.

ReplaceAllB Funktion

Ersetzt alle Vorkommen eines Strings mit einem anderen String.

Syntax

Ergebnis=**ReplaceAllB**(QuellString, altString, neuString)

ODER

Ergebnis=**StringVariable.ReplaceAllB**(altString, neuString)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Eine Kopie von <i>QuellString</i> , in der alle Vorkommen von <i>altString</i> durch <i>neuString</i> ersetzt wurden.
<i>QuellString</i>	<u>String</u>	Der Original-String.
<i>altString</i>	<u>String</u>	Die zu ersetzenden Zeichen.
<i>neuString</i>	<u>String</u>	Die ersetzenden Zeichen.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **ReplaceAllB** ersetzt alle Vorkommen von *altString* in *QuellString* durch *neuString*. **ReplaceAllB** ist case-sensitiv, da es den Quellstring als eine Reihe von Bytes behandelt. Es sollte an Stelle von ReplaceAll verwendet werden, wenn der String aus binären Daten besteht oder wenn Ihre Applikation auf einem Ein-Byte-Character-Set (wie dem US-System) läuft und Sie Case-Sensitivität möchten.

Wenn *neuString* ein leerer String ("") ist, entfernt die Funktion **ReplaceAllB** jedes Vorkommen von *altString* im *QuellString*. Ist *altString* in leerer String (""), so liefert die Funktion **ReplaceAllB** eine unveränderte Kopie von *QuellString* zurück.

Beispiele

Dies sind Beispiele, die Ergebnisse der Funktion **ReplaceAllB** zeigen:

```
Dim Ergebnis As String
Ergebnis=ReplaceAllB("xyxyxy","x","z") //liefert "zyzyzy"
Ergebnis=ReplaceAllB("Der schnelle Fuchs"," ","") //liefert "DerschnelleFuchs"
```

Siehe auch

[Replace](#), [ReplaceAll](#), [ReplaceB](#) Funktionen.

ReplaceB Funktion

Ersetzt das erste Vorkommen eines [Strings](#) mit einem anderen [String](#).

Syntax

Ergebnis=**ReplaceB**(QuellString, altString, neuString)

ODER

Ergebnis=**StringVariable.ReplaceB**(altString, neuString)

Teil	Art	Beschreibung
<i>Ergebnis</i>	String	Eine Kopie von <i>QuellString</i> , in der das erste Vorkommen von <i>altString</i> durch <i>neuString</i> ersetzt wurde.
<i>QuellString</i>	String	Der Original-String.
<i>altString</i>	String	Die zu ersetzenden Zeichen.
<i>neuString</i>	String	Die ersetzenden Zeichen.
<i>StringVariable</i>	String	Eine beliebige Variable vom Typ String .

Hinweise

Ersetzt das erste Vorkommen von *altString* im *QuellString* mit *neuString*. **ReplaceB** ist die binäre Version von [Replace](#).

Wenn *neuString* ein leerer String (""), entfernt die Funktion **ReplaceB** das erste Vorkommen von *altString* im *QuellString*.

Ist *altString* ein leerer String (""), so liefert **ReplaceB** eine unveränderte Kopie von *QuellString* zurück.

ReplaceB ist case-sensitiv, da es den Quellstring als eine Reihe von Bytes behandelt. Es sollte an Stelle von [Replace](#) verwendet werden, wenn der String aus binären Daten besteht oder wenn Ihre Applikation auf einem Ein-Byte-Character-Set (wie dem US-System) läuft und Sie Case-Sensitivität möchten.

Beispiele

Dies sind Beispiele, die Ergebnisse der **ReplaceB**-Funktion zeigen:

```
Dim Ergebnis As String
Ergebnis=ReplaceB("Der schnelle Fuchs","Fuchs","Hase") //liefert "Der schnelle Hase"
Ergebnis=ReplaceB("Der schnelle Fuchs","F","L") //liefert "Der schnelle Luchs"
Ergebnis=ReplaceB("Der schnelle Fuchs","schnelle","") //liefert "Der Fuchs"
```

Unter Verwendung der zweiten Syntax:

```
Dim result,s As String
s="Der schnelle Fuchs"
result=s.ReplaceB("Fuchs","Hase") //liefert "Der schnelle Hase"
MsgBox result
```

Siehe auch

[ReplaceAll](#), [ReplaceAllB](#), [Replace](#) Funktionen.

ReplaceLineEndings Funktion

Ersetzt die Zeilenenden im übergebenen [String](#) durch die festgelegten Zeilenenden.

Syntax

Ergebnis=**ReplaceLineEndings**(SourceString,LineEnding)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	String	Eine Kopie von <i>sourceString</i> mit neuen Zeilenenden, die die ursprünglichen Zeilenenden ersetzen.
<i>SourceString</i>	String	Der Original-String.
<i>LineEnding</i>	String	Die Zeilenenden, die zum Ersetzen der Zeilenenden in <i>SourceString</i> verwendet werden sollen..

Hinweise

ReplaceLineEndings führt ein globales Suchen und Ersetzen aller Zeilenende-Zeichen in *SourceString* durch und ersetzt diese durch die festgelegten Zeilenenden. Die Suche erkennt automatisch Windows-, Macintosh- und Unix-Zeilenenden. Verwenden Sie diese Funktion, um mehrzeiligen (oder mehrabsätzigen) Text crossplattformkompatibel zu machen. Am einfachsten legt man den Zeilenende-Ersetzen-String mit dem EndOfLine-Objekt fest.

Beispiel

Dieses Beispiel ersetzt die Zeilenenden in dem Text in einem EditField durch Macintosh-Zeilenenden.

```
Dim s as String
s=ReplaceLineEndings(EditField1.Text,EndOfLine.Macintosh)
```

Siehe auch

EndOfLine Objekt.

ResourceFork Klasse

Objekte der ResourceFork-Klasse werden verwendet, um Daten in Ressourcen zu schreiben oder aus ihnen zu lesen. Sie können Resource-Dateien in das Projekt-Fenster importieren (Dateien vom Typ „rsrc“), indem Sie diese aus dem Finder in das Projekt-Fenster draggen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Geticl</i>	ID als <u>Integer</u>	Lädt ein großes (32 x 32) icl-Icon mit der angegebenen ID. Liefert den Typ <u>Picture</u> .
<i>Getics</i>	ID als <u>Integer</u>	Lädt ein kleines (16 x 16) ics-Icon mit der angegebenen ID. Liefert den Typ <u>Picture</u> .
<i>ResourceLocked</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Wird verwendet, um das Attribut locked der Resource zu setzen oder zu löschen.
<i>ResourcePreload</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Wird verwendet, um das Attribut Preload der Resource zu setzen oder zu löschen.
<i>ResourceProtected</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Wird verwendet, um das Attribut Protected der Resource zu setzen oder zu löschen.
<i>ResourcePurgeable</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Wird verwendet, um das Attribut Purgeable der Resource zu setzen oder zu löschen.
<i>ResourceSysHeap</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Wird verwendet, um das Attribut SysHeap der Resource zu setzen oder zu löschen.
TypCount	<u>Integer</u>	Anzahl der Resource-Typen, die im Resourcefork enthalten sind.

Methoden

Name	Parameter	Beschreibung
<i>AddPicture</i>	Bild als <u>Picture</u> , ID als <u>Integer</u> , Name als <u>String</u>	Legt eine PICT Resource mit dem angegebenen Namen und der ID an und trägt das angegebene Bild ein.
<i>AddResource</i>	Data als <u>String</u> , Typ als <u>String</u> , ID als <u>Integer</u> , Name als <u>String</u>	Legt eine Resource vom angegebenen Typ, Namen und ID an und trägt die angegebenen Daten ein.
<i>Close</i>		Schließt den offenen Resourcefork. Der Resourcefork wird automatisch geschlossen, wenn die Instanz gelöscht wird.
<i>GetCicn</i>	ID als <u>Integer</u>	Liefert die Resource CICN (Farbicon) mit der angegebenen ID als <u>Picture</u> zurück.
<i>GetCursor</i>	ID als <u>Integer</u>	Liefert die CURS-Resource passend zu ID als <u>MouseCursor</u> -Objekt.

Name	Parameter	Beschreibung
<i>GetHandle</i>	Type als <u>String</u> , ID als <u>Integer</u>	Liefert ein Resource-Handler als ein Integer, wenn Type und ID der gewünschten Resource übergeben werden. Ist für die Verwendung mit Declare-Befehlen gedacht. Die Resource wird freigegeben, wenn der Resource-Fork geschlossen wird oder wenn Sie ReleaseHandle aufrufen.
<i>GetNamedPicture</i>	Name als <u>String</u>	Liefert die PICT-Resource mit dem angegebenen Namen als <u>Picture</u> zurück.
<i>GetNamedResource</i>	Typ als <u>String</u> , Name als <u>String</u>	Liefert die angegebene Resource als <u>String</u> zurück.
<i>GetPicture</i>	ID als <u>Integer</u>	Liefert die PICT-Resource mit der angegebenen ID als <u>Picture</u> zurück.
<i>GetResource</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Liefert die angegebene Resource als <u>String</u> zurück.
<i>GetSound</i>	ID als <u>Integer</u>	Liefert die snd-Resource mit der angegebenen ID als <u>Sound</u> -Objekt zurück.
<i>ReleaseHandle</i>	Handle als <u>Integer</u>	Gibt einen Resource-Handle frei, der von GetHandle erhalten wurde.
<i>RemoveResource</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Entfernt die angegebene Resource aus dem Resource-Fork.
<i>ResourceCount</i>	Typ als <u>String</u>	Liefert die Anzahl von Ressourcen mit dem angegebenen Typ zurück.
<i>ResourceID</i>	Typ als <u>String</u> , Index als <u>Integer</u>	Liefert die über <i>Typ</i> und <i>Index</i> spezifizierte Resource-ID als <u>Integer</u> zurück. Achtung: die Liste beginnt bei Null.
<i>ResourceName</i>	Typ als <u>String</u> , ID als <u>Integer</u>	Liefert den Namen der über Typ und ID spezifizierten Resource als String zurück. Achtung: Die Liste beginnt bei Null.
<i>ResourceTyp</i>	Index als <u>Integer</u>	Liefert die über den Index spezifizierte Ressourcenart als <u>String</u> zurück. Achtung: Die Liste beginnt bei Null.

Hinweis

Mit einer Ausnahme wird der Zugriff auf den Resourcefork des Programmes nur auf dem Macintosh unterstützt. Prüfen Sie den Wert der Konstante TargetMacOS, bevor Sie versuchen, einen Resourcefork zu öffnen.

Die eine Ausnahme besteht darin, daß Sie benutzerdefinierte Cursor durch eine CURS-Resource einem Projektfenster hinzufügen können und auf diese in einem für Windows erzeugten Programm zugreifen können. Siehe Hinweis bei der MouseCursor-Klasse und Abschnitt „Benutzerdefinierte Maus-Cursor in Windows-Applikationen“ im Entwicklerhandbuch.

In erzeugten Applikationen werden alle Informationen des „Informationen“-Bereichs der Dialogbox „Applikation erzeugen“ in einer Standard-„vers“-Resource gespeichert. Sie können auf diese Informationen über die GetResource-Methoden der ResourceFork-Klasse zugreifen – also z.B. auf Versionsnummer, Länder-Code und Entwicklungsstadium.

Wenn Sie Resource-Nummern verwenden, die mit internen REALbasic-Ressourcen in Konflikt geraten, kann das unvorhersehbare Folgen haben. Eine Möglichkeit, Resource-ID-Konflikte zu vermeiden, ist, hohe Werte, z.B. 1128 statt 128 zu nehmen.

Ein alternativer Weg, benutzerdefinierte Icons dem erzeugten Programm hinzuzufügen, ist, die Icon-Daten (inklusive „icns“ 32-Bit-Icons) in eine Resource-Datei zu packen, die Sie Ihrem Projekt hinzufügen. Diese Icons überschreiben die Icon-Resource, die von REALbasic selbst angeboten wird.

Weitere Informationen zu den Eigenschaften der Resource-Attribute (ResourceLocked, ResourcePreLoad, ResourceProtected, ResourcePurgeable, ResourceSysHeap) finden Sie in Apples „Inside Macintosh“ Reihe.

Beispiele

Dieses Beispiel setzt das Attribut locked der PICT-Resource mit der ID 128:

```
Dim rf as ResourceFork
rf=GetFolderItem("MyApp").OpenResourceFork
rf.ResourceLocked("PICT",128)=True
```

Dieses Beispiel prüft, ob die Eigenschaft purgeable gesetzt ist:

```
Dim rf as ResourceFork
rf=GetFolderItem("MyApp").OpenResourceFork
If rf.ResourcePurgeable("PICT",128) Then
    Beep
    MsgBox "Diese Resource ist purgeable."
End If
```

Dieses Beispiel öffnet die Resourcefork des Projekts. Wenn Sie Ihrem Projekt eine Resource-Datei hinzugefügt haben, können Sie diese Technik verwenden, um auf ihre Resource-Datei zuzugreifen (vorausgesetzt, das Programm läuft auf dem Macintosh):

```
Dim rf as ResourceFork
rf=App.ResourceFork

Dieses Beispiel holt das Schreibtisch-Applikations-Icon von Photoshop® und zeigt es in einem Canvas-Steuerelement an:

Dim rf as ResourceFork
Dim f as FolderItem
Dim p as Picture
f=GetFolderItem("Photoshop")
If f <> Nil then
    rf=f.OpenResourceFork
    If rf <> Nil then
        p=rf.GetPic(128)
        Canvas1.backdrop=p
    Else
        MsgBox "Resource nicht gefunden!"
    End if
Else
    MsgBox "Datei nicht gefunden!"
End if
```

Dieses Beispiel lädt einen Cursor aus der Photoshop-Applikation und verwendet ihn als Mauszeiger für die eigene Anwendung.

```
Dim rf as ResourceFork
Dim m as MouseCursor
Dim f as FolderItem
f=GetFolderItem("Photoshop")
If f <> Nil then
    rf=f.OpenResourceFork
    If rf <> Nil then
        m=rf.GetCursor(1525)
        App.MouseCursor=m
    Else
        MsgBox "Resource nicht gefunden!"
    End if
Else
    MsgBox "Datei nicht gefunden!"
End if
```

Siehe auch

[App](#) Objekt,[Window](#), [MouseCursor](#) Klasse.

Return Befehl

Kehrt in der Ausführung zur aufrufenden Methode zurück.

Syntax

Return [Wert]

Teil	Typ	Beschreibung
Wert	Beliebig	Wert, der zur aufrufenden Funktion zurückgegeben wird. Der Datentyp muss bei der Funktionsdeklaration spezifiziert werden.

Hinweise

Verwenden Sie **Return**, um zur aufrufenden Funktion zurückzukehren und den angegebenen Wert dabei zurückzugeben. Wird **Return** in einer Methode aufgerufen, so kehrt die Programmausführung zur aufrufenden Methode zurück, ohne einen Rückgabewert zurückzugeben.

Der Wert, den eine Funktion zurückgibt, kann ein Array oder ein einzelner Wert sein. Wenn Sie ein Array zurückgeben wollen, fügen Sie der Typenbezeichnung leere Klammern an. Wenn Sie zum Beispiel ein Array vom Typ Integer zurückgeben wollen, schreiben Sie Integer() statt nur Integer. Wenn Sie ein mehrdimensionales Array benötigen, schreiben Sie Dimensionen-1 Kommata in die Klammern. Für ein zweidimensionales Array vom Typ Double wäre das Double(,).

Siehe auch

[Function](#) Befehl

RGB Funktion

Liefert ein [Color](#) anhand der übergebenen Werte für rot, grün und blau zurück. Sie eine Farbe auch mit den Modellen [HSV](#) oder [CMY](#) spezifizieren.

Syntax

Ergebnis = **RGB**(rot, gruen, blau)

Teil	Art	Beschreibung
<i>rot</i>	Integer	Der Rot-Anteil der Farbe (0-255).
<i>gruen</i>	Integer	Der Grün-Anteil der Farbe (0-255).
<i>blau</i>	Integer	Der Blau-Anteil der Farbe (0-255).
<i>Ergebnis</i>	Color	Ein Objekt, das die Farbe repräsentiert, die sich aus den Farbanteilen von <i>rot</i> , <i>gruen</i> und <i>blau</i> ergibt.

Hinweise

Die Funktion **RGB** liefert ein Color-Objekt, das die Farbe repräsentiert, die durch die übergebenen Rot-, Grün- und Blau-Farbanteile definiert wird. Diese Anteile sind Integers zwischen 0 und 255.

Sie können eine Farbe im RGB-Modell auch mit Hilfe des [&c](#)-Operators festlegen. Mit dem [&c](#)-Operator legen Sie die Werte von Rot, Grün und Blau mit hexadezimalen statt mit dezimalen Werten fest.

Wenn Sie eine Farbe aus der Farben-Palette draggen, enthält das DragItem die RGB-Werte der Farbe. Diese werden automatisch verwendet, wenn Sie die Farbe in eine Zeile im Code-Editor draggen. Der Code für die Farbe verwendet den [&c](#)-Operator eher als die RGB-Funktion.

Beispiele

Dieses Beispiel verwendet die Funktion **RGB**, um der Vordergrundfarbe eines [Canvas](#) Steuerelements verschiedene Farben zuzuweisen.

```
Dim s As String
Canvas1.Graphics.forecolor=RGB(0,0,0)
//setzt sie auf schwarz
Canvas1.Graphics.forecolor=RGB(255,0,0)
//setzt sie auf rot
Canvas1.Graphics.forecolor=RGB(255,255,255)
//setzt sie auf weiß
```

Siehe auch

[Color](#) Datentyp; [CMY](#), [HSV](#), [SelectColor](#) Funktionen; [&c](#)-Operator.

RGBSurface Object

Wird für direkte Pixel-Manipulationen verwendet. Die RGBSurface-Eigenschaft eines Picture-Objekts erlaubt es, das Bild auf Pixelebene zu manipulieren. Sie kann nur für Bilder verwendet werden, die durch [NewPicture](#) mit einer Farbtiefe von 16 oder 32 erzeugt wurden.

Pixelmanipulationen, die durch die [RGBSurface](#)-Methode vorgenommen werden, sind schneller als gleiche Manipulationen, die mit den Methoden der [Graphics](#)-Klasse durchgeführt werden.

Methoden

Name	Parameter	Beschreibung
<i>Pixel</i>	x als Integer , y als Integer	Liefert die Farbe eines Objektes an der durch die Parameter angegebenen Stelle. Liefert ein Color -Objekt.

Beispiele

Dieses Beispiel ersetzt alle schwarzen Pixel im [Picture](#)-Objekt somePicture mit weißen:

```

Dim surf As RGBSurface
Dim x, y As Integer
surf = somePicture.RGBSurface
For y = somePicture.height - 1 downTo 0
  For x = somePicture.width - 1 downTo 0
    If surf.Pixel(x,y) = &c000000 then surf.Pixel(x,y) = &cFFFFFFF
  next
next

```

Hier wird die FloodFill-Methode verwendet, um ein **RGBSurface**-Objekt in einer zufälligen Farbe zu zeichnen. Dieser Code befindet sich im MouseDown-Event eines Canvas-Steuerelements. Die Variable p ist ein Picture-Objekt.

```

Dim randomColor as Color
randomColor=RGB(Rnd*255,Rnd*255,Rnd*255) //erzeuge zufällige Farbe
p.RGBSurface.FloodFill x,y,randomColor //fülle den Bereich mit der Farbe
Me.Graphics.DrawPicture p,0,0 //zeichne das Canvas-Steuerelement neu

```

Im folgenden Beispiel wird die Transform-Methode verwendet, um ein Bild zu invertieren:

```

Dim map(255), i as Integer
for i = 0 to 255
  map(i) = 255 - i
next
somePicture.RGBSurface.Transform map

```

Hinweis

32-bit RGBSurface-Objekte werden unter Windows in 24-Bit-Objekte umgewandelt.

Siehe auch

RGBSurface Eigenschaft eines Picture-Objekts.

Right Funktion

Liefert die gewünschte Anzahl Zeichen der rechten Seite eines Strings zurück.

Syntax

Ergebnis=**Right**(quelle, anzahl)

ODER

Ergebnis=**StringVariable.Right**(anzahl)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die letzten <i>anzahl</i> Zeichen aus <i>quelle</i> .
<i>quelle</i>	<u>String</u>	Der String, aus dem die Zeichen geholt werden.
<i>anzahl</i>	<u>Integer</u>	Die Anzahl an Zeichen, die Sie aus <i>quelle</i> erhalten möchten. Ist <i>anzahl</i> größer als die Anzahl an Zeichen im String <i>quelle</i> , so werden alle Zeichen zurückgeliefert.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **Right** liefert *anzahl* Zeichen der rechten Seite des Strings *quelle* zurück (daher auch der Name dieser Funktion).

Wenn Sie Bytes anstatt von Zeichen benötigen, sollten Sie die Funktion RightB verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **Right**, um die letzten 4 Zeichen aus einem String zurückzuliefern.

```

Dim s As String
s=Right("Hallo Welt", 4) //liefert "Welt"

```

Siehe auch

Asc, Chr, InStr, Left, Len, Mid Funktionen; String Datentyp.

RightB Funktion

Liefert die gewünschte Anzahl Bytes der rechten Seite eines Strings zurück.

Syntax

Ergebnis=**RightB**(quelle, anzahl)

ODER

Ergebnis=**StringVariable.RightB**(anzahl)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Die letzten <i>anzahl</i> Bytes aus <i>quelle</i> .
<i>quelle</i>	<u>String</u>	Der String, aus dem die Bytes geholt werden.
<i>anzahl</i>	<u>Integer</u>	Die Anzahl an Bytes, die Sie aus <i>quelle</i> erhalten möchten. Ist <i>anzahl</i> größer als die Anzahl an Bytes im String <i>quelle</i> , so werden alle Bytes zurückgeliefert.
<i>StringVariable</i>	<u>String</u>	Eine beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **RightB** liefert *anzahl* Bytes der rechten Seite des Strings *quelle* zurück (daher auch der Name dieser Funktion).

RightB behandelt *quelle* als eine Reihe von Bytes und nicht Buchstaben. Es sollte verwendet werden, wenn *quelle* aus binären Daten besteht.

Wenn Sie Zeichen anstatt von Bytes einlesen möchten, sollten Sie die Funktion Right verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **RightB**, um die letzten 4 Bytes aus einem String zurückzuliefern.

```
Dim s As String
s=RightB("Hallo Welt", 4) //ergibt "Welt"
```

Siehe auch

AscB, ChrB, InStrB, LeftB, LenB, MidB, Right Funktionen; String Datentyp.

Rnd Funktion

Liefert eine zufällige Zahl zwischen 0 und 1. Diese Funktion ist obsolet. Verwenden Sie stattdessen die Random Klasse.

Siehe auch

Random Klasse.

Round Funktion

Rundet einen bestimmten Wert zum nächsten Integer.

Syntax

Ergebnis=**Round** (Wert)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Der gerundete <i>Wert</i> .
<i>Wert</i>	<u>Double</u>	Der zu rundende Wert.

Hinweis

Die Funktion **Round** rundet den übergebenen Wert zum nächsten Integer und liefert diesen zurück.

Beispiele

Dieses Beispiel verwendet die Funktion **Round**, um eine gerundete Zahl zurückzuliefern.

```
Dim d as double
d=Round(1.499) //liefert 1
d=Round(1.500) //liefert 2
```

Siehe auch

Ceil, Floor Funktionen.

RoundRectangle Steuerelement



Zeichnet ein Rechteck mit abgerundeten Ecken.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Eigenschaften

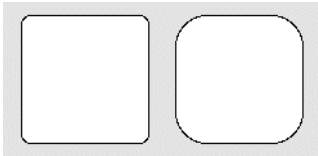
Name	Art	Beschreibung
<i>BorderWidth</i>	Integer	Die Breite des Randes des abgerundeten Rechtecks in Pixeln.
<i>FillColor</i>	Color	Die Füllfarbe des Rechtecks.
<i>OvalHeight</i>	Integer	Die Höhe der abgerundeten Ecken.
<i>OvalWidth</i>	Integer	Die Breite der abgerundeten Ecken.

Events

Name	Parameter	Beschreibung
<i>MouseDown</i>	x als Integer , y als Integer	Die Maustaste wurde innerhalb der Region des RoundRectangle an der gelieferten Stelle x,y gedrückt. Wenn Sie MouseDown auswerten möchten, liefern Sie True zurück.
<i>MouseDrag</i>	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des RoundRectangle-Steuerelements gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als Integer , y als Integer	Die Maustaste wurde innerhalb der Region des RoundRectangle an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht True im MouseDown-Event liefern.

Hinweis

Wenn Sie OvalWidth und OvalHeight erhöhen, bekommen die abgerundeten Ecken einen gleichmäßigeren Bogen. Im Bild hat das linke Rechteck für beide den Wert 16, das rechte Rechteck den Wert 50.



Beispiel

Folgender Code erzeugt ein benutzerdefiniertes RoundRectangle:

```
roundrectangle1.fillColor=RGB(255,0,0) //Rot
roundrectangle1.ovalheight=30
roundrectangle1.ovalwidth=16
roundrectangle1.borderwidth=1
```

Siehe auch

[Rectangle](#) Steuerelement, [RectControl](#) Klasse.

RoundRectShape Klasse

Zeichnet ein (zweidimensionales) Rechteck mit abgerundeten Ecken in einer Vektorgrafikumgebung.

Übergeordnete Klasse

[RectShape](#)

Eigenschaften

Name	Typ	Beschreibung
Cornerheight	<u>Double</u>	Höhe des ovalen Eckeneinsatzes.
Cornerwidth	<u>Double</u>	Breite des ovalen Eckeneinsatzes.
Segments	<u>Integer</u>	Die Anzahl der Polygon-Seiten, die verwendet werden, wenn eine Annäherung benötigt wird. Null bedeutet, daß es dem Rendereer überlassen wird.

Beispiel

Folgende Methode im Paint-Event eines Fensters zeichnet ein Quadrat mit abgerundeten Ecken.

```
Dim r as RoundRectShape
r=New RoundRectShape
r.width=75
r.height=75
r.border=100
r.bordercolor=RGB(0,0,0)
r.fillcolor=RGB(255,102,102)
r.cornerHeight=15
r.cornerWidth=15
r.borderWidth=2.5
g.DrawObject r,150,150
```

Siehe auch

[ArcShape](#), [CurveShape](#), [FigureShape](#), [FolderItem](#), [Group2D](#), [Graphics](#), [OvalShape](#), [Picture](#), [PixmapShape](#), [RectShape](#), [RoundRectShape](#), [StringShape](#) Klassen.

RTrim Funktion

Liefert den übergebenen String-Datentyp ohne nachfolgende Leerzeichen zurück.

Syntax

Ergebnis=RTrim(uellString)

ODER

Ergebnis=StringVariable.RTrim

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Der quellString ohne nachfolgende Leerzeichen.
<i>quellString</i>	<u>String</u>	Der String, der um nachfolgende Leerzeichen bereinigt werden soll.

Beispiele

Dieses Beispiel entfernt die nachfolgenden Leerzeichen (von der rechten Seite) des übergebenen Strings:

```
Dim s as String
s=RTrim(" Hallo Welt ") //liefert „ Hallo Welt“
```

Siehe auch

[LTrim](#), [Trim](#), [Asc](#), [Chr](#), [InStr](#), [Len](#), [Left](#), [Mid](#) Funktionen; String Datentyp.

Runtime Objekt

Liefert Informationen über den aktuellen Zustand der Laufzeitumgebung. Diese Informationen können für Debug-Zwecke nützlich sein.

Eigenschaften

Name	Art	Beschreibung
<i>ObjectCount</i>	<u>Integer</u>	Aktuelle Anzahl existierender Objekte.
<i>ObjectClass</i>	<u>String</u>	Erwartet ObjectNumber (<u>Integer</u>) als Parameter, wobei ObjectNumber größer oder gleich null und kleiner als ObjectCount ist. Liefert die Klasse des übergebenen Objekts.

Name	Art	Beschreibung
<i>ObjectRefs</i>	<u>Integer</u>	Erwartet ObjectNumber (<u>Integer</u>) als Parameter, wobei ObjectNumber größer oder gleich null und kleiner als ObjectCount ist. Liefert die Anzahl an Referenzen auf das übergebene Objekt.
<i>ObjectID</i>	<u>Integer</u>	Erwartet ObjectNumber (<u>Integer</u>) als Parameter, wobei ObjectNumber größer oder gleich null und kleiner als ObjectCount ist. Liefert einen eindeutigen Bezeichner für das übergebene Objekt.
<i>MemoryUsed</i>	<u>Integer</u>	Liefert die Gesamtzahl an genutztem Speicher (in Bytes), der von den angelegten Objekten belegt wird.

Beispiel

Folgender Code zeigt die Eigenschaften von den zugehörigen individuellen Objekten in einer mehrspaltigen ListBox an und die Gesamtzahl an benutztem Speicher in einem EditField:

```
Dim i, total as integer
total=Runtime.ObjectCount
For i=0 to total-1
    Listbox1.addrow Str(Runtime.ObjectID(i))
    Listbox1.Cell(i,1)=Runtime.ObjectClass(i)
    Listbox1.Cell(i,2)=Str(Runtime.ObjectRefs(i))
Next
EditField1.text=Str(Runtime.memoryUsed)
```

Siehe auch

System Objekt.

RuntimeException Klasse

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
ErrorNumber	<u>Integer</u>	Enthält eine Fehlernummer, die den Laufzeitfehler beschreibt.
<i>Message</i>	<u>String</u>	Wird für beschreibenden Text verwendet, der angezeigt wird, wenn die Laufzeitfehlermeldung auftritt.

Hinweise

Verwenden Sie Runtime-Exceptions, um Fehler aufzudecken und auf diese zu reagieren. Wenn Sie beispielsweise auf ein Array-Element zugreifen wollen, das nicht existiert, wird eine OutOfBoundsException ausgelöst. Exceptions haben keine Methoden oder Events.

Name	Beschreibung
<u>IllegalCastException</u>	Sie haben einem Objekt eine andere Klasse zugewiesen und ihm eine Nachricht geschickt, die seine richtige Klasse nicht annehmen kann.
<u>KeyNotFoundException</u>	Es wurde versucht, auf ein Dictionary-Objekt mit einem Schlüssel zuzugreifen, den das Dictionary nicht enthält.
<u>NilObjectException</u>	Es wurde versucht, auf ein nicht existierendes Objekt zuzugreifen.
<u>OutOfBoundsException</u>	Es wurde versucht, einen Wert auszulesen oder zu schreiben, der außerhalb der Grenzen des Objekts oder des Datentyps liegt.
<u>OutOfMemoryException</u>	Wird in bestimmten Fällen aufgerufen, wenn eine Operation wegen Speichermangel nicht vollendet werden kann.
<u>RbScriptAlreadyRunningException</u>	Der Anwender hat versucht, ein RbScript zu modifizieren, das schon ausgeführt wird oder versucht, den Kontext des Skriptes zu ändern, während es läuft.

Name	Beschreibung
<u>RegexException</u>	Die RegEx-Engine hat einen Laufzeitfehler bemerkt. Derzeit bedeutet das, daß sie ein ungültiges SearchPattern in einem Regulären Ausdruck verwendet haben. In künftigen Versionen werden weitere Arten von Fehlerbehandlungen hinzugefügt.
<u>ShellNotRunningException</u>	Sie haben versucht, auf eine asynchrone oder interaktive Shell-Session zuzugreifen, die Shell war aber nicht am Laufen.
<u>StackOverflowException</u>	Wenn eine Routine (Methode/Event-Handler/Menü-Handler) eine andere Routine aufruft, merkt sich der Rechner im Speicher, von wo aus die Routine aufgerufen wurde und welche lokalen Werte sie kennt. Dies geschieht, damit aus der Routine zur Ursprungsroutine zurückgekehrt werden kann und die Ursprungsroutine alles in dem Zustand vorfindet, wie sie verlassen wurde. Der Speicher, in dem diese Informationen gespeichert werden, ist ein Stapel (im Englischen „Stack“ genannt, in Anlehnung an Heuballen, die in Bauernhöfen mitunter gestapelt werden, um die Effizienz bei der Raumnutzung möglichst optimal zu halten, weniger aber, um die Aufmerksamkeit von Passanten auf den Hof zu lenken). Die Informationen aller ineinander aufgerufenen Routinen werden im Stack „aufeinandergestapelt“. Wenn Ihrer Anwendung dieser Speicherplatz ausgeht, wird eine StackOverflowException ausgelöst. Um das Auftreten dieses Fehlers zu vermeiden, sollten Sie Ihre Anwendung sehr genau prüfen.
<u>TypeMismatchException</u>	Sie haben versucht, einem Objekt den falschen Datentyp zuzuweisen.
<u>UnsupportedFormatException</u>	Sie haben einen String-Ausdruck verwendet, der nicht zu einer Zahl ausgewertet werden kann oder Sie haben versucht, ein nicht unterstütztes Bild-Format zu öffnen oder zu speichern.

Falls es nicht klappt, einen Laufzeitfehler in einem Exception Block abzufangen, wird das UnhandledException-Event der Application-Klasse aufgerufen. Sie können dann alle Laufzeitfehler generell innerhalb dieses Event-Handlers behandeln.

Wenn Sie die Office Automation Plugins installieren, werden Exception-Klassen für jedes der drei Plugins Teil der Sprache: WordException, ExcelException und PowerPointException. Jede dieser Klassen besitzt zwei Eigenschaften:

Name	Art	Beschreibung
Message	<u>String</u>	Weist auf den Befehl hin, der ausgeführt wurde, als der Fehler auftrat.
Code	<u>Integer</u>	Ein OLE Fehler-Code.

Wenn Sie die Office Automation Plugins verwenden, können Sie einen Exception Block einbauen, der auf diese Klassen verweist. Zum Beispiel:

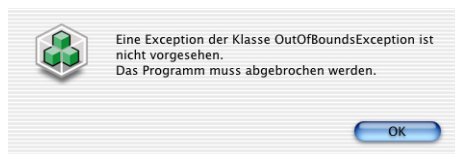
```
Exception err as WordException
MsgBox err.message+" Fehler-Nummer: "+Str(err.code)
```

Plugins von Fremdanbietern können ebenfalls eigene Arten von Runtime-Exceptions einbringen.

Wenn Sie Ihre Applikation in der IDE testen, wird die Ausführung bei der betreffenden Zeile stoppen und eine Fehlermeldung in Ihrem Code-Editor anzeigen. Wenn der Fehler in einer eigenständigen Applikation auftritt, wird der Anwender eine Fehler-Meldung sehen und das Programm wird beendet – es sei denn, Sie bauen einen Exception-Handler ein.

Hinweis

Runtime Exceptions werden üblicherweise verwendet, um Programmierfehler aufzudecken. Ohne einen Handler für so eine Exception bekommt der Anwender von REALbasic einen vorgefertigten Dialog gezeigt, daß eine bestimmte Art von Exception aufgetreten ist und die Anwendung beendet wird.



Das Programm wird beendet, wenn der Anwender den Dialog bestätigt.

Indem Sie Exception-Handler in ihrem Programm verwenden, können Sie stattdessen auch aussagekräftigere Informationen ausgeben lassen, mit deren Hilfe Sie dem Grund des Problems leichter auf die Spur kommen können und Ihre Anwendung auch weiterlaufen lassen können.

Wenn Sie beispielsweise beim Zugriff auf ein Element eines Arrays eine „[OutOfBoundsException](#)“ erhalten, können Sie in einem Exception-Handler die Anzahl der Arrayelemente und die Nummer des Elements ausgeben, auf das Sie zugreifen wollten. Dadurch kommen Sie der Quelle des Fehlers näher. Ein Beispiel können Sie im Abschnitt [OutOfBoundsException](#) finden.

Es gibt ein weiteres Beispiel für die Behandlung von Exceptions im Abschnitt von [IllegalCastException](#).

Beispiele

Dieses Beispiel beinhaltet einen [If](#)-Befehl, der die Art der [Exception](#) überprüft und eine Warnbox anzeigt, die angibt, welcher Fehler-Typ aufgetreten ist. Der [Exception](#) Block beginnt nach der letzten Zeile der Methode. Beachten Sie, daß seine Einrückung die gleiche Ebene wie der Sub-Befehl hat.

```
Sub Action
    Dim a(3) as String
    a(4)="geoff"
Exception err
    If err IsA NilObjectException then
        MsgBox "Nil Object Exception"
    elseif err IsA OutOfBoundsException then
        MsgBox "Out of Bounds"
    elseif err IsA TypeMismatchException then
        MsgBox "Type Mismatch"
    End if
End Sub
```

Anstelle des [If](#)-Befehls können Sie auch verschiedene [Exception](#) Blocks verwenden, von denen jeder einen spezifischen Typ von Laufzeitfehler behandelt.

```
Dim f as FolderItem
f.delete
Exception err as NilObjectException
MsgBox "Nilobjectexception"
Exception err as OutOfBoundsException
MsgBox "Out of Bounds"
Exception err as TypeMismatchException
MsgBox "typemismatch"
```

Folgendes Beispiel prüft den Typ der Exception und behandelt die Exception, als wäre es eine [NilObjectException](#). Wenn die Exception keine [NilObjectException](#) ist, läßt es die Ausführung durch dem Befehl [Raise](#) fortfahren.

```
as FolderItem
f.Delete
Exception err
If err IsA NilObjectException then
    MsgBox "Hoppala!"
else
    Raise err
end if
```

Siehe auch

[Function](#), [Raise](#) Befehl; [NilObjectException](#), [IllegalCastException](#), [TypeMismatchException](#), [OutOfBoundsException](#), [StackOverflowException](#) Fehler; [Exception](#) Block, [Nil](#) Funktion.

SaveAsDialog Klasse

Erzeugt eine benutzerdefinierbare „Sichern unter“-Dialogbox. Die nicht-benutzerdefinierbare Version dieses Dialogs wird mit der Funktion [GetSaveFolderItem](#) angezeigt.

Übergeordnete Klasse

[FolderItem](#) Dialog

Hinweise

Durch die Eigenschaften der FolderItemDialog-Klasse können Sie folgende Aspekte einer „Sichern unter“-Dialogbox selbst definieren:

- Position (Eigenschaften Left und Top)
- Ausgangsverzeichnis (Eigenschaft Initial Directory)
- Dateinamen-Vorgabe (Eigenschaft SuggestedFileName)
- Gültige Dateitypen, die angezeigt werden sollen (Eigenschaft Filter)
- Beschriftung der Bestätigen- und Abbrechen-Knöpfe (Eigenschaften ActionButtonCaption und CancelButtonCaption)
- Text, der im Titelfeld des Dialogs erscheint (Eigenschaft Title)
- Text, der innerhalb des Dialogs erscheint (Eigenschaft PromptText)

Unter Mac OS X erscheint eine Checkbox „Suffix ausblenden“ in der „Sichern unter“-Dialogbox. Das zurückgelieferte FolderItem bekommt die ExtensionVisible-Eigenschaft passend zur Wahl des Anwenders gesetzt.

Beispiel

Folgender Code öffnet eine benutzerdefinierbare „Sichern unter“-Dialogbox und zeigt den Inhalt des Dokumente-Verzeichnisses im Datei-Browser an.

```
Dim dlg as SaveAsDialog
Dim f as FolderItem
dlg=new SaveAsDialog
dlg.InitialDirectory=Volume(0).Child("Dokumente")
dlg.promptText="Geben Sie einen Dateinamen ein"
dlg.SuggestedFileName="RTF Datei"
dlg.Title="Speichern Sie Ihre Datei"
f=dlg.ShowModal()
If f <> Nil then
    //Datei gespeichert
Else
    //Benutzerabbruch
End if
```

Siehe auch

[FolderItemDialog](#), [OpenDialog](#), [SelectFolderDialog](#), [FolderItem](#) Klassen; [GetSaveFolderItem](#) Funktion.

Screen Klasse

Wird verwendet, um Informationen über die Monitore Ihres Rechners abzufragen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Art	Beschreibung
<i>AvailableHeight</i>	<u>Integer</u>	Die verfügbare Höhe in Pixeln des angegebenen Bildschirms unter Berücksichtigung der Menüleiste und/oder des Docks, falls vorhanden.
<i>AvailableLeft</i>	<u>Integer</u>	Die Anzahl der nutzbaren Pixel von der linken Kante des angegebenen Bildschirms. Dies wäre Null, außer, der Anwender hätte das Dock an der linken Seite platziert.
<i>AvailableTop</i>	<u>Integer</u>	Die Anzahl der nutzbaren Pixel von der oberen Kante des angegebenen Bildschirms unter Berücksichtigung der Höhe der Menüleiste, falls eine auf dem angegebenen Bildschirm vorhanden ist.
<i>AvailableWidth</i>	<u>Integer</u>	Die verfügbare Breite in Pixeln des angegebenen Bildschirms unter Berücksichtigung des Docks, falls dieses auf dem angegebenen Bildschirm links oder rechts vorhanden ist.
<i>Depth</i>	<u>Integer</u>	Die Farbtiefe des Bildschirms.
<i>Height</i>	<u>Integer</u>	Die Höhe des Bildschirms.

Name	Art	Beschreibung
Left	<u>Integer</u>	Die Koordinaten der oberen, linken Ecke des Bildschirms relativ zum Hauptbildschirm.
Top	<u>Integer</u>	Die Koordinaten der oberen, rechten Ecke des Bildschirms relativ zum Hauptbildschirm.
Width	<u>Integer</u>	Die Breite des Bildschirms.

Hinweis

Obwohl Sie kein Screen-Objekt anlegen können, können Sie auf Screen-Objekte mit Hilfe der Funktion Screen zugreifen. Bei der Beschreibung der Funktionen Screen und ScreenCount finden Sie weitere Informationen.

Beispiel

Folgendes Beispiel ermittelt die Werte von AvailableLeft, AvailableHeight, AvailableTop und AvailableWidth für den Hauptbildschirm. Falls der Anwender die Option eingeschaltet hat, daß das Dock ausgeblendet wird, wird die Größe des Docks, wenn es vorübergehend sichtbar gemacht wurde, nicht in die Berechnung miteinbezogen.

```
Dim s as String
s="Left="+Str(Screen(0).availableleft)+Chr(13)
s=s+"Width="+Str(Screen(0).availablewidth)+Chr(13)
s=s+"Top="+Str(Screen(0).availabletop)+Chr(13)
s=s+"Height="+Str(Screen(0).availableheight)+Chr(13)
Me.text=s
```

Screen Funktion

Wird verwendet, um auf die Eigenschaften eines Screen-Objektes zugreifen zu können. Liefert einen Verweis auf den übergebenen Bildschirm zurück.

Syntax

Screen(index).property

Teil	Art	Beschreibung
<i>index</i>	<u>Integer</u>	Die Nummer des Bildschirms, dessen Eigenschaften Sie auslesen möchten. Hierbei steht 0 für den Hauptbildschirm (der Bildschirm mit der Menüleiste).
<i>property</i>	beliebiger Name einer Eigenschaft eines Screen-Objekts	Die Eigenschaft, die Sie auslesen möchten. Siehe Beschreibung der <u>Screen</u> Klasse.

Hinweis

Obwohl Sie kein Screen-Objekt anlegen können, können Sie mit der Funktion **Screen** die Eigenschaften des am Rechner des Anwenders angeschlossenen Monitors abfragen. Bildschirm 0 ist der Hauptbildschirm (der mit der Menüleiste). Sie können die Funktion ScreenCount verwenden, um herauszufinden, wieviele Bildschirme existieren.

Beispiele

Dieses Beispiel zeigt die Größe des Hauptbildschirms des Anwenders an.

```
MsgBox "Ihr Bildschirm hat die Größe "
+Str(screen(0).width)+" * "
+Str(screen(0).height)+". "
```

Siehe auch

Screen Klasse.

ScreenCount Funktion

Wird verwendet, um die Anzahl von Bildschirmen herauszufinden, die am Rechner des Anwenders angeschlossen sind.

Syntax

Ergebnis = **ScreenCount**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Etwas, was einen <u>Integer</u> -Wert aufnehmen kann.

Hinweis

Die Funktion **ScreenCount** liefert die Anzahl der Bildschirme (Monitore), die am Rechner des Anwenders angeschlossen sind.

Beispiel

Dieses Beispiel meldet die Anzahl der am Rechner des Anwenders angeschlossenen Bildschirme.

```
Dim myscreens as Integer
myscreens=ScreenCount
If (myscreens=1) then
    MsgBox "Sie haben nur einen Monitor. Wo liegt das Problem?"
else
    MsgBox "Sie haben "+Str(myscreens)+" Bildschirme!"
end if
```

Siehe auch

Screen Klasse; Screen Funktion.

Scrollbar Steuerelement



Das Scrollbar-Steuerelement.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Art	Beschreibung
LiveScroll	<u>Boolean</u>	Wenn <u>True</u> , wird bei jeder Änderung des Anwenders ein ValueChanged-Event ausgelöst. Andernfalls wird ein einzelnes ValueChanged-Event ausgelöst, sobald der Anwender den Schieberegler losläßt.
<i>Maximum</i>	<u>Integer</u>	Der Maximalwert der Scrollbar. Voreinstellung ist 100. Für Mac OS 8.5 oder neuer ist das Maximum weitaus größer.
<i>Minimum</i>	<u>Integer</u>	Der Minimalwert der Scrollbar. Voreinstellung ist 0. Bitte beachten Sie den Hinweis bezüglich Mac OS.
<i>PageStep</i>	<u>Integer</u>	Der Betrag, um den sich der Wert verändert, wenn der Anwender in den Scrollbalkenbereich klickt.
<i>LineStep</i>	<u>Integer</u>	Der Betrag, um den sich der Wert verändert, wenn ein Scrollpfeil angeklickt wird. Voreinstellung ist 1.
<i>Value</i>	<u>Integer</u>	Der aktuelle Wert der Scrollbar.

Events

Name	Beschreibung
<i>GotFocus</i>	(Nur Win32) Das Steuerelement hat den Fokus bekommen und besitzt einen Selektionsrahmen um die Beschriftung.
<i>LostFocus</i>	(Nur Win32) Das Steuerelement hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u> Der Mausknopf wurde innerhalb der ScrollBar an der Position x,y gedrückt. Gibt ein Boolean zurück. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen. Die Koordinaten x und y sind lokal zum Steuerelement. Der Punkt 0,0 ist die linke obere Ecke des gesamten Steuerelements (links von der Beschriftung und über der Box, die gezeichnet wurde).

Name		Beschreibung
<i>MouseDown</i>	x als <u>Integer</u>	Der Mausknopf wurde innerhalb der ScrollBar gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt).
	y als <u>Integer</u>	Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u>	Der Mausknopf wurde innerhalb der ScrollBar an der Position x,y losgelassen.
	y als <u>Integer</u>	Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>ValueChanged</i>		Der Wert hat sich geändert.

Hinweis

Die Steuerelemente-Palette hat separate Werkzeuge für horizontale und vertikale Scrollbalken. Jeder Scrollbalken kann in die andere Ausrichtung rotiert werden, nachdem er einem Fenster hinzugefügt wurde.

Beispiele

Ändern des Maximalwerts eines **Scrollbars** zur Laufzeit:

```
ScrollBar1.maximum=200
```

Setzt den Text eines StaticText-Steuerelements auf den Wert der **Scrollbars**, während der Benutzer scrollt:

```
Sub ValueChanged()  
    staticText1.text=Str(scrollbar1.Wert)  
End Sub
```

Sehen Sie auch im Abschnitt über das horizontale Scrolling bei ListBox-Steuerelementen mit einem **Scrollbars**-Steuerelement nach.

Siehe auch

Slider Steuerelement.

Select Case Befehl

Führt in Abhängigkeit vom Wert eines Ausdrucks eine von mehreren Befehlsgruppen aus.

Syntax

Select Case TestAusdruck

[Case Ausdruck-n

Anweisungen-n]

[Else

elseAnweisungen]

End [Select]

Teil	Beschreibung
<i>TestAusdruck</i>	Beliebiger Ausdruck, der einen Wert zurückliefert. Dieser kann ein beliebiger Datentyp oder ein Objekt sein.
<i>Ausdruck-n</i>	Beliebiger Ausdruck oder Liste von Ausdrücken. Sie können eine Funktion verwenden, um den Datentyp von TestAusdruck zu ermitteln.
<i>Anweisungen-n</i>	Befehle, die ausgeführt werden, wenn Ausdruck-n <u>True</u> ist.
<i>elseAnweisungen</i>	Befehle, die ausgeführt werden, wenn kein Ausdruck <u>True</u> ist.

Hinweis

Der Befehl **Select Case** ist dann sinnvoll, wenn es mehrere Bedingungen gibt, die überprüft werden müssen. Im Gegensatz zum Befehl **If** beendet der Befehl **Select Case** seine Tätigkeit, sobald ein passender **Case Ausdruck (Expression)** gefunden wurde und führt daraufhin alle Befehle aus, die dem **Case-Ausdruck** bis zum nächsten **Case-Ausdruck** folgen. Wenn kein **Case-Ausdruck** zutrifft, werden die *elseAnweisungen* ausgeführt.

Der **Case**-Befehl akzeptiert verschiedene Arten von Ausdrücken. Ein Ausdruck kann ein einzelner Wert, eine komma-separierte Liste von Werten, eine Funktion mit einem Rückgabewert, ein Wertebereich bei der Verwendung des „To“- Schlüsselwortes sein. Ausdrücke

mit dem „Is“-Schlüsselwort sind ebenfalls erlaubt. Sie können die Ausdrücke auch kombinieren, indem Sie diese mit einem Komma voneinander trennen.

Hier sind einige Beispiele:

```

Case 2, 4, 6, 8 //verschiedene Werte
Case 2 To 5 //ein Bereich von Werten unter Verwendung von To
Case 2 To 5, 7,9,11 //Beides, verschiedene Werte und Bereich
Case myFunction(x) // eine Funktion
Case Is >= 42 // größer als/gleich-Operator
Case Is <19 //kleiner als-Operator

```

Beispiele

Dieses Beispiel verwendet den Befehl **Select Case**, um herauszufinden, welchen Wochentag der Anwender in ein EditField eingetragen hat und zeigt dementsprechend eine Nachricht an.

```

Dim dayZahl As Integer
Dim msg As String
dayZahl=Val(EditField1.text)
Select Case dayZahl
Case 2
    msg="Es ist Montag."
Case 3
    msg="Es ist Dienstag."
Case 4
    msg="Es ist Mittwoch."
Case 5
    msg="Es ist Donnerstag."
Case 6
    msg="Es ist Freitag."
Else
    msg="Es ist Wochenende."
End Select
MsgBox msg

```

Das folgende Beispiel verwendet einen **Select Case**-Ausdruck, um festzustellen, welcher Knopf in einem MessageDialog gedrückt wurde. Beachten Sie, daß hier Objekte vom Typ MessageDialogButton verglichen werden.

```

Dim d as New MessageDialog //deklariere das MessageDialog-Objekt
Dim b as MessageDialogButton //um das Ergebnis zu verwalten
d.icon=1 //zeigt Warn-Icon
d.ActionButton.Caption="Speichern"
d.CancelButton.Visible=True //zeigt den Abbrechen-Knopf
d.CancelButton.Cancel=True //esc-Taste funktioniert für Abbrechen
d.AlternateActionButton.Visible=True //zeigt den "Nicht sichern"-Knopf
d.Message="Änderungen vor dem Schließen speichern?"
d.Explanation="Wenn Sie Ihre Änderungen nicht sichern, geht "+_
    +" die gesamte wichtige Arbeit seit der letzten Kaffeepause verloren."
b=d.ShowModal //Dialog anzeigen
Select Case b //b ist einMessageDialogButton
Case d.ActionButton //ermittelt, welcher Knopf gedrückt wurde.
    //Speichern wurde angeklickt
Case d.AlternateActionButton
    //Nicht speichern wurde angeklickt
Case d.CancelButton
    //Abbrechen wurde angeklickt
End Select

```

Siehe auch

If...Then...Else Befehl.

Select4DDatabaseByADSP Funktion

Wird verwendet, um mit Hilfe eines Öffnen-Dialogs eine 4D-Datenbank über ADSP zu öffnen. Diese Funktion ist obsolet. Verwenden Sie bitte die ADSP4DServer Klasse.

Siehe auch

ADSP4DServer Klasse.

Select4DDatabaseByTCPIP Funktion

Wird verwendet, um mit Hilfe eines Öffnen-Dialogs eine 4D-Datenbank über TCP/IP zu öffnen. Diese Funktion ist obsolet. Bitte verwenden Sie die [TCPIP4DServer](#) Klasse.

Siehe auch

[TCPIP4DServer](#) Klasse.

SelectColor Funktion

Zeigt den Standard-Color-Picker an und erlaubt dem Anwender, eine Farbe zu wählen.

Syntax

result=**SelectColor**([ByRef](#) col, prompt)

Teil	Typ	Beschreibung
<i>result</i>	Boolean	Liefert True , wenn der Anwender OK klickt; False , wenn er auf Abbrechen klickt.
<i>col</i>	Color	Enthält beim Aufruf die Farbe, die dem Color-Picker übergeben wird und bei Rückkehr aus der Funktion die Farbe, die der Anwender ausgewählt hat. Beachten Sie die Verwendung von ByRef in der Parameterliste.
<i>prompt</i>	String	Text, der im Color-Picker angezeigt werden soll.

Hinweise

Sie legen die Vorgabefarbe, die beim Öffnen des Color-Pickers eingestellt sein soll, mit dem Wert *col* fest. Wenn *col* kein Wert zugewiesen wurde, wird schwarz als Vorgabe verwendet. Die Farbe, die der Anwender wählt, wird in *col* zurückgeliefert. Wenn der Anwender auf Abbrechen klickt, ohne eine Auswahl getroffen zu haben, wird result auf *False* gesetzt; andernfalls wird es auf *True* gesetzt.

Beispiel

Folgendes Beispiel erlaubt dem Anwender, eine Farbe auszuwählen, die dann als Füllfarbe in einem Rectangle verwendet wird.

```
Dim c as Color
Dim b as Boolean
c=CMY(.35,.9,.6)
b=SelectColor(c,"Wähle eine Farbe")
Rectangle1.FillColor=c
```

Siehe auch

[Color](#) Datentyp; [CMY](#), [RGB](#), [HSV](#) Funktionen.

SelectFolder Funktion

Zeigt eine Datei-Öffnen-Dialogbox an, die dem Anwender die Auswahl eines Ordners statt einer Datei erlaubt.

Syntax

Ergebnis=**SelectFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Der vom Anwender ausgewählte Ordner.

Hinweis

Die Funktion **SelectFolder** zeigt eine Dialogbox, die der Dialogbox aus der Funktion [GetOpenFolderItem](#) stark ähnelt. Der Unterschied ist, daß die **SelectFolder**-Dialogbox die Auswahl eines Ordners statt einer Datei erlaubt.

Wenn der Anwender Mac OS 8.5 oder neuer installiert hat, wird automatisch der neue Navigation Services Browser als verschiebbarer modaler Dialog angezeigt. Er wird auf der folgenden Abbildung gezeigt.



Beispiele

Dieses Beispiel zeigt den Namen des vom Anwender ausgewählten Ordners:

```
Dim f as FolderItem
f=SelectFolder
If f<> Nil Then
    MsgBox f.name
End If
```

Siehe auch

[GetOpenFolderItem](#), [GetSaveFolderItem](#) Funktionen.

SelectFolderDialog Klasse

Erzeugt eine benutzerdefinierbare „Ordner wählen“-Dialogbox. Die nicht-benutzerdefinierbare Version dieses Dialogs wird mit der Funktion [SelectFolder](#) angezeigt.

Übergeordnete Klasse

[FolderItemDialog](#) Klasse

Hinweise

Durch die Eigenschaften der FolderItemDialog-Klasse können Sie folgende Aspekte einer „Ordner wählen“-Dialogbox selbst definieren:

- Position (Eigenschaften Left und Top)
- Ausgangsverzeichnis (Eigenschaft Initial Directory)
- Gültige Dateitypen, die angezeigt werden sollen (Eigenschaft Filter)
- Beschriftung der Bestätigen- und Abbrechen-Knöpfe (Eigenschaften ActionButtonCaption und CancelButtonCaption)
- Text, der im Titelbalken des Dialogs erscheint (Eigenschaft Title)
- Text, der innerhalb des Dialogs erscheint (Eigenschaft PromptText)

Beispiel

Folgender Code öffnet eine benutzerdefinierbare „Ordner wählen“-Dialogbox und zeigt den Inhalt der Ordner Dokumente auf dem Start-Volumen im Browser an.

```
Dim dlg as SelectFolderDialog
Dim f as FolderItem
dlg=New SelectFolderDialog
dlg.ActionButtonCaption="Wähle"
dlg.InitialDirectory=Volume(0).Child("Dokumente")
f=dlg.ShowModal()
if f <> Nil then
    //Hier das FolderItem verwenden
else
    //Benutzer-Abbruch
end if
```

Siehe auch

[FolderItemDialog](#), [OpenDialog](#), [SaveAsDialog](#), [FolderItem](#) Klassen; [SelectFolder](#) Funktion.

SelectODBCDatabase Funktion

Ermöglicht es dem Benutzer eine ODBC-Datenbank zu öffnen. Diese Funktion ist obsolet. Bitte verwenden Sie die [ODBCDatabase](#) Klasse und übergeben einen leeren String in die DataSource-Eigenschaft.

Siehe auch

[ODBCDatabase](#) Klasse.

Self Schlüsselwort

Self ist eine Referenz auf das einem Objekt übergeordnete Objekt.

Hinweis

Wenn **Self** aus der Methode eines Objekts heraus aufgerufen wird, ist es immer eine Referenz auf das dem Objekt übergeordnete Fenster. Dies bedeutet für den Event-Handler „Action“ eines PushButton-Steuerelements:

Left=100 und **Self**.Left=100

bewirken das gleiche, da kein Objekt-Bezeichner (object identifier) existiert und deshalb das übergeordnete Objekt (in diesem Fall das Fenster, dessen Teil der PushButton ist) verwendet wird.

Bei Code für ein Steuerelement in einem Fenster verweist „Self“ auf das Fenster und „Me“ auf das Steuerelement.

Wenn **Self** aus der Methode einer Klasse heraus aufgerufen wird, ist **Self** eine Referenz auf die Instanz der verwendeten Klasse. Dies ist normalerweise nicht notwendig, da – wie bereits beschrieben – ohne Object-Bezeichner das übergeordnete Objekt angenommen wird.

Siehe auch

[Me](#) Schlüsselwort.

Self kann nicht in der Methode eines Modules verwendet werden Fehler

Sie können [Self](#) oder [Me](#) nicht in einer Methode oder Funktion in einem Modul verwenden, da es kein Eltern-Fenster oder Eltern-Steuerelement gibt.

Siehe auch

[Self](#) Funktion.

Semaphore Klasse

Wird verwendet, um den Zugriff auf Shared Code in einer Multithreading-Umgebung zu kontrollieren.

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
<i>Signal</i>		Wird aufgerufen, um in den Besitz einer Semaphore zu gelangen.
<i>Release</i>		Wird aufgerufen, um den Zugang zu einer Semaphore freizugeben.

Hinweise

Eine Semaphore ist ein Objekt, das verwendet werden kann, um den Zugang zu gemeinsam genutzten Ressourcen zu synchronisieren.

Um in den Besitz einer Semaphore zu gelangen, ruft ein Thread die Signal-Methode auf. Wenn die Semaphore verfügbar ist, wird sie dem Thread zugeordnet, andernfalls wird der Thread gezwungen zu warten, bis die Semaphore mit der Release-Methode freigegeben wird.

Siehe auch

[CriticalSection](#), [Thread](#) Klasse

Separator Steuerelement



Wird verwendet, um Objekte auf einem Fenster optisch voneinander zu separieren.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

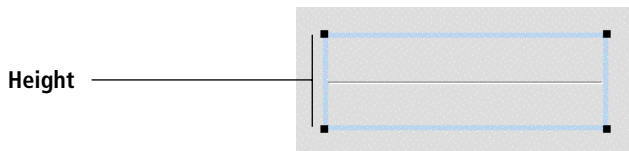
Eigenschaften

Name	Typ	Beschreibung
<i>Height</i>	Integer	Die Höhe des Separators.
<i>Left</i>	Integer	Der Abstand in Pixeln vom linken Rand des Separators zum linken Rand des Fensters.
<i>LockBottom</i>	Boolean	Legt fest, ob sich die untere Kante des Steuerelements immer in einer festen Distanz zur unteren Kante des zugehörigen Fensters befinden soll.
<i>LockLeft</i>	Boolean	Legt fest, ob sich die linke Kante des Steuerelements immer in einer festen Distanz zur linken Kante des zugehörigen Fensters befinden soll. LockLeft hat keine Auswirkung, solange LockRight True ist.
<i>LockRight</i>	Boolean	Legt fest, ob sich die rechte Kante des Steuerelements immer in einer festen Distanz zur rechten Kante des zugehörigen Fensters befinden soll.
<i>LockTop</i>	Boolean	Legt fest, ob sich die obere Kante des Steuerelements immer in einer festen Distanz zur oberen Kante des zugehörigen Fensters befinden soll. LockTop hat keine Auswirkung, solange LockBottom True ist.
<i>Top</i>	Integer	Der Abstand in Pixeln vom oberen Rand des Fensters zum oberen Rand des Separators.
<i>Width</i>	Integer	Die Breite des Separators.

Hinweis

Wenn die Breite größer als die Höhe ist, wird der Separator horizontal gezeichnet; ansonsten vertikal. In jedem Fall bestimmt der kleinere Wert die Dicke des Separator-Steuerelements, nicht die Dicke der sichtbaren Linien in der Mitte des Steuerelements. Mausbewegungs-Events werden über den gesamten Bereich des Separator-Steuerelements registriert, nicht nur über der sichtbaren Linie.

Folgende Abbildung zeigt einen sehr dicken Separator.



Siehe auch

[RectControl](#) Klasse.

Serial Steuerelement



Serial-Steuerelemente dienen zur Kommunikation über die serielle Schnittstelle.

Übergeordnete Klasse

Object

Das Serial-Steuerelement kann durch Code erzeugt werden, da es keine Unterklasse eines Control ist. Dies erlaubt es Ihnen, sehr einfach Code zu schreiben, der die Kommunikation übernimmt, ohne daß Sie das Steuerelement einem Fenster hinzuzufügen.

Eigenschaften

Name	Art	Beschreibung
<i>Baud</i>	<u>Integer</u>	Die Geschwindigkeit, mit der die Daten gesendet und empfangen werden (siehe unten).
<i>Bits</i>	<u>Integer</u>	0=5 Datenbits, 1=6 Datenbits, 2=7 Datenbits, 3=8 Datenbits
<i>CTS</i>	<u>Boolean</u>	Aktiviert CTS Flußkontrolle.
<i>DTR</i>	<u>Boolean</u>	Aktiviert DTR Flußkontrolle.
<i>Index</i>	<u>Integer</u>	Die Nummer des Steuerelements, falls es ein Teil eines Steuerelement-Arrays ist.
<i>Left</i>	<u>Integer</u>	Die linke Seite des Steuerelements in lokalen Koordinaten (relativ zum Fenster).
<i>MacInDriverRefNumber</i>	<u>Integer</u>	Macintosh In Driver Reference Number. Zur Verwendung für Toolbox-Programmierer.
<i>MacOutDriverRefNum</i>	<u>Integer</u>	Macintosh Out Driver Reference Number. Zur Verwendung für Toolbox-Programmierer.
<i>Name</i>	<u>String</u>	Der Name des Objektes.
<i>Parity</i>	<u>Integer</u>	0=keine Parität, 1=ungerade Parität, 2=gerade Parität
<i>Port</i>	<u>Integer</u>	Der zu benutzende Port. 0=Modemanschluß, 1=Druckeranschluß.
<i>SerialPort</i>	<u>SerialPort</u>	Wird verwendet, um den seriellen Port festzulegen, mit dem das Serial-Steuerelement kommunizieren soll.
<i>Stop</i>	<u>Integer</u>	0=1 Stopbits, 1=1.5 Stopbits, 2=2 Stopbits
<i>Super</i>	Object Class	Die Objektklasse, auf der das Objekt basiert.
<i>Top</i>	<u>Integer</u>	Die obere Seite des Steuerelements in lokalen Koordinaten (relativ zum Fenster)
<i>Win32DriverHandle</i>	<u>Integer</u>	Windows-Driver-Handle.
<i>XON</i>	<u>Boolean</u>	Aktiviert XON-Flußkontrolle.

Events

Name	Parameter	Beschreibung
<i>DataAvailable</i>		Es sind neue Daten eingetroffen, die nun im Puffer bereitliegen.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Schließt den seriellen Port.
<i>Flush</i>		Löscht alle Daten aus dem Puffer des seriellen Ports.
<i>LeaveDTROnClose</i>		Teilt dem Serial-Steuerelement mit, daß beim Schließen des Ports DTR eingeschaltet bleiben soll. Der serielle Port muß offen sein, damit diese Methode funktioniert.
<i>LookAhead</i>	[Encoding als <u>String</u>]	Liefert einen String. Liefert alle nicht ausgelesenen Zeichen aus dem Puffer, ohne sie aus dem Puffer zu entfernen. Der optionale Parameter <i>Encoding</i> erlaubt Ihnen das Text-Encoding der Daten festzulegen, die zurückgeliefert werden. Verwenden Sie das <u>Encodings</u> -Objekt, um ein Text-Encoding festzulegen.

Name	Parameter	Beschreibung
<i>Open</i>		Versucht den seriellen Port zu öffnen und liefert <u>True</u> zurück, wenn dies geklappt hat, <u>False</u> bei Mißerfolg.
<i>Poll</i>		Sorgt für die Aktualisierung der Eigenschaften des Steuerelements und löst ein Event <code>DataAvailable</code> aus, falls neue Daten vorliegen.
<i>Read</i>	Bytes als <u>Integer</u>	Liest <i>Bytes</i> aus dem Puffer und liefert sie als String zurück.
<i>ReadAll</i>		Liest alle zur Verfügung stehenden Daten aus dem Puffer und liefert sie als String zurück.
<i>Reset</i>		Setzt Baud- und Byte-Format des Ports zurück. Der serielle Port muß offen sein, damit diese Methode funktioniert.
<i>Write</i>	Data als <u>String</u>	Schickt an die serielle Schnittstelle asynchron Daten.
<i>XmitWait</i>		Wartet, bis alle mit der Write-Methode verschickten Daten an den seriellen Port geschickt werden konnten.

Baud Raten

Um die Baud-Rate zu setzen, weisen Sie die gewünschte Konstante der Baud-Eigenschaft zu. Um die Baud-Rate zu ermitteln, vergleichen Sie die Baud-Eigenschaft mit den Konstanten in dieser Tabelle.

Baud-Rate	Wert	Konstante	Baud-Rate	Wert	Konstante
300	0	Baud300	9600	8	Baud9600
600	1	Baud600	14400	9	Baud14400
1200	2	Baud1200	19200	10	Baud19200
1800	3	Baud1800	28800	11	Baud28800
2400	4	Baud2400	38400	12	Baud38400
3600	5	Baud3600	57600	13	Baud57600
4800	6	Baud4800	115200	14	Baud115200
7200	7	Baud7200	230400	15	Baud230400

Hinweis

Das Serial-Steuerelement kann verwendet werden, um mit mehreren seriellen Ports auf einmal zu kommunizieren. Sie können die Eigenschaften des System-Objekts verwenden, um die Anzahl der seriellen Ports auf dem Rechner zu ermitteln und ein Array dieser Ports zu erhalten.

Wenn während einer seriellen Verbindung Daten empfangen werden, wird automatisch der Event-Handler `DataAvailable` ausgeführt. In diesem Event-Handler können Sie die Funktionen `Read` oder `ReadAll` verwenden, um die Daten aus dem seriellen Puffer auszulesen. Diese Funktionen entfernen die Daten aus dem Puffer des seriellen Ports, sobald die Daten zurückgeliefert werden. Wenn Sie Daten aus dem Puffer auslesen wollen, ohne daß diese dort entfernt werden, sollten Sie die Eigenschaft `LookAhead` verwenden. Dieser Puffer verwendet soviel vom verfügbaren Speicher, wie benötigt wird, so daß er nicht in der Größe angepaßt werden muß.

Weil die Write-Methode asynchron arbeitet, kann es vorkommen, daß Sie die Methode `XmitWait` gebrauchen müssen, um `REALbasic` anzuweisen, solange zu warten, bis die Daten an den seriellen Port geschickt wurden.

Die seriellen Ports sind auch zur Verwendung unter Mac OS X verfügbar. Verwenden Sie die `Serialport`-Eigenschaft des `System`-Objekts, um wie gewohnt auf die serielle Schnittstelle zuzugreifen.

Das **Serial**-Steuerelement implementiert das „Readable“- und „Writable“-Klassen-Interface. Wenn Sie eines oder beide Interfaces implementieren, müssen Sie eine Read- und/oder Write-Methode vorhalten, die der folgenden Spezifikation entspricht.

Ergebnis=Read (count, [enc])

Teil	Typ	Beschreibung
Ergebnis	<u>String</u>	Der Text, der zurückgegeben wird.
count	<u>Integer</u>	Anzahl der Bytes, die gelesen werden sollen.
enc	<u>TextEncoding</u>	Optionaler Parameter, der die Kodierung des Textes angibt.

Das Readable Klassen-Interface hat zwei Methoden.

Name	Parameter	Beschreibung
EOF		Liefert <u>True</u> wenn das Ende des Streams erreicht ist. Liefert ebenfalls <u>True</u> , wenn der Stream geschlossen wurde und keine Daten im internen Puffer sind.
ReadError		Liefert <u>True</u> wenn ein Fehler aufgetreten ist.

Write(data)

Teil	Typ	Beschreibung
data	<u>String</u>	Der Text, der geschrieben werden soll.

Das Writeable Klassen-Interface hat folgende Methoden:

Name	Parameter	Beschreibung
Flush		Schreibt verbleibende Daten im internen Puffer auf das Volume und löscht den Puffer. Für Sockets und SerialStreams wird ein letztes Mal Poll aufgerufen und dann der Puffer geleert.
WriteError		Liefert <u>True</u> wenn ein Fehler aufgetreten ist.

Für mehr Informationen über Klassen-Interfaces und wie solche implementiert werden, lesen Sie das Kapitel „Klassen-Interfaces“ im Benutzerhandbuch durch.

Beispiel

Folgender Code weist ein Serial-Steuerelement an, unter Verwendung des Serial-Port Null (Modem-Port) auf einem Macintosh zu kommunizieren.

```
If TargetMacOS then // Programm läuft auf einem Mac
    serial.SerialPort=system.SerialPort(0)
end if
```

Siehe auch

SerialPort, Object Klassen; System Objekt.

SerialPort Klasse

Wird verwendet, um die Eigenschaften einer beliebigen seriellen Schnittstelle zu ermitteln.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Name</i>	<u>String</u>	Name der seriellen Schnittstelle. Auf einem Macintosh ist der Port Null der Modem-Port und Port 1 der Drucker-Port.
<i>InputDriverName</i>	<u>String</u>	Name des Eingabetreibers
<i>OutputDriverName</i>	<u>String</u>	Name des Ausgabebetreibers
<i>RatedSpeed</i>	<u>Integer</u>	Geschätzte Geschwindigkeit (Baud)
<i>MaximumSpeed</i>	<u>Integer</u>	Maximale Geschwindigkeit (Baud)

Beispiel

Siehe Beispiel für das System-Objekt. Sie bekommen die Werte der SerialPort-Eigenschaften, indem Sie zuerst ein System-Objekt erzeugen.

Siehe auch

System Objekt.

ServerSocket Klasse

Wird verwendet, um mehrfache Verbindungen auf dem gleichen Port zu unterstützen. **ServerSocket** ist nur in der Professional-Version von REALbasic verfügbar.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>IsListening</i>	<u>Boolean</u>	<u>True</u> , wenn das ServerSocket auf eingehende Verbindungen wartet.
LocalAddress	<u>String</u>	Die lokale Adresse, die das ServerSocket verwendet.
<i>MaximumSocketsConnected</i>	<u>Integer</u>	Die Maximalzahl an Verbindungen die ServerSocket erlaubt. Nachdem ein Socket die Verbindung von ServerSocket beendet, erlaubt der Server einer weitere Verbindung.
<i>MinimumSocketsAvailable</i>	<u>Integer</u>	Die kleinste Anzahl von verfügbaren Sockets, die im Socket-Verbindungen-Pool des Servers verfügbar ist. Falls ServerSocket unter diese Zahl fällt, ruft es das AddSocket-Event auf, um den Vorrat an Sockets aufzufüllen.
<i>Port</i>	<u>Integer</u>	Der Port, an dem ServerSocket lauscht.

Events

Name	Parameter	Beschreibung
<i>AddSocket</i>		Der ServerSocket fordert, daß Sie seinem internen Pool ein Socket hinzufügen. Es wird aufgerufen, wenn der ServerSocket anfangs zu lauschen beginnt. Anfangs wird es <i>MaximumSocketsConnected</i> -mal aufgerufen, um den internen Pool an Sockets aufzufüllen. Dieser Socket muß resident bleiben (darf keine lokale Variable sein) und muß nicht- <u>Nil</u> sein. Liefert ein <u>TCPSocket</u> . Da UDP ein verbindungsloses Protokoll ist, ergibt es für ein ServerSocket keinen Sinn, sich mit UDPSockets zu beschäftigen.
<i>Error</i>	ErrorCode als <u>Integer</u>	Tritt auf, wenn der Socket einen Fehler hat. Es werden Standard-Fehlermeldungen zurückgegeben. Auf dem Macintosh sind diese in MacErrors.h dokumentiert, Windows-Fehlercodes in WinSock.h.

Methoden

Name	Parameter	Beschreibung
<i>Listen</i>		Beginnt den Lausch-Prozeß für den ServerSocket mit den von Ihnen angegebenen Eigenschaften.
<i>StopListening</i>		Beendet das Lauschen auf dem ServerSocket . Bricht keine bestehenden Verbindungen ab.

Hinweise

Ein **ServerSocket** ist ein permanenter Socket, der auf einem einzelnen Port nach mehrfachen Verbindungen lauscht. Wenn ein Verbindungsversuch auf diesem Port stattfindet, gibt der **ServerSocket** die Verbindung an einen anderen Socket weiter und lauscht weiterhin auf dem gleichen Port. Ohne den **ServerSocket** ist es schwer, diese Funktionalität einzubauen, da es Wartezeiten gibt zwischen den eingehenden Verbindungen, dem Weitergeben, dem Erzeugen eines neuen lauschenden Sockets und dem Wiederstarten des Lausch-Prozesses. Wenn es zwei Verbindungen gibt, die etwa zur gleichen Zeit kommen, kann es passieren, daß eine fallengelassen wird, da auf diesem Port kein weiterer lauschender Socket verfügbar war.

Sie können die Eigenschaften *MinimumSocketsAvailable* und *MaximumSocketsConnected* ändern, nachdem der lauschende Socket eingerichtet wurde. Wenn Sie die *MaximumSocketsConnected*-Eigenschaft ändern, wird keine bestehende Verbindung zerstört (es kann nur sein, daß es keine weiteren Verbindungen erlaubt, bis die vorhandenen Verbindungen gelöst wurden). Wenn Sie die *MinimumSocketsAvailable*-Eigenschaft ändern, kann es sein, daß es das AddSocket-Event auslöst, um seinen internen Puffer aufzufüllen.

Unter Mac OS X kann ein **ServerSocket** nicht auf Ports unterhalb von 1024 zugreifen, es sei denn, der Anwender hat Root-Rechte. Das ist Teil des Designs von Mac OS X.

Der **ServerSocket** und SSLSocket sind nur in der Professional-Version von REALbasic verfügbar.

Siehe auch

UDPSocket, SSLSocket, SocketCore Klassen.

ServiceApplication Klasse

Wird verwendet, um eine Service-Applikation zu erstellen, die als Hintergrundprozeß läuft und keine Benutzerinteraktion benötigt.

Übergeordnete Klasse

ConsoleApplication

Events

Name	Parameter	Beschreibung
Continue		Tritt auf, wenn der Benutzer „Fortführen“ im Service-Control-Manager unter Windows klickt, unter der Annahme, daß ein Pause-Event auftrat. Der Continue-Event tritt nicht auf, bevor nicht ein Pause-Event stattfand. Der Benutzer erwartet, daß die Applikation fortgeführt wird, wenn dieser Event auftrat.
Pause		Tritt auf, wenn der Benutzer „Pause“ im Service-Control-Manager unter Windows klickt. Wenn der Benutzer „Pause“ anwählt, erwartet er, daß der Dienst alle Aktivitäten einstellt, bis ein „Continue“-Event auftrat.
Stop	ShuttingDown als <u>Boolean</u>	Tritt in zwei Situationen auf: Wenn der Benutzer Stop im Service-Control-Manager unter Windows anwählt (in dem Fall ist ShuttingDown <u>False</u>) oder der Computer heruntergefahren wird. In dem Fall ist ShuttingDown <u>True</u> . Und der Event tritt auf, wenn die <u>Quit</u> -Methode aufgerufen wurde oder der Dienst beendet wurde (alle Plattformen). Der ShuttingDown-Parameter wird nur unter Windows unterstützt.

Hinweise

Eine Service-Applikation ist eine spezielle Art von Konsolen-Applikation, die kein Benutzerinterface benötigt und auch läuft, wenn kein Benutzer eingeloggt ist. Ein typisches Beispiel wäre ein FTP- oder HTTP-Server.

Konsolen-Applikationen stützen sich auf den Print- und Input-Befehl, um mit dem Benutzer zu kommunizieren. Service-Applikation verhalten sich ähnlich (abhängig, wie diese installiert sind und welche Plattform verwendet wird). Wenn Ihre Service-Applikation zum Beispiel unter Mac OS X oder Linux unter der Verwendung von inetd laufen, so entsprechen Print und Input den Sockets, die das System für Sie bereit stellt. Das bedeutet, daß die Write-Methode der StandardOutputStream-Klasse und die Read-Methode der StandardInputStream-Klasse sich wie Sockets verhalten.

Dieses Verhalten ist nicht für alle Plattformen garantiert. Wenn Sie Ihre Service-Applikation für mehrere Plattformen entwickeln wollen, verwenden Sie am besten direkt einen TCPSocket.

Um eine Service-Applikation zu erzeugen, müssen Sie erst eine reguläre Konsolen-Applikation erstellen. Wählen Sie den Menüpunkt „Ablage/Neu...“ und dort „Konsolen-Applikation“. Eine Konsolen-Applikation besitzt ein einzelnes Objekt, dessen übergeordnete Klasse ConsoleApplication ist. Ändern Sie nun im Eigenschaften-Fenster die Super-Eigenschaft des App-Objekts in **ServiceApplication**.

Die **ServiceApplication**-Klasse ist eine abgeleitete Klasse der ConsoleApplication-Klasse, deshalb sind alle Events und Methoden der ConsoleApplication-Klasse verfügbar. Zusätzlich gibt es die drei Events der **ServiceApplication**-Klasse. Einige der Events werden nur auf bestimmten Plattformen ausgelöst, die diese unterstützen. Die einzige Plattform, die diese momentan unterstützt, ist Windows.

Siehe auch

ConsoleApplication, StandardInputStream, StandardOutputStream Klassen; Input, Print, StdErr, StdIn, StdOut Methoden; TargetHasGUI Konstante.

Shell Klasse

Wird verwendet, um Unix- oder DOS-Shell-Befehle unter Mac OS X oder Windows auszuführen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>ErrorCode</i>	<u>Integer</u>	Liefert 0, wenn die Execute-Methode erfolgreich ausgeführt wurde. Andernfalls wird ein Fehlercode vom System geliefert. Unter Windows wird -1 geliefert, wenn die Ausführung mißlang.
<i>IsRunning</i>	<u>Boolean</u>	True, wenn ein asynchroner oder interaktiver Shell-Prozeß läuft.
<i>Mode</i>	<u>Integer</u>	Steuert die Arbeitsweise der Shell. Asynchroner und Interaktiver Modus werden nur unter Mac OS X unterstützt. 0=Synchron. Die Shell führt ihren Befehl aus und liefert das Ergebnis. 1=Asynchron. Die Shell führt ihre Befehl aus und liefert Daten über das DataAvailable-Event. 2=Interaktiv. Daten können mit der Write-Methode an eine laufende Shell-Session gesendet werden und Daten werden über das DataAvailable-Event zurückgeliefert.
<i>PID</i>	<u>Integer</u>	ID des interaktiven Shell-Prozesses.
<i>Result</i>	<u>String</u>	Enthält Ergebnisse des Ausgabe-Puffers, wenn Sie sich im Synchronen Modus befinden (Modus 0), ohne den Puffer zu leeren.
<i>TimeOut</i>	<u>Integer</u>	(nur Windows) Zeit (Millisekunden), die festlegt, wie lange ein Prozeß laufen darf, bevor er automatisch terminiert wird. Voreinstellung ist 2000, bzw. 2 Sekunden.

Events

Name	Parameter	Beschreibung
<i>Completed</i>		Wird nur im Modus 1 und 2 ausgelöst. Wird ausgelöst, wenn der ausgeführte Befehl beendet ist.
<i>DataAvailable</i>		Wird nur im Modus 1 und 2 ausgelöst. Wird ausgelöst, wenn das Shell-Objekt Daten zurückliefert – so wie bei der Socket-Klasse.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Beendet den laufenden Befehl.
<i>Execute</i>	Command als <u>String</u>	Führt einen einzeiligen Unix- oder DOS-Shell-Befehl aus. Wird im Synchronen Modus verwendet. Wenn Sie sich im Synchronen Modus (Modus 0) befinden, enthält die Result-Eigenschaft die Ergebnisse.
<i>Poll</i>		Schaut nach zurückgelieferten Daten von der laufenden Shell und löst ggfs. das DataAvailable-Event aus.
<i>ReadAll</i>		Liefert den Inhalt des Ausgabe-Puffers und leert diesen.
<i>Write</i>	Text als <u>String</u>	Sendet einen String an den Eingabe-Puffer der Shell.
<i>WriteLine</i>	Text als <u>String</u>	Sendet ein String-Ende in einem Zeilenvorschub an den Eingabe-Puffer der Shell.

Hinweise

Verwenden Sie die Shell-Klasse, um DOS- oder Unix-Befehle auszuführen und die Ergebnisse zu erhalten. Sie hat eine einzige Methode namens Execute, die ein einzeliges Kommando ausführt. Dies führt dazu, daß sich zwei Eigenschaften des Shell-Objekts ändern: ErrorCode, welches ein vom System gelieferter Fehlercode oder 0 für keinen Fehler ist; und „Result“, welches ein String ist, der die Ausgabe des Befehls enthält. Die TimeOut-Eigenschaft legt fest, wie lange (in Millisekunden) ein Prozeß laufen darf, bis er automatisch terminiert wird. Ein Wert von -1 bedeutet, daß der Prozeß endlos laufen darf. Diese Eigenschaft betrifft derzeit nur Windows (unter Mac OS X nicht benötigt).

Nur unter Mac OS X kann die Shell-Klasse Befehle im asynchronen oder interaktiven Modus ausführen.

Beispiele

Folgender Code listet unter Verwendung des synchronen Modus die Dateien des aktuellen Verzeichnisses auf (Mac OS X oder Linux):

```
Dim s As Shell
s=New Shell
If TargetCarbon then
    s.execute "ls -la"
    If s.errorCode=0 then
        EditField1.text=s.result
    else
        MsgBox "Fehler-Code: "+Str(s.errorCode)
    end if
end if
```

Folgende Terminal-Applikation erlaubt Ihnen, Unix-Befehle unter Verwendung des interaktiven Modus zu übermitteln. Das Interface besteht aus zwei EditFields, InputField, in die der Anwender einen Befehl eingeben kann und OutputField, das das Ergebnis zeigt.

Das Open-Event für das Fenster initialisiert das Shell-Objekt (deklariert als eine Eigenschaft des Fensters).

```
mShell = New Shell
mShell.Mode = 2
```

Der Anwender kann Unix-Befehle in das EditField eingeben. Wenn er Return drückt, läuft der folgende Code im Keydown-Event des EditFields. Die Write-Methode sendet den Befehl in den Eingabepuffer der Shell.

```
If Key = Chr(13) Then
    If Not mShell.IsRunning Then
        mShell.Execute "sh"
    End If
    mShell.Write InputField.Text
    mShell.Write Chr(13)
    InputField.Text = ""
    Return True
Else
    Return False
End If
```

Ein **Timer** ruft die ReadAll-Methode in deren Action-Event auf:

```
If mShell <> Nil Then
    OutputField.SetText = mShell.ReadAll
End If
```

Siehe auch

[TargetCarbon](#), [TargetWin32](#), [TargetPPC](#), [TargetMacOS](#), [TargetLinux](#) Konstanten.

ShellNotRunningException Fehler

Sie haben versucht, einer Shell einen String zu übergeben, die nicht läuft.

Übergeordnete Klasse

[RuntimeException](#)

Hinweise

Der ShellNotRunningException Fehler tritt auf, wenn Sie versuchen, einen String an die Shell-Klasse zur Ausführung zu übergeben, diese aber nicht läuft. Das kann passieren, wenn Sie die Shell interaktiv benutzen und sie vor der Übergabe des Strings geschlossen wurde.

Siehe auch

Shell-Klasse.

ShowURL Methode

Verwendet die passende Anwendung, um die angegebene URL zu öffnen. Die passende Anwendung wird durch die Einstellungen im Kontrollfeld „Internet“ bzw. im Programm „Internet Config“ definiert.

Syntax

ShowURL URL

Teil	Art	Beschreibung
URL	<u>String</u>	Die zu öffnende URL.

Hinweis

URL steht für Uniform Resource Locator. URLs sind Internetadressen wie FTP-Server und Websites. Die Methode **ShowURL** verwendet die Anwendereinstellungen des Kontrollfelds „Internet“ bzw. des Programms „Internet Config“ um festzustellen, welches Programm für welche Art von URL verwendet wird. Wenn Sie beispielsweise eine URL mit „http“ angeben, wird der Webbrowser des Anwenders gestartet.

Beispiele

Dieses Beispiel startet den Webbrowser des Anwenders und öffnet danach unsere Website.

```
ShowURL "http://application-systems.de/"
```

Dieses Beispiel öffnet den FTP-Client des Anwenders, öffnet den FTP-Server von REAL Software und geht in das Verzeichnis mit den aktuellen Releases.

```
ShowURL "ftp://ftp.realsoftware.com/current_release"
```

Folgendes Beispiel startet das vom Anwender eingestellte E-Mail-Programm und erzeugt eine neue E-Mail-Nachricht, mit der Empfänger-Adresse im „An“-Bereich:

```
ShowURL "mailto:batman@batcave.org"
```

Shuffle Methode

Sortiert ein Array in zufälliger Reihenfolge

Syntax

array.Shuffle

Teil	Typ	Beschreibung
array	Beliebiger Datentyp	Das Array, das neu sortiert werden soll.

Hinweise

Shuffle funktioniert mit Arrays beliebigen Typs und Dimensionen. Es verwendet eine zufällige Zahl als Grundlage. Ein Element im ursprünglichen Array kann in jeder Zelle des Arrays platziert werden, wobei die Chancen für alle Zellen gleich sind.

Beispiel

Das folgende Beispiel mischt ein eindimensionales Array und zeigt das Resultat in einer **ListBox** an.

```
Dim aTest() as Integer
Dim i as Integer
aTest=Array(0,1,2,3,4,5,6,7,8,9)
aTest.Shuffle
For i = 0 to Ubound(aTest)
    ListBox1.Addrow Str(aTest(i))
Next
```

Hier wird ein zweidimensionales Array gemischt.

```
Dim myArray(5,5) as String
Dim i, j as Integer
For i=0 to 5
    For j=0 to 5
        myArray(i,j)=Str(i)+Str(j)
    Next
Next
myArray.Shuffle
```

Siehe auch

Dim Befehl; Array; Split; Join; Ubound Funktionen; Append; Insert; Redim; Remove; Pop; IndexOf; Insert; Sort Methoden; ParamArray Schlüsselwort.

ShutDownItemsFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Ordner „Ausschaltobjekte“ im Mac OS 9-Systemordner zu bekommen.

Syntax

Ergebnis=ShutDownItemsFolder

Teil	Art	Beschreibung
Ergebnis	FolderItem	Ein FolderItem, das den Ordner repräsentiert.

Hinweis

Verwenden Sie die ShutDownItemsFolder-Funktion immer dann, wenn Sie auf Objekte aus dem Ausschaltobjekte-Ordner zugreifen müssen.

Die Funktion ShutDownItemsFolder bietet eine Möglichkeit, auf die Einträge im Ordner „Ausschaltobjekte“ zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

ShutDownItemsFolder liefert unter Windows und Linux Nil.

Beispiel

Dieses Beispiel führt in einer Listbox alle Objekte des Ordner Ausschaltobjekte auf.

```
Dim i as integer
Dim f as FolderItem
f=ShutDownItemsFolder
For i=1 to f.count
  ListBox1.addrow f.item(i).name
Next
```

Siehe auch

ControlPanelsFolder, DesktopFolder, ExtensionsFolder, FontsFolder, PreferencesFolder, StartupItemsFolder, SystemFolder, TemporaryFolder, TrashFolder Funktionen.

Sie haben einen Operator verwendet, der nicht zu den spezifizierten Datentypen kompatibel ist Fehler

Sie haben versucht, mit Datentypen eine Operation durchzuführen, die diese nicht unterstützen. Zum Beispiel Strings zu multiplizieren, Booleans zu dividieren, etc.

Beispiele

```
Dim c as Integer
Dim d,e as String
Dim g,h as Boolean
d="Ted"
e="Alice"
c=d*e //Fehler
c=g/h //Fehler
```

Sie können den New-Operator nicht mit dieser Klasse verwenden Fehler

Sie haben versucht, eine Instanz von einer abstrakten Klasse zu erzeugen. Sie können den New-Operator nur verwenden, um eine neue Instanz eines in einem Fenster bereits vorhandenen Steuerelements, einem dem Projekt über **Ablage/Neues Fenster** hinzugefügten Fenster oder einem vorhandenen Menüeintrag zu erzeugen. Das vorhandene Objekt fungiert als Vorlage für die neue Instanz.

Sie können ebenso keine Instanz einer Klasse erzeugen, die nur als Basis-Klasse für andere Klassen verwendet wird, wie z.B. Socket-Core.

Sie können diesen Variablen- oder Eigenschaften-Namen nicht verwenden, da es ein reserviertes Wort ist

Beispiel

Dies sollte man nicht machen:

```
Dim w as Window  
w=New Window
```

Verwenden Sie stattdessen den New-Operator mit einer Instanz der Window-Klasse, die über den Befehl **Ablage/Neues Fenster** erzeugt wurde.

```
Dim w as FindWindow  
w=New FindWindow
```

Siehe auch

New Operator.

Sie können diesen Variablen- oder Eigenschaften-Namen nicht verwenden, da es ein reserviertes Wort ist Fehler

Reservierte Wörter können nicht als Namen für selbstdefinierte Objekte verwendet werden. Wenn Sie ein reserviertes Wort als Variablennamen in einem Dim-Befehl oder als Eigenschaften-Namen verwendet haben, erscheint diese Meldung.

Beispiele

Die Verwendung eines reservierten Worts, wie z.B. ein Datentyp (Integer, Double, Boolean, etc.) als Name einer Eigenschaft in der Dialogbox **Neue Eigenschaft**.

Sie können ein Array nicht an eine externe Funktion übergeben Fehler

Sie erhalten diesen Fehler bei einem Declare-Befehl, wenn Sie einen der Parameter als ein Array definieren.

Siehe auch

Declare Befehl.

Sie können ein Array nicht einem anderen Array zuweisen Fehler

Sie haben versucht, ein Array einem anderen zuzuweisen. Um ein Array zu erzeugen, das die gleichen Elemente wie ein anderes hat, müssen Sie jeden Wert des einen Arrays dem gleichen Index des anderen Arrays zuweisen. Verwenden Sie dazu den For-Befehl und gehen Sie in einer Schleife alle Werte des Array-Index durch.

Beispiel

```
Dim a(5) as Integer  
Dim b(5) as Integer  
a=b
```

Siehe auch

Dim Befehl.

Sie können einen Ausdruck nicht als Parameter, der als ByRef definiert ist, übergeben Fehler

Sie haben in einem Funktionsaufruf versucht, einen Ausdruck an Stelle einer Variablen oder Eigenschaft einem Parameter zu übergeben, der als ByRef in der aufgerufenen Methode deklariert war.

Beispiel

Folgende Methode bekommt einen Parameter, der als ByRef deklariert wurde:

```
Sub Squarit(ByRef c as Double)  
c=c*c  
End Sub
```

Die aufrufende Funktion ist:

378 Sie können keinen Wert zurückgeben, da für diese Methode kein Rückgabetyt definiert wurde Fehler

```
Dim a,b as Double  
a=5  
b=10  
Squarit(a*b) //hier kann man keinen Ausdruck verwenden
```

Siehe auch

ByRef Operator.

Sie können keinen Wert zurückgeben, da für diese Methode kein Rückgabetyt definiert wurde Fehler

Sie können den Return-Befehl nicht in einer Methode verwenden, so lange Sie nicht in der Funktions-Deklaration einen Datentyp für den Wert definieren, der zurückgeliefert wird.

Beispiel

```
Sub myMethod(a as Integer, b as Integer)  
    Return a*b  
End Sub
```

Um einen Wert zurückzuliefern, müssen Sie den Datentyp des zurückgelieferten Werts deklarieren. In diesem Fall in diesem Beispiel würde der Deklarations-Befehl folgendermaßen aussehen:

```
Function myMethod(a as Integer, b as Integer) as Integer
```

Siehe auch

Sub, Function Befehle.

Sie können nicht ein nichtexistierendes Event implementieren Fehler

Tritt nur in RBScript auf. Sie haben versucht, ein Event zu verwenden, das in REALbasic nicht definiert ist.

Siehe auch

RBScript Steuerelement.

Sie können Super nicht verwenden, da diese Klasse keine übergeordnete Klasse besitzt Fehler

Sie können nicht auf die übergeordnete Klasse einer Klasse verweisen, die keine solche besitzt.

Sie müssen den Datentyp des Rückgabewertes angeben Fehler

Sie haben eine externe Funktion unter Verwendung des Function-Schlüsselwortes deklariert, aber keinen Rückgabewert festgelegt. Dies kann passieren, wenn man den Datentyp des Rückgabewertes vergißt anzugeben oder weil man eigentlich vorhatte ein **Sub** zu deklarieren aber **Function** stattdessen verwendete.

Siehe auch

Function, Sub Befehle.

Sign Funktion

Liefert das Vorzeichen der übergebenen Zahl.

SyntaxErgebnis=**Sign**(Wert)

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Das Vorzeichen von <i>Wert</i> . Liefert - 1, wenn <i>Wert</i> negativ ist, 0, wenn <i>Wert</i> Null ist und 1, wenn <i>Wert</i> positiv ist.
<i>Wert</i>	<u>Double</u>	Die Zahl, die der Funktion übergeben wird.

Beispiel

Folgendes Beispiel ermittelt das Vorzeichen der übergebenen Zahl.

```
Dim d as Double
Dim result as Integer
If EditField1.text <> "" then
    result=Sign(Val(EditField1.text))
    MsgBox Str(result)
Else
    MsgBox "Bitte eine Zahl eingeben!"
End if
```

Siehe auch

Abs, Val Funktionen.

Sin Funktion

Gibt den Sinus des übergebenen Wertes zurück.

SyntaxErgebnis=**Sin**(Wert)

Teil	Art	Beschreibung
<i>Wert</i>	<u>Double</u>	Der Wert, dessen Sinus Sie ermitteln wollen.
<i>Ergebnis</i>	<u>Double</u>	Der Sinus von <i>Wert</i> .

Hinweis

Die **Sin**-Funktion gibt den Winkel (im Bogenmaß) des übergebenen Wertes zurück. Wenn der Wert im Gradmaß vorliegt, multiplizieren Sie ihn mit Pi/180, um ihn ins Bogenmaß umzurechnen.

Beispiele

Folgendes Beispiel verwendet die **Sin**-Funktion, um den Sinus einer Zahl zu ermitteln.

```
Dim d As Double
d=Sin(0.5) //liefert 0.4794255
```

Siehe auch

Asin Funktion.

Single Datentyp

Ein **Single** ist ein wesentlicher Datentyp in REALbasic. Ein **Single** ist eine Zahl, die einen Dezimalwert besitzen kann, es ist eine reelle Zahl. Sie kann Werte zwischen -1.175494 e-38 und 3.402823 e+38 annehmen. In anderen Sprachen würde das **Single** von REALbasic als Realzahl mit einfacher Präzision bezeichnet. Da **Single** Zahlen sind, können Sie mathematische Berechnungen mit diesen durchführen. **Singles** belegen 4 Byte Speicher.

Beispiel

```
Dim s as Single
d=3.1416
```

Siehe auch

Integer, Double, String, Boolean, Color Datentypen; +, -, *, /, <, <=, =, >=, >, <>, \, Mod, Val, Str, VarType, IsNumeric Funktionen; Dim Befehl.

Slider Steuerelement



Implementiert das Slider-Steuerelement auf Macintosh-Rechnern mit Mac OS 8.x oder neuer, Windows und Linux.

Übergeordnete Klasse

RectControl

Da dies ein RectControl ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle RectControl-Objekte gemeinsam gelten, unter RectControl nach.

Eigenschaften

Name	Typ	Beschreibung
LineStep	<u>Integer</u>	Unter Windows bestimmt diese Eigenschaft die Schrittweite, mit der der Slider sich bewegt, wenn die Pfeil-Tasten benutzt werden, während der Slider den Fokus hat.
LiveScroll	<u>Boolean</u>	Wenn <u>True</u> , tritt ein ValueChanged-Event auf, wenn der Benutzer den Slider zieht. Andernfalls tritt das Event nur einmal auf, wenn der Benutzer den Slider los lässt.
Maximum	<u>Integer</u>	Der Maximalwert des Sliders. Voreingestellt ist 100.
Minimum	<u>Integer</u>	Der Minimalwert des Sliders.
PageStep	<u>Integer</u>	Die Schrittweite, wenn der Benutzer in den Balken des Sliders klickt.
Value	<u>Integer</u>	Aktueller Wert des Sliders.

Events

Name	Beschreibung
<i>GotFocus</i>	(Nur Win32) Das Steuerelement hat den Fokus bekommen und besitzt einen Selektionsrahmen um die Beschriftung.
<i>IsConnected</i>	Liefert ein Boolean. Liefert True, wenn der Socket eine bestehende Verbindung besitzt (nur Win32).
<i>LostFocus</i>	(Nur Win32) Das Steuerelement hat den Fokus verloren.
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u> Der Mausknopf wurde innerhalb des Sliders an der Position x,y gedrückt. Gibt ein Boolean zurück. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen. Die Koordinaten x und y sind lokal zum Steuerelement. Der Punkt 0,0 ist die linke obere Ecke des gesamten Steuerelements (links von der Beschriftung und über der Box, die gezeichnet wurde).
<i>MouseDown</i>	x als <u>Integer</u> , y als <u>Integer</u> Der Mausknopf wurde innerhalb des Sliders gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als <u>Integer</u> , y als <u>Integer</u> Der Mausknopf wurde innerhalb des Sliders an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>ValueChanged</i>	Der Wert des Sliders hat sich geändert.

Beispiele

Ändern des Maximalwerts eines **Sliders** zur Laufzeit:

```
Slider1.maximum=200
```

Setzt den Text von staticText1 auf den Wert des Sliders, während der Benutzer scrollt:

```
Sub ValueChanged()  
    staticText1.text=Str(Slider1.Wert)
```


End Sub

Siehe auch

RectControl Klasse; Scrollbar Steuerelement.

SMTPSecureSocket Klasse

Wird zum Senden von sicherer E-Mail über das SMTP-Protokoll unter Verwendung von SSL- oder TLS-Verschlüsselung verwendet.

Übergeordnete Klasse

SSLSocket

Eigenschaften

Name	Typ	Beschreibung
<i>Messages</i>	EmailMessage	Die Warteschlange aus Mitteilungen, die darauf wartet, verschickt zu werden. Dies ist ein Array, in dem das 0 ^{te} Element das ist, welches als nächstes verschickt wird. Wenn eine Mitteilung verschickt wurde, wird sie aus der Warteschlange entfernt und an das MessageSent-Event übergeben. Um an die Warteschlange eine Mitteilung anzuhängen, müssen Sie die <u>Append</u> -Methode verwenden. Um eine Mitteilung zu entfernen, müssen Sie die <u>Remove</u> -Methode verwenden.

Methoden

Name	Parameter	Beschreibung
<i>DeleteAllMessages</i>		Löscht alle Mitteilungen aus der Sende-Warteschlange.
<i>DisconnectFromServer</i>		Schließt die Verbindung mit dem Mail-Server.
<i>SendMail</i>		Stellt eine Verbindung mit dem Mail-Server her und sendet die Mitteilung in der Warteschlange.

Events

Name	Parameter	Beschreibung
<i>ConnectionEstablished</i>	Greeting als <u>String</u>	Wird ausgeführt, wenn die Verbindung mit dem Mail-Server hergestellt wurde und übergibt den Gruß-String, den der Server lieferte.
<i>Error</i>		Ein Fehler ist aufgetreten.
<i>MailSent</i>		Wird ausgeführt, wenn alle Mitteilungen in der Warteschlange gesendet wurden.
<i>MessageSent</i>	Email als <u>EmailMessage</u>	Wird ausgeführt, wenn eine Mitteilung gesendet wurde; <i>Email</i> enthält die Mitteilung, die gesendet wurde.
<i>ServerError</i>	ErrorID als <u>Integer</u> , ErrorMessage als <u>String</u> , Email als <u>EmailMessage</u>	Wird ausgeführt, wenn ein Fehler beim Senden einer Mitteilung auftrat. <i>ErrorID</i> und <i>ErrorMessage</i> wurden vom Mail-Server gesendet und <i>Email</i> ist die Mitteilung, die gesendet wurde, als der Fehler auftrat. <i>Email</i> wird zu der Zeit aus der Warteschlange entfernt, zu der das Event ausgeführt wird.

Hinweise

Die **SMTPSecureSocket**-Klasse ist die gleiche wie die SMTPSocket-Klasse, mit Ausnahme, daß diese von SSLSocket an Stelle von TCPSocket abgeleitet ist. Als Folge davon können Sie die Secure-Eigenschaft der SSLSocket-Klasse verwenden, um gesicherte Kommunikation zu unterstützen.

Siehe auch

POP3Socket, POP3SecureSocket, EmailMessage, TCPSocket, HTTPSocket, HTTPSecureSocket, SMTPSocket, SocketCore Klassen.

SMTPSocket Klasse

Wird zum Senden von E-Mails über das SMTP-Protokoll verwendet.

Übergeordnete Klasse

TCPSocket

Eigenschaften

Name	Typ	Beschreibung
<i>Messages</i>	EmailMessage	Die Warteschlange aus Mitteilungen, die darauf wartet, verschickt zu werden. Dies ist ein Array, in dem das 0 ^{te} Element das ist, welches als nächstes verschickt wird. Wenn eine Mitteilung verschickt wurde, wird sie aus der Warteschlange entfernt und an das MessageSent-Event übergeben. Um an die Warteschlange eine Mitteilung anzuhängen, müssen Sie die <u>Append</u> -Methode verwenden. Um eine Mitteilung zu entfernen, müssen Sie die <u>Remove</u> -Methode verwenden.

Methoden

Name	Parameter	Beschreibung
<i>DeleteAllMessages</i>		Löscht alle Mitteilungen aus der Sende-Warteschlange.
<i>DisconnectFromServer</i>		Schließt die Verbindung mit dem Mail-Server.
<i>SendMail</i>		Stellt eine Verbindung mit dem Mail-Server her und sendet die Mitteilung in der Warteschlange.

Events

Name	Parameter	Beschreibung
<i>ConnectionEstablished</i>	Greeting als <u>String</u>	Wird ausgeführt wenn die Verbindung mit dem Mail-Server hergestellt wurde und übergibt den Gruß-String, den der Server lieferte.
<i>Error</i>		Ein Fehler ist aufgetreten.
<i>MailSent</i>		Wird ausgeführt, wenn alle Mitteilungen in der Warteschlange gesendet wurden.
<i>MessageSent</i>	Email als <u>EmailMessage</u>	Wird ausgeführt, wenn eine Mitteilung gesendet wurde; <i>Email</i> enthält die Mitteilung, die gesendet wurde.
<i>ServerError</i>	ErrorID als <u>Integer</u> , ErrorMessage als <u>String</u> , Email als <u>EmailMessage</u>	Wird ausgeführt, wenn ein Fehler beim Senden einer Mitteilung auftrat. <i>ErrorID</i> und <i>ErrorMessage</i> wurden vom Mail-Server gesendet und <i>Email</i> ist die Mitteilung, die gesendet wurde, als der Fehler auftrat. <i>Email</i> wird zu der Zeit aus der Warteschlange entfernt, zu der das Event ausgeführt wird.

Hinweise

Die Klassen POP3Socket und **SMTPSocket** werden zusammen verwendet, um die Basis eines E-Mail-Programms zu bilden. POP3 ist das Standard-Internet-Protokoll für das Empfangen von Mitteilungen und SMTP (Simple Mail Transfer Protocol) ist das Standard-Internet-Protokoll für das Versenden von E-Mails.

SMTPSocket unterstützt die Autorisation über „login“ und „plain“. Dies erhöht die Kompatibilität mit einigen Servern.

Beispiel

Folgender Code im Action-Event eines PushButtons sendet eine E-Mail-Nachricht. Es arbeitet mit einem Formular mit Feldern für jeden Teil der E-Mail. EditFields werden für Anwendername und Paßwort des E-Mail-Accounts, die Von-, An- und CC-Adressen, den Betreff der E-Mail, den Haupttext im normalen Text und im HTML-Format verwendet. Da der Anwender mehrere An- und CC-Adressen eingeben kann, arbeitet das Beispiel diese Strings ab und fügt die individuellen An- und CC-Adressen mit den AddRecipient- und AddCCRecipient-Methoden der EmailMessage hinzu.

Ein Anhang kann im Formular als ein vollständiger Pfad auf eine Datei festgelegt werden. Wenn der Anhang existiert, wird er der Attachments-Eigenschaft der EmailMessage hinzugefügt.

Ein **SMTPSocket** namens SMTPSocket1 wurde dem Fenster hinzugefügt.

```
Dim mail as EmailMessage
Dim file as EmailAttachment
Dim i as Integer
Dim s as String
```

```

SMTPSocket1.address = "mail.mySMTPServer.com"
SMTPSocket1.port = 25

SMTPSocket1.username = usernameFld.text //Anwendername aus dem Editfield ermitteln
SMTPSocket1.password = passwordFld.text //Paßwort aus dem Editfield ermitteln

mail = New EmailMessage
mail.fromAddress=fromAddressFld.text //Von-Adresse aus dem Editfield ermitteln
mail.subject=subjectFld.text //Betreff aus dem Editfield ermitteln
mail.bodyPlainText = bodyFld.text //Haupttext aus dem Editfield ermitteln
mail.bodyHTML = htmlFld.text //Haupttext als HTML aus dem Editfield ermitteln
mail.headers.appendHeader "X-Mailer","REALbasic SMTP Demo"

//mehrere An-Adressen können durch Kommas getrennt werden
//jede An-Adresse wird dem Empfänger-Array hinzugefügt
s = ReplaceAll(toAddressFld.text,",",Chr(13))
For i = 1 to CountFields(s,Chr(13))
  mail.addRecipient Trim(NthField(s,Chr(13),i))
next

//mehrere CC-Adressen können durch Kommas getrennt werden
//jede An-Adresse wird dem CC-Empfänger-Array hinzugefügt
s = ReplaceAll(ccAddressFld.text,",",Chr(13))
For i = 1 to CountFields(s,Chr(13))
  mail.addCCRecipient Trim(NthField(s,Chr(13),i))
next

If fileFld.text <> "" then //gibt es einen Pfad zu einer Attachment-Datei?
  file = New EmailAttachment
  file.loadFromFile GetFolderItem(fileFld.text)
  mail.attachments.append file //Füge Datei dem Attachments-Array hinzu
end

SMTPSocket1.messages.append mail //Füge E-Mail der Liste der Nachrichten hinzu
SMTPSocket1.SendMail //Sende Nachricht

```

Siehe auch

[POP3Socket](#), [POP3SecureSocket](#), [EmailMessage](#), [TCPSocket](#), [HTTPSocket](#), [HTTPSecureSocket](#), [SMTPSecureSocket](#), [SocketCore](#) Klassen.

Socket Klasse

Socket Steuerelemente werden verwendet, um über TCP/IP zu kommunizieren. Diese Klasse ist obsolet. Die Klasse [TCPSocket](#) bietet die Funktionalität der alten Socket-Klasse an. Das **Socket**-Steuerelement in der Steuerelementen-Palette ist nun ein [TCPSocket](#)-Steuerelement. Zusätzlich unterstützt REALbasic nun über die [UDPSocket](#)-Klasse das UDP-Protokoll.

Siehe auch

[TCPSocket](#), [SSLSocket](#), [ServerSocket](#), [SocketCore](#), [UDPSocket](#) Klassen.

SocketCore Klasse

Die Basis-Klasse für [TCPSocket](#) und [UDPSocket](#). Die [SSLSocket](#)-Klasse ist abgeleitet von [TCPSocket](#). **SocketCore** ist eine abstrakte Klasse und kann nicht direkt instanziiert werden. Diese Gruppe von Klassen ersetzt das [Socket](#)-Steuerelement und erweitert REALbasics Fähigkeiten der Kommunikation.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
<i>IsConnected</i>	<u>Boolean</u>	Zeigt an, ob ein Socket derzeit verbunden ist. Bei <u>TCP.Sockets</u> bedeutet eine Verbindung, daß Sie Daten senden und empfangen können und mit einem entfernten Rechner verbunden sind. Bei <u>UDP.Sockets</u> bedeutet dies, daß Sie an diesen Port gebunden sind und in der Lage sind, Multicast-Gruppen zu senden, zu empfangen, beizutreten oder zu verlassen oder Socket-Optionen zu setzen.
<i>LastErrorCode</i>	<u>Integer</u>	Der letzte Fehlercode für den Socket.
<i>LocalAddress</i>	<u>String</u>	Die lokale IP-Adresse des Computers.
<i>Port</i>	<u>Integer</u>	Der Port, über den verbunden werden soll. Unter Mac OS X führt ein Versuch, über einen Port kleiner 1024 zu verbinden, zu einem SocketCore.Error-Event mit einer Fehlernummer 105, außer die Applikation läuft mit Root-Rechten. Das liegt an Apples Implementation von Open Transport und derzeit gibt es da keinen Workaround.

Events

Name	Parameter	Beschreibung
<i>DataAvailable</i>		Tritt auf, wenn zusätzliche Daten in den internen Empfangs-Puffer kamen.
<i>Error</i>		Tritt auf, wenn mit dem Socket ein Fehler auftrat.
<i>SendComplete</i>	UserAborted als <u>Boolean</u>	Tritt auf, wenn ein Versenden abgeschlossen ist. Verwenden Sie dies, um festzustellen, ob alle Ihre Daten gesendet wurden. <i>UserAborted</i> ist <u>True</u> , wenn der Anwender das Senden durch Zurückgeben von <u>True</u> aus dem SendProgress-Event abbricht. Sie können diese Information verwenden, um verschiedene Status-Variablen zu aktualisieren oder den Anwender auf einen Erfolg oder Mißerfolg hinzuweisen. Wenn das Senden fertig ist, ist dieser Wert <u>False</u> . <i>UserAborted</i> ist bei UDP-Sockets immer <u>False</u> . <i>UserAborted</i> ist <u>True</u> , wenn der der SendProgress-Event <u>True</u> zurückgibt. Sie können diese Information verwenden, um Statusvariablen zu aktualisieren oder dem Benutzer eine Rückmeldung zu geben, wenn die Übertragung fehlschlägt. War die Übertragung schon vollendet, ist <i>UserAborted</i> <u>False</u> . Bei UDP-Sockets ist es immer <u>False</u> .

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Schließt die Verbindung des Sockets, schließt alle Verbindungen, die der Socket haben kann und setzt den Socket zurück. Die einzigen Informationen, die nach dem Aufruf von Close erhalten bleiben, sind der Port, die Adresse (im Falle eines <u>TCP.Sockets</u>), die LastErrorCode-Eigenschaften des Sockets und die restlichen Daten im Empfangspuffer der Socket. Alle anderen Informationen werden verworfen.
<i>Connect</i>		Versucht, eine Verbindung herzustellen. Bei <u>TCP.Sockets</u> müssen die Adresse- und die Port-Eigenschaften gesetzt werden. Bei <u>UDP.Sockets</u> muß die Port-Eigenschaft gesetzt werden. Die Connect-Methode bindet ein Socket an einen Port. Nach dem Aufruf von Connect gibt die Port-Eigenschaft den Port an, an den Sie momentan gebunden sind.
<i>Poll</i>		Pollt den Socket manuell, was einem Socket erlaubt, synchron verwendet zu werden.
<i>Purge</i>		Entfernt alle Daten aus dem internen Empfangs-Puffer des Sockets. Es wirkt sich nicht auf den internen Sende-Puffer des Sockets aus.

Fehler-Codes

Die LastErrorCode-Eigenschaft enthält einen Integer-Wert, der dem letzten Fehler-Code entspricht. Diese Fehler-Codes geben Ihnen wichtige Informationen über Ihren Socket und Sie sollten dies keinesfalls ignorieren.

Jeder Fehler-Code entspricht einer Klassen-Konstante. Wenn Sie herausfinden wollen, welcher Fehler aufgetreten ist, so können Sie den Wert der LastErrorCode-Eigenschaft mit diesen Konstanten vergleichen.

Verwenden Sie die folgenden Konstanten für den Vergleich mit der `LastErrorCode`-Eigenschaft.

Fehler-Code	Klassen-Konstante	Beschreibung
100	<code>OpenDriverError</code>	Der Treiber konnte nicht geöffnet werden. Das heißt, daß entweder Open Transport (Macintosh) oder WinSock (Windows) nicht installiert sind oder die Version nicht den Anforderungen entspricht.
101	<code>LostConnection</code>	Dieser Fehler-Code wird seit Version 5.0 nicht länger verwendet.
102		Die Verbindung wurde geschlossen. Entweder von der Gegenstelle mit einem ordentlichen Aufruf der <code>Close</code> -Methode der <code>SocketCore</code> -Klasse oder durch ein gewaltsames Abziehen des Netzkabels. Dies ist nicht unbedingt ein echter Fehler. Dieser Fehler tritt auch auf, wenn Sie die <code>Disconnect</code> -Methode des <code>TCP Sockets</code> aufrufen.
103		<code>REALbasic</code> konnte die Adresse nicht auflösen. Dies kann eine unbekannte IP-Adresse sein oder ein Hostname/Domainname, der nicht aufgelöst werden konnte.
104	<code>AddressInUseError</code>	Dieser Fehler-Code wird seit Version 5.0 nicht länger verwendet.
105		Die Adresse wird bereits benutzt. Dieser Fehler tritt auf, wenn Sie sich an einen Port binden wollen, an den Sie bereits gebunden sind. Ein Beispiel wäre, wenn Sie an einem Socket lauschen (listen) und versuchen, mit einem zweiten Socket an diesem Port lauschen.
106		Dieser Fehler bedeutet, daß der Socket nicht in dem Status ist, um die geforderte Aktion durchzuführen. Ein Beispiel wäre, mit einem Socket Lesen oder Schreiben zu wollen, ohne daß dieser eine Verbindung aufgebaut hat.
107	<code>InvalidPortError</code>	Dieser Fehler besagt, daß der angegebene Port ungültig ist. Dies bedeutet, daß Sie eine Port-Nummer unter 0 oder über 65.535 angegeben haben. Es kann aber auch bedeuten, daß Sie nicht die erforderlichen Rechte besitzen, um diesen Port zu benutzen. Unter Mac OS X oder Linux kann nur der Benutzer „root“ Ports im Bereich von 0 bis 1024 benutzen.
108	<code>OutOfMemoryError</code>	Dieser Fehler bedeutet, daß Ihre Applikation nicht genügend Speicher zur Verfügung hat.

Dies sind nicht die einzigen Fehler, die `LastErrorCode` zugewiesen werden können. Wenn `REALbasic` einen Fehler-Code nicht korrekt umwandeln kann, wird der Fehler-Code der aktuellen Plattform in diese Eigenschaft geschrieben. Auf dem Macintosh haben diese einen Bereich von -3211 bis -3285. Unter Windows einen Bereich von 10004 bis 11004. Für eine Beschreibung dieser Fehler-Codes können Sie einen Blick in `MacErrors.h`, beziehungsweise `WinSock.h` werfen.

Hinweise

Es gibt Situationen, in denen Sie das System einen Port auswählen lassen möchten. Zum Beispiel bei einem passiven FTP-Transfer, da hier ganze Port-Bereiche zur Verfügung stehen. Wenn Sie als Port 0 angeben und dann `TCP Socket.Listen` oder `UDP Socket.Connect` aufrufen, wird das System einen offenen Port für Sie auswählen. Meistens werden diese Ports im Bereich von 49512 bis 65535 (inklusive) liegen.

Wenn Sie diese Art der Portauswahl benutzen, möchten Sie wahrscheinlich wissen, welcher Port tatsächlich ausgewählt wurde. Nach dem Aufruf der `Connect`- oder `Listen`-Methode der `TCP Socket`-Klasse können Sie die `Port`-Eigenschaft auslesen. Das heißt, daß die Portnummer von 0 auf die zugewiesene Nummer wechselt.

Es gibt einen weiteren Vorteil. Es gibt eine Methode zum Hacken, bei der einer Applikation der aktuelle Port entzogen wird. Diese Technik wird „port hijacking“ genannt. Wenn Sie die `Port`-Eigenschaft prüfen, können Sie feststellen, ob dieser Fall eingetreten ist. Es kann eine gute Idee sein, den Port sporadisch zu überprüfen. Zum Beispiel: Wenn Sie für HTTP-Verbindungen auf Port 80 lauschen und feststellen, daß die `Port`-Eigenschaft aber 2113 zurückliefert, wissen, daß etwas schief läuft.

Der Parameter des `SendComplete`-Event läßt Sie wissen, ob der Transfer abgeschlossen wurde oder ob er durch einen Rückgabewert von `True` vom `SendProgress`-Event der `TCP Socket`-Klasse abgebrochen wurde. Sie können diese Informationen verwenden, um Ihre Status-Variablen entsprechend zu setzen. Wenn der Benutzer die Übertragung abgebrochen hat, ist dieser Parameter `True` und wenn

die Übertragung vollständig war False. Dies ist ein neues Feature für Macintosh und Windows. Der `SendComplete`-Parameter ist immer False, wenn `UDPSockets` verwendet werden, da diese keinen `SendProgress`-Event haben.

Byte-Reihenfolge (Endianess)

Es gibt zwei Standards für die Bytes-Reihenfolge. Macintoshs verwenden „big“, Windows und x86 Linux „little“. Sockets arbeiten mit Datenströmen und verändern die Byte-Reihenfolge der Daten nicht. Dies ist hilfreich, wenn man Daten zwischen verschiedenen Plattformen (zum Beispiel Macintosh und Windows) verschickt. Wenn Sie aber binäre Daten wie BinaryStream oder einen MemoryBlock verschicken wollen, müssen Sie sicherstellen, daß jeweils die Byte-Reihenfolge des Servers oder des Clients verwendet wird. Sie können dies, indem Sie die `LittleEndian`-Eigenschaft eines MemoryBlocks oder BinaryStreams auf den gleichen Wert setzen wie der Server oder Client. Ist die Byte-Reihenfolge nicht gewährleistet, kann es zu unterschiedlichsten Fehlern in Ihrer Applikation kommen, wenn diese eine bestimmte Reihenfolge voraussetzt.

Siehe auch

TCPSocket, ServerSocket, UDPSocket, Datagram Klassen.

Sort Methode

Sortiert die Elemente eines Arrays in aufsteigender Reihenfolge.

Syntax

`array.Sort`

Teil	Beschreibung
<code>array</code>	Das zu sortierende Array.

Hinweis

Die Methode **Sort** arbeitet ausschließlich mit Integern, Strings sowie eindimensionale Arrays vom Typ Single oder Double.

Beispiele

Dieses Beispiel sortiert das Array Namen.

```
Namen.Sort
```

Siehe auch

Dim Befehl; Array, Split, Join, Ubound Funktionen; Append, Insert, Redim, Remove, Pop, IndexOf, Insert, Shuffle, Methoden; ParamArray Schlüsselwort.

Sound Klasse

Wird verwendet, um Klänge abzuspielen.

Übergeordnete Klasse

Object

Methoden

Name	Parameter	Beschreibung
<i>IsPlaying</i>		Liefert <u>True</u> während der Klang abgespielt wird.
<i>Pan</i>		Legt die relative Lautstärke (Balance) zwischen linkem und rechtem Lautsprecher fest. Der Wert reicht von -100 bis +100, wobei 0 die gleiche Lautstärke für den linken und rechten Kanal bedeutet. -100 spielt den Klang nur über den linken Kanal, 100 nur über den rechten Kanal. Voreingestellt ist 0.
<i>Play</i>		Spielt den Klang.
<i>PlayLooping</i>		Spielt den Klang in einer Endlosschleife.
<i>Stop</i>		Stoppt den Klang, der gespielt wird.

Name	Parameter	Beschreibung
<i>Volume</i>		Steuert die Lautstärke des Klangs. Der Wert reicht von 0 bis 100; 0 stellt den Klang auf Stumm, 100 spielt den Klang mit „normaler“ Lautstärke, die bei den Lautstärke-Einstellungen auf dem Rechner eingestellt ist. Voreingestellt ist 100.

Hinweis

Auf Klänge, die ins Projekt-Fenster gezogen wurden, kann über ihren Objektnamen zugegriffen werden. Auf Klänge kann auch über die Methode `OpenAsSound` mit einem `FolderItem` zugegriffen werden. Es gibt für Klangobjekte keine Eigenschaften oder Events.

Die Macintosh-Version unterstützt alle Formate, die Quicktime bekannt sind. Sie können beliebig viele Töne gleichzeitig abspielen und sind nur durch die Systemressourcen begrenzt.

Beispiele

Dieses Beispiel spielt einen Klang „Sledgehammer“ ab, der ins Projekt-Fenster gezogen wurde.

```
sledgehammer.play
```

Dieses Beispiel liest den Klang aus einer Datei „Hurz!“ aus dem aktuellen Ordner (Verzeichnis) in ein Sound-Objekt und spielt ihn ab.

```
Dim f as FolderItem
Dim s as Sound
f = GetFolderItem("Hurz!")
s = f.OpenAsSound
s.play
```

Folgendes Beispiel spielt den Klang „Giggle“, der ins Projekt-Fenster gedraggt wurde, in einer Endlosschleife:

```
Giggle.playlooping
```

Dieses Beispiel stoppt „Giggle“:

```
Giggle.stop
```

Siehe auch

[NotePlayer](#) Steuerelement.

Speak Methode

Verwendet den internen Synthesizer, um den angegebenen Text auszusprechen.

Syntax

Speak (phrase, [Interrupt])

Teil	Typ	Beschreibung
phrase	<u>String</u> oder <u>Variant</u>	Text, der gesprochen werden soll. Speak akzeptiert einen beliebigen <u>Variant</u> , der als <u>String</u> interpretiert werden kann.
Interrupt	<u>Boolean</u>	Optionaler Parameter. Wenn <u>True</u> , unterbricht der aktuelle Aufruf einen vorhergehenden. Wird er weggelassen, wird <u>False</u> angenommen.

Hinweise

Speak benutzt einen string (oder beliebigen Variant der als String interpretiert werden kann) und verwendet die Macintosh- oder Windows-Text-To-Speech-Engine (Text-zu-Sprache), um den Text als Sprache auszugeben. Die Ausgabe erfolgt asynchron und erlaubt einen normalen Programmfluß. In der Voreinstellung werden kontinuierliche Aufrufe von **Speak** gepuffert und nacheinander ausgeführt, ohne den vorherigen Aufruf zu unterbrechen. Dies kann mit dem optionalen Interrupt-Parameter geändert werden. Geben Sie True an, wenn Sie den vorherigen Aufruf abbrechen wollen.

Speak ist unter Linux nicht verfügbar.

Beispiel

Der folgende Code eines `PushButtons` liest den eingegebenen Text in einem `EditField` vor:

```
If EditField1.text <> "" then
    Speak EditField1.text
Else
    Speak "Bitte geben Sie etwas Text in das Feld ein!"
End if
```

Split Funktion

Erzeugt ein eindimensionales Array aus dem angegebenen String.

Syntax

Ergebnis=**Split**(source, [delimiter])

ODER

source.**Split**([delimiter])

Teil	Typ	Beschreibung
Ergebnis	<u>String</u> array	Ein Array, das die Textteile enthält, nachdem der angegebene String mit dem angegebenen Trennzeichen zerteilt wurde.
source	<u>String</u>	Der String, der zerteilt werden soll.
delimiter	<u>String</u>	Optionales Trennzeichen mit dem der String zerteilt werden soll. Wird keines angegeben, wird ein Leerzeichen verwendet.

Hinweise

Verwenden Sie die Split-Funktion, wenn Sie ein String-Array aus einer Liste von Elementen erzeugen wollen, die mit einem Trennzeichen getrennt sind. Über den optionalen Paramter können Sie ein Trennzeichen angeben. Ist kein Trennzeichen angegeben, wird ein Leerzeichen verwendet. Ist das Trennzeichen ein leerer String, wird der String in seine Buchstaben zerteilt.

Beispiele

Das erste Beispiel gibt ein Komma-Trennzeichen an und das zweite verwendet das voreingestellte Trennzeichen. Sie legen jeden Teil des String als ein Array-Element ab. Das letzte Beispiel erzeugt ein Array mit vier Elementen.

```
Dim anArray(-1) as String
anArray=Split("Adam,Aardvark,Accountant",",")
anArray=Split("Adam Aardvark Accountant")
anArray=Split("Adam",",")
//Die ersten beiden mit der alternativen Syntax:
Dim s as String
s="Adam,Aardvark,Accountant"
anArray=s.Split("",",") //erzeugt ein 3-Element-Array
anArray=s.Split("")//erzeugt ein Array mit einzelnen Zeichen
```

Siehe auch

Dim Befehl; String Datentyp; NthField, Join, Array Funktionen.

Dim Befehl; Array, Join, Ubound Funktionen; Append, Insert, Redim, Remove, Pop, IndexOf, Insert, Shuffle, Sort Methoden; ParamArray Schlüsselwort.

Sprite Klasse

Ein Bild kann mit dem Steuerelement SpriteSurface animiert werden.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Group</i>	<u>Integer</u>	Entscheidet, ob der Event-Handler „Collision“ für das Steuerelement <u>SpriteSurface</u> aufgerufen wird, wenn das Sprite mit einem anderen Sprite in Kontakt gerät. Sprites mit einem negativen Group-Wert kollidieren mit jedem Sprite (außer mit denen der Group 0). „Group 0“ bedeutet, daß das Sprite mit anderen Sprites nicht kollidieren wird. Ein positiver Wert für Group bedeutet, daß das Sprite mit Sprites einer anderen Group kollidieren wird, nicht aber mit denen, die den gleichen Group-Wert haben.
<i>Image</i>	<u>Picture</u>	Das Bild, das gezeichnet wird, wenn die Methode Run für das SpriteSurface aufgerufen wird.
<i>Priority</i>	<u>Integer</u>	Definiert die Reihenfolge, in der das Sprite im Event-Handler „NextFrame“ gezeichnet wird.

Name	Art	Beschreibung
ScreenX	Integer	Der Abstand des Sprites zur linken Seite des SpriteSurface in Pixeln.
ScreenY	Integer	Der Abstand des Sprites zur oberen Kante des SpriteSurface in Pixeln.
X	Integer	Die Position der linken Seite des Bildes im SpriteSurface -Bereich.
Y	Integer	Die Position der oberen Seite des Bildes im SpriteSurface -Bereich.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Entfernt das Sprite aus dem SpriteSurface , in dem es angelegt wurde.

Hinweise

Ein Sprite-Objekt ist ein Bild, das mit Hilfe eines [SpriteSurface](#) Steuerelements animiert werden kann. Das Sprite kann über den Bildschirm bewegt werden, indem Sie im Event-Handler „NextFrame“ des entsprechenden [SpriteSurface](#)-Steuerelements die Sprite-Eigenschaften X und Y verändern. Sollten sich zwei Sprites während der Animation berühren, so besteht die Möglichkeit einer *Kollision*. „Kollision“ bedeutet, daß der [SpriteSurface](#)-Event-Handler „Collision“ aufgerufen wird und ihm die beiden kollidierenden Sprites übergeben werden. Dies gibt Ihnen die Möglichkeit zu entscheiden, was passieren soll, wenn sich die beiden Sprites berühren. Ob eine Kollision auftritt oder nicht entscheidet sich anhand der Group-Werte der Eigenschaften der beiden Sprites:

Group Werte	Kollision
<i>0 für ein beliebiges Sprite</i>	Nein
<i>Beide Sprites haben den gleichen positiven Wert</i>	Nein
<i>Beide Sprites haben verschiedene positive Werte</i>	Ja
<i>Ein Sprite hat einen negativen Wert</i>	Ja

Sprites werden während des Event-Handlers „NextFrame“ in der Reihenfolge ihrer [SpriteSurface](#)-Eigenschaft Priority gezeichnet. Diese Eigenschaft ist in der Reihenfolge vorbelegt, in der die Sprites angelegt werden. Die Priority des ersten Sprites ist 1, die Priority des zweiten Sprites ist 2, usw. In diesem Fall wird das zweite Sprite über dem ersten Sprite gezeichnet.

Alle Eigenschaften eines Sprite-Objekts können zur Laufzeit geändert werden. Dadurch können Sie Bild, Position, Group und Priority eines Sprites während dessen Animation verändern.

Ein Sprite kann mit dem New-Operator erzeugt und mit der Attach-Methode der SpriteSurface-Klasse einem SpriteSurface verbunden werden.

Vergleichen von ScreenX und X (und ScreenY und Y)

ScreenX und X (und ScreenY und Y) sind bei einem SpriteSurface, das nicht gescrollt wurde, gleich. Wenn es gescrollt wurde, ändern sich die Werte von ScreenX und ScreenY der Sprites, da sich deren Positionen auf dem Bildschirm geändert haben. Deren X- und Y-Werte bleiben gleich, da sich deren Position in „Welt“-Koordinaten nicht geändert hat.

Masken- und Transparenz-Eigenschaften

Sprites ignorieren sowohl die Mask- als auch die Transparency-Eigenschaft des Pictures. Weiße Pixel sind immer transparent und nicht-weiße Pixel sind immer undurchsichtig.

Beispiele

Bei der Beschreibung des Steuerelements [SpriteSurface](#) finden Sie ein Beispiel.

Siehe auch

[SpriteSurface](#) Steuerelement

SpriteSurface Steuerelement



Wird verwendet, um eine Sprite-Animation anzulegen.

Übergeordnete Klasse

[RectControl](#)

Eigenschaften

Name	Art	Beschreibung
<i>BackDrop</i>	<u>Picture</u>	Ein Bild, daß im durch SurfaceHeight, SurfaceLeft, SurfaceTop und SurfaceWidth definierten Bereich gezeichnet wird, wenn die Run-Methode von SpriteSurface aufgerufen wird. <u>Sprite</u> -Objekte werden vor dem Backdrop gezeichnet.
<i>ClickToStop</i>	<u>Boolean</u>	Wenn <u>True</u> , so stoppt ein Mausklick die Animation des SpriteSurface.
<i>Depth</i>	<u>Integer</u>	Farbtiefe der SpriteSurface-Bereichs in Bits. Kann die Werte 0, 8, 16 und 32 annehmen. Die Angabe von 0 führt dazu, daß automatisch beim Öffnen des SpriteSurface seine Farbtiefe an die der aktuellen Monitoraufösung angepaßt wird. Die Depth-Eigenschaft kann während des Betriebs geändert werden.
<i>FrameSpeed</i>	<u>Integer</u>	FrameSpeed legt fest, wie oft der Bildschirminhalt neu gezeichnet werden soll, bevor ein Frame neu gezeichnet wird. Eine FrameSpeed von 0 sorgt dafür, daß der Rechner die Frames mit der höchstmöglichen Geschwindigkeit neu zeichnet. Um die passende FrameSpeed zu berechnen, teilen Sie 60 durch die Anzahl der Frames, die pro Sekunde gezeichnet werden sollen und runden Sie das Ergebnis auf den nächsten Integer-Wert. Jedes Neuzeichnen eines Frames löst ein NextFrame-Event aus. Bitte beachten Sie, daß sich die Definition von FrameSpeed im Vergleich zur Version 1.0 geändert hat und passen Sie Ihre unter REALbasic 1.0 entwickelten Projekte entsprechend an.
<i>Graphics</i>	<u>Graphics</u>	Wird verwendet, um auf den Bildschirm zu zeichnen, während das SpriteSurface läuft. Vermeiden Sie es, in den Bereich zu zeichnen, der über die Eigenschaften SurfaceHeight, SurfaceLeft, SurfaceTop und SurfaceWidth definiert ist, da das SpriteSurface diesen Bereich behandelt.
<i>Height</i>	<u>Integer</u>	Die Höhe des SpriteSurface-Steuerelements in lokalen Koordinaten (relativ zum Fenster).
Index		Die Nummer des Steuerelements, wenn es Teil eines Steuerelement-Arrays ist.
<i>Left</i>		Die linke Seite des Steuerelements in lokalen Koordinaten (relativ zum Fenster).
<i>LockBottom</i>	<u>Boolean</u>	Legt fest, ob die untere Kante des SpriteSurface in einer festen Entfernung zur unteren Kante des zugehörigen Fensters bleiben soll.
<i>LockLeft</i>	<u>Boolean</u>	Legt fest, ob die linke Kante des SpriteSurface in einer festen Entfernung zur linken Kante des zugehörigen Fensters bleiben soll. <i>LockLeft</i> hat keine Auswirkung, solange <i>LockRight</i> nicht True ist.
<i>LockRight</i>	<u>Boolean</u>	Legt fest, ob die rechte Kante des SpriteSurface in einer festen Entfernung zur rechten Kante des zugehörigen Fensters bleiben soll.
<i>LockTop</i>	<u>Boolean</u>	Legt fest, ob die obere Kante des SpriteSurface in einer festen Entfernung zur oberen Kante des zugehörigen Fensters bleiben soll. <i>LockTop</i> hat keine Auswirkung, solange <i>LockBottom</i> nicht True ist.
Name		Der Name des Steuerelements.
<i>ScrollX</i>	<u>Integer</u>	Entfernung, die das SpriteSurface in horizontaler Richtung gescrollt wurde.
<i>ScrollY</i>	<u>Integer</u>	Entfernung, die das SpriteSurface in vertikaler Richtung gescrollt wurde.
Super		Die Objektklasse, auf der das Steuerelement basiert.
<i>SurfaceHeight</i>	<u>Integer</u>	Nun das gleiche wie Height.
<i>SurfaceLeft</i>	<u>Integer</u>	Nun das gleiche wie Left.
<i>SurfaceTop</i>	<u>Integer</u>	Nun das gleiche wie Top.
<i>SurfaceWidth</i>	<u>Integer</u>	Nun das gleiche wie Width.
<i>Top</i>		Die obere Seite des SpriteSurface-Steuerelements in lokalen Koordinaten (relativ zum Fenster).
<i>Width</i>		Die Breite des SpriteSurface-Steuerelements in lokalen Koordinaten (relativ zum Fenster).

Events

Name	Parameter	Beschreibung
<i>Collision</i>	S1 als <u>Sprite</u> , S2 als <u>Sprite</u>	Zwei <u>Sprites</u> mit verschiedenen Group-Eigenschaften (oder negativen Group-Eigenschaften) sind zusammengestoßen (berührt sich).
<i>NextFrame</i>		Das SpriteSurface ist bereit, das nächste Bild der Animation zu zeichnen. <u>Sprite</u> -Objekte werden am Ende dieses Event-Handlers gezeichnet.
<i>PaintTile</i>	g als <u>Graphics</u> xpos als <u>Integer</u> ypos als <u>Integer</u>	Tritt auf, wenn die Scroll-Methode aufgerufen wurde. g ist ein <u>Graphics</u> -Objekt und xpos und ypos sind die Koordinaten des Rechtecks, das neu gezeichnet werden muß.

Methoden

Name	Parameter	Beschreibung
<i>Attach</i>	S als <u>Sprite</u>	Erlaubt es, ein Sprite mit einem SpriteSurface zu verbinden.
<i>Close</i>		Hält die Animation an und übergibt die Kontrolle des Bildschirms wieder komplett an das Fenster zurück, das das SpriteSurface Steuerelement enthält.
<i>KeyTest</i>	KeyCode als <u>Integer</u>	Liefert <u>True</u> , wenn die über ihren KeyCode angegebene Taste gedrückt wurde.
<i>NeuSprite</i>	Bild als <u>Picture</u> , X als <u>Integer</u> , Y als <u>Integer</u>	Liefert ein neues <u>Sprite</u> -Objekt und zeichnet dessen Bild mit den angegebenen Koordinaten im angegebenen SpriteSurface.
<i>PaintTile</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Sorgt dafür, daß der durch x,y angegebene Hintergrundbereich neu gezeichnet wird und löst Ihr Paint-Tile-Event aus. Wenn Sie Teile des Hintergrundes während eines Spiels ändern müssen, ist dies die beste Möglichkeit, zu erreichen, daß der Hintergrund aktualisiert wird. (Beachten Sie, daß die Aktualisierung auf dem Bildschirm nicht vor dem nächsten Frame erscheint).
<i>Run</i>		Startet die Animation.
<i>Scroll</i>	dx als <u>Integer</u> dy als <u>Integer</u>	Scrollt das SpriteSurface um den Wert von dx und dy.
<i>Stop</i>		Stoppt die Animation.
<i>Update</i>		Läßt die Animation pro Aufruf einen Schritt weiterlaufen. Es zeichnet das Sprite neu, das bewegt wurde oder sich geändert hat, versendet Kollisionsmeldungen und ruft einmal das NextFrame-Event auf.

Hinweise

Größe und Position des SpriteSurface-Bereichs festlegen

Das SpriteSurface-Steuerelement ist ein normales sichtbares Steuerelement, das aus der Steuerelemente-Palette in ein Fenster gedraggt werden kann. Sie können es zu beliebiger Größe aufziehen oder indem Sie den Eigenschaften Left, Top, Width und Height Werte zuweisen. Diese Eigenschaften können gelesen und geschrieben werden und können somit auch zur Laufzeit die Größe und Position eines SpriteSurface verändern.

Sie können außerdem die zwei Eigenschaften FullScreen und MenuBarVisible des Elternfensters einsetzen, damit das SpriteSurface sich über den gesamten Bildschirm erstreckt. Setzen Sie Set FullScreen auf True und MenuBarVisible auf False und setzen Sie die Lock...-Eigenschaften des SpriteSurface-Steuerelements, um es fest an das Elternfenster zu binden, so daß sich der Bereich des SpriteSurface automatisch vergrößert. Dies paßt automatisch die Eigenschaften Left, Top, Width und Height an.

Animationen mit Sprites anlegen

Das Steuerelement SpriteSurface wird verwendet, um eine Animation anzulegen, bei der Bilder auf dem Bildschirm bewegt werden müssen und das Neuzeichnen (Redraw) automatisch vom SpriteSurface gehandhabt werden soll. Jedes Bild ist ein Sprite-Objekt. Sprite-Objekte haben X- und Y-Eigenschaften, die ihre aktuelle Position auf dem Bildschirm angeben, während das SpriteSurface die aktuelle Sprite-Animation durchläuft.

Bewegen und Ändern von Sprites

Der Event-Handler „NextFrame“ wird jedes Mal aufgerufen, wenn das SpriteSurface bereit ist, das nächste Bild einer Animation zu zeichnen. Wenn Sie möchten, daß sich ein Sprite im nächsten Bild relativ zum letzten Sprite bewegt, ändern Sie die Eigenschaften X und Y im Event-Handler „NextFrame“. Wenn Sie das Bild für den nächsten Frame austauschen möchten, ändern Sie hierzu die Eigenschaft „Image“ im Event-Handler „NextFrame“. Sie können die Update-Methode verwenden, um die Animation schrittweise ablaufen zu lassen. Dies ruft den NextFrame-Eventhandler nur einmal auf.

Um ein Sprite aus einer Animation zu löschen, müssen Sie nur die Clode-Methode des Sprites aufrufen.

Frame-Redraw

Die Geschwindigkeit, mit der die Frames gezeichnet werden, basiert auf der Eigenschaft FrameSpeed. Je höher FrameSpeed, desto schneller wird die Animation ablaufen.

Um die Animation zu starten, rufen Sie die Run-Methode des SpriteSurface auf. Um die Animation zu beenden, rufen Sie die Close-Methode des SpriteSurface auf.

Momentane Implementation

Die Größe und Position des SpriteSurface-Bereichs wird durch die Eigenschaften Left, Top, Height und Width gesteuert. Eine andere Möglichkeit ist das Setzen der FullScreen-Eigenschaft des Elternfensters auf True und das Verwenden der Eigenschaften LockLeft, LockTop, LockRight und LockBottom, damit sich die Größe des SpriteSurface automatisch mit der des Fensters ändert.

Wenn die Run-Methode einmal aufgerufen wurde, werden solange NextFrame-Events aufgerufen, bis der Anwender die Maustaste drückt (wenn die Eigenschaft ClickToStop True ist). Ist ClickToStop False, so läuft die Animation immer weiter, bis das Fenster, in dem sich das SpriteSurface befindet, geschlossen wird.

Beispiele

In diesem Beispiel wird ein Sprite horizontal auf dem Bildschirm vor- und zurückbewegt. Der PushButton-Event-Handler „Action“ legt ein Sprite an und sichert es unter dem Namen LisaSprite in einer Fenster-Eigenschaft vom Typ Sprite. Das Bild für das Objekt LisaSprite ist ein PICT-Bild namens LisaPict, das ins Projekt-Fenster gezogen wurde. Das Steuerelement SpriteSurface namens SpriteSurface1 überwacht die Animation. Der Pushbutton startet über die Run-Methode die Animation des Steuerelements SpriteSurface1. Eine andere Fenster-Eigenschaft vom Typ Integer mit den Namen LisaDirection wird verwendet, um die momentane Richtung des LisaSprite festzulegen. Dies sind die Event-Handler:

PushButton1:

```
Sub Action()
    LisaSprite=SpriteSurface1.NewSprite(LisaPict,200,200)
    LisaDirection=0
    SpriteSurface1.Run
End Sub
```

SpriteSurface1:

```
Sub NextFrame()
    If LisaSprite.x>=640 Then
        LisaDirection=1
    ElseIf LisaSprite.x<=0 then
        LisaDirection=0
    End if
    Select Case LisaDirection
    Case 0
        LisaSprite.x=LisaSprite.x+1
    Case 1
        LisaSprite.x=LisaSprite.x-1
    End Select
End Sub
```

Dieses Beispiel beendet (mit Hilfe der Close-Methode) das aktuell laufende SpriteSurface-Steuerelement, wenn der Anwender die Return-Taste drückt:

```
Sub NextFrame()
    If Me.KeyTest(&h24) Then
        Me.Close
    End if
End Sub
```

Ein wesentlich ausgereifteres Beispiel namens „REALSprites“ finden Sie auf dem FTP-Server von REAL Software und auf der REALbasic-CD. Dieses Beispiel verwendet auch eine neue Klasse, um Sprites zu kontrollieren.

Folgendes Beispiel verwendet das PaintTile-Event, um ein Bild auf dem Bildschirm zu bewegen, wenn der Anwender die Pfeiltasten drückt. Das PaintTile-Event tritt auf, wenn die Scroll-Methode aufgerufen wird.

Das NextFrame-Event von SpriteSurface hat folgenden Code:

```
If spriteSurface1.keytest(&h7E) then //hoch
    me.scroll 0, -10
elseif spriteSurface1.keytest(&h7D) then //runter
    me.scroll 0, 10
elseif spriteSurface1.keytest(&h7B) then //links
    me.scroll -10,0
elseif spriteSurface1.keytest(&h7C) then //rechts
    me.scroll 10,0
elseif spriteSurface1.keytest(&h24) then //Ende
    me.close
end if
```

Jede Bedingung testet, ob der Anwender eine Pfeiltaste gedrückt hat und ruft dann die Scroll-Methode mit den passenden x- und y-Werten auf. Das PaintTile-Event verwendet folgenden Code:

```
Sub PaintTile (g as Graphics, xpos as integer, ypos as integer)
    g.foreColor=rgb(0,0,0) //setzt die Vordergrundfarbe auf schwarz
    g.fillRect 0,0,64,64 //löscht die Kachel
    g.foreColor=rgb(255,255,255) //setzt die Vordergrundfarbe auf weiß
    g.drawString str(xpos)+" "/" "+str(ypos), 0,32
end sub
```

Siehe auch

[Sprite](#) Objekt.

Sqrt Funktion

Liefert die Quadratwurzel des übergebenen Wertes.

Syntax

Ergebnis = **Sqrt** (Wert)

Teil	Art	Beschreibung
<i>Wert</i>	<u>Double</u>	Der Wert, dessen Quadratwurzel Sie benötigen.
<i>Ergebnis</i>	<u>Double</u>	Die Quadratwurzel von <i>Wert</i> .

Hinweis

Die Funktion **Sqrt** liefert die Quadratwurzel des übergebenen Wertes.

Beispiele

Dieses Beispiel verwendet die Funktion **Sqrt**, um die Quadratwurzel einer Zahl zu ermitteln.

```
Dim d As Double
d=Sqrt(16) //ergibt 4
```

SSLSocket Steuerelement

Verwenden Sie das **SSLSocket**-Steuerelement, um eine gesicherte Kommunikation über TCP/IP unter Verwendung der Secure Sockets Layer (SSL)-Technologie durchzuführen. Das **SSLSocket**-Steuerelement ist nur in der Professional-Version von REALbasic verfügbar. die Standard-Version von REALbasic compiliert keine Verweise auf das **SSLSocket**-Steuerelement.

SSLSocket unterstützt derzeit nicht das Lauschen an Secure Sockets.

Übergeordnete Klasse

[TCPSocket](#) Klasse.

Eigenschaften

Name	Typ	Beschreibung
<i>ConnectionType</i>	<u>Integer</u>	Legt die Art der SSL-Verbindung fest. <i>ConnectionType</i> kann folgende Werte annehmen: 0 - SSLv2: SSL (Secure Sockets Layer) Version 2. 1 - SSLv23: SSL Version 3, kann aber auf Version 2 zurückfallen, falls nötig. 2 - SSLv3: SSL Version 3. 3 - TLSv1: TLS (Transport Layer Security) Version 1. Der Defaultwert ist 1 (Verbindungstyp SSLv23). Nachdem Sie die <i>SSLSocket.Connect</i> -Methode einmal aufgerufen haben, können Sie den Verbindungstyp nicht mehr ändern, ohne zuvor <i>SSLSocket.Close</i> aufzurufen, Sie können aber jederzeit den Verbindungstyp erfragen. Wenn Sie unsicher sind, ob der Server den Verbindungstyp SSLv23 unterstützt, können Sie die Verbindung mehrmals öffnen. Schlägt der Verbindungsaufbau mit Fehler 102 fehl, versuchen Sie einen anderen Verbindungstyp.
<i>Secure</i>	<u>Boolean</u>	Setzen Sie das auf <u>True</u> , um eine SSL-Verbindung festzulegen. Falls <i>Secure</i> <u>False</u> ist, übermittelt der <i>SSLSocket</i> Daten genau so wie <u>TCPSocket</u> .
<i>SSLConnected</i>	<u>Boolean</u>	<u>True</u> , falls Sie eine SSL-Verbindung haben.
<i>SSLConnecting</i>	<u>Boolean</u>	<u>True</u> , falls REALbasic beim Ausführen eines Handshakes ist, um eine SSL-Verbindung aufzubauen.

Events

Name	Parameter	Beschreibung
<i>Connected</i>		Eine Verbindung wurde unter Verwendung der Address- und Port-Eigenschaften hergestellt.
<i>DataAvailable</i>		Im Puffer sind nun neue einkommende Daten verfügbar.
<i>Error</i>		Ein Fehler ist aufgetreten.
<i>SendComplete</i>		Alle Daten im Socket-Puffer wurden geschrieben (gesendet).

Hinweise

Um eine SSL-Verbindung aufzubauen, setzen Sie die *Secure*-Eigenschaft auf True und verwenden die *Connect*-Methode.

Derzeit ist es so, daß das Setzen der *Secure*-Eigenschaft auf True und ein anschließendes Aufrufen der *Listen*-Methode kein lauschenden Secure Socket erzeugt. Wenn Sie dies machen, erzeugen Sie einen nicht-sicheren lauschenden Port auf Ihrem Rechner auf dem angegebenen Port und alle eingehenden Verbindungen werden nicht sicher sein. Eine Funktion für sicheres Lauschen wird vielleicht in einer künftigen Version von REALbasic eingebaut.

Das **SSLSocket**-Steuerelement besitzt in der Steuerelemente-Palette kein eigenes Icon. Es gibt zwei Möglichkeiten, ein **SSLSocket**-Steuerelement einem Fenster hinzuzufügen:

- Draggen Sie ein TCPSocket-Steuerelement auf ein Fenster und ändern dann seine Super-Klasse in **SSLSocket**.
- Rufen Sie das Kontextmenü des Fensters durch ctrl-Klick (Macintosh) oder Rechtsklick (Windows) auf das Fenster auf und wählen Sie dann **Hinzufügen/SocketCore/TCPSocket/SSLSocket** aus.

Wenn Sie es aus der Steuerelemente-Palette draggen, erhalten Sie das Icon des TCPSocket-Steuerelements. Dieser Unterschied ist nur kosmetisch und beeinträchtigt die Funktionsweise des *SSLSocket* nicht.

Das **SSLSocket**-Steuerelement kann mit Code instanziiert werden, da es nicht von einem Control abgeleitet ist. Dadurch können Sie leicht Code schreiben, der Kommunikationen durchführt, ohne dem Fenster ein Steuerelement hinzuzufügen.

Das Schreiben in ein Socket ist asynchron. Das bedeutet, daß jedesmal, wenn die *Write*-Methode aufgerufen wird, landen die übergebenen Daten in einen Puffer im Speicher, bevor sie wirklich gesendet werden und werden danach aus dem Puffer entfernt. Nachdem der Socket das Senden der Daten aus dem Puffer an den Rechner am anderen Ende der Socket-Verbindung beendet hat, wird der *SendComplete*-Event-Handler ausgeführt. Dadurch wissen Sie, wann alle Daten wirklich gesendet wurden.

Das Aufrufen von *Read*, *ReadAll*, oder *Lookahead* erwischt eventuell nicht alle Daten im internen Puffer. Das liegt daran, daß SSL Daten in Blöcken lesen muß (wegen der Verschlüsselung) und sich eventuell kein kompletter Block im Puffer befindet. Es könnten z.B. 700 verfügbare Bytes im Puffer sein, SSL kann aber nur 512 Bytes entschlüsseln, da der Rest ein nicht kompletter Block ist. In die-

sem Fall verbleiben einige Daten wartend im Puffer. Wenn mehr Daten eintreffen, führt REALbasic ein DataAvailable-Event aus. Wenn keine weiteren DataAvailable-Events mehr verfügbar sind, führt REALbasic auch nach Verbindungstrennung zusätzliche DataAvailable-Events aus, damit Sie alle verbliebenen Daten aufsammeln können, die SSL zurückliefern kann. Es gibt zwei Dinge, auf die man deswegen achten sollte:

- 1) Wenn es nicht genug Daten zum entschlüsseln für SSL gibt, erhalten Sie eventuell ein DataAvailable-Event aber keine Daten.
- 2) Der Aufruf von SSLSocket.Close könnte DataAvailable-Events ausführen.

Siehe auch

[SocketCore](#), [TCPSocket](#), [Object](#) Klassen.

StackOverflowException Fehler

Tritt auf, wenn die Aufrufkette zu lang wird.

Übergeordnete Klasse

[RuntimeException](#)

Hinweise

Eine StackOverflowException tritt auf, wenn der Stack überläuft. Das passiert, wenn die Aufrufkette zu lang wird. Das kann leicht passieren, wenn Sie in Ihrem Code rekursive Aufrufe tätigen, ohne dafür Sorge zu tragen, die Rekursion zu beenden – oder die Bedingung für den Beendigung benötigt zu viele Aufrufe, bis sie eintritt.

Beispiel

Folgende Methode ruft sich selbst auf bis der Stack überläuft:

```
Function Square (value as Integer) as Integer
    Return Square(value)
End Function
```

Sie können die Funktion mit folgendem einfachen [Exception](#)-Handler verwalten:

```
Function Square (value as Integer) as Integer
    Return Square(value)
Exception err
    If err IsA StackOverflowException then
        MsgBox "Der Stack ist übergelaufen!"
    end if
End Function
```

Siehe auch

[RuntimeException](#) Klasse; [OutOfBoundsException](#), [IllegalCastException](#), [TypeMismatchException](#), [NilObjectException](#) Fehler; [Function](#), [Raise](#) Befehle; [Nil](#) Funktion, [Exception](#) Block.

StandardInputStream Klasse

Wird zum Lesen von Texteingabe in einer [Konsolen-Applikation](#) verwendet.

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
Read	Count als Integer , [enc als TextEncoding]	Liefert einen String . Liest countbytes vom Stream. Der optionale Parameter enc gibt die Kodierung des Strings an.
ReadLine		Liefert einen String . Liest vom Stream, bis der Benutzer Return oder Enter drückt. ReadLine liefert den String ohne Zeilenumbruch mit zurück.

Hinweise

Das es nur einen **StandardInputStream** für das System gibt, muß kein **StandardInputStream**-Objekt erzeugt werden. Benutzen Sie einfach [StdIn](#).

Die **StandardInputStream**-Klasse implementiert das „Readable“-Klassen-Interface. Wenn Sie dieses Interface selbst implementieren, müssen Sie eine Methode anbieten, die der obigen entspricht.

Siehe auch

[ConsoleApplication](#), [StandardOutputStream](#) Klassen; [Print](#), [Input](#), [StdErr](#), [StdIn](#), [StdOut](#) Methoden; [TargetHasGUI](#) Konstante.

StandardOutputStream Klasse

Wird bei Konsolen-Applikationen für die Ausgabe (stdout) und auch Fehlerausgabe(stderr) verwendet.

Übergeordnete Klasse

[Object](#)

Methoden

Name	Parameter	Beschreibung
Write	data als String	Schreibt <i>data</i> in den Stream.
WriteLine	data als String	Schreibt data in den Stream. WriteLine fügt einen Zeilenumbruch an.

Hinweise

Da es nur ein **StandardOutputStream** für das System gibt, besteht keine Notwendigkeit, ein StandardOutputStream-Objekt zu erzeugen. Verwenden Sie einfach StdOut oder StdErr für Fehlerausgabe (Error stream).

Die **StandardOutputStream**-Klasse implementiert das „Writable“-Klassen-Interface. Wenn Sie dieses implementieren wollen, müssen Sie eine Methode anbieten, die der obigen entspricht.

Siehe auch

[ConsoleApplication](#), [StandardInputStream](#), [Print](#), [Input](#), [StdErr](#), [StdIn](#), [StdOut](#) Methoden; [TargetHasGUI](#) Konstante.

StandardToolBarItem Steuerelement



Wird zum Erzeugen einer Toolbar in einem Fenster verwendet. Wird nur unter Mac OS X 10.2 und neuer unterstützt. StandardToolBarItems können für andere Plattformen nur für Layoutzwecke im Fenster platziert werden.

Übergeordnete Klasse

[Control](#)

Eigenschaften

Name	Typ	Beschreibung
<i>ControlOrder</i>	Integer	Legt die Links-nach-Rechts-Plazierung des Objekts in der Toolbar fest.
<i>Enabled</i>	Boolean	True , falls das Objekt aktiviert ist.
<i>Handle</i>	Integer	Eine in Declare -Befehlen verwendete ToolbarItemRef, nur für Toolbox-Programmierer von Interesse.
<i>IdentifierOther</i>	String	Ein ToolbarItem-Identifizier, den REALbasic nicht anbietet (Eine andere Art, einen Separator zu erzeugen, ist z.B., diese Eigenschaft auf “com.apple.hitoolbox.toolbar.separator” zu setzen).
<i>Left</i>	Integer	Der Abstand (in Pixeln) von der linken Seite des Fensters im Fenster-Editor. Im selbstständig laufenden Programm basiert die Reihenfolge der Toolbar-Objekte strikt auf der von links nach rechts Reihenfolge.
<i>Top</i>	Integer	Der Abstand (in Pixeln) oberen Rand des Fensters im Fenster-Editor. Im selbstständig laufenden Programm basiert die Reihenfolge der Toolbar-Objekte strikt auf der von links nach rechts Reihenfolge.

Name	Typ	Beschreibung
<i>Type</i>	<u>Integer</u>	Type - Einer der von Apple vordefinierten Toolbar-Typen oder ein benutzerdefinierter Toolbar-Typ: 0 - Anderer (Sie müssen IdentifierOther verwenden, um den zu verwendenden Identifier festzulegen) 1 - Separator (rein kosmetische Vorrichtung) 2 - Leerraum, trennt Objekte durch einen kleinen Zwischenraum 3 - Flexibler Leerraum, Objekte hinter diesem werden rechtsbündig ausgerichtet

Events

Name	Parameter	Beschreibung
<i>Close</i>		Tritt auf, wenn ein Objekt zerstört wird.
<i>Open</i>		Tritt auf, wenn das Objekt anfangs erzeugt wird.

Hinweis

Wenn sich mindestens ein [ToolbarItem](#) oder **StandardToolbarItem** in einem Fenster befindet, wird ein Toolbar erzeugt und der Toolbar-Knopf wird dem Fenster hinzugefügt. Jedes ToolbarItem wird in der ControlOrder von links nach rechts hinzugefügt.

Siehe auch

[ToolbarItem](#) Steuerelement.

StartupItemsFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Ordner „Startobjekte“ im Systemordner von Mac OS 9 und Windows zu bekommen.

Syntax

Ergebnis = **StartupItemsFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>FolderItem</u>	Ein <u>FolderItem</u> das den Ordner repräsentiert

Hinweis

Verwenden Sie die **StartupItemsFolder**-Funktion, wenn Sie auf Objekte aus dem Startobjekte-Ordner zugreifen müssen.

Objekte im Ordner „Startobjekte“ werden automatisch geöffnet, wenn der Anwender seinen Macintosh startet. Wenn Sie dem Anwender die Möglichkeit geben wollen, Ihre Anwendung automatisch beim Systemstart zu öffnen, müssen Sie die Anwendung (oder einen Alias davon) in den Ordner „Startobjekte“ legen.

Die Funktion **StartupItemsFolder** bietet eine Möglichkeit, auf die Einträge im Ordner „Startobjekte“ zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

Unter Windows greift **StartupItemsFolder** auf das Verzeichnis des Anwenders Programme\Autostart zu. Unter Linux liefert diese Funktion [Nil](#) zurück.

Beispiele

Bei der Beschreibung der Funktion [AppleMenuFolder](#) finden Sie ein passendes Beispiel.

Siehe auch

[AppleMenuFolder](#), [ApplicationSupportFolder](#), [ControlPanelsFolder](#), [DesktopFolder](#), [ExtensionsFolder](#), [FontsFolder](#), [PreferencesFolder](#), [StartupItemsFolder](#), [SystemFolder](#), [TemporaryFolder](#), [TrashFolder](#) Funktionen.

StaticText Steuerelement



Zeigt vom Anwender nicht bearbeitbaren Text an.

Übergeordnete Klasse

[RectControl](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Eigenschaften

Name	Art	Beschreibung
<i>Bold</i>	Boolean	Stellt den Text fett dar.
<i>Caption</i>	String	Der darzustellende Text. Ist mit der Text-Eigenschaft identisch. Diese Eigenschaft wurde zur Kompatibilität mit Visual Basic hinzugefügt.
<i>DataField</i>	String	Nur relevant, wenn das StaticText in Verbindung mit einem DataControl verwendet wird, um den Inhalt von Feldern einer Datenbanktabelle anzuzeigen. Der Name eines Feldes in der Tabelle, auf die das DataControl verweist.
<i>Data-Source</i>	DataControl	Das DataControl , das auf eine Tabelle in einer Datenbank verweist, dessen Feld mittels eines StaticText dargestellt wird (TableName als String). Wenn es innerhalb des Codes zugewiesen wird, verwenden Sie die Name-Eigenschaft eines DataControl. Verwenden Sie die DataField-Eigenschaft, um jedes anzuzeigende Feld mit einem StaticText zu verbinden. In der IDE erscheint ein Popup-Menü mit den Feldnamen. Database-Felder können auch mit Hilfe von EditFields , PopupMenu s und ListBoxes angezeigt werden.
<i>Italic</i>	Boolean	Stellt den Text kursiv dar.
<i>Multiline</i>	Boolean	Wenn True , so wird Text über mehrere Zeilen hinweg umbrochen.
<i>Text</i>	String	Der darzustellende Text.
<i>TextAlign</i>	Integer	Die Ausrichtung des Textes (0=links, 1=mittig, 2=rechts)
<i>TextColor</i>	Color	Die Textfarbe.
<i>TextFont</i>	String	Die Schriftart des darzustellenden Textes.
<i>TextSize</i>	Integer	Die Schriftgröße des darzustellenden Textes.
<i>Underline</i>	Boolean	Stellt den Text unterstrichen dar.

Events

Name	Parameter	Beschreibung
<i>Accelerator-Key</i>		Das Tastenkürzel für das StaticText wurde gedrückt.
<i>MouseDown</i>	x als Integer , y als Integer	Die Maustaste wurde innerhalb der Region des StaticText an der gelieferten Stelle x,y gedrückt. Wenn Sie MouseDown auswerten möchten, liefern Sie True zurück.
<i>MouseDown</i>	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des StaticText -Steuerelements gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
<i>MouseUp</i>	x als Integer , y als Integer	Die Maustaste wurde innerhalb der Region des StaticText an der gelieferten Stelle x,y losgelassen. Dieses Event tritt nicht auf, bis Sie nicht True im MouseDown-Event liefern.

Hinweise

Unter Windows können Sie ein Tastenkürzel für dieses Steuerelement definieren. In der Caption-Eigenschaft fügen Sie vor das Kürzel das Kaufmanns-Und-Zeichen ein. Die Beschriftung „&Home“ zum Beispiel legt das „H“ als Kürzel fest. Wenn Sie unter Windows alt+H drücken, wird das Action-Event ausgelöst.

Siehe auch

[RectControl](#) Klasse; [EditField](#) Steuerelement.

StdErr Methode

Hiermit erhalten Sie eine globale Referenz auf die Standard-Fehler-Ausgabe (stderr). Dies ist nur für Konsolen-Applikation verfügbar.

Hinweise

Die **StdErr**-Methode implementiert das „Writable“-Klassen-Interface. Wenn Sie dies selbst implementieren, müssen Sie eine Write-Methode anbieten, die der folgenden entspricht:

Write(data)

Teil	Typ	Beschreibung
<i>data</i>	<u>String</u>	Der Text, der geschrieben werden soll.

Das Writable-Klassen-Interface hat folgende Methoden:

Name	Parameter	Beschreibung
Flush		Leert den internen Puffer schreibt alle Daten auf das Volume. Dann wird der Puffer gelöscht.
WriteError		Liefert <u>True</u> wenn ein Fehler auftrat.

Für weitere Informationen über Klassen-Interfaces lesen Sie bitte das Kapitel „Klassen-Interfaces“ im Benutzerhandbuch durch.

Syntax

Ergebnis=StdOut

Teil	Typ	Beschreibung
<i>Ergebnis</i>	<u>StandardOutputStream</u>	Der Standard-Fehler-Stream einer Konsolen-Applikation und ist vom Typ <u>StandardOutputStream</u> .

Siehe auch

ConsoleApplication, StandardInputStream, StandardOutputStream Klassen; TargetHasGUI Konstante; Print, Input, StdIn, StdErr Methoden.

StdIn Methode

Liefert eine Referenz auf den globalen Standard-Eingabe-Stream. Wird in Applikationen vom Typ ConsoleApplications verwendet.

Syntax

result=StdIn

Teil	Typ	Beschreibung
<i>result</i>	<u>StandardInputStream</u>	Der Standard-Eingabe-Stream wird typischerweise für Tastatureingaben in einer Konsolen-Applikation verwendet. Es kann aber auch ein serieller Port, ein Socket oder ein anderes Eingabegerät sein.

Hinweise

Die **Stdin**-Methode implementiert das „Readable“-Klassen-Interface. Wenn Sie dies selbst implementieren, müssen Sie eine Read-Methode anbieten, die der folgenden entspricht.

Ergebnis=Read (count, [enc])

Teil	Typ	Beschreibung
Ergebnis	<u>String</u>	Der Text, der zurückgegeben wird.
count	<u>Integer</u>	Anzahl der Bytes, die gelesen wurden.
enc	<u>TextEncoding</u>	Optionaler Parameter, der die Kodierung des Strings angibt.

Das Readable-Klassen-Interface hat zwei Methoden.

Name	Parameter	Beschreibung
EOF		Liefert <u>True</u> , wenn der Stream zu Ende ist.

Name	Parameter	Beschreibung
ReadError		Liefert <u>True</u> wenn ein Fehler aufgetreten ist.

Für weitere Informationen über Klassen-Interfaces lesen Sie bitte das Kapitel „Klassen-Interfaces“ im Benutzerhandbuch durch.

Siehe auch

[ConsoleApplication](#), [StandardInputStream](#), [StandardOutputStream](#) Klassen; [TargetHasGUI](#) Konstante; [Print](#), [Input](#), [StdOut](#), [StdErr](#) Methoden.

StdOut Methode

Liefert eine Referenz auf den globalen Standard-Ausgabe-Stream. Wird in Applikationen vom Typ [ConsoleApplications](#) verwendet und ist typischerweise ein Terminal-Fenster.

Syntax

Ergebnis=StdOut

Teil	Typ	Beschreibung
Ergebnis	StandardOutputStream	Der Standard-Output-Stream ist normalerweise die Terminal-Applikation.

Hinweise

Die **StdErr**-Methode implementiert das „Writable“-Klassen-Interface. Wenn Sie dies selbst implementieren, müssen Sie eine Write-Methode anbieten, die der folgenden entspricht:

Write(data)

Teil	Typ	Beschreibung
data	String	Der Text, der geschrieben werden soll.

Das Writable-Klassen-Interface hat folgende Methoden:

Name	Parameter	Beschreibung
Flush		Gibt alle verbliebenen Daten aus und löscht den Puffer.
WriteError		Liefert <u>True</u> wenn ein Fehler auftrat.

Für weitere Informationen über Klassen-Interfaces lesen Sie bitte das Kapitel „Klassen-Interfaces“ im Benutzerhandbuch durch.

Siehe auch

[ConsoleApplication](#), [StandardInputStream](#), [StandardOutputStream](#) Klassen; [TargetHasGUI](#) Konstante; [Print](#), [Input](#), [StdIn](#), [StdErr](#) Methoden.

Str Funktion

Liefert den übergebenen Wert als [String](#) zurück.

Syntax

Ergebnis=Str(Zahl)

Teil	Art	Beschreibung
Ergebnis	String	Die String-Variante der übergebenen Zahl.
Zahl	Integer , Single oder Double	Ein beliebiger numerischer Ausdruck

Hinweis

Verwenden Sie die Funktion [Format](#), wenn Sie numerische Werte in formatierte Strings wie Datum, Uhrzeit oder Währung umwandeln möchten.

Str liefert sieben (7) signifikante Stellen. Es verwendet die wissenschaftliche Notation, wenn es mehr als 6 Stellen zurückgeben muß. **Str**(123456789) liefert 1.234568e+8.

Str nimmt an, daß der Punkt (.) Dezimaltrennzeichen ist. Wenn Ihre Anwendung andere Dezimaltrennzeichen benötigt, sollten Sie die Funktion CStr verwenden.

Beispiele

Dieses Beispiel verwendet die Funktion **Str**, um verschiedene Zahlen in einen String umzuwandeln.

```
Dim s As String
s=Str(123)           //liefert "123"
s=Str(-123,44)       //liefert "-123,44"
s=Str(123.0045)       //liefert "123.0045"
Const Pi=3.14159265358979323846264338327950
s= Str(pi)           // liefert "3.141593"
```

Siehe auch

CDbl, CStr, Val, Format Funktionen.

StrComp Funktion

Führt einen Binärvergleich (case sensitive) oder Textvergleich (lexikographisch) von zwei Strings durch.

Syntax

Ergebnis = **StrComp**(string1, string2, modus)

ODER

Ergebnis = **StringVariable.StrComp**(string2, modus)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Wenn <i>string1</i> < <i>string2</i> liefert die Funktion -1 Wenn <i>string1</i> = <i>string2</i> liefert die Funktion 0 Wenn <i>string1</i> > <i>string2</i> liefert die Funktion 1
<i>string1</i>	<u>String</u>	Der erste Vergleichsstring.
<i>string2</i>	<u>String</u>	Der zweite Vergleichsstring.
<i>modus</i>	<u>Integer</u>	Der Vergleichsmodus. 0=binär (case sensitive) und 1=Text (lexikographisch)
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ <u>String</u> .

Hinweis

Der Textvergleich arbeitet lexikographisch, nicht case insensitive. Dies bedeutet, daß die Groß-/Kleinschreibung genau dann bewertet wird, wenn die Strings selbst gleich sind.

Wenn Sie den Modus 0 (Binär) verwenden, vergleicht **StrComp** immer die Bytes des Strings, wie sie sind und nimmt keine Encoding-Konvertierung vor. Es ist denkbar, daß zwei Strings auf dem Bildschirm gleich aussehen, im Modus 0 von **StrComp** aber als unterschiedlich gemeldet werden, falls sie in unterschiedlichen Kodierungen vorliegen (und daher unterschiedliche Bytes enthalten).

Beispiele

Das folgende Beispiel liefert -1, da die beiden Strings bis auf die Groß/Kleinschreibung gleich sind.

```
StrComp("Spam", "spam", 1)
```

Das folgende Beispiel liefert -1, da beim Textvergleich der beiden Strings string1 größer ist als string2. Der ASCII-Wert von „s“ ist größer als der ASCII-Wert von „S“.

```
StrComp("Spam", "spam", 0)
```

Siehe auch

InStr Funktion; $\leq$, $=$, $\geq$, $\leq$, $\geq$, $\leq$ Operatoren.

String Datentyp

Ein **String** ist ein wesentlicher Datentyp in REALbasic. Er ist eine Serie von numerischen oder alphabetischen Zeichen, in Gänsefüßchen eingeschlossen. Jede Art von alphabetischer oder numerischer Information kann als ein String gespeichert werden. "Jean Marie", "17.3.98", "45.90" sind alles Beispiele von Strings. Wenn Sie Gänsefüßchen um Information in Ihrem Code platzieren, teilen Sie

REALbasic mit, diese Daten als eine Serie von Zeichen und sonst nichts anderes zu betrachten. Die Maximallänge eines Strings ist nur durch den verfügbaren Speicher begrenzt.

REALbasic beinhaltet eine umfangreiche Bibliothek von Funktionen, um Strings zu vergleichen und zu manipulieren. Diese finden Sie im Abschnitt „Siehe auch“ aufgelistet.

Die VarType-Funktion liefert als Wert 8, wenn ein **String** übergeben wird.

Hinweise

Alle Computer verwenden eine Kodierung, um Zeichenketten als eine Reihe von Bytes zu speichern. Die älteste und bekannteste ist die ASCII-Kodierung. Diese wird im Anhang A der PDF- oder gedruckten Version des Benutzerhandbuchs näher beschrieben. Sie definiert, daß Zeichen in einem Wertebereich von 0 bis 127 liegen. Dieser Bereich schließt Groß- und Klein-Buchstaben ein, sowie Ziffern, einige Symbole und nicht druckbare Zeichen, die in frühen Computern verwendet wurden.

Viele Erweiterungen wurden für ASCII eingeführt, um zusätzliche Symbole, Zeichen mit Akzent, nicht römische Zeichen und so weiter, zu unterstützen. Die Unicode-Kodierung wurde entwickelt, um nicht nur alle Sprachen abzudecken, sondern auch das Mischen von verschiedenen Sprachen in einer Zeichenkette zu ermöglichen. REALbasic unterstützt zwei verschiedene Unicode-Kodierungen, UTF-8 und UTF-16. Alle eingegebenen Konstanten, Strings und so weiter sind UTF-8 kodiert.

Wenn Ihre Strings mit REALbasic erzeugt, gespeichert und wieder gelesen wurden, brauchen Sie sich keine Gedanken über die Kodierung machen, da REALbasic die Kodierung immer mit speichert. Dies gilt auch für die REALdatenbank.

Da ASCII-Zeichen mit einem Wert über 127 unterschiedliche Zeichen repräsentieren, die von der Kodierung abhängig sind, ist es wichtig, daß Sie die Kodierung für alle Strings angeben, die außerhalb REALbasic erzeugt werden. Wenn Sie einen String über die TextInputStream-Klasse einlesen, setzen Sie die Encoding-Eigenschaft auf die korrekte Kodierung oder den optionalen Encoding-Parameter der Read-, ReadLine- oder ReadAll-Methode. Sie ermitteln die Kodierung eines Strings mit der Encoding-Funktion.

Wenn Sie einen String mit einer bestimmten Kodierung speichern möchten, können Sie die ConvertEncoding-Funktion benutzen. Geben Sie die gewünschte Kodierung als ein Encodings-Objekt an.

Wenn Sie ein Zeichen benötigen, das einen bestimmten Wert hat, können Sie dies mit der Funktion Chr erhalten. Dies funktioniert aber nur mit der ASCII-Kodierung. In allen anderen Fällen können Sie die Chr-Methode der TextEncoding-Klasse verwenden. Dabei können Sie, zusätzlich zum Wert, auch eine gewünschte Kodierung angeben.

REALbasic besitzt eine Vielzahl von Funktionen für die String-Manipulation und -Vergleiche, die unter „Siehe auch“ zu finden sind.

Die Funktion VarType liefert 8, wenn ein **String** übergeben wurde.

Beispiel

```
Dim s as String
d="Was tun, sprach Zeus"
```

Siehe auch

+, -, <=, <, >, =, >=, Asc, AscB, Chr, ChrB, CStr, InStr, InStrB, Left, LeftB, Len, LenB, Lowercase, LTrim, Mid, MidB, NthField, Replace, ReplaceAll, Right, RightB, RTrim, Str, StrComp, Titlecase, Trim, Uppercase, Val, VarType, IsNumeric Funktionen; Dim Befehl; Encodings Objekt; Encoding, ConvertEncoding Funktionen; TextEncoding, TextInputStream Klassen; Single, Double, Integer, Boolean, Color Datentypen.

StringProvider Interface Class

Wird verwendet, um benutzerdefinierte Objektverknüpfungen zu programmieren.

Methoden

Name	Typ	Beschreibung

<u>getString</u>	<u>String</u>	Ermittelt den aktuellen Wert von String.

<u>setString</u>	<u>String</u>	Setzt den Inhalt von String.

StringShape Klasse

Zeichnet einen Textstring in einer Vektorgrafikumgebung.

Übergeordnete Klasse

[Object2D](#)

Eigenschaften

Name	Typ	Beschreibung
Bold	Boolean	Setzen Sie es auf True , um den Schriftstil Fett zu verwenden.
Italic	Boolean	Setzen Sie es auf True , um den Schriftstil Kursiv zu verwenden.
Text	String	Der zu zeichnende Text. Der Text kann nur aus einer Zeile bestehen und nur einen Textstil (Font, Fontgröße, Stil) besitzen, der durch die anderen StringShape -Eigenschaften festgelegt wird.
TextFont	String	Name des zu verwendenden Fonts.
TextSize	Double	Fontgröße in Punkten.
Underline	Boolean	Setzen Sie es auf True , um den Schriftstil Unterstrichen zu verwenden.

Hinweise

Die Eigenschaften X,Y legen den Mittelpunkt der Grundlinie des Strings fest. Strings, die Zeilenumbrüche enthalten, werden nicht unterstützt. **StringShapes** können rotiert werden, dies ist jedoch sehr speicherintensiv, vor allem bei großen Strings.

Beispiele

Dieses Beispiel zeichnet eine Textzeile, wenn der Anwender die Maustaste drückt. Schreiben Sie folgenden Code in das [MouseDown](#)-Event eines [Fensters](#).

```
Dim s as StringShape
s=New StringShape
s.Text="Hallo Welt"
s.TextFont="Helvetica"
s.Bold=True
Graphics.DrawObject s,x,y
```

Siehe auch

[ArcShape](#), [CurveShape](#), [FigureShape](#), [FolderItem](#), [Group2D](#), [Graphics](#), [Object2D](#), [OvalShape](#), [Picture](#), [PixmapShape](#), [RectShape](#), [RoundRectShape](#) Klassen.

StyledText Klasse

Wird verwendet, um einen Stil-Text unabhängig von einem [EditField](#) zu bearbeiten. Für ein [EditField](#) können Sie die Eigenschaften und Methoden dieser Klasse über die [StyleText](#)-Eigenschaft erreichen.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
Text	String	Der Text, dessen Stilattribut über diese Instanz der Klasse verwaltet werden. Die <i>Start</i> - und <i>Index</i> -Parameter der Methoden dieser Klasse sind Null-basiert.

Methoden

Name	Parameter	Beschreibung
AppendStyleRun	Run als StyleRun	Hängt ein StyleRun -Objekt and das Ende des Texts an.
Bold	Start als Integer Length als Integer	Liest oder setzt den Stil des ausgewählten Texts auf Fett. <i>Start</i> ist die Startposition im Text, beginnend bei Null. Voreingestellt ist 1. Die Methode liefert ein Boolean zurück. Verwenden Sie eine Zuweisung zum Setzen der Eigenschaft.
Font	Start als Integer Length als Integer	Liest oder setzt die Schriftart des ausgewählten Texts. <i>Start</i> ist die Startposition im Text. Voreingestellt ist dies 1. Diese Methode liefert einen String . Verwenden Sie eine Zuweisung zum Setzen der Eigenschaft.

Name	Parameter	Beschreibung
InsertStyleRun	Run als <u>StyleRun</u> Index als <u>Integer</u>	Fügt ein <u>StyleRun</u> -Objekt an die angegebene Position ein. <i>Index</i> ist Null-basiert. Sie erhalten die gesamte Anzahl von StyleRuns im <i>Text</i> über die <u>StyleRunCount</u> -Methode.
Italic	Start als <u>Integer</u> Length als <u>Integer</u>	Liest oder setzt den Stil des ausgewählten Texts auf Italic. <i>Start</i> ist die Startposition im Text. Voreingestellt ist dies 1. Diese Methode liefert ein <u>Boolean</u> . Verwenden Sie eine Zuweisung, wenn Sie diese Eigenschaft setzen wollen.
Paragraph	Index als <u>Integer</u>	Liefert einen Absatz als Objekt vom Typ <u>Paragraph</u> . Der Absatz wird über den Index angegeben, wobei der Index Null-basiert ist. Die Eigenschaften der <u>Paragraph</u> -Klasse geben Ihnen Möglichkeit, die Position und Textausrichtung zu ändern.
ParagraphCount		Liefert ein <u>Integer</u> mit der Anzahl der Absätze (<u>Paragraph</u> -Objekte) im Text.
RemoveStyleRun	Index als <u>Integer</u>	Entfernt ein <u>StyleRun</u> -Objekt. Index ist Null-basiert. Sie ermitteln die gesamte Anzahl der StyleRun-Objekt im Text über die <u>StyleRunCount</u> -Methode.
SetParagraphAlignment	Index als <u>Integer</u> Alignment als <u>Integer</u>	Setzt die Textausrichtung eines Absatzes, der mittels Index spezifiziert wird. Index ist Null-basiert. Die möglichen Werte der Textausrichtung sind: 0 – linksbündig 1 – zentriert 2 – rechtsbündig (unter Linux nicht verfügbar)
Size	Start als <u>Integer</u> Length als <u>Integer</u>	Liest oder setzt die Schriftgröße des ausgewählten Texts. Start ist die Startposition im Text. Voreingestellt ist dies 1. Diese Methode liefert ein <u>Integer</u> . Verwenden Sie eine Zuweisung um diese Eigenschaft zu setzen.
StyleRun	Index als <u>Integer</u>	Liefert ein <u>StyleRun</u> -Objekt, das mittels Index spezifiziert ist. Index ist Null-basiert.
StyleRunCount		Liefert ein <u>Integer</u> mit der Anzahl der StyleRun-Objekte im Text. Verwenden Sie dies in Verbindung mit der StyleRun- und StyleRunRange-Methode, um weitere Informationen über die einzelnen StyleRun-Objekte zu erhalten.
StyleRunRange	Index als <u>Integer</u>	Liefert ein <u>Range</u> -Objekt für das angegebene <u>StyleRun</u> -Objekt. Index ist Null-basiert. Die Eigenschaften der <u>Range</u> -Klasse geben Ihnen Zugriff auf die Start- und Endposition eines <u>StyleRun</u> -Objekts sowie dessen Länge.
TextColor	Start als <u>Integer</u> Length als <u>Integer</u>	Liest oder setzt die Textfarbe des ausgewählten Texts. <i>Start</i> ist die Startposition im Text. Voreingestellt ist dies 1. Diese Methode liefert ein <u>Color</u> -Objekt. Verwenden Sie eine Zuweisung um diese Eigenschaft zu setzen.
Underline	Start als <u>Integer</u> Length als <u>Integer</u>	Liest oder setzt den Stil des ausgewählten Texts auf Unterstreichen. <i>Start</i> ist die Startposition im Text. Voreingestellt ist dies 1. Diese Methode liefert ein <u>Boolean</u> . Verwenden Sie eine Zuweisung, wenn Sie diese Eigenschaft setzen wollen.

Hinweise

Die **StyledText**-Klasse gibt Ihnen die Möglichkeit, Stilattribute unabhängig von einem EditField und dessen Eigenschaften zu verwenden. Zusätzlich können Sie über die **StyledText**-Klasse die Textausrichtung (linksbündig, zentriert, rechtsbündig) für einen

Absatz bestimmen. Die Eigenschaften und Methoden der **StyledText**-Klasse sind auch über die **StyledText**-Eigenschaft eines **EditField**-Objekts erreichbar, aber Sie können mit dieser Klasse auch Text bearbeiten, der nicht in einem Steuerelement angezeigt wird.

Die Methoden der **StyledText**-Klasse führen zu einem Update des Inhalts eines **EditFields**. Das bedeutet, wenn Sie bei einem **StyleRun**-Objekt eines Texts die Attribute ändern, werden diese Änderungen nicht im **EditField** angezeigt. Wenn Sie also Änderungen in einem **StyleRun**-Objekt vornehmen wollen, müssen Sie ein neues **StyleRun**-Objekt erzeugen und das alte ersetzen. Damit ändern Sie das **StyledText**-Objekt, dessen Änderungen sofort im **EditField** angezeigt werden. Einfacher geht dies natürlich über die Methoden des **StyledText**-Objekts statt der **StyleRun**-Eigenschaften.

Die Stilattribute, die ein **EditField** für markierten Text hat, werden von der **StyledText**-Klasse berücksichtigt. Das heißt, wenn Sie diesen Stil für markierten Text ändern und dann ein **StyleRun**-Objekt ermitteln, sind diese Informationen enthalten (und umgekehrt). Nicht alle Stile der **EditField**-Klasse für markierten Text sind in der **StyleRun**-Klasse verfügbar, wie zum Beispiel Outline, „komprimiert“, Schatten, gesperrt und ohne Stil. Die ersten vier Stile sind nur unter Mac OS 8-9 verfügbar.

Beispiel

Das folgende Beispiel verwendet die Methoden der **StyledText**-Klasse, um Text zu markieren, ihn auszurichten und in einem **EditField** anzuzeigen.

```
Dim Text as String // text, der im EditField angezeigt werden soll
Dim st,ln as Integer // Start- und Länge-Werte eines Absatzes
//Legt vier Text-Absätze fest
Text = "This is the text that we are going to save " _
      +"into our file from the EditField."+EndOfLine _
      +"Isn't that interesting?"&EndOfLine _
      +"Man, I sure do love using RB to take care of my projects."&EndOfLine _
      +"That's because the RS Engineering staff is just so awesome!"
EditField1.StyledText.Text = text //vier Text-Absätze

EditField1.StyledText.Bold(5, 2) = True
EditField1.StyledText.TextColor(5, 2) = &cFF0000 //fett und rot

EditField1.StyledText.Size( 7, 10) = 16
EditField1.StyledText.Underline(12, 4) = True //16 Punkt unterstrichen

EditField1.StyledText.Size(100, 4) = 18
EditField1.StyledText.TextColor(100, 4) = &cFF00FF //18 Punkt und Magenta

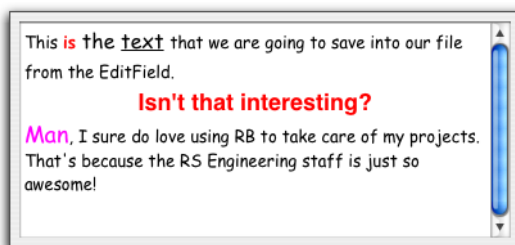
EditField1.StyledText.Font(0, Len(text) ) = "Comic Sans MS"

EditField1.StyledText.SetParagraphAlignment(1,1) //Zweiten Absatz zentrieren

//Diesen Absatz auf Helveticita, 18 Punkt, fett, rot setzen
//als erstes die Start- und Länge-Werte für den Absatz ermitteln...
st=EditField1.StyledText.Paragraph(1).StartPos-1
ln=EditField1.StyledText.Paragraph(1).Length

//als nächstes die Attribute zuweisen...
EditField1.StyledText.Bold(st,ln)=True
EditField1.StyledText.Font(st,ln)="Helvetica"
EditField1.StyledText.Size(st,ln)=18
EditField1.StyledText.TextColor(st,ln)=&cFF0000
```

Im Ergebnis sieht das **EditField**-Steuerelement so aus:



Siehe auch

[StyleRun](#), [Paragraph](#), [Range](#), [EditField](#) Klassen.

StyledTextPrinter Klasse

Wird verwendet, um Styled Text, der in einem [EditField](#) gespeichert ist, zu drucken.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Eigenschaft	Typ	Beschreibung
EOF	Boolean	True, wenn im EditField kein Text mehr zu drucken ist. Verwenden Sie dies, um festzustellen, wann der Aufruf von DrawBlock gestoppt werden soll. Siehe Beispiel.
<i>Width</i>	Integer	Breite des Textes, der gedruckt werden soll.

Methoden

Methode	Parameter	Beschreibung
<i>DrawBlock</i>	X als Integer , Y als Integer , Height als Integer	Wird verwendet, um Styled Text zu drucken. x und y sind die Koordinaten des Punktes, an dem der Ausdruck des Textes beginnen soll. X und Y sind der Abstand von der linken oberen Ecke des druckbaren Bereichs, wie im Seitenformat definiert.

Beispiele

Folgendes Beispiel druckt den Styled Text eines EditFields (die MultiLine-Eigenschaft des EditFields muß auf True gesetzt sein, um Styled Text zu akzeptieren und zu drucken):

```
Dim stp as StyledTextPrinter
Dim g as Graphics
g=OpenPrinterDialog()
If g <> Nil then
    stp=EditField1.StyledTextPrinter(g,72*7.5)
    stp.DrawBlock 0,0,72*10
End if
```

Dieses Beispiel druckt den Inhalt von EditField1 in zwei Spalten mit einem viertel Zoll Zwischenraum. Die EOF-Eigenschaft wird verwendet, um festzustellen, wann der gesamte Text gedruckt wurde. Wenn EOF False ist, wird die zweite Spalte oder eine weitere Seite gedruckt.

```
Dim g as Graphics
Dim stp as StyledTextPrinter
Dim ColumnWidth, SpaceBetweenColumns, PageHeight as Integer
Dim ColumnToPrint as Integer

ColumnWidth = 261
//7.5 Zoll minus 1/4 Zoll für Zwischenraum
//geteilt durch 2
SpaceBetweenColumns = 18
//18 pixels is 1/4 inch
PageHeight = 10 * 72
//10 Zoll * 72 Pixel pro Zoll

g=OpenPrinterDialog()
If g <> Nil then
    stp=EditField1.StyledTextPrinter(g,540)
    stp.width = ColumnWidth

    ColumnToPrint = 1
    Do until stp.eof
        stp.DrawBlock (columnWidth+SpaceBetweenColumns)*(columnToPrint-1),0,PageHeight
        If ColumnToPrint = 2 then //drucke letzte Spalte
            If Not stp.eof then // weiterer Text ist zu drucken
                g.nextPage
                ColumnToPrint = 1
```

```

        End if
    Else // weitere Spalten sind auf dieser Seite zu drucken
        ColumnToPrint = columnToPrint + 1
    End if
Loop
End if

```

Siehe auch

[EditField](#) Steuerelement; [OpenPrinter](#), [OpenPrinterDialog](#) Funktionen; [Graphics](#), [PrinterSetup](#) Klassen.

StyleRun Klasse

Wird verwendet, um Stilattribute in einem [StyledText](#)-Objekt zu organisieren und zu bearbeiten.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
Bold	Boolean	Auslesen oder Setzen des Stils.
Font	String	Auslesen oder Setzen der Schriftart.
Italic	Boolean	Auslesen oder Setzen des Stils.
Size	Integer	Auslesen oder Setzen der Schriftgröße.
Text	String	Auslesen oder Setzen des Texts eines StyleRun-Objekts.
TextColor	Color	Auslesen oder Setzen der Farbe des Texts.
Underline	Boolean	Auslesen oder Setzen des Stils.

Hinweise

Ein **StyleRun** ist eine Folge von hintereinanderliegenden Zeichen, die die gleichen Stilattribute besitzen, welche über die Eigenschaften des **StyleRun**-Objekts spezifiziert sind. Die Text-Eigenschaft eines [StyledText](#)-Objekts ist eine Reihe von StyleRun-Objekten. Sie ermitteln die gesamte Anzahl von StyleRun-Objekten über die StyleRunCount-Methode der [StyledText](#)-Klasse und referenzieren ein **StyleRun** über die StyleRun-Methode. Sie können die Start- und Endposition eines StyleRun-Objekts über die Methode StyleRunRange ermitteln und haben somit auch die Länge. Die [Range](#)-Klasse bietet Ihnen Zugriff auf die Start-, Endposition und Länge des **StyleRun**-Objekts.

Wenn der Stiltext in einem [EditField](#) angezeigt wird, werden Änderungen an einem StyleRun-Objekt nicht angezeigt. Dazu müssen Sie ein neues StyleRun-Objekt erzeugen und das alte ersetzen. Einfacher geht dies über die [StyledText](#)-Methoden statt der Eigenschaften eines **StyleRun**-Objekts.

Beispiel

Im folgenden Beispiel wird der Stiltext eines [EditField](#) als eine Folge von StyleRun-Objekten gespeichert, die aneinandergehängt sind.

```

Dim out as BinaryStream
out = DesktopFolder.Child("TestFile").CreateBinaryFile("")

//Wenn die Datei nicht erzeugt werden konnte, dann abbrechen
If out = Nil then return

//Nun werden alle StyleRuns abgearbeitet
//und in die Datei ausgegeben.
Dim mb as MemoryBlock
Dim sr as StyleRun
Dim i, Count as Integer
Dim TextLen, FontLen as Integer
Dim StaticInfo as Integer

//Es ist bekannt, daß ein StyleRun einen bestimmten Platz einnimmt
//3 Bytes für die Booleans, 2 für die Größe und 4 für die Farbe
staticInfo = 1 + 1 + 1 + 2 + 4

```

```
//Ermittle die Anzahl von Styles
count = EditField1.StyledText.StyleRunCount
For i = 0 to Count-1 //get the StyleRuns
  sr = EditField1.StyledText.StyleRun( i )

  textLen = Len(sr.Text) + 1
  fontLen = Len(sr.Font) + 1

//Speichere den StyleRun-Inhalt und Style-Info in einen Memoryblock
mb = New MemoryBlock( staticInfo + fontLen + textLen )
mb.BooleanValue(0) = sr.Bold
mb.BooleanValue(1) = sr.Italic
mb.BooleanValue(2) = sr.Underline
mb.Short(3) = sr.Size
mb.ColorValue(5, 32) = sr.TextColor
mb.CString(9) = sr.Font
mb.CString(9 + fontLen) = sr.Text

//Nun können die Informationen in die Datei geschrieben werden
out.Write(mb)
Next

out.Close //Datei schließen
```

Siehe auch

[StyledText](#), [Range](#), [Paragraph](#), [EditField](#) Klassen.

Sub Befehl

Deklariert den Namen, die Parameter und den Programmcode, die das Gerüst einer Unterroutine (Subroutine oder Methode) bilden.

Syntax

Sub name([ParameterListe])

[lokale Variable-Deklarationen]

[Anweisungen]

[Return]

[Anweisungen]

[exception handlers]

[Finally]

End Sub

Teil	Beschreibung
<i>name</i>	Benötigt. Dies ist der Name der Subroutine (Methode) und folgt den üblichen Konventionen, nach denen auch Variablen benannt werden.
<i>ParameterListe</i>	Optional. Dies ist eine Liste von Werten, die die Parameter repräsentieren, die beim Aufruf an die Subroutine übergeben werden sollen. Mehrere Parameter werden durch Kommata getrennt.

Hinweis

Der Befehl **Sub** wird verwendet, um eine Methode zu definieren. Das Word **Sub** wird hierbei verwendet, da in der Programmiersprache BASIC eine Subroutine einer Methode entspricht. Methoden werden normalerweise mit einem Objekt verbunden (Exceptions sind globale Methoden oder Funktionen, die Teil eines Moduls sind). Der gesamte ausführbare Code muß von einem **Sub** oder einem [Function](#)-Befehl umschlossen werden.Ein **Sub** kann nicht innerhalb eines anderen **Sub** oder einer [Function](#) definiert werden. Ein **Sub** führt jede Zeile des Programmcodes von oben nach unten hin aus, vorausgesetzt, Sie haben kein GOTO verwendet. Wenn die letzte Zeile des Programmcodes einer **Sub**-Routine abgearbeitet wurde, gibt diese die Kontrolle über den Programmablauf wieder an die Stelle im Programmcode ab, die die **Sub**-Routine aufgerufen hat.

Eine **Sub**-Routine wird aufgerufen, indem ihr Name gefolgt von den Parametern in Klammern angegeben wird.

Alle in einer **Sub**-Routine verwendeten lokalen Variablen müssen deklariert werden, bevor sie in einem Aufruf verwendet werden können. Variablen und Parameter, die in einer **Sub**-Routine verwendet werden, können sowohl global als auch lokal sein. Sie existieren vom Start der Anwendung an bis zu deren Ende.

Globale Variablen werden erzeugt, indem Eigenschaften eines Moduls erzeugt. Lokale Variablen sind Variablen, die bei jedem Start einer **Sub**-Routine neu angelegt und beim Ende zerstört werden. Als Folge kann auf sie nur innerhalb der **Sub**-Routine zugegriffen werden. Sie werden durch den Befehl Dim aus einer **Sub**-Routine oder einer Function heraus angelegt.

Der Befehl **Return** kann verwendet werden, um sofort die Kontrolle über den Programmablauf an den Befehl zurückzugeben, der die **Sub**-Routine aufgerufen hat.

Exception-Handler sind Befehle, die Fehler zur Laufzeit verarbeiten. In der Beschreibung der RuntimeException-Klasse und des Exception Block finden Sie weitere Informationen.

Manchmal muß eine Funktion einige Aufräumarbeiten vornehmen, je nachdem ob sie normal beendet wurde oder ob sie verfrüht durch eine Exception verlassen werden mußte. Der optionale „Finally“-Block am Ende der Funktion ist für diesen Zweck gedacht. Code in diesem Block wird ausgeführt, auch wenn eine Exception auftrat, egal ob die Exception abgefangen und bearbeitet wurde oder nicht.

Wenn Ihre Methode einen Wert zurückliefern soll, sollten Sie sie als Function deklarieren. Eine Function ist eine Methode, die einen Wert zurückliefern kann. Dieser Rückgabewert kann auch ein eindimensionales Array sein. Bei der Beschreibung des Befehls Function finden Sie weitere Informationen.

Standardmäßig werden Parameter als Werte an Methoden übergeben. Die Methode erhält eine Kopie der Daten und kann Operationen mit diesen Daten ausführen. Wenn eine Methode mehr als einen Wert zurückliefern soll, können Sie Informationen optional als Referenz übergeben. In diesem Fall wird ein Zeiger auf das Objekt übergeben, das die Informationen enthält, es wird also keine Kopie der Daten angelegt und Ihre Methode arbeitet mit den Originaldaten. Der Vorteil der Parameterübergabe als Referenz besteht darin, daß die Methode die Werte der Parameter verändern kann, was nicht möglich ist, wenn die Parameter als Werte übergeben werden, da sie dann ja nur Kopien der Originaldaten sind.

Parameter als Referenz zu übergeben ist vor allem dann von Nutzen, wenn Ihre Methode mehrere Werte zurückliefern muß. Wenn die Parameter als Werte übergeben werden, kann die Methode nur einen einzigen Wert zurückliefern (indem Sie die Methode als Function deklarieren und der Wert als Ergebnis der Funktion zurückgegeben wird).

Mit den Schlüsselwörtern ByVal und ByRef können Sie festlegen, welche Art der Parameterübergabe verwendet werden soll. Um einen Parameter als Referenz zu übergeben, müssen Sie in der Methodendeklaration das Schlüsselwort ByRef verwenden, z.B. „ByRef MyInt as Integer“.

Eine **Sub** Methode kann sich selbst aufrufen, was eine Rekursion hervorrufen kann. Eine zu tief gehende Rekursion kann zu Stack-OverflowException-Fehlern führen.

Siehe auch

Exit, Break, Return, Function Befehle, RuntimeException Klasse, Exception Block.

Super kann nicht in einer Modul-Methode verwendet werden Fehler

Das Konzept von Super beschränkt sich nur auf die Klassen-Hierarchie, daher hat es keine Bedeutung in einer Methode eines Moduls.

Syntax Error Fehler

Ein Fehler aus einer Reihe verschiedener Fehler, der nicht durch eine genauere Fehlermeldung erklärt werden kann, ist aufgetreten.

Beispiele

Eine schließende Klammer wurde vergessen:

```
If (d > 0 then
.
end if
```

Es wurde ein Dim-Befehl innerhalb eines bedingten Ausdrucks verwendet:

```
Dim s as String
s="A"
```

```

if s="A" then
  Dim b as Boolean
  b=False
end if

```

Ein nicht korrekter Dim-Befehl:

```

Dim d Double // 'as' weggelassen
Dim f // 'as' und Datentyp weggelassen
Dim As Boolean //kein Variablenname
Dim Double as Double //'Double' ist ein reserviertes Wort
Dim a b as Integer //sollte 'a,b' sein
Dim c as //Typ fehlt
Dim myFlag as True //falscher Typ

```

Das 'Then'-Schlüsselwort fehlt in einem If-Befehl:

```

Dim a as Integer
a=5
If a >0 //müßte sein: If a>0 then
  MsgBox "Huhu"
end if

```

Dem Elseif-Befehl ging kein If-Befehl voraus.

```

Dim i,j,k as Integer
// mache irgendwas
elseif j>0 then
//mache irgendwas anderes
end if
//Korrekt
If j=0 then
.
Elseif j>0 then
.
End if

```

Das End If-Schlüsselwort fehlt bei einem If-Befehl:

```

Dim dayNum as Integer
if dayNum=1 then
  MsgBox "Es ist Montag"
// end if wird benötigt

```

#else wurde an Stelle von Else in einem #If-Befehl verwendet (bedingte Compilierung).

```

Dim c as Color
#If TargetMacOS then
  b=SelectColor( c , "Wähle eine Farbe")
else // müßte #else sein
  Beep
#EndIf

```

Das Next-Schlüsselwort fehlt bei einer For-Schleife:

```

Dim i,j as Integer
Dim aInts(2,2) as Integer
For i=0 to 2
  For j=0 to 2
    aInts(i,j)=i*j
  Next
//abschließendes 'Next' fehlt hier

```

Ein Next-Schlüsselwort wurde in diesem Beispiel zu viel verwendet:

```

Dim i,j as Integer
Dim aInts(2,2) As Integer
For i=0 to 2
  For j=0 to 2
    aInts(i,j)=i*j
  Next
Next
Next //zu viele "nexts"

```

Das Wend-Schlüsselwort wurde bei einer While-Schleife weggelassen:

```

While i<10
  Beep

```

Next //müßte Wend heißen

Der While-Befehl wurde weggelassen:

```
Dim i as Integer
i=1
//While-Befehl fehlt hier
  Beep
  i=i+1
Wend
```

Das Loop-Schlüsselwort fehlt bei einem Do-Befehl:

```
Dim i as Integer
Do until i=5
  Beep
  i=i+1
//Loop-Befehl fehlt
```

Ein Loop-Schlüsselwort wurde ohne vorangegangenen (passenden) Do-Befehl verwendet:

```
Dim x as Integer
// Do statement missing
x=x+1
Loop Until x<100
```

Das End Select-Schlüsselwort fehlt bei einem Select Case-Befehl:

```
Select case x
case 1
  MsgBox "Die Party des ersten Teils."
case 2
  MsgBox "Die Party des zweiten Teils."
//kein zugehöriger End Select Befehl
```

Der Select Case-Befehl fehlt:

```
Dim Day as String
case 1 //Select Case-Befehl fehlt
  Day = "Montag"
case 2
  Day = "Dienstag"
end select
```

Die Befehle Sub und Function können nicht innerhalb einer Methode verwendet werden.

```
Sub myMethod(x as Integer,y as Integer)
  Sub myMethod (x as Integer,y as Integer)
    .
  end sub
end sub
```

Ein Komma zeigt an, daß der zweite, benötigte Parameter fehlt:

```
ListBox1.insertRow 2,
```

Es wurde ein Gleichheitszeichen verwendet, wo es nicht erforderlich war:

```
Return=x*y //Das Gleichheitszeichen müßte ein Leerzeichen sein
```

Ein Schlüsselwort wurde falsch geschrieben:

```
If dayNum=1 then
  MsgBox "Es ist Montag"
end fi //sollte end if heißen
```

System Objekt

Die Eigenschaften des System-Objekts liefern Informationen über den Rechner des Anwenders.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>Keychain</i>	<u>Keychain</u>	Liefert den Default- <u>KeyChain</u> (Schlüsselbund). Sie können den Default-Schlüsselbund festlegen, indem Sie diesen auf einen anderen <u>KeyChain</u> setzen. Wenn kein <u>KeyChain</u> existiert, versucht es, einen zu erzeugen. Wenn Sie keinen erzeugen möchten, sollten Sie zuerst <code>System.KeyChainCount</code> aufrufen und auf 0 überprüfen.
<i>KeyChainCount</i>	<u>Integer</u>	Liefert die Anzahl verfügbarer Schlüsselbunde. Diese Zahl enthält sowohl alle Schlüsselbunde innerhalb des Schlüsselbund-Ordners als auch alle anderen dem Schlüsselbund-Manager bekannten Schlüsselbunde.
<i>KeyScript</i>	<u>Integer</u>	Erlaubt das Ermitteln und Festlegen der Tastaturbelegung. Die Werte sind die Werte für Skript-Systeme, z.B. 0 für MacRoman, 1 für MacJapanese etc. Zusätzlich können Sie negative Zahlen zuweisen, um verschiedene Spezial-Effekte hervorzurufen, z.B. wechselt das Zuweisen von -1 zum nächsten Tastatur-Layout im Tastatur-Menü. Weitere Informationen finden Sie unter: http://developer.apple.com/techpubs/mac/Text/Text-386.html
<i>MouseX</i>	<u>Integer</u>	Die X-Koordinate der Maus (Pixel). Gemessen von der linken oberen Ecke des Bildschirms.
<i>MouseY</i>	<u>Integer</u>	Die Y-Koordinate der Maus (Pixel). Gemessen von der linken oberen Ecke des Bildschirms.
<i>MouseDown</i>	<u>Boolean</u>	<u>True</u> , wenn Mausknopf gedrückt.
<i>PPPStatus</i>	<u>Integer</u>	Aktueller Status der PPP-Verbindung. Bitte schauen Sie in der Tabelle im Abschnitt „PPPStatus-Werte“ wegen einer Beschreibung der Bedeutung der von PPPStatus gelieferten Werte nach.
<i>QuickTime</i>	<u>QuickTime</u>	Liefert Zugriff auf das <u>QuickTime</u> -Objekt, welches bestimmte Eigenschaften der auf dem Rechner des Anwenders installierten QuickTime-Software liefert.
<i>SerialPortCount</i>	<u>Integer</u>	Anzahl der vorhandenen seriellen Schnittstellen.
<i>SerialPort</i>	<u>SerialPort</u>	Ein Read-only-Array der seriellen Schnittstellen, die der Rechner des Anwenders besitzt.

Methoden

Name	Typ	Beschreibung
<i>Gestalt</i>	Code als <u>String</u> ByRef Result als <u>Integer</u>	Liefert durch <i>Code</i> angegebene Eigenschaften des Systems. Gibt ein <u>Boolean</u> zurück, welches <u>True</u> ist, wenn der Aufruf erfolgreich war.
<i>Pixel</i>	X als <u>Integer</u> , Y als <u>Integer</u>	Liefert die Farbe eines Bildschirmpixels (als <u>Color</u>) in globalen Koordinaten. Wenn die angegebenen Koordinaten auf keinem Bildschirm existieren, wird als Farbe schwarz zurückgegeben.
<i>PPPConnect</i>	[UserIntervention als <u>Boolean</u>]	Stellt eine Wahlverbindung her. Der optionale Parameter ermöglicht Ihnen Benutzereingriff während des Einwahlprozesses. Es kann sein, daß mancher Ihrer Anwender mehr als einen ISP hat, bei dem er sich einwählen kann und der voreingestellte ist nicht immer unbedingt der, bei dem er sich einwählen möchte. Wenn das der Fall ist, setzen Sie den optionalen Parameter auf <u>True</u> , um Benutzereingriff zu gewähren. Dieses Feature ist auf dem Macintosh nicht verfügbar, da es dort keinen Standard-Dialog für dieses Feature gibt. Wenn der Anwender Windows 95/98/ME verwendet, Sie <u>False</u> übergeben oder den Parameter weglassen, versucht das System eine Verbindung mit dem ersten Telefonbuch-Eintrag, den es finden kann, herzustellen.
<i>PPPODisconnect</i>		Trennt eine PPP-Verbindung.

PPP

Das Point-to-Point Protokoll (PPP) wird zum Beispiel für Verbindung zum Internet über ein Modem verwendet. Dies ist eine systemweite Funktionalität, die Sie verwenden können, um eine Internetverbindung zu Ihrem ISP (Internet Service Provider) aufzubauen. Nach einem erfolgreichen Verbindungsaufbau können Sie mit Hilfe eines TCPSocket oder UDPSocket mit der Außenwelt kommunizieren.

Wenn Sie die Methode `PPPODisconnect` aufrufen, wird die Verbindung getrennt. Dies bedeutet, daß alle Verbindungen ins Internet geschlossen werden. Deshalb sollten Sie vor dem Schließen der Verbindung den Benutzer um Bestätigung bitten.

Die Methode `PPPConnect` verhält sich etwas anders. Wenn Sie auf dem Macintosh `True` angeben, wird keine Benutzerintervention benötigt. Das heißt, es gibt auf dem Macintosh keine standardisierten Benutzer-Dialoge, die angezeigt werden, um eine Verbindung auszuwählen. Unter Windows NT 4 oder höher wird Ihnen bei der Angabe von `True` der Standarddialog des RAS-Managers angezeigt. Unter Windows 98/ME wird die erste eingetragene Verbindung verwendet. Ebenso, wenn Sie unter Windows keine Angabe machen.

PPP Statuswerte

Die folgende Tabelle faßt die möglichen Statuswerte zusammen. Um den Status zu ermitteln, vergleichen Sie die `PPPStatus`-Eigenschaft mit den hier aufgeführten Konstanten.

Wert	Klassen-Konstante	Beschreibung
0	<code>PPPONoConnection</code>	Keine Verbindung vorhanden. Das bedeutet, Sie haben die <code>PPPConnect</code> -Methode noch nicht aufgerufen oder der Verbindungsaufbau ist fehlgeschlagen. Es kann ebenso bedeuten, daß Sie bereits die <code>PPPODisconnect</code> -Methode aufgerufen haben und die Verbindung geschlossen wurde.
1		Nicht verwendet.
3	<code>PPPConnectionClosing</code>	Die Verbindung wird geschlossen. Das bedeutet, Sie haben Die <code>PPPODisconnect</code> -Methode aufgerufen und der Verbindungsabbau wird gerade durchgeführt und ist noch nicht beendet.
4	<code>PPPConnectionOpening</code>	Eine Verbindung wird aufgebaut. Dies bedeutet nicht, daß Sie eine gültige Verbindung haben. Die Verwendung der <code>TCPsocket</code> oder <code>UDPSocket</code> ohne eine gültige Verbindung kann zu Fehlern führen. Testen Sie die Verbindung deshalb vor der Verwendung eines Sockets.
5	<code>PPPConnected</code>	Sie haben eine gültige Verbindung. Nun können Sie Ihren <code>TCPsocket</code> oder <code>UDP-Socket</code> verwenden, da die Verbindung vollständig aufgebaut ist.

Hinweise

`MouseX` und `MouseY` liefern Koordinaten in Bezug auf die linke obere Ecke des Monitors des Anwenders. Die Klassen `Window` und `Control` enthalten `MouseX`- und `MouseY`-Eigenschaften, die sich aufs Fenster beziehen (im Fall der `Control`-Klasse aufs Eltern-Fenster des Steuerelements).

Die Eigenschaften `SerialPort` und `SerialPortCount` arbeiten normal in carbonisierten Applikationen, die unter Mac OS X laufen. Carbon-Applikationen, die im Classic Mac OS laufen, können jedoch nicht auf die seriellen Ports zugreifen. Für diese Rechner sollten Sie eine separate Classic-Version Ihrer Applikation erzeugen.

Das Point-to-Point-Protokoll (PPP) ist das Protokoll, das verwendet wird, um über ein Einwahlmodem eine Internetverbindung herzustellen. Nachdem eine Verbindung hergestellt ist, können Sie Objekte verwenden, die von den Klassen `TCPsocket` und/oder `UDPSocket` abgeleitet sind, um die Verbindung zu verwalten.

In Version 5.0 wurden die `PPPStatus`-Eigenschaft und die `PPPConnect`- und `PPPODisconnect`-Methoden aus der `Socket`-Klasse in das **System**-Objekt verschoben. Dies liegt daran, daß der Gültigkeitsbereich von `PPPStatus`, `PPPODisconnect` und `PPPConnect` systemweit ist. Sie beziehen sich auf alle Verbindungen, die auf dem Rechner des Anwenders aufgebaut wurden. Zum Beispiel trennt der Aufruf der `PPPODisconnect`-Methode eine bestehende Internetverbindung, egal von welcher Applikation diese aufgebaut wurde.

Unterstützung für mehrere Netzwerkkarten

Eine Neuerung in der Version 5.5 ist, daß Sie nun mehrere Netzwerkkarten unter Mac OS X, Windows und Linux verwenden können. Dies erlaubt es Ihnen, eine Applikation zu schreiben, die verschiedene Netzwerkkarten benutzt, um zum Beispiel sogenanntes „tunneling“ durchzuführen. Dies ist mit früheren Versionen von REALbasic nicht möglich. Um die vorhandenen Netzwerkkarten zu ermitteln, können Sie die Methode `GetNetworkInterface` verwenden und ein `NetworkInterface`-Objekt der `NetworkInterface`-Eigenschaft der `SocketCore`-Klasse zuweisen.

Beispiel

Folgendes Beispiel zeigt die Anzahl und Namen der seriellen Schnittstellen an, die der Rechner des Anwenders besitzt. Auf einem „normalen“ Macintosh ist `SerialPort 0` der Modemanschluß und `SerialPort 1` der Druckeranschluß:

```
Dim system as systemclass
editfield1.text=Str(system.serialportcount)
editfield2.text=system.serialport(0).Name
editfield3.text=system.serialport(1).Name
```

Folgendes Beispiel liefert die Versionsnummer des installierten QuickTime:

```
Dim s as string
Dim b as boolean
Dim i,resp as integer
b=system.gestalt("qtim",resp)
If b then
    s=Hex(resp)
    For i =Len(s) downto 1
        s=Left(s,i)+". "+Mid(s,i+1)
    Next
    MsgBox "QuickTime Version "+s
End if
```

Das Ergebnis ist:



Folgendes Beispiel überprüft, ob das Programm unter Mac OS 8.5 oder neuer läuft:

```
// Ermittelt die Betriebssystemversion, Da diese
// Funktionen nur unter Mac OS 8.5
// oder neuer vorhanden sind
If System.Gestalt("sysv", sysVersion) and sysVersion >= &h0850 then
    // mache irgendwas tolles hier
end if
```

Um festzustellen, ob das Programm unter Mac OS X läuft, verwenden Sie den gleichen Code, jedoch mit der Konstanten &h1000.

Siehe auch

[SerialPort](#), [Application](#) Klassen; [App](#), [NetworkInterface](#) Objekte.

SystemFolder Funktion

Wird benutzt, um Zugriff auf den Systemordner zu bekommen. Unter Windows greift es auf das Verzeichnis WINDOWS\SYSTEM zu.

Syntax

Ergebnis=**SystemFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Ein FolderItem , das den Systemordner repräsentiert.

Hinweis

Verwenden Sie die Funktion **SystemFolder** immer dann, wenn Sie Zugriff auf den Systemordner des Anwenders benötigen.

Die Funktion **SystemFolder** bietet eine Möglichkeit, auf den Systemordner zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert und die auch bei künftigen Versionen, falls Apple den Systemordner umorganisieren sollte, funktionieren wird.

Wenn Ihr Programm auch unter Windows verwendet wird, können Sie vor Aufruf von SystemFolder mit der globalen boole'schen Konstanten [TargetWin32](#) überprüfen, ob das Programm auf einem Windows-Rechner läuft.

Beispiele

Bei der Beschreibung der Funktion [AppleMenuFolder](#) finden Sie ein passendes Beispiel.

Siehe auch

[AppleMenuFolder](#), [ApplicationSupportFolder](#), [ControlPanelsFolder](#), [DesktopFolder](#), [ExtensionsFolder](#), [FontsFolder](#), [PreferencesFolder](#), [StartupItemsFolder](#), [TemporaryFolder](#), [TrashFolder](#) Funktionen.

TabPanel Steuerelement



Das Standard-Karteireiter-Steuerelement.

Übergeordnete Klasse

[PagePanel](#)

Da dies ein [RectControl](#) ist, schauen Sie wegen der anderen Eigenschaften und Events, die für alle [RectControl](#)-Objekte gemeinsam gelten, unter [RectControl](#) nach.

Eigenschaften

Name	Art	Beschreibung
Facings	Integer	Ausrichtung (nur bei Mac OS 8.5 und neuer) 0 – Nord 1 – Süd 2 – Ost 3 – West
SmallTabs	Boolean	Wenn True , verringern sich die vertikalen Ausmaße des Karteireiters (nicht unter Windows verfügbar).

Methoden

Name	Parameter	Beschreibung
Caption	Index als Integer	Erlaubt Ihnen, die Beschriftung des angegebenen Tabs zur Laufzeit zu setzen. Der erste Tab hat die Nummer Null.

Events

Name	Parameter	Beschreibung
Change		Der vorne liegende Tab hat sich geändert.
GotFocus		(Nur Win32) Das TabPanel hat den Fokus bekommen. Wenn der Anwender dieses Steuerelement durch Drücken der Taste Tab angewählt hat, bekommt das oberste Tab den Selektionsrahmen.
LostFocus		(Nur Win32) Das TabPanel hat den Fokus verloren.
MouseDown	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des TabPanels an der Position x,y gedrückt. Gibt ein Boolean zurück. Geben Sie True zurück, wenn Sie das MouseDown-Event auswerten wollen. Die Koordinaten x und y sind lokal zum Steuerelement. Der Punkt 0,0 ist die linke obere Ecke des gesamten Steuerelements (links von der Beschriftung und über der Box, die gezeichnet wurde).
MouseDrag	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des TabPanels gedrückt und die Maus an die Position x,y relativ zum Steuerelement bewegt (gedraggt). Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.
MouseUp	x als Integer , y als Integer	Der Mausknopf wurde innerhalb des TabPanels an der Position x,y losgelassen. Dieses Event tritt erst auf, wenn Sie im MouseDown-Event True zurückliefern.

Hinweise

Ermitteln und Festlegen des angezeigten Panels

Benutzen Sie die Value-Eigenschaft, um das angezeigte Panel zu ermitteln oder zu setzen. Das erste Panel hat den Index 0. Das TabPanel ist eine Unterklasse vom [PagePanel](#). Die Value-Eigenschaft wird dort näher erklärt.

Hinzufügen, Benennen und Umbenennen von Tabs

Sie modifizieren die Tab-Reiter in einem TabPanel, indem Sie auf den letzten Tab-Reiter namens „...“ klicken. Das ruft den Tab-Panel-Editor auf:



Hier können Sie nun:

- Den ersten Tab-Reiter umbenennen, indem Sie ihn selektieren und auf „Ändern“ klicken.
- Tab-Reiter und deren Beschriftung hinzufügen, indem Sie auf den „Neu“-Knopf klicken.
- Einen beliebigen Tab-Reiter löschen, indem Sie ihn selektieren und auf „Löschen“ klicken.
- Die Tab-Reiter umarrangieren, indem Sie einen Tab-Reiter selektieren und „Rauf“ oder „Runter“ anklicken.

Sie fügen einem bestimmten Panel Steuerelemente hinzu, indem Sie auf seinen Tab-Reiter klicken und Steuerelemente in dieses Panel draggen.

Ausrichtung und SmallTabs

Die Facing-Eigenschaft erlaubt Ihnen, die Tab-Reiter an jeder beliebigen Seite des TabPanel zu positionieren. Die SmallTab-Eigenschaft reduziert die vertikale (Nord oder Süd) oder die horizontale (Ost oder West) Größe des Tabs. Folgender Vergleich zeigt den Effekt der SmallTabs-Eigenschaft in einem TabPanel, das die Süd-Ausrichtung verwendet.



SmallTabs aus



SmallTabs Ein

Plazieren von Steuerelementen auf Tab Panels

Sie können andere Steuerelemente auf die Oberfläche eines jeden Tab-Panels plazieren. Solche Steuerelemente werden dann sichtbar, wenn der Anwender auf diesen Karteireiter klickt, um ihn nach oben zu bringen. Sie können die TabPanelIndex-Eigenschaft der Control-Klasse verwenden, um das Panel zu ermitteln, auf der sich das Steuerelement befindet. Sie können ein Steuerelement nicht mittels Code auf ein anderes Panel verschieben. Bitte beachten Sie, daß TabPanelIndex auf einem anderen Weg durch das Auslesen des Elternobjekts des Steuerelements oder Code, der TabPanelIndex verwendet, ersetzt werden könnte und vielleicht neu geschrieben werden muß.

Wenn die Caption-Eigenschaft ein Und-Zeichen enthält, wird dieses nur angezeigt, wenn ein weiteres Und-Zeichen folgt. Dies ist so, damit sich Macintosh- und Windows-Programme gleich verhalten.

Siehe auch

RectControl Klasse, PagePanel Steuerelement.

Tan Funktion

Gibt den Tangens des übergebenen Winkels zurück.

Syntax

Ergebnis=Tan (Wert)

Teil	Art	Beschreibung
Ergebnis	<u>Double</u>	Der Tangens von Wert.
Wert	<u>Double</u>	Der Wert (im Bogenmaß), dessen Tangens Sie ermitteln wollen.

Hinweis

Die **Tan**-Funktion gibt den Tangens des übergebenen Winkels (im Bogenmaß) zurück.

Beispiele

Dieses Beispiel verwendet die **Tan**-Funktion, um den Tangens einer Zahl zu ermitteln. Wenn der Winkel im Gradmaß vorliegt, multiplizieren Sie ihn mit $\pi/180$, um ihn ins Bogenmaß umzurechnen.

```
Dim d As Double
Const PI=3.14159265358979323846264338327950
d=Tan(45*PI/180) //ergibt 1.0
```

Siehe auch

[Atn](#) Funktion.

Target68K Konstante

Wird verwendet, um festzustellen, ob Sie die Applikation auch für 68K compilieren. Diese Konstante ist obsolet. REALbasic 4.0 und höher erzeugen keine 68K-Applikationen mehr.

Siehe auch

[TargetWin32](#), [DebugBuild](#), [TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetMachO](#), [TargetLinux](#), [RBVersion](#), [RBVersionString](#), [TargetHasGUI](#) Funktionen; [#If](#) Befehl.

TargetCarbon Konstante

Wird verwendet, um hinzuweisen, daß Sie Carbon-Code compilieren.

Syntax

result=**TargetCarbon**

Teil	Typ	Beschreibung
<i>result</i>	Boolean	Liefert True, wenn Sie Carbon-Code compilieren, entweder als ein eigenständiges Programm oder innerhalb von REALbasic.

Hinweise

Die Carbon-Version von REALbasic compiliert für Carbon, wenn man ein Projekt in der Entwicklungsumgebung testet.

[Target68K](#), [TargetPPC](#), und **TargetCarbon** sind sich gegenseitig ausschließende Untermengen von [TargetMacOS](#). [TargetPPC](#) ist nur True, wenn PPC-Code auf einem PowerPC unter einem Classic Mac OS ausgeführt wird.

Siehe auch

[TargetWin32](#), [DebugBuild](#), [TargetLinux](#), [TargetMacOS](#), [TargetMachO](#), [TargetPPC](#), [Target68K](#), [RBVersion](#), [RBVersionString](#), [TargetHasGUI](#) Funktionen; [#If](#) Befehl.

TargetHasGUI Konstante

Wird verwendet, um festzustellen, ob eine Applikation eine grafische Benutzeroberfläche hat oder nicht. Applikationen vom Typ [ConsoleApplications](#) haben keine grafische Oberfläche.

Syntax

result=**TargetHasGUI**

Teil	Typ	Beschreibung
<i>result</i>	Boolean	Liefert <u>True</u> , wenn Sie REALbasic-Code compilieren, der ein grafische Benutzeroberfläche besitzt.

Beispiel

Benutzen Sie **TargetHasGUI**, um Code-Teile zu isolieren, die für eine Version Ihrer Konsolen-Applikation gedacht sind, die eine grafische Benutzeroberfläche besitzt. In dem folgenden Beispiel wird eine Fehlermeldung in die Standard-Fehler-Ausgabe geschrieben, wenn die Applikation keine GUI besitzt.

```
Sub DisplayError(err as String)
  #if TargetHasGUI
    MsgBox err
  #else
    StdErr.WriteLine err
  #endif
End Sub
```

Siehe auch

[TargetWin32](#), [DebugBuild](#), [TargetLinux](#), [TargetMacOS](#), [TargetPPC](#), [TargetCarbon](#), [TargetMachO](#), [Target68K](#), [RBVersion](#), [RBVersionString](#) Funktionen; [#If](#) Befehl; [ConsoleApplication](#), [ServiceApplication](#) Klassen.

TargetLinux Konstante

Wird verwendet, um festzustellen, ob für Linux compiliert wird.

Syntax

result=**TargetLinux**

Teil	Typ	Beschreibung
<i>result</i>	Boolean	Liefert True , wenn Sie für Linux compilieren.

Siehe auch

[TargetWin32](#), [DebugBuild](#), [TargetMacOS](#), [TargetCarbon](#), [TargetMachO](#), [TargetPPC](#), [TargetHasGUI](#), [RBVersion](#), [RBVersionString](#) Funktionen; [#If](#) Befehl.

TargetMachO Konstante

Wird verwendet, um festzustellen, ob für Mac OS X Mach-O compiliert wird. Eine Mach-O-Applikation funktioniert nur unter Mac OS X, da dies das native Format des Mac OS X-Kernels ist.

Syntax

result=**TargetMachO**

Teil	Typ	Beschreibung
<i>result</i>	Boolean	Liefert True , wenn Sie für Mach-O compilieren.

Siehe auch

[TargetWin32](#), [DebugBuild](#), [TargetLinux](#), [TargetHasGUI](#), [TargetMacOS](#), [TargetPPC](#), [RBVersion](#), [RBVersionString](#) Funktionen; [#If](#) Befehl.

TargetMacOS Konstante

Wird verwendet, um hinzuweisen, daß Sie Code für ein beliebiges Mac OS, 68K oder PPC Classic oder Carbon/Mac OS X compilieren.

Syntax

result=**TargetMacOS**

Teil	Typ	Beschreibung
<i>result</i>	Boolean	Liefert True , wenn Sie Code für ein beliebiges Mac OS compilieren, egal ob als eigenständiges Programm oder innerhalb von REALbasic.

Hinweise

[Target68K](#), [TargetPPC](#), und [TargetCarbon](#) sind sich gegenseitig ausschließende Untermengen von **TargetMacOS**. [TargetPPC](#) ist nur True, wenn PPC-Code auf einem PowerPC unter einem Classic Mac OS ausgeführt wird.

Die letzte Version von REALbasic, die 68K-Code compilieren kann, ist die 3.5.2.

Siehe auch

[TargetWin32](#), [DebugBuild](#), [Target68K](#), [TargetLinux](#), [TargetPPC](#), [TargetCarbon](#), [TargetHasGUI](#), [TargetMachO](#), [RBVersion](#), [RBVersionString](#), Funktionen; [#If](#) Befehl.

TargetPPC Konstante

Wird verwendet, um hinzuweisen, daß Sie PPC-Code compilieren.

Syntax

result=**TargetPPC**

Teil	Typ	Beschreibung
<i>result</i>	<u>Boolean</u>	Liefert <u>True</u> , wenn Sie PPC-Code compilieren, entweder als ein eigenständiges Programm oder innerhalb von REALbasic.

Hinweise

[Target68K](#), **TargetPPC**, und [TargetCarbon](#) sind sich gegenseitig ausschließende Untermengen von [TargetMacOS](#). [TargetPPC](#) ist nur True, wenn PPC-Code auf einem PowerPC unter einem Classic Mac OS ausgeführt wird.

Siehe auch

[TargetWin32](#), [DebugBuild](#), [TargetLinux](#), [TargetMacOS](#), [Target68K](#), [TargetCarbon](#), [TargetHasGUI](#), [TargetMachO](#), [RBVersion](#), [RBVersionString](#) Funktionen; [#If](#) Befehl.

TargetWin32 Konstante

Wird verwendet, um hinzuweisen, daß Sie Code für Windows-Betriebssysteme compilieren.

Syntax

result=**TargetWin32**

Teil	Typ	Beschreibung
<i>result</i>	<u>Boolean</u>	Liefert <u>True</u> , wenn Sie Windows-Code compilieren.

Siehe auch

[TargetMacOS](#), [DebugBuild](#), [Target68K](#), [TargetCarbon](#), [TargetPPC](#), [RBVersion](#), [RBVersionString](#) Funktionen; [#If](#) Befehl.

TCPIP4DServer Klasse

Wird zum Öffnen einer 4th Dimension/4D Server Datenquelle verwendet. Verwenden Sie diese Klasse anstelle von [Open4DDatabaseByTCPIP](#).

Übergeordnete Klasse

[Database](#)

Eigenschaften

Name	Typ	Beschreibung
Task	<u>String</u>	Beschreibung der Verbindung. Der Task-Parameter ist der Name des Prozesses, der auf dem 4D-Server erzeugt wird, wenn Sie eine Verbindung aufbauen. Dieser Name wird im 4D-Server-Fenster angezeigt.

Hinweise

Um die Klasse verwenden zu können, muß das Plugin für die 4D-Version, die Sie verwenden, in Ihrem REALbasic Plugin-Ordner installiert sein. Die Versionsnummer ist an den Klassennamen angehängt, z.B. „TCPIP4DServer60“.

Wenn Sie eine 4D-Datenquelle unter Windows verwenden, sollten Sie die 4DOpen.DLL-Datei (von 4D bereitgestellt) in das gleiche Verzeichnis wie Ihre Applikation kopieren. Das stellt sicher, daß die Applikation diese Library finden kann und daß die passende Version der Library verwendet wird.

Sie finden eine Reihe von Database-Plugins auf der REALbasic-CD, neuere Versionen könnte es aber auf der REAL Software Web-Site unter www.realsoftware.com geben.

Um Zugriff auf einen 4D-Server zu erhalten, muß dieser entsprechend konfiguriert sein und der Benutzer *username* muß die erforderlichen Zugriffsrechte besitzen, um eine Verbindung aufbauen zu können.

Beispiel

Folgender Code stellt eine Verbindung zu einem 4D Server über TCP/IP her:

```
Dim db as TCPIP4DServer67
.
db = New TCPIP4DServer67
db.host = "127.0.0.1"
db.username = "ich"
db.password = "geheim"
.
If db.connect then
//weiter mit Datenbankoperationen.
else
//Fehler beim Verbindungsaufbau
end if
```

Siehe auch

[ADSP4DServer](#), [Database](#) Klassen.

TCPSocket Steuerelement



Wird für TCP/IP-Kommunikation verwendet. Diese Klasse ersetzt die [Socket](#)-Klasse.

Übergeordnete Klasse

[SocketCore](#) Klasse

Eigenschaften

Name	Typ	Beschreibung
<i>Address</i>	String	Die TCP/IP-Adresse, mit der verbunden werden soll.
<i>BytesAvailable</i>	Integer	Die Anzahl an Bytes von im internen Empfangspuffer verfügbaren Daten.
<i>PPPStatus</i>	Integer	Diese Eigenschaft wurde durch die PPPStatus-Eigenschaft des System -Objekts ersetzt.
<i>RemoteAddress</i>	String	Die Remote-Adresse des anderen Rechners, mit dem Sie verbunden sind. Verwenden Sie dies an Stelle der Address-Eigenschaft, um die Adresse des Rechners, mit dem Sie momentan verbunden sind, herauszufinden.

Events

Name	Parameter	Beschreibung
<i>Connected</i>		Tritt auf, wenn eine Verbindung aufgebaut wurde.
<i>Error</i>		Ein Fehler ist aufgetreten.
<i>SendComplete</i>	UserAborted als Boolean	Alle Daten im Socket-Puffer wurden geschrieben (gesendet). Der Parameter <i>UserAborted</i> erlaubt Ihnen, festzustellen, ob der Anwender die Übertragung durch Rückgabe von True vom SendProgress-Event abgebrochen hat. Wenn der Anwender sie abgebrochen hat, ist der Wert True , andernfalls ist er False .

Name	Parameter	Beschreibung
<i>SendProgress</i>	BytesSent as <u>Integer</u> BytesLeft as <u>Integer</u>	Tritt auf, wenn Ihr Netzwerkprovider Ihre Daten in „Happen“ liefert und dabei ist, den nächsten Happen zu schicken. Die Parameter zeigen das Ausmaß des Fortschritts an, der während des Sendes gemacht wurde. Liefert ein <u>Boolean</u> . Das Zurückgeben von <u>True</u> aus diesem Event führt zum Abbrechen des Sendens.

Methoden

Name	Parameter	Beschreibung
<i>Disconnect</i>		Bricht die Verbindung des Sockets ab, setzt es zurück und löst einen Socket 102-Fehler aus, um Ihnen mitzuteilen, daß die Verbindung des Socket unterbrochen wurde.
<i>Listen</i>		Versucht, nach eingehenden Verbindungen auf dem derzeit festgelegten Port zu lauschen. Nach dem Aufruf von Listen liefert die Port-Eigenschaft den tatsächlichen Port, der gerade verwendet wird.
<i>LookAhead</i>	[Encoding als <u>String</u>]	Liefert einen String. Liefert alle nicht ausgelesenen Zeichen aus dem Puffer, ohne sie aus dem Puffer zu entfernen. Der optionale Parameter <i>Encoding</i> erlaubt Ihnen das Text-Encoding der Daten festzulegen, die zurückgeliefert werden.
<i>PPPConnect</i>		Wurde durch die PPPConnect-Methode des <u>System</u> -Objekts ersetzt.
<i>PPPODisconnect</i>		Wurde durch die PPPODisconnect-Methode des <u>System</u> -Objekts ersetzt.
<i>Read</i>	Bytes als <u>Integer</u> ; [Encoding als <u>TextEncoding</u>]	Liest die angegebene Menge an Bytes aus dem internen Empfangspuffer. Der optionale Parameter <i>Encoding</i> erlaubt Ihnen das Text-Encoding der Daten festzulegen, die zurückgeliefert werden. Verwenden Sie das <u>Encodings</u> -Objekt, um ein Text-Encoding festzulegen.
<i>ReadAll</i>	[Encoding als <u>TextEncoding</u>]	Liest alle Daten aus dem internen Puffer. Der optionale Parameter <i>Encoding</i> erlaubt Ihnen das Text-Encoding der Daten festzulegen, die zurückgeliefert werden. Verwenden Sie das <u>Encodings</u> -Objekt, um ein Text-Encoding festzulegen.
<i>Write</i>	Data als <u>String</u>	Schreibt <i>Data</i> an die Verbindungsgegenstelle.

Hinweise

Das *Transmission Control Protocol*, oder TCP, ist die Basis für den Großteil des Internet-Verkehrs. Es ist ein verbindungsorientiertes Protokoll, das einen verlässlichen Weg bietet, Daten über ein Netzwerk zu auszutauschen.

Um eine Verbindung zwischen zwei Rechnern unter Verwendung der TCP Sockets aufzubauen, muß einer der Rechner so eingestellt werden, daß er an einem bestimmten Port lauscht. Der andere Rechner (genannt Client) versucht dann, eine Verbindung herzustellen, indem es die Netzwerk-Adresse (oder IP-Adresse) des anderen Rechners und den Port, über den die Verbindung laufen soll, festlegt. Damit Daten an den anderen Rechner gesendet oder von diesem empfangen werden können, müssen beide Rechner wissen, daß diese Verbindung aufgebaut wird. Das geht entweder, indem man einen wohlbekannten Port für den Lauscher (oder Server) nimmt, an dem gelauscht wird oder durch eine andere vorzeitige Absprache (z.B., wenn Sie der Autor sowohl von Client und Server sind).

Wenn ein Server einen Verbindungsversuch auf dem Port, an dem er lauscht, erhält, nimmt er die eingehende Verbindung an und sendet eine Empfangsbestätigung zurück an den anderen Rechner. Nachdem beide Rechner eine Vereinbarung getroffen haben (oder „verbunden“ sind), können Sie mit dem Senden und Empfangen von Daten beginnen. Wenn Sie Ihre Verbindung mit dem anderen Rechner beenden, gibt es einen ähnlichen Handshake-Prozeß, so daß beide Rechner wissen, daß die Verbindung dabei ist, beendet zu werden.

Aufgrund des Umfangs an Fehlerkontrolle und Handshakes ist TCP ein sehr verlässliches Protokoll. Wenn Sie ein Paket mit Informationen senden, ist es garantiert, daß es den anderen Rechner erreicht, außer die Verbindung wurde unterbrochen, entweder zufällig oder absichtlich. Dieses Feature bedeutet aber eine Menge Overhead. Ein typisches TCP-Paket, das über das Netzwerk versendet wird, besitzt einen 40-Byte-Header, der es begleitet. Dieser Header wird von allen Rechnern auf der Route bis zum Bestimmungsort überprüft und verändert. Dadurch ist TCP ein recht langsames Protokoll. Der Kompromiß lautet hier Geschwindigkeit gegen Sicherheit, wobei Sicherheit bevorzugt wird. Wenn Sie Geschwindigkeit benötigen (z.B. für ein Netzwerkspiel), sollten Sie sich das UDP-Protokoll ansehen.

Die Sendegeschwindigkeit von **TCPSocket** ist schneller als die der Socket-Klasse. Dieses neue Feature gibt es jedoch nur auf dem Macintosh.

Der **TCPSocket** verwaltet den Prozeß der Verbindungstrennung eleganter. Er versucht, die Verbindung, wenn möglich, normal zu trennen (wodurch alle Datentransfers normal beendet werden können) und greift nur dann auf einen vorzeitigen Abbruch zurück, wenn ein normaler nicht möglich ist. **TCPSockets** behandelt nun eine Flußkontrolle, so daß beim Senden einer großen Datenmenge beim Sende- oder Empfangs-Prozeß keine Daten verloren gehen.

Es ist möglich, daß Datenpakete eintreffen, bevor der Socket den Verbindungsvorgang zu Ende geführt hat. Sollte dies passieren, ist **TCPSocket** darauf vorbereitet, das zu behandeln, wodurch Ihre Anfangspakete nicht verloren gehen. Dieses Feature gibt es nur auf dem Macintosh; Windows-Sockets konnten diese Situationen schon vorher meistern.

Das SendProgress-Event erlaubt es Ihnen, die Sendegeschwindigkeit festzulegen und teilt Ihnen mit, wieviele Bytes an Daten Sie seit des letzten SendProgress-Events gesendet haben und wieviele noch zu senden sind. Wenn Sie True vom SendProgress-Event zurückliefern, brechen Sie den aktuellen Transfer ab. Das schließt nicht die Verbindung des Sockets; es leert nur den Puffer. Sie können dieses Event verwenden, um herauszufinden, ob eine Verbindung zu langsam ist und diese dann abbrechen. Nachdem alle Daten übertragen wurden, erhalten Sie ein letztes SendProgress-Event, gefolgt von einem SendComplete-Event.

Die RemoteAddress-Eigenschaft für ein Verbindungs-Socket liefert die IP-Adresse der entfernten Host-Verbindung bei Windows-Sockets. In früheren REALbasic-Versionen würde ein Verbindungs-Socket nur einen leeren String liefern, wenn Sie es nach der RemoteAddress-Eigenschaft fragen würden. Eine nützliche Sache, die Sie mit der RemoteAddress-Eigenschaft machen können, ist ein Test, ob die IP-Adresse, mit der Sie eine Verbindung herstellen möchten, die gleiche wie die IP-Adresse ist, mit der Sie momentan verbunden sind.

In Version 5.0 (und später) können **TCPSockets** nun verwaisten. Ein verwaister Socket sperrt sich selbst, nachdem Sie die Connect- oder Listen-Methode aufrufen. Der Socket wird wieder freigegeben, wenn die Verbindung unterbrochen wurde. Das bedeutet, das ein verwaister Socket so lange weiter arbeitet, wie die Verbindung besteht. Das bedeutet außerdem, daß Sie die Close-Methode aufrufen müssen, bevor die Sperrung aufgehoben wird.

Wenn Sie einen Socket nicht länger benötigen, sollten Sie sicherstellen, daß dieser letztendlich eine „Verbindung schließen“-Nachricht erhält. Ein Webserver: zum Beispiel wird seinen Socket erst dann schließen, wenn alle Daten verschickt wurden. Wenn Sie nun Ihre Daten empfangen haben, bleibt der Socket so lange verbunden, bis Sie explizit die Close-Methode aufrufen. Versichern Sie sich also, daß Ihre Sockets geschlossen werden, wenn diese nicht weiter benötigt werden. Sie können stattdessen auch die Disconnect-Methode aufrufen.

Unter Mac OS X kann ein Anwender, der keine Root-Rechte besitzt, nicht auf Ports kleiner 1024 zugreifen. Sie können einen Socket verwenden, um mit einem Server auf Ports kleiner 1024 eine Verbindung aufzunehmen, Sie können aber keine Mac OS X-Applikation schreiben, die auf diesen Ports lauscht, ohne Root zu sein.

Die **TCPSocket**-Klasse implementiert das „Readable“- und „Writable“-Klassen-Interface. Wenn Sie diese selbst implementieren wollen, müssen Sie eine Read- und Write-Methode anbieten, die den folgenden entspricht:

Syntax

Ergebnis=**Read** (count, [enc])

Teil	Typ	Beschreibung
result	<u>String</u>	Der Text, der zurückgegeben wird.
count	<u>Integer</u>	Die Anzahl der Bytes die gelesen werden sollen.
enc	<u>TextEncoding</u>	Optionaler Parameter, mit dem die Kodierung des Strings angegeben werden kann.

Das Readable-Klassen-Interface besitzt zwei Methoden.

Name	Parameter	Beschreibung
EOF		Liefert <u>True</u> , wenn das Ende des Streams erreicht ist oder der Socket geschlossen ist und keine Daten im internen Puffer vorhanden sind.
ReadError		Liefert <u>True</u> , wenn ein Fehler aufgetreten ist.

Write(data)

Teil	Typ	Beschreibung
data	<u>String</u>	Der Text, der geschrieben werden soll.

Das Writeable-Klassen-Interface besitzt folgende Methoden:

Name	Parameter	Beschreibung
Flush		Schreibt die verbleibenden Daten im internen Puffer auf das Volume und löscht den Puffer. Für Sockets und serielle Streams wird ein letztes Mal Poll aufgerufen und dann der Puffer gelöscht.
WriteError		Liefert <u>True</u> , wenn ein Fehler aufgetreten ist.

Weitere Informationen über Klassen-Interfaces und wie diese implementiert werden, finden Sie im Benutzerhandbuch im Kapitel „Klassen-Interfaces“.

Siehe auch

[SocketCore](#), [SSLSocket](#), [ServerSocket](#), [UDPSocket](#), [EasyTCPSocket](#) Klassen.

TemporaryFolder Funktion

Wird benutzt, um Zugriff auf die Einträge des temporären Ordners zu bekommen. Unter Windows greift sie auf das TEMP-Verzeichnis des Anwenders zu. Unter Linux wird das /tmp/-Verzeichnis verwendet.

Syntax

Ergebnis=**TemporaryFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Ein FolderItem , das den Ordner repräsentiert.

Hinweis

Der temporäre Ordner ist ein unsichtbarer Ordner auf dem Start-Volume des Anwenders. Verwenden Sie diesen, wenn Sie eine Datei sichern möchten, die der Anwender nicht auf dem Schreibtisch sehen soll. [GetTemporaryFolderItem](#) erzeugt ein [FolderItem](#) in diesem Ordner.

Die Funktion **TemporaryFolder** bietet eine Möglichkeit, auf den temporären Ordner zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert.

Beispiele

Bei der Beschreibung der Funktion [AppleMenuFolder](#) finden Sie ein passendes Beispiel.

Siehe auch

[AppleMenuFolder](#), [ApplicationSupportFolder](#), [ControlPanelsFolder](#), [DesktopFolder](#), [ExtensionsFolder](#), [FontsFolder](#), [PreferencesFolder](#), [StartupItemsFolder](#), [SystemFolder](#), [TrashFolder](#), [GetTemporaryFolderItem](#) Funktionen.

TextColor Funktion

Liefert die momentan im Erscheinungsbild-Manager eingestellte Farbe zurück, die zum Zeichnen von Text verwendet wird.

Syntax

Ergebnis=**TextColor**

Teil	Art	Beschreibung
<i>Ergebnis</i>	Color	Die zum Zeichnen von Text verwendete Farbe.

Hinweis

Dieser Wert ist bei der Arbeit mit [Canvas](#) Steuerelementen wichtig, um deren Aussehen mit den Einstellungen des Erscheinungsbild-Managers zu synchronisieren. Wenn Sie Text zeichnen, sollten Sie diese Farbe als Vordergrundfarbe für den Text verwenden.

Dieser Wert kann vom Anwender geändert werden, weshalb Sie in jedem Zeichen-Event diesen Wert neu abrufen sollten, statt diesen Wert einmal abzurufen und zu sichern.

Siehe auch

[DarkBevelColor](#), [DarkTingeColor](#), [FillColor](#), [FrameColor](#), [HighlightColor](#), [LightBevelColor](#), [LightTingeColor](#) Funktionen; [Color](#) Datentyp.

TextConverter Klasse

Wird verwendet, um Text von einem Kodierungssystem in ein anderes zu konvertieren.

Methoden

Name	Parameter	Beschreibung
<i>Clear</i>		Löscht den Zustand des TextConverter-Objekts für die Konvertierung eines neuen Strings.
<i>Convert</i>	InputText als <u>String</u>	Konvertiert InputText

Beispiel

Folgendes Beispiel konvertiert den Text in einem TextField:

```
Dim c as TextConverter
c=GetTextConverter(GetTextEncoding(&h500), GetTextEncoding(0))
EditField2.text=c.convert(EditField1.text)
```

Siehe auch

GetTextConverter; ConvertEncoding; DefineEncoding; Encoding Funktionen; TextEncoding Klasse; Encodings Objekt.

TextEncoding Klasse

Wird beim Text-Encoding zum Festlegen der Textkodierung eines String verwendet.

Eigenschaften

Name	Typ	Beschreibung
<u>Base</u>	<u>Integer</u>	Der Kodierungstyp.
<i>Code</i>	<u>Integer</u>	(Mac OS). Der Mac OS-TextEncoding-Wert, nützlich für Declares. Sie können dies ebenso verwenden, um zwei Encodings zu vergleichen: Wenn die Code-Eigenschaften zweier TextEncoding-Objekte gleich sind, präsentieren sie die gleiche Textkodierung (inklusive Base, Variation und Format).
<u>Variant</u>	<u>Integer</u>	Eine Variante des <i>Base</i> -Text-Encodings
<u>Format</u>	<u>Integer</u>	Ein Format für das <i>Base</i> -Text-Encoding. Wird von Unicode für die Festlegung des Unicode-Formats verwendet, das benutzt werden soll.
<i>InternetName</i>	<u>String</u>	Name der Internet-Text-Kodierung.

Methoden

Name	Parameter	Beschreibung
Chr	CodePoint als <u>Integer</u>	Liefert ein Zeichen in der angegebenen Kodierung. Der „Code-Punkt“ des Zeichens ist der gleiche wie der Wert, den Asc liefert. Um z.B. einen String zu erhalten, der das Apple-Logo in UTF-8 enthält, müssen Sie folgendes verwenden: s = Encodings.UTF8.Chr(&hF8FF). Es ist sicherer diese Methode statt der Chr-Funktion zu verwenden, außer Sie sind sich sicher, daß Sie mit ASCII-Codes arbeiten(0-127).
Equals	otherEncoding als TextEncoding	Vergleicht das angegebene Encoding mit dem übergebenen Encoding. Liefert ein <u>Boolean</u> .

Hinweise

Wenn ein Computer Text speichert, wird jedes Zeichen als numerischer Wert gespeichert. Wenn diese Zeichen nun angezeigt oder gedruckt werden müssen, wird eine bestimmte Kodierung verwendet, um festzustellen, welches Zeichen für diesen Wert ausgegeben werden muß.

Die ersten Computer verwendeten eine Kodierung die ASCII genannt wurde. ASCII steht für American Standard Code for Information Exchange. Es definiert 128 Werte, die Groß- und Kleinbuchstaben umfassen, sowie Ziffern und einige Symbole und nicht druckbare Zeichen. Die ASCII-Kodierung wird im Anhang A der PDF-Version oder gedruckten Sprachreferenz beschrieben.

Die Computer wurden weiterentwickelt und wurden auch in nicht-englisch-sprachige Länder eingeführt und man wurde sich der Grenzen der ASCII-Kodierung bewußt. Sie enthält keine Zeichen, die einen Akzent benötigen oder ideografische Sprachen wie Japanisch oder Chinesisch, die tausende Zeichen benötigen.

Im Ergebnis wurde die ASCII-Kodierung erweitert. Oberhalb des Bereichs von 0 bis 127 sind die Zeichen aber von Kodierung zu Kodierung unterschiedlich. Zum Beispiel unterscheiden sich die Zeichen zwischen Windows und Mac OS in dem Bereich von 128 bis 255. Es wurde weitere Kodierungen entwickelt, die diesen Bereich unterschiedlich interpretieren, aber nicht-ASCII-Zeichen unterstützen.

Eine allgemein verwendete Lösung dieses Problems stellt die Kodierung namens Unicode dar. Sie wurde entwickelt, um alle möglichen Zeichen aller Sprachen verarbeiten zu können. Sie ermöglicht sogar eine Mixtur von Sprachen innerhalb einer Zeichenkette. Wie auch immer. Nicht alle Strings, mit denen Sie zu tun haben, verwenden Unicode.

Wenn Sie einen String verwenden, müssen Sie dessen Kodierung kennen, um die Aneinanderreihung von Bytes korrekt zu interpretieren, die letztendlich den String darstellen. In REALbasic besteht ein String aus zwei Bestandteilen. Einmal der Text und zusätzlich dessen verwendete Kodierung. REALbasic unterstützt zwei Formate der Unicode-Kodierung. Diese sind UTF-8 und UTF-16. Alle Strings in REALbasic werden als UTF-8 compiliert. Dieses Unicode-Format verwendet ein Byte für ASCII-Zeichen und bis zu vier Bytes für nicht-ASCII-Zeichen.

Wenn Sie mit Strings arbeiten, die in REALbasic erzeugt wurden, müssen Sie sich keine Gedanken über deren Kodierung machen, da REALbasic grundsätzlich UTF-8 verwendet. Für Strings, die nicht in REALbasic erzeugt wurden, die zum Beispiel über das Internet eingelesen werden, müssen Sie eine Kodierung angeben, damit REALbasic die Bytefolge korrekt interpretieren kann.

Dies können Sie auf unterschiedlichen Wegen bewerkstelligen. Wenn Sie zum Beispiel einen Text mit Hilfe der `TextInputStream`-Klasse einlesen, können Sie die `Encoding`-Eigenschaft verwenden, indem Sie dieser ein `Encodings`-Objekt zuweisen, das die gewünschte Kodierung repräsentiert. Hier ist ein Beispiel, das einen Text mit der MacRoman-Kodierung einliest:

```
Dim f As FolderItem
Dim t As TextInputStream
f=GetOpenFolderItem("text") //Dateityp, der im Dateitypen-Dialog festgelegt wurde
If f <> Nil then
    t=f.OpenAsTextFile
    t.Encoding=Encodings.MacRoman //Encoding des Input-Streams festlegen.
    EditField1.text=t.ReadAll
    t.Close
End if
```

Auch die `Read`-, `ReadLine`- oder `ReadAll`-Methoden akzeptieren einen optionalen Parameter für die Kodierung.

Wenn Sie einen String in einer bestimmten Kodierung ausgeben müssen, können Sie die Funktion `ConvertEncoding` verwenden. Dieses Beispiel konvertiert den Text in einem `EditField` in die Kodierung `WindowsANSI`:

```
Dim s as String
s=EditField1.Text.ConvertEncoding(Encodings.WindowsANSI)
```

Sie werden Textkodierung hilfreich finden, wenn Sie sich mit folgenden Anwendungsbereichen beschäftigen:

- Internet-Applikationen, also Web-Browser oder Email-Programme
- Applikationen, die Text über Plattformgrenzen hinweg übertragen
- Applikationen, die auf Unicode basieren

Beispiel

Die `Encoding`-Funktion macht es einfach, das `TextEncoding` eines Strings zu ermitteln. Verwenden Sie das `Encodings`-Objekt, um ein bestimmtes Text-Encoding zu erhalten. Die gebräuchlichsten sind UTF8, UTF16, UCS4, ASCII, MacRoman, MacJapanese und WindowsLatin1. Verwenden Sie die Auto-Vervollständigen-Funktion im Code-Editor, um sich die komplette Liste anzeigen zu lassen.

Beispiele

Das folgende Beispiel ermittelt das Text-Encoding des der `Encoding`-Funktion übergebenen Strings.

```
Dim t as TextEncoding
t=Encoding(Editfield1.text)
If t <> Nil then
    staticText1.text="Base="+Str(t.base)
    staticText2.text="Format="+Str(t.format)
    staticText3.text="Variant="+Str(t.variant)
end if
```

Folgender Befehl verwendet das **Encodings**-Objekt, um das UTF8-Text-Encoding für einen Text in einem EditField zu ermitteln.
`Editfield2.text=DefineEncoding(EditField1.text, Encodings.UTF8)`

Siehe auch

Appendix A, GetTextEncoding, GetInternetTextEncoding, Encoding, GetTextConverter, ConvertEncoding, DefineEncoding, Chr Funktionen; Encodings Objekt.

TextInputStream Klasse

Um Text aus einer Datei lesen zu können, müssen Sie vorher ein TextInputStream Objekt anlegen. TextInputStreams bieten Methoden zum Lesen aus einer Datei, zum Überprüfen, ob Sie am Dateiende angelangt sind und zum Schließen der Datei. TextInputStreams werden mit der Methode OpenAsTextFile und einem FolderItem angelegt. Wenn Sie mit einer Datei arbeiten, deren Kodierung nicht UTF-8 ist, sollten Sie die Encoding-Eigenschaft entsprechend setzen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>EOF</i>	<u>Boolean</u>	Der Stream hat das Ende der Datei erreicht.
Encoding	<u>TextEncoding</u>	Legt die Kodierung fest, die für einen String, der von ReadLine oder ReadAll zurückgeliefert wird, festgelegt werden soll. Es konvertiert nicht die Bytes, sondern weist diesen nur ein Encoding zu, so als ob Sie <u>DefineEncoding</u> aufgerufen hätten. Verwenden Sie das <u>Encoding</u> -Objekt, um das Text-Encoding festzulegen. Die Voreinstellung ist UTF-8, Sie können aber ein anderes Encoding zuweisen, das zu Ihrer Datei paßt oder auch <u>Nil</u> zuweisen, falls aus irgendwelchen Gründen möchten, daß die Strings ein undefiniertes Encoding besitzen.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Schließt die Datei.
<i>ReadAll</i>	[Encoding] als <u>TextEncoding</u>	Liefert den gesamten Text aus dem TextInputStream (als <u>String</u>). Der optionale <i>Encoding</i> -Parameter ermöglicht es, die Kodierung des Textes festzulegen. Sie können das Encoding auch über die Encoding-Eigenschaft setzen.
<i>ReadLine</i>	[Encoding] als <u>TextEncoding</u>	Liefert die nächste Zeile aus dem TextInputStream (als <u>String</u>). Der optionale <i>Encoding</i> -Parameter ermöglicht es, die Kodierung des Textes festzulegen. Sie können das Encoding auch über die Encoding-Eigenschaft setzen.

Hinweise

Die **TextInputStream**-Klasse implementiert das „Readable“-Klassen-Interface. Wenn Sie dieses selbst implementieren wollen, müssen Sie eine Read-Methode anbieten, die der folgenden entspricht:

Ergebnis=**Read** (count, [enc])

Teil	Typ	Beschreibung
Ergebnis	<u>String</u>	Der Text, der zurückgegeben wird.
count	<u>Integer</u>	Die Anzahl der Bytes, die gelesen werden sollen.
enc	<u>TextEncoding</u>	Optionaler Parameter, mit dem die Kodierung des Strings angegeben werden kann.

Das Readable-Klassen-Interface besitzt zwei Methoden.

Name	Parameter	Beschreibung
EOF		Liefert <u>True</u> , wenn das Ende des Streams erreicht ist oder der Socket geschlossen ist und keine Daten im internen Puffer vorhanden sind.
ReadError		Liefert <u>True</u> , wenn ein Fehler aufgetreten ist.

Für weitere Informationen lesen Sie bitte das Kapitel „Klassen-Interface“ im Benutzerhandbuch durch.

Wenn Sie eine Textdatei von einem anderen Betriebssystem oder einer anderen Sprache einlesen, müssen Sie die Kodierung angeben, die beim Speichern dieser Datei verwendet wurde. Wenn Ihnen die Kodierung bekannt ist, können Sie der Encoding-Eigenschaft des **TextInputStream**-Objekts ein Encoding-Objekt zuweisen. Hier ist ein Beispiel, das eine Datei mit der Kodierung MacRoman einliest:

```
Dim f As FolderItem
Dim t As TextInputStream
f=GetOpenFolderItem("text") //Dateityp, der im Dateitypen-Dialog festgelegt wurde
If f <> Nil Then
    t=f.OpenAsTextFile
    t.Encoding=Encodings.MacRoman //Encoding des Input-Streams festlegen
    EditField1.Text=t.ReadAll
    t.Close
End if
```

Um eine Kodierung anzugeben, können Sie auch den optionalen Parameter der ReadAll-Methode verwenden:

```
EditField1.Text=t.ReadAll(Encodings.MacRoman)
```

anstatt:

```
t.Encoding=Encodings.MacRoman
```

Beispiele

Dieses Beispiel liest die Zeilen und Spalten mit Daten aus einer Datei, die diese, mit Tabulator getrennt, enthält und zeigt diese in einer ListBox an:

```
Dim f As FolderItem
Dim textInput As TextInputStream
Dim rowFromFile,oneCell As String
Dim i As Integer

f=GetOpenFolderItem("text/plain")

If f <> Nil Then
    textInput = f.OpenAsTextFile
    textInput.Encoding=Encodings.MacRoman //Strings liegen in MacRoman vor
    Do
        rowFromFile=textInput.ReadLine
        If ListBox1.ColumnCount < CountFields(rowFromFile,Chr(9)) Then
            ListBox1.ColumnCount=CountFields(rowFromFile,Chr(9))
        End If
        ListBox1.AddRow NthField(rowFromFile,Chr(9),1)
        For i=1 to CountFields(rowFromFile,Chr(9))
            oneCell=NthField(rowFromFile,Chr(9),i)
            ListBox1.Cell(ListBox1.ListCount-1,i-1)=oneCell
        Next
    Loop Until textInput.EOF
    textInput.Close
End If
```

Siehe auch

ConvertEncoding, DefineEncoding, Encoding Funktionen; BinaryStream, TextOutputStream, TextEncoding Klassen; Encodings Objekt.

TextOutputStream Klasse

Um in eine Datei schreiben zu können, müssen Sie vorher ein TextOutputStream-Objekt anlegen. TextOutputStreams kennen Methoden, die in eine Datei schreiben und diese schließen können, wenn Sie mit dem Schreiben fertig sind. Sie werden mit den Methoden CreateTextFile und AppendToTextFile sowie einem FolderItem aufgerufen. Müssen Sie eine bestimmte Kodierung angeben, so rufen Sie bitte vor dem Schreiben der Datei ConvertEncoding auf.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>Delimiter</i>	<u>String</u>	Das Zeichen, das ein Zeilenende in der Datei markiert. Carriage Return ist voreingestellt.

Methoden

Name	Parameter	Beschreibung
<i>Close</i>		Schließt die Datei.
<i>Write</i>	Text als <u>String</u>	Schreibt den Text in den <code>TextOutputStream</code> und damit in die Datei.
<i>WriteLine</i>	Text als <u>String</u>	Fügt den Delimiter (das Zeilenendezeichen) ans Ende der Zeile an und schreibt den Text in den <code>TextOutputStream</code> und damit in die Datei.

Hinweise

Die **TextOutputStream**-Klasse implementiert das "Writable" Klassen-Interface. Wenn Sie dies selbst implementieren wollen, müssen Sie eine Write-Methode anbieten, die der folgenden entspricht.

Write(data)

Teil	Typ	Beschreibung
data	<u>String</u>	Der Text, der geschrieben werden soll.

Das Writable-Klassen-Interface besitzt die folgenden Methoden:

Name	Parameter	Beschreibung
Flush		Schreibt die verbleibenden Daten im internen Puffer auf das Volume und löscht den Puffer.
WriteError		Liefert <u>True</u> , wenn ein Fehler auftrat.

Für weitere Informationen über Klassen-Interfaces und wie diese implementiert werden, finden Sie im Benutzerhandbuch im Kapitel „Klassen-Interfaces“.

Wenn Sie eine Datei in einer bestimmten Kodierung schreiben müssen, können Sie ConvertEncoding verwenden, um den Text zu konvertieren, bevor Sie ihn an die Write- oder WriteLine-Methode übergeben. Hier ist ein Beispiel:

```
Dim f As FolderItem
Dim t As TextOutputStream
f = GetFolderItem("Sample.txt")
t = f.CreateTextFile
t.Write ConvertEncoding(Editfield1.text, Encodings.MacRoman)
t.close
```

Beispiel

Dieses Beispiel zeigt den Datei-Sichern-Dialog. Danach wird eine Textdatei angelegt und die Text-Eigenschaften dreier EditFields werden in die neue Datei geschrieben. Schließlich wird die Datei geschlossen.

```
Dim file As FolderItem
Dim fileStream As TextOutputStream
file=GetSaveFolderItem("application/text","Mein Info")
fileStream=file.CreateTextFile
fileStream.WriteLine NameFeld.text
fileStream.WriteLine AdresseFeld.text
fileStream.WriteLine TelefonFeld.text
fileStream.Close
```

Siehe auch

BinaryStream, TextInputStream, FolderItem Klassen; ConvertEncoding Funktion; Encodings Objekt.

Thread Klasse

Threads werden verwendet, um unabhängig von der Benutzeroberfläche im Hintergrund Code auszuführen.

Übergeordnete Klasse

Object

Events

Name	Parameter	Beschreibung
<i>Run</i>		Der Code, der ausgeführt werden soll, wenn die Run-Methode aufgerufen wird.

Eigenschaften

Name	Parameter	Beschreibung
<i>StackSize</i>	<u>Integer</u>	Größe des Stacks in Bytes. Kann festgelegt werden, wenn der Thread nicht läuft. Die Default-Stackgröße ist 64K. Zuvor war sie 256K. Eine Stackgröße von Null bedeutet die Voreinstellung.
<i>ThreadID</i>	<u>Integer</u>	ID des Threads, zugewiesen zur Laufzeit.

Methoden

Name	Parameter	Beschreibung
<i>Run</i>		Führt den Run-Event-Handler des Threads aus.

Hinweise

Objekte der Thread-Klasse werden verwendet, um Code auszuführen, der unabhängig von der Benutzeroberfläche im Hintergrund läuft. Wenn Sie beispielsweise ein komplexes Bild zu zeichnen hätten, dessen Berechnung mehrere Minuten dauert, könnten Sie den Code hierzu in einem Thread ablaufen lassen, so daß der Anwender weiterhin im Vordergrund die Benutzeroberfläche benutzen kann, ohne auf das Ende der Berechnungen zu warten.

Ein Thread kann ebenso verwendet werden, um eine sehr zeitaufwendige Schleife im Hintergrund ausführen zu lassen. Da Schleifen, die nicht als Thread laufen, das gesamte Interface blockieren und den Anwender hindern, Menüs oder Steuerelemente zu bedienen, ist dies eine gute Lösung, dem Anwender zu erlauben, das Programm weiter bedienen zu können, während ein längerer Prozeß abläuft. Siehe Schleifen [For](#), [While](#), and [Do](#).

Threads geben an andere Threads und andere Applikation immer dann Zeit ab, wenn sie ein Schleifen-Konstrukt wie [For](#), [While](#) und [Do](#) ausführen.

Um einen Thread anzulegen, legen Sie eine neue Klasse an und machen Sie **Thread** zur übergeordneten Klasse. Schreiben Sie den auszuführenden Code in den Event-Handler „Run“. Um diesen Thread auszuführen, legen Sie lediglich ein neues Thread-Objekt an und rufen Sie dessen Run-Methode auf. Sie können dies, wie im Beispiel demonstriert, durch Hinzufügen der Instanz der Thread-Klasse zu einem Fenster oder auf üblichem Weg machen.

Sie können die DoEvents-Methode aufrufen, um Laufzeit an REALbasic zurückzugeben.

Beispiel

Folgender Thread namens Calculate, ruft eine Funktion in einem Module auf, die eine zeitaufwendige Berechnung ausführt.

```
Dim r as double
r=Interpolate(1, 100,000,000)
```

Um die Berechnung auszuführen, draggen Sie die Thread-Klasse, Calculate, aus dem Projekt-Fenster in ein Fenster. Dies erzeugt eine Instanz des Threads, Calculate1. Fügen Sie ein Steuerelement einem Fenster hinzu, das folgenden Code besitzt:

```
Calculate1.run
```

Sie können die ID dieser Instanz des Threads ermitteln, indem Sie folgenden Wert auslesen:

```
Calculate1.ThreadID
```

Statt daß Sie die Calculate-Klasse in ein Fenster draggen, können Sie die Instanz auch mittels Code erzeugen und seine Run-Methode aufrufen:

```
Dim t as Calculate
t=new Calculate
t.run
```

Siehe auch

[Semaphore](#), [CriticalSection](#), [Application](#) Klassen; [#Pragma](#) Direktive.

Ticks Funktion

Liefert die Anzahl an Ticks (60tel einer Sekunde), die der Computer seit dem Einschalten arbeitet.

Syntax

Ergebnis=**Ticks**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Anzahl von Ticks, die der Computer, seit dem Einschalten, arbeitet.

Beispiele

Dieses Beispiel zeigt in einer MessageBox an, seit wie vielen Minuten der Rechner läuft.

```
Dim minutes As Integer
minutes=Ticks/60/60
Msgbox "Ihr Computer ist seit "+Str(minutes)+" Minuten eingeschaltet."
```

Siehe auch

Microseconds Funktion.

Timer Objekt



Ein **Timer** ist ein Objekt, das nach einer definierten Zeitspanne oder in einem wiederholten Intervall Code ausführt.

Übergeordnete Klasse

Object

Da es von Object, nicht von RectControl abgeleitet ist, können Sie es auch im Code instanzieren.

Eigenschaften

Name	Art	Beschreibung
<i>Enabled</i>	<u>Boolean</u>	Erlaubt es Ihnen, den Timer ein- und auszuschalten. Wenn Enabled <u>True</u> ist, ist der Timer eingeschaltet.
<i>Index</i>	<u>Integer</u>	Die Nummer des Steuerelements, wenn es Teil eines Steuerelement-Arrays ist.
<i>Left</i>	<u>Integer</u>	Die linke Seite des Steuerelements in lokalen Koordinaten (relativ zum Fenster).
<i>Mode</i>	<u>Integer</u>	Die Intervallart, in der das Event „Action“ ausgeführt wird. 0=nie, 1=einmal, 2=mehrmals. Voreinstellung ist 2.
<i>Name</i>	<u>String</u>	Der Objektname.
<i>Period</i>	<u>Integer</u>	Die Zeit zwischen den Aufrufen (in Millisekunden). Ein Period-Wert kleiner gleich null setzt Period auf den Wert von einer Millisekunde. Voreinstellung ist 1000. Die Rate, in der ein Timer Signale senden kann, hängt von der Geschwindigkeit des Rechners, dem Betriebssystem und anderen Aufgaben ab, die der Rechner gerade erledigen muß. Es ist möglich, Period auf einen Wert zu setzen, den der Rechner, auf dem das Programm läuft, nicht erreichen kann.
<i>Top</i>	<u>Integer</u>	Die obere Seite des Steuerelements in lokalen Koordinaten (relativ zum Fenster).

Methoden

Name	Parameter	Beschreibung
<i>Reset</i>		Setzt den Timer zurück und startet ihn neu.

Events

Name	Parameter	Beschreibung
<i>Action</i>		Der Code, der nach der angegebenen Zeit im angegebenen Modus ausgeführt wird.

Hinweise

Die Eigenschaft Mode bestimmt das Intervall, in dem der Event-Handler „Action“ aufgerufen wird. Ist der Mode nicht 0 (null), so wartet das Timer-Objekt die angegebene Zeitspanne ab, bevor der Event-Handler „Action“ aufgerufen wird. Wird Mode zur Laufzeit auf 0 (null) gesetzt, so bricht das Timer-Objekt das Warten sofort ab und der Event-Handler wird nicht ausgeführt.

Das Timer-Objekt führt seinen Event-Handler „Action“ aus, egal, ob sein Fenster das oberste ist oder nicht (unter der Voraussetzung, daß Mode nicht auf 0 gesetzt wurde). Auch die Sichtbarkeit des Fenster hat auf die Ausführung des Event-Handlers „Action“ keinerlei Auswirkungen. Der Timer wurde so implementiert, daß er Zeit an andere laufende Applikationen abgibt, so daß der Rechner nicht ins Stocken gerät.

Siehe auch

Object Klasse.

Titlecase Funktion

Gibt von einem übergebenen String alle Wörter mit großem Anfangsbuchstaben und den Rest des Wortes in Kleinbuchstaben zurück, weil man im englischsprachigen Raum Überschriften so schreibt.

Syntax

Ergebnis = Titlecase(Wert)

ODER

Ergebnis = StringVariable.Titlecase

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>String</u>	Eine Kopie des Originalstrings mit allen passend konvertierten Zeichen.
<i>Wert</i>	<u>String</u>	Der zu verändernde String.
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ String.

Hinweis

Konvertiert alle Buchstaben in einem String zuerst in kleingeschriebene Zeichen und dann den ersten Buchstaben eines jeden Wortes in Großbuchstaben. Zahlen bleiben unverändert.

Beispiele

Dieses Beispiel konvertiert den String.

```
Dim s As String
s=Titlecase("der Schnelle fUCHS") //liefert "Der Schnelle Fuchs"
s=Titlecase("DIE 5 LAHMEN HUNDE") //liefert "Die 5 Lahmen Hunde"
```

Siehe auch

Lowercase, Uppercase Funktionen.

ToolStripItem Steuerelement



Wird zum Erzeugen eines Toolbar in einem Fenster verwendet. Wird nur unter Mac OS X ab 10.2 unterstützt. ToolStripItems können für andere Plattformen in Fenstern nur für Layoutzwecke positioniert werden.

Übergeordnete Klasse

Control

Eigenschaften

Name	Typ	Beschreibung
<i>Caption</i>	<u>String</u>	Das Textlabel unter dem ToolStripItem. Siehe Hinweise.
<i>ControlOrder</i>	<u>Integer</u>	Legt die linke oder rechte Platzierung der Icons auf der Toolbar fest.
<i>Enabled</i>	<u>Boolean</u>	True, falls das ToolStripItem aktiviert ist.
<i>Handle</i>	<u>Integer</u>	Ein toolStripItemRef, das in <u>Declare</u> -Befehlen verwendet wird; nur für Toolbox-Programmierer nützlich.
<i>HelpTag</i>	<u>String</u>	Standard-Helptag für das Objekt.

Name	Typ	Beschreibung
<i>HelpTagLong</i>	<u>String</u>	Helptag, das erscheint, wenn der Anwender die Apfel-Taste gedrückt hält.
<i>Image</i>	<u>Picture</u>	Das Bild, das durch das ToolbarItem angezeigt werden soll. Wenn dem ToolbarItem kein Bild zugewiesen ist, wird ein „?“-Icon angezeigt.
<i>Left</i>	<u>Integer</u>	Der Abstand (in Pixeln) von der linken Seite des Fensters im Fenster-Editor. Im selbständig laufenden Programm basiert die Reihenfolge der Toolbar-Objekte strikt auf der von links nach rechts Reihenfolge.
<i>Top</i>	<u>Integer</u>	Der Abstand (in Pixeln) oberen Rand des Fensters im Fenster-Editor. Im selbstständig laufenden Programm basiert die Reihenfolge der Toolbar-Objekte strikt auf der von links nach rechts Reihenfolge.

Events

Name	Parameter	Beschreibung
<i>Action</i>		Tritt auf, wenn der Anwender auf das ToolbarItem klickt.
<i>Open</i>		Tritt auf, wenn das Objekt anfangs erzeugt wird.
<i>Close</i>		Tritt auf, wenn ein Objekt zerstört wird.

Methoden

Name	Parameter	Beschreibung
<i>SetImageFromIcon</i>	Creator als <u>String</u> Type als <u>String</u>	Weist ein Bild der Image-Eigenschaft unter Verwendung eines im System registrierten Icons unter Verwendung der übergebenen Type und Creator.

Im Folgenden ist eine Liste der gängigsten Icons, die mit SetImageFromIcon verwendet werden:

Creator	Typ	Beschreibung
macs	tdel	Löschen
macs	tfav	Favoriten
macs	thom	Privat
macs	burn	Brennen
macs	ejec	Auswerfen
macs	CLIP	Clipboard
macs	desk	Schreibtisch
macs	FNDR	Finder
macs	trsh	Papierkorb
macs	ftrh	Voller Papierkorb
macs	docu	Dokument
macs	gurl	URL
macs	udsk	iDisk
macs	ilht	HTTP
macs	ilft	FTP
macs	gnet	Netzwerk

Eine komplette Liste der von Apple angebotenen Icons finden Sie in Icons.h bei den Universal Headers.

Hinweis

Falls die Caption-Eigenschaft ein Kaufmanns-Und ist, wird es nur angezeigt, wenn eine weiteres Kaufmanns-Und vorangestellt wird. Wenn Sie die Caption-Eigenschaft z.B. auf „Bob & Ray“ setzen, wird nur „Bob Ray“ angezeigt. Sie müssen es auf „Bob && Ray“ setzen. Das Kaufmanns-Und-Zeichen wird unter Windows für Tastenkürzel verwendet.

Siehe auch

[StandardToolBarItem](#) Steuerelement.

TrashFolder Funktion

Wird benutzt, um Zugriff auf die Einträge im Papierkorb zu bekommen. Unter Windows liefert der Aufruf von TrashFolder ein [FolderItem](#), das auf den Papierkorb des Anwenders verweist. Unter Linux wird auf das Verzeichnis „Trash“ im Home-Verzeichnis des Benutzers zugegriffen. Existiert dieses Verzeichnis nicht wird [Nil](#) zurückgegeben.

Syntax

Ergebnis = **TrashFolder**

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Ein folderItem , das den Papierkorb repräsentiert.

Hinweis

Der Papierkorb wird auf dem Finder-Schreibtisch durch ein Symbol repräsentiert. Wenn ein Anwender Objekte in den oder aus dem Papierkorb zieht, werden diese in Wirklichkeit in den oder aus dem Trash-Ordner gezogen.

Die Funktion **TrashFolder** bietet eine Möglichkeit, auf den Papierkorb zuzugreifen, die auch unter anderen Sprachversionen des Mac OS funktioniert.

Beispiele

Bei der Beschreibung der Funktion [AppleMenuFolder](#) finden Sie ein passendes Beispiel.

Siehe auch

[AppleMenuFolder](#), [ApplicationSupportFolder](#), [ControlPanelsFolder](#), [DesktopFolder](#), [ExtensionsFolder](#), [FontsFolder](#), [PreferencesFolder](#), [StartupItemsFolder](#), [SystemFolder](#), [TemporaryFolder](#) Funktionen.

Trim Funktion

Liefert den übergebenen String ohne führende und nachfolgende Leerzeichen zurück.

Syntax

Ergebnis = **Trim**(quellString)

ODER

Ergebnis = **StringVariable.Trim**

Teil	Art	Beschreibung
<i>Ergebnis</i>	String	Der quellString ohne führende oder nachfolgende Leerzeichen.
<i>quellString</i>	String	Der String, der um führende und nachfolgende Leerzeichen bereinigt werden soll.
<i>StringVariable</i>	String	Beliebige Variable vom Typ String.

Beispiele

Dieses Beispiel entfernt führende wie nachfolgende Leerzeichen des übergebenen Strings:

```
Dim s as String
s=Trim("  Hallo Welt  ") //liefert „Hallo Welt“
```

Siehe auch

[LTrim](#), [RTrim](#), [Asc](#), [Chr](#), [InStr](#), [Len](#), [Left](#), [Mid](#), [Right](#) Funktionen.

True Schlüsselwort

Wird verwendet, um ein [boole'sches](#) Objekt auf den Wert True zu setzen oder um zu testen, ob ein vorhandener [boole'scher](#) Ausdruck gleich „True“ ist. Ein Vergleich von zwei Werten, die gleich sind, ergibt True.

Syntax

expression=**True**

Expression ist ein beliebiger, gültiger boole'scher Ausdruck.

Siehe auch

False Schlüsselwort, And, Or, Not Operatoren.

Try Block

Wird verwendet, um eine RuntimeException vor dem Exception Block zu behandeln.

Syntax

Try

```
//REALbasic-Befehle-- Code der eine Ausnahme auslösen kann
Catch [ErrorParameter] [As ErrorType]
//exception handlers -- behandeln der Ausnahme
[Finally]
//Code, der ausgeführt wird, auch wenn eine Ausnahme ausgelöst wurde
End [Try]
```

Teil	Beschreibung
<i>ErrorParameter</i>	Optional, um den Typ der Ausnahme zu ermitteln
<i>ErrorType</i>	Optional, wenn benutzt, muss es mit <i>ErrorParameter</i> verwendet werden. Damit kann ein bestimmter Ausnahmetyp abgefangen werden. Der Try-Block braucht dann nur diesen Ausnahmetyp behandeln.

Hinweise

Der **Try**-Block ist ähnlich zum Exception-Block. Ausnahmen, die in einem **Try**-Block auftreten, werden mit dem Catch-Befehl abgefangen. Es ist die gleiche Syntax wie die des Exception-Befehls. Lesen Sie die Beschreibung des RuntimeException-Befehls für eine Liste der Fehler und deren Erläuterungen.

Im Unterschied zum Exception-Block kann der **Try**-Blocks verschachtelt werden. Wenn ein **Try**-Block einen Ausnahmetyp nicht abhandelt, kann dies der äußere Block abhandeln oder der Exception-Block selbst, wenn keine weiteren **Try**-Blöcke vorhanden sind.

Ein **Try**-Block kann einen eigenen Finally-Befehl haben. Der Code im Finally-Teil wird selbst dann ausgeführt, wenn keine Ausnahme im Try-Block auftrat. Der Finally-Befehl wird bei den Befehlen Sub und Function erläutert.

Beispiel

Dieses Beispiel verwendet einen Catch-Befehl in einem Open-Event von einem Fenster, um Speicher-Ausnahmen (memory exception) abzuhandeln, die beim Zeichnen eines importierten GIF-Bilds auftreten könnten. Die Variable myPicture ist eine globale Eigenschaft vom Typ Picture. Das Bild „Logo“ wurde in das Projektfenster gezogen.

```
Sub Open
myPicture=New Picture(Logo.width,Logo.height,32)
myPicture.Graphics.DrawPicture Logo,0,0
Catch err as OutOfMemoryException
  MsgBox "Nicht genug Speicher um das Bild zu zeichnen!"
  Quit
End Sub
```

Siehe auch

Exception Block, Sub Befehl, Function Befehl. RuntimeException Klasse.

TypeMismatchException Fehler

Tritt auf, wenn Sie versuchen, einem Objekt einen Wert zuzuweisen, das einen falschen Datentyp hat.

Übergeordnete Klasse

RuntimeException

Hinweise

Ein **TypeMismatchException** Fehler tritt auf, wenn Sie versuchen, einen Wert von einem nicht korrekten Datentyp einem Objekt zuzuweisen. Dieser Fehler kann nur auftreten, wenn der Compiler den Typ des Wertes während des Compilierens nicht feststellen kann – z.B., wenn Sie Varianten verwenden. Normalerweise aber erkennt der Compiler falsche Typen, wenn Sie versuchen, die Applikation zu compilieren.

Beispiel

Folgender Code weist ein Bild, das ins Projekt-Fenster gedraggt wurde, der Variante „v“ zu und versucht dann, „v“ einem Integer zuzuweisen. Der Exception-Handler zeigt eine Wanbox, die es dem Entwickler erlaubt, dem Problem auf den Grund zu gehen. Exception-Handling bewahrt eine erzeugte Applikation vor dem Beenden, wenn der Fehler auftritt.

```
Dim v as Variant
Dim i as Integer
v=Produktlogo // ein Bild
i=v
Exception err
If err IsA TypeMismatchException then
    MsgBox "Der Variablen i muß eine Zahl zugewiesen werden!"
end if
```

Siehe auch

Falscher Typ Fehler; RuntimeException Klasse; OutOfBoundsException, IllegalCastException, StackOverflowException, NilObjectException Fehler; Function, Raise Befehle, Nil Schlüsselwort, Exception Block.

Ubound Funktion

Liefert den Index des letzten Elements eines Arrays.

Syntax

Ergebnis = **Ubound**(array, [Dimension])

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Der Index des letzten Elements des angegebenen Arrays.
<i>array</i>	Array-Objekt	Das Array.
<i>Dimension</i>	<u>Integer</u>	Optional: Nur bei mehrdimensionalen Arrays wichtig. Wird verwendet, um die Dimension anzugeben, von der Sie das letzte Element benötigen. Die erste Dimension trägt die Nummer 1. Wenn -1 übergeben wird, wird die Anzahl der Dimensionen des Arrays geliefert. Wenn eine nicht vorhandene Dimension übergeben wird, wird ein OutOfBoundsException Fehler ausgelöst.

Hinweis

Die Funktion **Ubound** wird verwendet, um das letzte Element eines Arrays zu ermitteln, kann aber auch dazu verwendet werden, die Größe eines Arrays festzustellen. Der Unterschied liegt im Detail: Alle Arrays haben ein nulltes Element. In einigen Fällen wird das nullte Element verwendet, in anderen nicht. Wenn Sie die Funktion **Ubound** verwenden, müssen Sie beachten, um was für ein Array es sich handelt, wenn Sie die Anzahl der Werte im Array wissen möchten. Ist das Array nullbasiert, dann wird das Element Null verwendet, um einen Wert zu sichern. In diesem Fall müssen Sie zu dem von **Ubound** gelieferten Wert eins hinzuzählen.

Beispiele

Dieses Beispiel ersetzt in einem Array alle Vorkommen von X durch Y.

```
Dim i As Integer
For i=0 to Ubound(Names)
    If Names(i)="X" Then
        Names(i)="Y"
    End If
Next
```

Siehe auch

Append, IndexOf Methoden; Dim Befehl; Insert Methode; Redim Methode; Remove Methode; Sort Methode.

Dim Befehl; Array, Split, Join Funktionen; Append, Insert, Redim, Remove, Pop, IndexOf, Insert, Shuffle, Sort Methoden; ParamArray Schlüsselwort.

UDPSocket Klasse

Wird für UDP-Kommunikation verwendet (User Datagram Protocol).

Übergeordnete Klasse

[SocketCore](#) Klasse.

Eigenschaften

Name	Typ	Beschreibung
<i>BroadcastAddress</i>	<u>String</u>	Die Broadcast-Adresse des Rechners. Sie können diese Adresse bei einem Write festlegen und die Daten werden über das Netzwerk übertragen (Sie erhalten die Daten, die Sie senden, jedoch nicht zurück).
<i>PacketsAvailable</i>	<u>Integer</u>	Die Anzahl verfügbarer Pakete im internen Empfangspuffer.
<i>RouterHops</i>	<u>Integer</u>	Legt fest, wie viele Router-Sprünge die versendeten Daten machen dürfen. Das ist auch als Time To Live (oder TTL) bekannt. Diese Eigenschaft kann nur auf Multicast-Sendungen angewendet werden.
<i>SendToSelf</i>	<u>Boolean</u>	Legt fest, ob die Daten, die Sie versenden, auch zu Ihnen geschickt werden. Das ist auch als Loopback bekannt. Diese Eigenschaft kann nur auf Multicast-Sendungen angewendet werden. Das Setzen der SendToSelf -Eigenschaft auf <u>True</u> funktioniert nicht unter allen Windows-Versionen. MSDN legt fest, daß die SendToSelf-Eigenschaft unter Windows 95 und NT 4 ausgeschaltet werden kann. Ein Multicasting-Socket erhält in jedem Fall die Daten, die er versendet.

Methoden

Name	Parameter	Beschreibung
<i>JoinMulticast Group</i>	Group als <u>String</u>	Versucht, mit der angegebenen Multicast-Gruppe eine Verbindung herzustellen. Liefert ein <u>Boolean</u> . Wenn der Socket die Verbindung erfolgreich hergestellt hat, liefert er <u>True</u> . Sie können mit so vielen Multicast-Gruppen verbinden, wie Sie möchten. Der <i>Group</i> -Parameter legt eine Klasse-D-IP-Adresse im Bereich von 224.0.0.0 bis 239.255.255.255 fest.
<i>LeaveMulticastGroup</i>	Group als <u>String</u>	Verläßt die angegebene Multicast-Gruppe.
<i>Read</i>	[Encoding als <u>TextEncoding</u>]	Liefert ein Datagramm aus dem internen Empfangspuffer. Die Adresse-Eigenschaft des Datagrams ist die Remote-Adresse, von der die Daten gesendet wurden. Der optionale <i>Encoding</i> -Parameter ermöglicht es, die Kodierung der zurückgegebenen Daten festzulegen. Sie können das Encoding auch über die Encoding-Eigenschaft setzen.
<i>Write</i>	Data als <u>DataGram</u>	Schreibt die Daten an die angegebene Adresse.
<i>Write</i>	Address als <u>String</u>	Konstruiert ein Datagramm und sendet die Daten an die angegebene Adresse.
	Data als <u>String</u>	Wenn die Adresse Teil einer Multicast-Gruppe ist, erhalten alle Mitglieder der Gruppe diese Daten. Falls die Adresse eine Broadcast-Adresse ist, erhält jeder im Netzwerk diese Daten. Wenn die Adresse eine IP-Adresse ist, werden die Daten nur an diese Adresse gesendet.

Hinweise

Der **UDPSocket** hat kein eigenes Icon in der Steuerelemente-Palette. Der einfachste Weg, es einem Fenster hinzuzufügen, ist, das Fenster-Kontextmenü durch ctrl-Klick (Macintosh) oder über Rechtsklick (Windows) aufzurufen und **Hinzufügen/SocketCore/UDPSocket** auszuwählen. Sie können auch ein TCPSocket-Steuerelement auf ein Fenster draggen und dann im Eigenschaften-Fenster seine SuperClass auf UDPSocket ändern. Sie müssen jedoch kein UDPSocket-Steuerelement einem Fenster hinzuzufügen, um es zu benutzen. Da es keine Unterklasse von Control ist, können Sie es über Code instanzieren.

Das User Datagram Protocol, oder UDP, ist meistens die Basis für den schnellsten und sehr verbreiteten Netzwerkverkehr. Es ist ein verbindungsloses Protokoll das sehr wenig Overhead hat, ist aber nicht so sicher wie TCP. Da es keine Verbindung gibt, benötigen Sie nicht so viele Schritte zur Vorbereitung, wenn Sie ein UDPSocket verwenden wollen.

Um ein **UDPSocket** zu verwenden, muß es an einem bestimmten Port auf Ihren Rechner gebunden werden. Dies wird über die Connect-Methode der **SocketCore**-Klasse vorgenommen. Nachdem diese Bindung hergestellt wurde, ist der **UDPSocket** zum Einsatz bereit. Er beginnt sofort, beliebige Daten zu akzeptieren, die er auf dem an ihn gebundenen Port sieht. Er erlaubt Ihnen auch, Daten zu versenden oder UDP-Socket-Optionen zu setzen (welche später beschrieben werden).

Um zu unterscheiden, welcher Rechner was für Daten sendet, verwendet ein **UDPSocket** eine als Datagram bezeichnete Datenstruktur. Ein Datagram besteht aus zwei Teilen, der IP-Adresse des entfernten Rechners, der Ihnen die Daten sendet und der Nutzlast – oder den Istdaten selbst. Wenn Sie versuchen, Daten zu versenden, müssen Sie die Informationen in Form eines Datagrams festlegen. Diese Information besteht normalerweise aus der Remote-Adresse des Rechners, der Ihre Daten empfangen soll, dem Port, auf den sie gesendet werden und den Daten, die Sie an den entfernten Rechner senden wollen.

UDP-Sockets können in verschiedenen Modi arbeiten, welche sich alle sehr ähnlich sind, aber sehr unterschiedlich verwendet werden können. Der Modus, der einer TCP-Kommunikation am meisten ähnelt, wird „Unicasting“ genannt. Dies ist der Fall, wenn die IP-Adresse, die Sie angeben, wenn Sie Daten schreiben, die eines einzelnen Rechners ist. Ein Beispiel wäre, Daten an www.realsoftware.com oder eine andere Netzwerk-Adresse zu senden. Es ist ein Datagram, welches einen beabsichtigten Empfänger hat.

Der zweite Operations-Modus wird „Broadcasting“ genannt. Wie der Name schon andeutet, wird an alle gesendet. Das ist ähnlich wie das Rufen in ein Megaphon. Jeder bekommt die Mitteilung, ob er will oder nicht. Wenn der Rechner an dem Port lauscht, den Sie festgelegt haben, erhält er auf diesem Port die Daten. Diese Art zu senden ist sehr netzwerkintensiv. Wie Sie sich vorstellen können, kann Broadcasting sich zu einer Riesenmenge Netzwerk-Traffic anhäufen. Der Vorteil ist, daß Daten, die über Broadcast verschickt werden, Ihr Sub-Netz nicht verlassen. Grundsätzlich verläßt ein Broadcast-Versenden Ihr Netzwerk nie in die weite Welt. Wenn Sie Daten als Broadcast versenden möchten, anstatt die Daten an eine IP-Adresse eines entfernten Rechners zu schicken, legen Sie als Broadcast-Adresse Ihren Rechner fest. Diese Adresse variiert von Rechner zu Rechner, daher bietet REALbasic eine Eigenschaft der **UDP-Socket**-Klasse an, welche Ihnen die korrekte Broadcast-Adresse mitteilt.

Der dritte Operations-Modus für UDP-Sockets ist „Multicasting“. Dies ist eine Kombination aus Unicasting und Broadcasting, die sehr leistungsfähig und praktisch ist. Multicasting ist ein wenig wie ein Chatroom: Sie betreten den Chatroom und sind in der Lage, mit jedem anderen im Chatroom Konversation zu betreiben. Wenn Sie den Chatroom betreten möchten, rufen Sie `JoinMulticastGroup` auf und Sie müssen nur die Gruppe festlegen, der Sie beitreten möchten. Der Gruppen-Parameter ist eine spezielle Art von IP-Adresse, genannt *Klasse-D-IP*. Diese reicht von 224.0.0.0 bis 239.255.255.255. Stellen Sie sich die IP-Adresse als den Namen des Chatrooms vor. Wenn Sie mit zwei anderen Leuten zu chatten beginnen wollen, müssen alle drei `JoinMulticastGroup` mit der gleichen Klasse-D-IP-Adresse wie als Gruppe festgelegt aufrufen. Wenn Sie den Chatroom verlassen möchten, müssen Sie nur `LeaveMulticastGroup` aufrufen und auch hier wieder die Gruppe festlegen, die Sie verlassen möchten. Sie können so vielen Multicast-Gruppen beitreten, wie Sie möchten, Sie sind nicht auf eine einzelne gleichzeitig beschränkt. Wenn Sie Daten an die Multicast-Gruppe senden möchten, müssen Sie nur die IP-Adresse der Multicast-Gruppe angeben. Alle Leute, die der gleichen Gruppe beigetreten sind, werden die Nachricht erhalten.

Multicasting besitzt ein paar extra Features, die es sogar zu einem sehr leistungsfähigen Utility für Netzwerk-Applikationen macht. Wenn Sie die Daten, die Sie mit einem Multicast-Send versenden, empfangen möchten, können Sie die `SendToSelf`-Eigenschaft des **UDPSocket** setzen (auch bekannt als Loopback). Wenn sie auf `True` gesetzt ist, erhalten Sie, wenn Sie (an eine Multicast-Gruppe) Daten senden, diese Daten zurück. Sie können außerdem die Anzahl an Router-Sprüngen, die ein Multicast-Datagramm benötigen wird (auch als Time to Live bekannt) festlegen. Wenn Ihr Datagram versendet wird, durchläuft es eine Reihe von Routern auf dem Weg zu seinem Bestimmungsort. Jedesmal, wenn das Datagram einen Router passiert, wird seine `RouterHops`-Eigenschaft erniedrigt. Wenn diese Zahl Null erreicht, wird das Datagram zerstört. Das bedeutet, Sie können sehr viel genauer steuern, wer Ihre Datagrame bekommt. Hier folgen einige Vorschläge, auf welche Werte `RouterHops` gesetzt werden könnte:

Wert	Beschreibung
0	Gleicher Host
1	Gleiches Sub-Netz
32	Gleiche Site
64	Gleiche Region
128	Gleicher Kontinent

Wert	Beschreibung
255	Unbeschränkt

Bedenken Sie, daß, wenn Ihr Datagram einen Router passieren muß, der Multicasting nicht unterstützt, es sofort zerstört wird.

Aufgrund der verbindungslosen Arbeitsweise von UDP gibt es keinerlei Garantie, daß Ihre Daten den Bestimmungsort erreichen werden. Sie können das Problem lösen, indem Sie Ihr eigenes Protokoll auf Basis des UDP-Protokoll erzeugen, welches den Empfang bestätigt.

Der UDPSocket arbeitet im asynchronen Modus, die Connect-Methode jedoch synchron. Wenn die Verbindung fehlschlägt, erhalten Sie sofort ein Error-Event und die IsConnected-Eigenschaft wird sofort gesetzt.

Siehe auch

[SocketCore](#), [Datagram](#), [EasyUDPSocket](#) Klassen.

Unbekanntes Pragma Fehler

Sie haben eine unbekannte Bezeichnung für ein [#Pragma](#) Schlüsselwort angegeben. Sie können nur Pragmas verwenden, die Teil der REALbasic-Sprache sind.

Beispiel

Der folgende String ist ungültig.

```
#pragma myGreatPragma
```

Siehe auch

[#Pragma](#) Schlüsselwort

UnsupportedFormatException Fehler

Tritt auf, wenn Sie einen [String](#)-Ausdruck verwenden, der nicht ausgewertet werden kann.

Übergeordnete Klasse

[RuntimeException](#)

Hinweise

REALbasic erlaubt das Verwenden eines String-Ausdrucks, um die Spaltenbreiten über die Column-Klasse oder die ColumnWidths-Eigenschaft der [ListBox](#) einzustellen. Sie können das Prozentzeichen oder das "*" -Symbol wie in den entsprechenden Abschnitten erklärt verwenden. Wenn Sie einen nicht erlaubten Buchstaben verwenden, tritt eine **UnsupportedFormatException** auf.

Beispiel

Folgende Festlegung führt zu einem **UnsupportedFormatException** Laufzeitfehler:

```
ListBox1.ColumnWidths="50d,50i"
```

Siehe auch

[RuntimeException](#) Klasse; [Exception Block](#).

Uppercase Funktion

Wandelt alle Buchstaben in einem [String](#) in großgeschriebene Buchstaben.

Syntax

Ergebnis=**Uppercase**(Wert)

ODER

Ergebnis=**StringVariable.Uppercase**

Teil	Art	Beschreibung
<i>Ergebnis</i>	String	Eine Kopie des Originalstrings mit allen passend konvertierten Zeichen.

Teil	Art	Beschreibung
<i>Wert</i>	<u>String</u>	Der Originalstring.
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ <u>String</u> .

Hinweis

Liefert den Wert aller Buchstaben in großgeschriebener Schreibweise.

Beispiele

Dieses Beispiel konvertiert die String in Großbuchstaben.

```
Dim s As String
s=Uppercase("dEr Schnelle fUCHS") //liefert "DER SCHNELLE FUCHS"
s=Uppercase("die 5 lahmen hunde") //liefert "DIE 5 LAHMEN HUNDE"
```

Siehe auch

Titlecase, Lowercase Funktionen.

UserCancelled Funktion

Wird verwendet um festzustellen, ob der Anwender die Tastenkombination Befehlstaste-Punkt (auf dem Macintosh) oder Escape (unter Windows) gedrückt hat, um die Ausführung des Codes abzubrechen.

Syntax

Ergebnis=**UserCancelled**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Boolean</u>	<i>Ergebnis</i> ist <u>True</u> , wenn der Anwender Befehlstaste-Punkt (Macintosh) oder Escape (Windows) gedrückt hat und <u>False</u> , wenn nicht.

Hinweis

Die Funktion **UserCancelled** wird solange True liefern, bis der Event-Handler beendet wurde, der beim Drücken der Tastenkombination Befehlstaste-Punkt/Escape ausgeführt wurde.

Beispiele

Dieses Beispiel verwendet die Funktion **UserCancelled** um eine For-Schleife zu beenden, wenn der Anwender die Tastenkombination Befehlstaste-Punkt/Escape drückt.

```
Dim i As Integer
For i = 0 to 10000
    ProgressBar1.Wert = i
    If UserCancelled Then
        Exit
    End If
Next
```

Val Funktion

Liefert die numerische Form eines Strings.

Syntax

Ergebnis=**Val**(string)

ODER

Ergebnis=**StringVariable.Val**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Double</u>	Die numerische Entsprechung des übergebenen Strings.
<i>string</i>	<u>String</u>	Ein beliebiger, gültiger String-Ausdruck.
<i>StringVariable</i>	<u>String</u>	Beliebige Variable vom Typ <u>String</u> .

Hinweis

Die Funktion **Val** hört auf, den String zu untersuchen, sobald es ein Zeichen nicht mehr als Teil einer Zahl erkennt. Alle anderen Zeichen werden automatisch entfernt.

Es erkennt die Präfixe &0 (oktal), &b (Binär) und &H (hexadezimal) . Leerzeichen vor dem Kaufmanns-Und sind jedoch nicht erlaubt. Das bedeutet, " &HFF" liefert keinen Wert, "&H FF" jedoch liefert 255.

Die CDbI-Funktion ist das gleiche wie die **Val**-Funktion, wird aber verwendet, wenn Sie einen String übergeben müssen, der ein anderes Zeichen als Dezimal-Trenner als den Punkt (.) verwendet. Es wird das Zeichen verwendet, das im Zahlenformat-Kontrollfeld festgelegt wurde. **Val** sollte generell verwendet werden, um internationale Daten zu konvertieren, nicht jedoch für Daten, die der Anwender eingibt. **Val** ist nicht für den internationalen Gebrauch geeignet, CDbI jedoch ist es.

Die CStr-Funktion ist das gleiche wie die Str-Funktion, wird jedoch verwendet, wenn Sie eine Zahl übergeben müssen, die ein anderes Zeichen als Dezimal-Trenner als den Punkt (.) verwendet. Es wird das Zeichen verwendet, das im Zahlenformat-Kontrollfeld festgelegt wurde.

Val liefert Null, wenn *string* keine Zahlen enthält.

Beispiele

Diese Beispiele verwenden die Funktion **Val**, um die in einem String enthaltenen Zahlen zurückzuliefern.

```
Dim n As Integer
n = Val("12345") //liefert 12345
n = Val(" 12345") //liefert 12345
n = Val("123 45") //liefert 123
n = Val("&HFFF") //liefert 4095
```

Siehe auch

Str, CDbI und CStr Funktionen; & Operator.

Variant Datentyp

Ein Variant ist ein spezieller Datentyp, der eine beliebige Art von Daten enthalten kann. Verwenden Sie die Funktion VarType (s.u.) oder die Type-Methode der Variant-Klasse, um den Datentyp eines Variant zu bestimmen.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
<i>BooleanValue</i>	<u>Boolean</u>	Liefert den Wert des Variant als ein Boolean.
<i>ColorValue</i>	<u>Color</u>	Liefert den Wert des Variant als ein Color.
<i>DateValue</i>	<u>Date</u>	Liefert den Wert des Variants als Datum. Wenn Sie den String-Wert eines Datums zurückerhalten, wird der String im SQL-Datumsformat YYYY-MM-DD HH:MM geliefert.
<i>DoubleValue</i>	<u>Double</u>	Liefert den Wert des Variant als ein Double.
<i>IntegerValue</i>	<u>Integer</u>	Liefert den Wert des Variant als ein Integer.
<i>ObjectValue</i>	<u>Object</u>	Liefert den Wert des Variant als ein Object.
<i>StringValue</i>	<u>String</u>	Liefert den Wert des Variant als ein String. Enthält der Variant ein Datum, wird dieses in das SQLDateTime-Format konvertiert. Lesen Sie hierzu die Beschreibung der Date-Klasse.

Methoden

Name	Parameter	Beschreibung
<i>Equals</i>	v als Variant	Liefert ein <u>Boolean</u> . Gibt True zurück, wenn der aktuelle Variant vom gleichen Datentyp und Wert ist wie der als Parameter übergebene Variant.

Name	Parameter	Beschreibung
<i>Hash</i>	<u>Integer</u>	Liefert einen Hash-Wert für diesen Variant; das ist einer, der immer das gleiche für jeden Variant mit dem gleichen Wert ist. Der Hash-Wert ist garantiert eindeutig für Integer und Colors. Bei Objekten ist er garantiert eindeutig, so lange alle Objekte momentan existieren (wenn jedoch ein Objekt einmal gelöscht wird, könnte ein künftiges Objekt den gleichen Hash-Wert bekommen).
<i>IsNull</i>		Liefert ein <u>Boolean</u> . Liefert True, wenn der Variant <u>Null</u> ist.
<i>IsNumeric</i>		Liefert ein <u>Boolean</u> . Liefert True, wenn der Variant ein numerischer Datentyp ist. Enthält der Variant einen String einer Zahl, zum Beispiel "8", statt 8, so liefert IsNumeric ebenfalls True.
<i>Type</i>		Liefert ein <u>Integer</u> , der den Datentyp von <i>v</i> angibt. Wenn der Variant nicht <u>Nil</u> ist, bietet dies die gleiche Funktionalität wie die globale Funktion VarType. Siehe Tabelle der Datentypen für die VarType-Funktion.

Hinweise

Wenn Sie einem String ein Variant zuweisen, numerisch oder variabel, konvertiert REALbasic den Wert des Variant in den Datentyp der Variablen. Zum Beispiel:

```
Dim v as Variant
Dim n as Integer
Dim s as String
v=25
s=v // s ist "25"
v="25"
n=v // n ist 25
```

Wenn es irgendwelche Zweideutigkeiten bei der Typkonvertierung von Variants gibt, sollten Sie eine der Eigenschaften der **Variant**-Klasse verwenden, um den REALbasic-Compiler dazu zu zwingen, den Variant in den gewünschten Typ zu konvertieren. In diesem Beispiel soll die Produktnummer ein String sein, zusammengesetzt aus der Modellnummer und der Teilenummer. Um dies zu machen, verwenden Sie die StringValue-Eigenschaft, um REALbasic mitzuteilen, was Sie möchten:

```
Dim Modell,Teilenummer,Produkt as Variant
Modell=100
Teilenummer=546
Produkt=Modell+Teilenummer //Produkt=646
Produkt=Modell.StringValue+Teilenummer.StringValue //Produkt=100546
```

Sie können die Variablen aber auch einfach als String festlegen und das Zusammenbauen des Strings direkt machen.

Die Type-Methode liefert den Typ des Variant, vorausgesetzt, er ist nicht Nil. Folgender Code löst z.B. einen NilObjectException Fehler aus:

```
Dim v as Variant
Dim i as Integer
i=v.type
```

Andererseits liefert folgendes Beispiel 0.

```
Dim v as Variant
Dim i as Integer
i=VarType(v)
```

Variants, die in einem Ausdruck verwendet wurden, konvertieren sich selbst in den Typ des anderen Operanden, egal was für Daten dieser enthält und schieben die Typ-Konvertierung nicht bis zur Laufzeit vor sich her. Dies betrifft nur Ausdrücke mit einem Variant-Operanden und einem Operanden eines anderen Typs, nicht Ausdrücke, in denen beide Operanden Variants sind.

Siehe auch

VarType, IsNumeric Funktion; Dictionary, DatabaseField Klassen; Integer, Single, Double, String, Boolean, Color Datentyp; Dim Befehl.

VarType Funktion

Wird verwendet, um den Datentyp einer Variablen zu bestimmen.

Syntax

result=**VarType**(value)

Teil	Typ	Beschreibung
<i>result</i>	<u>Integer</u>	Liefert den Datentyp von <i>value</i> .
<i>value</i>	<u>Variant</u>	Wert, der typisiert werden soll.

Hinweise

Result nimmt folgende Werte an:

Result	Beschreibung
0	<u>Nil</u>
2	<u>Integer</u>
5	<u>Double</u> oder <u>Single</u>
7	<u>Date</u>
8	<u>String</u>
9	<u>Object</u>
11	<u>Boolean</u>
16	<u>Color</u>

Hinweis: Bitte beachten Sie, daß manche frühere REALbasic-Versionen für **Object** den Wert **13** zurücklieferten.

Beispiel

Folgende Code-Zeile liefert 8 zurück (Datentyp String):

```
EditField1.text=str(VarType( "Hermann" ))
```

Siehe auch

Variant Klasse, Integer, Single, Double, String, Boolean, Color Datentyp; IsNumeric Funktion.

Vector3D Klasse

Wird verwendet, um absolute oder relative Positionen eines Vektors in einem dreidimensionalen RB3DSpace zu repräsentieren.

Eigenschaften

Name	Art	Beschreibung
<i>X</i>	<u>Double</u>	X-Koordinate eines Punktes im 3D-Raum.
<i>Y</i>	<u>Double</u>	Y-Koordinate eines Punktes im 3D-Raum.
<i>Z</i>	<u>Double</u>	Z-Koordinate eines Punktes im 3D-Raum.

Methoden

Name	Parameter	Beschreibung
<i>Add</i>	vec als Vector3D	Fügt den Vektor <i>vec</i> dem aktuellen Vektor hinzu und speichert das Ergebnis in diesem Objekt.
<i>Copy</i>	vec als Vector3D	Sorgt dafür, daß dieser Vector3D dem übergebenen entspricht.
<i>Cross</i>	vec als Vector3D	Liefert das Kreuzprodukt zweier Vektoren. <i>v1.Cross(v2)</i> liefert Ihnen einen neuen Vektor, der senkrecht zur Ebene ist, die durch <i>v1</i> und <i>v2</i> definiert ist. Liefert ein Vector3D.
<i>Dot</i>	vec als Vector3D	Liefert das Skalarprodukt (als ein <u>Double</u>) zweier Vektoren. <i>v1.Dot(v2)</i> liefert Ihnen den Cosinus des Winkels zwischen <i>v1</i> und <i>v2</i> . Liefert ein Double.
<i>Length</i>		Liefert die Länge des Vektors im 3D-Raum. Liefert ein Double.
<i>LenSquared</i>		Liefert die quadrierte Länge des Vektors im 3D-Raum. Liefert ein Double.
<i>Minus</i>	vec als Vector3D	Liefert die Differenz dieses Vektors und des Vektors <i>vec</i> .

Name	Parameter	Beschreibung
<i>MultiplyBy</i>	<i>d</i> als <u>Double</u>	Multipliziert diesen Vektor mit dem Skalar <i>d</i> und speichert das Ergebnis in diesem Objekt. Dies skaliert den Vektor; eine andere Möglichkeit, einen Vektor zu skalieren ist eine Zuweisung zur <code>Length</code> - oder <code>LenSquared</code> -Eigenschaft.
<i>Normalize</i>		Skaliert den Vektor so, daß seine quadrierte Länge (und Länge) 1 ist.
<i>Plus</i>	<i>vec</i> als Vector3D	Liefert ein Vector3D . Liefert die Summe dieses Vektors und des Vektors <i>vec</i> .
<i>Subtract</i>	<i>vec</i> als Vector3D	Subtrahiert den übergebenen Vektor von diesem und speichert das Ergebnis in diesem Objekt.
<i>Times</i>	<i>d</i> als <u>Double</u>	Liefert ein Vector3D . Liefert das Produkt dieses Vektors und des Skalars <i>d</i> .

Konstruktor-Syntax

Syntax	Parameter	Beschreibung
<u>New</u> Vector3D (<i>x</i> , <i>y</i> , <i>z</i>)	<i>x</i> als <u>Double</u> <i>y</i> als <u>Double</u> <i>z</i> als <u>Double</u>	Erzeugt ein Vector3D und initialisiert <i>x</i> , <i>y</i> , und <i>z</i> mit den übergebenen Werten.
<u>New</u> Vector3D (<i>vec</i>)	<i>vec</i> als Vector3D	Erzeugt einen Vector3D und initialisiert den neuen Vektor mit den gleichen Werten wie den übergebenen.

Hinweise

Die **Vector3D**-Klasse wird verwendet, um eine absolute oder relative Position zu repräsentieren. Es sind einfach die X-, Y-, und Z-Koordinaten eines Punktes im 3D-Raum. Je nachdem, wie Sie es nutzen möchten, können diese die absolute Position von etwas im 3D-Raum bedeuten oder eine Positionsveränderung. Sie können seine Länge (das ist sein Abstand vom Ursprung) oder seine quadrierte Länge (womit einfacher zu rechnen ist) ermitteln. `Normalize()` skaliert den Vektor so, daß seine Länge 1 wird. Die `Cross`- und `Dot`-Methoden erlauben Ihnen, zwei verschiedene Vektor-Multiplikationen auszuführen.

Vergleich von Vektor-Operationen

Die **Vector3D**-Klasse besitzt einen kompletten Satz von Methoden, um Vektor-Arithmetik durchzuführen, so daß Sie nicht mit den individuellen Elementen der Vektoren arbeiten müssen. Sie können das Ergebnis der Berechnung entweder im aufrufenden Vektor speichern oder in einem neuen Vektor. Folgende Tabelle vergleicht die sechs Methoden:

Operation	Speichert Ergebnis am gleichen Ort	Speichert Ergebnis in neuen Vektor
Addition	<i>a.Add(b)</i>	<i>c = a.Plus(b)</i>
Subtraktion	<i>a.Subtract(b)</i>	<i>c = a.Minus(b)</i>
Skalierung	<i>a.MultiplyBy(f)</i>	<i>c = a.Times(f)</i>

Siehe auch

Object3D, Group3D, Quaternion Klassen; RB3DSpace Steuerelement.

Vererbung von einer Klasse ist nicht möglich Fehler

Tritt nur in RBScript auf. Tritt auf, wenn Sie die übergeordnete Klasse einer Klasse auf ein Interface an Stelle einer echten Klasse setzen.

VirtualVolume Klasse

Erlaubt das Erzeugen und Verwalten einer Hierarchie von „virtuellen“ Dateien innerhalb einer physikalischen Datei. Grundlegendes Schreiben und Lesen von Text- und Binary-Streams werden unterstützt.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
Root	FolderItem	Referenz auf das Wurzelverzeichnis des virtuellen Volumes.

Hinweise

Die [FolderItem](#)-Klasse besitzt zwei Methoden und eine Eigenschaft für die Arbeit mit virtuellen Volumes. Verwenden Sie die Methoden [CreateAsVirtualVolume](#) und [OpenAsVirtualVolume](#), um virtuelle Volumes zu erzeugen und zu öffnen. Die [VirtualVolume](#)-Eigenschaft der [FolderItem](#)-Klasse liefert ein **VirtualVolume**, falls sich das [FolderItem](#) in einem virtuellen Volume befindet.

Nachdem Sie das **VirtualVolume** erzeugt haben, können Sie dessen Wurzelverzeichnis bekommen und darin mit den gleichen [FolderItem](#)-Methoden wie bei echten Dateien navigieren. Virtuelle Dateien sind plattformübergreifend kompatibel und unterstützen das grundlegende Lesen und Schreiben als Text- und Binär-Dateien. Die [SaveAsPicture](#)- und [SaveStyledEditField](#)-Methoden der [FolderItem](#)-Klasse arbeiten nicht auf Virtual Volumes. Virtual Volumes unterstützen keine [ResourceForks](#).

Virtuelle Volumes werden ihre Daten immer auf ein Volume speichern wenn sie geschlossen werden. Dies erhöht deren Datensicherheit im Falle eines Stromausfalls.

Dateinamen können bis zu 223 Byte lang sein. Pfade werden nicht unterstützt, Ordner dagegen schon (erzeugen Sie z.B. ein virtuelles Verzeichnis mit der [CreateAsFolder](#)-Method der [FolderItem](#)-Klasse und navigieren Sie mit den [Child](#)- oder [Item](#)-Methoden dieser Klasse). Das virtuelle Dateisystem unterstützt Vier-Byte-Type-Codes (erreichbar über [f.MacType](#)), erzeugt aber keine [Creator](#)-Codes.

Beispiel

Folgendes Beispiel erzeugt ein **VirtualVolume**, ermittelt dessen [Root](#)-Eigenschaft und schreibt eine Textdatei in das Virtual Volume.

```

Dim vv As VirtualVolume
Dim realFile As FolderItem
Dim virtFile As FolderItem
Dim outp As TextOutputStream

realFile = GetFolderItem("").Child("VV")
vv = realFile.CreateVirtualVolume
If vv = Nil then
    MsgBox "Kann Virtual Volume nicht erzeugen"
else
    virtFile = vv.Root.Child("Virtual File.txt")
    outp = virtFile.CreateTextFile
    outp.Write "Hallo Welt!"
end if

```

Um auf ein vorhandenes Virtual Volume zuzugreifen, verwenden Sie die [OpenAsVirtualVolume](#)-Methode an Stelle der [CreateVirtualVolume](#)-Methode der [FolderItem](#)-Klasse.

Siehe auch

[FolderItem](#) Klasse.

Volume Funktion

Liefert ein [FolderItem](#), das ein gemountetes Volume repräsentiert.

Syntax

Ergebnis=**Volume**(VolumeNummer)

Teil	Art	Beschreibung
<i>Ergebnis</i>	FolderItem	Das gemountete Volume, dessen Nummer übergeben wurde.
<i>VolumeNummer</i>	Integer	Die Nummer des Volumes, für das Sie ein FolderItem benötigen.

Hinweis

Die Funktion **Volume** liefert ein [FolderItem](#), das das Volume repräsentiert, dessen Nummer Sie übergeben haben. Volume 0 (null) ist hierbei das Start-Volume.

Zusammen mit der Funktion [VolumeCount](#) kann man diese Funktion verwenden, um in einer Schleife alle gemounteten Laufwerke zu finden.

Beispiele

Dieses Beispiel trägt die Namen aller gemounteten Volumes in eine ListBox ein:

```
Dim i,n as Integer
n= VolumeCount-1
For i=0 to n
  ListBox1.AddRow Volume(i).Name
Next
```

Folgendes Beispiel liefert ein FolderItem für den Dokumente-Ordner auf dem Startvolume.

```
Dim f as FolderItem
f=Volume(0).Child("Dokumente")
```

Siehe auch

VolumeCount Funktion, FolderItem Klasse.

VolumeCount Funktion

Liefert die Anzahl gemounteter Volumes zurück.

Syntax

Ergebnis=**VolumeCount**

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Integer</u>	Die Anzahl gemounteter Volumes

Hinweis

Die Funktion **VolumeCount** liefert die Anzahl gemounteter Volumes zurück.

Beispiele

Bei der Beschreibung der Funktion Volume finden Sie ein passendes Beispiel.

Siehe auch

Volume Funktion.

While...Wend Befehl

Führt eine Reihe von Befehlen wiederholt aus, solange eine bestimmte Bedingung erfüllt (True) ist.

Syntax

While Bedingung

Anweisungen

Wend

Bedingung ist ein beliebiger, gültiger, boole'scher Ausdruck.

Hinweis

Wenn *Bedingung* True ist, werden alle *Anweisungen* ausgeführt, bis der **Wend** Befehl erreicht wurde. Ist *Bedingung* dann immer noch True, so werden die *Anweisungen* noch einmal ausgeführt. Ist *Bedingung* False, so wird das Programm in der Zeile fortgesetzt, die dem **Wend** Befehl folgt.

While...Wend Befehle können beliebig tief ineinander verschachtelt werden. Zu jedem **Wend** Befehl gehört ein **While** Befehl.

Wenn die Schleife läuft, übernimmt sie die Benutzerschnittstelle, um den Anwender daran zu hindern, mit Menüs oder Steuerelementen zu arbeiten. Normalerweise macht das nichts aus. Wenn jedoch die Schleife sehr lange läuft, sollten Sie deren Code in einen separaten Thread verschieben, damit sie im Hintergrund läuft.

Beispiel

Dieses Beispiel verwendet den Befehl **While...Wend**, um eine Variable zu erhöhen.

```
Dim x As Integer
While x<100
```

$x=x+1$

Wend

Siehe auch

Do...Loop, For...Next Befehle.

Window Funktion

Liefert eine Referenz auf ein offenes Fenster.

Syntax

Ergebnis=**Window**(WindowNumber)

Teil	Art	Beschreibung
<i>Ergebnis</i>	<u>Window</u>	Eine Referenz auf ein Fenster, dessen Nummer übergeben wurde.
<i>WindowNumber</i>	<u>Integer</u>	Die Nummer des Fensters, zu dem Sie eine Referenz haben möchten.

Hinweise

Die **Window**-Funktion liefert eine Referenz auf die übergebene Fenster-Nummer. Die Fenster-Liste enthält alle Fenster, die erzeugt wurden. Fenster 0 ist das oberste Fenster. Schwimmende Fenster befinden sich immer oberhalb von Dokumentfenstern. Um zum Beispiel das oberste Dokumentfenster zu bekommen, wenn es auch ein schwimmendes Fenster gibt, müssen Sie außerdem die Frame-Eigenschaft jedes Fensters überprüfen. Wenn Sie die Reihenfolge der Fenster in der Liste von vorne nach hinten auswerten, sollten Sie daran denken, daß die Visible-Eigenschaft eines Fensters vielleicht nicht auf True gesetzt ist. Sollte das der Fall sein, sollten Sie die Visible-Eigenschaft eines Fensters überprüfen, wenn Sie das oberste sichtbare Fenster suchen. Falls Sie das nicht machen, kann es passieren, daß Sie als oberstes Fenster eines erwischen, das nicht sichtbar ist.

Diese Funktion kann in Verbindung mit der Funktion WindowCount verwendet werden, um nacheinander alle offenen Fenster zu bearbeiten.

Beispiel

Dieses Beispiel listet die Titel aller offenen Fenster in einer ListBox auf:

```
Dim i as Integer
Dim WindowTitle as String
If WindowCount > 1 then
  For i=0 to WindowCount-1
    WindowTitle=Window(i).Title
    ListBox1.AddRow windowTitle
  Next
Return Nil
End If
```

Dieses Beispiel ermittelt das oberste Dokument-Fenster. Es berücksichtigt nur Dokument-Fenster (Frame=0) und beachtet, ob die Möglichkeit besteht, daß ein Fenster unsichtbar ist:

```
Dim i,wc as Integer
' Berechne die Anzahl der Fenster für die Schleife.
wc = WindowCount - 1
' Die Fenster-Liste durchgehen.
For i = 0 to wc
  ' Handelt es sich um ein unsichtbare Dokumentfenster?
  If (Window(i).Frame = 0) and Window(i).visible then
    ' Das ist das erste sichtbare Dokumentfenster.
    return Window(i)
  end if
Next
```

Siehe auch

WindowCount -Funktion, Window Klasse.

Window Klasse

Ein beliebiges Fenster.

Übergeordnete Klasse

Object

Eigenschaften

Name	Art	Beschreibung
<i>BackColor</i>	<u>Color</u>	Die Hintergrundfarbe des Fensters.
<i>Backdrop</i>	<u>Picture</u>	Ein Picture-Objekt, das im Fenster dargestellt wird, sobald sich dieses öffnet.
<i>Balloon Help</i>	<u>String</u>	Der Text für die Sprechblasenhilfe
Closebox	<u>Boolean</u>	Wenn <u>True</u> , so hat das Fenster eine Box zum Schließen des Fensters.
<i>Composite</i>	<u>Boolean</u>	Falls <u>True</u> , erlaubt es, Steuerelemente auf anderen Steuerelementen (darüberliegend) oder als Hintergrundbild ohne das Steuerelement umgebendes Rechteck. Ohne Wirkung bei folgenden Fenstertypen: Modal, Movable Modal, PlainBox und ShadowedBox. Bei Drawer- und Metal-Fenstern ist Composite immer aktiv. Funktioniert nur unter Mac OS X ab 10.2. Weitere Hinweise zur Composite-Eigenschaft finden Sie im Hinweise-Abschnitt.
<i>Control</i>	Array von Controls	Nullbasiertes Array von Steuerelementen eines Fensters.
<i>ControlCount</i>	<u>Integer</u>	Die Anzahl der Steuerelemente im Fenster.
<i>DockItem</i>		(nur Mac OS X). Erlaubt Ihnen die Manipulation des Dock-Eintrags, das zu der Applikation gehört. Die DockItem-Eigenschaft besitzt zwei Methoden, UpdateRow und ResetIcon. ResetIcon setzt das Icon in seinen ursprünglichen Zustand zurück (Standardaussehen). Das Aussehen des Icons können Sie mit den Methoden der <u>Graphics</u> -Klasse ändern. Da ein Mac OS X-Icon automatisch skaliert wird, sollten Sie das Icon als in Pixel-Icon in der Größe 128*128 entwerfen. Um das Icon neu zu zeichnen, rufen Sie die UpdateNow-Methode auf. Sie können auch mit Hilfe der ClearRect-Eigenschaft der <u>Graphics</u> -Klasse mit einem leeren Icon beginnen. Alles, was man mit einem <u>Graphics</u> -Objekt machen kann, kann man mit dem Grafik-Objekt im Dock machen (Zeichnen in einem Bild oder ein Shape 2D verwenden). Die DockItem-Eigenschaft der <u>Application</u> -Klasse erlaubt Ihnen, den Dock-Eintrag jedes einzelnen Fensters zu steuern.
<i>FloaterProcess</i>	<u>String</u>	Der Mac-Creator-Code der Anwendung, in der das Fenster sichtbar sein soll. Ist dieser leer, so ist das Fenster in allen Anwendungen sichtbar. Der Fenstertyp muß „Global schwimmendes Fenster“ sein, damit dies funktioniert.
Frame	<u>Integer</u>	Der Fenstertyp. 0 - Dokument-Fenster 1 - Bewegbares modales Fenster 2 - Modale Dialogbox 3 - Schwimmendes Fenster 4 - Einfache Box 5 - Schattierte Box 6 - Abgerundetes Fenster 7 - Globales schwimmendes Fenster. 8 - Sheet-Fenster (als solches nur unter Mac OS X angezeigt) 9-Metall-Fenster (als solches nur unter Mac OS X ab 10.2 angezeigt) 10-Drawer-Fenster (als solches nur unter Mac OS X ab 10.2 angezeigt)
<i>FullScreen</i>	<u>Boolean</u>	Wenn <u>True</u> , vergrößert sich das Fenster auf den gesamten Bildschirm. Die Werte von Left, Top, Width und Height werden an die neue Fensterposition angepaßt. Sie können nicht geändert werden, so lange FullScreen <u>True</u> ist. Wenn MenuBarVisible <u>True</u> ist, wird unter Mac OS der obere Rand des Fensters teilweise von der Menüleiste verdeckt. Wenn Sie mehrere Monitor angeschlossen haben, ändert sich das Fenster in der Größe so, daß es den Monitor ausfüllt, der den größten Teil des Fensters darstellt. FullScreen ist auf <u>False</u> voreingestellt.

Name	Art	Beschreibung
GrowIcon	<u>Boolean</u>	Wenn <u>True</u> , ist das Fenster in der Größe veränderbar.
<i>HasBackColor</i>	<u>Boolean</u>	Wenn <u>True</u> , so ist die Hintergrundfarbe des Fensters auf den Wert der Eigenschaft Backcolor eingestellt.
<i>Height</i>	<u>Integer</u>	Die Höhe des Fensters.
<i>Left</i>	<u>Integer</u>	Der Abstand (in Pixeln) zwischen linker Bildschirmkante und der linken Seite des Fensters.
<i>LiveResize</i>	<u>Boolean</u>	Falls <u>True</u> , wird das Fenster „live“ neu gezeichnet, wenn es durch Draggen in der Größe geändert wird. LiveResize sollte auf <u>True</u> gesetzt werden, wenn Sie Drawer-Fenster verwenden. Funktioniert nur unter Mac OS X.
MacProcID	<u>Integer</u>	Apple Window Definition ID. Kann verwendet werden, um Fenstertypen zu definieren, die nicht von der Frame-Eigenschaft unterstützt werden. Fenstertypen, die durch MacProcID definiert werden zeigen nur auf der Pacintosh-Plattform Wirkung. Unter Windows wird der Fenstertyp von der Frame-Eigenschaft gesteuert. Siehe Abschnitt <u>Hinweise</u> .
<i>MacWindowPtr</i>	<u>Integer</u>	Liefert einen Zeiger auf das Fenster. Kann von Toolbox-Programmierern verwendet werden, um das Fenster zu steuern.
<i>MaxHeight</i>	<u>Integer</u>	Die Maximalhöhe, auf die das Fenster vergrößert werden kann.
<i>MaxWidth</i>	<u>Integer</u>	Die Maximalbreite, auf die das Fenster vergrößert werden kann.
<i>MenuBar</i>	MenuItem	Die Menüleiste, die mit Fenster zugewiesen ist. Wenn das Fenster aktiv ist, wird die Menüleiste angezeigt. Wenn dem Fenster keine Menüleiste zugewiesen ist wird die globale Menüleiste der Applikation verwendet. Das ist die Menüleiste, die dem Projekt als Voreinstellung hinzugefügt wurde oder durch die MenuBar-Eigenschaft der <u>Application</u> -Klasse zugewiesen wurde. Unter Windows hat jedes Fenster unter der Titelleiste eine Menüleiste. Es verwendet die Menüleiste, die dem Fenster zugewiesen wurde oder die Menüleiste der Applikation, wenn dem Fenster keine Menüleiste zugewiesen wurde. Wenn es sich um eine MDI-Applikation handelt, verwendet das MDI-Fenster die Menüleiste der Applikation. Individuelle Dokument-Fenster können keine Menüleiste haben, das ist aber eine Einschränkung von Windows. Schwimmende Fenster können eine Menüleiste haben und diese Fenster zeigen die Menüleiste an, die Sie ihnen zuweisen.
<i>MenuBarVisible</i>	<u>Boolean</u>	Wenn auf <u>False</u> gesetzt, wird die Menüleiste immer dann ausgeblendet, wenn das Fenster das oberste nicht-schwimmende ist. Setzen Sie MenuBarVisible auf <u>False</u> und FullScreen auf <u>True</u> , damit das Fenster den gesamten Bildschirm einnehmen kann. Die Menüleiste wird wieder dargestellt, wenn ein anderes Fenster (dessen MenuBarVisible-Eigenschaft <u>True</u> ist) nach oben geholt wird oder wenn die Applikation in den Hintergrund geschickt wird.
<i>MinHeight</i>	<u>Integer</u>	Die Minimalhöhe, auf die das Fenster verkleinert werden kann.
<i>MinWidth</i>	<u>Integer</u>	Die Minimalbreite, auf die das Fenster verkleinert werden kann.
<i>MouseCursor</i>	<u>MouseCursor</u>	Der Mauszeiger, der dargestellt wird, solange sich der Mauszeiger im Fenster befindet und die MouseCursor-Eigenschaft der Application-Klasse Nil ist. Wenn die <u>Application</u> -Klasse MouseCursor nicht Nil ist, werden die Nicht-Nil-MouseCursor-Eigenschaften, die zu irgendeinem <u>Window</u> oder <u>Control</u> gehören, ignoriert. Wenn die MouseCursor-Eigenschaft des Fensters nicht Nil ist, werden die MouseCursor-Eigenschaften aller Steuerelemente ignoriert. Mögliche Namen sind: WatchCursor, IBeamCursor und ArrowCursor. MouseCursors können ebenso aus Resource-Dateien zugewiesen werden, die dem Projekt hinzugefügt wurden.
<i>MouseX</i>	<u>Integer</u>	Die x-Koordinate der Maus in Pixeln. Gemessen von der linken oberen Ecke des Fensters.

Name	Art	Beschreibung
<i>MouseY</i>	<u>Integer</u>	Die y-Koordinate der Maus in Pixeln. Gemessen von der linken oberen Ecke des Fensters.
<i>Placement</i>	<u>Integer</u>	Die Position, an der das Fenster geöffnet wird. 0 – Standard 1 – Parent Window 2 – Hauptbildschirm 3 – Parent Window Screen 4 – Stagger Bei Drawer-Fenster haben die Werte folgende Bedeutung: 0 - Oben/Links 1 - Zentriert 2 - Unten/Rechts
<i>Title</i>	<u>String</u>	Der Fenstertitel.
<i>Top</i>	<u>Integer</u>	Der Abstand (in Pixeln) zwischen der oberen Kante des Bildschirms und der oberen Kante des Fensters.
<i>Visible</i>	<u>Boolean</u>	Wenn <u>True</u> , ist das Fenster nach dem Öffnen sichtbar.
<i>Width</i>	<u>Integer</u>	Die Breite des Fensters.
<i>WinHWND</i>	<u>Integer</u>	Liefert ein Handle auf das Fenster (nur unter Win 32).
<i>ZoomIcon</i>	<u>Boolean</u>	Wenn <u>True</u> , so hat das Fenster eine Zoom-Box.

Methoden

Name	Parameter	Beschreibung
<i>AcceptFileDrop</i>	FileType als <u>String</u>	Erlaubt das Darauf-Ziehen (Droppen) von Dokumenten mit dem Dateityp <i>FileType</i> . <i>FileType</i> muß ein Dateityp sein, den Sie im Dateitypen-Dialog festgelegt haben. Dies funktioniert sogar korrekt, wenn Sie mehrere Dateitypen mit dem selben Dateitypen-Code definiert haben und Sie nicht speziell AcceptFileDrop für den ersten solchen Typ aufrufen.
<i>AcceptMacData</i>	Type als <u>String</u>	Erlaubt das Ziehen von Daten des angegebenen Typs auf dieses Fenster. <i>Type</i> ist ein vierbuchstabiger Resource-Code, z.B. „snd“ oder „TEXT“.
<i>AcceptPictureDrop</i>		Erlaubt das Ziehen von Bildern auf dieses Fenster.
<i>AcceptTextDrop</i>		Erlaubt das Ziehen von Text auf dieses Fenster.
<i>Close</i>		Schließt das Fenster. Nachdem es geschlossen wurde, kann es nicht mehr refreshed oder neu gezeichnet werden.
<i>DrawInto</i>	g als <u>Graphics</u> x als <u>Integer</u> y als <u>Integer</u>	Zeichnet den Inhalt des Fensters in das angegebene <u>Graphics</u> -Objekt.
<i>Hide</i>		Blendet das Fenster aus.
<i>Refresh</i>	LöscheHintergrund als <u>Boolean</u>	Zeichnet den gesamten Fensterinhalt neu. Der optionale Parameter LöscheHintergrund gibt an, ob vor dem Zeichnen des Fensters der Hintergrund gelöscht werden soll. Voreingestellt ist True.
<i>RefreshRect</i>	X als <u>Integer</u> Y als <u>Integer</u> Width als <u>Integer</u> Height als <u>Integer</u> [LöscheHintergrund als <u>Boolean</u>]	Zeichnet nur die angegebenen Bereiche des Fensters neu. Der optionale Parameter LöscheHintergrund gibt an, ob vor dem Zeichnen des Fensters der Hintergrund gelöscht werden soll. Voreingestellt ist True.

Name	Parameter	Beschreibung
<i>SetFocus</i>		Entfernt den Fokus von dem Steuerelement, das ihn momentan besitzt, so daß kein Steuerelement den Fokus hat. Verwenden Sie die SetFocus-Methode der RectControl-Klasse, um den Fokus auf ein bestimmtes Steuerelement innerhalb des Fensters zu setzen.
<i>Show</i>		Zwingt das Fenster sofort sichtbar zu werden statt darauf zu warten, daß die Anwendung im Ruhezustand arbeitet.
<i>ShowModal</i>		Zwingt die aktuelle Methode dazu, ihre Arbeit zu unterbrechen, bis der Dialog ausgeblendet oder geschlossen wird.
<i>ShowModalWithin</i>	ElternFenster als Window	Stellt das Fenster innerhalb des <i>ElternFenster</i> dar. Wird zum Anzeigen von Sheet-Fenstern und Drawern innerhalb eines bestimmten Eltern-Fenster verwendet.
<i>ShowWithin</i>	ElternFenster als Window [facing als <u>Integer</u>]	Stellt das Fenster innerhalb des <i>ElternFenster</i> dar. Wird zum Anzeigen von Sheet-Fenstern innerhalb eines bestimmten Eltern-Fenster verwendet. Bei einem Drawer-Fenster legen die optionalen facing-Parameter fest, auf welcher Seite der Drawer herausfährt: -1 = default (vom System festgelegt - normalerweise links), 0 = oben, 1 = unten, 2 = rechts, 3 = links (gleiche Reihenfolge wie TabPanels). Wenn Sie facing weglassen, wird der Defaultwert verwendet.
<i>UpdateNow</i>		Leert den Hintergrund-Puffer bei Mac OS X Fenstern. Da Mac OS X eine Doppel-Pufferung vornimmt, werden die Zeichenoperationen im Hintergrund gesammelt, bevor sie auf dem Bildschirm erscheinen. Unter Umständen möchten Sie den Puffer manuell leeren, nachdem Sie einen Fortschrittsbalken aktualisiert haben, das Bild einer Animation oder eines Canvas gezeichnet haben oder anderes ähnlicher Natur. UpdateNow hat keine Auswirkung unter Windows und dem Classic Mac OS.

Events

Name	Parameter	Beschreibung
<i>Activate</i>		Das Fenster wurde aktiviert.
<i>CancelClose</i>	appBeenden als <u>Boolean</u>	Das Fenster soll durch die Methode <u>Quit</u> oder durch Klick auf die Closebox geschlossen werden. Liefern Sie <u>True</u> zurück, um das Schließen des Fensters (im Falle von <u>Quit</u> auch das Schließen anderer offener Fenster) zu verhindern. Der optionale Parameter appBeenden ist True, wenn die Applikation geschlossen werden soll und False, wenn nur ein Fenster der Applikation geschlossen werden soll.
<i>Close</i>		Das Fenster wird geschlossen.
<i>Deactivate</i>		Das Fenster wurde ausgeblendet.
<i>DropObject</i>	Obj als <u>DragItem</u>	Das durch Obj vertretene Objekt wurde auf das Fenster gezogen.
<i>EnableMenuItems</i>		Der Anwender hat einen Menüeintrag angeklickt oder ein Tastaturkürzel eines Menüeintrags betätigt.
<i>KeyDown</i>	Key als <u>String</u>	Die Taste Key wurde gedrückt und wurde von keinem Objekt des Fensters bearbeitet.
<i>MouseDown</i> (Funktion)	X als <u>Integer</u> , Y als <u>Integer</u>	Die Maustaste wurde im Fenster an den angegebenen Koordinaten gedrückt. Geben Sie <u>True</u> zurück, wenn Sie MouseDown auswerten wollen.
<i>MouseDown</i> (Funktion)	X als <u>Integer</u> , Y als <u>Integer</u>	Die Maustaste wurde im Fenster gedrückt und der Mauszeiger mit gedrückt gehaltener Maustaste an die Koordinaten x und y (relativ zum Fenster) verschoben. Dieses Event wird nicht auftreten, wenn Sie nicht im Event MouseDown <u>True</u> zurückliefern.
<i>MouseEnter</i>		Der Mauszeiger hat das Fenster betreten.

Name	Parameter	Beschreibung
<i>MouseExit</i>		Der Mauszeiger hat das Fenster verlassen
<i>MouseMove</i>	X als <u>Integer</u> Y als <u>Integer</u>	Der Mauszeiger hat sich innerhalb des Fensters an die lokalen Koordinaten x und y bewegt.
<i>MouseUp</i>	X als <u>Integer</u> Y als <u>Integer</u>	Die Maustaste wurde im Fenster an den lokalen Koordinaten x und y losgelassen.
<i>Moved</i>		Das Fenster wurde verschoben.
<i>Open</i>		Das Fenster wird sich öffnen.
<i>Paint</i>	G als <u>Graphics</u>	Das Fenster muß neu gezeichnet werden. Die angegebenen Parameter geben Ihnen hierzu Zugriff auf das Objekt.
<i>Resized</i>		Das Fenster wurde in der Größe geändert.
<i>Resizing</i>		Der Anwender ist dabei, die Größe des Fensters zu ändern. Resizing wird ausgelöst, egal ob die LiveResize-Eigenschaft in Carbon True ist. (HINWEIS: Die Eigenschaften Top, Left, Width und Height wurden zu diesem Zeitpunkt schon angepaßt, wenn LiveResize eingeschaltet ist, die Steuerelemente wurden noch nicht gezeichnet. Sie könnten z.B. eine Boolean-Eigenschaft verwenden, um festzulegen, ob ein Canvas in seinem Paint-Event neu gezeichnet werden soll oder nicht.)

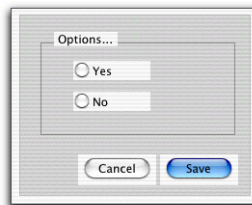
Hinweise

Window-Konstruktoren werden aufgerufen, nachdem das Fenster und seine Steuerelemente erzeugt wurde, jedoch bevor das Open-Event aufgerufen wird.

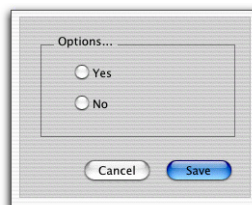
Die Composite-Eigenschaft

Die Composite-Eigenschaft löst ein Darstellungsproblem, das unter Mac OS X auftritt.

In bestimmten Fällen stellen Steuerelemente, die in einem Metal- oder Drawer-Fenster oder auf einem anderen Steuerelement platziert wurden, ihren rechteckigen Umriß als einen weißen Hintergrund dar, solange die Composite-Eigenschaft des Fensters nicht eingeschaltet wurde. Das Problem tritt nicht auf, solange die Steuerelemente wirklich rechteckig sind. Bei Fenstern vom Typ Metal oder Drawer ist die Composite-Eigenschaft automatisch aktiviert. Das Problem kann auch bei normalen Fenstern auftreten. Wenn z.B. Steuerelemente auf einem Placard-Steuerelement in einem Dokument-Fenster platziert werden, zeigt sich dieses Problem.



In diesem Beispiel befinden sich PushButtons, RadioButtons und eine GroupBox innerhalb eines Placard-Steuerelements und die Composite-Eigenschaft des Fensters ist ausgeschaltet. Wenn Sie die Composite-Eigenschaft einschalten, verschwinden die häßlichen Rechtecke, die sich um jedes Steuerelement befinden.



Benutzerdefinierte Fenstertypen

Wenn die Eigenschaft `MacProcID` auf einen Wert ungleich Null gesetzt ist (was man nur in der IDE machen kann), wird das Fenster in Übereinstimmung mit diesem Wert gezeichnet. Wenn Sie dies so machen, wird das Verhalten des Fensters dem Wert der `Frame`-Eigenschaft entsprechen, das Aussehen wird jedoch von der Fensterdefinition gesteuert, die `MacProcID` zur Verfügung stellt. Dies erlaubt es Ihnen, Fenstertypen zu definieren, die nicht direkt in REALbasic definiert sind (z.B. über die `Frame`-Eigenschaft).

Weitere Infos zu `MacProcID`-Werten sehen Sie in der Apple-Entwicklerdokumentation unter „Window Definition IDs“ unter <http://developer.apple.com/techpubs/macos8/HumanInterfaceToolbox/WindowManager/WindowMgr8Ref/WindowMgrRef.b.html> nach.

Folgende Tabelle führt die möglichen Werte von `MacProcID` auf. Werte zwischen 1024 und 1087 benötigt der Appearance Manager (Mac OS8 und neuer); andere Werte sind Werte aus Zeiten vor dem Appearance Manager. Unter Windows steuert der Wert der `Frame`-Eigenschaft den Fenstertyp, egal welchen Wert `MacProcID` hat.

Wert	Beschreibung
1	Modal dialog box
2	Modeless dialog box
3	Modeless dialog box with shadow
4	Movable window with no size box
5	Movable dialog box with title bar
8	Movable window with zoom box and size box
12	Movable window with zoom box and no size box
16	Round corner window (adding 2, 4, or 6 to 16 changes the curvature of the corners)
1024	Movable window with no size box or zoom box
1025	Movable window with size box
1026	Movable window with vertical zoom box and no size box
1027	Movable window with vertical zoom box and size box
1028	Movable window with horizontal zoom box and no size box
1029	Movable window with horizontal zoom box and size box
1030	Movable window with full zoom box and no size box
1031	Movable window with full zoom box and size box
1040	Modeless dialog box
1041	Modeless dialog box with shadow
1042	Modal dialog box
1043	Movable modal dialog box
1044	Alert box
1045	Movable alert box
1046	Movable modal dialog box with size box
1057	Floating window with no size box or zoom box
1059	Floating window with size box
1061	Floating window with vertical zoom box
1063	Floating window with vertical zoom box and size box
1065	Floating window with horizontal zoom box
1067	Floating window with horizontal zoom box and size box
1069	Floating window with full zoom box
1071	Floating window with full zoom box and size box
1073	Floating window with side title bar
1075	Floating window with side title bar and size box

Wert	Beschreibung
1077	Floating window with side title bar and vertical zoom box
1079	Floating window with side title bar, vertical zoom box, and size box
1081	Floating window with side title bar and horizontal zoom box
1083	Floating window with side title bar, horizontal zoom box, and size box
1085	Floating window with side title bar and full zoom box
1087	Floating window with side title bar, full zoom box, and size box
1985	Floating window with no size box or zoom box
1987	Floating window with size box
1989	Floating window with zoom box
1991	Floating window with zoom box and size box
1993	Floating window with side title bar and no size or zoom boxes
1995	Floating window with side title bar and size box
1997	Floating window with side title bar and zoom box
1999	Floating window with side title bar, size box, and zoom box.

Benutzerdefinierte Mauszeiger

Die MouseCursor-Eigenschaft eines Steuerelements legt die Form des Mauszeigers fest, wenn dieser sich in einem Fenster befindet, vorausgesetzt, die MouseCursor-Eigenschaften der Application-Klasse Nil ist. Wenn Sie auch möchten, daß sich das Aussehen des Mauszeigers ändert, wenn es sich über einem Steuerelement innerhalb des Fensters befindet, müssen Sie entweder der MouseCursor-Eigenschaft des Fenster einen neuen Wert zuweisen oder vorübergehend die MouseCursor-Eigenschaft des Fensters auf Nil setzen und die MouseCursor-Eigenschaft des Steuerelements auf das gewünschten Aussehen. Für ein Beispiel schauen Sie im Abschnitt der MouseCursor-Klasse nach.

Beispiele

Dieses Beispiel setzt die Hintergrundfarbe des Fensters auf Grau, vorausgesetzt, die BackColor-Eigenschaft ist auf True gesetzt.

```
backcolor=RGB(80,80,80)
```

Dieses Beispiel verbreitert das Fenster um 20 Pixel.

```
width=width+20
```

Dieses Beispiel ändert den Fenstertitel.

```
title="Dokument 1"
```

Drawinto Beispiel

Dieses Beispiel zeichnet den Inhalt eines Fensters in ein Graphics-Objekt. Wenn Sie beispielsweise ein neues Projekt erzeugen, ein zweites Fenster hinzufügen und in diesem ein Canvas-Steuerelement platzieren, können Sie mit folgender Programmzeile, die im Paint-Event des ersten Fensters stehen muß, den Inhalt des ersten Fensters im Canvas-Steuerelement des zweiten Fensters abbilden:

```
self.drawinto(dialog1.canvas1.graphics,0,0)
```

Zugriff auf Eigenschaften und Steuerelemente anderer Fenster

Wenn Sie die Eigenschaften eines anderen Fensters ändern, können Sie hierzu zwei Wege wählen. Wenn Sie nur eine Instanz eines zu referenzierenden Fensters haben, können Sie den Objektnamen des Fensters als Referenz verwenden. Wenn Sie Ihr Fenster beispielsweise window1 genannt haben und in diesem ein EditField namens EditField1 liegt, dessen Text-Eigenschaft Sie auf den Wert „Fred“ ändern möchten, können Sie folgende Syntax verwenden:

```
window1.EditField1.text="Fred"
```

Wenn Sie mehrere Instanzen der gleichen Fensterklasse zur gleichen Zeit offen haben, müssen Sie wie in diesem Beispiel eine Referenz auf das Zielfenster mit angeben. In diesem Beispiel wird ein Fenster geöffnet und sein Fenstertitel geändert. „anotherWindow“ ist hier eine Fensterklasse aus dem Projekt-Fenster.

```
Dim w As anotherWindow
w=New anotherWindow
w.title="Ihr Ergebnis"
```

Den Mauszeiger verändern

Die MouseCursor-Eigenschaft eines Steuerelements legt die Form des Mauszeigers fest, wenn dieser sich in einem Fenster befindet, vorausgesetzt, die MouseCursor-Eigenschaften der Application-Klasse Nil ist. Wenn MouseCursor der Application-Klasse nicht Nil ist, wird die MouseCursor-Eigenschaft des Fensters ignoriert. Eine ausführliche Beschreibung finden Sie bei den Klassen Application und Control.

Drag und Drop

Eine Erklärung, wie Sie mit Drag und Drop umgehen können, finden Sie in der Beschreibung der Control und der DragItem-Klasse.

WindowCount Funktion

Liefert die Anzahl offener Fenster.

Syntax

Ergebnis=WindowCount

Teil	Art	Beschreibung
Ergebnis	<u>Integer</u>	Die Anzahl offener Fenster.

Hinweis

Die Funktion **WindowCount** liefert die Anzahl offener Fenster. Dies beinhaltet auch nicht sichtbare (ausgeblendete) Fenster.

Beispiele

Bei der Beschreibung der Funktion Window finden Sie ein entsprechendes Beispiel.

Siehe auch

Window Funktion.

WordApplication Klasse

Wird zur Automatisierung von Microsoft Word verwendet.

Übergeordnete Klasse

OLEObject

Siehe auch

Office, ExcelApplication, PowerPointApplication, OLEObject Klassen.

XMLAttribute Klasse

Repräsentiert einen XML-Attribut-Knoten.

Übergeordnete Klasse

XMLNode

Eigenschaften

Name	Typ	Beschreibung
OwnerElement	<u>XMLElement</u>	Eine Referenz auf das <u>XMLElement</u> von dem dieses Attribut ein Teil ist.

Hinweise

Um Attribute über den Index zu ermitteln, können Sie die GetAttributeNode-Methode von XMLElement verwenden.

Siehe auch

XMLNode, XMLElement, XMLDocument Klassen.

XMLAttributList Klasse

Eine Helfer-Klasse, die im [XMLReader-StartElement-Event](#) und im [XMLXsltHandler-StartElement-Event](#) verwendet wird. Es handelt sich um eine grundlegende Struktur für eine Sammlung von Attributen. Die Attribute können über den Index oder den Namen ermittelt werden.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
Count	Integer	Die Anzahl von Attributen in der Liste.

Methoden

Name	Parameter	Beschreibung
Key	Index als Integer	Ermittelt ein Attribut über die Index-Position. Liefert das Attribut als einen String. <i>Index</i> ist Null-basiert.
Value	Index als Integer	Ermittelt einen Attribut-Wert über die Index-Position. Liefert das Attribut als einen String. <i>Index</i> ist Null-basiert.
Value	Key als String	Ermittelt einen Attribut-Wert.

Siehe auch

[XMLAttribute](#), [XMLNode](#), [XMLDocument](#) Klassen.

XMLCDATASection Klasse

XMLCDATASection erlaubt es Ihnen, den [IsA](#)-Operator zu verwenden, um festzustellen, ob sich ein [XMLNode](#) in einer **XMLCDATASection** befindet.

Übergeordnete Klasse

[XMLNode](#)

Eigenschaften

Keine

Methoden

Keine

Siehe auch

[XMLNode](#), [XMLAttribute](#), [XMLComment](#), [XMLElement](#), [XMLDocument](#) Klassen.

XMLComment Klasse

XMLComment erlaubt Ihnen, den [IsA](#)-Operator zu verwenden, um festzustellen, ob ein [XMLNode](#) ein **XMLComment** ist.

Übergeordnete Klasse

[XMLNode](#)

Eigenschaften

Keine

Methoden

Keine

Siehe auch

[XMLNode](#)

XMLContentModel Klasse

Repräsentiert eine gelesene DTD-Element-Deklaration. Expat legt diese Struktur fest und **XmlContentModel** ist eine direkte Repräsentation davon. Es wird an den ElementDecl-Event von XMLReader übergeben. Dies ist hilfreich für jeden, der DTD-Überprüfungs-Routinen entwickeln will.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
Name	<u>String</u>	Name des Objekts.
NumChildren	<u>Integer</u>	Anzahl von Kindern.
Quant	<u>Integer</u>	XML-Inhalts-Quant. Im Abschnitt Konstanten finden Sie die gültigen Quants.
Type	<u>Integer</u>	XML-Inhals-Typ. Im Abschnitt Konstanten finden Sie die gültigen Typen.

Methoden

Name	Parameter	Beschreibung
Child	Index als <u>Integer</u>	Liefert ein XMLContentModel.

Konstanten

Dies sind die Konstanten, die mit dem Wert der Type-Eigenschaft verglichen werden können:

Wert	Konstante
1	XML_CTYPE_EMPTY
2	XML_CTYPE_ANY
3	XML_CTYPE_MIXED
4	XML_CTYPE_NAME
5	XML_CTYPE_CHOICE
6	XML_CTYPE_SEQ

Dies sind die Konstanten, die mit dem Wert der Quant-Eigenschaft verglichen werden:

Wert	Konstante
0	XML_CQUANT_NONE
1	XML_CQUANT_OPT
2	XML_CQUANT_REP
3	XML_CQUANT_PLUS

Hinweise

Wenn der Typ XML_CTYPE_EMPTY oder XML_CTYPE_ANY ist, dann erhält der Quant den Wert XML_CQUANT_NONE und die anderen Felder werden 0 oder NULL. Wenn der Typ XML_CTYPE_MIXED ist, dann erhält Quant den Wert *None* oder *REP* und numChildren enthält die Anzahl an Elementen, die vermischt werden können und *children* zeigt auf ein Array aus XMLContent-Zellen, die alle den Typ XML_CTYPE_NAME haben ohne Quantifizierung.

Wenn der Typ XML_CTYPE_NAME ist, dann zeigt *name* auf den Namen und das *numChildren*-Feld wird 0 und *children* wird NULL. Die *quant*-Felder weisen auf Quantifier hin, die auf diesen Namen platziert sind.

CHOICE und SEQ erhalten den Namen NULL, die Anzahl der Kinder ist in *numChildren* und *children* zeigt, rekursiv, auf ein Array von XMLContent-Zellen.

Die Typen EMPTY, ANY und MIXED treten nur in der obersten Ebene auf.

Siehe auch

XMLReader Klasse.

XMLDocument Klasse

Repräsentiert ein XML-Dokument. Es wird sowohl zum Einlesen von vorhandenen XML-Daten in ein DOM-Dokument als auch zum Erzeugen eines neuen XML-Dokuments verwendet.

Übergeordnete Klasse

XMLNode

Konstruktor

Name	Parameter	Beschreibung
XMLDocument		Erzeugt ein leeres Dokument.
XMLDocument	XML als <u>String</u>	Liest den übergebenen XML-String in das Dokument ein.
XMLDocument	f als <u>FolderItem</u>	Liest die übergebene XML-Datei in das Dokument ein.

Eigenschaften

Name	Typ	Beschreibung
DocumentElement	<u>XMLElement</u>	Verweist auf das Element in der obersten Ebene im Dokument.
PreserveWhiteSpace	<u>Boolean</u>	Wenn dies auf <u>False</u> gesetzt wird, werden Return und Tabs zwischen den Elementen entfernt. Setzt man es auf <u>True</u> , verbleiben diese an ihren Stellen. Die Voreinstellung ist <u>False</u> .
ToString	<u>String</u>	Enthält eine String-Repräsentation des Dokuments.

Methoden

Name	Parameter	Beschreibung
CreateAttribute	Name als <u>String</u>	Erzeugt einen Attribut-Knoten als ein <u>XMLAttribute</u> .
CreateAttribute	URI als <u>String</u> , Name als <u>String</u>	Erzeugt einen Attribut-Knoten als ein <u>XMLAttribute</u> mit einer Namensraum-Deklaration.
CreateCDATASection	Data als <u>String</u>	Erzeugt einen CDATA-Abschnitt aus <i>Data</i> und liefert diesen als ein <u>XMLCDATASection</u> .
CreateComment	Data als <u>String</u>	Erzeugt ein <u>XMLComment</u> aus <i>Data</i> und liefert es als ein <u>XMLComment</u> .
CreateElement	TagName als <u>String</u>	Erzeugt ein Element als ein <u>XMLElement</u> .
CreateElement	URI als <u>String</u> , Tagname als <u>String</u>	Erzeugt ein Element aus einer Namensraum-Deklaration als ein <u>XMLElement</u> .
CreateProcessingInstruction	Target als <u>String</u> , Data als <u>String</u>	Erzeugt eine <u>XMLProcessingInstruction</u> aus dem übergebenen Schlüsselwort <i>target</i> und <i>data</i> .
CreateTextNode	Data als <u>String</u>	Erzeugt einen Textknoten aus <i>Data</i> und liefert ihn als <u>XMLTextNode</u> .
ImportNode	foreignNode als <u>XMLNode</u> , [deep als <u>Boolean</u>]	Kopiert einen Knoten aus einem anderen XMLDocument in das aktuelle Dokument. Wenn der optionale Parameter <i>deep True</i> ist, importiert ImportNode alle Kindknoten von <i>foreignNode</i> . Nachdem Sie sie importiert haben, müssen Sie diese irgendwohin platzieren.
LoadXML	Doc als <u>String</u>	Liest den übergebenen XML-String in das Dokument ein.
LoadXML	f als <u>FolderItem</u>	Liest die übergebene XML-Datei in das Dokument ein.

Name	Parameter	Beschreibung
Transform	xsl als <u>String</u> oder xsl als <u>XMLStyleSheet</u>	Erzeugt ein neues XML-Dokument, das das Ergebnis der Zuweisung eines SXLT-Stylesheets ist. Liefert ein XML-Document .
	[saxHandler als <u>XMLSxltHandler</u>]	Verwenden Sie optional den Event-Handler <i>saxHandler</i> , um SAX-Style-Events zum Ausgabe-XML während der Umwandlung zu erhalten.

Beispiele

Dieses Beispiel erzeugt ein sehr einfaches XML-Dokument.

```
Dim xml as XmlDocument
Dim root, person, firstname, lastname as XMLNode
xml = New XmlDocument
root = xml.AppendChild(xml.CreateElement("root"))
person = root.AppendChild(xml.CreateElement("person"))
firstname = person.AppendChild(xml.CreateElement("firstname"))
firstname.AppendChild(xml.CreateTextNode("John"))
lastname = person.AppendChild(xml.CreateElement("lastname"))
lastname.AppendChild(xml.CreateTextNode("Doe"))
//create_output is a muti-line EditField
create_output.text = xml.ToString
```

Dieses Beispiel fügt die Attribute „ID“ und „active“ einem Element hinzu.

```
Dim xml as XmlDocument
Dim root, node, attNode as XMLNode

xml = New XmlDocument
root = xml.AppendChild(xml.CreateElement("root"))
node = root.AppendChild(xml.CreateElement("person"))

attNode = xml.CreateAttribute("id")
attNode.value = "55"

node.SetAttribute("active", "true")

create_output.text = xml.ToString
```

Dieses Beispiel liest ein XML-Dokument ein und zeigt die Anzahl der „Person“-Knoten und den Dokument-Text in einem EditField an.

```
Dim xdoc as XmlDocument
Dim node as XMLNode
Dim i, count as Integer
Dim out as String

// create a new document
xdoc = New XmlDocument

xdoc.PreserveWhitespace = False

// load and parse the xml in the EditField, parse_inputText
xdoc.LoadXml(parse_inputText.Text)

// lets first do some simple, static traversing of the xml
// this doesn't do any error checking to make sure stuff exists
// so if the document changes, it will break.

count = xdoc.DocumentElement.childCount
out = "People: " + Str(count) //number of Person nodes
out = out + EndOfLine

// childNodes are 0 based.
For i = 0 to count - 1

    node = xdoc.DocumentElement.Child(i)

    // this gets the textnode of the <firstname> node
    out = out + node.FirstChild.FirstChild.Value
```

```
    // this gets the textnode of the <lastname> node
    out = out + " " + node.LastChild.FirstChild.Value
    out = out + EndOfLine
```

```
next
```

```
//display results in the EditField parse_output
parse_output.text = out
```

Siehe auch

[XMLNode](#), [XMLElement](#), [XMLAttribute](#), [XMLTextNode](#), [XMLCDATASection](#), [XMLComment](#), [XMLProcessingInstruction](#), [XMLNodeList](#), [XMLException](#), [XMLDOMException](#) Klassen.

XMLDOMException Error

Diese Exception kann während des Erzeugens eines DOM-Dokuments auftreten, wenn dabei etwas schief ging. Es tritt zum Beispiel auf, wenn mehrere Elemente der obersten Ebene eines Dokumentes angehängt werden.

Übergeordnete Klasse

[RuntimeException](#)

Eigenschaften

Keine. Verwenden Sie die Eigenschaften Message und ErrorNumber der [RuntimeException](#)-Klasse.

Konstanten

Vergleichen Sie ErrorNumber mit diesen Klassen-Konstanten anstatt mit den Werten, um Ihren Code besser lesbar zu halten.

Wert	Konstante
0	OK
1	INDEX_SIZE_ERR
2	DOMSTRING_SIZE_ERR
3	HIERARCHY_REQUEST_ERR
4	WRONG_DOCUMENT_ERR
5	INVALID_CHARACTER_ERR
6	NO_DATA_ALLOWED_ERR
7	NO_MODIFICATION_ALLOWED_ERR
8	NOT_FOUND_ERR
9	NOT_SUPPORTED_ERR
10	INUSE_ATTRIBUTE_ERR
11	INVALID_STATE_ERR
12	SYNTAX_ERR
13	INVALID_MODIFICATION_ERR
14	NAMESPACE_ERR
15	INVALID_ACCESS_ERR
16	INVALID_NODE_TYPE
17	QUERY_PARSE_ERR
18	QUERY_EXECUTION_ERR
19	NOT_OK

XMLElement Klasse

Wird zur Repräsentation eines XML-Elements verwendet.

Übergeordnete Klasse

[XMLNode](#)

Eigenschaften

Name	Typ	Beschreibung
AttributeCount	Integer	Die Anzahl an Attributen, die dieses Element enthält.

Methoden

Name	Parameter	Beschreibung
GetAttribute	Name als String oder URI als String , Name als String	Ermittelt den Wert des durch <i>Name</i> festgelegten Attributs.
GetAttributeNode	Name als String oder Index als Integer oder URI als String , Name als String	Ermittelt einen XML-Attribut-Knoten des durch <i>Name</i> oder durch die <i>Index</i> -Position festgelegten Attributs.
RemoveAttribute	Name als String	Entfernt das durch <i>Name</i> oder durch die Knoten-Referenz festgelegte Attribut. Liefert den entfernten Knoten optional als ein XMLAttribute .
RemoveAttribute	attributeNode als XMLAttribute	Entfernt das durch die Knoten-Referenz festgelegte Attribut. Liefert den entfernten Knoten optional als ein XMLAttribute .
SetAttribute	Name als String , Value als String oder URI als String , Name als String , Value als String	Setzt ein Attribut des aktuellen Elements. Die zweite Syntax setzt ein Attribut und deklariert einen Namensraum.
SetAttributeNode	AttributeNode als XMLAttribute , [ns als Boolean]	Setzt einen Attribut-Knoten. Liefert optional eine Referenz auf einen Knoten als ein XMLAttribute , das den gleichen Namen hatte und ersetzt wurde. Der optionale Parameter <i>ns</i> legt fest, ob Namensraum-Daten, die in dem übergebenen XMLAttribute gefunden wurden, verwendet und deklariert werden sollen.

Siehe auch

[XMLNode](#), [XMLDocument](#) Klassen.

XMLException Fehler

Diese Exception kann aus mehreren Gründen auftreten, ebenso wie Fehler beim Einlesen von XML-Dokumenten. Weitere Informationen über den Fehler finden Sie in den Eigenschaften Message und ErrorNumber der [RuntimeException](#)-Klasse.

Übergeordnete Klasse

[RuntimeException](#)

Eigenschaften

Name	Typ	Beschreibung
Line	<u>String</u>	Wenn es ein Fehler beim Einlesen ist, ist dies die Zeilennummer, in der das Einlesen scheiterte.
Node	<u>String</u>	Wenn es ein Fehler beim Einlesen ist, ist dies der Knoten, an dem das Einlesen scheiterte.

Siehe auch

RuntimeException Klasse

XMLNode Klasse

XMLNode ist die Basis-Klasse für die meisten XML-Document Object Model (DOM)-Klassen und implementiert viele der gängigen Eigenschaften und Funktionen eines DOM-Knoten. Diese Klasse wird normalerweise nicht direkt instanziiert. Die meisten Objekte werden aus Methoden der XMLDocument-Klasse erzeugt.

Beachten Sie, daß nicht jede Eigenschaft bei jedem Knoten-Typ verwendet wird. Die Value-Eigenschaft wird zum Beispiel bei XMLAttribute und XMLTextNode, nicht jedoch bei XMLElement verwendet.

Übergeordnete Klasse

Object

Eigenschaften

Name	Typ	Beschreibung
ChildCount	<u>Integer</u>	Die Anzahl an Kind-Knoten, die dieser Knoten enthält.
FirstChild	XMLNode	Das erste Kind dieses Knotens.
LastChild	XMLNode	Das letzte Kind dieses Knotens.
LastError	<u>Integer</u>	Speichert einen Error-Code, wenn ein Fehler auftritt.
LocalName	<u>String</u>	Der Name des Knotens ohne das Präfix.
Name	<u>String</u>	Der Name dieses Knotens.
NamespaceURI	<u>String</u>	Der NamespaceURI dieses Knotens.
NextSibling	XMLNode	Der Text-Knoten.
OwnerDocument	<u>XMLDocument</u>	Das XMLDocument, das diesen Knoten enthält.
Parent	XMLNode	Das Elter dieses Knotens.
Prefix	<u>String</u>	Der Namensraum-Präfix dieses Knotens.
PreviousSibling	XMLNode	Der vorhergehende Knoten.
ToString	<u>String</u>	Eine String-Repräsentation dieses Knotens.
Type	<u>Integer</u>	<u>Integer</u> -Konstante, die den Typ, wie Element, Attribut, Textnode und so weiter markiert. Verwenden Sie die Konstanten des <u>XMLNodeType</u> -Objekts um die Werte zu ermitteln oder zu setzen.
Value	<u>String</u>	Wird in manchen Knoten zum Setzen oder Ermitteln von Werten, wie z.B. TextNode verwendet. Wird nicht für alle Knotentypen unterstützt.

Methoden

Name	Parameter	Beschreibung
AppendChild	NewChild als XMLNode	Fügt nach dem letzten Kind ein weiteres hinzu. Es liefert optional eine Referenz auf das neue Kind als ein XMLNode .
Child	Index als <u>Integer</u>	Liefert das Kind XMLNode an der durch <i>Index</i> angegebenen Position. <i>Index</i> ist Null-basiert.
Clone	Deep als <u>Boolean</u>	Dupliziert den aktuellen Knoten. Der Parameter Deep legt fest, ob auch alle Kind-Knoten dupliziert werden sollen. Es liefert ein XMLNode .

Name	Parameter	Beschreibung
Compare	NodeToCompare als XMLNode	Vergleicht zwei Knoten. Der Rückgabewert arbeitet wie <u>StrComp</u> . Es liefert einen <u>Integer</u> : Wenn <i>string1</i> < <i>string2</i> dann liefert es -1 Wenn <i>string1</i> = <i>string2</i> dann liefert es 0 Wenn <i>string1</i> > <i>string2</i> dann liefert es 1
Insert	NewChild als XMLNode , RefChild als XMLNode	Fügt <i>NewChild</i> vor der Position von <i>RefChild</i> ein. Es liefert optional eine Referenz auf den eingefügten Knoten als ein XMLNode .
RemoveChild	OldChild als XMLNode	Entfernt <i>OldChild</i> .
ReplaceChild	NewChild als XMLNode , OldChild als XMLNode	Ersetzt oldChild mit NewChild. Liefert optional eine Referenz auf das neue Kind als ein XMLNode .
XQL	Query als <u>String</u> , [Map als <u>String</u>]	Führt eine XPath-Abfrage aus und liefert eine <u>XMLNodeList</u> der resultierenden Knoten. Wenn die Abfrage die Namensraum-Referenzen beinhaltet, müssen Sie Deklarationen für diese in Form eines String-Arrays anbieten, z.B. („ns1“ „http://foo“ „ns2“ „http://bar“).

Hinweise

Die REALbasic XML-Library kann XML-Dokumente einlesen und erzeugen, bietet DOM Level 2-Unterstützung, XPath, XSLT und volle Namensraum-Unterstützung, inklusive Überprüfung während der Erzeugung.

Siehe auch

[XMLDocument](#), [XMLElement](#), [XMLNodeList](#) Klassen.

XMLNodeList Klasse

Speichert das Ergebnis einer XQL-Abfrage, die durch die XQL-Methode der XMLNode-Klasse durchgeführt wurde.

Übergeordnete Klasse

[Object](#)

Eigenschaften

Name	Typ	Beschreibung
LastError	<u>Integer</u>	Der letzte Fehlercode, der erhalten wurde.
Length	<u>Integer</u>	Die Anzahl der Knoten in der Liste.

Methoden

Name	Parameter	Beschreibung
Item	Index als <u>Integer</u>	Ermittelt den angegebenen XML-Knoten aus der Liste und gibt ihn als ein XMLNode zurück.
ToString	[WrapperTag als <u>String</u>]	Liefert eine <u>String</u> -Repräsentation der Knoten. Der optionale Parameter <i>WrapperTag</i> plazierte ein Tag um die NodeList um sie lesbar zu machen.

Siehe auch

[XMLNode](#) Klasse

XMLNodeType Objekt

Erlaubt den Zugriff auf die Typ-Konstanten für die Type-Eigenschaft von XMLNode.

Konstanten

Das **XMLNodeType**-Objekt enthält Konstanten für die Type-Eigenschaft. Diese werden in der folgenden Tabelle aufgelistet.

Wert	Konstante
1	Element_Node
2	Attribute_Node
3	Text_Node
4	CData_Section_Node
5	Entity_Reference_Node
6	Entity_Node
7	Processing_Instruction_Node
8	Comment_Node
9	Document_Node
10	Document_Type_Node
11	Document_Fragment_Node
12	Notation_Node
13	Other_Node

Siehe auch

[XMLNode](#) Klasse.

XMLProcessingInstruction Klasse

Repräsentiert eine XML-Ausführungsanweisung. Diese Klasse erlaubt es, den IsA-Operator zu verwenden, um zu prüfen, ob ein XML-Node eine XML-Ausführungsanweisung ist.

Übergeordnete Klasse

[XMLNode](#)

Eigenschaften

Keine

Methoden

Keine

Siehe auch

[XMLNode](#)

XMLReader Klasse

Dies ist eine Abstraktionsklasse der neuesten Version von Expat (XML Parser Toolkit). Expat ist eine Library zum Einlesen von XML-Dokumenten. Erzeugen Sie eine Unterklasse von **XMLReader**, um Zugriff auf die Events zu erhalten. Die meisten der Events, Methoden und Eigenschaften sind genaue Entsprechungen der Expat-Funktionen.

Übergeordnete Klasse

[Object](#)

Konstruktor

Name	Parameter	Beschreibung
XMLReader		Erzeugt ein XMLReader-Objekt.

Name	Parameter	Beschreibung
XMLReader	[XMLEncoding als String], [namespaceSeparator als String]	Erzeugt ein XMLReader-Objekt. Der XMLEncoding-Parameter stellt das Encoding für den Parser ein. Das voreingestellte Encoding ist UTF-8. Der Parameter <i>namespaceSeparator</i> zwingt Expat die Namensraum-Definitionen zu expandieren.

Eigenschaften

Name	Typ	Beschreibung
Base	String	Stellt die Basis ein, die für das Auflösen relativer URLs in System-Identifiern in Deklarationen verwendet werden soll.
CurrentByteCount	Integer	Die Anzahl an Bytes im aktuellen Event.
CurrentByteIndex	Integer	Die Byte-Position, an der der Fehler auftrat.
CurrentColumnNumber	Integer	Die Spalten-Nummer, in der sich der Parser momentan befindet.
CurrentLineNumber	Integer	Die Zeilen-Nummer, in der sich der Parser derzeit befindet.
ErrorCode	Integer	Der letzte Fehler-Code.
SetDefaultHandler	Boolean	Stellt Expat so ein, daß er unbekannte Daten an den Default-Event schickt Entities werden nicht eingelesen.
SetDefaultHandlerExpand	Boolean	Stellt Expat so ein, daß er unbekannte Daten an den Default-Event schickt und Entities expandiert werden.

Events

Name	Parameter	Beschreibung
AttListDecl	elname als String , attname als String , att_type als String , dflt als String , isrequired als Boolean	Wird bei einer DTD-Attliis-Deklaration ausgelöst.
Characters	s als String	Wird bei Zeichen-Daten, s, ausgelöst.
Comment	data als String	Wird bei einem XML-Kommentar <i>data</i> ausgelöst.
Default	s als String	Wird bei jedem Zeichen im XML-Dokument ausgelöst, für das es keinen passenden Handler gibt.
ElementDecl	name als String , Content als XMLContentModel	Wird bei einer DTD-Element-Deklaration ausgelöst.
EndCDATA		Wird am Ende von CDATA ausgelöst.
EndDoctypeDecl		Wird am Ende einer DocType-Deklaration ausgelöst.
EndDocument		Wird am Ende eines Dokuments ausgelöst.
EndElement	Name als String	Wird am Ende des Elements <i>Name</i> ausgelöst.
EndPrefixMapping	prefix als String	Wird am Ende einer Namensraum-Deklaration ausgeführt.
EntityDecl	entityName als String , is_parameter_entity als Boolean , Value als String , Base als String , SystemID als String , publicID als String , notationName als String	Wird bei einer Entität-Deklaration ausgelöst. is_parameter_entity wird True, wenn die Entität eine Parameter-Entität ist. Bei internen Entitäten (<!ENTITY foo "bar">) wird <i>Value</i> ausgefüllt und systemID, publicID und notation-Name werden leere Strings. Bei externen Entitäten wird <i>Value</i> ein leerer String und systemID wird ausgefüllt. Das publicID-Argument bleibt so lange leer bis ein öffentlicher Identifier angeboten wird. Das notationName-Argument bekommt nur für ungelesene Entitäts-Deklarationen einen veröffentlichten Wert.

Name	Parameter	Beschreibung
ExternalEntityRef	Context as <u>String</u> , Base as <u>String</u> , SystemID as <u>String</u> , PublicID as <u>String</u>	Wird bei einer externen Entitäts-Referenz ausgelöst. Liefert ein Boolean. Geben Sie False zurück, um das Einlesen zu stoppen und True, um anzuzeigen, daß die Verwaltung der externen Entität gelang.
NotationDecl	NotationName as <u>String</u> , Base as <u>String</u> , SystemID as <u>String</u> , PublicID as <u>String</u>	Wird bei einer Notation ausgelöst.
NotStandalone		Dies wird aufgerufen, wenn das Dokument nicht Standalone ist, das bedeutet, es besitzt ein externes Subset oder eine Referenz auf eine Parameter-Entität, es ist jedoch nicht Standalone = Yes. Liefert ein Boolean. Geben Sie False zurück, um das Einlesen zu stoppen..
ProcessingInstruction	Target as <u>String</u> , Data as <u>String</u>	Wird bei einer Ausführungsanweisung <i>Target</i> und <i>Data</i> ausgelöst
SkippedEntity	EntityName as <u>String</u> , is_parameter_entity as <u>Boolean</u>	Wird ausgelöst, wenn eine Entität erreicht wird, aber Entitäten momentan nicht expandiert werden.
StartCDATA		Wird beim Beginn von CDATA ausgelöst.
StartDoctypeDecl	DoctypeName as <u>String</u> , SystemID as <u>String</u> , PublicID as <u>String</u> , has_internal_subset as <u>Boolean</u>	Wird zu Beginn einer DocType-Deklaration ausgelöst.
StartDocument		Wird bei Beginn eines Dokumentes ausgelöst.
StartElement	Name as <u>String</u> , AttributeList as <u>XMLAttributeList</u>	Wird bei Beginn eines XML-Elements, Name und AttributeList ausgelöst.
StartPrefixMapping	Prefix as <u>String</u> , uri as <u>String</u>	Wird zu Beginn einer Namensraum-Deklaration, Prefix und URI ausgelöst.
XMLDecl	version as <u>String</u> , XMLEncoding as <u>String</u> , Standalone as <u>Boolean</u>	Wird bei einem XML-Deklarations-Knoten ausgelöst.

Methoden

Name	Parameter	Beschreibung
CreateExternalEntityReader	XMLEncoding als <u>String</u>	
DefaultCurrent		Wenn dies innerhalb eines Events aufgerufen wird, werden die aktuellen Daten an das Default-Event umgeleitet.
FreeExternalEntityReader		
Parse	s als <u>String</u> , [isFinal als <u>Boolean</u>]	Liest die Daten aus dem übergebenen String s. Der optionale Parameter <i>isFinal</i> teilt Expat mit, was zu tun ist, wenn das Ende der Daten erreicht wird und das XML nicht korrekt aufgebaut ist, z.B., wenn es in der Mitte eines Elements endet. Wenn Sie <i>isFinal</i> auf False setzen, wird Expat angewiesen, keinen Fehler auszulösen, da in einem weiteren Aufruf weitere Daten folgen werden.

Name	Parameter	Beschreibung
Parse	Datei als <u>FolderItem</u> , [isFinal als <u>Boolean</u>]	Liest die Daten aus dem übergebenen Dokument Datei. <i>isFinal</i> weist Expat an, was zu tun ist, wenn, wie oben beschrieben, das Ende der Daten erreicht wird, weil das XML nicht wohlgeformt ist.
Reset	[XMLEncoding als <u>String</u>]	Setzt den Parser zurück. Der optionale Parameter <i>XMLEncoding</i> legt das Encoding der zu lesenden Daten fest. Liefert ein <u>Boolean</u>

Konstanten

Wert	Konstante
0	XML_ERROR_NONE
1	XML_ERROR_NO_MEMORY
2	XML_ERROR_SYNTAX
3	XML_ERROR_NO_ELEMENTS
4	XML_ERROR_INVALID_TOKEN
5	XML_ERROR_UNCLOSED_TOKEN
6	XML_ERROR_PARTIAL_CHAR
7	XML_ERROR_TAG_MISMATCH
8	XML_ERROR_DUPLICATE_ATTRIBUTE
9	XML_ERROR_JUNK_AFTER_DOC_ELEMENT
10	XML_ERROR_PARAM_ENTITY_REF
11	XML_ERROR_UNDEFINED_ENTITY
12	XML_ERROR_RECURSIVE_ENTITY_REF
13	XML_ERROR_ASYNC_ENTITY
14	XML_ERROR_BAD_CHAR_REF
15	XML_ERROR_BINARY_ENTITY_REF
16	XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
17	XML_ERROR_MISPLACED_XML_PI
18	XML_ERROR_UNKNOWN_ENCODING
19	XML_ERROR_INCORRECT_ENCODING
20	XML_ERROR_UNCLOSED_CDATA_SECTION
21	XML_ERROR_EXTERNAL_ENTITY_HANDLING
22	XML_ERROR_NOT_STANDALONE
23	XML_ERROR_UNEXPECTED_STATE
24	XML_ERROR_ENTITY_DECLARED_IN_PE
25	XML_ERROR_FEATURE_REQUIRES_XML_DTD
26	XML_ERROR_CANT_CHANGE_FEATURE_ONCE_PARSING

XMLReaderException Klasse

Kann auftreten, wenn während des Einlesens einer XML-Datei mit XMLReader ein Fehler auftritt. Überprüfen Sie die ErrorCode- und Message-Eigenschaften der RuntimeException-Klasse sowie die Eigenschaften dieser Klasse um Informationen zu dem Fehler zu erhalten.

Übergeordnete Klasse

RuntimeException

Eigenschaften

Name	Typ	Beschreibung
CurrentByteCount	Integer	Die Anzahl an Bytes im aktuellen Event.
CurrentByteIndex	Integer	Die Byte-Position, an der der Fehler auftrat.
CurrentColumnNumber	Integer	Die Spalten-Nummer, in der der Fehler auftrat.
CurrentLineNumber	Integer	Die Zeilen-Nummer, in der der Fehler auftrat.

Siehe auch

[XMLReader](#), [RuntimeException](#) Klassen.

XMLStyleSheet Klasse

Diese wird für XML-Dokumente verwendet, die XSLT-Stylesheets repräsentieren, die einem anderen XML-Dokument mit der Transform-Methode der [XMLDocument](#)-Klasse zugewiesen werden.

Übergeordnete Klasse

[XMLDocument](#)

Konstruktoren

Name	Parameter	Beschreibung
XMLStyleSheet		Erzeugt ein leeres XSL-Dokument
XMLStyleSheet	XML als String	Liest einen XML-String in ein XSL-Dokument ein
XMLStyleSheet	f als FolderItem	Liest eine XML-Datei in ein XSL-Dokument ein.

Methoden

Name	Parameter	Beschreibung
LoadXML	XML als String oder f als FolderItem	Liest einen XML-String oder eine XML-Datei in ein XSL-Dokument ein.

XMLTextNode Klasse

Repräsentiert einen XML-Text-Knoten.

Übergeordnete Klasse

[XMLNode](#)

Eigenschaften

Keine

Methoden

Keine

Hinweis

Verwenden Sie die **XMLTextNode**-Klasse mit dem [IsA](#)-Operator, um festzustellen, ob es sich bei einem [XMLNode](#) um einen **XML-TextNode** handelt.

Siehe auch

[XMLNode](#), [XMLComment](#) Klassen, [IsA](#) Operator

XMLXslHandler Klasse

Wird zum Verwalten von SAX (Simple API for XML) Style-Events für die XML-Ausgabe einer XSLT-Transformation verwendet.

Events

Name	Parameter	Beschreibung
Characters	Inhalt als <u>String</u>	Wird bei Zeichen-Daten ausgelöst.
Comment	Inhalt als <u>String</u>	Wird bei XML-Kommentaren ausgelöst.
EndDocument		Wird am Ende des Dokuments ausgelöst.
EndElement	Name als <u>String</u>	Wird am Ende eines Elements ausgelöst.
EndNamespace	prefix als <u>String</u>	Wird am Ende einer Namensraum-Deklaration ausgelöst.
ProcessingInstruction	Target als <u>String</u> , Contents als <u>String</u>	Wird bei einer XML-Verarbeitungsanweisung ausgelöst.
StartDocument		Wird am Beginn eines Dokuments ausgelöst.
StartElementName	Name als <u>String</u> , attributeList als <u>XMLAttributeList</u>	Wird am Beginn eines Elements ausgelöst.
StartNamespace	prefix als <u>String</u> , URI als <u>String</u>	Wird am Beginn einer Namensraum-Deklaration ausgelöst.

Hinweise

Erzeugen Sie zum Verwenden von **XMLXslHandler** eine Instanz dieser Klasse und fügen Sie Code in die Events ein. Übergeben Sie dann eine Instanz Ihrer Unterklasse als zweiten Parameter der Transform-Methode der XMLDocument-Klasse.

Siehe auch

XMLDocument Klasse.

Anhang A: REALdatabase SQL Sprachreferenz

Ein REALbasic Datenbank-Frontend verwendet zur Kommunikation mit seiner Datenquelle die Strukturierte Abfragesprache SQL (Structured Query Language). Das Datenbank-Plugin, das Sie zum Zugriff auf Ihre Datenquelle verwenden, empfängt eine SQL-Anweisung von REALbasic, übersetzt diese – falls notwendig – in eine für Ihre Datenquelle verständliche Form und sendet diese an die Datenquelle.

Die mitgelieferte REALdatabase verwendet die in diesem Abschnitt beschriebene Untermenge des SQL-Standards. Im Fall der REALdatabase wird das Plugin als integraler Bestandteil von REALbasic mitgeliefert, muss also nicht als zusätzliche Datei im Plugins-Verzeichnis installiert werden.

REALdatabase unterstützt einen Teil des SQL/92 und SQL/99-Standards, einschließlich Abfragen mit Verknüpfungen, Aggregatfunktionen und mehr. Für die unterstützten Funktionen gilt, dass sie bis auf wenige unwesentliche Einschränkungen (wie z.B. beim Datentyp Boolean) voll SQL-kompatibel sind, einschließlich der Standard-SQL-Fehlercodes.

Jede REALdatabase-Tabelle verfügt über ein spezielles Feld „_rowID“, das einen Datensatz eindeutig identifiziert. Dieses Feld wird automatisch mitgeführt und aktualisiert und ist ideal zum Erzeugen von relationalen Datenbanken geeignet.

Fast alle von REALbasic unterstützten Datenbankserver verfügen über native SQL-Backends (die einzige Ausnahme ist 4DServer). Wenn Sie mit einem nativen SQL-Backend arbeiten, übergibt das REALbasic-Plugin Ihren SQL-Befehl unmodifiziert an die Datenquelle. Das heißt, dass Sie alle für diese Datenquelle gültigen SQL-Befehle in Ihrem REALbasic-Frontend verwenden können. Welche SQL-Kommandos und Datentypen Ihr Datenbankserver unterstützt, entnehmen Sie bitte dessen Dokumentation.

Falls SQL für Sie Neuland ist, sollten Sie sich zunächst mit den Grundlagen von SQL beschäftigen, bevor Sie ein REALbasic Datenbank-Frontend programmieren. Diese Anleitung kann kein Ersatz für ein SQL-Handbuch sein, sie beschreibt lediglich die SQL-Untermenge, die derzeit von der REALdatabase und dem 4th Dimension-Plugin unterstützt wird. Konsultieren Sie also eines der vielen guten SQL-Bücher, wie z.B. *SQL für Dummies* von Allen G. Taylor. Außerdem sind im Internet viele gute SQL-Online-Handbücher zu finden.

Unterstützte SQL Syntax

In diesem Abschnitt finden Sie eine semiformale Spezifikation der von REALbasic unterstützten SQL-Syntax. Dabei gelten folgende Konventionen: Ein vertikaler Strich („|“) ist Trennzeichen zwischen Begriffen, die alternativ verwendet werden können. Begriffe, die optional verwendet oder weggelassen werden können, werden in eckige Klammern („[“ und „]“) eingeschlossen. Begriffe, die angegeben werden müssen, sind in geschweifte Klammern („{“ und „}“) eingeschlossen. Begriffe, die zwischen „<“ und „>“ stehen, sind von komplexerer Natur und werden mit einem eigenen Eintrag detailliert beschrieben. Alle Begriffe, die auf „- string“ enden, sind einfache Bezeichner.

```
valid-sql-input:
{
  <select-statement>
  <insert-statement>
  <update-statement>
  <delete-statement>
  <create-table-statement>
  <create-index-statement>
  <alter-table-statement>
  <drop-table-statement>
  <drop-index-statement>
  COMMIT
  ROLLBACK
}
```

```
select-statement:
  SELECT [ ALL | DISTINCT ] <select-item-list>
    FROM <table-list>
    [ WHERE <expression> ]
    [ GROUP BY <column-ref-list> ]
    [ HAVING <expression> ]
    [ ORDER BY <expression> ]
```

```
select-item-list: one or more comma-separated items of the form:
  <column-expression> [ [ AS ] <column-name-string> ]
```

```
table-list: one or more comma-separated items of the form:
  <table-name-string> [ [ AS ] <range-variable-string> ]
```

```

insert-statement:
    INSERT INTO <table-name-string> [ ( <column-name-list> ) ]
    {
        | DEFAULT VALUES
        | VALUES ( <expression-list> )
        | <select-statement>
    }

update-statement:
    UPDATE <table-name-string> SET <assignment-list>
    [ WHERE <expression> ]

assignment-list: one or more comma-separated items of the form:
    <column-name-string> = { <expression> | DEFAULT | NULL }

delete-statement:
    DELETE FROM <table-name-string>
    [ WHERE <expression> ]

create-table-statement:
    CREATE TABLE <table-name-string> ( <column-def-list> )

column-def-list: one or more comma-separated items of the form:
    <column-name-string> <column-type-string>

create-index-statement:
    CREATE INDEX <index-name-string> ON <table-name-string>
    ( <column-name-list> )

alter-table-statement:
    ALTER TABLE <table-name-string>
    {
        | ADD [ COLUMN ] <column-name-string> <column-type-string>
        | DROP [ COLUMN ] <column-name-string> { RESTRICT | CASCADE }
    }

drop-table-statement:
    DROP TABLE <table-name-string> { RESTRICT | CASCADE }

drop-index-statement:
    DROP INDEX <table-name-string>.<index-name-string>

expression:
    {
        | <expression> OR <expression>
        | <expression> AND <expression>
        | NOT <expression>
        | <expression> { = | < | <= | > | >= } <expression>
        | <expression> [ NOT ] LIKE <expression> [ ESCAPE <expression> ]
        | <expression> { + | - } <expression>
        | <expression> { * | / } <expression>
        | -<expression>
        | ( <expression> )
        | <function>
        | { <number> | <string-literal> | TRUE | FALSE }
        | DATE <date-string>
        | TIME <time-string>
        | TIMESTAMP <timestamp-string>
        | <column-reference>
    }

column-reference:
    {
        | <column-name-string>
        | <range-variable-string>.<column-name-string>
        | *
        | <range-variable-string>.*
    }

function:
    {
        | BIT_LENGTH ( <expression> )
        | OCTET_LENGTH ( <expression> )
        | CHAR_LENGTH ( <expression> )
        | CURRENT_DATE
    }

```

```

CURRENT_TIME
CURRENT_DATETIME
LOWER ( <expression> )
UPPER ( <expression> )
POSITION ( <expression> IN <expression> )
SUBSTRING ( <expression> FROM <expression> [ FOR <expression> ] )
TRIM ( [ LEADING | TRAILING | BOTH ]
      [ <expression> ] [ FROM ] <expression> )
AVG ( [ DISTINCT | ALL ] <expression> )
MAX ( [ DISTINCT | ALL ] <expression> )
MIN ( [ DISTINCT | ALL ] <expression> )
SUM ( [ DISTINCT | ALL ] <expression> )
COUNT ( [ DISTINCT | ALL ] <expression> )
COUNT (*)
LAST_ROWID ( 'table-name-string' )
}

```

Unterstützte Datentypen

Die SQL-Untermenge der REALdatabase kann mit folgenden Datentypen umgehen:

Feldtyp	Numerischer Code	Beschreibung
Integer	3	Numerischer Ganzzahl-Datentyp. Die maximale Stellenanzahl ist implementierungsabhängig. Die REALdatabase verwendet 4-Byte-Integer-Werte (Wertebereich $\pm 2,000,000,000$). Für andere Datenquellen konsultieren Sie bitte deren Dokumentation.
Text oder VarChar	5	Speichert Textdaten, bei denen die Anzahl der Zeichen von Datensatz zu Datensatz variiert, ungenutzte Zeichen aber nicht mit Leerzeichen überschrieben werden. VARCHAR(20) legt also ein VARCHAR-Feld mit der Maximallänge von 20 Zeichen an. Die REAL-Database unterstützt den VarChar-Feldtyp, nicht den Text-Typ. Die theoretische Maximallänge eines VarChar-Feldes beträgt 2^{31} Bytes. Wenn Sie in einem CREATE TABLE oder ALTER TABLE-Befehl eine Maximallänge festlegen, wird diese ignoriert.
Double	7	Speichert Fließkommazahlen mit doppelter Genauigkeit. Die REAL-Database verwendet 8-Byte Doubles.
Date	8	Speichert Jahr, Monat und Tag eines Datums im Format YYYY-MM-TT.
Time	9	Speichert die Stunden, Minuten und Sekunden einer Uhrzeit im Format HH:MM:SS. Die Stunden und Minuten sind zweistellig. Die Sekunden sind auch zweistellig, können aber einen optionalen Nachkommateil haben, z.B. 09:55:25.248. Die Standardlänge des Nachkommateils ist 0.
TimeStamp	10	Speichert Datum und Uhrzeit im Format YYYY-MM-TT HH:MM:SS. Die Einzelkomponenten entsprechen den Datentypen Date und Time mit dem Unterschied, dass der Nachkommateil der Sekunden standardmäßig 6 und nicht 0 Stellen lang ist. Wenn ein TimeStamp-Wert keinen Nachkommaanteil hat, ist er 19 Stellen lang. Verfügt er über einen Nachkommaanteil, ist seine Länge 20 Stellen plus die Anzahl der Nachkommastellen.
Boolean	12	Speichert die Werte von TRUE und FALSE.
Binary	14	Speichert Code, Bilder und hexadezimale Daten. Schlagen Sie in der Dokumentation Ihrer Datenquelle nach, um die maximale Länge des Binary-Feldes zu erfahren. Die REALdatabase speichert bis zu 2^{31} Bytes.
String	18	Bis zu 2^{31} Bytes Text. Identisch mit VarChar.

Die REALdatabase unterstützt ein Fluchtsymbol („Escape Character“) in Zeichenketten. Ein Fluchtsymbol entwertet ein direkt darauf folgendes Sonderzeichen und sorgt dafür, dass dieses als normales Zeichen gewertet wird. Das Fluchtsymbol kann mittels SET ESCAPE frei definiert werden. Z.B. stellt „SET ESCAPE p“ den Backslash als Fluchtsymbol ein. Mit „SET ESCAPE“ ohne Argument wird das Fluchtsymbol wieder deaktiviert.

SQL in REALbasic

Dieser Abschnitt zeigt eine Übersicht der SQL-Befehle für die REALdatabase.

Befehl	Beschreibung
<u>ALTER TABLE</u>	Dient zum Hinzufügen und Löschen von Feldern aus einer Tabelle.
<u>COMMIT</u>	Schreibt Modifikationen in die Datenbank zurück, z.B. neu eingefügte, gelöschte oder geänderte Daten. Speichert auch mit CREATE TABLE oder CREATE INDEX vorgenommene Änderungen an der Datenbankstruktur.
<u>CREATE INDEX</u>	Erzeugt einen Index in einer Tabelle.
<u>CREATE TABLE</u>	Erzeugt eine Tabelle und spezifiziert deren Felder und Feldattribute.
<u>DELETE</u>	Löscht die angegebene Gruppe von Datensätzen.
<u>DROP TABLE</u>	Löscht eine Tabelle aus der Datenbank.
<u>DROP INDEX</u>	Löscht den angegebenen Index aus der Tabelle.
<u>INSERT</u>	Fügt einen Datensatz in die Tabelle ein.
<u>ROLLBACK</u>	Nimmt durch Einfügen, Löschen und Aktualisieren an den Daten vorgenommene Änderungen zurück. Das gilt auch für Änderungen an der Datenbankstruktur (CREATE TABLE, CREATE INDEX...).
<u>SELECT</u>	Liefert eine Gruppe von Datensätzen, in der SQL-Sprache auch Cursor genannt. REALbasic verwendet den Ausdruck RecordSet. Sie können die Spalten (Felder), Tabelle(n), Suchbedingungen, Gruppierung und Spaltensortierung festlegen.
<u>UPDATE</u>	Aktualisiert einen in der Tabelle vorhandenen Datensatz.

Funktionen für alphanumerische Zeichen

Folgende Funktionen können auf einzelne Zeichen oder Zeichenketten angewendet werden:

Funktion	Beispiel	Beschreibung
<u>BIT_LENGTH</u>	BIT_LENGTH('zz') ergibt 32	Liefert die Länge eines Zeichens oder einer Zeichenkette in Bits. BIT_LENGTH (<i>string</i>)
<u>CHAR_LENGTH</u>	CHAR_LENGTH ('480') ergibt 3.	Liefert die Anzahl der in einer Zeichenkette enthaltenen Zeichen zurück. CHAR_LENGTH (<i>string</i>)
<u>LOWER</u>	LOWER('GREENHORNET') ergibt "greenhornet".	Wandelt alle Zeichen einer Zeichenkette in Kleinbuchstaben um. LOWER (<i>string</i>)
<u>OCTET_LENGTH</u>	OCTET_LENGTH ('FF') ergibt 2.	Liefert die Größe einer Zeichenkette oder eines Bitstring-Ausdrucks in Byte: OCTET_LENGTH (<i>string</i>)
<u>POSITION</u>	POSITION ('G' IN 'Fred Gorman') ergibt 6.	Sucht eine Zeichenkette in einer anderen Zeichenkette und liefert die Position des ersten Zeichens der gefundenen Zeichenkette als Integer zurück. Wenn die gesuchte Zeichenkette nicht gefunden wurde, wird null zurückgeliefert. POSITION (<i>targetString</i> IN <i>sourceString</i>)
<u>SUBSTRING</u>	SUBSTRING ('Samantha' FROM 1 For 3) liefert 'Sam'.	Extrahiert einen Teilstring aus dem übergebenen Quellstring. SUBSTRING (<i>string</i> FROM <i>start</i> [FOR <i>length</i>]) <i>Start</i> ist die Startposition; wird die optionale FOR-Bedingung weggelassen, liefert SUBSTRING alle Zeichen von <i>start</i> bis zum Ende des Quellstrings. Wird FOR angegeben, liefert SUBSTRING <i>length</i> Zeichen, beginnend bei <i>start</i> zurück.
<u>TRIM</u>	TRIM BOTH from ' The Matrix ' liefert 'The Matrix'. Identisch mit TRIM BOTH '' from ' The Matrix '.	Entfernt Leer- oder andere spezifizierte Zeichen vom Anfang oder Ende der übergebenen Zeichenkette (Vorgabe ist das Leerzeichen). TRIM (LEADING TRAILING BOTH [<i>trimstring</i>] FROM <i>targetString</i>)
<u>UPPER</u>	UPPER('Grand Blanc') liefert "GRAND BLANC".	Liefert den übergebenen String in Großbuchstaben zurück. UPPER (<i>string</i>)

SQL Funktionen

Folgende Datums-, Zeit- und ID-Funktionen sind verfügbar:

Funktion	Beispiel	Beschreibung
<u>CURRENT_DATE</u>	CURRENT_DATE	Liefert das aktuelle Datum im SQL-Datumsformat YYYY-MM-DD als SQL Date-Datentyp zurück. CURRENT_DATE
<u>CURRENT_TIME</u>	CURRENT_TIME	Liefert die aktuelle Uhrzeit im SQL-Format HH:MM:SS als SQL Time-Datentyp zurück. CURRENT_TIME
<u>CURRENT_TIMESTAMP</u>	CURRENT_TIMESTAMP	Liefert den aktuellen Zeitstempel im SQL TimeStamp-Format YYYY-MM-DD HH:MM:SS als SQL TimeStamp Datentyp zurück.
<u>LAST_ROWID</u>	LAST_ROWID ('Movies')	Liefert den letzten Wert von _rowID für die angegebene Tabelle. Dieser Wert kann für verschiedene relationale Verknüpfungen benutzt werden. LAST_ROWID (tablename)

Aggregatfunktionen

Bei den folgenden Funktionen handelt es sich um Aggregatfunktionen, die auch als Mengenfunktionen bezeichnet werden. Aggregatfunktionen berechnen ein einziges Ergebnis aus einer Menge von Datensätzen.

Die Datensätze, auf die eine Aggregatfunktion angewendet werden soll, werden mit einer standardmäßigen WHERE-Bedingung festgelegt, die zusammen mit der Aggregatfunktion in einen SELECT-Befehl eingebettet ist.

In einem SELECT-Befehl können mehrere Aggregatfunktionen enthalten sein, jede liefert ein eigenes Ergebnis für die durch die WHERE-Bedingung festgelegten Datensätze. Allerdings darf das Schlüsselwort DISTINCT nur einmal vorkommen. Der SELECT-Befehl liefert ein RecordSet zurück. Wird die WHERE-Bedingung weggelassen, wird die Aggregatfunktion über alle Datensätze der Tabelle berechnet.

Wird die GROUP BY-Bedingung zusammen mit einer Aggregatfunktion verwendet, besteht das Ergebnis aus einem Datensatz pro Gruppe. Die Berechnung erfolgt dann pro Gruppe.

Ohne GROUP BY-Bedingung wird ein Datensatz mit dem Rechenergebnis über alle abgefragten Datensätze zurück geliefert.

Funktion	Beschreibung
<u>AVG</u>	Ermittelt über mehrere Datensätze den numerischen Durchschnittswert eines Ausdrucks. Der Ausdruck muss einen numerischen Datentyp besitzen und kann ein oder mehrere Felder enthalten. Mit dem Schlüsselwort DISTINCT kann die Berechnung so eingeschränkt werden, dass jeder Wert für Ausdruck nur einmal berücksichtigt wird. Beispiel: Ermittelt nur für Datenbank-Produkte den Durchschnittspreis in der Produkttabelle. SELECT AVG (Preis) from Produkte WHERE Produkt='Datenbank'
<u>COUNT</u>	Zählt die Datensätze, auf die der Ausdruck zutrifft. Der Ausdruck kann ein oder mehrere Felder enthalten. Mit dem Schlüsselwort DISTINCT kann die Berechnung so eingeschränkt werden, dass jeder Wert für Ausdruck nur einmal berücksichtigt wird. Beispiel: Zählt die Kundenzahl für die Postleitzahl 69123. SELECT COUNT (*) from Kunden WHERE PLZ='69123'
<u>MAX</u>	Gibt den größten Wert des oder der spezifizierten Felder zurück. Arbeitet mit numerischen, Datums-, Zeit- und Zeichen-Datentypen zusammen. Beispiel: Findet die größte Einwohnerzahl in der Städte-Tabelle. SELECT MAX (Einwohner) from Staedte
<u>MIN</u>	Gibt den kleinsten Wert des oder der spezifizierten Felder zurück. Arbeitet mit numerischen, Datums-, Zeit- und Zeichen-Datentypen zusammen. Beispiel: Findet den niedrigsten Preis in der Produkte-Tabelle. SELECT MIN (Preis) from Produkte

Funktion	Beschreibung
<u>SUM</u>	Ermittelt die Gesamtsumme der spezifizierten Felder. Deshalb muss der Ausdruck einen numerischen Datentyp besitzen. Der Ausdruck kann ein oder mehrere Felder beinhalten. Mit dem Schlüsselwort DISTINCT kann die Berechnung so eingeschränkt werden, dass jeder Wert für Ausdruck nur einmal berücksichtigt wird. Beispiel: Findet die Einwohnerzahl der Staaten der NE-Region. SELECT SUM (Einwohner) from Staaten WHERE Region='NE'

Ausführen von SQL-Kommandos

Der SELECT-Befehl wird als Zeichenkette an die SQLSelect-Methode der Database-Klasse übergeben. Diese liefert ein RecordSet-Objekt zurück, das die gesuchten Datensätze enthält.

Alle anderen SQL-Befehle werden über die SQLExecute-Methode der Database-Klasse aufgerufen. Keiner diese Befehle liefert Daten zurück.

Die Beispiele dieses Abschnitts setzen voraus, dass Sie eine Instanz der Database-Klasse erzeugt haben. Das SQL-Kommando wird in Form einer Zeichenkette entweder an SQLExecute oder SQLSelect übergeben.

Als Syntax wird im Folgenden ausschließlich die Syntax des SQL-Befehls angegeben. Der REALbasic-Code, der den SQL-Befehl an die Datenquelle weiter gibt, wird ausgeblendet.

Fehlercodes

Die REALdatabase liefert in den ErrorCode- und ErrorMessage-Eigenschaften der Database-Klasse einen Fehlercode und eine Fehlerbeschreibung.

Die ErrorMessage-Eigenschaft enthält nach der Fehlerbeschreibung in Klammern auch den Fehlercode.

Code	Beschreibung
00000	Erfolg. Dieser Code wird von einem erfolgreichen SQL-Befehl geliefert.
02000	Keine Datensätze gefunden. Wird zurückgegeben, wenn ein SELECT-Befehl keine Datensätze findet.
01004	Warnung: Zeichenkette rechts abgeschnitten.
01S09	Warnung: Ungültiges SQL-Schlüsselwort
07000	Fehler: Dynamischer SQL-Fehler.
42S01	Fehler: Die Tabelle existiert bereits.
42S02	Fehler: Tabelle nicht gefunden.
42S11	Fehler: Der Index existiert bereits.
42S12	Fehler: Index nicht gefunden.
42S21	Fehler: Feld (Spalte) existiert bereits.
42S22	Fehler: Feld (Spalte) nicht gefunden.
21S01	Fehler: Die Werteliste passt nicht zur Feldliste.
22000	Fehler: Datenausnahme.
22023	Fehler: Ungültiger Parameter.
25000	Fehler: Ungültiger Transaktionsstatus.
0A000	Fehler: Nicht unterstütztes SQL-Feature.
72000	Fehler: Fehler des internen SQL-Prozessors.

SQL Befehlsreferenz

Dieser Abschnitt ist eine Sprachreferenz für die von der REALdatabase unterstützte SQL-Untermenge.

ALTER TABLE Befehl

Fügt ein Feld in eine Tabelle ein oder löscht ein Feld aus einer Tabelle.

Syntax

ALTER TABLE *tablename* **ADD** [*Column*] *fieldname* *fieldtype*

oder

ALTER TABLE *tablename* **DROP** [*Column*] *fieldname* {**RESTRICT**|**CASCADE**}

Argument	Beschreibung
<i>TableName</i>	Name einer vorhandenen Tabelle.
<i>FieldName</i>	Name des neuen bzw. zu entfernenden Feldes.
<i>FieldType</i>	Datentyp. Von der REALdatabase werden folgende Datentypen unterstützt: Integer, VarChar, Double, Boolean, Date, Time, TimeStamp, Binary (blob) und String. Welche Datentypen von anderen Datenquellen unterstützt werden, entnehmen Sie bitte der Dokumentation der betreffenden Datenquelle.

Das Schlüsselwort „Column“ kann auch weggelassen werden.

Beispiel

Das Beispiel fügt das Feld „VacationDays“ in die Tabelle „Employees“ ein.

```
Dim dbFile as FolderItem
Dim db as RealDatabase
db=New RealDatabase
dbFile=GetFolderItem("myCompany")
db.DatabaseFile=dbFile
if db.connect then
  db.SQLExecute("ALTER TABLE Employees ADD COLUMN VacationDays Integer")
  if db.error then
    MsgBox db.errorMessage
  else
    db.SQLExecute("Commit") //no error, save change
  end if
else
  MsgBox "Connection to database failed."
end if
```

Siehe auch

[CREATE TABLE](#), [DROP TABLE](#)

COMMIT Statement

Speichert Änderungen in der Datenbank.

Syntax

COMMIT

Anmerkung

Die REALdatabase arbeitet mit Transaktionen. Eine Transaktion umfasst entweder Änderungen an der Datenbankstruktur oder an den gespeicherten Daten. Eine Transaktion wird automatisch mit der ersten derartigen Änderung gestartet und erst dann in der Datenbank gespeichert, wenn der COMMIT-Befehl ausgelöst wird. Wenn Sie statt COMMIT den ROLLBACK-Befehl auslösen, wird der Zustand der Datenbank vor Beginn der Transaktion wieder hergestellt.

Änderungen an der Datenbank können auch durch Aufrufen der Commit-Methode der Database-Klasse gespeichert werden.

Beispiel

Das folgende Beispiel verwendet Commit, um eine Änderung in einer Tabelle zu speichern. Einige andere Beispiele in diesem Kapitel verwenden alternativ die Commit-Methode der Database-Klasse.

```
Dim dbFile as FolderItem
Dim db as RealDatabase
db=New RealDatabase
dbFile=GetFolderItem("myCompany")
db.DatabaseFile=dbFile
if db.connect then
```

```
db.SQLExecute ("ALTER TABLE Employees ADD COLUMN VacationDays Integer")
if db.error then
  MsgBox db.errorMessage
else
  db.SQLExecute("Commit") //no error, save change
end if
else
  MsgBox "Connection to database failed."
end if
```

Siehe auch

[ROLLBACK](#)

CREATE INDEX Befehl

Erzeugt einen Index in einer Tabelle.

Syntax

CREATE INDEX *indexname* **ON** *tablename* (*fieldlist*)

Argument	Beschreibung
<i>IndexName</i>	Name des neuen Index.
<i>TableName</i>	Name der Tabelle, die das Feld enthält, das indiziert werden soll.
<i>Fieldlist</i>	Die Namen eines oder mehrerer Felder, aus denen der Index erzeugt werden soll. <i>FieldList</i> ist eine kommaseparierte Liste von Feldnamen/Datentypen.

Anmerkung

Indizieren Sie die Felder, die Sie am häufigsten für Such- und Sortiervorgänge benötigen und die Felder, die für relationale Verknüpfungen verwendet werden. Suchvorgänge können durch die Verwendung eines Index drastisch beschleunigt werden, da nicht mehr die gesamte Tabelle sequentiell durchsucht werden muss.

Beispiel

Das folgende Beispiel erzeugt in der Tabelle „Employees“ einen Index „NameIndex“ aus dem Feld „Name“.

```
Dim dbFile as FolderItem
Dim db as REALdatabase
db=New REALdatabase
dbFile = GetFolderItem("myCompany")
db.DatabaseFile=dbFile
If db.Connect then
  db.SQLExecute ("Create Index NameIndex on Employees (Name)")
  If db.error then
    MsgBox db.errorMessage
  Else
    db.SQLExecute ("Commit")
  end if
else
  MsgBox "Database connection failed."
End if
```

Siehe auch

[DROP INDEX](#)

CREATE TABLE Befehl

Erzeugt eine neue Tabelle in einer Datenbank.

Syntax

CREATE TABLE *tablename* (*fieldname1 fieldtype* [not null],...,*FieldnameN fieldtype* [not null])

Argument	Beschreibung
<i>TableName</i>	Name der neuen Tabelle.
<i>FieldName</i>	Name eines Feldes
<i>FieldType</i>	Datentyp. Die REALdatabase unterstützt folgende Datentypen: integer, varchar, double, boolean, date, time, timestamp, binary (blob) und string.
<i>Not Null</i>	Optional. Wenn <i>Not Null</i> verwendet wird, verlangt die Datenbank einen Wert für dieses Feld in jedem Datensatz, es darf also nicht leer bleiben.

Anmerkung

Der Befehl Create Table erzeugt eine Tabellenstruktur in der aktuellen Datenbank. Verwenden Sie die SQLExecute-Methode der Database-Klasse, um REALbasic einen Create Table-Befehl zu übergeben. Die REALdatabase ergänzt Ihre Angaben automatisch um das Primärschlüsselfeld „_RowID“. Somit müssen Sie keinen eigenen Primärschlüssel mitführen. Verwenden Sie statt dessen „_RowID“ für relationale Verknüpfungen und andere Aufgaben, bei denen ein Datensatz eindeutig identifiziert werden muss.

Beispiel

Der folgende Codeschnipsel öffnet die vorhandene Datenbank „myDVDs“ und fügt dieser eine Tabelle mit drei Feldern hinzu. Der Unterstrich („_“) wird verwendet, um den SQL-Befehl zur besseren Lesbarkeit auf zwei Zeilen zu verteilen. Der REALbasic-Compiler behandelt diese wie eine logische Zeile.

```
Dim dbFile as FolderItem
Dim rs as New Recordset
Dim db as REALdatabase
db=New REALdatabase
dbFile = GetFolderItem("myDVDs")
db.DatabaseFile=dbFile
If db.Connect() then //check if the database was opened successfully
    db.SQLExecute ("Create Table Movies (Title Varchar,Year Integer,"_
        +"Director Varchar)" // "_" used to break this into two lines
    if db.Error then
        MsgBox db.ErrorMessage
    else
        db.Commit
        MsgBox "Table created successfully.")
    end if
else //Connect method failed.
    Beep
    MsgBox "The database couldn't be opened."
end if
```

Siehe auch

[ALTER TABLE](#), [DROP TABLE](#).

DELETE Befehl

Löscht die mit der WHERE-Bedingung spezifizierten Datensätze.

Syntax

DELETE FROM *tablename* **WHERE** *searchcondition*

Argument	Beschreibung
<i>tablename</i>	Name der Tabelle, aus der Datensätze gelöscht werden sollen.
<i>searchcondition</i>	Boole'scher Ausdruck, der die zu löschenden Datensätze beschreibt.

Beispiel

Folgendes Beispiel löscht aus der Tabelle „Movies“ den Datensatz mit dem Inhalt „Star Wars“ im Titelfeld.

```
Dim dbFile as FolderItem
```

```
Dim db as RealDatabase
db=New RealDatabase
dbFile=New FolderItem("myMovies")
db.databaseFile=dbFile
If db.error then
    MsgBox db.errorMessage
else
    db.SQLExecute ("Delete From Movies Where Title='Star Wars'")
    If db.error then
        MsgBox db.ErrorMessage
    else
        db.Commit //save deletion
        MsgBox "Record deleted successfully."
    end if
end if
```

Siehe auch

[INSERT](#), [UPDATE](#).

DROP INDEX Befehl

Löscht einen Index aus einer Tabelle.

Syntax

DROP INDEX *tablename.indexname*

Argument	Beschreibung
<i>tablename</i>	Name der Tabelle, die den zu löschenden Index enthält.
<i>indexname</i>	Name des zu löschenden Index.

Beispiel

Folgendes Beispiel löscht den Index „NameIndex“ aus der Tabelle „Employees“, der für das Feld „Name“ angelegt worden ist.

```
Dim dbFile as FolderItem
Dim db as REALdatabase
db=New REALdatabase
dbFile = GetFolderItem("myCompany")
db.DatabaseFile=dbFile
If db.Connect then
    db.SQLExecute ("Drop Index Employees.NameIndex")
    If db.error then
        MsgBox db.errorMessage
    Else
        db.SQLExecute ("Commit")
    end if
else
    MsgBox "Database connection failed."
End if
```

Siehe auch

[CREATE INDEX](#)

DROP TABLE Befehl

Löscht eine Tabelle aus einer Datenbank.

Syntax

DROP TABLE *tablename* {**RESTRICT**| **CASCADE**}

Argument	Beschreibung
<i>tablename</i>	Name der Tabelle, die aus der Datenbank gelöscht werden soll.

Beispiel

Das folgende Beispiel löscht die Tabelle „Employees“ aus der Datenbank.

```
Dim dbFile as FolderItem
Dim db as RealDatabase
db=New RealDatabase
dbFile=GetFolderItem("myCompany")
db.DatabaseFile=dbFile
If db.connect then
    db.SQLExecute ("Drop Table Employees")
    If db.error then
        MsgBox db.errorMessage
    Else
        db.Commit //save changes to the schema
        MsgBox "Table dropped successfully."
    end if
Else
    MsgBox "Connection to database failed."
End if
```

Siehe auch

[CREATE TABLE](#), [ALTER TABLE](#).

INSERT Befehl

Fügt einen Datensatz in eine Tabelle ein.

Syntax

INSERT INTO *tablename* [*fieldname1*,...,*fieldnameN*]**VALUES** (*value1*,...,*valueN*)

Argument	Beschreibung
<i>tablename</i>	Name einer vorhandenen Tabelle.
<i>fieldname1 to N</i>	Optional. Feldnamen aus <i>tablename</i> . Wird keine Liste von Feldnamen angegeben, muss nacheinander für jedes Feld ein Wert angegeben werden. Wird eine Feldliste angegeben, korrespondiert jedes Wert der Werteliste mit dem entsprechenden Feld der Feldliste. So ist value1 der Wert für field1, value2 für field2 etc. Sie können auch dem <code>_rowID</code> -Feld einen Wert zuweisen. Dieser darf vorher allerdings noch nicht verwendet worden sein. Wird kein Wert für <code>_rowID</code> angegeben, wird dieser von der Datenbank-Engine erzeugt.
<i>value1 to N</i>	Werte für die Felder. Statt eines Wertes kann auch das Schlüsselwort „Default“ angegeben werden, um dem Feld einen Vorgabewert zuzuweisen.

Beispiel

Das folgende Beispiel fügt vier Datensätze in die Tabelle „Employees“ ein. Mit dem Unterstrich („_“) können lange logische Zeilen im Codeeditor auf mehrere Textzeilen verteilt werden, um die Lesbarkeit zu verbessern.

```
Dim dbFile as FolderItem
Dim db as RealDatabase
db=New RealDatabase
dbFile=New FolderItem("myCompany")
db.databaseFile=dbFile
If db.error then
    MsgBox db.errorMessage
else
    db.SQLExecute ("Insert into Employees (Name,Department,Salary,VacationDays)"_
        +"Values ('Seymore','Accounting',5000,0)")
    db.SQLExecute ("Insert into Employees (Name,Department,Salary,VacationDays)"_
        +"Values ('Throckmorton','Marketing',4400,0)")
    db.SQLExecute ("Insert into Employees (Name,Department,Salary,VacationDays)"_
        +"Values ('Dilbert','Engineering',6400,10)")
    db.SQLExecute ("Insert into Employees (Name,Department,Salary,VacationDays)"_
        +"Values ('Winnie','Sales',3500,0)")
    If db.error then
```

```
MsgBox db.errorMessage
Else
  db.Commit //save new records to database
  MsgBox "Records added successfully."
End if
rs=db.sqlSelect ("Select * from Employees") //select all Employee records
If db.error then
  MsgBox db.errorMessage
Else
  //proceed with database operations here
End if
End if
```

Siehe auch

[DELETE](#), [COMMIT](#), [ROLLBACK](#).

ROLLBACK Befehl

Stellt den Zustand der Datenbank vor Beginn der letzten Transaktion wieder her.

Syntax

ROLLBACK

Anmerkungen

Die REALdatabase arbeitet mit Transaktionen. Als Transaktion wird eine nicht trennbare Abfolge von Änderungen an der Datenbankstruktur (also an Tabellen, Feldern und Feldtypen) oder den Daten (mittels INSERT, DELETE oder UPDATE) bezeichnet. Eine Transaktion startet automatisch, wenn eine der genannten Aktionen ausgeführt wird und wird durch Aufrufen von COMMIT in der Datenbank gespeichert. Statt die Transaktion zu speichern (und damit zu beenden), kann sie mit ROLLBACK abgebrochen werden. Dies stellt den Zustand vor Start der Transaktion wieder her.

Sie können alternativ die Rollback-Methode der Database-Klasse aufrufen.

Siehe auch

[COMMIT](#).

SELECT Befehl

Der SQL Select-Befehl wird verwendet, um Abfragen an die Datenbank zu richten. Dazu wird ein gültiger SQL SELECT-Befehl als Zeichenkette an die SQLSelect-Methode der Database-Klasse übergeben. Der SQL-Select-Befehl selektiert, sortiert und gruppiert wahlweise Daten in einer oder mehreren Tabellen.

Syntax

result=SELECT [ALL | DISTINCT] *expressionList* FROM *tablelist* WHERE [LIKE] *searchcondition* GROUP BY *groupingcondition* HAVING *filtercondition* ORDER BY *sortCondition*

Teil	Beschreibung
result	Die vom SQL SELECT-Befehl ausgewählten Datensätze werden als RecordSet-Objekt zurückgeliefert. Der SQL SELECT-Befehl muss an die SQLSelect-Methode der Database-Klasse und darf nicht an SQLExecute übergeben werden.
ALL DISTINCT	Optional. Während ALL alle gefundenen Datensätze zurückliefert (Vorgabe), liefert DISTINCT nur solche Datensätze, die sich von allen vorher gelieferten unterscheiden, blendet also Dupletten aus. So liefert DISTINCT Cities nur einen Datensatz pro Stadt, auch wenn diese in mehreren Datensätzen vorkommt.

Teil	Beschreibung														
<i>expressionList</i>	Durch Kommata getrennte Liste von Spalten (Feldern). Verwenden Sie den *, um alle Felder aus <i>table</i> zu erhalten. Falls <i>tablelist</i> mehrere Tabellen enthält, müssen Sie die Syntax <i>tablename.fieldname</i> verwenden, um ein bestimmtes Feld zu referenzieren. Beispiel: Die Datenbank besteht aus der Tabelle Schauspieler mit den Feldern Name und Rolle und der Tabelle Filme mit den Feldern Regisseur, Titel und Jahr. Dann referenzieren Sie den Namen eines Schauspielers mit <i>Schauspieler.Name</i> und den Titel eines Films mit <i>Filme.Titel</i>. Wenn Sie Abfragen über mehrere Tabellen ausführen, werden Aliase für Tabellennamen akzeptiert, z.B.: „SELECT A.NAME, B.NAME FROM SCHAUSPIELER A, SCHAUSPIELER B WHERE A.FILM=B.FILM“. Die <i>expressionList</i> kann sowohl eine einfache Liste von Feldnamen sein, als auch Berechnungen mit mathematischen Operatoren oder SQL- und Aggregatfunktionen enthalten.														
<i>tablelist</i>	Durch Kommata getrennte Liste von Tabellennamen. Falls ein Tabellename Leerzeichen enthält, muss er in eckige Klammern eingeschlossen werden, z.B. [Product Groups].														
LIKE	Optionales Schlüsselwort, mit dem Zeichenketten auf teilweise Übereinstimmung geprüft werden können. Für den Teil, der sich unterscheiden darf, verwenden Sie im Suchbegriff die SQL-Platzhalter „%“ (für eine beliebige Zeichenfolge) und „_“ für ein beliebiges Zeichen. Beispiel: „WHERE FILME LIKE 'LIEBE%'“ passt auf alle Filme, die mit „Liebe“ beginnen. „WHERE Name LIKE '%son%'“ findet alle Namen, die die Zeichenfolge „son“ enthalten. Mehr dazu im Anschluss. Indem Sie dem Schlüsselwort LIKE den NOT-Operator voran stellen, negieren Sie den Suchbegriff. So findet „TITEL NOT LIKE 'Assistent%'“ nur Datensätze, deren Feld TITEL nicht mit „Assistent“ beginnt.														
<i>searchcondition</i>	Boole'scher Ausdruck, der eine Untermenge von Datensätzen in einer oder mehreren Tabellen beschreibt. Wenn <i>tablename</i> auf mehrere Tabellen verweist, müssen Sie die Syntax <i>tablename.fieldname</i> in <i>searchcondition</i> verwenden, z.B.: <i>Kunden.KuNr</i> = <i>Rechnungen.KuNr</i>. <i>Suchbedingung</i> kann folgende Vergleichssymbole verwenden: <table> <tr> <th>Vergleich</th><th>Symbol</th></tr> <tr> <td>Gleichheit</td><td>=</td></tr> <tr> <td>Nicht gleich</td><td><></td></tr> <tr> <td>Kleiner als</td><td><</td></tr> <tr> <td>Kleiner oder gleich</td><td><=</td></tr> <tr> <td>Größer als</td><td>></td></tr> <tr> <td>Größer oder gleich</td><td>>=</td></tr> </table>	Vergleich	Symbol	Gleichheit	=	Nicht gleich	<>	Kleiner als	<	Kleiner oder gleich	<=	Größer als	>	Größer oder gleich	>=
Vergleich	Symbol														
Gleichheit	=														
Nicht gleich	<>														
Kleiner als	<														
Kleiner oder gleich	<=														
Größer als	>														
Größer oder gleich	>=														
<i>groupingcondition</i>	Feld oder Felder, nach denen die Datensätze im RecordSet gruppiert werden. Wenn eine GROUP BY-Bedingung verwendet wird, werden die zurückgelieferten Datensätze entsprechend dieser Bedingung in Gruppen organisiert. Die Gruppen erscheinen in alphabetischer Reihenfolge.														
<i>filtercondition</i>	Einer GROUP BY-Bedingung kann optional eine HAVING-Bedingung folgen. Diese sorgt dafür, dass nur spezifische Werte für die GROUP BY-Felder berücksichtigt werden. Die <i>filtercondition</i> „filtert“ also die von der GROUP BY-Bedingung gruppierten Datensätze. Das Kommando „GROUP By Movies.Director HAVING Movies.Director='Kubrick' OR Movies.Director='Fellini' OR Movies.Director='Wood'“ gruppiert die Datensätze nach dem Regisseur, liefert aber nur drei Gruppen für Kubrick, Fellini und Wood. Die HAVING-Bedingung muss den Wert eines gruppierten Feldes mit einem konstanten Wert vergleichen.														
<i>sortcondition</i>	Feld oder Felder in denen Sie die individuellen Datensätze im RecordSet sortieren möchten. Verwenden Sie ORDER BY anstelle von GROUP BY, wenn das Feld unüblich zum Definieren von Gruppen ist, so wie z.B. Nachname. Normalerweise erscheinen die Datensätze in aufsteigender Reihenfolge. Wenn Sie es in absteigender wünschen, fügen Sie den Befehl DESC ein, also z.B. 'ORDER BY Rechnungen.Datum DESC'.														

Bemerkungen

Über den SQL Select-Befehl sind ganze Bücher geschrieben worden, speziell über das Thema relationale Datenbankoperationen.

Suchen mit Wildcards

Die REAL-Database unterstützt die beiden SQL-Wildcard-Zeichen, Prozent „%“ und Unterstrich „_“. Das Prozentzeichen steht für eine beliebige Anzahl von Zeichen und der Unterstrich steht für genau ein einzelnes beliebiges Zeichen. Wenn Sie das % an den Anfang oder das Ende des Suchstrings anfügen, lösen Sie eine „beginnt mit“ bzw. „endet mit“ Suche aus. Wenn Sie das % auf beiden Seiten des Strings angeben, legen Sie eine „enthält“ Suche fest. Hier ein Beispiel:

SELECT Befehl	Suchtyp
SELECT * from Autoren where au_lname LIKE "Jon%"	Beginnt mit
SELECT * from Autoren where au_lname LIKE "%son"	Endet mit
SELECT * from Autoren where au_lname LIKE "%man%"	Enthält

Das Suchkriterium ist case-sensitiv.
Der Unterstrich ist das Wildcard-Zeichen für ein beliebiges einzelnes Zeichen. Zum Beispiel liefert
SELECT * from Autoren where au_lname LIKE "B_r"
Liefert „Bar“, „Bor“, etc.

SELECT Beispiele

Relationale Operationen („Joins“ oder Verknüpfungen) werden ausgeführt, indem man die zu verknüpfenden Tabellen in der Tabellenliste aufführt und in der WHERE-Bedingung festlegt, über welche (Schlüssel-) Felder die Verknüpfung erfolgt. Für detaillierte Informationen über relationale Operationen konsultieren Sie bitte ein SQL-Handbuch.

Betrachten Sie die beiden folgenden Tabellen, Movies und Actors:

RowID	Title	Director	Year
1	Star Wars	George Lucas	1977
2	Raiders of the Lost Ark	George Lucas	1981
3	American Graffiti	George Lucas	1973
4	Apollo 13	Ron Howard	1995
5	Cast Away	Robert Zemeckis	2000

RowID	Name	Movie
1	Harrison Ford	1
2	Mark Hamill	1
3	Carrie Fisher	1
4	Harrison Ford	2
5	Ron Howard	3
6	Richard Dreyfuss	3
7	Tom Hanks	4
8	Tom Hanks	5

Die dritte Spalte der Actors-Tabelle identifiziert einen Film, in dem der Schauspieler mitwirkte. Die folgenden SELECT Befehle können auf diese Daten angewendet werden.

Einfache Selektion

Diese Anweisungen wählen alle Datensätze und alle Spalten einer Tabelle aus:

```
Select * from Movies
Select * from Actors
```

Diese Anweisungen liefern die oben abgebildeten Tabellen zurück.

Die folgende Anweisung verwendet „distinct“, damit jeder Schauspieler nur genau einmal zurückgeliefert wird:

```
Select distinct Name from Actors
```

Sie liefert folgendes RecordSet:

Name

Harrison Ford
Mark Hamill
Carrie Fisher
Ron Howard
Richard Dreyfuss
Tom Hanks

Relationale Operationen

Es folgen Beispiele für Abfragen, die WHERE und GROUP verwenden, um Daten aus beiden Tabellen zu extrahieren.

Schauspieler finden, die in „Star Wars“ mitgewirkt haben:

Für jeden Tabellennamen wird ein Buchstabe als Alias eingesetzt. Es werden diejenigen Datensätze zurückgeliefert, auf die die WHERE-Bedingung zutrifft. Diese legt fest, dass das Feld Titel den Inhalt „Star Wars“ haben muss und dass genau die Einträge aus Actors geliefert werden, deren Movie-Feld identisch ist mit der RowID des Star Wars-Datensatzes aus der Movies-Tabelle. Es werden also die ersten drei Datensätze aus Actors zurück geliefert.

```
Select a.name from Actors A, Movies M where a.movie=m._rowid and m.title='Star Wars'
```

Dieser Select-Befehl legt außerdem fest, dass nur der Darstellernamen geliefert wird. Das Ergebnis ist also:

Name

Harrison Ford
Mark Hamill
Carrie Fisher

Filme finden, in denen Lucas Regie führte und die Schauspieler, den Film und das Entstehungsjahr zurückliefern:

Dieses Beispiel unterscheidet sich von dem vorhergehenden darin, dass die zurück gelieferten Datensätze sowohl aus der Actors- als auch der Movies-Tabelle stammen:

```
Select A.Name,M.Title,M.Year from Actors A, Movies M where A.movie=M._rowid and M.Director='George Lucas'
```

Die WHERE-Bedingung arbeitet wie im vorigen Beispiel. Sie selektiert alle Schauspieler, deren Movies-Spalte mit der RowID eines Films übereinstimmt, bei dem Lucas Regie führte. Das Resultat der Abfrage:

Name	Movie	Year
Ron Howard	American Graffiti	1973
Richard Dreyfuss	American Graffiti	1973
Harrison Ford	Raiders of the Lost Ark	1981
Harrison Ford	Star Wars	1977
Mark Hamill	Star Wars	1977
Carrie Fisher	Star Wars	1977

Schauspieler finden, die zusammen mit Harrison Ford vor der Kamera standen:

```
Select A.name from Actors A, Actors B where A.movie = B.movie and B.name='Harrison Ford' and A.name<>'Harrison Ford'
```

Ergebnis:

Name

Mark Hamill
Carrie Fisher

Personen finden, die sowohl Schauspieler als auch Regisseur sind:

```
Select A.Name from Actors A, Movies M where A.Name = M.Director
```

Ergebnis :

Name
Ron Howard

Mit Wildcards suchen

Filme finden, deren Titel das Wort „War“ enthalten (Groß-/Kleinschreibung ignorieren):

```
Select title, year from Movies where Upper(title) like '%WAR%'
```

Ergebnis:

Title	Year
Star Wars	1977

Aggregatfunktionen

Für jeden Film ermitteln, wie viele Schauspieler mitwirken:

```
Select A.Title, Count(B.Name) from Movies A, Actors B where B.Movie = A._rowID group by B.Movie
```

Ergebnis:

Title	Count(Name)
Star Wars	3
Raiders of the Lost Ark	1
American Graffiti	2
Apollo 13	1
Cast Away	1

Für jeden Regisseur den ersten und letzten in der Datenbank erfassten Film ermitteln:

```
Select Director, Min(Year), Max(Year) from Movies Group by Director
```

Ergebnis:

Director	Min(Year)	Max(Year)
George Lucas	1973	1981
Robert Zemeckis	2000	2000
Ron Howard	1995	1995

Alle Schauspieler heraussuchen, die mehr als einen Film gedreht haben:

```
Select Name, Count(*) from Actors group by Name having Count(*) > 1
```

Ergebnis:

Name	Count
Harrison Ford	2
Tom Hanks	2

UPDATE Befehl

Modifiziert die in einer Tabelle gespeicherten Daten.

Syntax

```
UPDATE tablename SET fieldname1=expression1,... fieldnameN=expressionNWHERE searchcondition
```

Parameter	Beschreibung
tablename	Dame der Tabelle, die aktualisiert werden soll.

Parameter	Beschreibung
<i>fieldname</i>	Feld aus <i>TableName</i>
<i>expression</i>	Wert, der <i>FieldName</i> zugewiesen werden soll. Dabei kann es sich um einen Wert oder eines der Schlüsselworte DEFAULT oder NULL handeln.
<i>searchcondition</i>	Boole'scher Ausdruck, der die zu aktualisierende(n) Spalte(n) beschreibt.

Bemerkung

Rufen Sie den Commit-Befehl oder die Commit-Methode der Database-Klasse auf, um die die mit UPDATE vorgenommenen Änderungen in der Datenbank zu speichern. Oder verwenden Sie Rollback, um den Zustand vor Aufruf von UPDATE wieder herzustellen. Sie können auch die Edit- und Update-Methoden der RecordSet-Klasse von REALbasic verwenden, um Datensätze zu ändern.

Beispiel

Folgendes Beispiel ändert für den Film „Rabid“ das Feld Year auf 1976.

```
Dim dbFile as FolderItem
Dim db as RealDatabase
db=New RealDatabase
dbFile=New FolderItem("myMovies")
db.databaseFile=dbFile
If db.error then
    MsgBox db.errorMessage
else
    db.sqlexecute ("UPDATE Movies SET Year=1976 WHERE title='Rabid'")
    If db.error then
        MsgBox db.errorMessage
    Else
        db.Commit
        MsgBox "Records updated successfully."
    End if
End if
```

SQL Funktionen

BIT_LENGTH Function

Ermittelt die Anzahl von Bits, die ein Zeichen bzw. ein Bit-String belegt.

Syntax

result=BIT_LENGTH(String)

Teil	Typ	Beschreibung
<i>result</i>	Integer	Anzahl von Bits in <i>String</i> .
<i>String</i>	String	Zeichen oder Bit-String.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect ("SELECT BIT_LENGTH('zz')") //returns 16
rs=db.SQLSelect ("SELECTBIT_LENGTH('cat')") /returns 24
```

CHAR_LENGTH Function

Ermittelt die Anzahl von Zeichen in einer Zeichenkette.

Syntax

result=CHAR_LENGTH(*String*)

Teil	Typ	Beschreibung
<i>result</i>	Integer	Zeichenzahl in <i>String</i> .
<i>String</i>	String	Zu prüfende Zeichenkette.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect ("SELECT CHAR_LENGTH('Madison Ave.')" returns 12.
rs=db.SQLSelect ("SELECT CHAR_LENGTH('480' )" //returns 3.
```

Siehe auch

[BIT_LENGTH](#), [OCTET_LENGTH](#).

LOWER Funktion

Wandelt die übergebene Zeichenkette in Kleinbuchstaben und liefert diese zurück.

Syntax

result=LOWER(*String*)

Teil	Typ	Beschreibung
<i>result</i>	String	Die Zeichenkette in Kleinbuchstaben.
<i>String</i>	String	Zeichenkette, die umgewandelt werden soll.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect ("SELECT LOWER('GREENHORNET')")
//returns "greenhornet".
```

Siehe auch

[UPPER](#).

OCTET_LENGTH Funktion

Liefert die von einer Zeichenkette oder einem Zeichenkettenausdruck belegte Byte-Anzahl zurück.

Syntax

result=OCTET_LENGTH(*string*)

Teil	Typ	Beschreibung
<i>result</i>	Integer	Byte-Anzahl von <i>String</i> .
<i>String</i>	String	Zeichenkette, deren Byte-Anzahl ermittelt werden soll.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
.
rs=db.SQLSelect ("SELECT OCTET_LENGTH ('F')) //returns 1.
rs=db.SQLSelect ("SELECT OCTET_LENGTH ('FF')) //returns 2.
```

Siehe auch

[BIT_LENGTH](#), [CHAR_LENGTH](#).

POSITION Funktion

Sucht im Quellstring nach dem Zielstring und liefert dessen Startposition im Quellstring zurück.

Syntax

result=POSITION (*targetString* IN *sourceString*)

Teil	Typ	Beschreibung
<i>result</i>	Integer	Zeichenposition, an der <i>targetString</i> erstmals innerhalb von <i>sourceString</i> auftritt. Wenn <i>targetString</i> gar nicht in <i>sourceString</i> enthalten ist, wird null zurück geliefert.
<i>targetString</i>	String	Zeichenkette, nach der gesucht wird.
<i>sourceString</i>	String	Zeichenkette, die durchsucht wird.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
'
rs=db.SQLSelect ("SELECT POSITION ('G' IN 'Fred Gorman')") //returns 6.
```

Siehe auch

[SUBSTRING](#), [TRIM](#).

SUBSTRING Funktion

Liefert einen Teil einer Zeichenkette zurück.

Syntax

result=SUBSTRING (*targetString* FROM *start* [FOR *length*])

Teil	Typ	Beschreibung
<i>result</i>	String	Die Zeichenkette, die aus <i>targetString</i> extrahiert wurde.
<i>targetString</i>	String	Zeichenkette, aus der ein Teil extrahiert werden soll.
<i>Start</i>	Integer	Startposition innerhalb von <i>targetString</i> , ab der ein Teil extrahiert werden soll.
<i>Length</i>	Integer	Optionaler Parameter, der die Anzahl der ab <i>Start</i> zu extrahierenden Zeichen festlegt. Fehlt <i>Length</i> , werden alle Zeichen von <i>Start</i> bis zum Ende von <i>targetString</i> zurück geliefert.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
'
rs=db.SQLSelect ("SELECT SUBSTRING('Samantha' FROM 1 For 3)")
//returns Sam'.
```

Siehe auch

[POSITION](#), [TRIM](#).

TRIM Funktion

Entfernt überflüssige Zeichen vom Anfang und/oder Ende einer Zeichenkette.

Syntax

result=TRIM (LEADING|TRAILING|BOTH [*trimstring*] FROM *targetString*)

Teil	Typ	Beschreibung
<i>result</i>	String	Die von überflüssigen Zeichen am Anfang und Ende von <i>targetString</i> befreite Zeichenkette.
<i>trimString</i>	String	Optional. Zeichen, das am Anfang und/oder Ende von <i>targetString</i> entfernt werden soll. Falls nicht angegeben, wird das Leerzeichen angenommen.
<i>targetString</i>	String	Zeichenkette, die „getrimmt“ werden soll.

Bemerkungen

Sie können festlegen, ob führende und/oder abschließende Zeichen abgeschnitten werden sollen. Das Schlüsselwort FROM ist erforderlich, wenn ein *trimString* oder eine der Optionen LEADING | TRAILING | BOTH angegeben wurde.

Beispiele

```
Dim db as New REALdatabase
Dim rs as New RecordSet

rs=db.SQLSelect(TRIM (BOTH from ' The Matrix ')) //returns 'The Matrix'.
rs=db.SQLSelect(TRIM (Leading ' ' from ' Matrix')) //areturns 'Matrix'.
```

Siehe auch

[POSITION](#), [SUBSTRING](#).

UPPER Funktion

Liefert den übergebenen String in Großbuchstaben zurück.

Syntax

result=UPPER (*String*)

Teil	Typ	Beschreibung
<i>result</i>	String	Der String in Großbuchstaben.
<i>String</i>	String	Zeichenkette, die verarbeitet werden soll.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet

rs=db.SQLSelect ("SELECT UPPER('Grand Blanc')")
//returns "GRAND BLANC".
```

Siehe auch

[LOWER](#).

CURRENT_DATE Funktion

Liefert das aktuelle Datum im SQL-Format YYYY-MM-DD.

Syntax

result=CURRENT_DATE

Teil	Typ	Beschreibung
<i>result</i>	SQLDate	Das aktuelle Datum im Format YYYY-MM-DD.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
'
rs=db.SQLSelect ("SELECT CURRENT_DATE") //returns '2003-10-07'
```

CURRENT_TIME Funktion

Liefert die aktuelle Uhrzeit im SQL-Zeitformat: HH:MM:SS.

Syntax

result=CURRENT_TIME

Teil	Typ	Beschreibung
<i>result</i>	SQLTime	Die aktuelle Uhrzeit im Format HH:MM:SS.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
'
rs=db.SQLSelect ("SELECT CURRENT_TIME") //returns '13:51:30', for example
```

Siehe auch

[CURRENT_TIMESTAMP](#)

CURRENT_TIMESTAMP Function

Liefert die Kombination aus dem aktuellen Datum und der aktuellen Zeit im SQLTimestamp-Format YYYY-MM-DD HH:MM:SS.

Syntax

result=CURRENT_TIMESTAMP

Teil	Typ	Beschreibung
<i>result</i>	SQLTimestamp	Der aktuelle Zeitstempel im Format YYYY-MM-DD HH:MM:SS.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
'
rs=db.SQLSelect ("SELECT CURRENT_TIMESTAMP")
//returns '2003-10-07 13:52:57' for example
```

Siehe auch

[CURRENT_TIME](#)

LAST_ROWID Funktion

Liefert den letzten Wert von `_rowID` für die angegebene Tabelle. Dieser Wert kann für diverse relationale Verknüpfungen genutzt werden. Die REALdatabase fügt jeder Tabelle automatisch ein Feld `_rowID` zur eindeutigen Identifizierung jedes Datensatzes hinzu. `row_ID` wird automatisch mit einem Wert belegt, es sei denn, der Anwender weist einen Wert zu.

Syntax

result=LAST_ROWID (*tablename*)

Teil	Typ	Beschreibung
<i>result</i>	Integer	Der letzte Wert des <i>_rowID</i> -Feldes, der in der angegebenen Tabelle gespeichert wurde.
<i>tablename</i>	String	Der Name der Tabelle, für die der letzte vergebene <i>_rowID</i> -Wert ermittelt werden soll.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect ("SELECT LAST_ROWID ('Employees')")
```

Aggregatfunktionen

AVG Funktion

Ermittelt über mehrere Datensätze den Durchschnittswert eines Ausdrucks.

Syntax

result=AVG [DISTINCT|ALL] (*expression*)

Teil	Typ	Beschreibung
<i>result</i>	Double	Der Durchschnittswert von <i>expression</i> für die durch <i>searchExpression</i> spezifizierten Datensätze.
<i>expression</i>	Numeric	Feld oder Funktion von einem oder mehreren Feldern, die einen numerischen Wert zurückliefert.
SearchExpression		SQL WHERE Bedingung, die die zu verarbeitenden Datensätze auswählt.

Mit dem optionalen Schlüsselwort DISTINCT sorgen sie dafür, dass jeder für *expression* ermittelte Wert nur einmal für die Berechnung des Durchschnitts verwendet wird. Wenn Sie ALL verwenden, werden alle Werte für Expression einbezogen, also auch Duplikate. Vorgabe ist ALL.

Beispiel

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect("Select AVG (Population) FROM Cities")
rs=db.SQLSelect("Select AVG (Income) FROM Cities WHERE State='CA'")
```

Siehe auch

[COUNT](#), [SUM](#).

COUNT Funktion

Die COUNT-Funktion ermittelt, wie viele Datensätze einer Tabelle eine bestimmte Bedingung erfüllen.

Syntax

result=COUNT [DISTINCT] *expression* FROM *searchExpression*

Teil	Typ	Beschreibung
<i>result</i>	Integer	Die Summe von <i>expression</i> über die durch <i>SearchExpression</i> spezifizierten Datensätze.
<i>expression</i>	Any	Feld oder Funktion über mehrere Felder.
SearchExpression		SQL WHERE-Bedingung, die die Datensätze für die COUNT-Funktion selektiert.

Mit dem optionalen Schlüsselwort **DISTINCT** sorgen sie dafür, dass jeder für *expression* ermittelte Wert nur einmal gezählt wird. Wenn Sie * als *expression* verwenden, werden alle Datensätze gezählt.

Wenn *expression* angegeben ist, liefert COUNT die Anzahl der Datensätze, auf deren Felder *expression* zutrifft.

Beispiel

Das erste Beispiel liefert die Anzahl der Datensätze zurück, in denen sowohl im Title- als auch im Director-Feld der Movies-Tabelle Einträge vorgenommen wurden.

Das zweite Beispiel zählt die Datensätze, deren Title-Feld nicht leer und deren Director-Feld identisch mit „Lucas“ ist.

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect ("SELECT COUNT (DISTINCT Director) from Movies")
rs=db.SQLSelect ("Select Count (*) from Movies WHERE Director = 'Lucas'")
rs=db.SQLSelect ("Select Count (*) from Movies")
```

Siehe auch

[AVG](#), [SUM](#).

MAX Funktion

Liefert den größten Wert eines Ausdrucks zurück.

Syntax

result=MAX *expression* FROM *searchExpression*

Teil	Typ	Beschreibung
<i>result</i>	Double	Der größte Wert von <i>expression</i> für die mit <i>SearchExpression</i> spezifizierten Datensätze.
<i>expression</i>	Numeric or String	Feld oder Funktion über ein oder mehrere Felder, die einen numerischen Wert oder eine Zeichenkette zurückgibt.
<i>SearchExpression</i>		SQL WHERE-Bedingung, die die zu verarbeitenden Datensätze selektiert.

Beispiel

Dieses Beispiel finden die größte Einwohnerzahl in der Tabelle Stadt.

```
Dim db as New REALdatabase
Dim rs as New RecordSet
rs=db.SQLSelect("SELECT MAX (Einwohner) from Stadt")
```

Siehe auch

[MIN](#).

MIN Funktion

Liefert den kleinsten Wert eines Ausdrucks zurück.

Syntax

result=MIN *expression* FROM *searchExpression*

Teil	Typ	Beschreibung
<i>result</i>	Double	Der kleinste Wert von <i>expression</i> für die mit <i>SearchExpression</i> spezifizierten Datensätze.
<i>expression</i>	Numeric or String	Feld oder Funktion über ein oder mehrere Felder, die einen numerischen Wert oder eine Zeichenkette zurückgibt.
<i>SearchExpression</i>		SQL WHERE-Bedingung, die die zu verarbeitenden Datensätze selektiert.

Beispiel

Dieses Beispiel findet den kleinsten Preis in der Produktetabelle.

```
Dim db as New REALdatabase
Dim rs as New RecordSet
.
rs=db.SQLSelect("SELECT MIN (Preis) from Produkte")
```

Siehe auch

[MAX](#).

SUM Funktion

Liefert die Summe der angegebenen numerischen Felder zurück.

Syntax

result=SUM [DISTINCT|ALL] *expression* FROM *searchExpression*

Teil	Typ	Beschreibung
<i>result</i>	Double	Die Summe aller Werte von <i>expression</i> über die durch <i>SearchExpression</i> bestimmten Datensätze.
<i>expression</i>	Numeric	Feld oder Funktion über mehrere Felder, die einen numerischen Wert liefert.
<i>SearchExpression</i>		SQL WHERE-Bedingung, die die zu verarbeitenden Datensätze selektiert.

Mit dem optionalen Schlüsselwort DISTINCT sorgen sie dafür, dass jeder für *expression* ermittelte Wert nur einmal für die Berechnung verwendet wird. Wenn Sie ALL verwenden, werden alle Werte von Expression einbezogen, also auch Duplikate. Vorgabe ist ALL.

Bemerkung

Expression muss einen numerischen Wert ergeben und kann ein oder mehrere Felder enthalten. Das Schlüsselwort DISTICT kann die Berechnung auf solche Werte beschränken, die eindeutig sind.

Beispiel

Dieses Beispiel findet die Einwohnerzahl aller Bundesstaten in der Region NE.

```
Dim db as New REALdatabase
Dim rs as New RecordSet
.
rs=db.SQLSelect("SELECT SUM (Einwohner) from States WHERE Region='NE'")
```

Siehe auch

[AVG](#), [COUNT](#).

Anhang B: Text Encoding Base- und Variant-Codes

Die Tabellen in diesem Anhang listen die möglichen Werte für die *Base*- und *Variant*-Parameter auf, die von verschiedenen Funktionen aus dem Bereich Textcodierung verwendet werden. Tabelle 1 enthält die Codes für den *Base*-Parameter, Tabelle 2 die Codes für den *Variant*-Parameter.

Tabelle 1: Text Encoding Base

Base	Wert	Base	Wert	Base	Wert
MacRoman	0	MacVT100	&hFC	WindowsHebrew	&h0505
MacJapanese	1	MacHFS	&hFF	WindowsArabic	&h0506
MacChineseTrad	2	UnicodeDefault	&h0100	WindowsBalticRim	&h0507
MacKorean	3	UnicodeV1_1	&h0101	WindowsVietnamese	&h0508
MacArabic	4	ISO10646_1933	&h0101	WindowsKoreanJohab	&h0510
MacHebrew	5	UnicodeV2_0	&h0103	US_ASCII	&h0600
MacGreek	6	Unicodev2_1	&h0103	JIS_X0201_76	&h0620
MacCyrillic	7	ISOLatin1	&h0201	JIS_X0208_83	&h0621
MacDevanagari	9	ISOLatin2	&h0202	JIS_X0208_90	&h0622
MacGurmukhi	10	ISOLatin3	&h0203	JIS_X0212_90	&h0623
MacGujarati	11	ISOLatin4	&h0204	JIS_C6226_78	&h0624
MacOriya	12	ISOLatinCyrillic	&h0205	GB_2312_80	&h0630
MacBengali	13	ISOLatinArabic	&h0206	GBK_95	&h0631
MacTamil	14	ISOLatinGreek	&h0207	KSC_5601_87	&h0640
MacTelugu	15	ISOLatinHebrew	&h0208	KSC_5601_92_Johab	&h0641
MacKannada	16	ISOLatin5	&h0209	CNS_11643_92_P1	&h0651
MacMalayalam	17	DOSLatinUS	&h0400	CNS_11643_92_P2	&h0652
MacSinhalese	18	DOSGreek	&h0405	CNS_11643_92_P3	&h0653
MacBurmese	19	DOSBalticRim	&h0405	ISO_2022_JP	&h0820
MacKhmer	20	DOSLatin1	&h0410	ISO_2022_JP_2	&h0821
MacThai	21	DOSGreek1	&h0411	ISO_2022_CN	&h0830
MacLaotian	22	DOSLatin2	&h0412	ISO_2022_CN_EXT	&h0831
MacGeorgian	23	DOSCyrillic	&h0413	ISO_2022_KR	&h0840
MacArmenian	24	DOSTurkish	&h0414	EUC_JP	&h0920
MacChineseSimp	25	DOCPortuguese	&h0415	EUC_CN	&h0930
MacTibetan	26	DOSIslandic	&h0416	EUC_TW	&h0931
MacMongolian	27	DOSHebrew	&h0417	EUC_KR	&h0940
MacEthiopic	28	DOSCanadianFrench	&h0418	ShiftJIS	&h0A01
MacCentralEurRoman	29	DOSArabic	&h0419	KOI8_R	&h0A02
MacVietnamese	30	DOSNordic	&h041A	Big5	&h0A03
MacExtArabic	31	DOSRussian	&h041B	MacRomanLatin1	&h0A04
MacSymbol	33	DOSGreek2	&h041C	HZ_GB_2312	&h0A05
MacDingbats	34	DOSThai	&h041D	NextStepLatin	&h0B01
MacTurkish	35	DOSJapanese	&h0420	EBCDIC_US	&h0C01
MacCroatian	36	DOSChineseSimplif	&h0421	EBCDIC_CP037	&h0C02
MacIcelandic	37	DOSKorean	&h0422	MultiRun	&h0FFF
MacRomanian	38	DOSChineseTrad	&h0423	MacTradChinese	2
MacCeltic	39	WindowsLatin1	&h0500	MacRSSymbol	8
MacGaelic	40	WindowsANSI	&h0500	MacSimpChinese	25
MacUnicode	&h7E	WindowsLatin2	&h0501	MacGeez	28
MacFarsi	&h8C	WindowsCryillic	&h0502	MacEastEurRoman	29
MacUkrainian	&h98	WindowsGreek	&h0503	MacUninterp	32
MacInuit	&hEC	WindowsLatin5	&h0504		

Tabelle 2: Variants

Variant	Wert	Variant	Wert
Text Encoding Default Variant	0	Hebrew Figure Space Variant	1
Mac Roman Default Variant	0	Unicode Max Decomposed Variant	2
Mac Roman Currency Sign Variant	1	Japanese NoOneByteKanaOption	0020
Mac Roman Euro Sign Variant	2	Japanese Use ASCII Backslash Option	&h40
Mac Icelandic Std Default Variant	0		
Mac Icelandic TT Default Variant	1		
Mac Icelandic Std Euro Sign Variant	4		
Mac Icelandic TT Euro Sign Variant	5		
Mac Croatian Default Variant	0		
Mac Croatian Currency Sign Variant	1		
Mac Croatian Euro Sign Variant	2		
Mac Romanian Default Variant	0		
Mac Romanian Currency Sign Variant	1		
Mac Romanian Euro Sign Variant	2		
Mac Japanese Standard Variant	0		
Mac Japanese StdNoVerticals Variant	1		
Mac Japanese Basic Variant	2		
Mac JapanesePostScript Scrn Variant	3		
Mac Japanese PostScript Print Variant	4		
Mac Japanese VartAtKuPlusTen Variant	5		
Mac Arabic Standard Variant	0		
Mac Arabic TrueType Variant	1		
Mac Arabic Thuluth Variant	2		
Mac Arabic AlBayan Variant	3		
Mac Farsi Standard Variant	0		
Mac Farsi TrueType Variant	1		
Mac Hebrew Standard Variant	0		
Mac Hebrew FigureSpace Variant	1		
Mac VT100 Default Variant	0		
Mac VT100 CurrencySign Variant	1		
Mac VT100 EuroSign Variant	2		
Unicode No Subset	0		
Unicode Canonical Decomp Variant	2		
Big5_Basic Variant	0		
Big5_Standard Variant	1		
Big5_ETen Variant	2		
Unicode No Compatibility Variant	1		
Unicode No Composed Variant	3		
Unicode No Corporate Variant	4		
Mac Roman Standard Variant	0		
Mac Icelandic Standard Variant	0		
Mac Icelandic TrueType Variant	1		
Japanese Standard Variant	0		
Japanese Std No Verticals Variant	1		
Japanese Basic Variant	2		
Japanese PostScript Scrn Variant	3		
Japanese PostScript Print Variant	4		
Japanses VertAtKuPlusTen Variant	5		
Hebrew Standard Variant	0		

Anhang C: ASCII-Codes

Die Tabellen in diesem Anhang listen die ASCII-Codes für Macintosh- und Windows-Systeme auf. Tabelle 1 zeigt die Dezimal-, Hexadezimal- und Oktal-Werte für die Standard-ASCII-Zeichen (0 bis 127) an, Tabelle 2 zeigt den „erweiterten“ Zeichenvorrat für die Macintosh- und Windows-Plattformen für die US-Version der aktuellen Macintosh- und Windows-Betriebssysteme. Die Zeichen, die zu Codes größer als 127 gehören, hängen vom verwendeten Text-Encoding ab.

Tabelle 1: Standard ASCII-Codes

Decimal	Hex	Result	Decimal	Hex	Result
0	0	NUL	64	40	@
1	1	SOH	65	41	A
2	2	STX	66	42	B
3	3	ETX	67	43	C
4	4	EOT	68	44	D
5	5	ENQ	69	45	E
6	6	ACK	70	46	F
7	7	BEL	71	47	G
8	8	BS	72	48	H
9	9	HT	73	49	I
10	A	LF	74	4A	J
11	B	VT	75	4B	K
12	C	FF	76	4C	L
13	D	CR	77	4D	M
14	E	SO	78	4E	N
15	F	SI	79	4F	O
16	10	DLE	80	50	P
17	11	DC1	81	51	Q
18	12	DC2	82	52	R
19	13	DC3	83	53	S
20	14	DC4	84	54	T
21	15	NAK	85	55	U
22	16	SYN	86	56	V
23	17	ETB	87	57	W
24	18	CAN	88	58	X
25	19	EM	89	59	Y
26	1A	SUB	90	5A	Z
27	1B	ESC	91	5B	[
28	1C	FS	92	5C	\
29	1D	GS	93	5D	]
30	1E	RS	94	5E	^
31	1F	US	95	5F	-
32	20	SP	96	60	`
33	21	!	97	61	a
34	22	"	98	62	b
35	23	#	99	63	c

Tabelle 1: Standard ASCII-Codes (Fortsetzung)

Decimal	Hex	Result	Decimal	Hex	Result
36	24	\$	100	64	d
37	25	%	101	65	e
38	26	&	102	66	f
39	27	'	103	67	g
40	28	(	104	68	h
41	29	)	105	69	i
42	2A	*	106	6A	j
43	2B	+	107	6B	k
44	2C	,	108	6C	l
45	2D	-	109	6D	m
46	2E	.	110	6E	n
47	2F	/	111	6F	o
48	30	0	112	70	p
49	31	1	113	71	q
50	32	2	114	72	r
51	33	3	115	73	s
52	34	4	116	74	t
53	35	5	117	75	u
54	36	6	118	76	v
55	37	7	119	77	w
56	38	8	120	78	x
57	39	9	121	79	y
58	3A	:	122	7A	z
59	3B	;	123	7B	{
60	3C	<	124	7C	
61	3D	=	125	7D	}
62	3E	>	126	7E	~
63	3F	?	127	7F	DEL

Tabelle 2: Erweiterte ASCII-Codes

Decimal	Hex	Result (Mac)	Result (Windows)	Decimal	Hex	Result (Mac)	Result (Windows)
128	80	Ä	⌘	192	C0	ı	À
129	81	Å		193	C1	ì	Á
130	82	Ç	,	194	C2	¬	Â
131	83	É	f	195	C3	√	Ã
132	84	Ñ	„	196	C4	ƒ	Ä
133	85	Ö	...	197	C5	≈	Å
134	86	Ü	†	198	C6	Δ	Æ
135	87	á	‡	199	C7	«	Ç
136	88	à	^	200	C8	»	È
137	89	â	‰	201	C9	...	É
138	8A	ä	§	202	CA	(space)	Ê
139	8B	ã	‹	203	CB	À	Ë
140	8C	å	œ	204	CC	Ã	Ì
141	8D	ç		205	CD	Ö	Í
142	8E	é		206	CE	œ	Î
143	8F	è		207	CF	œ	Ï
144	90	ê		208	D0	–	Ð
145	91	ë	'	209	D1	—	Ñ
146	92	í	'	210	D2	“	Ò
147	93	ì	”	211	D3	”	Ó
148	94	î	”	212	D4	'	Ô
149	95	ï	•	213	D5	'	Õ
150	96	ñ	–	214	D6	÷	Ö
151	97	ó	—	215	D7	◊	×
152	98	ò	~	216	D8	ÿ	Ø
153	99	ô	™	217	D9	ÿ	Ù
154	9A	ö	š	218	DA	/	Ú
155	9B	õ	›	219	DB		Û
156	9C	ú	œ	220	DC	‹	Ü
157	9D	ù		221	DD	›	Ý
158	9E	û		222	DE	fi	Þ
159	9F	ü	ÿ	223	DF	fl	ß
160	A0	†		224	E0	‡	à
161	A1	°	ı	225	E1	·	á
162	A2	¢	¢	226	E2	,	â
163	A3	£	£	227	E3	„	ã
164	A4	§	⌘	228	E4	‰	ä
165	A5	•	¥	229	E5	À	å
166	A6	¶	ı	230	E6	Ê	æ
167	A7	ß	§	231	E7	Á	ç

Tabelle 2: Erweiterte ASCII-Codes (Fortsetzung)

Decimal	Hex	Result (Mac)	Result (Windows)	Decimal	Hex	Result (Mac)	Result (Windows)
168	A8	®	™	232	E8	Ë	è
169	A9	©	©	233	E9	È	é
170	AA	™	ª	234	EA	Í	ê
171	AB	´	«	235	EB	Î	ë
172	AC	¨	¬	236	EC	Ï	ì
173	AD	≠	-	237	ED	Ì	í
174	AE	Æ	®	238	EE	Ó	î
175	AF	Ø	-	239	EF	Ô	ï
176	B0	∞	°	240	F0	🍏	ð
177	B1	±	±	241	F1	Ò	ñ
178	B2	≤	²	242	F2	Ú	ò
179	B3	≥	³	243	F3	Û	ó
180	B4	¥	´	244	F4	Ü	ô
181	B5	μ	μ	245	F5	ı	õ
182	B6	∂	¶	246	F6	ˆ	ö
183	B7	Σ	·	247	F7	˜	÷
184	B8	Π	„	248	F8	-	ø
185	B9	π	¡	249	F9	˘	ù
186	BA	∫	º	250	FA	˙	ú
187	BB	ª	»	251	FB	°	û
188	BC	º	¼	252	FC	¸	ü
189	BD	Ω	½	253	FD	˝	ý
190	BE	æ	¾	254	FE	˚	þ
191	BF	ø	¿	255	FF	ˇ	ÿ

Alphabetische Befehlsübersicht

#If ...#Else...#Endif Befehl	6
#Pragma Direktive	7
& Operator	8
' Operator	9
* Operator	10
+ Operator	10
/ Operator	11
// Operator	11
< Operator	12
<= Operator	12
<> Operator	13
= Operator	13
> Operator	14
>= Operator	15
\ Operator	15
_ Schlüsselwort	7
Abs Funktion	16
Abschließende Gänsefüßchen fehlen Fehler	16
Abstrakte Klasse kann nicht vererbt werden Fehler	16
Acos Funktion	17
AddressBook Klasse	17
AddressBookAddress Klasse	18
AddressBookContact Klasse	19
AddressBookContact Klasse	20
AddressBookData Klasse	21
AddressBookGroup Klasse	22
AddressBookRecord Klasse	23
AddressOf Operator	24
ADSP4DServer Klasse	24
And Operator	25
App Objekt	25
Append Methode	26
AppleEvent Klasse	26
AppleEventDescList Klasse	29
AppleEventObjectSpecifier Klasse	29
AppleEventRecord Klasse	30
AppleEventTarget Klasse	31
AppleEventTemplate Klasse	31
AppleMenuFolder Funktion	31
Application Klasse	32
ApplicationSupportFolder Funktion	35
ArcShape Klasse	35
Array Funktion	36
Arraygrenzen müssen Integer sein Fehler	37
Asc Funktion	37
AscB Funktion	38
Asin Funktion	38
Assigns kann nur auf den letzten Parameter angewendet werden Fehler	39
Assigns Schlüsselwort	39
Atan Funktion	39
Atan2 Funktion	40
AutoDiscovery Klasse	40
Beep Methode	41
BevelButton Steuerelement	42
Bezeichner erwartet Fehler	45
Bin Funktion	45
BinaryStream Klasse	45
Bitwise Klasse	47
BitwiseAnd Funktion	48
BitwiseOr Funktion	49
BitwiseXor Funktion	49
Boolean Datentyp	49

Bounds3D Klasse	49
Break Ausdruck	50
ByRef Schlüsselwort	51
ByVal Schlüsselwort	52
Call Befehl	52
Canvas Steuerelement	53
CDbl Funktion	55
Ceil Funktion	55
ChasingArrows Steuerelement	56
Checkbox Steuerelement	56
Chr Funktion	57
ChrB Funktion	58
ClearFocus Methode	58
Clipboard Klasse	58
CMY Funktion	60
Collection Klasse	60
Color Datentyp	61
ComboBox Steuerelement	62
ConsoleApplication Klasse	64
Const Befehl	65
ContextualMenu Steuerelement	65
Control Klasse	66
ControlPanelsFolder Funktion	69
ConvertEncoding Funktion	69
Cos Funktion	70
CountFields Funktion	70
CriticalSection Klasse	71
Cross-Plattform-Entwicklung	5
CStr Funktion	71
CurveShape Klasse	72
DarkBevelColor Funktion	73
DarkTingeColor Funktion	73
DataAvailableProvider Interface Klasse	74
Database Klasse	74
DatabaseCursor Klasse	81
DatabaseCursorField Klasse	81
DatabaseField Klasse	81
DatabaseQuery Steuerelement	82
DataBaseRecord Klasse	83
DataControl Steuerelement	84
Datagram Klasse	88
DataNotificationReceiver Interface Klasse	88
DataNotifier Interface Klasse	88
Date Klasse	88
DebugBuild Konstante	90
DebugDumpObjects Methode	91
Declare Befehl	91
DecodeBase64 Funktion	93
DecodeQuotedPrintable Funktion	94
DefineEncoding Funktion	94
Deklaration Enthält ein reserviertes Wort	95
Der New Operator kann nicht mit einem Klassen-Interface verwendet werden Fehler	95
DesktopFolder Funktion	95
Destructoren können keine Parameter haben Fehler	96
Dictionary Klasse	96
Die Größe eines Arrays muss eine Konstante oder eine Zahl sein Fehler	97
Die Klassen-Hierarchie	4
Die Zählervariable und die Variable nach Next müssen übereinstimmen Fehler	98
Dies ist kein Array, aber Sie benutzen es wie eines Fehler	98
Diese Eigenschaft besitzt den gleichen Namen wie eine Klassen-Methode Fehler	100
Diese Eigenschaft hat den selben Namen wie ein Event Fehler	100
Diese Eigenschaft hat den selben Namen wie eine Methode Fehler	100
Diese Eigenschaft ist geschützt. Sie kann nur innerhalb ihrer Klasse verwendet werden Fehler	101

Diese Form eines Array kann nicht sortiert werden Fehler	98
Diese globale Variable hat den selben Namen wie eine globale Funktion Fehler	99
Diese globale Variable hat den selben Namen wie eine Klasse Fehler	99
Diese lokale Variable oder Konstante hat den selben Namen wie ein Declare in dieser Methode Fehler	99
Diese Methode benötigt mehr Parameter als übergeben wurden Fehler	101
Diese Methode benötigt weniger Parameter als übergeben wurden Fehler	100
Diese Methode ist geschützt. Sie kann nur innerhalb ihrer Klasse verwendet werden Fehler	101
Diese Methode liefert keinen Wert zurück Fehler	101
Diese Methode, Eigenschaft oder Variable existiert nicht Fehler	99
Diese Zeile ist zu komplex Fehler	101
Diese(r) lokale Variable/Parameter hat den selben Namen wie eine Konstante Fehler	99
Dieser Klasse fehlt eine oder mehrere Methoden eines Interfaces, das sie erbt Fehler	103
Dieser Name wird bereits verwendet Fehler	103
Dieses #else hat kein vorhergehendes passendes #if Fehler	102
Dieses #endif hat kein vorhergehendes passendes #if Fehler	102
Dieses Array hat mehr Dimensionen als Sie angegeben haben Fehler	102
Dieses Array hat weniger Dimensionen als Sie angegeben haben Fehler	102
Dieses Objekt steht im Konflikt zu einem anderen Objekt gleichen Namens Fehler	103
Dim Befehl	104
DisclosureTriangle Steuerelement	106
Do...Loop Befehl	106
DocumentsFolder Funktion	106
Double Datentyp	107
DragItem Klasse	108
Dynamische Steuerelemente können nur als Teil eines Steuerelement-Arrays erzeugt werden Fehler	110
EasyTCPSocket Klasse	110
EasyUDPSocket Klasse	112
EditableMovie Klasse	114
EditField Steuerelement	117
Eigenschaft hat keinen Wert Fehler	123
Ein Assigns-Parameter kann keinen Standardwert haben Fehler	124
Ein Erweiterter Parameter kann keinen Standardwert haben Fehler	123
Ein Interface dieser Klasse ist nicht vom Typ Klassen-Interface Fehler	124
Ein ParamArray kann keine Arrays enthalten Fehler	124
Ein Parameter nach einem Parameter-Array Fehler	124
Ein Parameter-Array kann keinen Standardwert haben Fehler	124
Ein Parameter-Array kann nicht als Referenz übergeben werden Fehler	125
Ein Referenz-Parameter kann keinen Standardwert haben Fehler	125
Eine Methode mit einem Parameter-Array, kann keine optionalen Parameter haben Fehler	124
Einführung	3
EmailAttachment Klasse	125
EmailHeaders Klasse	126
EmailMessage Klasse	127
EnableMenuItems Methode	128
EncodeBase64 Funktion	128
EncodeQuotedPrintable Funktion	128
Encoding Funktion	129
Encodings Objekt	129
EndOfLine Klasse	130
Es gibt keine Klasse mit diesem Namen Fehler	131
ExcelApplication Klasse	131
Exception Block	132
Exception Objekte müssen vom Typ RuntimeException sein Fehler	134
Exit Befehl	134
Exp Funktion	134
ExportPicture Funktion	134
Extends kann nur auf den ersten Parameter angewendet werden Fehler	136
Extends Schlüsselwort	135
ExtensionsFolder Funktion	136
Externe Funktionen können Objekte nicht als Parameter verwenden Fehler	136
Externe Funktionen können String-Datentypen nicht als Parameter verwenden Fehler	137
Falscher Typ Fehler	137
False Schlüsselwort	137

FigureShape Klasse	137
FillColor Funktion	138
Floor Funktion	139
FolderItem Klasse	139
FolderItemDialog Klasse	147
Font Funktion	149
FontCount Funktion	150
FontsFolder Funktion	150
For Each...Next Befehl	150
For...Next Befehl	151
Format Funktion	153
FrameColor Funktion	154
Function Befehl	154
Für die Deklaration von Bibliotheken können nur String-Konstanten verwendet werden Fehler	154
GameInputElement Klasse	156
GameInputManager Klasse	156
GetAppleEventTarget Funktion	158
GetFolderItem Funktion	158
GetFontTextEncoding Funktion	159
GetIndexedObjectDescriptor Funktion	160
GetInternetTextEncoding Funktion	160
GetNamedObjectDescriptor Funktion	160
GetOpenFolderItem Funktion	161
GetOrdinalObjectDescriptor Funktion	162
GetPropertyObjectDescriptor Funktion	163
GetQTCrossFadeEffect Funktion	163
GetQTGraphicsExporter Funktion	163
GetQTSMPTEEffect Funktion	164
GetRangeObjectDescriptor Funktion	167
GetSaveFolderItem Funktion	168
GetStringComparisonObjectDescriptor Funktion	169
GetTemporaryFolderItem	169
GetTestObjectDescriptor Funktion	169
GetTextConverter Funktion	170
GetTextEncoding Funktion	170
GetTrueFolderItem Funktion	171
GetUniqueIDObjectDescriptor Funktion	172
GOTO Befehl	172
GOTO Ziel nicht gefunden Fehler	172
Graphics Klasse	173
Group2D Klasse	176
Group3D Klasse	177
GroupBox Steuerelement	177
GuessJapaneseEncoding Funktion	178
Hex Funktion	179
HighlightColor Funktion	179
HSV Funktion	179
HTTPSecureSocket Klasse	182
HTTPSocket Klasse	180
If...Then...Else Befehl	184
IllegalCastException Laufzeitfehler	185
ImageWell Steuerelement	186
IndexOf Methode	186
Input Methode	187
Insert Methode	188
InStr Funktion	188
InStrB Funktion	189
Integer Datentyp	189
InternetHeaders Klasse	190
InvalidParentException Fehler	187
IPCSocket Klasse	190
Is Operator	193
IsA Operator	193

IsCMMClick Funktion	195
IsNumeric Funktion	195
Join Funktion	196
Kann dieser Eigenschaft keinen Wert zuweisen Fehler	196
Keyboard Objekt	196
KeyChain Klasse	198
KeyChainException Fehler	200
KeyChainItem Klasse	201
KeyNotFoundException Fehler	203
Konventionen	3
Left Funktion	203
LeftB Funktion	204
Len Funktion	204
LenB Funktion	205
Light3D Klasse	205
LightBevelColor Funktion	206
LightTingeColor Funktion	206
Line Steuerelement	206
ListBox Steuerelement	207
ListColumn Klasse	223
ListSelectionNotificationReceiver Interface Klasse	224
ListSelectionNotifier Interface Klasse	225
LittleArrows Steuerelement	225
Log Funktion	225
Logische Operationen benötigen boole'sche Werte Fehler	225
Lowercase Funktion	226
LTrim Funktion	226
Max Funktion	226
MD5 Funktion	227
MD5Digest Klasse	227
MDIWindow Klasse	228
Me kann nicht in einer Methode oder einem Modul verwendet werden Fehler	229
Me Schlüsselwort	229
MemoryBlock Klasse	229
MenuItem Klasse	231
MessageDialog Klasse	233
MessageDialogButton Klasse	235
Microseconds Funktion	235
Mid Funktion	236
MidB Funktion	237
Min Funktion	237
Mit #if können nur Konstanten vom Typ Boolean verwendet werden Fehler	238
Mod Operator	238
MouseCursor Klasse	238
Movie Klasse	239
MoviePlayer Control	240
MsgBox Funktion	243
MySQLDatabase Klasse	245
NetworkInterface Objekt	246
New Operator	247
NewAppleEvent Funktion	248
NewAppleEventTarget Funktion	248
NewDragItem Funktion	249
NewMemoryBlock Funktion	250
NewPicture Funktion	250
NewREALdatabase Funktion	252
Nil Schlüsselwort	252
NilObjectException Laufzeitfehler	252
Not Operator	253
NotePlayer Steuerelement	254
NthField Funktion	256
Object Klasse	257
Object2D Klasse	257

Object3D Klasse	258
Oct Funktion	260
ODBCDatabase Klasse	260
Office Klasse	261
OLEContainer Steuerelement	263
OLEException Fehler	264
OLEObject Klasse	264
OLEParameter Klasse	265
Open4DDatabasebyADSP Funktion	265
Open4DDatabasebyTCPIP Funktion	265
OpenBaseDatabase Klasse	265
OpenCSVCursor Funktion	266
OpenDBFCursor Funktion	267
OpenDialog Klasse	267
OpenDTFDatabase Funktion	268
OpenODBCDatabase Funktion	269
OpenOpenBaseDatabase Funktion	269
OpenOracleDatabase Funktion	269
OpenPostgreSQLDatabase Funktion	269
OpenPrinter Funktion	270
OpenPrinterDialog Funktion	270
OpenREALdatabase Funktion	271
OpenURLMovie Funktion	271
Operator_ Schlüsselwort	271
Or Operator	273
OracleDatabase Klassen	274
OutOfBoundsException Fehler	274
OutOfMemoryException Fehler	275
Oval Steuerelement	276
OvalShape Klasse	276
PagePanel Steuerelement	277
Paragraph Klasse	277
ParamArray Schlüsselwort	278
Parameter sind nicht zu dieser Funktion kompatibel	279
Parameter und Standardwert müssen vom selben Typ sein	279
ParseDate Funktion	279
Picture Klasse	280
PixmapShape Klasse	281
Placard Steuerelement	282
Pop Methode	283
POP3SecureSocket Klasse	284
POP3Socket Klasse	286
PopupArrow Steuerelement	287
PopupMenu Steuerelement	288
PostgreSQLDatabase Klasse	290
Pow Funktion	291
PowerPointApplication Klasse	291
PreferencesFolder Funktion	291
PrefsMenuItem Klasse	292
Print Methode	293
PrinterSetup Klasse	293
ProgressBar Steuerelement	295
PushButton Steuerelement	296
QT3DAudio Klasse	298
QTEffect Klasse	299
QTEffectSequence Klasse	299
QTGraphicsExporter Klasse	299
QTSoundTrack Klasse	301
QTTrack Klasse	302
QTUserData Klasse	303
QTVideoTrack Klasse	304
Quaternion Klasse	306
QuickTime Objekt	297

Quit Methode	308
QuitMenuItem Klasse	308
RadioButton Steuerelement	309
Raise Befehl	310
Random Klasse	310
Range Klasse	311
RB3DSpace Steuerelement	312
RBScript Steuerelement	316
RBScriptAlreadyRunningException Fehler	323
RBVersion Konstante	323
RBVersionString Konstante	324
REALdatabase Klasse	324
REALdatabaseOldFormat Klasse	327
RecordSet Klasse	327
Rectangle Steuerelement	329
RectControl Klasse	330
RectShape Klasse	334
ReDim Methode	335
RegEx Klasse	336
RegExceptionHandler	341
RegExMatch Klasse	341
RegExOptions Klasse	341
RegExSearchPatternException Fehler	342
RegistryAccessErrorException Fehler	343
RegistryItem Klasse	343
Rem Befehl	344
Remove Methode	344
Replace Funktion	345
ReplaceAll Funktion	345
ReplaceAllB Funktion	346
ReplaceB Funktion	347
ReplaceLineEndings Funktion	347
ResourceFork Klasse	348
Return Befehl	350
RGB Funktion	351
RGBSurface Object	351
Right Funktion	352
RightB Funktion	352
Rnd Funktion	353
Round Funktion	353
RoundRectangle Steuerelement	354
RoundRectShape Klasse	354
RTrim Funktion	355
Runtime Objekt	355
RuntimeException Klasse	356
SaveAsDialog Klasse	358
Screen Funktion	360
Screen Klasse	359
ScreenCount Funktion	360
Scrollbar Steuerelement	361
Select Case Befehl	362
Select4DDatabaseByADSP Funktion	363
Select4DDatabaseByTCPIP Funktion	364
SelectColor Funktion	364
SelectFolder Funktion	364
SelectFolderDialog Klasse	365
SelectODBCDatabase Funktion	366
Self kann nicht in der Methode eines Modules verwendet werden Fehler	366
Self Schlüsselwort	366
Semaphore Klasse	366
Separator Steuerelement	367
Serial Steuerelement	368
SerialPort Klasse	370

ServerSocket Klasse	371
ServiceApplication Klasse	372
Shell Klasse	373
ShellNotRunningException Fehler	374
ShowURL Methode	374
Shuffle Methode	375
ShutDownItemsFolder Funktion	376
Sie haben einen Operator verwendet, der nicht zu den spezifizierten Datentypen kompatibel ist Fehler	376
Sie können den New-Operator nicht mit dieser Klasse verwenden Fehler	376
Sie können diesen Variablen- oder Eigenschaften-Namen nicht verwenden, da es ein reserviertes Wort ist Fehler	377
Sie können ein Array nicht an eine externe Funktion übergeben Fehler	377
Sie können ein Array nicht einem anderen Array zuweisen Fehler	377
Sie können einen Ausdruck nicht als Parameter, der als ByRef definiert ist, übergeben Fehler	377
Sie können keinen Wert zurückgeben, da für diese Methode kein Rückgabotyp definiert wurde Fehler	378
Sie können nicht ein nichtexistierendes Event implementieren Fehler	378
Sie können Super nicht verwenden, da diese Klasse keine übergeordnete Klasse besitzt Fehler	378
Sie müssen den Datentyp des Rückgabewertes angeben Fehler	378
Sign Funktion	378
Sin Funktion	379
Single Datentyp	379
Slider Steuerelement	380
SMTPSecureSocket Klasse	381
SMTPSocket Klasse	381
Socket Klasse	383
SocketCore Klasse	383
Sort Methode	386
Sound Klasse	386
Speak Methode	387
Split Funktion	388
Sprachreferenz	6
Sprite Klasse	388
SpriteSurface Steuerelement	389
Sqrt Funktion	393
SSLSocket Steuerelement	393
StackOverflowException Fehler	395
StandardInputStream Klasse	395
StandardOutputStream Klasse	396
StandardToolBarItem Steuerelement	396
StartupItemsFolder Funktion	397
StaticText Steuerelement	397
StdErr Methode	399
StdIn Methode	399
StdOut Methode	400
Str Funktion	400
StrComp Funktion	401
String Datentyp	401
StringProvider Interface Class	402
StringShape Klasse	402
StyledText Klasse	403
StyledTextPrinter Klasse	406
StyleRun Klasse	407
Sub Befehl	408
Super kann nicht in einer Modul-Methode verwendet werden Fehler	409
Syntax Error Fehler	409
System Objekt	411
SystemFolder Funktion	414
TabPanel Steuerelement	415
Tan Funktion	416
Target68K Konstante	417
TargetCarbon Konstante	417
TargetHasGUI Konstante	417
TargetLinux Konstante	418
TargetMachO Konstante	418

TargetMacOS Konstante	418
TargetPPC Konstante	419
TargetWin32 Konstante	419
TCP4Server Klasse	419
TCP4Socket Steuerelement	420
TemporaryFolder Funktion	423
TextColor Funktion	423
TextConverter Klasse	424
TextEncoding Klasse	424
TextInputStream Klasse	426
TextOutputStream Klasse	427
Thread Klasse	428
Ticks Funktion	429
Timer Objekt	430
Titlecase Funktion	431
ToolBarItem Steuerelement	431
TrashFolder Funktion	433
Trim Funktion	433
True Schlüsselwort	433
Try Block	434
TypeMismatchException Fehler	434
Ubound Funktion	435
UDPSocket Klasse	436
Unbekanntes Pragma Fehler	438
UnsupportedFormatException Fehler	438
Uppercase Funktion	438
UserCancelled Funktion	439
Val Funktion	439
Variant Datentyp	440
VarType Funktion	441
Vector3D Klasse	442
Vererbung von einer Klasse ist nicht möglich Fehler	443
VirtualVolume Klasse	443
Volume Funktion	444
VolumeCount Funktion	445
While...Wend Befehl	445
Window Funktion	446
Window Klasse	446
WindowCount Funktion	454
WordApplication Klasse	454
XMLAttribute Klasse	454
XMLAttributeList Klasse	455
XMLCDATASection Klasse	455
XMLComment Klasse	455
XMLContentModel Klasse	456
XMLDocument Klasse	457
XMLDOMException Error	459
XMLElement Klasse	460
XMLException Fehler	460
XMLNode Klasse	461
XMLNodeList Klasse	462
XMLNodeType Objekt	462
XMLProcessingInstruction Klasse	463
XMLReader Klasse	463
XMLReaderException Klasse	466
XMLStyleSheet Klasse	467
XMLTextNode Klasse	467
XMLXslHandler Klasse	467

Befehle nach Themen

3D-Grafiken

Bounds3D	49
Group3D Klasse	177
Light3D	205
Object3D Klasse	258
Quaternion Klasse	306
RB3DSpace Steuerelement	312
Vector3D Klasse	442

Adressbücher

AddressBook Klasse	17
AddressBookAddress Klasse	18
AddressBookContact Klasse	19, 20
AddressBookData Klasse	21
AddressBookGroup Klasse	21
AddressBookRecord Klasse	23

AppleEvents

AppleEvent Klasse	26
AppleEventDescList Klasse	29
AppleEventObjectSpecifier Klasse	29
AppleEventRecord Klasse	30
AppleEventTarget Klasse	31
AppleEventTemplate Klasse	31
Application Klasse	32
GetAppleEventTarget (Prompt, ListLabel) → AppleEventTarget	158
GetIndexedObjectDescriptor(gewünschte Klasse, Objekt, Index) → AppleEventObjectSpecifier	160
GetNamedObjectDescriptor(gewünschte Klasse, Objekt, Name) → AppleEventObjectSpecifier	160
GetOrdinalObjectDescriptor(gewünschte Klasse, Objekt, Ordnungsschlüssel) → AppleEventObjectSpecifier	162
GetPropertyObjectDescriptor(Objekt, Name) → AppleEventObjectSpecifier	163
GetRangeObjectDescriptor(Des. Class, object, rangeStart, rangeEnd) → AppleEventObjectSpecifier	167
GetStringComparisonObjectDescriptor (comp. Key, comp. Form, field, value) → AppleEventObjectSpecifier	169
GetTestObjectDescriptor(desiredClass, object, comparison) → AppleEventObjectSpecifier	169
GetUniqueIDObjectDescriptor(DesiredClass, Object, ID) → AppleEventObjectSpecifier	172
NewAppleEvent(eventClass, eventID, creatorCode) → AppleEvent	248
NewAppleEventTarget (Name, Computer, Zone, PortType) → AppleEventTarget	248

Arrays

Append	26
Array(ElementList) → Array	36
Dim Befehl	104
IndexOf	186
Insert	188
Join(sourceArray, delimiter) → String	196
OutOfBoundsException Fehler	274
OutOfMemoryException Fehler	275
ParamArray Schlüsselwort	278
Pop	283
Redim	335
Remove	344
Shuffle	375
Sort	386

Split(source, delimiter) → String array	388
Ubound(array) → Integer	435

Boolean

False	137
Nil	252
Not Operator	253
Or Operator	273
True	433

Code-Ausführung

#If ... #Else ... #Endif	6
' Operator	9
// Operator	11
_ Schlüsselwort	7
Assigns Schlüsselwort	39
ByRef Schlüsselwort	51
ByVal Schlüsselwort	52
Call Befehl	52
Do...Loop	106
EnableMenuItems	128
Exit	134
Extends Schlüsselwort	135
For Each ... Next	150
For...Next	151
Function Befehl	154
GOTO Befehl	172
If...Then...Else	184
Pragma Direktive	7
Quit	308
RBScript Steuerelement	316
Rem	344
Return Befehl	350
Select Case	362
Semaphore Klasse	366
Sub Befehl	408
Thread Klasse	428
Timer Steuerelement	430
UserCancelled → Boolean	439
While...Wend	445

Dateien

AppleMenuFolder → FolderItem	31
Application Klasse	32
ApplicatonSupportFolder → FolderItem	35
BinaryStream	45
ControlPanelsFolder → FolderItem	69
CountFields(Quelle,Separator) → Integer	70
DesktopFolder → FolderItem	95
DocumentsFolder → FolderItem	106
ExportPicture (pic) → Boolean	134
ExtensionsFolder → FolderItem	136
FolderItem Klasse	139
FolderItemDialog Klasse	147
GetFolderItem(Pfad) → String	158
GetOpenFolderItem(Filter) → FolderItem	161

GetSaveFolderItem(filter, default file name) → FolderItem	168
GetTemporaryFolderItem → FolderItem	169
GetTrueFolderItem (Name) → FolderItem	171
OpenDialog Klasse	267
PreferencesFolder → FolderItem	291
SaveAsDialog Klasse	358
SelectFolder → FolderItem	364
SelectFolderDialog Klasse	365
ShutDownItemsFolder → FolderItem	376
StandardInputStream Klasse	395
StandardOutputStream Klasse	396
StartupItemsFolder → FolderItem	397
SystemFolder → FolderItem	414
TemporaryFolder → FolderItem	423
TextInputStream Klasse	426
TextOutputStream Klasse	427
TrashFolder → FolderItem	433
VirtualVolume Klasse	443
Volume(volumeNumber) → FolderItem	444
VolumeCount → Integer	445

Datenbank

ADSP4DServer Klasse	24
Database Klasse	74
DatabaseCursor Klasse	81
DatabaseCursorField Klasse	81
DatabaseField Klasse	81
DatabaseQuery Steuerelement	82
DatabaseRecord Klasse	83
DataControl Steuerelement	84
MySQLDatabase Klasse	245
NewREALDatabase (FolderItem) → Database	252
ODBCDatabase Klasse	260
Open4DDatabaseByADSP (connectionstring, task, username, password) → Database	265
Open4DDatabaseByTCPIP (connectionstring, task, username, password) → Datenbank	265
OpenBaseDatabase Klasse	265
OpenCSVCursor Methode	266
OpenDBFCursor (FolderItem) → DatabaseCursor	267
OpenDTFDatabase (dbFile, blobFile, username, password) → Database	268
OpenODBCDatabase (datasource) → Datenbank	269
OpenOpenBaseDatabase (host, database, username, password) → Datenbank	269
OpenOracleDatabase (username, password) → Database	269
OpenPostgreSQLDatabase (host, port, database, user, password) → Datenbank	269
OpenREALDatabase (file) → Datenbank	271
OracleDatabase Klasse	274
PostgreSQLDatabase Klasse	290
REALDatabase Klassen	324
REALDatabaseOldFormat Klasse	327
RecordSet Klasse	327
Select4DDatabaseByADSP (task, username, password) → Datenbank	363
Select4DDatabaseByTCPIP (IPaddress, username, password) → Datenbank	364
SelectODBCDatabase → Datenbank	366
TCPIP4DServer Klasse	419

Datentypen

Boolean	49

Color	61
Double	107
Integer	189
IsNumeric(arg) ➔ Boolean	195
Single	379
String	401
Variant	440

Datum und Uhrzeit

Date Klasse	88
Microseconds ➔ Double	235
ParseDate (Text, ByRef ParsedDate) ➔ Boolean	279
Ticks ➔ Integer	429

Debugging

DebugDumpObjects(Dateiname)	91
Runtime Objekt	355

Drucken

OpenPrinter(pageSetup) ➔ Graphics	270
OpenPrinterDialog(pageSetup) ➔ Graphics	270
PrinterSetup	293
StyledTextPrinter Klasse	406

Einfaches Netzwerk

AutoDiscovery Klasse	40
EasyTCPSocket Klasse	110
EasyUDPSocket Klasse	112

Einführung

Cross-Plattform-Entwicklung	5
Die Klassen-Hierarchie	4
Konventionen	3
Überblick	3

Email

EmailAttachment Klasse	125
EmailHeaders Klasse	126
EmailMessage Klasse	127
POP3SecureSocket Klasse	284
POP3Socket Klassen	286
SMTPSecureSocket Klasse	381
SMTPSocket Klasse	381

Fehlerbehandlung

Exception Block	132
IllegalCastException Laufzeitfehler	185
InvalidParentException Fehler	187
KeyChainException Fehler	200
KeyNotFoundException Fehler	203
Nil Funktion	252
NilObjectException Laufzeitfehler	252
OLEException Fehler	264
OutOfBoundsException Fehler	274
OutOfMemoryException Fehler	275

Raise	310
RbScriptAlreadyRunningException Fehler	323
RegexException Fehler	341
RegexSearchPatternException Fehler	342
RegistryAccessErrorException Fehler	343
Runtime Exception Klasse	356
ShellNotRunningException Fehler	374
StackOverflowException Fehler	395
Try Block	434
TypeMismatchException Fehler	434

Fehlermeldungen

Abschließende Gänsefüßchen fehlen Fehler	16
Abstrakte Klasse kann nicht vererbt werden	16
Arraygrenzen müssen Integer sein	37
Assigns kann nur auf den letzten Parameter angewendet werden Fehler	39
Bezeichner erwartet	45
Deklaration Enthält ein reserviertes Wort Fehler	95
Der New Operator kann nicht mit einem Klassen-Interface verwendet werden Fehler	95
Destructoren können keine Parameter haben Fehler	96
Die Größe eines Arrays muss eine Konstante oder eine Zahl sein Fehler	97
Die Zählervariable und die Variable nach Next müssen übereinstimmen Fehler	98
Dies ist kein Array, aber Sie benutzen es wie eines Fehler	98
Diese Eigenschaft besitzt den gleichen Namen wie eine Klassen-Methode Fehler	100
Diese Eigenschaft hat den selben Namen wie ein Event Fehler	100
Diese Eigenschaft hat den selben Namen wie eine Methode Fehler	100
Diese Eigenschaft ist geschützt. Sie kann nur innerhalb ihrer Klasse verwendet werden Fehler	101
Diese Form eines Array kann nicht sortiert werden Fehler	98
Diese globale Variable hat den selben Namen wie eine globale Funktion Fehler	99
Diese globale Variable hat den selben Namen wie eine Klasse	99
Diese lokale Variable oder Konstante hat den selben Namen wie ein Declare in dieser Methode Fehler	99
Diese Methode benötigt mehr Parameter als übergeben wurden Fehler	101
Diese Methode benötigt weniger Parameter als übergeben wurden Fehler	100
Diese Methode ist geschützt. Sie kann nur innerhalb ihrer Klasse verwendet werden Fehler	101
Diese Methode liefert keinen Wert zurück Fehler	101
Diese Methode, Eigenschaft oder Variable existiert nicht Fehler	99
Diese Zeile ist zu komplex Fehler	101
Diese(r) lokale Variable/Parameter hat den selben Namen wie eine Konstante Fehler	99
Dieser Klasse fehlt eine oder mehrere Methoden eines Interfaces, das sie erbt Fehler	103
Dieser Name wird bereits verwendet	103
Dieses #else hat kein vorhergehendes passendes #if Fehler	102
Dieses #endif hat kein vorhergehendes passendes #if Fehler	102
Dieses Array hat mehr Dimensionen als Sie angegeben haben Fehler	102
Dieses Array hat weniger Dimensionen als Sie angegeben haben Fehler	102
Dieses Objekt steht im Konflikt zu einem anderen Objekt gleichen Namens Fehler	103
Dynamische Steuerelemente können nur als Teil eines Steuerelement-Arrays erzeugt werden	110
Eigenschaft hat keinen Wert Fehler	123
Ein Assigns-Parameter kann keinen Standardwert haben Fehler	124
Ein Erweiterter Parameter kann keinen Standardwert haben Fehler	123
Ein Interface dieser Klasse ist nicht vom Typ Klassen-Interface Fehler	124
Ein ParamArray kann keine Arrays enthalten Fehler	124
Ein Parameter nach einem Parameter-Array Fehler	124
Ein Parameter-Array kann keinen Standardwert haben Fehler	124
Ein Referenz-Parameter kann keinen Standardwert haben Fehler	125
Eine Methode mit einem Parameter-Array, kann keine optionalen Paramet haben Fehler	124
Es gibt keine Klasse mit diesem Namen Fehler	131

Exception Objekte müssen vom Typ RuntimeException sein Fehler	134
Extends kann nur auf den ersten Parameter angewendet werden Fehler	136
Externe Funktionen können Objekte nicht als Parameter verwenden	136
Externe Funktionen können String-Datentypen nicht als Parameter verwenden	137
Falscher Typ	137
Für die Deklaration von Bibliotheken können nur String-Konstanten verwendet werden Fehler	154, 265
GOTO Ziel nicht gefunden Fehler	172
Kann dieser Eigenschaft keinen Wert zuweisen Fehler	196
Logische Operationen benötigen boole'sche Werte Fehler	225
Me kann nicht in einer Methode oder einem Modul verwendet werden Fehler	229
Mit #if können nur Konstanten vom Typ Boolean verwendet werden Fehler	238
Parameter sind nicht zu dieser Funktion kompatibel Fehler	279
Parameter und Standardwert müssen vom selben Typ sein Fehler	279
Self kann nicht in der Methode eines Modules verwendet werden Fehler	366
Sie haben einen Operator verwendet, der nicht zu den spezifizierten Datentypen kompatibel ist Fehler	376
Sie können den New-Operator nicht mit dieser Klasse verwenden Fehler	376
Sie können diesen Variablen- oder Eigenschaften-Namen nicht verwenden, da es ein reserviertes Wort ist Fehler	377
Sie können ein Array nicht an eine externe Funktion übergeben Fehler	377
Sie können ein Array nicht einem anderen Array zuweisen Fehler	377
Sie können einen Ausdruck nicht als Parameter, der als ByRef definiert ist, übergeben Fehler	377
Sie können keinen Wert zurückgeben, da für diese Methode kein Rückgabetyt definiert wurde Fehler	378
Sie können nicht ein nichtexistierendes Event implementieren Fehler	378
Sie können Super nicht verwenden, da diese Klasse keine übergeordnete Klasse besitzt Fehler	378
Sie müssen den Datentyp des Rückgabewertes angeben Fehler	378
Super kann nicht in einer Modul-Methode verwendet werden Fehler	409
Syntax Error Fehler	409
Unbekanntes Pragma Fehler	438
Vererbung von einer Klasse ist nicht möglich Fehler	443

Fenster

ClearFocus Methode	58
MDIWindow Klasse	228
MessageDialog Klasse	233
MessageDialogButton Klasse	235
MsgBox(message,[buttons],[title]) → Integer	243
Window Funktion	446
Window Klasse	446
WindowCount → Integer	454

Fonts

Font(index) → String	149
FontCount → Integer	150
FontsFolder → FolderItem	150

Grafik

&cRRGGBB → Color	9
ArcShape Klasse	35
Canvas Steuerelement	53
CMY(cyan, magenta, yellow) → Color	60
Color Datentyp	61
CurveShape Klasse	72
DarkBevelColor → Color	73
DarkTingeColor → Color	73
ExportPicture (pic) → Boolean	134
FigureShape Klasse	137
FillColor → Color	138

FrameColor → Color	154
Graphics Klasse	173
Group2D Klasse	176
HighlightColor → Color	179
HSV(hue, saturation, value) → Color	179
ImageWell Steuerelement	186
LightBevelColor → Color	206
LightTingeColor → Color	206
Movie Klasse	239
MoviePlayer Steuerelement	240
NewPicture(width, height, depth) → Picture	250
Object2D Klasse	257
OpenPrinter(pageSetup) → Graphics	270
OpenPrinterDialog(pageSetup) → Graphics	270
Oval Steuerelement	276
OvalShape Klasse	276
Picture Klasse	280
PixmapShape Klasse	281
RectShape Klasse	334
Rgb(red, green, blue) → Color	351
RGBSurface Objekt	351
RoundRectShape Klasse	354
SelectColor (ByRef color, prompt) → Boolean	364
Sprite Klasse	388
SpriteSurface Steuerelement	389
StringShape Klasse	402
TextColor → Color	423

Internet

Datagram Klasse	88
DecodeBase64(str) → String	93
EmailAttachment Klasse	125
EmailHeaders Klasse	126
EmailMessage Klasse	127
EncodeBase64(str) → String	128
HTTPSecureSocket Klasse	182
HTTPSocket Klasse	180
InternetHeaders Klasse	190
NetworkInterface Objekt	246
POP3SecureSocket Klasse	284
POP3Socket Klasse	286
ShowURL Methode	374
SMTPSecureSocket Klasse	381
SMTPSocket Klasse	381
SocketCore Klasse	383
SSLSocket	393
TCPSocket Steuerelement	420
UDPSocket Klasse	436

Klassen

AddressBook	17
AddressBookAddress	18
AddressBookContact	19, 20
AddressBookData	21
AddressBookGroup	21
AddressBookRecord	23

ADSP4DServer	24
AppleEvent	26
AppleEventDescList	29
AppleEventObjectSpecifier	29
AppleEventRecord	30
AppleEventTarget	31
AppleTemplate	31
Application	32
ArcShape	35
AutoDiscovery	40
BevelButton	42
BinaryStream	45
Bitwise	47
Canvas	53
ChasingArrows	56
Checkbox	56
Clipboard	58
Collection	60
ComboBox	62
ConsoleApplication	64
ContextualMenu	65
Control	66
CriticalSection	71
CurveShape	72
DataAvailableProvider	74
Database	74
DatabaseCursor	81
DatabaseCursorField	81
DatabaseField	81
DatabaseQuery Steuerelement	82
DatabaseRecord	83
DataControl Steuerelement	84
Datagram	88
DataNotificationReceiver	88
DataNotifier Interface Klasse	88
Date	88
Dictionary	96
DisclosureTriangle Steuerelement	106
DragItem	108
EasyTCPSocket	110
EasyUDPSocket	112
EditableMovie	114
EditField Steuerelement	117
EmailAttachment	125
EmailHeaders	126
EmailMessage	127
EndOffline	130
ExcelApplication	131
FigureShape	137
FolderItem	139
FolderItemDialog	147
GameInputElement	156
GameInputManager	156
Graphics	173
Group2D	176
Group3D	177

GroupBox Steuerelement	177
HTTPSecureSocket	182
HTTPSocket	180
ImageWell Steuerelement	186
InternetHeaders	190
IPCsocket	190
IsA Operator	193
KeyChain	198
KeyChainItem	201
Line Steuerelement	206
ListBox Steuerelement	207
ListColumn	223
ListSelectionNotificationReceiver	224
ListSelectionNotifier	225
LittleArrows Steuerelement	225
MD5Digest	227
MDIWindow	228
MemoryBlock	229
MenuItem	231
MouseCursor	238
Movie	239
MoviePlayer Steuerelement	240
MySQLDatabase	245
New Operator	247
NewMemoryBlock(size) ➔ MemoryBlock	250
NotePlayer Steuerelement	254
Object	257
Object2D	257
Object3D	258
ODBCDatabase	260
Office	261
OLEContainer	263
OLEObject	264
OLEParameter	265
OpenBaseDatabase	265
OpenDialog	267
OracleDatabase	274
Oval Steuerelement	276
OvalShape	276
PagePanel Steuerelement	277
Paragraph	277
Picture	280
PixmapShape	281
Placard Steuerelement	282
POP3SecureSocket	284
POP3Socket	286
PopupArrow Steuerelement	287
PopupMenu Steuerelement	288
PostgreSQLDatabase	290
PowerPointApplication Klasse	291
PrefsMenuItem	292
PrinterSetup	293
ProgressBar Steuerelement	295
Pushbutton Steuerelement	296
QT3DAudio	298
QTEffect	299

QTEffectSequence	299
QTGraphicsExporter	299
QTSoundTrack	301
QTTrack	302
QTUserData	303
QTVideoTrack	304
Quaternion	306
QuitMenuItem	308
RadioButton Klasse	309
Random	310
Range	311
RBScript Steuerelement	316
REALDatabase	324
REALDatabaseOldFormat	327
RecordSet	327
Rectangle Steuerelement	329
RectControl	330
RectShape	334
RegEx	336
RegExMatch	341
RegExOptions	341
RegistryItem	343
ResourceFork	348
RoundRectangle Steuerelement	354
RoundRectShape	354
RuntimeException	356
SaveAsDialog	358
Screen	359
Scrollbar Steuerelement	361
SelectFolderDialog	365
Self	366
Semaphore	366
Separator Steuerelement	367
Serial Steuerelement	368
SerialPort	370
ServerSocket	371
ServiceApplication	372
Shell	373
Slider Steuerelement	380
SMTPSecureSocket	381
SMTPSocket	381
Socket Steuerelement	383
SocketCore	383
Sound	386
Sprite	388
SpriteSurface Steuerelement	389
SSLSocket Steuerelement	393
StandardInputStream	395
StandardOutputStream	396
StaticText Steuerelement	397
StringProvider	402
StringShape	402
StyledText	403
StyledTextPrinter Klasse	406
StyleRun	407
TabPanel Steuerelement	415

TCPIP4DServer	419
TCPSocket	420
TextConverter	424
TextEncoding	424
TextInputStream	426
TextOutputStream	427
Thread	428
Timer Steuerelement	430
UDPSocket	436
Vector3D	442
VirtualVolume	443
Window	446
WordApplication	454
XMLAttribute	454
XMLAttributeList	455
XMLCDATASection	455
XMLComment	455
XMLContentModel	456
XMLDocument	457
XMLDOMException	459
XMLElement	460
XMLException	460
XMLNode	461
XMLNodeList	462
XMLProcessingInstruction	463
XMLReader	463
XMLReaderException	466
XMLStyleSheet	467
XMLTextNode	467
XMLXslHandler	467

Konsolen-Applikation

ConsoleApplication Klasse	64
Input → String	187
Print Methode	293
ServiceApplication Klasse	372
StandardInputStream Klassen	395
StandardOutputStream Klasse	396
StdErr → StandardOutputStream	399
StdIn → StandardInputStream	399
StdOut → StandardOutputStream	400
TargetHasGUI → Boolean	417

Konstanten

DebugBuild → Boolean	90
RBVersion → Double	323
RBVersionString → String	324
Target68K → Boolean	417
TargetCarbon → Boolean	417
TargetHasGUI → Boolean	417
TargetLinux → Boolean	418
TargetMachO → Boolean	418
TargetMacOS → Boolean	418
TargetPPC → Boolean	419
TargetWin32 → Boolean	419

Laufzeitfehler

Exception Block	132
InvalidParentException Fehler	187
KeyChainException Fehler	200
KeyNotFoundException Fehler	203
NilObjectException Fehler	252
OLEException Fehler	264
OutOfBoundsException Fehler	274
OutOfMemoryException Fehler	275
RbScriptAlreadyRunningException Fehler	323
RegexSearchPatternException Fehler	342
RegexException Fehler	341
RegistryAccessErrorException Fehler	343
ShellNotRunningException Fehler	374
StackOverflowException Fehler	395
Try Block	434
UnsupportedFormatException	438
XMLDOMException Fehler	459
XMLException Fehler	460
XMLReaderException Klasse	466

Mathematisch

Abs(Wert) → Double	16
Acos(Wert) → Double	17
AscB(String) → Integer	38
Asin(Wert) → Double	38
Atan(Wert) → Double	39
Atan2(Wert) → Double	40
Bin(Wert) → String	45
CDbl(string) → Double	55
Ciel(Wert) → Double	55
Cos(Wert) → Double	70
Exp(Wert) → Double	134
Floor(Wert) → Double	139
Hex(Wert) → String	179
Log(Wert) → Double	225
Max(Wert1, Wert2) → Double	226
Min(Wert1, Wert2) → Double	237
Mod Operator	238
Oct(Wert) → String	260
Pow(value, power) → Double	291
Random Klasse	310
Rnd → Double	353
Round(Wert) → Double	353
Sign(value) → Integer	378
Sin(Wert) → Double	379
Sqrt(Wert) → Double	393
Tan(Wert) → Double	416
Val(string) → Double	439

Menüs

IsCMMClick → Boolean	195
MenuItem Klasse	231
PopupMenu Steuerelement	288
PrefsMenuItem Klasse	292

QuitMenuItem Klasse	308

Monitore/Bildschirme

Screen	360
Screen Klasse	359
ScreenCount → Integer	360

Multithreading

CriticalSection Klasse	71
Semaphore Klasse	366

Netzwerk

AutoDiscovery Klasse	40
Datagram Klasse	88
EasyTCPSocket Klasse	110
EasyUDPSocket Klasse	112
EmailAttachment Klasse	125
EmailHeaders Klasse	126
EmailMessage Klasse	127
HTTPSecureSocket Klasse	182
HTTPSocket Klasse	180
IPCSocket Klasse	190
NetworkInterface Objekt	246
POP3SecureSocket Klasse	284
POP3Socket Klasse	286
ServerSocket Klasse	371
ShowURL Methode	374
SMTPSecureSocket Klasse	381
SMTPSocket Klasse	381
Socket Klasse	383
SocketCore Klasse	383
SSLSocket Steuerelement	393
TCPSocket Steuerelement	420
UDPSocket Klasse	436

Objekte

App	25
Keyboard	196
Me Funktion	229
New Operator	247
Self	366
System	411

Objektverknüpfung

DataAvailableProvider	74
DataNotificationReceiver Interface Klasse	88
DataNotifier Interface Klasse	88
ListSelectionNotificationReceiver Interface Klasse	224
ListSelectionNotifier Interface Klasse	225
StringProvider Interface Klasse	402

Office Automatisierung

ExcelApplication Klasse	131
Office Klasse	261
PowerPointApplication Klasse	291

WordApplication Klasse	454

OLE

OLE Parameter Klasse	265
OLEContainer Klasse	263
OLEException	264
OLEObject Klasse	264

Operatoren

.....	12, 12, 13, 15
-	6
&	8
&cRRGGBB → Color	9
*	10
+	10
/	11
=	13
>	14
>=	15
And	25
Bitwise Klasse	47
BitwiseAnd(value1, value2) → Integer	48
BitwiseOr(Wert1, Wert2) → Double	49
BitwiseXor(Wert1, Wert2) → Double	49
Is	193
IsA	193
Mod	238
New	247
Not	253
Operator_ Schlüsselwort	271
Or	273

OS

#If...#Else...#Endif	6
AddressOf Operator	24
AppleMenuFolder → FolderItem	31
Application Klasse	32
ApplicationSupportFolder → FolderItem	35
Beep	41
Clipboard Klasse	58
ControlPanelsFolder → FolderItem	69
DebugBuild → Boolean	90
Declare Befehl	91
DesktopFolder → FolderItem	95
DocumentsFolder → FolderItem	106
EndOffline Klasse	130
ExtensionsFolder → FolderItem	136
Font(index) → String	149
FontCount → Integer	150
FontsFolder → FolderItem	150
GetOpenFolderItem(Filter) → FolderItem	161
GetSaveFolderItem(filter, default file name) → FolderItem	168
GetTemporaryFolderItem → FolderItem	169
HighlightColor → Color	179
Keyboard	196
MDIWindow Klasse	228

MemoryBlock Klasse	229
Microseconds → Double	235
MouseCursor Klasse	238
NewDragItem (Left, Top, Width, Height) → DragItem	249
OpenPrinterDialog(pageSetup) → Graphics	270
PreferencesFolder → FolderItem	291
Quicktime	297
RBVersion → Double	323
RBVersionString → String	324
ResourceFork Klasse	348
Screen	360
ScreenCount → Integer	360
SelectFolder → FolderItem	364
Serial Steuerelement	368
SerialPort Klasse	370
Shell Klasse	373
ShowURL Methode	374
ShutDownItemsFolder → FolderItem	376
Speak Method	387
StartupItemsFolder → FolderItem	397
System Objekt	411
SystemFolder → FolderItem	414
Target68K → Boolean	417
TargetCarbon → Boolean	417
TargetHasGUI → Boolean	417
TargetLinux → Boolean	418
TargetMachO → Boolean	418
TargetMacOS → Boolean	418
TargetPPC → Boolean	419
TargetWin32 → Boolean	419
TemporaryFolder → FolderItem	423
Ticks → Integer	429
TrashFolder → FolderItem	433
UserCancelled → Boolean	439
Volume(volumeNumber) → FolderItem	444
VolumeCount → Integer	445

QuickTime

EditableMovie Klasse	114
GetQTCrossFadeEffect Funktion → QTEffect	163
GetQTGraphicsExporter Methode	163
GetQTSMPTEEEffect (ID) → QTEffect	164
OpenURLMovie → Movie	271
QT3DAudio Klasse	298
QTEffect Klasse	299
QTEffectSequence Klasse	299
QTGraphicsExporter Klasse	299
QTSoundTrack Klasse	301
QTTrack Klasse	302
QTUserData Klasse	303
QTVideoTrack Klasse	304

Registry-Einträge

RegistryAccessErrorException Fehler	343
RegistryItem Klasse	343

Reguläre Ausdrücke

RegEx Klasse	336
RegException Fehler	341
RegExMatch Klasse	341
RegExOptions Klasse	341
RegExSearchPatternException Fehler	342

Ressourcen

ResourceFork Klasse	348

Schlüsselbund

KeyChain Klasse	198
KeyChainException Fehler	200
KeyChainItem Klasse	201

Sound

Beep	41
Movie Klasse	239
MoviePlayer Steuerelement	240
NotePlayer Steuerelement	254
Sound Klasse	386
Speak Methode	387

Spielsteuerung

GameInputElement Klasse	156
GameInputManager Klasse	156

Steuerelemente

BevelButton	42
Canvas	53
ChasingArrows	56
Checkbox	56
ClearFocus Methode	58
ComboBox	62
ContextualMenu	65
DatabaseQuery	82
DataControl	84
DisclosureTriangle	106
EditField	117
GroupBox	177
ImageWell	186
IsCMMClick → Boolean	195
Line	206
ListBox	207
LittleArrows	225
NotePlayer	254
OLEContainer	263
Oval	276
PagePanel	277
Placard	282
PopupArrow	287
PopupMenu	288
ProgressBar	295
Pushbutton	296
RadioButton	309

RB3DSpace	312
RBScript	316
Rectangle	329
RectControl	330
RoundRectangle	354
Scrollbar	361
Separator	367
Serial	368
Slider	380
SpriteSurface	389
SSLSocket	393
StandardToolbarItem	396
StaticText	397
TabPanel	415
Timer	430
ToolbarItem	431

Stiltext

Paragraph Klasse	277
Range Klasse	311
StyledText Klasse	403
StyleRun Klasse	407

Strings

.....	12, 12, 13
=	13
>	14
>=	15
Asc(string) → Integer	37
CDBl(string) → Double	55
Chr(Wert) → String	57
ChrB(value) → String	58
CountFields(Quelle,Separator) → Integer	70
CStr(Wert) → String	71
EndOfLine Klasse	130
Format(Zahl, formatSpec) → String	153
InStr([start], source, find) → Integer	188
InStrB([start], source, find) → Integer	189
Left(source, count) → String	203
LeftB(source, count) → String	204
Len(string) → Integer	204
LenB(string) → Integer	205
Lowercase(Wert) → String	226
LTrim(sourceString) → String	226
Mid(source, start, [length]) → String	236
MidB(source, start, [length]) → String	237
NthField(source, separator, fieldNumber) → String	256
Replace(sourceString, oldString, newString) → String	345
ReplaceAll(sourceString, oldString, newString) → String	345
ReplaceAllB(sourceString, oldString, newString) → String	346
ReplaceB(sourceString, oldString, newString) → String	347
ReplaceLineEndings(sourceString, LineEnding) → String	347
Right(source, count) → String	352
RightB(source, count) → String	352
RTrim(sourceString) → String	355
Split(source, delimiter) → String array	388

Split(sourceArray, delimiter) → String	196
Str(number) → String	400
StrComp(string1, string2, mode) → Integer	401
String Datentyp	401
StringShape Klasse	402
Titlecase(Wert) → String	431
Trim(sourceString) → String	433
Uppercase(Wert) → String	438
Val(string) → Double	439

Textcodierung

ConvertEncoding(str, newEncoding) → String	69
DecodeQuotedPrintable(str) → String	94
DefineEncoding(str, enc) → String	94
EncodeQuotedPrintable(str) → String	128
Encoding(str) → TextEncoding	129
Encodings object	129
GetFontTextEncoding (FontName) → TextEncoding	159
GetInternetTextEncoding (InternetEncoding) → TextEncoding	160
GetTextConverter(inputEncoding, outputEncoding) → TextConverter	170
GetTextEncoding (Base, Variant, Format) → TextEncoding	170
GuessJapaneseEncoding → TextEncoding	178
MD5(str) → String	227
MD5Digest Klasse	227
TextConverter Klasse	424
TextEncoding Klasse	424

Toolbars

StandardToolbarItem Steuerelement	396
ToolbarItem Steuerelement	431

Variablen

Append	26
Const Befehl	65
Declare Befehl	91
Dim Befehl	104
Pop	283
VarType (Wert) → Integer	441

Vektorgrafik

ArcShape Klasse	35
CurveShape Klasse	72
FigureShape Klasse	137
Group2D Klasse	176
Object2D Klasse	257
OvalShape Klasse	276
PixmapShape Klasse	281
RectShape Klasse	334
RoundRectShape Klasse	354
StringShape Class	402

XML

XMLAttribute Klasse	454
XMLAttributeList Klasse	455
XMLCDATASection Klasse	455

XMLComment Klasse	455
XMLContentModel Klassen	456
XMLDocument Klasse	457
XMLDOMException Klasse.	459
XMLElement Klasse	460
XMLException Klasse	460
XMLNode Klasse	461
XMLNodeList Klasse	462
XMLNodeType Object	462
XMLProcessingInstruction Klasse	463
XMLReader Klasse.	463
XMLReaderException Klasse	466
XMLStyleSheet Klasse.	467
XMLTextNode Klasse	467
XMLXslHandler Klasse	467