

Nützliche Klassen

Für MS-Access Entwickler

Armin Hess

1.1 vom 16.06.2013

Inhalt

Versionshinweise	2
1. Die Klassen	3
1.1. Allgemeine Eigenschaften der Klassen	3
1.1.1. Eigenschaften (Properties)	3
2. clsParas	4
3. clsWhere	5
4. clsSort	9
5. clsMeter	10
6. clsExcel	12
7. Nutzung	13

Versionshinweise

Version	Datum	Was ist neu oder anders?
1.0	03.02.2011	clsParas, clsWhere, clsSort
1.1		clsMeter, clsExcel

1. Die Klassen

1.1. Allgemeine Eigenschaften der Klassen

Alle Klassen des Frameworks haben einige Verwaltungseigenschaften, die im Folgenden beschrieben werden.

Bei allen Beschreibungen von Eigenschaften steht in Klammer der Datentyp der Eigenschaft. Dahinter ist vermerkt ob die Eigenschaften gelesen oder geschrieben werden kann.

R	Die Eigenschaft kann nur gelesen werden,
W	Die Eigenschaft kann nur geschrieben werden.
R/W	Die Eigenschaft kann sowohl gelesen, als auch geschrieben werden.

1.1.1. Eigenschaften (Properties)

<i>className</i>	Liefert den Namen der Klasse.(String, R)
<i>classVersion</i>	Liefert die Version der Klasse.(String, R)
<i>classLastChanged</i>	Liefert den Zeitpunkt der letzten Änderung der Klasse. .(String, R)

2. clsParas

Diese Klasse erlaubt es mehrere Texte zu verketteten und wieder zu zerlegen. Als Trennzeichen wird standardmäßig das Pipezeichen (|) verwendet. Sollte das Trennzeichen im text vorkommen, wählt die Klasse einen anderen Trenner. Der gewählte Trenner wird dem String vorangestellt.

Funktionen

Add strText

strText Der Text wird zum Parameterstring hinzugefügt.

Eigenschaften (Properties)

asString liefert den Zusammenhängenden String (R)
 asString= der zusammenhängende String wird der Klasse übergeben und in seine Bestandteile zerlegt. (W)
 Count Anzahl der Parameter (R, long)
 Item(Index) liefert den durch Index adressierten String. Die Zählung beginnt bei 1. (R, string)

Beispiel

```
Dim cArgs as clsParas
Set cArgs=new clsParas
cArgs.add "Parameter1"
cArgs.add "Parameter2"
doCmd.openform "Formular",,,,cArgs.asString
Im Formular kann dann folgendermaßen darauf zugegriffen werden.
Dim cArgs as clsParas
Set cArgs=new clsParas
cArgs.asString=me.openargs()
debug.print cargs.item(1) ' liefert Parameter1
```

3. clsWhere

Eine Klasse zum Erzeugen von Where-Klauseln. Dabei werden dem Anwender eine ganze Reihe von Komfortfunktionen zur Verfügung gestellt. Der allgemeine Aufruf lautet:

```
Add<Type> strField, ctrControl, [strDescription]
```

strField	Der Name des Feldes, nach dem gesucht werden soll in der Tabelle oder der Abfrage.
Control	Ein Control, in das der Anwender das Suchkriterium eintragen kann. Die Klasse prüft die Gültigkeit und färbt im Fehlerfall das Feld ein.
strDescription	Optionalen Parameter, bei dem die Bezeichnung, die im Beschreibungsstring verwendet wird, angegeben werden kann. Normalerweise ermittelt die clsWhere die Bezeichnung des Controls aus dem davor stehenden Text. Sollte dies nicht funktionieren oder ein anderer Text gewünscht sein, kann er hier angegeben werden.

Diese Funktion gibt es für verschiedene Datentypen. Bei den verschiedenen Typen werden verschiedene Funktionalitäten zur Verfügung gestellt. Es gilt jeweils folgende Syntax.

Text

Eingabe in der Textbox	Erzeugte Where-Klausel	Hinweis
text	LIKE '*text*'	Es wird grundsätzlich eine Teilstringsuche ausgeführt.
-text	NOT LIKE '*text*'	
text*	Like 'text*'	Fügt der Anwender Jokerzeichen ein, werden vom System keine mehr hinzugefügt.
'text'	'text'	Wird der Suchtext in Hochkommas oder Anführungszeichen eingeschlossen, wird nach dem exakten Text gesucht.
Null	Is null	Durch Eingabe des Textes Null wird nach leeren Feldern gesucht.

Date

Eingabe in der Textbox	Erzeugte Where-Klausel	Hinweis
3.2.2011	=#2/3/2011#	
>3.2.2011	>#2/3/2011#	Alle Operatoren (>, <, >=, <=)

		<= und <>) sind zugelassen.
24.12.10-31.12.10	BETWEEN #12/24/2010 AND #12/31/2010#	
2010	BETWEEN #1/1/2010# AND #12/31/2010#	Suche nach einem Jahr.
3/2010	BETWEEN #3/1/2010# AND #3/31/2010#	Suche nach einem Monat
2009-2010	BETWEEN #1/1/2009# AND #12/31/2010#	Suche nach mehreren Jahren. Analog kann auch mit Monaten oder exaktem Datum verfahren werden.
KW5	BETWEEN #1/31/2011# AND #2/6/2011#	Suche nach Kalenderwochen im aktuellen Jahr
KW5/2010	BETWEEN #02/01/2010# AND #02/07/2010#	Suche nach Kalenderwochen in einem anderen Jahr. Bei KW-Angaben können natürlich auch alle Operatoren und das Minus (für zwischen) verwendet werden.
Null	Is null	Durch Eingabe des Textes Null wird nach leeren Feldern gesucht.
*	Is not null	Suche nach Feldern, die nicht leer sind.

Number

Eingabe in der Textbox	Erzeugte Where-Klausel	Hinweis
123	=123	
>123	>123	Alle Operatoren (>, <, >=, <= und <>) sind zugelassen.
123-234	BETWEEN 123 AND 234	
Null	Is null	Durch Eingabe des Textes Null wird nach leeren Feldern gesucht.
*	Is not null	Suche nach Feldern, die nicht leer sind.

```
addBool strField, ctrControl, [strDescription]
```

strField

Der Name des Feldes, nach dem in der Abfrage gesucht werden soll.

Control	Eine Checkbox, in dem der Anwender das Suchkriterium anklicken kann.
strDescription	Optionalen Parameter, bei dem die Bezeichnung, die im Beschreibungsstring verwendet wird, angegeben werden kann. Normalerweise ermittelt die clsWhere die Bezeichnung des Controlls aus dem davor stehenden Text. Sollte dies nicht funktionieren oder ein anderer Text gewünscht sein, kann er hier angegeben werden.

addRefx strField, ctrControl, [strDescription],[fNegation]

strField	Der Name des Feldes, nach dem in der Abfrage gesucht werden soll.
Control	Ein Combobox, in dem der Anwender das Suchkriterium auswählen kann.
strDescription	Optionalen Parameter, bei dem die Bezeichnung, die im Beschreibungsstring verwendet wird, angegeben werden kann. Normalerweise ermittelt die clsWhere die Bezeichnung des Controlls aus dem davor stehenden Text. Sollte dies nicht funktionieren oder ein anderer Text gewünscht sein, kann er hier angegeben werden.
fNegation	Boolscher Wert, der angibt ob die Abfrage verneint werden soll.

addSpez strText

strText	Text, der in die Where-Klausel eingefügt wird. Die über Bool gewählte Art wird vorangestellt.
---------	---

addParenthesisLeft

Eine öffnende Klammer wird in die Where-Klausel eingefügt.

addParenthesisRight

Eine schließende Klammer wird in die Where-Klausel eingefügt.

Sollte auf diese Weise ein leeres Klammerpaar entstehen, wird nichts eingefügt.

Run frmForm

frmForm	Formular, auf das die Where Klausel ausgeführt werden soll. Dies geschieht durch Setzen der Filter Eigenschaft des Formulars. Wenn die Query Eigenschaft von clsWhere gesetzt wurde, wird die Where-Klausel in die Abfrage integriert und die Recordsource Eigenschaft des Formulars gesetzt.
---------	---

Eigenschaften

Query	Name einer vordefinierten Abfrage. Die erzeugte Where-Klausel wird eingefügt und die Funktion Run weißt sie so erhaltene Abfrage der Eigenschaft Recordsource zu. (W)
Bool	legt fest ob die einzelnen Komponenten mit AND oder OR verknüpft werden sollen. Dazu stehen die Enums <i>ahwAND</i> und <i>ahwOR</i> zur Verfügung. (RW)

Count	Anzahl der Felder, die in die Where-Klausel aufgenommen wurden (R)
ErrorColor	Longwert der Farbe, die zum Markieren von Fehleingaben. Default ist 255 (rot) (RW)
forceNoCase	Bool, wenn True wird Lcase bei Textabfragen eingefügt. Dies ist notwendig, wenn das Backend Groß- und Kleinschreibung unterscheidet. (z.B. Oracle über ODBC) (RW)
as String	die Where-Klausel als String (R)
Description	Beschreibung der Where-Klausel in Klartext (R)
Verbose	Boolscher Wert, der festlegt ob die Beschreibung in die Statuszeile geschrieben werden soll (RW)
forDebug	Boolscher Wert. Ist er TRUE, wird, wenn Verbose gesetzt ist, die Where-Klausel anstelle der Beschreibung ausgegeben. (RW)
wildcard	Longwert, der bei addRefx als "alle" interpretiert wird. (RW)

Ein Beispiel

```
Sub runQuery()
```

```
Dim cWhere As clsWhere
```

```
' Klasse deklarieren
```

```
Set cWhere = new clsWhere
```

```
' Klasse instanzieren
```

```
cWhere.addText "strName", Me.txtNameReq
```

```
' Text suchen
```

```
cWhere.addNumber "lngNumber", Me.txtNumberReq
```

```
' Zahlen suchen
```

```
cWhere.addRefx "lngProjectID", Me.cboProjectIDreq
```

```
' Comboboxen
```

```
cWhere.addDate "datDatum", Me.txtDateReq
```

```
' in Datums suchen
```

```
cWhere.verbose=true
```

```
' Text in Statuszeile
```

```
cWhere.forDebug=true
```

```
' als SQL
```

```
cwhere.run me
```

```
' Ausführen
```

```
End Sub
```


4. clsSort

Klasse zum Sortieren von Spalten in Endlosformularen. Die Spalten müssen einen Button als Überschrift haben. In den onClick Event der Buttons wird die Funktion goSort der Formularglobalen Instanz von clsSort aufgerufen.

Beim Klicken wird die Spalte direkt unter dem Button aufsteigend sortiert. Ein weiterer Klick sortiert absteigend.

Funktion

```
goSort [ctr1][,ctr2][,...]
```

Wenn kein Parameter angegeben ist, wird die Spalte direkt unter dem Button sortiert. Wenn Parameter angegeben werden, wird nach den Feldern, an die die Controls gebunden sind sortiert.

Ein Beispiel:

```
Private cSort as clsSort      ' Globales Objekt
Private Sub Form_Open()
    set cSort = new clsSort    ' Sortierobjekt instanzieren
End Sub

Private Sub cmdButton1_Click()
    cSort.goSort              ' sortiere nach dem Feld direkt
                             ' unter dem Button
End Sub

Private Sub cmdButton2_Click()
    cSort.goSort me.txt1,me.txt2 ' sortiere nach txt1 und txt2
End Sub

Private mcSort as clsSort
Private Function cSort() as clsSort
    if mcSort is nothing then
        set mcSort = new clsSort
    End If
    set cSort = mcSort
end Function
```

Leider verliert Access bei einer Unterbrechung (durch einen Laufzeitfehler, oder Umschalten in den Entwurfmodus globale Klassen. Dem kann man entgegenwirken, indem man für die globale clsSort Instanz eine kleine Funktion verwendet, die das Objekt gegebenenfalls wieder neu instanziiert. Der Aufruf des Sortierobjekt bleibt dabei völlig gleich.

5. clsMeter

Klasse, die einen Fortschrittsbalken zur Verfügung stellt

Funktionen

Init strTitle, lngHighWater

strTitle	Titel des Fortschrittbalkens
lngHighWater	Der Wert, dem 100% des Balkens entsprechen soll.

Update [lngValue][,strText]

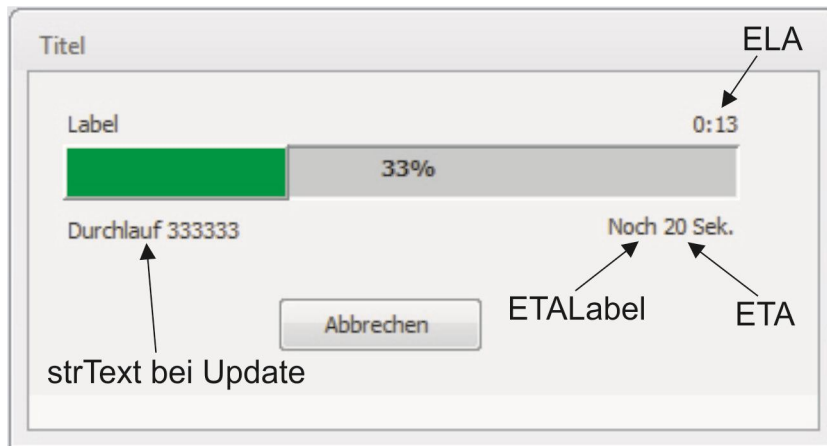
lngValue	Wert auf den der Balken gesetzt wird. Wenn lngValue nicht angegeben wird, wird der Wert vom letzten Aufruf um eins erhöht.
strText	Text der unterhalb des Balkens ausgegeben wird.
Update	Die Funktion liefert FALSE, wenn der Anwender den Abbruch Knopf gedrückt hat, ansonsten TRUE

Remove

Der Fortschrittsbalken verschwindet.

Eigenschaften

Mode	Art des Fortschrittbalkens. Mögliche Werte sind <i>ahmSystem</i> : Der Balken in der Statuszeile wird verwendet. Dann sind alle anderen Eigenschaften ohne Bedeutung <i>ahmFormAnalog</i> : Der Balken wird in einem Formular (frmMeter) dargestellt. (W)
Color	Farbe des Balkens (W)
Percent	Boolean, legt fest ob der Prozentuale Wert angezeigt werden soll. (W)
Title	Titel des Fensters (W)
Label	Bezeichnung des Balkens (W)
ETA	Boolean, legt fest ob die geschätzte Restzeit (estimated time of arrival) angezeigt werden soll. (W)
ETALabel	legt den Text vor der restzeit fest. (W)
ETADelay	legt fest nach wie vielen Sekunden die Restzeit angezeigt werden soll.
ELA	Boolean, legt fest ob die bisher verstrichene Zeit (Elapsed) angezeigt werden soll.
CancelButton	Boolean, legt fest of ein Abbrechenknopf angezeigt werden soll.



Fenster frmMeter mit der Positionierung der einzelnen Eigenschaften.

Ein Beispiel

```
Dim cMeter as clsMeter
```

```
Dim lngRows as long, lngRow as long
```

```
Set cMeter = new clsMeter
```

```
cMeter.init "Import", lngRows
```

```
For lngRow = 1 to lngRows
```

```
...
```

```
if not cMeter.update() then exit For ' Abbruch
```

```
Next
```

```
cMeter.remove
```

6. clsExcel

Eine Klasse zum Fernsteuern von Excel

Funktionen

getCell lngX, lngY

Liefert den Inhalt einer Zelle.

lngX Nummer der Spalte (A=1, B=2, usw.)

lngY Nummer der Zeile (beginnend bei 1)

getCell Inhalt der Zelle

setCell lngX, lngY, varNewValue

Schreibt einen Wert in eine Zelle.

lngX Nummer der Spalte (A=1, B=2, usw.)

lngY Nummer der Zeile (beginnend bei 1)

varNewValue Wert, der in die Zelle geschrieben werden soll.

deleteRows lngRow1[, lngRow2]

löscht Zeilen

lngRow1 Zeilennummer, ab der gelöscht wird

lngRow2 Zeilennummer, bis zu der gelöscht wird (optional, wenn dieser Parameter nicht angegeben wird, wird bis zum Ende der Datei gelöscht).

save [strFile]

Sichert die Exceldatei

strFile Dateiname unter dem gesichert wird (Optional)

Eigenschaften

file Dateiname, der geöffnet werden soll

lastRow Nummer der letzten verwendeten Zeile

worksheet Name des Tabellenblatts, das bearbeitet werden soll.

worksheetIndex Nummer des Tabellenblatts, das bearbeitet werden soll.

7. Nutzung

1. Die Klassen können frei, in beliebigen Anwendungen verwendet werden.
2. Der Autorenhinweis darf nicht entfernt werden.
3. Fehler und Anregungen bitte an armin@gaeuwetter.de
4. Die Programme werden so überlassen wie sie sind. Es wird keinerlei Garantie für ein bestimmtes Verhalten übernommen. Vor allem kann keine Garantie für fehlerfreiheit übernommen werden.