



db2000 measure conversion

© 2008 Massimo Mascalchi

massimo.mascalchi@db2000web.net



DESCRIZIONE DELLE FUNZIONI

ATTENZIONE: per l'utilizzo di queste funzioni è necessario disporre del file "**db2000 - measure units.dat**" (archivio delle unità di misura) oppure di un'altro ma avente la stessa struttura dei dati.

ConvertUnit

restituisce la conversione del valore/quantità indicato dall'unità di misura origine nell'unità di misura di conversione per il tipo di misura anch'esse indicate purché le due unità siano presenti nell'archivio delle unità di misura

```
sintassi:  r = fMC.ConvertUnit([tipo misura], _  
                                [unità di misura origine], _  
                                [valore/quantità unità di misura origine], _  
                                [unità di misura di conversione])
```

```
esempi:  r = fMC.ConvertUnit("5", "8", 10, "23") ' 0.1  
         r = fMC.ConvertUnit("length", "centimeter", 10, "meter") ' 0.1
```

CountConversionUnits

restituisce, per il tipo di misura indicato, il numero delle unità di conversione attualmente presenti nell'archivio

```
sintassi:  r = fMC.CountConversionUnits([tipo misura])
```

```
esempi:  r = fMC.CountConversionUnits("5")  
         r = fMC.CountConversionUnits("length")
```

CountMeasureCategories

restituisce il numero dei tipi di misura (categorie) attualmente presenti nell'archivio

```
sintassi:  r = fMC.CountMeasureCategories
```

```
esempio:  r = fMC.CountMeasureCategories
```

DataMeasureUnitsFileName

restituisce/imposta il nome del file contenente i dati necessari per effettuare le conversioni tra le varie unità di misura (archivio delle unità di misura)

```
sintassi:  r = fMC.DataMeasureUnitsFileName ' restituisce il nome del file  
         fMC.DataMeasureUnitsFileName = [nome del file] ' imposta il nome del file
```

```
esempi:  r = fMC.DataMeasureUnitsFileName ' restituisce il nome del file  
         fMC.DataMeasureUnitsFileName = AppPath & "\db2000 - measure units.dat"  
         ' imposta il nome del file
```



ExistConversionUnit

verifica nell'archivio l'esistenza dell'unità di misura indicata, restituisce **True** se trovata altrimenti **False**

sintassi: `r = fMC.ExistConversionUnit([tipo misura], [unità di misura])`

esempi: `r = fMC.ExistConversionUnit("5", "8")`

`r = fMC.ExistConversionUnit("length", "centimeter")`

ExistMeasureCategory

verifica nell'archivio l'esistenza del tipo di misura indicata, restituisce **True** se trovata altrimenti **False**

sintassi: `r = fMC.ExistMeasureCategory([tipo misura])`

esempi: `r = fMC.ExistMeasureCategory("5")`

`r = fMC.ExistMeasureCategory("length")`

New1

inizializza gli oggetti della libreria

sintassi: `fMC.New1`

esempio: `fMC.New1`

ReadConversionUnitParameters

restituisce in un array di stringhe i parametri di conversione dell'unità di misura indicata purché presente nell'archivio

sintassi: `ra() = fMC.ReadConversionUnitParameters([tipo misura], [unità di misura])`

l'array restituito è una matrice di 5 elementi:

(0)	=	ID dell'unità di misura
(1)	=	nome dell'unità di misura
(2)	=	simbolo o abbreviazione
(3)	=	rapporto del valore di conversione
(4)	=	rettifica del valore di conversione

esempi: `ra() = fMC.ReadConversionUnitParameters("5", "8")`

```
' se è stato dichiarato di usare come archivio dei dati necessari alle conversioni
' il file "db2000 - measure units.dat" verranno restituiti i seguenti parametri:
'
'                                     ra(0) = "8"
'                                     ra(1) = "centimeter"
'                                     ra(2) = "cm"
'                                     ra(3) = "0.01"
'                                     ra(4) = "0"
```

`ra() = fMC.ReadConversionUnitParameters("length", "centimeter")`

```
' se è stato dichiarato di usare come archivio dei dati necessari alle conversioni
' il file "db2000 - measure units.dat" si otterrà lo stesso risultato dell'esem-
' pio precedente...
```



ReadConversionUnitsIdentifiers

restituisce in un'array di stringhe le identificazioni di tutte le unità di conversione presenti nell'archivio per il tipo di misura e i dati richiesti indicati

```
sintassi:  ra() = fMC.ReadConversionUnitsIdentifiers([tipo di misura], [dati richiesti])

           [dati richiesti] =  (0) = ID delle unità di misura
                               (1) = nomi dell'unità di misura
```

```
esempi:   ra() = fMC.ReadConversionUnitsIdentifiers("5", 1)
           ra() = fMC.ReadConversionUnitsIdentifiers("length", 1)
```

ReadMeasureCategories

restituisce in un array di stringhe tutti gli identificativi del tipo di misura presenti nell'archivio per il tipo di campo richiesto

```
sintassi:  ra() = fMC.ReadMeasureCategories([dati richiesti])

           [dati richiesti] =  (0) = ID delle unità di misura
                               (1) = nomi dell'unità di misura
```

```
esempi:   ra() = fMC.ReadMeasureCategories(0)
           ra() = fMC.ReadMeasureCategories(1)
```

ReadMeasureCategory

se presente nell'archivio restituisce una stringa contenente l'identificativo della misura (categoria), se il tipo della misura indicato è l'identificativo numerico ritorna il nome della misura stessa; invece, se il tipo della misura indicato è il nome, ritorna l'identificativo numerico

```
sintassi:  r = fMC.ReadMeasureCategory([tipo di misura])

esempi:   r = fMC.ReadMeasureCategory("5")           ' "length"
           r = fMC.ReadMeasureCategory("length")     ' 5
```

RemoveConversionUnit

rimuove dall'archivio l'unità di misura indicata associata al tipo di misura indicato (categoria), ritorna **True** se l'operazione termina correttamente altrimenti **False**

```
sintassi:  r = fMC.RemoveConversionUnit([tipo di misura], [unità di misura])

esempi:   r = fMC.RemoveConversionUnit("5", "8")
           r = fMC.RemoveConversionUnit("length", "centimeter")
```

RemoveMeasureCategory

rimuove dall'archivio il tipo di misura indicato (categoria) e tutte le unità di misura associate, ritorna **True** se l'operazione termina correttamente altrimenti **False**

```
sintassi:  r = fMC.RemoveMeasureCategory([tipo di misura])

esempi:   r = fMC.RemoveMeasureCategory("5")
           r = fMC.RemoveMeasureCategory("length")
```



WriteConversionUnitParameters

aggiorna i parametri contenuti in un array dell'unità di misura indicata o ne inserisce una nuova, ritorna **True** se l'operazione termina correttamente altrimenti **False**

```
sintassi: r = fMC.WriteConversionUnitParameters([tipo di misura], _  
                                                [unità di misura], _  
                                                [array parametri unità di misura], _  
                                                [nuovo inserimento])
```

l'array dei parametri è una matrice di 5 elementi:

(0)	=	ID dell'unità di misura
(1)	=	nome dell'unità di misura
(2)	=	simbolo o abbreviazione
(3)	=	rapporto del valore di conversione
(4)	=	rettifica del valore di conversione

[nuovo inserimento] = False = aggiorna i dati dell'unità di misura
True = inserisce una nuova unità di misura

ATTENZIONE: quando si tratta di un nuovo inserimento è necessario indicare in **[unità di misura]** il valore corrispondente al primo elemento dell'array dei parametri (0) o al contenuto del secondo elemento (1) indifferentemente

```
esempi: r = fMC.WriteConversionUnitParameters("5", "23", ra(), False)  
r = fMC.WriteConversionUnitParameters("length", "meter", ra(), False)  
r = fMC.WriteConversionUnitParameters("5", "37", ra(), True)
```

WriteMeasureCategory

aggiorna i parametri del tipo di misura indicata (categoria) o ne inserisce una nuova, ritorna **True** se l'operazione termina correttamente altrimenti **False**

```
sintassi: r = fMC.WriteMeasureCategory([tipo di misura], _  
                                         [ID misura], _  
                                         [nome misura], _  
                                         [nuovo inserimento])
```

[nuovo inserimento] = False = aggiorna i dati dell'unità di misura
True = inserisce una nuova unità di misura

ATTENZIONE:

1. quando si tratta di un nuovo inserimento è necessario indicare in **[tipo di misura]** il valore corrispondente a **[ID misura]** o a **[nome misura]** indifferentemente
2. si consiglia di evitare di cambiare il nome ad un tipo di misura (categoria) alla quale sono state già assegnate delle unità di misura, quest'ultime, anche se ancora presenti nell'archivio, non saranno più rintracciabili

```
esempi: r = fMC.WriteMeasureCategory("5", "5", "length", False)  
r = fMC.WriteMeasureCategory("length", "5", "length", True)
```