

QWaccess ActiveX DLL

(December 11, 2009 v5.0.412)

QWaccess ActiveX DLL is a high level tool for accessing Quality Window databases. With a handful of methods, a user familiar with Visual Basic can easily retrieve all sorts of information stored in Quality Window templates, as well as data stored in Quality Window databases. To initialize the control, one must issue a statement similar to: **Dim QWfile As New Qwaccess**. Note that a reference to Quality Window Data Engine (qwaccess.dll or qwaccessSDK.dll) must be established inside a project.

After this point, QWfile becomes the container holding all of the control's internal properties. Any number of containers may be defined to access multiple databases.

Methods

CloseFile

Closes an opened Quality Window database and clears all collections
QWfile.**CloseFile**

CopyTemplate *QWfile*

Creates a clone of another open template (*QWfile*) into the current template. This is usually done when the file will be opened with a mode of FileCreate, and must be done prior to actually Opening the file.

GetNext

Reads the next non-deleted record in the database into the **Value** collection.
Sets EOF after last record is read.
QWfile.**GetNext**

GetNextSummarized

Reads the next grouping of records within the sorted list according to the **SummarizedFields** parameter, and sets the **Statistics** collection.

GetPrev

Reads the previous non-deleted record in the database into the **Value** collection.
Sets BOF after first record is read.
QWfile.**GetPrev**

GetRecord *RecordNo*

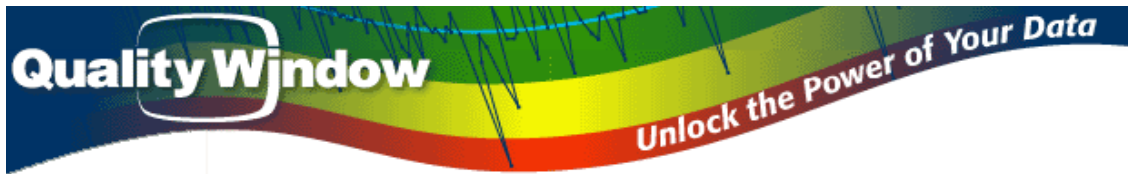
Reads a specific record in the database, denoted by *RecordNo*. If the record is deleted, then the Deleted property is also turned on. Each field in the record is then stored in the Value collection ie **Value(15)**
QWfile.**GetRecord** 127

InitRecord

Initializes the record buffer. Useful when creating a new record

InsertRecord

Writes a record into the database. Method determines automatically based on the Date/Time of the record if the process is an Insert or an Add to the database. RecordNo of the new record is set upon completion



OpenFile *Filename*

Opens a Quality Window database specified as *Filename*. The filename may contain path information, but need not specify any extension. The filename may also contain a network UNC prefix. All descriptive information related to the database are loaded into properties, and the database is positioned at the beginning of the file.

QWfile.**OpenFile** "\\Server\c\qwddata\monitor.qwt"

PutRecord *RecordNo*

Writes a record into the database at RecordNo position. If RecordNo is Zero, then the record is appended to the end of the file.

SeekFirst

Positions the Read pointer to just before the first record, allowing a **GetNext** to actually read the first record.

QWfile.**SeekFirst**

SeekLast

Positions the Read pointer to just after the last record, allowing a **GetPrev** to actually read the last record.

QWfile.**SeekLast**

SortFile

Sort all non-deleted records in the database based on the **SortFields** parameters.

Respects any **Filter** parameters specified.

Respects any **SummarizedFields** parameters specified.

Sets the **Sorted** collection

Statistics on the current recordset are calculated if **CalcStatistics** is True

QWfile.**SortFile**

CopyField *SourceField, TargetField*

Copies an entire Field construct to another. ie to make Field(10) look like Field(3) you would issue **CopyField .Fields(3), .Fields(10)**

LoadViewsDirectory

Loads all of the Views associated with the opened database into the **Views** collection

SaveViewsDirectory

Saves all of the Views in the **Views** collection to disk for later retrieval

ReCalcRecord *RecordNo*

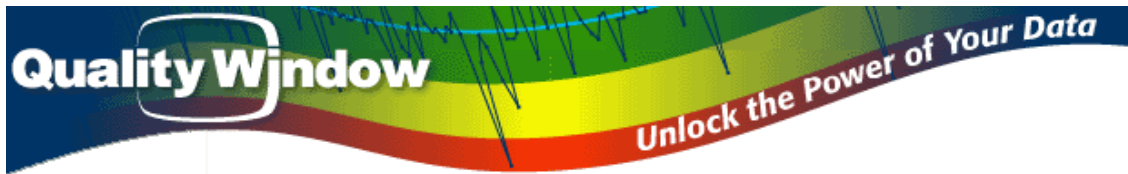
Recalculates a specific record in the database. All Lookup and Calculated fields are computed, but not yet written to the database.

SetupReCalc *RecordNo*

Creates a temporary buffer to handle @Moving type calculations so that subsequent **ReCalcRecord** calls are more efficient.

UpdateQWT

Writes all non-structural template changes back to the Opened template. Useful for quickly updating properties such as Limits



Properties

Title As String

Sets/Returns the Application Title defined in the template

Version As Double

Sets/Returns the version number of the Quality Window template ie 4.5

LockVar As Integer

Sets/Returns the number of variables to lock on the QW logsheet

History As Integer

Sets/Returns the number of records to initially load within QW

Xaxis As Integer

Sets/Returns the field number to use as the X-Axis

Annotate1 As Integer

Sets/Returns the field number used for the first annotation

Annotate2 As Integer

Sets/Returns the field number used for the second annotation

PreProcess As String

Sets/Returns the Batch file to execute when the 'Add' key is pressed

LogFile As String

Sets/Returns the application Log Filename defined in the template

PostProcess As String

Sets/Returns the Batch file to execute after saving a record

HelpFile As String

Sets/Returns the application Help Filename defined in the template

DefaultView As Integer

Sets/Returns the View to be used as default

LimitPercent As Double

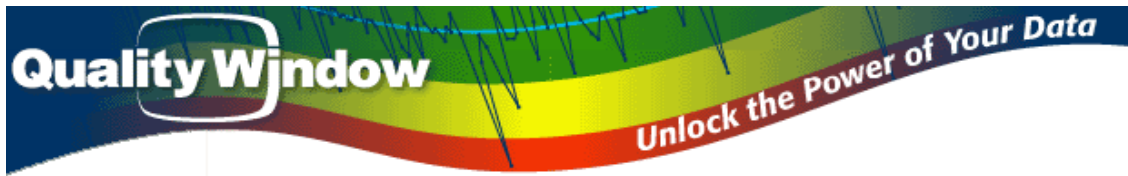
Sets/Returns the Percentage of points used to calculate some of the Limits.
Used in conjunction with **LimitMin** and **LimitMax**

LimitMin As Long

Sets/Returns the minimum number of points used to calculate some of the limits.

LimitMax As Long

Sets/Returns the maximum number of points used to calculate some of the limits.
In determining Limits Based on Low Variation, High Variation, Low Average and High Average the following formula is used:
 $SegmentSize = Population\ size * (LimitPercent / 100)$
If SegmentSize is less than LimitMin then SegmentSize = LimitMin
If SegmentSize is greater than LimitMax then SegmentSize = LimitMax



SecLevel As String

Sets/Returns the Security Level of a QW template
Can be a string of any of the following characters:
T, A, E, C, I, D, V

SecVariable As Integer

Sets/Returns the field number used to store the Security UserID

AttrUnits As Integer

Sets/Returns the variable number used for Attribute Units

AttrUnitsDesc As String

Sets/Returns the description for Units

AttrFreqDesc As String

Sets/Returns the description for Frequency

AttrWeight As Integer

Sets/Returns the variable used for Weight

AttrWeightDesc As String

Sets/Returns the description for Weight

AttrDowntime As Integer

Sets/Returns the variable used for Downtime

AttrNormal As Integer

Sets/Returns the variable used for Normalizing

AttrNormalDesc As Integer

Sets/Returns the description for Normalizing

AttrInterval As Double

Sets/Returns the Shift Interval in Hours

AttrIntervalTime As String

Sets/Returns the Start Time of an Interval (Shift)

AttrIntervalDesc As String

Sets/Returns the description of an Interval

Filename As String

Returns the currently opened Quality Window database without extension.

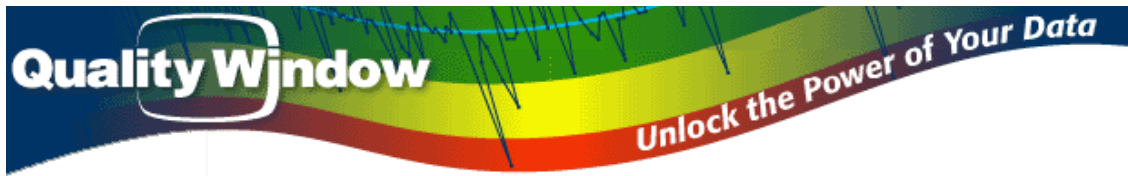
FilePath As String

Returns the full path of the currently opened Quality Window database

RuleFilename As String

Returns the full path and filename of the current Rules file.

GetSystemDir As String Returns the full path of the Windows System directory.



AutoNumber As Integer

Sets/Returns the variable number used to store the auto-generated sequence number. A file with the extension '.QWN' is used to maintain the next available sequence number. A new sequence number is only generated when the **Value** of the **AutoNumber** field is empty at the time of a **PutRecord**.

RecordCount As Long

Returns the total number of records in the database (including deleted records).

RecordLen As Long

Returns the Logical Record Length of the current database.

RecordNo As Long

Returns the current Record Number of record just read.

RecBuffer As String

Returns the current record buffer of the record just read or Sets the current record buffer to be written

OpenMode As Integer

Must be set prior to **OpenFile**. The following values can be specified:

0-**FileReadOnly** – Opens the database in a read only mode (default)

1-**FileCreate** – Creates a new template and database

2-**FileUpdate** – Opens the database in a read/write mode

FileExists As Boolean

Returns whether the QWT file actually exists. This should be checked just after an **OpenFile** is issued.

Deleted As Boolean

Sets/Returns whether the record just written/read is Deleted or not.

Filtered As Boolean

Returns whether the record just read satisfies the current **Filter** options.
Returns True if no Filtering defined.

EOF As Boolean

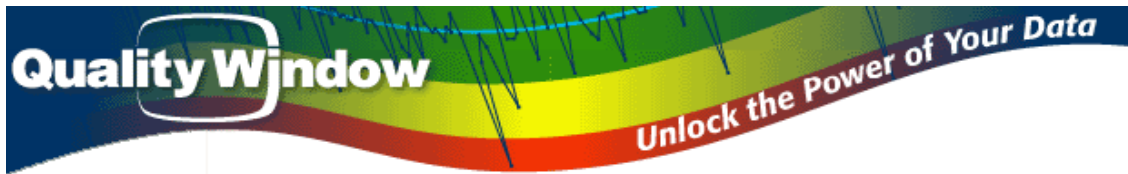
Returns whether the Read pointer is at the end of the database.

BOF As Boolean

Returns whether the Read pointer is at the beginning of the database.

CalcStatistics as Boolean

Sets/Returns whether Statistics should be calculated during the **SortFile** process
Default is False



FlushBuffers As Boolean

Sets/Returns whether the QWD, QWI and QWA files should be closed and reopened after each **PutRecord** to ensure that buffers get flushed and database date/time stamps are updated.

Default is False.

QWTdateTime As Date

Returns the Date and Time stamp of the Opened Quality Window Template

QWDdateTime As Date

Returns the Date and Time stamp of the Opened Quality Window Data file

Value(index) As String

Sets/Returns the field's actual data within the current database record. Numeric fields (N, A, R, C) without data will contain Null, Text fields (T, P, L) without data will contain the empty string ("").

The Date Value(1) should be formatted as yyyy-mm-dd

The Time Value(2) should be formatted as hh:mm:ss

Value(0) returns the record's date/date stamp in serial date format

ReCalcMaxMax As Integer

Examines all calculated fields and determines minimum number of records that must be buffered. If no @M functions or @RULE functions are performed, then the value = 1.

SummarizedEOF As Boolean

Returns True after the last **GetNextSummarized** reaches the end of the sorted list.

SummarizedInterval As String

Sets/Returns the interval to use when summarizing records

Format: hh:mm:ss Default: 24:00:00

SummarizedStartTime As String

Sets/Returns the start or alignment time for the summarizing interval

Format: hh:mm:ss Default: 00:00:00

SummarizedShiftDuration As Double

Sets/Returns the duration in hours of a Shift.

SummarizedDayBegins As String

Sets/Returns the start or alignment time for a Shift, Day, Week, Month, Quarter or Year

Format: hh:mm:ss Default: 00:00:00

SummarizedWeekBegins As Integer

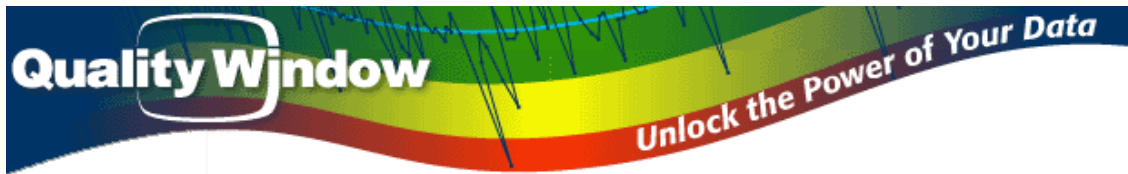
Sets/Returns the day on which Weeks should start on

0=Sunday, 1=Monday, 2=Tuesday, ... 6=Saturday

SummarizedYearBegins As Integer

Sets/Returns the Month in which Years start on

1=January 1st, 2= February 1st, 3=March 1st, ...



SummarizedIncludeNulls As Boolean

Determines whether the summarize function returns intervals with no records.
Default is False

SummarizedBegPtr As Long

Returns the beginning pointer within **Sorted().RecordNo** for the interval after a **GetNextSummarized** has executed. Useful for reading the detail records that the summarized record is made up from.

SummarizedEndPtr As Long

Returns the Ending pointer within **Sorted().RecordNo** for the interval after a **GetNextSummarized** has executed. Useful for reading the detail records that the summarized record is made up from.

Fields.Add *Name, FieldType, Length [, Decimals]*

Adds a new Field to the Fields collection where *Name* is the name of the variable, its *FieldType*, its total *Length*, and optionally its *Decimals*.

Fields.Clear

Clears the entire **Fields** collection

Fields.Count As Long

Returns the total number of **Fields** defined in the database

Fields(index).Name As String

Sets/Returns the Field Name

Fields(index).FieldType As String

Sets/Returns the Field Type such as T, N, P, L, A, R or C.

T = Text

N = Numeric

P = Popup

L = Lookup

A = Average

R = Range

C = Calculated

Fields(index).Length As Integer

Sets/Returns the field length

Fields(index).Decimals As Integer

Sets/Returns the number of Decimals defined if Numeric

Fields(index).XRef As String

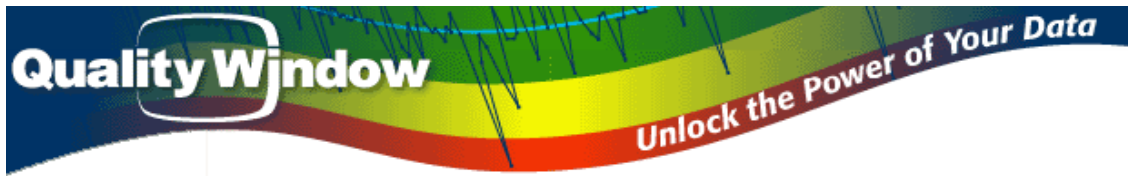
Sets/Returns the field's cross-reference if specified

Fields(index).Sample As Integer

Sets/Returns the field's sampling plan

Fields(index).ReportType As String

Sets/Returns the field's Reporting Type
Can be V, VT, A, A1, A3, A5, A10, A20, A30



Fields(index).UpperCase As String

Sets/Returns if the field should be converted to Upper Case. Can be Y (yes) or N (no).

Fields(index).Required As String

Sets/Returns whether the field is required. Can be Y (yes), N (no) or P (protected).

Fields(index).LimitMethod As Long

Sets/Returns the Field's method of determining Limits

The following Enum (LimitMethods) values may be specified:

LimitsBasedOnPopulation = 0

LimitsBasedOnMrEstimate = 1

LimitsBasedOnLowVariation = 2

LimitsBasedOnHighVariation = 3

LimitsBasedOnLowAverage = 4

LimitsBasedOnHighAverage = 5

Fields(index).TargetMethod As Long

Sets/Returns the Field's method of centering Limits

The following Enum (TargetMethods) values may be specified:

TargetBasedOnFixed = 0

TargetBasedOnCalc = 1

Fields(index).Maximum As Variant

Sets/Returns the field's Maximum value if numeric

Fields(index).USLflag As String

Sets/Returns the field's USL type. Can be F (fixed), C (calculated) or B (blank)

Fields(index).USL As Variant

Sets/Returns the field's USL if Fixed

Fields(index).UCLflag As String

Sets/Returns the field's UCL type. Can be F (fixed), C (calculated) or B (blank)

Fields(index).UCL As Variant

Sets/Returns the field's UCL if Fixed

Fields(index).UWLflag As String

Sets/Returns the field's UWL type. Can be F (fixed), C (calculated) or B (blank)

Fields(index).UWL As Variant

Sets/Returns the field's UWL if Fixed

Fields(index).TGTflag As String

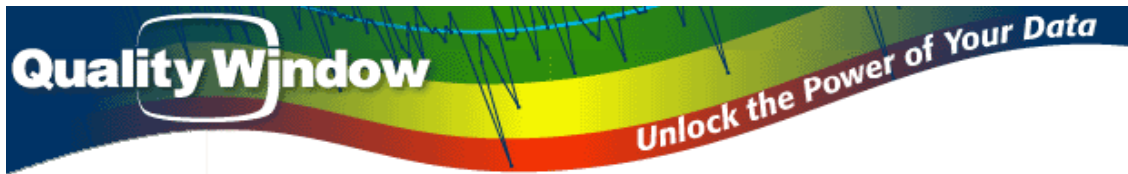
Sets/Returns the field's TGT type. Can be F (fixed), C (calculated) or B (blank)

Fields(index).TGT As Variant

Sets/Returns the field's TGT if Fixed

Fields(index).LWLflag As String

Sets/Returns the field's LWL type. Can be F (fixed), C (calculated) or B (blank)



Fields(index).LWL As Variant

Sets/Returns the field's LWL if Fixed

Fields(index).LCLflag As String

Sets/Returns the field's LCL type. Can be F (fixed), C (calculated) or B (blank)

Fields(index).LCL As Variant

Sets/Returns the field's LCL if Fixed

Fields(index).LSLflag As String

Sets/Returns the field's LSL type. Can be F (fixed), C (calculated) or B (blank)

Fields(index).LSL As Variant

Sets/Returns the field's LSL if Fixed

Fields(index).Minimum As Variant

Sets/Returns the field's Minimum value if numeric

Fields(index).InputFile As String

Sets/Returns the field's Input File if specified

Fields(index).OutputFile As String

Sets/Returns the field's Output File if specified

Fields(index).AvgCount As Integer

Sets/Returns the field's Number of Values to Average if Type Average

Fields(index).Popup As String

Sets/Returns the field's Popup filename if Popup or Lookup

Fields(index).Lookup As Integer

Sets/Returns the field's Lookup Key if Lookup

Fields(index).CompData As Integer

Sets/Returns the field's companion data key if Popup or Lookup

Fields(index).KeyID As Integer

Sets/Returns the field's number of keys to use if Popup or Lookup

Fields(Index).Rules As String

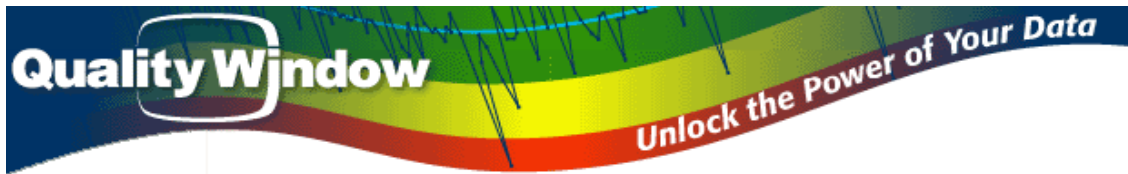
Sets/Returns the Rules enabled for this variable. Each defined rule is given a letter of the alphabet, therefore this string will contain a list of letters where the particular rules have been enabled for this field.

Fields(Index).Range As Integer

Sets/Returns the Field Number associated with this Range variable

Fields(Index).Relation As String

Sets/Returns the Field Numbers defined as Relationships with this variable
Default is: 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000



Fields(Index).Calculation As String

Sets/Returns the Field's calculation statement.

Fields(Index).Units As String

Sets/Returns the Units designation for this variable

Fields(Index).HelpFile As String

Sets/Returns the Help Filename associated with this variable

Fields(Index).Category As String

Sets/Returns a description (ie Test Method) associated with this variable

Fields(index).Pos As Integer

Sets/Returns the field's relative start position within the database RecBuffer

Fields(index).VariableNo As Integer

Sets/Returns the field's variable number ie field 1, field 2, etc

Filter.Add *AndOr, ParBeg, FieldNo, Condition, Value, ParEnd*

Filter.Count As Long

Returns the number of Filter statements in effect

Filter(Index).AndOr As Integer

Sets/Returns whether an AND / OR is used on the condition

Can be: **Logic_NONE** = 0

Logic_AND = 1

Logic_OR = 2

Note: The first filter defined ignores the **AndOr** statement

Note: AND has precedence over OR unless parenthesis are used.

Filter(Index).ParBeg As String

Sets/Returns whether an Open parenthesis is used ie “(“

Filter(Index).FieldNo As Integer

Sets/Returns the Field Number to compare with

Filter(Index).Condition As String

Sets/Returns one of the following conditions:

= Equal To

<> Not Equal To

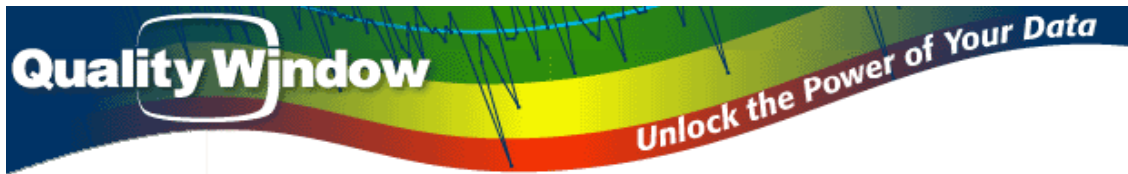
> Greater Than

>= Greater than or Equal To

< Less Than

<= Less Than or Equal To

LIKE Contains the string



Filter(Index).Value As String

Sets/Returns the Value to compare against
“NULL” may also be specified to signify no entry in the field
A field name may be specified within square brackets ie [Team]
If value is specified within single quotes, they are stripped
If using LIKE then string may contain ‘*’ or ‘%’ as wildcard characters

Filter(Index).ParEnd As String

Sets/Returns whether a Close parenthesis is used ie “)”

Filter.Clear

Clears the **Filter** collection

Sorted.Count As Long

Returns the total number of records sorted

Sorted(Index).RecordNo As Long

Returns each individual record number of the sorted result This can then be used as a parameter to the **GetRecord** function to retrieve that particular record.

For I = 1 To QWfile.Sorted.Count

 QWfile.GetRecord QWfile.Sorted(I).RecordNo

Next I

Sorted.Clear **Clears the Sorted collection**

SortFields.Add *FieldNo, SortSeq*

Adds a Sort Field and Sequence to later be used by the **SortFile** function.
Sequence can be 1 = Ascending, 2 = Descending, 3 = Distinct

SortFields.Count As Long

Returns the number of Sort Fields defined

SortFields(Index).FieldNo As Integer

Returns the Field Number to use as Key to the sort

SortFields(Index).SortSeq As Integer

Returns the Sequence to sort the Key Field .

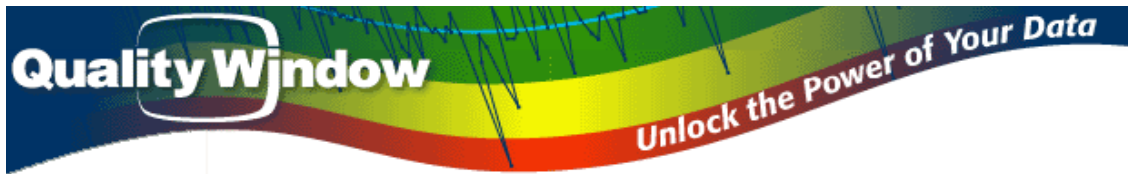
Sort_ASC = 1 sorts the field in Ascending order

Sort_DESC = 2 sorts the field in Descending order

Sort_DISTINCT = 3 extracts all distinct values for that field

SortFields.Clear

Clears the **SortFields** collection



SummarizedFields.Add FieldNo

Sets the Field Number to summarize on

Note that summarizing is done after any filtering and after any sorting

All Statistics are calculated based on each grouping of records.

If the Field Number is greater than 0, then it represents a true Field Number

Field Numbers less than zero means that the summarizing is to be done based on the date/time of each record as follows:

-1	BySecond
-2	ByMinute
-3	ByHour
-4	ByShift
-5	ByDay
-6	ByWeek
-7	ByMonth
-8	ByQuarter
-9	ByYear

Note that in the above list, the settings for **SummarizedShiftDuration**, **SummarizedDayBegins**, **SummarizedWeekBegins** and **SummarizedYearBegins** are respected in determining the interval starting point.

A Field Number specified as zero means that records should be summarized based on a time interval. The properties **SummarizedStartTime** and **SummarizedInterval** are used to calculate the intervals. Null intervals will be suppressed if the property **SummarizedIncludeNulls** is set to False.

SummarizedFields.Count As Integer

Returns the number of fields to summarize on

SummarizedFields(Index).FieldNo

Sets/Returns the field number to summarize on

SummarizedFields.Clear

Clears the SummarizeFields collection

Views.Add *Name, No*

Adds a View to the **Views** collection.

Default view is based on property **DefaultView**

Views.Count As Long

Returns the number of **Views** defined

Views(Index).Name As String

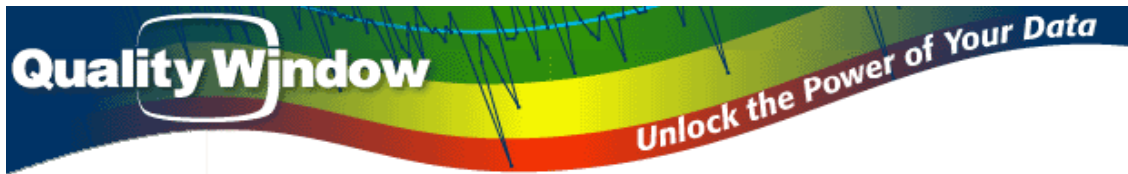
Sets/Returns the name of the **view**

Views(Index).No As Long

Sets/Returns the number of the **view** used as the extension for the view
ie monitor.v15

Views.Clear

Clears the **Views** collection



Rules.Add *Key, Name*

Adds a Rule to the **Rules** collection. This is automatically done on initialize.
Key is alphabet equivalent of the rule ie A-Z
Name is the 3 character mnemonic given to the rule

Rules.Count As Long

Returns the number of **Rules** defined

Rules(Index).Name As String

Sets/Returns the 3 character mnemonic given to the **rule**

Rules(Index).Min As String

Sets/Returns the Minimum number of points to satisfy the **rule**

Rules(Index).Max As String

Sets/Returns the Maximum number of points to satisfy the **rule**

Rules(Index).Cond As String

Sets/Returns the Condition to test against each point

Rules(Index).Zone As String

Sets/Returns the Zone (value) to be tested against

Rules(Index).Desc1 As String

Sets/Returns the first line of the **Rule** description

Rules(Index).Desc2 As String

Sets/Returns the second line of the **Rule** description

Rules(Index).Desc3 As String

Sets/Returns the third line of the **Rule** description

Rules.Clear Clears the **Rules** collection

Statistics.Count as Long

Returns the same value as **Fields.Count**

Statistics(Index).N

Returns the number of non-null entries for the variable

Statistics(Index).Tdev

Returns the Target Deviation of the variable

Statistics(Index).Sum

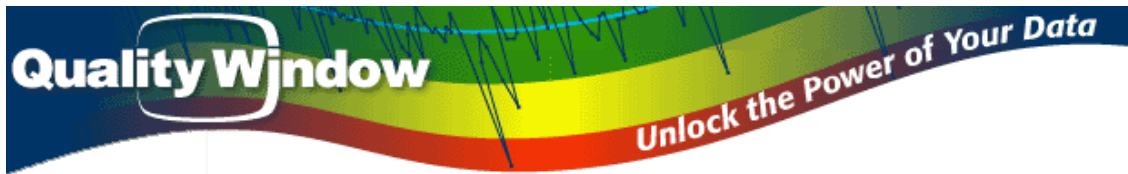
Returns the Sum of the variable

Statistics(Index).SD

Returns the population Standard Deviation of the variable

Statistics(Index).Maximum

Returns the Maximum value of all points for the variable



Statistics(Index).CalcUSL

Returns +4SD for the variable

Statistics(Index).CalcUCL

Returns +3SD for the variable

Statistics(Index).Mean

Returns the mean or Average for the variable

Statistics(Index).CalcLCL

Returns -3SD for the variable

Statistics(Index).CalcLSL

Returns -4SD for the variable

Statistics(Index).Minimum

Returns the Minimum value of all points for the variable

Statistics(Index).OOOLpct

Returns the Observed Out Of Limit Percent for the variable

Statistics(Index).OOOLppm

Returns the Observed Out Of Limit Parts Per Million for the variable

Statistics(Index).COOLpct

Returns the Calculated Out Of Limit Percent for the variable

Statistics(Index).COOLppm

Returns the Calculated Out Of Limit Parts Per Million for the variable

Statistics(Index).CR

Returns the Capability Ratio for the variable

Statistics(Index).TZ

Returns the Target-Z for the variable

Statistics(Index).CPK

Returns the CPK for the variable

Statistics(Index).CV

Returns the % Coefficient of Variation for a variable $(SD / Mean) * 100$

Statistics(Index).FirstValue

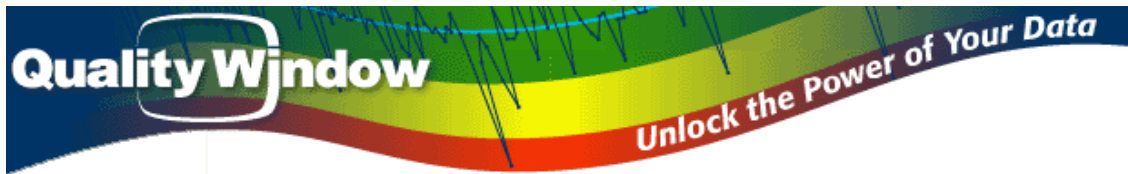
Returns the first non-null value for a variable

If summarizing by Time Intervals, then this returns the lowest possible Date/Time for the interval

Statistics(Index).LastValue

Returns the last non-null value for a variable

If summarizing by Time Intervals, then this returns the highest possible Date/Time for the interval



Statistics(*Index*).AvgTimeBetween

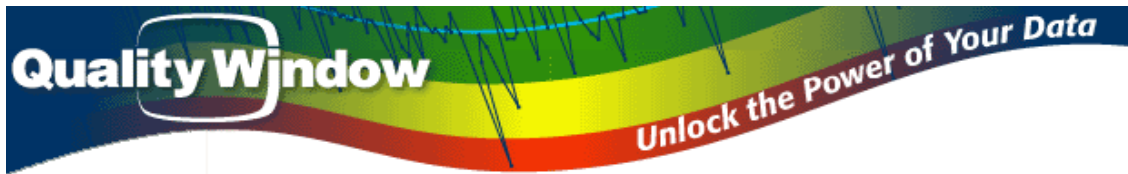
Returns the average time in seconds between non-null values for a variable

Statistics(*Index*).Range

Returns the Range for a variable (Maximum – Minimum)

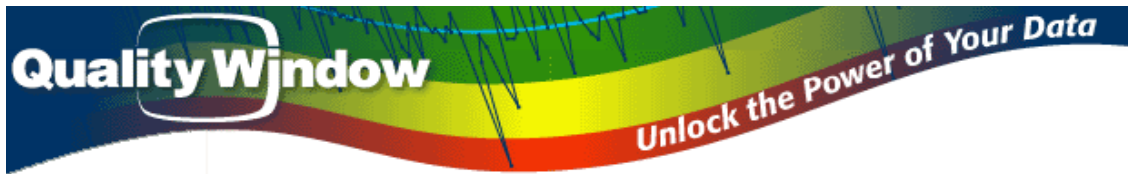
Statistics(*Index*).VariableNo

Returns the Variable Number of the variable



The following table outlines the various Field properties that need to be set in order to create a valid Quality Window Template. For a more detailed description of each property, see the Quality Window documentation.

	Text	Popup	Lookup	Numeric	Average	Range	Calc.
Name	Required	Required	Required	Required	Required	Required	Required
FieldType	Required	Required	Required	Required	Required	Required	Required
Length	Required	Required	Required	Required	Required	Required	Required
Decimals	N/A	N/A	N/A	[0]	[0]	[0]	[0]
Xref	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]
Sample	[0]	[0]	[0]	[0]	[0]	[0]	[0]
ReportType	N/A	N/A	N/A	[V]	[V]	[V]	[V]
UpperCase	[N]	[N]	[N]	N/A	N/A	N/A	N/A
Required	[N]	[N]	[N]	[N]	[N]	[N]	[N]
LimitMethod	N/A	N/A	N/A	[0]	[0]	[0]	[0]
TargetMethod	N/A	N/A	N/A	[0]	[0]	[0]	[0]
InputFile	[blank]	[blank]	N/A	[blank]	[blank]	[blank]	[blank]
OutputFile	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]
Maximum	N/A	N/A	N/A	[0]	[0]	[0]	[0]
USLflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
USL	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
UCLflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
UCL	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
UWLflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
UWL	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
TGTflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
TGT	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
LWLflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
LWL	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
LCLflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
LCL	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
LSLflag	N/A	N/A	N/A	[C]	[C]	[C]	[C]
LSL	N/A	N/A	N/A	[blank]	[blank]	[blank]	[blank]
Minimum	N/A	N/A	N/A	[0]	[0]	[0]	[0]
AvgCount	N/A	N/A	N/A	N/A	[0]	N/A	N/A
Popup	N/A	[blank]	[blank]	N/A	N/A	N/A	N/A
Lookup	N/A	N/A	[0]	N/A	N/A	N/A	N/A
CompData	N/A	[0]	[0]	N/A	N/A	N/A	N/A
KeyID	N/A	[0]	[0]	N/A	N/A	N/A	N/A
Rules	N/A	N/A	N/A	[all rules]	[all rules]	[all rules]	[all rules]
Range	N/A	N/A	N/A	N/A	N/A	[0]	N/A
Relation	N/A	N/A	N/A	[none]	[none]	[none]	[none]
Calculation	N/A	N/A	N/A	N/A	N/A	N/A	[blank]
Units	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]
HelpFile	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]
Category	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]	[blank]



Example 1: This example opens a Quality Window database, displays all Field Names, then displays the 3rd field value of each record, then closes the file.

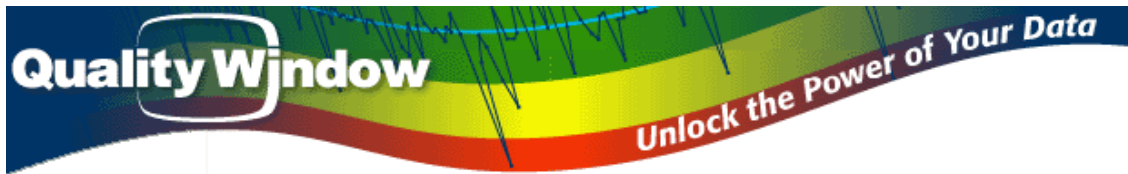
```
Dim QWfile As New QWaccess
Dim I As Integer
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    For I = 1 To .Fields.Count
        Debug.Print .Fields(I).Name
    Next I
    .SeekFirst
    While Not .EOF
        .GetNext
        Debug.Print .RecordNo, .Value(3)
    Wend
    .CloseFile
End With
```

Example 2: This example opens a Quality Window database, displays all Field Names, then displays the 3rd field value of each record but in descending sequence in the file, then closes the file.

```
Dim QWfile As New QWaccess
Dim I As Integer
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    For I = 1 To .Fields.Count
        Debug.Print.Fields(I).Name
    Next I
    .SeekLast
    While Not .BOF
        .GetPrev
        Debug.Print .RecordNo, .Value(3)
    Wend
    .CloseFile
End With
```

Example 3: This example opens a Quality Window database, sorts all records in descending order by Date and Time, displays the results, then closes the file.

```
Dim QWfile As New QWaccess
Dim Rec As Long
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .SortFields.Add 1, Sort_DESC
    .SortFields.Add 2, Sort_DESC
    .SortFile
    For Rec = 1 To .Sorted.Count
        .GetRecord .Sorted(Rec).RecordNo
        Debug.Print .RecordNo, .Value(1), .Value(2)
    Next Rec
    .CloseFile
End With
```



Example 4: This example opens a Quality Window database, selects all records where Team = A, B or D, sorts these records in descending order by Date and Time, displays the results, then closes the file.

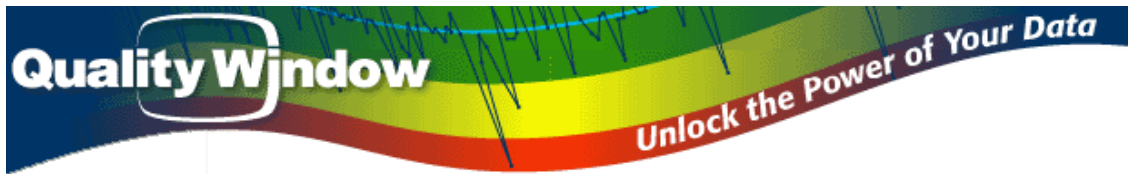
```
Dim QWfile As New QWaccess
Dim Rec As Long
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .Filter.Add Logic_NONE,      "", 3, "=", "10-A Team"
    .Filter.Add Logic_OR,       "", 3, "=", "20-B Team"
    .Filter.Add Logic_OR,       "", 3, "=", "30-D Team"
    .SortFields.Add 1, Sort_DESC
    .SortFields.Add 2, Sort_DESC
    .SortFile
    For Rec = 1 To .Sorted.Count
        .GetRecord .Sorted(Rec).RecordNo
        Debug.Print .RecordNo, .Value(1), .Value(2), .Value(3)
    Next Rec
    .CloseFile
End With
```

Example 5: This example opens a Quality Window database, selects all distinct Teams (Field 3), displays the results, then closes the file.

```
Dim QWfile As New QWaccess
Dim Rec As Long
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .SortFields.Add 3, Sort_DISTINCT
    .SortFile
    For Rec = 1 To .Sorted.Count
        .GetRecord .Sorted(Rec).RecordNo
        Debug.Print .RecordNo, .Value(3)
    Next Rec
    .CloseFile
End With
```

Example 6: This example opens a Quality Window database, Sets Statistics to be calculated, then displays each variable name, Count and Mean.

```
Dim QWfile As New QWaccess
Dim I As Integer
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .CalcStatistics = True
    .SortFile
    For I = 1 To .Statistics.Count
        Debug.Print .Fields(I).Name, .Statistics(I).N, .Statistics(I).Mean
    Next I
    .CloseFile
End With
```

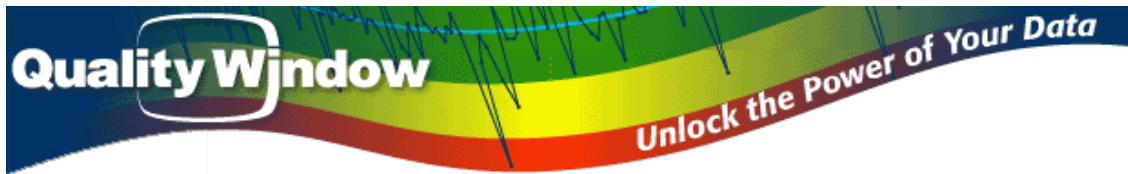


Example 7: This example opens a Quality Window database, Summarizes the entire database by intervals of 8 hours beginning at midnight, and returns a record for each interval containing the start Date/Time of the interval as well as the Mean of variable 5 for each interval.

```
Dim QWfile As New QWaccess
Dim I As Integer
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .SummarizedFields.Add 0
    .SummarizedIncludeNulls = True
    .SummarizedStartTime = "00:00:00"
    .SummarizedInterval = "08:00:00"
    .CalcStatistics = True
    .SortFile
    While Not .SummarizedEOF
        .GetNextSummarized
        Debug.Print .Statistics (1).FirstValue, .Statistics (2).FirstValue, .Statistics (5).Mean
    Wend
    .CloseFile
End With
```

Example 8: This example opens a Quality Window database, Summarizes the entire database grouping by variable 3, and for each change in variable 3, displays the first value Date/Time, the first value of variable 3, and the sum of variable 5 for that grouping. Note that if the database would have been first sorted by variable 3, then the result would have returned the sum of variable 5 for each distinct variable 3.

```
Dim QWfile As New QWaccess
Dim I As Integer
With QWfile
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .SummarizedFields.Add 3
    .CalcStatistics = True
    .SortFile
    While Not .SummarizedEOF
        .GetNextSummarized
        Debug.Print .Statistics (1).FirstValue, .Statistics (2).FirstValue, .Statistics (3).FirstValue,
        .Statistics (5).Sum
    Wend
    .CloseFile
End With
```

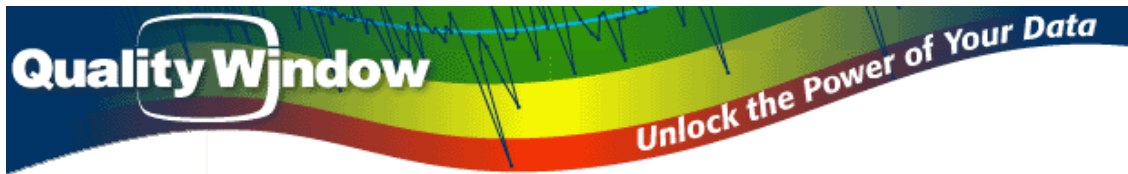


Example 9: This example opens a Quality Window database, clones the template and creates a new database, then filters all records from the first database with Team = "10-A Team" and writes only those records to the second database.

```
Dim QWfile1 As New QWaccess
Dim QWfile2 As New QWaccess
Dim Rec As Long
QWfile1.OpenFile "e:\qwdata\qw45data\monitor.qwt"
QWfile2.CopyTemplate QWfile1
QWfile2.OpenMode = FileCreate
QWfile2.OpenFile "e:\qwdata\qw45data\monitor2.qwt"
QWfile1.Filter.Add Logic_NONE, "", 3, "=", "10-A Team"
QWfile1.SortFile
For Rec = 1 to QWfile1.Sorted.Count
    QWfile1.GetRecord QWfile1.Sorted(Rec).RecordNo
    QWfile2.RecBuffer = QWfile1.RecBuffer
    QWfile2.PutRecord 0
Next Rec
QWfile1.CloseFile
QWfile2.CloseFile
```

Example 10: This example creates a new Quality Window database with a filename of MONITOR2, an Application Title of "New Application", and containing 3 Fields: Date, Time and Conv_Speed. Notice how Name, FieldType, Length and Decimals are normally specified as parameters. Any other parameters can be specified on separate lines using the QWfield object.

```
Dim QWfile As New QWaccess
Dim QWfield as Field
With QWfile
    .Title = "New Application"
    Set Qwfield = .Fields.Add ("Date", "T", 8)
    Set Qwfield = .Fields.Add ("Time", "T", 8)
    Set QWfield = .Fields.Add ("Conv_Speed", "N", 5, 1)
    QWfield.Required = "Y"
    QWfield.Sample = 2
    .OpenMode = FileCreate
    .OpenFile "e:\qwdata\qw45data\monitor2.qwt"
    .Value(1) = "2000-05-01"
    .Value(2) = "13:01:00"
    .Value(3) = 150.1
    .PutRecord 0
    .Value(1) = "2000-05-02"
    .Value(2) = "13:02:00"
    .Value(3) = 150.2
    .PutRecord 0
    .CloseFile
End With
```

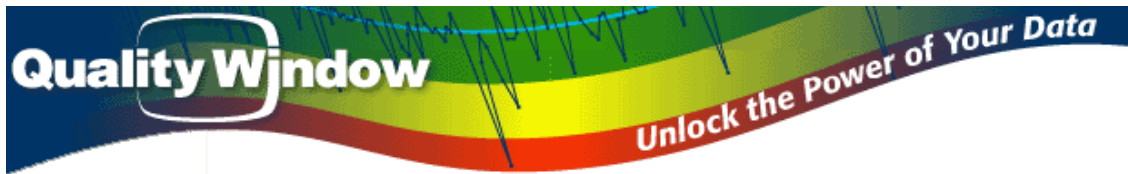


Example 11: This example updates the Target limit of the 3rd variable of an existing Quality Window database (MONITOR) and creates a new one called MONITOR2. Note that the data is not carried with it.

```
Dim QWfile As New QWaccess
Dim QWfile2 As New QWaccess
Dim QWfield as Field
QWfile.OpenFile "e:\qwdata\qw45data\monitor.qwt"
QWfile2.CopyTemplate QWfile
QWfile2.Fields(3).TGTflag = "F"
QWfile2.Fields(3).TGT = "150.1"
QWfile.CloseFile
QWfile2.OpenMode = FileCreate
QWfile2.OpenFile "e:\qwdata\qw45data\monitor2.qwt"
QWfile2.CloseFile
```

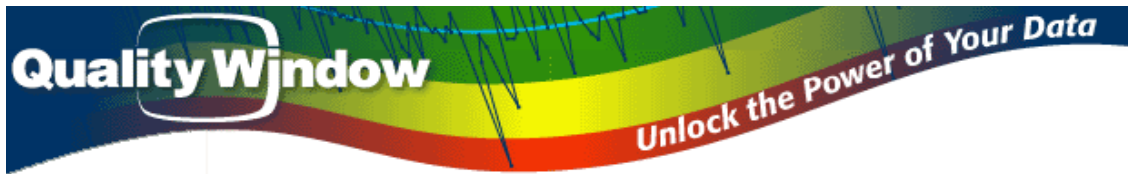
Example 12: This example reads record number 10 from the database MONITOR and marks the record for deletion.

```
Dim QWfile As New QWaccess
With QWfile
    .OpenMode = FileUpdate
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .GetRecord 10
    .Deleted = True
    .PutRecord 10
    .CloseFile
End With
```



Example 13: This example will write a record to the database. Depending on the Date and Time of the record, it will either Add the record to the end of the database or Insert the record at the appropriate place. Then it will ReCalculate the new record and check if the following records needs to be ReCalculated as well (depending on calculations involving @M functions).

```
Dim QWfile As New QWaccess
Dim SavRec As Long, Rec As Long
With QWfile
    .OpenMode = FileUpdate
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
    .InitRecord
    .Value(1) = "2003-01-02": .Value(2) = "11:20:30": .Value(3) = "Team A"
    .InsertRecord
    SavRec = qwfile.RecordNo
    qwfile.SetupReCalc qwfile.RecordNo
    qwfile.ReCalcRecord SavRec
    qwfile.PutRecord SavRec
    If qwfile.ReCalcMaxBack > 1 Then
        For Rec = SavRec + 1 To qwfile.RecordCount
            qwfile.GetRecord Rec
            If qwfile.Deleted = False Then
                qwfile.ReCalcRecord Rec
                qwfile.PutRecord Rec
            End If
        Next Rec
    End If
    .CloseFile
End With
```



Example 14: This example will create a filter to extract all records from last month's production period. It assumes that a day starts at 6:30AM. In other words, if the current date is Oct 22, 2004 the example will extract all records from 2004-09-01 06:30:00 to 2004-10-01 06:29:59. Notice the special use of Field 0 within a Filter statement – the Date/Time string must be in “yyyy-mm-dd hh:mm:ss” format.

```
Dim QWfile As New QWaccess
```

```
Dim datBegDateTime As Date, datEndDateTime As Date
```

```
Dim strBegDateTime As String, strEndDateTime As String
```

```
Dim Rec As Long
```

```
With QWfile
```

```
    .OpenMode = FileReadOnly
```

```
    .OpenFile "e:\qwdata\qw45data\monitor.qwt"
```

```
    datBegDateTime = DateSerial(Year(Now), Month(Now), 1) 'Get Beginning of Current Month
```

```
    datBegDateTime = DateAdd("m", -1, datBegDateTime)    'Subtract 1 Month
```

```
    datBegDateTime = datBegDateTime + TimeSerial(6, 30, 0) 'Start at 6:30AM
```

```
    datEndDateTime = DateAdd("m", 1, datBegDateTime)    'Add 1 Month
```

```
    datEndDateTime = datEndDateTime - TimeSerial(0, 0, 1) 'Subtract 1 Second
```

```
    strBegDateTime = Format(datBegDateTime, "yyyy-mm-dd hh:mm:ss")
```

```
    strEndDateTime = Format(datEndDateTime, "yyyy-mm-dd hh:mm:ss")
```

```
    .Filter.Clear
```

```
    .Filter.Add Logic_NONE, "(", 0, ">=", strBegDateTime, ""
```

```
    .Filter.Add Logic_AND, "", 0, "<=", strEndDateTime, ""
```

```
    .SortFile
```

```
    For Rec = 1 To .Sorted.Count
```

```
        .GetRecord .Sorted(Rec).RecordNo
```

```
        Debug.Print .Value(1), .Value(2)
```

```
    Next Rec
```

```
    .CloseFile
```

```
End With
```