advanced logic technology

WeICAD®

AUTOMATION SYNTAX

VERSION 4.2

What is Automation ?

Some Windows applications such as MS Excel, MS Access or WellCAD expose COM components (Component Object Model). Instances of COM classes, so called objects, can be created in program or script code to access methods and properties of these applications. E.g. a WellCAD.Application object can be created through a simple VBS script to start WellCAD "remotely" and trigger the creation of Borehole Documents.
Properties of an object can be set (e.g. the title of a Borehole Document object) while methods can be called to execute a certain task (e.g. importing a file or inserting a new log). Parameters can be passed with methods and they can return a value (e.g. another object).

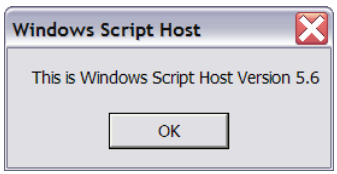
Scripting Host and VBS

Windows released the Windows Scripting Host (WSH) to replace the DOS command line programming. The WSH reads the lines of a script file and interprets them at runtime. The current version of the WSH is 5.6 and is installed with the Windows XP operating system. Win98 / ME/ NT and Win2000 operating systems can be updated with the latest release of the WSH.
BASIC (Beginners All Purpose Symbolic Instruction Code) has a long history and its dialect Visual Basic Script (VBS) has become the most used scripting language in the Windows world. VBS is easy to learn and in conjunction with the WSH provides the tools necessary to automate WellCAD.
A simple text editor, such as Notepad, can be used to create scripts. VBS script files must carry the extension *.VBS and can easily be started with a double click on the file icon.
Books are available explaining VBS in various levels of complexity (one very popular application of scripts is the automation of System / Network Administration). Online documentation is available from the Microsoft homepage (try www.msdn.com and search for "VBS Reference"). This does not include documentation about the COM-classes exposed by applications. It is the best to refer to manuals provided by the applications developer.

Does my WellCAD installation support automation?

To find out if the WSH is installed on your computer:

- Open Notepad (or any other text editor).
- Create a new text file and type in the following line
  `WScript.Echo "This is " & WScript.Name & " Version " WScript.Version`
- Save the file with an *.VBS extension.
- Double click on the file and see a message box popping up on your screen.

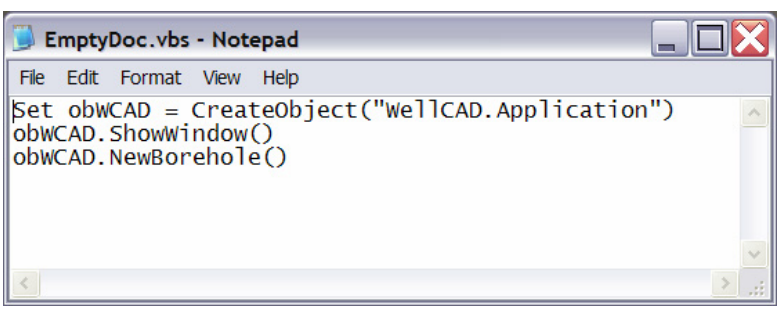


The message box called through the *WScript.Echo* method displays the name and version of the scripting host installed on your computer.

Getting Started - my first script?

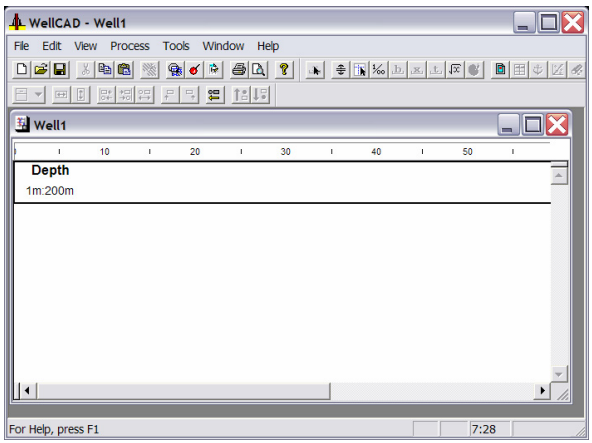
Before running scripts to automate your current WellCAD version, you first have to register WellCAD as automation server in the system registry. This is simply done by starting WellCAD on your computer. Please execute the following steps to create a first script to automate WellCAD:

- Open Notepad (or any other text editor).
- Create a new text file and type in the following lines:



- Save the file with the *.VBS extension.
- Double click on the file to run the script.

The WellCAD application should open with an empty borehole document created for you.



The first line of the script is your main entry point to WellCAD. Calling the *CreateObject* method instantiated and returned an object of the *WellCAD.Application* class. In line 2 and 3 of the script this object is used to call the application's methods to show the main window of WellCAD on screen and to create a new borehole document.
Please extend the script by a few lines so it corresponds to the one shown in the following example (please do not type the line numbers):

```
1 Set obWCAD = CreateObject("WellCAD.Application")
2 obWCAD.ShowWindow()
3 Set obBHole = obWCAD.NewBorehole()
4 obBHole.Name = "MyDoc"
5 Set obLithoLog = obBHole.InsertNewLog(7)
6 obLithoLog.Name = "Litho"
```

Save the file and run the script by double clicking on the file icon.
WellCAD will open a new borehole document again. But the document title has been set and an empty Lithology Log has been inserted with the log title set to "Litho".
Line three of the script has been altered from its former version to return an object of type *WellCAD.Borehole*. In line **4** of the script we set the *Name* property of the *Borehole* object to "MyDoc". The following line **5** calls the *InsertNewLog* method of the *Borehole* object to create a new Lithology Log and return an object of type *WellCAD.Log*. Specific indices are used to create logs of different types. To set the log title, line **6** of the script sets the *Name* property of the *Log* object to "Litho".

Note: If the execution of the script fails ensure the WellCAD application does not continue to run in the background. Open the Task Manager and look in the *Process* tab for *WellCAD.exe*. If WellCAD is still running, stop the process.

Getting Started – a more advanced script?

The following listing is from a script that creates a new borehole document using a LAS file import. The script accesses the imported gamma ray log and applies a filter. Finally a template is applied and the WCL file stored. The sample is available on the WellCAD installation CD or contact ALT to retrieve a copy.

```
1 Set obWCAD = CreateObject("WellCAD.Application")
2 obWCAD.ShowWindow()
3 Set obBHole = obWCAD.FileImport("C:\Automation\AutoIntro\AutoIntro.LAS",_
      FALSE, "C:\Automation\AutoIntro\AutoIntro.ini")
4 Set obGRLogFilt = obBHole.FilterLog("GR", FALSE,_
      "C:\Automation\AutoIntro\AutoIntro.ini")
5 obGRLogFilt.Name = "GR Filtered"
6 obBHole.ApplyTemplate "C:\Automation\AutoIntro\AutoIntro.WDT"
7 obBhole.SaveAs "C:\Automation\AutoIntro\AutoIntro.WCL"
```

We will examine the script line by line now.
With `Set obWCAD = CreateObject("WellCAD.Application")` in line **1** we trigger the execution of WellCAD and get a handle to the application (`obWCAD`) in return. Using this handle we call the `ShowWindow()` method in line **2** which opens the applications workspace on our screen.
The WellCAD application object supports a `FileImport` (line **3**) method which takes up to three parameters. The first parameters is a string indicating the location and filename of the dataset to import. In the above example we want to load a *LAS* file named *AutoIntro* from the *C:\Automation\AutoIntro* folder. When importing a file into WellCAD a dialog box opens prompting for parameters. To avoid the display of dialog boxes and allow a "silent" import the second parameter of the `FileImport` method has been set to `FALSE`. If this parameter is set to `TRUE` (which is the default) you would get the LAS Import Dialog Box displayed. Nevertheless, the import parameters must be defined somewhere. A configuration file (a simple *.ini file) can be setup and passed as third paramter of the `FileImport` method. A file import in WellCAD creates a new borehole document. The succesfull execution of the `FileImport` method will return a handle to a Borehole object (`obBHole`).
We want to filter the log with the title "GR" that has been imported. Using the `obBHole` object we can call the `FilterLog` method in line **4**. Similar to the method for the file import this method takes three parameters as well. The first parameter is the title of the log, followed by a flag that allows us to suppress the display of the processing dialog box. The third parameter is the configuration file from which the filter length and filter type will be read. You can open and edit the configuration file in a text editor like Notepad. The `FilterLog` method returns a handle to the new log containing the filtered data set. We rename the log and set the `Name` property of the `obGRLogFilt` object to `"GR Filtered"` in line **5**.

By calling `obBHole.ApplyTemplate "C:\Automation\AutoIntro\AutoIntro.WDT"` in line **6** we load and apply a WellCAD Document Template. Please refer to the syntax hereafter to learn more about the different parameters of the *ApplyTemplate* method.
The last step performed by the script is the saving of an WCL file using the `SaveAs` method (line **7**) of the application object. Note the different syntax for methods returning no values. Method parameters are called without parentheses.

How do I continue from here?

The remaining pages of this document describe the syntax of each method and property exposed by WellCAD. You can build them into scripts or applicaton code. Declaration of variables, loops, conditional tests etc. are specific to the programming language used and are not part of WellCAD Automation. Refer to books dealing with the programming language in more detail. Please note that VBS (Visual Basic Scripting) is probably one of the easiest ways to use Automation for batch processing purposes as it does not require a special programming environment or compiler and is easy to learn. If you intend to build comprehensive graphical user interfaces you should consider using a more sophisticated programming language like Visual Basic or Visual C++.
A couple of VBS script examples are available on the WellCAD installation CD and from ALT. Treat them as a starting point to develop your own scripts. The example scripts cover topics like building a loop to read files out of a specific folder and repeat a process for each of them, how to access a configuration file used as lookup table to rename log names to standard titles so that they comply with the template requirements or how start print jobs. The scripts are kept simple so you can easily find your way through it.

W A R N I N G !

Files with a VBS extension will automatically start the Windows Scripting Host and execution of the commands contained in the script as soon as you double click on it. Objects are available in VBS allowing access to various components of your operating system (e.g. the file system). You should **never** execute a VBS from an unknown source just to see what happens. Open the script in a text editor first and check it contents.

Navigation Tree

Application

Borehole

Log

LithoBed

LithoDictionary

LithoPattern

CommentBox

Structure

DrillItem

EqpItem

IntervalItem

CrossBox

SchmitBox

FossilItem

StackItem

ODBC

MarkerBox

NewBorehole *(VARIANT Template)*

Creates and returns new Borehole Document.

Object: Application

Return Value: Borehole Object

Parameter:

Template [optional]	A string specifying the path to the template file (*.WDT) from which the new borehole should be created. If the path is not set an empty borehole document will be generated.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.NewBorehole("C:\Temp\Demo.WDT")

'Or

Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.NewBorehole()
```

OpenBorehole *(VARIANT File)*

Creates and returns new Borehole Document.

Object: Application

Return Value: Borehole Object

Parameter:

File [optional] A string specifying the path to the WellCAD file (*.WCL). If the path is not set the File Open dialog box will be displayed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.WCL")

'Or

Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole()
```

GetBorehole *(VARIANT Document)*

Returns a Borehole Document currently open in WellCAD.

Object: Application

Return Value: Borehole Object

Parameter:

Document [optional]	*Document* is the zero based index of the borehole document to be retrieved. The first document created got the index 0, the second one got the index 1 and so on. If the *Document* parameter is empty the last borehole document opened in the WellCAD window will be returned. Which borehole document will be returned is entirely based on the index and does not depend on the status (activated, minimized, maximized etc.)

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole(0)

'Or

Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
```

ShowWindow ()

Call this method to display the WellCAD window on screen. The example below shows two possible applications.

Object: Application

Return Value: Bool

Example:

```
'Displays the WellCAD window after the process finished
Set obWCAD = CreateObject("WellCAD.Application")
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, CONFIGFILE
obWCAD.ShowWindow()

'Displays the WellCAD window before the process starts
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, CONFIGFILE
```

RefreshWindow ()

Call this method to update the display of the current borehole document.

Object: Borehole

Return Value: -

Remarks:

Usually the call of a method triggers the refresh of the data display. Nevertheless the user can disable this behaviour setting the *AutoUpdate* property of the Borehole object to FALSE. In this case an update of the display can be achieved by calling the *RefreshWindow* method.

Example:

```
'Displays the WellCAD window after the process finished
Set obWCAD = CreateObject("WellCAD.Application")
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, CONFIGFILE
obWCAD.ShowWindow()

'Displays the WellCAD window before the process starts
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, CONFIGFILE
```

FileImport *(VARIANT File, VARIANT PromptUser, VARIANT ConfigFile, VARIANT LogFile)*

Imports data from the specified file and returns a new Borehole Document.
Supported file formats are: WA*, TXT, CSV, ASC, RD, TFD, LAS, BMP, TIF, JPG, GIF, ASCII (AppleCore)

Object: Application

Return Value: Borehole Object

Parameter:

FileName [optional]	A string specifying the input file. If no parameter is passed in this method the File Import dialog box will be opened.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).
LogFile [optional]	The full path and filename of a log file. Error or warning messages generated during the process will be written into the specified ASCII file.

Remarks:

This method automatically recognizes the file extension and initiates the corresponding import algorithm. The configuration file must contain the import options. Each file format has its own field in the *.ini file.

WA* files: No entry in the configuration file is needed except for the FWS log. For this log type the `DefaultTimeSamplingRate` and `DefaultTimeOffset` from the `[TextImport]` field are read. The log type generated will be determined from the file extension (e.g. WAL = Lithology Log, WAM = Mud Log).

TXT, CSV, ASC files:

```
[TextImport]
Delimiter=tab space , : ;
SkipLines=0
TitleEnabled=YES
UnitsEnabled=YES
MultiLogEnabled=NO
DepthUnit=ft
ResampleEnabled=NO
ResampleRate=0.1
LogType= MudLog
'LogType= WellLog
'LogType= IntervalLog
'LogType= FWSLog
'LogType= VspLog
'LogType= ImgLog
'LogType= ImgFloat2Log
'LogType= ImgFloat4Log
'LogType= RGBLog
'LogType= StructureLog
'LogType= AnalysisLog
'LogType= PercentLog
'LogType= BioLog
'LogType= LithoLog
'LogType= CommentLog
'LogType= StrataLog
'LogType= EngineeringLog
'LogType= StackLog
'LogType= OLELog
'LogType= PolarRoseLog
'LogType= CrossLog
'LogType= DepthLog
'LogType= FossilLog

'Single Log Import
LogName= Caliper
LogUnit= mm

' specific to FWS import
DefaultTimeSamplingRate= 5
DefaultTimeOffset= 0

'Multi Log Import
Col1_LogName= Depth
Col1_LogUnit= m
Col1_ReferenceEnabled= YES
```

```
Col1_Type= Depth
'Col1_Type= Data
'Col1_Type= DD/MM/YYYY
Col1_DateTimePartner= -1
Col1_GMTOffset= 0
Col2_LogName= GR
Col2_LogUnit= API
Col2_ReferenceEnabled= NO
Col2_Type= Data
…
```

LAS files:

```
[LASImport]
DepthTolerance=10.0
MaxDepthRange=YES
TopDepth=0.0
BottomDepth=150.0
```

ASCII (AppleCore):

```
[AppleCore]
MergeSameItems= TRUE
Header= C:\Temp\Demo.wch
Template= C:\Temp\Demo.wdt
```

RD files:

```
[RDImport]
UseExternalTolFile= NO
TolFileName= C:/Temp/Demo.TOl

[FAC40]
OrientTo= None
'OrientTo= North
'OrientTo= Highside
EnableAmplitude= YES
EnableTravelTime= YES
EnableDeviation= YES
EnableRawMagIncl= NO
EnableCalMagIncl= NO
EnableRawTemp= NO
EnableCalTemp= NO
IgnoreSyncError= FALSE
MaxDepthRange= YES
TopDepth= 0.0
BottomDepth= 10.0

[OBI40]
OrientTo= None
'OrientTo= North
```

```
'OrientTo= Highside
EnableColorImage= YES
EnableBWImage= YES
EnableDeviation= YES
EnableRawMagIncl= NO
EnableCalMagIncl= NO
EnableRawTemp= NO
EnableCalTemp= NO
MaxDepthRange= YES
TopDepth= 0.0
BottomDepth= 10.0

[ABI40]
OrientTo= None
'OrientTo= North
'OrientTo= Highside
EnableEcho1= YES
EnableEcho2= NO
EnableThickness= NO
EnableWave= NO
EnableWavelet= NO
EnableSlowChannels= NO
EnableDeviation= NO
IgnoreSyncError= NO
MaxDepthRange= YES
TopDepth= 0.0
BottomDepth= 10.0

[MIT]
OrientTo= None
'OrientTo= Highside
EnableRawFinger= YES
EnableCalFinger= YES
EnableRawIncl= YES
EnableDeviation= YES
EnableSlowChannels= NO
MaxDepthRange= YES
TopDepth= 0.0
BottomDepth= 10.0

[2Saf1000]
MaxDepthRange= YES
TopDepth= 0.0
BottomDepth= 10.0
```

TFD files:

```
[TFDImport]
UseExternalTolFile= NO
TolFileName= C:/Temp/Demo.TOL

[FAC40]
OrientTo= None
'OrientTo= North
'OrientTo= Highside
EnableEcho= YES
EnableDeviation= YES
EnableRawMagIncl= NO
EnableCalMagIncl= NO
EnableSlowChannels= NO
IgnoreSyncError= FALSE

[OBI40]
OrientTo= None
'OrientTo= North
'OrientTo= Highside
EnableColorImage= YES
EnableBWImage= YES
EnableDeviation= YES

[ABI40]
OrientTo= None
'OrientTo= North
'OrientTo= Highside
EnableEcho1= YES
EnableEcho2= NO
EnableThickness= NO
EnableWave= NO
EnableWavelet= NO
EnableSlowChannels= NO
EnableDeviation= NO
IgnoreSyncError= NO

[MIT]
OrientTo= None
'OrientTo= Highside
EnableRawFinger= YES
EnableCalFinger= YES
EnableRawIncl= YES
EnableDeviation= YES
```

BMP files:

```
[RGBImport]
DepthUnit= m
'DepthUnit= ft
TopDepth= 0.0
BottomDepth= 10.0
Rotate= 0
'Rotate= 90
'Rotate= 180
'Rotate= 270
Resample= YES
SampleRate= 0.1
ReferencePosition= 0.0
```

CIB files:

```
[ReevesFileImport]
SolidToadCategory= 1
HollowToadCategory= 0
DipNullValue= -999.25
DefaultTimeSamplingRate= 5
DefaultTimeOffset= 0
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.FileImport("C:\Demo.LAS",FALSE,"C:\Import.ini")
```

FileExport *(VARIANT File, VARIANT PromptUser, VARIANT ConfigFile, VARIANT LogFile)*

Exports data to the specified file.
Supported file formats are: LAS, EMF, CGM

Object: Borehole

Return Value: -

Parameter:

FileName [optional]	A string specifying the output path and filename. If the string is empty the standard File Export dialog box will be displayed.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).
LogFile [optional]	The full path and filename of a log file. Error or warning messages generated during the process will be written into the specified ASCII file.

Remarks:

This method automatically recognizes the supported file extensions and initiates the corresponding data export algorithm. The configuration file must contain the import options. Each file format has its own field in the *.ini file. The parameters in the configuration file reflect the options shown in the file export dialog boxes. As they are mostly self explanatory and in depth discussion will be skipped at this point.

Note: WA* files can only be created from a Log object.

LAS files:

```
[LASExport]
Log1=GR
Log2=RHO
'Log3=…
'SamplingRate=0.1
EnableHeader=FALSE
EnableWrap=TRUE
MaxDepthRange=YES
TopDepth=0.0
BottomDepth=150.0
```

EMF files:

```
[EMFExport]
MaxDepthRange=YES
TopDepth= 0.0
BottomDepth= 10.0
```

CGM files:

```
[CGMExport]
MaxDepthRange=YES
TopDepth= 0.0
BottomDepth= 10.0
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Automation\Test\Demo.WCL")
obBHDoc.FileExport "C:\Automation\Test\Demo.LAS", FALSE,_
"C:\Automation\Test\Test.ini"
obBHDoc.FileExport "C:\Automation\Test\Demo.EMF", FALSE,_
"C:\Automation\Test\Test.ini"
obBHDoc.FileExport "C:\Automation\Test\Demo.CGM", FALSE,_
"C:\Automation\Test\Test.ini"
```

FileExport *(VARIANT Path)*

Exports data to the specified file.
Supported file formats are: WA#

Object: Log

Return Value: Bool

Parameter:

Path	A string specifying the output path. File title and file extension will automatically set according to log title and log type.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Automation\Test\Demo.WCL")
Set obGRLog = obBHDoc.Log("GR")
obGRLog.FileExport  "C:\Automation\Test\"
```

CloseBorehole *(VARIANT PromptForSaving, VARIANT Document)*

Closes the specified borehole document with or without saving.

Object: Application

Return Value: -

Parameter:

PromptForSaving [optional]	Set this flag to TRUE (default) if you want to display the SaveAs dialog box. If the flag is set to FALSE the document closes without saving.
Document [optional]	*Document* is the zero based index of the borehole document to be closed. You can also pass the document title. If *Document* is empty the last document added to the WellCAD workspace will be closed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, CONFIGFILE
'Prompt user to close docs
If MsgBox("Do you want to close the document?",_
vbYesNo + vbDefaultButton2, "Close Document") = vbYes Then
obWCAD.CloseBorehole TRUE, 0
End If
```

Quit *(VARIANT PromptForSaving)*

Closes all open borehole documents and exits WellCAD. Optionally the user can be prompted for saving the borehole documents.

Object: Application

Return Value: -

Parameter:

PromptForSaving [optional]	Set this flag to TRUE (default) if you want to display the SaveAs dialog box for each document that will be closed. If the flag is set to FALSE the documents close without saving.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, CONFIGFILE
'Prompt user to close docs
If MsgBox("Do you want to close the document?",_
vbYesNo + vbDefaultButton2, "Close Document") = vbYes Then
obWCAD.Quit TRUE
End If
```

CheckFormula (BSTR Formula)

Verifies the syntax of a formula.

Object: Borehole

Return Value: Bool

Parameter:

Formula A string of the formula that needs to be checked.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
If obBHDoc.CheckFormula("({GR}-min({GR}))/(max({GR})-min({GR}))")Then
Set obGRIndexLog = obBHDoc.InsertNewLog(2)
obGRIndexLog.Name = "GR Index"
obGRIndexLog.Formula = "({GR} - min({GR})) / (max({GR}) - min({GR}))"
End If
```

DoPrint *(BOOL bPrintDlg, float fStartDepth, float fEndDepth, short iNbOfCopies)*

Sends the current document to the default or user selected printer.

Object: Borehole

Return Value: -

Parameter:

bPrintDlg	Set this flag to TRUE if you want to display the standard Print dialog box. If the flag is FALSE the document will be send to the printer choosen as the default one.
fStartDepth	Upper limit of the depth interval you would like to print in units of the master depth axis.
fEndDepth	Lower limit of the depth interval you would like to print in units of the master depth axis.
iNbOfCopies	Set the number of copies that you would like to print.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.DoPrint TRUE, obBHDoc.TopDepth, obBHDoc.BottomDepth, 1
```

InsertNewLog *(SHORT LogType)*

Creates a new WellCAD log. The type of log is determined from the *LogType* index.

Object: Borehole

Return Value: Log Object

Parameter:

LogType Index specifying the log type to be inserted.

Remarks:

This following list shows the different indizes for automation supported WellCAD logs:

UNDEFINED	0	INTERVALLOG	13
WELLLOG	1	ANALYSISLOG	14
FORMULALOG	2	PERCENTLOG	15
MUDLOG	3	COREDESCLOG	16
FWSLOG	4	DEPTHLOG	17
IMAGELOG	5	STRATALOG	18
STRUCTURELOG	6	STACKLOG	19
LITHOLOG	7	POLARROSELOG	20
COMMENTLOG	8	CROSSLOG	21
ENGINEERINGLOG	9	OLELOG	22
RGBLOG	10	SHADINGLOG	23
IMAGEFLOAT2LOG	11	MARKERLOG	24
IMAGEFLOAT4LOG	12		

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
If obBHDoc.CheckFormula("({GR}-min({GR}))/(max({GR})-min({GR}))")Then
Set obGRIndexLog = obBHDoc.InsertNewLog(2)
obGRIndexLog.Name = "GR Index"
obGRIndexLog.Formula = "({GR} - min({GR})) / (max({GR}) - min({GR}))"
End If
```

RemoveLog *(VARIANT Log)*

Removes the specified log from the borehole document.

Object: Borehole

Return Value: -

Parameter:

Log	Integer specifying the zero based index of the log or a string with the log title to be removed from the document.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.RemoveLog "GR"
```

Log *(VARIANT Log)*

This [Get] property of the Borehole object returns an log object of the specified channel.

Object: Borehole

Return Value: Log object

Parameter:

Log Integer specifying the zero based index of the log or a string with the log title to be retrieved.

Example:

```
'Access a log in a document
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
Set obLog = obBHDoc.Log("GR")
obLog.Name = "GR calibrated"
```

Log (VARIANT Log, IDispatch NewLog)*

The [Put] version of the Log property allows replacement of a *Log* in the current document with a log (*NewLog*) of the same or another document. This is the equivilent to the Copy'n Paste option.

Object: Borehole

Return Value: -

Parameter:

Log	Integer specifying the zero based index of the log or a string with the log title to be retrieved.
NewLog	A log object (such as the one returned by the InsertNewLog method or Log property) that will replace the log specified by name or index in the first parameter.

Example:

```
'Copy and Paste a log between documents
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc1 = obWCAD.OpenBorehole("C:\Temp\Demo1.wcl")
Set obBHDoc2 = obWCAD.OpenBorehole("C:\Temp\Demo2.wcl")
Set obNewLog = obBHDoc2.Log("GR")
obBHDoc1.Log "GR", obNewLog
```

AddLog *(IDispatch* Log)*

Adds the copy of a log passed as a log object into the calling borehole document. Another way of copy and pasting logs between documents.

Object: Borehole

Return Value: Log object

Parameter:

Log Log object from which a copy will be added to the borehole document.

Example:

```
'Copy and Paste a log
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc1 = obWCAD.OpenBorehole("C:\Temp\Demo1.wcl")
Set obBHDoc2 = obWCAD.OpenBorehole("C:\Temp\Demo2.wcl")
Set obLog = obBHDoc2.Log("GR")
Set obNewLog = obBHDoc1.AddLog(obLog)
```

ConvertLogTo (VARIANT LogToConvert, VARIANT LogType, VARIANT PromptUser, VARIANT ConfigFile)

Converts, where possible, one log type into another one.

Object: Borehole

Return Value: Log object

Parameter:

LogToConvert	A string specifying the log name or an integer representing the index of the Log to convert.
LogType	An integer specifying the target log type. (See *InsertNewLog* method for a list of codes.)
PromptUser [optional]	*Currently not supported.*
ConfigFile [optional]	*Currently not supported.*

Remarks:

Please note that currently a silent conversion is not possible. When required the conversion dialog box will open.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
Set obCommentLog = obBHDoc.ConvertLogTo("Litho",8)
```

MergeLogs *(VARIANT Log1, VARIANT Log2, VARIANT Average, VARIANT CreateNewLog)*

Merges the data of the two specified logs.

Object: Borehole

Return Value: -

Parameter:

Log1	Integer specifying the zero based index of the log or a string with the log title. If no new log is created this log will receive the data from *Log2.*
Log2	Integer specifying the zero based index of the log or a string with the log title.
Average [optional]	If set to TRUE the data within the depth overlap between *Log1* and *Log2* will be averaged. If the flag is set to FALSE the data from *Log1* is taken. The default is set to TRUE.
CreateNewLog [optional]	If set to TRUE a new log with the merged data set will be created (and returned as Log object). If the flag is set to FALSE the data of *Log2* will be added to *Log1*. The default is set to FALSE.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.MergeLogs "GR upper", "GR lower"
```

SaveAs (BSTR Filename)

Saves the calling borehole document using the name and location specified in *Filename*.

Object: Borehole

Return Value: Bool

Parameter:

FileName String specifying the path and name of the file to be saved.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
. . .
obBHDoc.SaveAs "C:\Temp\Demo_Proc.wcl"
```

ApplyTemplate *(BSTR Template, VARIANT PromptIfNotFound, VARIANT CreateNewLogs, VARIANT CreateNewLayers, VARIANT ApplyAnnotationSettings, VARIANT ReplaceHeader, VARIANT KeepExistingCharts, VARIANT CreateNewCharts)*

Merges the data of the two specified logs.

Object: Borehole

Return Value: Bool

Parameter:

Template	String specifying the name and path of the template (*.WDT) to be applied.
PromptIfNotFound [optional]	Boolean indicating whether the dialog to prompt the user for logs not found in the template is displayed or not. The default is set to TRUE.
CreateNewLogs [optional]	Boolean indicating whether all logs contained in the template will be transferred to the target document or not. The default is set to FALSE.
CreateNewLayers [optional]	Boolean indicating whether all annotation layers contained in the template will be transferred to the target document or not. The default is set to FALSE.
ApplyAnnotationSettings [optional]	Boolean indicating whether settings for operational symbols (e.g. symbol offsets) in annotation layers in the template will be applied to the target document or not. The default is set to FALSE.

ReplaceHeader [optional]	Boolean indicating whether the current document header will be replaced by the header contained in the template. The default is set to FALSE. If the header will be replaced header key words of bothe headers will be tested and information transferred if the key words are equal.
KeepExistingCharts [optional]	Boolean indicating whether cross-plot charts in the target document will be kept. The default is set to TRUE.
CreateNewCharts [optional]	Boolean indicating whether cross-plot charts contained in the template will be transferred to the target document. The default is set to FALSE.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
. . .
obBHDoc.ApplyTemplate "C:\Temp\Demo.wdt", FALSE, TRUE
```

ReadDatabase *(BSTR Scriptname)*

Opens and interprets an SQL script to download data from a database. It is the equivalent to the Open Database option in the WellCAD / File menu.

Object: Borehole

Return Value: Bool

Parameter:

FileName String specifying the path and name of the SQL script to be called.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.NewBorehole()
obBHDoc.ReadDatabase "C:\Temp\LoadData.SQL"
```

WriteDatabase *(BSTR Scriptname)*

Uploads the data of the calling borehole document to a database by executing the commands from the specified SQL script. This method is the equivalent to the Save Database option in the WellCAD / File menu.

Object: Borehole

Return Value: Bool

Parameter:

FileName String specifying the path and name of the SQL script to be called.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
. . .
obBHDoc.WriteDatabase "C:\Temp\SaveData.SQL"
```

DepthMatchLog *(VARIANT LogToMatch, VARIANT LinkLog)*

Depth matches the specified Log using the links created from the specified Depth Log.

Object: Borehole

Return Value: -

Parameter:

LogToMatch [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document that is going to be depth matched.
LinkLog [optional]	A string specifying the Depth Log name or an integer representing the index of the Depth Log in the Borehole Document from which the depth matching links will be created.

Remarks:

This method is the equivalent to the DepthMatcher tool provided in the WellCAD / Tools menu. Using a Depth Log providing the matcher links is equal to selecting the "Links from Depth Log" option from the Edit menu of the DepthMatcher window.
If the parameter list is empty or only the LogToMatch is provided the DepthMatcher dialog box will be displayed. A complete parameter list (unless the provided logs are not valid) will result in immediate application of the depth matching process.
Direction of depth matching:
Assuming the Depth Log contains the following data row 100, 98. With the WellCAD.ini file containing the following entry

```
[DepthMatcher]
ReversedDepthLog=FALSE,
```

The data values at a depth of 100 (or at 98 using the Depth Log axis) and above will be shifted upwards by 2 depth units. If the `ReversedDepthLog=TRUE` the data points would be shifted downwards.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Automation\Test\Demo.WCL")
obBHDoc.DepthMatchLog "PHI", "Core"
```

FilterLog (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Calculates a new filtered data set.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the Well Log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[FilterLog]
FilterType=Median
'FilterType=MovingAverage
'FilterType=WeightedAverage
FilterWidth=5
MaxDepthRange = YES
TopDepth = 5.0
BottomDepth = 10.0
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
Set obLog = obBHDoc.FilterLog("GR", TRUE, CONFIGFILE)
```

ResampleLog *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Resamples a data set according to a new constant sampling rate or sample point determined from a log.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the Well, Mud, RGB or Image Log to be resampled.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ResampleLog]
SamplingRate=0.1
ReferenceLog=Plugs
UseReferenceLog=TRUE
UseNearestPoint=FALSE
VerticalSamplingFactor=1
RadialSamplingFactor=1
RadialDownSampling=YES
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
Set obLog = obBHDoc.ResampleLog("GR", FALSE, CONFIGFILE)
```

BlockLog *(VARIANT Log , VARIANT PromptUser, VARIANT ConfigFile)*

Calculates statistical values for each depth interval determined from a reference log or specified by the user.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the Well Log t be processed.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[BlockLog]
ReferenceInterval=Lithology
'ReferenceInterval=10
Resolution=0.1
EmptyIntervalMode=Interpolate
'EmptyIntervalMode=Null
'EmptyIntervalMode=Max
'EmptyIntervalMode=Min
Interpolate=YES
Minimum=NO
Maximum=NO
Average=YES
StdDev=NO
AveragePlusStdDev=YES
AverageMinusStdDev=YES
```

Example:

```
Sum=No
SumNorm=No
Area=No
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.BlockLog "GR", FALSE, CONFIGFILE
```

MergeSameLogItems *(VARIANT Log)*

Merges equal items (e.g. consecutive beds of equal lithology) within a log.

Object: Borehole

Return Value: -

Parameter:

Log A string specifying the log name or an integer representing the index of the Lithology, Comment, CoreDesc or Interval Log to be processed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.FileImport("C:\Temp\Litho.wal", FALSE)
obBHDoc.MergeSameLogItems "Lithology"
```

ExtendLog *(VARIANT Log, float TopDepth, float BottomDepth)*

Extends the allocated depth ange of Well, Formula and Analysis Logs.

Object: Borehole

Return Value: -

Parameter:

Log	A string specifying the log name or an integer representing the index of the Well, Formula or Analysis Log to be processed.
TopDepth	New top depth of the log in units of the current depth axis.
BottomDepth	New bottom depth of the log in units of the current depth axis.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.FileImport("C:\Temp\GR.waw", FALSE)
obBHDoc.ExtendLog "GR", 10.5, 100.5
```

SetVisibleDepthRange (VARIANT TopDepth, VARIANT BottomDepth)

Adjusts the depth range of the data display on screen. This function does not truncate any data.

Object: Borehole

Return Value: -

Parameter:

TopDepth [optional]	New visible top depth of the borehole document in units of the current depth axis.
BottomDepth [optional]	New visible bottom depth of the borehole document in units of the current depth axis.

Remarks:

If no parameter is passed the visible depth range will be set to the minimum top and maximum bottom depth of the borehole document.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.OpenBorehole("C:\Temp\Demo.wcl")
obBHDoc.SetVisibleDepthRange 10.5, 100.5
```

CalculateBoreholeDeviation *(VARIANT PromptUser, VARIANT ConfigFile)*

Calculates borehole Azimuth, RBR and Tilt out of magnetometer and inclinometer data.

Object: Borehole

Return Value: -

Parameter:

PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[CalculateBoreholeDeviation]
MagX=Mag X
MagY=Mag Y
MagZ=Mag Z
InclX=Acc X
InclY=Acc Y
InclZ=
MagXPositive=TRUE
MagYPositive=TRUE
MagZPositive=TRUE
InclXPositive=TRUE
InclYPositive=TRUE
InclZPositive=TRUE
IsAccelerometer=TRUE
MarkerPosition=182.5
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
obBHDoc.CalculateBoreholeDeviation FALSE, "C:\Temp\Process.ini"
```

CalculateBoreholeCoordinates (VARIANT PromptUser, VARIANT ConfigFile)

Calculates the deviation path coordinates Northing, Easting and TVD.

Object: Borehole

Return Value: -

Parameter:

PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[CalculateBoreholeCoordinates]
Method=Classic Tangential
'Method=Balanced Tangential
'Method=Radius Of Curvature
Unit=m
AzimuthLog=AZI
TiltLog=TILT
NewDepthLog=TRUE
CountTVDFromLogTop=TRUE
TVDStartDepth=0
MagDeclination=11.5
EstimateErrors=FALSE
AziError=0.1
TiltError=0.1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
obBHDoc.CalculateBoreholeCoordinates FALSE, "C:\Temp\Process.ini"
```

CalculateBoreholeClosure *(VARIANT PromptUser, VARIANT ConfigFile)*

Calculates the deviation path closure distance, closure angle and DLS.

Object: Borehole

Return Value: -

Parameter:

PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[CalculateBoreholeClosure]
AzimuthLog=AZI
TiltLog=TILT
NorthingLog=NORTH
EastingLog=EAST
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
obBHDoc.CalculateBoreholeClosure FALSE, "C:\Temp\Process.ini"
```

ApplyNaturalGammaBoreholeCorrection *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Applies corrections to the count rates stored in the specified FWS, Well Log.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[BoreholeConditionCorrections]
DeadTime=7.2
EnableDeadTime=yes
EnableFactors=yes
K-Factor=2*0.00001028
FactorName1=Water Factor
FactorName2=Pipe Factor
Top1=top
Bot1=2.85
Factor1-1=1
Factor1-2=1.49
Top2=2.85
Bot2=bot
Factor2-1=1.12
Factor2-2=1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obLog = obBHDoc.ApplyNaturalGammaBoreholeCorrection ("Counts", FALSE,_
"C:\Temp\Process.ini")
```

ApplyTotalGammaCalibration *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Applies a calibration factor or equation to the values in the specified Well Log.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[BoreholeConditionCorrections]
DeadTime=7.2
EnableDeadTime=yes
EnableFactors=yes
K-Factor=2*0.00001028
FactorName1=Water Factor
FactorName2=Pipe Factor
Top1=top
Bot1=2.85
Factor1-1=1
Factor1-2=1.49
Top2=2.85
Bot2=bot
Factor2-1=1.12
Factor2-2=1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obLog = obBHDoc.ApplyTotalGammaCalibration("Counts_Cor", FALSE,_
"C:\Temp\Process.ini")
```

CalculateSpectrumTotalCount (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Extracts the total count, min, max, average or median from each spectrum trace of the specified log.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[SpectralGamma_Statistic]
Total=yes
Min=no
Max=no
Ave=no
Median=no
Channel=no
UseWindow=yes
WinLow=410
WinHigh=2850
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
obBHDoc.CalculateSpectrumTotalCount "Counts", FALSE, "C:\Temp\Process.ini"
```

ProcessSpectrumData *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Calculates window counts or concentrations according to a given calibration model.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[SpectralGamma]
OutputWindowCounts=yes
ProcessModel=C:\Temp\Test.sgm
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
obBHDoc.ProcessSpectrumData "Spectrum", FALSE, "C:\Temp\Process.ini"
```

CalculateBoreholeVolume *(VARIANT PromptUser, VARIANT ConfigFile)*

Calculates either hole or annular volume and outputs the result in various presentation formats.

Object: Borehole

Return Value: -

Parameter:

PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[VolumeProcess]
InnerDiam=CasingCaliper
OuterDiam=HoleCaliper
InnerDiamUnit=m
'InnerDiamUnit=cm
'InnerDiamUnit=mm
'InnerDiamUnit=in
'InnerDiamUnit=ft
'InnerDiamUnit=yd
OuterDiamUnit=m
VolumeUnit=litre
'VolumeUnit=cu.cm
'VolumeUnit=cu.m
'VolumeUnit=cu.ft
'VolumeUnit=cu.in
'VolumeUnit=cu.yd
AnnularVolume=yes
BottomToTop=yes
DisplayTick=yes
SmallTickFreq=1
MediumTickFreq=10
```

```
LargeTickFreq=100
DisplayNumerical=yes
NumericalFreq=5
DisplayCurve=yes
DisplayInterval=yes
IntervalRef=5
MaxDepthRange=yes
TopDepth=0
BottomDepth=1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
obBHDoc.CalculateBoreholeVolume FALSE, "C:\Temp\Process.ini"
```

FilterFWSLog (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Applys a moving average, weihted average or frequency filter to the data in an FWS Log.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[FilterFWSLog]
FilterType=MovingAverage
'FilterType=WeightedAverage
'FilterType=Frequency
FilterWidth=25
LowPass=5
LowCut=10
HighPass=25
HighCut=30
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.FilterFWSLog("Sonic", FALSE, "C:\Temp\AutoProcess.ini")
```

StackTraces (VARIANT IsSpectrum, VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Calculates an average sonic trace for the given stack interval.

Object: Borehole

Return Value: Log object

Parameter:

IsSpectrum [optional]	Boolean indicating whether the data contained in the Fws Log is sonic or spectrum data.
Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[StackTraces]
NumberOfStacks=5
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obLog = obBHDoc.StackTraces("Rx1",FALSE, CONFIGFILE)
obLog.Name = "Rx1 Stacked"
```

ReverseAmplitude *(VARIANT Log)*

Swaps the amplitude values of a FWS log (multiplies each amplitude values by -1).

Object: Borehole

Return Value: -

Parameter:

Log [optional] Can be an integer representing the log index or a string with the log name. The *Log* parameter is optional and can be empty.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.ReverseAmplitude "Sonic"
```

CorrectBadTraces *(VARIANT Log)*

Replaces NULL data traces in Image, RGB and FWS logs.

Object: Borehole

Return Value: -

Parameter:

Log [optional] Can be an integer representing the log index or a string with the log name. The *Log* parameter is optional and can be empty.

Remarks:

This funtion can be applied to Image, RGB and FWS Logs. If the the *Log* paramter is empty a log selection from a dialog box will be possible.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set Borehole = obWCAD.GetBorehole()
Borehole.CorrectBadTraces("Amplitude")
```

PickFirstArrival *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Picks the first arrival time using the standard threshold or advanced method.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[PickFirstArrival]
AdvancedMethod=TRUE
Blanking=0.0
Threshold=6.0
LargeWindow=120.0
SmallWindow=40.0
RatioThreshold=3.0
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.PickFirstArrival("Sonic", FALSE,_
 "C:\Temp\AutoProcess.ini")
```

ApplyStandOffCorrection *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Corrects the first arrival pick for the tool's stand off from the borehole wall and outputs a transit time in [m/s].

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the Well log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ApplyStandOffCorrection]
ToolSpacingUnit=m
'ToolSpacingUnit=ft
ToolSpacing=0.6
StandOffUnit=m
'StandOffUnit=in
StandOff=0.05
FluidVelocityUnit=m/s
'FluidVelocityUnit=us/m
'FluidVelocityUnit=us/ft
FluidVelocity=1440.0
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDTRx1 = obBHDoc.PickFirstArrival("RX1", FALSE, CONFIGFILE)
Set obTT = obBHDoc.ApplyStandOffCorrection(obDTRx1.Name, FALSE,_
"C:\Temp\AutoProcess.ini")
```

ExtractWindowPeakAmplitude (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Extracts the maximum amplitude found in a time window of a FWS log trace.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ExtractWindowPeakAmplitude]
WindowLength=50
PickMax=TRUE
WindowStart=DT
'WindowStart=250
```

The WindowsStart parameter can be a log title or a constant number.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.ExtractWindowPeakAmplitude("Sonic", FALSE,_
 "C:\Temp\AutoProcess.ini")
```

PickE1Arrival (VARIANT LogFws, VARIANT LogDt, VARIANT PromptUser, VARIANT ConfigFile)

Adjusts the arrival time given in *LogDt* to the next maximum or minimum amplitude in *LogFws.*

Object: Borehole

Return Value: Log Object

Parameter:

LogFws [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
LogDt [optional]	A string specifying the log name or an integer representing the index of the arrival time log (Well Log) in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[AdjustPickToExtremum]
PickPositivPolarity=TRUE
FilterWidth=5
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obDTRx1 = obBHDoc.PickFirstArrival("RX1", FALSE, CONFIGFILE)
obDTRx1.Name = "DT1"
Set obLog = obBHDoc.PickE1Arrival("RX1","DT1", FALSE, CONFIGFILE)
obLog.Name = "DT E1"
```

ExtractE1Amplitude *(VARIANT LogFws, VARIANT LogE1, VARIANT PromptUser)*

Reads the amplitude at the arrival time given in *LogE1* from the sonic log *LogFws*.

Object: Borehole

Return Value: Log Object

Parameter:

LogFws [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
LogDt [optional]	A string specifying the log name or an integer representing the index of the arrival time log (Well Log) in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obDTRx1 = obBHDoc.PickFirstArrival("RX1", FALSE, CONFIGFILE)
obDTRx1.Name = "DT1"
Set obE1 = obBHDoc.PickE1Arrival("RX1","DT1", FALSE, CONFIGFILE)
obE1.Name = "E1"
Set obE1Amp = obBHDoc.ExtractE1Amplitude("RX1", "E1", FALSE)
obE1Amp.Name = "E1 Amplitude"
```

CalculateMechanicalProperties (VARIANT LogP, VARIANT LogS, VARIANT LogDens)

Calculates

Object: Borehole

Return Value: -

Parameter:

LogP [optional] A string specifying the log name or an integer representing the index of the well log in the Borehole Document containing the compressional wave slowness (in us/m).

LogS [optional] A string specifying the log name or an integer representing the index of the well log in the Borehole Document containing the shear wave slowness (in us/m).

LogDens [optional] A string specifying the log name or an integer representing the index of the well log in the Borehole Document containing the density (in g/ccm).

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.CalculateMechanicalProperties "DTp", "DTs", "RHOb"
```

ApplySemblanceProcessing *(VARIANT PromptUser, VARIANT ConfigFile)*

Extracts the cumulative energy from reflected tube wave arrivals.

Object: Borehole

Return Value: Log object

Parameter:

PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ApplySemblanceProcessing]
Rx1_Log=Receiver 1
Rx1_Offset=0.0
Rx1_TxDistance=0.6
Rx1_Unit=m
Rx2_Log=Receiver 2
Rx2_Offset=0.0
Rx2_TxDistance=1
Rx2_Unit=m
Rx3_Log= . . .
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obLog = obBHDoc.ApplySemblanceProcessing(FALSE, CONFIGFILE)
obLog.Name = "Semblance"
```

AdjustPickToExtremum *(VARIANT LogFws, VARIANT LogPick, VARIANT PromptUser, VARIANT ConfigFile)*

Adjusts the pick given in *LogPick* to the next maximum or minimum amplitude in *LogFws*.

Object: Borehole

Return Value: Log Object

Parameter:

LogFws [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
LogPick [optional]	A string specifying the log name or an integer representing the index of the picks log (Well Log) in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[PickE1Arrival]
PickPositivPolarity=TRUE
FilterWidth=5
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obDTRx1 = obBHDoc.PickFirstArrival("RX1", FALSE, CONFIGFILE)
obDTRx1.Name = "DT1"
Set obE1 = obBHDoc.AdjustPickToExtremum("RX1","DT1", FALSE, CONFIGFILE)
obE1.Name = "E1"
```

ProcessReflectedTubeWave (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Extracts the cumulative energy from reflected tube wave arrivals.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the FWS log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ProcessReflectedTubeWave]
Side=both
'Side=upper
'Side=lower
Offset=0.0 'measured in us
Blanking=0.0 'measured in us
FluidSlowness=696.0 'measured in us/m
TxFrequency=15000.0 'measured in Hz
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBHDoc = obWCAD.GetBorehole()
Set obLog = obBHDoc.ProcessReflectedTubeWave("RX1", FALSE, CONFIGFILE)
obLog.Name = "Energy"
```

FilterImageLog (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Filters the data values of a Image Log. Displays the process dialog box if no parameter is passed.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[FilterImageLog]
FilterType=Average
'FilterType=Median
'FilterType=Despiking
FilterWidth=3
Filterheight=3
HighCutLimit=75
LowCutLimit=15
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.FilterImageLog("Amplitude", FALSE,_
 "C:\Temp\AutoProcess.ini")
```

ApplyConditionalTesting (VARIANT LogIf, VARIANT LogThen, VARIANT PromptUser, VARIANT ConfigFile)

Applies conditional testing (If Then Else) to image log values and returns a new log

Object: Borehole

Return Value: Log object

Parameter:

LogIf [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document. (IF *LogIf* …, THEN *LogThen* …)
LogThen [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document. (IF *LogIf* …, THEN *LogThen* …)
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

This funtion tests the values of the log *LogIf* against the values of another image log, constant value or against the NULL value. According to the test result the values of the *LogThen* will be modified and a copy of the *LogThen* with the modidies values will be returned.

The configuration file must contain the following section name and parameters:

```
[ApplyConditionalTesting]
'Condition= !=
'Condition= <=
'Condition= >=
Condition= <
'Condition= >
'Condition= ==
ConditionValue=100
ThenValue=NULL
```

```
ElseValue=Amplitude
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.ApplyConditionalTesting ("Travel Time", "Amplitude",_
 FALSE,"C:\Temp\AutoProcess.ini")
obLog.Name = "Tested"
```

NormalizeImage *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Normalizes the amplitudes of an Image Log. Displays the process dialog box if no parameter is passed.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[NormalizeImage]
Mode=Static
'Mode=Dynamic_1D
'Mode=Dynamic_2D
'Mode=HighPass
WindowHeight=0.3
WindowWidth=5
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = Borehole.NormalizeImage("Amplitude",FALSE,"C:\Temp\AutoProcess.ini")
```

ExtractImageLogStatistics (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Extracts minimum, maximum, average, median or a percentage of values fulfilling an optional condition from each image trace.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ExtractImageLogStatistics]
Minimum=FALSE
Maximum=FALSE
Average=FALSE
Median=TRUE
Percentage=FALSE
Condition=0
Value1=100
Value2=200
MaxDepthRange=TRUE
TopDepth=0
BottomDepth=10
```

For the condition flag the following option are available:
0 = no condition, 1 = lower than, 2 = larger than, 3 = lower or equal, 4 = larger or equal, 5 = equal, 6 = not equal, 7 = between, 8 = between or equal.

If the condition flag has been set to 7 or 8 valid entries for Value1 and Value2 are expected.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.ExtractImageLogStatistics "Amplitude", FALSE,_
 "C:\Temp\AutoProcess.ini"
Set obLog = obBorehole.Log("Amplitude - median")
obLog.Name = "Median"
```

CalculateApparentMetalLoss (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Calculates an apparent metal loss value for each trace of radius values stored in an image log.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[CalculateApparentMetalLoss]
InternalPipeRadius=1.9
ExternalPipeRadius=2.2
```

The units of the internal/external pipe radius values must be the same as the unit of the measured radius.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.CalculateApparentMetalLoss ("Multi Finger Caliper",_
 FALSE,_ "C:\Temp\AutoProcess.ini")
obLog.Name = "Metal Loss"
```

CalculateFluidVelocity *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Estimates a fluid velocity from travel time measurements and given calibration points.

Object: Borehole

Return Value: Log object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

This configuration file must contain the following section name and parameters:

```
[CalculateFluidVelocity]
TravelTimeUnit=0.1
ToolRadius=19
'TimeWindow=TWnd
TimeWindow=74
CalibrationPoint1=20.44, 96
CalibrationPoint2=36.85, 96
'CalibrationPoint3=…
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.CalculateFluidVelocity("TravelTime", FALSE,_
 "C:\Temp\AutoProcess.ini")
```

CalculateAcousticCaliper (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Calculates borehole radius and caliper values from acoustic traveltime measurements.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

This configuration file must contain the following section name and parameters:

```
[CalculateAcousticCaliper]
TravelTimeUnit=0.1
CaliperUnit=mm
'CaliperUnit=cm
'CaliperUnit=in
ToolRadius=19
TimeWindow=TWnd
'TimeWindow=74
FluidVelocity=FVelocity
'FluidVelocity=1448
FluidVelocityUnit= m/s
'FluidVelocityUnit= km/s
'FluidVelocityUnit= m/ms
'FluidVelocityUnit= m/us
'FluidVelocityUnit= ft/s
'FluidVelocityUnit= ft/ms
'FluidVelocityUnit= ft/us
'FluidVelocityUnit= s/km
```

```
'FluidVelocityUnit= s/m
'FluidVelocityUnit= ms/m
'FluidVelocityUnit= us/m
'FluidVelocityUnit= s/ft
'FluidVelocityUnit= ms/ft
'FluidVelocityUnit= us/ft
CurveOutput=FALSE
ImageOutput=TRUE
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.CalculateAcousticCaliper "Travel Time", FALSE,_
 "C:\Temp\AutoProcess.ini"
```

CentralizeImageData *(VARIANT Log)*

Corrects data for de-centralization effects and outputs a new image log.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional] A string specifying the log name or an integer representing the index of the log in the Borehole Document.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.CentralizeImageData ("Travel Time")
obLog.Name = "TT Centralized"
```

CalculateCasingThickness *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Calculates thickness values for a casing pipe from acoustic thickness traveltime measurements.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[CalculateCasingThickness]
TravelTimeUnit=0.01
ThicknessUnit=mm
'ThicknessUnit=cm
'ThicknessUnit=in
SteelVelocity=Vel
'SteelVelocity=5200
SteelVelocityUnit= m/s
'SteelVelocityUnit= km/s
'SteelVelocityUnit= m/ms
'SteelVelocityUnit= m/us
'SteelVelocityUnit= ft/s
'SteelVelocityUnit= ft/ms
'SteelVelocityUnit= ft/us
'SteelVelocityUnit= s/km
'SteelVelocityUnit= s/m
'SteelVelocityUnit= ms/m
'SteelVelocityUnit= us/m
```

```
'SteelVelocityUnit= s/ft
'SteelVelocityUnit= ms/ft
'SteelVelocityUnit= us/ft
CurveOutput=FALSE
ImageOutput=TRUE
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.CalculateCasingThickness "Travel Time Thickness", FALSE,_
 "C:\Temp\AutoProcess.ini"
```

OrientImageToNorth (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Rotates an image log to north according to the deviation channels provided.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the image log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[OrientImageToNorth]
MagX=Mag X
MagY=Mag Y
MagZ=Mag Z
InclX=Acc X
InclY=Acc Y
InclZ=
MagXPositive=TRUE
MagYPositive=TRUE
MagZPositive=TRUE
InclXPositive=TRUE
InclYPositive=TRUE
InclZPositive=TRUE
IsAccelerometer=TRUE
MarkerPosition=180
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.OrientImageToNorth "Amplitude", FALSE, "C:\Temp\AutoProcess.ini"
```

OrientImageToHighside (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Rotates an image log to high side according to the deviation channels provided.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the image log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[OrientImageToHighside]
InclX=Acc X
InclY=Acc Y
InclZ=
InclXPositive=TRUE
InclYPositive=TRUE
InclZPositive=TRUE
IsAccelerometer=TRUE
MarkerPosition=180
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.OrientImageToHighside "Amplitude", FALSE, "C:\Temp\AutoProcess.ini"
```

RotateImage *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Rotates the image data by adding an angle (clockwise) or subtracting a value to/from the data position.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the image log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[RotateImage]
RotateBy=1.2
'RotateBy=Log
RotateClockwise=TRUE
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.RotateImage "Amplitude", FALSE, "C:\Temp\AutoProcess.ini"
```

MirrorImage *(VARIANT Log)*

Rearranges the data within an image log so that the data appears mirrored when compared to the original iamge.

Object: Borehole

Return Value: -

Parameter:

Log [optional] A string specifying the log name or an integer representing the index of the log in the Borehole Document.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.MirrorImage "Amplitude"
```

AdjustImageBrightnessAndContrast (VARIANT Log, VARIANT PromptUser)

Adjusts the brightness and contrast values for the data in an RGB log.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.

Remarks:

If the *PromptUser* parameter is set to FALSE the new brightness and contrast values will be determined automatically.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.AdjustImageBrightnessAndContrast "Image"
```

ApplyStructureApparentToTrueCorrection (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Corrects the apparent azimuth and dip angles in a Structure log.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ApplyApparentToTrueCorrection]
AzimuthLog=AZI
TiltLog=TILT
ReferenceIsNorth=TRUE
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.ApplyStructureApparentToTrueCorrection("Structure App",_
 FALSE, "C:\Temp\AutoProcess.ini")
obLog.Name = "Structure True"
```

ApplyStructureTrueToApparentCorrection (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Recalculates the apparent azimuth and dip angles in a Structure log from the true structure angles.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ApplyTrueToApparentCorrection]
AzimuthLog=AZI
TiltLog=TILT
ReferenceIsNorth=TRUE
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.ApplyStructureTrueToApparentCorrection("Structure True",_
 FALSE, "C:\Temp\AutoProcess.ini")
obLog.Name = "Structure Apparent"
```

RemoveStructuralDip (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Removes a given dip from the data in a structure log and recalculates Dip and Azimuth angles.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the structure log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[RemoveStructuralDip]
'Azimuth=AziLog
Azimuth=45
'Dip=DipLog
Dip=10
MaxDepthRange=TRUE
TopDepth=0
BottomDepth=1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.RemoveStructuralDip("Test", FALSE,_
 "C:\Temp\AutoProcess.ini")
```

RecalculateStructureDip (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Corrects the dip angle data within a structure log for new caliper settings.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the structure log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[RecalculateStructureDip]
'Caliper=CalLog
Caliper=200
CaliperUnit=mm
'CaliperUnit=in
MaxDepthRange=TRUE
TopDepth=0
BottomDepth=1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.RecalculateStructureDip "Test", FALSE, "C:\Temp\AutoProcess.ini"
```

RecalculateStructureAzimuth *(VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)*

Recalculates the Azimuth data within a structure log.

Object: Borehole

Return Value: -

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the structure log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[RecalculateStructureAzimuth]
'Angle=Log
Angle=45
RotateClockwise=TRUE
MaxDepthRange=TRUE
TopDepth=0
BottomDepth=1
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.RecalculateStructureAzimuth "Test", FALSE, "C:\Temp\AutoProcess.ini"
```

ExtractStructureIntervalStatistic (VARIANT Log, VARIANT PromptUser, VARIANT ConfigFile)

Allows to determine statistical values (e.g. frequency of dips) per interval from a structure log.

Object: Borehole

Return Value: Log Object

Parameter:

Log [optional]	A string specifying the log name or an integer representing the index of the structure log in the Borehole Document.
PromptUser [optional]	Boolean specifying whether dialog boxes are displayed to interact with the user or not. If set to FALSE the processing parameters will be read out of the specified configuration file. The WellCAD.ini will be read or default values are used if no configuration file has been specified.
ConfigFile [optional]	The full path and filname of the configuration file (e.g. C:\Temp\Process.ini).

Remarks:

The configuration file must contain the following section name and parameters:

```
[ExtractStructureIntervalStatistic]
'Reference=Log
Reference=5.0
BlockingMode=Counts
'BlockingMode=Freq_m
'BlockingMode=Freq_ft
'BlockingMode=Ave_Dip
'BlockingMode=Ave_Azi
FilterOnDip=FALSE
DipLow=0
DipHigh=90
FilterOnAzimuth=FALSE
AziLow=0
AziHigh=360
FilterOnAperture=TRUE
ApertureLow=
ApertureHigh=
FilterOnAttributes=TRUE
```

```
Attribute1Name=Type
Attribute1Value1=RF
Attribute1Value2=MB
Attribute2Name=Remark
Attribute2Value1=Open
Attribute2Value2=Loose
```

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.ExtractStructureIntervalStatistic("Test", FALSE,_
 "C:\Temp\AutoProcess.ini")
```

BSTR ***Name***

Use this property to get or set the title of a borehole document.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.Name = "Automation Test"
```

Float ***TopDepth***

Use this property to get the top depth of a borehole document in units of the master depth axis (independent of any Depth Log being the current reference axis).

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Top = obBorehole.TopDepth
```

Float ***BottomDepth***

Use this property to get the bottom depth of a borehole document in units of the master depth axis (independent of any Depth Log being the current reference axis).

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Bottom = obBorehole.BottomDepth
```

Short ***NbOfLogs***

Use this property to get the number of logs contained in the borehole document.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Logs = obBorehole.NbOfLogs
```

*IDispatch** **Header**

Use this property to retrieve header of a borehole document. The Header object allows populationof the dynamic text fields in a borehole document header. You can also assign a header object from another borehole document to this property.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obHeader = obBorehole.Header
obHeader.ItemText "Well", "12345"
obHeader.ItemText "Company", "Go-For-Log"
. . .
```

*IDispatch** **Page**

Use this property to retrieve a Page object of a borehole document. Page objects allow setting properties for the document print out (similar to the Margin options in the Page Setup dialog box).

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
'Set paper mode to fanfold
obPage.PaperMode = 1
. . .
```

*IDispatch** **Depth**

Use this property to retrieve an object of the master depth azis of a borehole document. Depth objects allow setting properties for the master depth axis of a borehole document such as scale, unit, decimals,….

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
'Set depth unit
obPage.PaperMode = 1
. . .
```

Bool **AutoUpdate**

Use this property to get or set the update flag. If set to TRUE (which is the default) the screen will be updated each time a data value has been added or any other action took place.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
obBorehole.AutoUpdate = FALSE
. . .
```

Short ***VersionMajor***

Use this property to retrieve the major version number of your WellCAD application.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.NewBorehole()
Wscript.Echo obBorehole.VersionMajor
```

Short ***VersionMinor***

Use this property to retrieve the minor version number of your WellCAD application.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.NewBorehole()
Wscript.Echo obBorehole.VersionMinor
```

Short ***VersionBuild***

Use this property to retrieve the build number of your WellCAD application.

Object: Borehole

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.NewBorehole()
MajorVersion = obBorehole.VersionMajor
MinorVersion = obBorehole.VersionMinor
BuildNumber = obBorehole.VersionBuild
WScript.Echo "This is WellCAD version" & MajorVersion & "." & MinorVersion_
& " build " & BuildNumber
```

IDispatch ***ODBC***

Use this property to retrieve an ODBC Object. The ODBC object allows to pass SQL statements to the SQL interpretor within WellCAD (only accessable if the ODBC add-on module is activated).

Object: Borehole

Example:

```
Set obBHDoc = obWCAD.NewBorehole()
Set obODBC = obBHDoc.ODBC
obODBC.InterpretSQLStatement("$dsn = OpenDatabase(Test, Admin)")
obODBC.InterpretSQLStatement("$Log = select Depth, Value from tblGamma")
obODBC.InterpretSQLStatement("LoadWellLog($Log)")
obODBC.InterpretSQLStatement("Close($Log)")
obODBC.InterpretSQLStatement("Close($dsn)")
```

BSTR ***Name***

Use this property to get or set the title of a log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.Name = "GR Index"
```

BSTR ***TitleComment***

Use this property to get or set the comment of a log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.Name = "GR Index"
obLog.TitleComment = "Calculated Gamma Ray Index"
```

BSTR ***LogUnit***

Use this property to get or set the unit string of a log. Only applicable to supported log types.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.Name = "GR corrected"
obLog.LogUnit = "CPS"
```

Short ***Type***

Use this property to get the log type index (e.g. Well Log = 1, Mud Log = 3)

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
Index = obLog.Type
```

Float ***LeftPosition***

Set or get the position of the left border of the log column in percent (0 < position < 1).

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.Name = "GR corrected"
obLog.LeftPosition = 0.1
```

Float ***RightPosition***

Set or get the position of the right border of the log column in percent (0 < position < 1).

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.Name = "GR corrected"
obLog.RightPosition = 0.2
```

Float ***TopDepth***

Get the top depth of the log in current master depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
Start = obLog.TopDepth
```

Float ***BottomDepth***

Set or get the bottom depth of the log in current master depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
Stop = obLog.BottomDepth
```

Float ***DataMin***

Use this property to get the minimum of the log data.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
Min = obLog.DataMin
```

Float ***DataMax***

Use this property to get the maximum of the log data.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
Max = obLog.DataMax
```

Float ***ScaleLow***

Set or get the low scale value of the log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.ScaleLow = obLog.DataMin
```

Float ***ScaleHigh***

Set or get the high scale value of the log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.ScaleHigh = obLog.DataMax
```

Short ***PenStyle***

Set or get the pen style used in a Well or Mud Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.PenStyle = 0
```

Long ***PenColor***

Set or get the pen color used in a Well or Mud Logs as RGB color amplitude.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.PenColor = 0
```

Short ***PenWidth***

Set or get the pen width used in a Well or Mud Log in 1/10 mm.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.PenWidth = 5
```

Short ***Shading***

Set or get the shading option used in a Well or Mud Logs (0 = none, 1 = left, 2 = right)

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.Shading = 1
```

Short ***Filter***

Set or get the width (in samples) of the display filter for Well or Mud Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.Filter = 5
```

Bool ***GridEnable***

Set or get the flag to display the vertical grid in Well or Mud Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.GridEnable = TRUE
```

Short ***ScaleMode***

Set or get the scale type for the data display of Well or Mud Logs. (0 = linear, 1 = logarithmic)

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.ScaleMode = 0
```

Float ***GridSpacing***

Set or get the vertical grid spacing for Well or Mud Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.GridSpacing = 20
```

Bool ***ScaleReversed***

Set or get the flag to reverse the data display scale of Well or Mud Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.ScaleReversed = FALSE
```

BSTR ***Formula***

Set or get the formula string used in a Formula Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.Name = "GR corrected"
obLog.Formula = "{GR}/ (1 - {GR}* 0.0000072)"
```

Long ***NbOfData***

Retrieve the number of data points in a log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
Samples = obLog.NbOfData
```

Short ***Style***

Set or get the data display style for Mud (1 = fixed bar, 2 = dynamic bar, 3 = line) and Engineering Logs (0 = full, 1 = left, 2 = right)

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Permeability")
obLog.Style = 1
```

Short ***FixedBarWidth***

Set or get the fixed bar width in 1/10 mm for Mud Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Permeability")
obLog.Style = 5
```

IDispatch ***LithoDictionary***

Set or get the lithology library as LithoDictionary object.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obLithoLib = obLog.LithoDictionary
```

IDispatch ***Font***

Set or get the lithology library as Font object.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Remarks")
Set obFont = obLog.Font
```

Float ***NullValue***

Set or get the No Data value for a log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
obLog.NullValue = -999.25
```

Float ***SampleRate***

Set or get the sample interval of a log in current master depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
SmpInt = obLog.SampleRate
```

Float ***GroundDepth***

Set or get the ground depth for Engineering Logs in current master depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(9)
obLog.Name = "Well Layout"
obLog.GroundDepth = 12.5
```

Long ***NbDrillItem***

Set or get the number of drill item in a Engineering Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(9)
obLog.Name = "Well Layout"
obLog.NbDrillItem = 3
```

Float ***DiameterHigh***

Set or get the max diameter scaling value for Engineering Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(9)
obLog.Name = "Well Layout"
obLog.DiameterHigh = 0.5
```

Long ***BackgroundColor***

Set or get the background color for the Engineering Log as RGB color amplitude.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(9)
obLog.Name = "Well Layout"
obLog.BackgroundColor = 0
```

Short ***BackgroundStyle***

Set or get the background style for the Engineering Log (0 = none, 1 = solid, 2 = hatch).

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(9)
obLog.Name = "Well Layout"
obLog.BackgroundStyle = 0
```

Short ***BackgroundHatchStyle***

Set or get the background hatch style for the Engineering Log (0 = horizontal, 1 = vertical, 2 = forward diagonal, 3 = backward diagonal, 5 = cross, 6 = diagonal cross).

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(9)
obLog.Name = "Well Layout"
obLog.BackgroundHatchStyle = 3
```

Long ***NbOfEqpItem***

Get the number of equipment items in the Engineering Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
NbItems = obLog.NbOfEqpItem
```

Long ***TraceLength***

Get or set the length of a data trace in Image, RGB and FWS Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Travel Time")
NbTraceSamples = obLog.TraceLength
```

Float ***TraceSampleRate***

Get or set the trace sample interval in Image, RGB and FWS Logs.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Travel Time")
TraceSampleInterval = obLog.TraceSampleRate
```

Float ***TraceOffset***

Get or set the offset of a data trace in the FWS Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Sonic")
Offset = obLog.TraceOffset
```

Float ***ApertureUnit***

Get or set the unit the aperture in the Structure Log data is measured in (0.001 when in mm or 0.00254 when measured in 1/10 inches).

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(6)
obLog.ApertureUnit = 0.001
```

Float ***CaliperUnit***

Get or set the unit the caliper in the Structure Log data is measured in (0.001 when in mm or 0.0254 when measured in inches)

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(6)
obLog.CaliperUnit = 0.001
```

SetPosition *(float Left, float Right)*

Call this function to set the position of the left and right log border on the borehole document.

Object: Log

Return Value: -

Parameter:

Left	A value between 0 and 1 defining the position of the left log border.
Right	A value between 0 and 1 defining the position of the right log border.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.SetPosition(0.1,0.2)
```

DoSettingsDialog ()

Calling this function will open the main settings dialog box of the corresponding log.

Object: Log

Return Value: bool

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(2)
obLog.DoSettingsDialog()
```

RemoveLithoBed (Long Index)

This function will remove the specified lithology bed from the log.

Object: Log

Return Value: -

Parameter:

Index The index of the lithobed to be removed from the log.

Remarks:

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Litho")
ObLog.RemoveLithoBed 0
```

RemoveLithoBedAtDepth (float Depth)

Removes the lithology box from the Litho log at the specified depth.

Object: Log

Return Value: -

Parameter:

Depth The depth in current units at which the comment box will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Litho")
obLog.RemoveLithoBedAtDepth 10.5
```

InsertNewLithoBed (float TopDepth, float BottomDepth, BSTR LithoCode, float Value, float Position)

Inserts a new lithology bed into a Litho Log.

Object: Log

Return Value: LithoBed Object

Parameter:

TopDepth	The top depth of the new lithology interval in current depth units.
BottomDepth	The bottom depth of the new lithology interval in current depth units.
LithoCode	A String containing the lithology code identifying the pattern or symbol loaded from the attached library.
Value	Value between 0 and 1 that links a numerical value (e.g. hardness) to the lithology bed.
Position	The horizontal position within the log column if non repeated symbols are displayed (value betwenn 0 and 1).

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(7)
Set obLib = obLog.AttachLithoDictionary("C:\Temp\Litho.LTH")
obLog.InsertNewLithoBed 10, 12, Sst, 0.75, 0.5
```

AttachLithoDictionary *(BSTR FileName)*

Attaches a new symbol or pattern library (*.LTH file) to Litho, CoreDesc, Strata, Analysis or Percentage Log.

Object: Log

Return Value: LithoDictionary Object

Parameter:

FileName A string specifying path and file name of the LTH library to attach.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(7)
Set obLib = obLog.AttachLithoDictionary("C:\Temp\Litho.LTH")
```

InsertNewCommentBox *(float TopDepth, float BottomDepth, BSTR Text)*

Inserts a new box with the specified text into the Comment Log.

Object: Log

Return Value: Comment Box object

Parameter:

TopDepth	The top of the box in current depth units.
BottomDepth	The bottom of the box in current depth units.
Text	A String containing the text to be displayed in the new box.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(8)
Set obBox = obLog.InsertNewCommentBox(10.5, 15.5, "Hello World!")
```

RemoveCommentBox (long Index)

Removes the comment box from the Comment log at the specified index.

Object: Log

Return Value: -

Parameter:

Index The index at which the comment box will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Remarks")
obLog.RemoveCommentBox 0
```

RemoveCommentBoxAtDepth (float Depth)

Removes the comment box from the Comment log at the specified depth.

Object: Log

Return Value: -

Parameter:

Depth The depth in current units at which the comment box will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Remarks")
obLog.RemoveCommentBoxAtDepth 10.5
```

InsertNewStructure *(float Depth, float Azimuth, float Dip, Short Category, BSTR Description)*

Inserts a new structure into the Structure Log. This function supported the old Structure Log data format. Beginning with WellCAD 4.2 the Category and Description parameters will be added to the attributes Type and Description.

Object: Log

Return Value: Structure Object

Parameter:

Depth	The depth in current units of the structure to be inserted.
Azimuth	The azimuth angle measured in degree.
Dip	The dip angle of the structure measured in degree.
Category	Code identifying the structure.
Description	String containing the text describing the structure.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(6)
obLog.InsertNewStructure 10.5, 270.0, 10.0, 105, "joint"
```

InsertNewStructureEx *(float Depth, float Azimuth, float Dip, float Aperture)*

Inserts a new structure into the Structure Log supporting the new Structure Log data format (since WellCAD v4.2).

Object: Borehole

Return Value: Structure Object

Parameter:

Depth	The depth in current units of the structure to be inserted.
Azimuth	The azimuth angle measured in degree.
Dip	The dip angle of the structure measured in degree.
Aperture	Aperture of a fracture expressed in meter.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(6)
obLog.InsertNewStructure 10.5, 270.0, 10.0, 0.005
```

RemoveStructure *(long Index)*

Removes a Structure at the specified index from the Structure Log.

Object: Log

Return Value: -

Parameter:

Index The index at which the structure will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
obLog.RemoveStructure 0
```

RemoveStructureAtDepth *(float Depth)*

Removes a Structure at the specified depth in current units from the Structure Log.

Object: Log

Return Value: -

Parameter:

Depth The depth at which the structure will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
obLog.RemoveStructureAtDepth 0.5
```

InsertNewDrillItem *(float BottomDepth, float Diameter)*

Inserts a new drill item into the Engineering Log.

Object: Log

Return Value: DrillItem Object

Parameter:

BottomDepth The bottom depth of the drill in current depth units.

Diameter The diameter of the drill.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = obLog.InsertNewEqpItem(75.2, 96)
```

RemoveDrillItem *(long Index)*

Removes the drillitem at the spcified index from the Engineering Log.

Object: Log

Return Value: -

Parameter:

Index The index at which the drill item will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
obLog.RemoveDrillItem 0
```

InsertNewEqpItem *(float TopDepth, float BottomDepth, BSTR Name)*

Inserts a new equipment item of the specified name and depth interval of the Engineering Log.

Object: Log

Return Value: Equipment Item object

Parameter:

TopDepth	The top depth of the equipment item interval in current units.
BottomDepth	The bottom depth of the equipment item interval in current units.
Name	The name (code) of the equipment item to be inserted. Possible item are: `PlainCasing, WireWoundCasing, SlottedCasing, PerforatedCasing, Centralizer, Shoe, Packer, Water, Wedge, HeadWorks, Tranducer, Gauge, Cement, Gravel, NormalThread, ReverseThread, Plug.`

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = obLog.InsertNewEqpItem(0, 8.8, "PlainCasing")
```

RemoveEqpItem (long Index)

Removes an equipment item at the specified depth in dex from the Engineering Log.

Object: Log

Return Value: -

Parameter:

Index The index at which the item will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
obLog.RemoveEqpItem 0
```

InsertData *(long Index, float Value)*

Inserts a new data point at the specified index into a Well Log.

Object: Log

Return Value: -

Parameter:

Index	The index at which the new data point will be inserted. If necessary existing data point will be shifted. The index must be lower or equal to the number of data points in the log.
Value	The new data value.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Porosity")
obLog.InsertData 0, 14.5
```

RemoveData *(long Index)*

Removes the data point at the specified index from a Mud or Well Log. (For Well Logs the data value will be set to NULL.)

Object: Log

Return Value: -

Parameter:

Index The index at which the data point will be removed from the log.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Porosity")
obLog.RemoveData 0
```

InsertDataAtDepth *(float Depth, float Value)*

Inserts a new data point at the specified depth into a Well Log.

Object: Log

Return Value: -

Parameter:

Depth	Depth in current units at which the new data point should be inserted. If necessary existing datapoints will be shifted. The function fails if the added depth does not maintain the constant sample rate of the Well Log.
Value	The new data value.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Porosity")
obLog.DataAtDepth obLog.TopDepth, 14.5
```

RemoveDataAtDepth (float Depth)

Removes the data point at the specified depth from a Mud or Well Log. (For Well Logs the data value will be set to NULL.)

Object: Log

Return Value: -

Parameter:

Depth The depth in current units at which the data point should be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("CorePorosity")
obLog.RemoveDataAtDepth obLog.BottomDepth
```

Float ***Data****(long Index)*

Retrieve the data value at the specified index from a Well, Mud, Interval or Depth Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
NbData = obLog.NbOfData
For i = 0 To NbData-1
       Value = obLog.Data(i)
Next
```

Data*(long Index, float Value)*

Sets a new data value of a Well, Mud, Interval or Depth Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("GR")
NbData = obLog.NbOfData
For i = 0 To NbData-1
       obLog.Data i,-999
Next
```

Float ***DataAtDepth****(float Depth)*

Retrieves the data value at the specified depth from a Well, Mud, Interval or Depth Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Northing")
Value = obLog.DataAtDepth(obLog.BottomDepth)
```

DataAtDepth*(float Depth, float Value)*

Sets a new data value of a Well, Mud or Depth Log at the specified depth.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log()
obLog.DataAtDepth 0, -999
```

*Float **TraceData**(long DepthIndex, long TraceIndex)*

Retrieves the data value at the specified depth and position indizes from an Analysis, Percentage, FWS, Image or RGB Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Amplitude")
LastValue = obLog.TraceData(obLog.NbOfData - 1, obLog.TraceLength - 1)
```

***TraceData**(long DepthIndex, long TraceIndex, float Value)*

Sets the data value at the specified depth and position indizes within the trace of an Analysis, Percentage, FWS or Image Log

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Amplitude")
obLog.TraceData(obLog.NbOfData - 1, obLog.TraceLength - 1, -999)
```

*Float **TraceDataAtDepth**(float Depth, float Positon)*

Retrieves the data value at the specified depth and position within the trace of an Analysis, Percentage, FWS, Image or RGB Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("RX1")
FirstValue = obLog.TraceData(obLog.TopDepth, 0)
```

***TraceDataAtDepth**(float Depth, float Position, float Value)*

Sets the data value at the specified depth and position within the trace of an Analysis, Percentage, FWS or Image Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("RX1")
obLog.TraceData obLog.TopDepth, 0, -999
```

IDispatch ***LithoBed****(long Index)*

Returns a LithoBed object at the specified index from a Lithology Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Litho")
Set obFirstBed = obLog.LithoBed(0)
```

LithoBed*(long Index, IDispatch Value)*

Sets a LithoBed object at the specified index within a Lithology Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog1 = obBorehole.Log("Litho")
Set obFirstBed = obLog1.LithoBed(0)
Set obLog2 = obBorehole.Log("LithoCopy")
ObLog2.LithoBed 0, obFirstBed
```

IDispatch ***LithoBedAtDepth****(float Depth)*

Returns a LithoBed object at the specified depth from a Lithology Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Litho")
Set obFirstBed = obLog.LithoBed(obLog.TopDepth)
```

LithoBedAtDepth*(float Depth, IDispatch Value)*

Sets a LithoBed object at the specified index within a Lithology Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog1 = obBorehole.Log("Litho")
Set obFirstBed = obLog1.LithoBed(obLog1.TopDepth)
Set obLog2 = obBorehole.Log("LithoCopy")
ObLog2.LithoBed obLog2.TopDepth, obFirstBed
```

IDispatch ***CommentBox****(long Index)*

Retrieve a Comment Box Item object from the Comment Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Remarks")
Set obBox = obLog.CommentBox(0)
```

IDispatch ***CommentBoxAtDepth****(float Depth)*

Retrieve a Comment Box object from the Comment Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Remarks")
Set obBox = obLog.CommentBoxAtDepth(10.5)
```

IDispatch ***Structure****(long Index)*

Retrieve a Structure object from the Structure Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = obLog.Structure(0)
```

IDispatch ***StructureAtDepth****(float Depth)*

Retrieve a Structure object from the Structure Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Structure")
Set obItem = obLog.StructureAtDepth(10.5)
```

IDispatch ***DrillItem****(long Index)*

Retrieve a Drill Item object from the Engineering Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = obLog.DrillItem(0)
```

IDispatch ***DrillItemAtDepth****(float Depth)*

Retrieve a Drill Item object from the Engineering Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = obLog.DrillItemAtDepth(10.5)
```

IDispatch ***EqpItem****(long Index)*

Retrieve an Equipment Item object at the specified index from the Engineering Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = obLog.EqpItemItem(0)
```

IDispatch ***IntervalItem****(long Index)*

Retrieve an Interval Item object from the Interval Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("test")
Set obItem = obLog.IntervalItem(0)
```

IDispatch ***IntervalItemAtDepth****(float Depth)*

Retrieve an Interval Item object from the Interval Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
Set obItem = obLog.IntervalItemAtDepth(10.5)
```

IDispatch ***SchmitBox****(long Index)*

Retrieve a Schmit Box object from the Polar & Rose Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
Set obBox = obLog.SchmitBox(0)
```

IDispatch ***SchmitBoxAtDepth****(float Depth)*

Retrieve a Schmit Box object from the Polar & Rose Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
Set obBox = obLog.SchmitBoxAtDepth(10.5)
```

IDispatch ***CrossBox****(long Index)*

Retrieve a Cross Box object from the Cross Section Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("XSection")
Set obBox = obLog.CrossBox(0)
```

IDispatch ***CrossBoxAtDepth****(float Depth)*

Retrieve a Cross Box object from the Cross Section Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("XSection")
Set obBox = obLog.CrossBoxAtDepth(10.5)
```

IDispatch ***FossilItem****(long Index)*

Retrieve a Fossil Item object from the CoreDesc Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItem(0)
```

IDispatch ***FossilItemAtDepth****(float Depth)*

Retrieve a Fossil Item object from the CoreDesc Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItemAtDepth(10.5)
```

IDispatch ***StackItem****(long Index)*

Retrieve a Stack Item object from the Stacking Pattern Log at the specified index.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
Set obItem = obLog.StackItem(0)
```

IDispatch ***StackItemAtDepth****(float Depth)*

Retrieve a Stack Item object from the Stacking Pattern Log at the specified depth in current depth units.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
Set obItem = obLog.StackItemAtDepth(10.5)
```

BSTR ***ComponentName****(long Column)*

Get the name of the components used in the specified column of a Percentage or Analysis Log

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Analysis")
Component1 = obLog.ComponentName 0
```

ComponentName*(long Column, BSTR Name)*

Set the name (e.g. litho code) for the component used in the specified data column of a Percentage or Analysis Log.

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Analysis")
obLog.ComponentName 0, "Sst"
```

IDispatch ***Marker*** *(long Index)*

Retrieve a marker box object from the Marker Log at the specified index

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
Set obMarker = obLog.Marker(0)
```

IDispatch ***MarkerByName*** *(BSTR Name)*

Retrieve a marker box object of the specified Name from the Marker Log

Object: Log

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
Set obMarker = obLog.MarkerByName("Reservoir")
```

InsertTrace *(long Index)*

Inserts a new data trace into an Image, FWS or Analysis Log.

Object: Log

Return Value: -

Parameter:

Index	The index at which the trace should be added. Must be lower or equal the number of data traces within the log. If necessary existing traces will be shifted.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(14)
obLog.InsertTraceAtDepth 0
```

InsertTraceAtDepth (float Depth)

Inserts a new data trace into an Image, FWS or Analysis Log.

Object: Log

Return Value: -

Parameter:

Depth	The depth at which the new data trace will be inserted. If necessary existing traces will be shifted. The function fails if the specified depth lies without the constant sample rate of the log.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(14)
obLog.InsertTraceAtDepth 9.5
```

InsertNewIntervalItem *(float TopDepth, float BottomDepth, float Value)*

Insert a new data interval into an Interval Log.

Object: Log

Return Value: IntervalItem Object

Parameter:

TopDepth	The top depth of the data interval in current depth units.
BottomDepth	The bottom depth of the data interval in current depth units.
Value	The data value.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(13)
Set obInterval = obLog.InsertNewIntervalItem(9.5, 10, 100)
```

RemoveIntervalItem (long Index)

Removes an data interval at the specified index.

Object: Log

Return Value: -

Parameter:

Index The index of the data interval to be removed from the Interval Log.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
obLog.RemoveIntervalItem 0
```

RemoveIntervalItemAtDepth (float Depth)

Removes an data interval at the specified depth.

Object: Log

Return Value: -

Parameter:

Depth	The depth of the data interval in current units that should be removed from the Interval Log.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
obLog.RemoveIntervalItemAtDepth 10.5
```

InsertNewCrossBox (float TopDepth, float BottomDepth)

Inserts a new box into the Cross Section Log.

Object: Log

Return Value: CrossBox Object

Parameter:

TopDepth The top depth of the cross section box incurrent depth units.

Bottom The bottom depth of the cross section box in current depth units.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(21)
Set obBox = obLog.InsertNewFossilItem(9.5, 10)
```

RemoveCrossBox (long Index)

Removes a box from the Cross Section Log at the specified index.

Object: Log

Return Value: -

Parameter:

Index The zero based index of the box to be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("XSection")
obLog.RemoveCrossBox 0
```

RemoveCrossBoxAtDepth *(float Depth)*

Removes a box from the Cross Section Log at the specified depth.

Object: Log

Return Value: -

Parameter:

Depth The depth in current units at which the box should be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("XSection")
obLog.RemoveCrossBoxAtDepth 10.5
```

InsertNewSchmitBox *(float TopDepth, float BottomDepth, BSTR Text)*

Inserts a new box into a Polar & Rose Log.

Object: Log

Return Value: SchmitBox object

Parameter:

TopDepth	Top depth of the box in current depth units.
BottomDepth	Bottom depth of the box in current depth units.
Text	Description associated with the box in the Polar & Rose Log.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(20)
Set obBox = obLog.InsertNewFossilItem(0, 10, "Rose Diagram")
```

RemoveSchmitBox (long Index)

Removes a box from the Polar & Rose log at the specified index.

Object: Log

Return Value: -

Parameter:

Index The zero based index of the box to be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
obLog.RemoveSchmitBox 0
```

RemoveSchmitBoxAtDepth (float Depth)

Removes a box from the Polar & Rose log at the specified depth.

Object: Log

Return Value: -

Parameter:

Depth The the depth in current depth units at which the box should be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
obLog.RemoveSchmitBoxAtDepth 10.5
```

InsertNewFossilItem *(float TopDepth, float BottomDepth, BSTR Code, float Abundance, long Dominance, float Position)*

Inserts a new interval and symbol into the CoreDesc Log.

Object: Log

Return Value: FossilItem Object

Parameter:

TopDepth	The top of the interval in current depth units.
BottomDepth	The bottom of the interval in current depth units.
Code	A string correspoding to the code which identifies the desired symbol in the litho library.
Abundance	The abundance value associated with the symbol (e.g. between 0 and 9).
Dominance	The dominance value associated with the symbol (0 = undiff, 1 = minor, 2 = major).
Position	A value between 0 and 1 determining the horizontal position of the symbol within the log column.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(16)
Set obItem = obLog.InsertNewFossilItem(0, 1, "abc", 0, 2, 0.5")
```

RemoveFossilItem (long Index)

Removes an item at the specified index from the CoreDesc Log.

Object: Log

Return Value: -

Parameter:

Index The zero based index of the symbol to be removed from the log.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
obLog.RemoveFossilItem 0
```

RemoveFossilItemAtDepth *(float Depth)*

Removes an item at the specified depth from the CoreDesc Log.

Object: Log

Return Value: -

Parameter:

Depth	The depth of the symbol in current depth units at which it should be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
obLog.RemoveFossilItemAtDepth 10.5
```

InsertNewStackItem (float TopDepth, float BottomDepth, float TopWidth, float BottomWidth)

Inserts a new stacking pattern box and determines the symbol shape from top and bottom width (values between 0 and 1).

Object: Log

Return Value: StackItem Object

Parameter:

TopDepth	The depth at the top of the stacking pattern box in current depth units.
BottomDepth	The depth at the bottom of the stacking pattern box in current depth units.
TopWidth	A value between 0 and 1 expressig the width of the top of the symbol.
BottomWidth	A value between 0 and 1 expressing the width of the bottom of the symbol.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(19)
obLog.InsertNewStackItem 0, 10, 0.1, 0.8
```

RemoveStackItem (long Index)

Removes an item from the Stacking Pattern Log.

Object: Log

Return Value: -

Parameter:

Index The zero based index of the stacking pattern box to be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
obLog.RemoveStackItem 0
```

RemoveStackItemAtDepth *(float Depth)*

Removes an item from the Stacking Pattern Log at the specified depth.

Object: Log

Return Value: -

Parameter:

Depth	The depth in current units at which the stacking pattern box should be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
obLog.RemoveStackItemAtDepth 10.5
```

InterpretSQLStatement (BSTR Statement)

This method can be called from the ODBC object to execute an SQL statement using the WellCAD SQL Interpreter.

Object: ODBC

Return Value: bool

Parameter:

Statement A string containing the SQL statement such as a Select query or WellCAD specific SQL procedure.

Remarks:

This method can only be used if the ODBC add-on module is activated on your WellCAD license.

Example:

```
Set obBHDoc = obWCAD.NewBorehole()
Set obODBC = obBHDoc.ODBC
obODBC.InterpretSQLStatement("$dsn = OpenDatabase(Test, Admin)")
obODBC.InterpretSQLStatement("$Log = select Depth, Value from tblGamma")
obODBC.InterpretSQLStatement("LoadWellLog($Log)")
obODBC.InterpretSQLStatement("Close($Log)")
obODBC.InterpretSQLStatement("Close($dsn)")
```

RemoveStrataColumn (long Index)

Removes a column from a Strata Log.

Object: Log

Return Value: -

Parameter:

Index The zero based index of the column to be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("LithoStrata")
obLog.RemoveStrataColumn 3
```

RemoveTrace (long Index)

Removes a trace from an Image, FWS, Analysis or Percentage Log

Object: Log

Return Value: -

Parameter:

Index The zero based index of the data sample to be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Traveltime")
obLog.RemoveTrace 0
```

RemoveTraceAtDepth (float Depth)

Removes an entire data trace from an Image, FWS, Analysis or Percentage Log. Removing a trace from an Image, FWS or Analysis Log means setting all trace values to NULL.

Object: Log

Return Value: -

Parameter:

Depth The depth in current units at which the trace will be removed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Traveltime")
obLog.RemoveTraceAtDepth 10.5
```

InsertNewAttribute *(BSTR AttributeName)*

Inserts a new blank attribute class to the Structure Log.

Object: Log

Return Value: -

Parameter:

AttributeName The name of the new attribute.

Remarks:

It is not necessary to load the classification for picked structures from an LTH file. The user can enter information into an attribute class that is not linked to an LTH file.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(6)
obLog.AttachAttributeDictionary "Remarks"
```

AttachAttributeDictionary *(BSTR AttributeName, BSTR FileName)*

Attaches a new attribute library (*.LTH file) to a Structure Log

Object: Log

Return Value: -

Parameter:

AttributeName	A string specifying the name of the attribute class as it appears when ranking the picked features in the Structure Log.
FileName	Path and title of the LTH file to be loaded.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.InsertNewLog(6)
obLog.AttachAttributeDictionary "Fracture Form", "C:\Temp\Form.lth"
```

InsertNewOleBoxFromFile (BSTR FileName,VARIANT AllowPicture, float TopDepth, float BottomDepth)

Inserts a new box into an OLE Log displaying the data (e.g. JPEG image) contaied in the specified file.

Object: Log

Return Value: -

Parameter:

FileName	The path and title of the file containing the information to be displayed in the new box.
AllowPicture	Boolean specifying whether the loaded information should be displayed as picture or OLE object.
TopDepth	The top depth of the new OLE Log box in current depth units.
BottomDepth	The bottom depth of the new OLE Log box in current depth units.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Core Photographs")
Oblog.InsertNewOleBoxFromFile "C:\Temp\Photo1.jpg", TRUE, 100.5, 101.5
```

InsertNewMarker (float Depth, BSTR Name, BSTR Comment, BSTR Contact)

Inserts a new marker into a Marker Log

Object: Log

Return Value: MarkerBox Object

Parameter:

Depth	The value specifies the depth in current depth units at which the marer will be placed.
Name	The name and also idenfier of the marker to be inserted.
Comment	Contains additional information displayed along with the marker name.
Contact	The contact style that will be read from the contact library and displayed.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
obLog.InsertNewMarker 822.5, "Reservoir", "TVD 819.1 m", "Major"
```

RemoveMarker (long Index)

Remove the marker from a Marker Log at the specfied index.

Object: Log

Return Value: -

Parameter:

Index Zero based index of the marker within a Marker Log.

Remarks:

The total number of markers in a log can be retrieved with the "NbOfData" property.

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
obLog.RemoveMarker obLog.NbOfData - 1
```

Short ***NbOfItems***

Returns the number of items within the header object

Object: Header

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obHeader = obBorehole.Header
HeaderItems = obHeader.NbOfItems
```

BSTR ***ItemName*** *(short Index)*

Returns the title of the dynamic text field identified by the item index from the header.

Object: Header

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obHeader = obBorehole.Header
For i = 0 To NbHeaderItems-1
   Item = obHeader.ItemName(i)
Next
```

*BSTR **ItemText** (BSTR ItemName)*

Returns the text of the dynamic text field identified by the item name.

Object: Header

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obHeader = obBorehole.Header
For i = 0 To NbHeaderItems-1
   Item = obHeader.ItemName(i)
   Text = obHeader.ItemText(Item)
Next
```

***ItemText** (short Index)*

Returns the title of the dynamic text field identified by the item index from the header.

Object: Header

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obHeader = obBorehole.Header
For i = 0 To NbHeaderItems-1
   Item = obHeader.ItemName(i)
   obHeader.ItemText(Item, "Not Available")
Next
```

Short ***LeftMargin***

Set or get the left page margin of the printout in mm.

Object: Page

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
obPage.LeftMargin = 5
```

Short ***RightMargin***

Set or get the right page margin of the printout in mm.

Object: Page

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
obPage.RightMargin = 5
```

Short ***TopMargin***

Set or get the margin at the top of a page in mm

Object: Page

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
obPage.TopMargin = 10
```

Short ***BottomMargin***

Set or get the margin at the bottom of a page in mm

Object: Page

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
obPage.BottomMargin = 10
```

Short ***PaperMode***

Set or get the paper mode mode (0 = sheet, 1 = fanfold)

Object: Page

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
obPage.PaperMode = 1
```

Short ***Numbering***

Set or get the page numbering mode (0 = none, 1 = left, 2 = center, 3 = right, 4 = alternate)

Object: Page

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obPage = obBorehole.Page
obPage.Numbering = 2
```

Float ***Scale***

Get or set the depth scale (e.g. 100 = 1: 100)

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.Scale = 100
```

Float ***LeftPosition***

Get or set the left position of the master depth column border in percent of the document width (value between 0 and 1)

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.LeftPosition = 0
```

Float ***RightPosition***

Get or set the right position of the master depth column border in percent of the document width (value between 0 and 1)

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.RightPosition = 0.1
```

Short ***Decimals***

Set or get the number of decimals displayed in the depth string

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.Decimals = 1
```

Short ***Unit***

Set or get the depth unit (0 = meter, 1 = feet)

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.Unit = 1
```

Short ***HorizontalGrid***

Set or get the depth grid type (0 = none, 1 = Major, 2 = Minor)

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.HorizontalGrid = 1
```

SetPosition *(float Left, float Right)*

Get or set the left and right position of the master depth column border in percent of the document width (value between 0 and 1)

Object: Depth

Return Value: -

Parameter:

Left	A value between 0 and 1 indicating the left depth column border
Right	A value between 0 and 1 indicating the right depth column border

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.SetPosition 0, 0.1
```

Float *HorizontalGridSpacing*

Get or set the depth grid spacing for the master depth axis

Object: Depth

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obDepth = obBorehole.Depth
obDepth.HorizontalGridSpacing = 1
```

Float ***TopDepth***

Get or set the top depth of a lithology bed of the Litho Log in current depth units

Object: LithoBed

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obItem = obLog.LithoBed(0)
Top = obItem.TopDepth
```

Float ***BottomDepth***

Get or set the bottom depth of a lithology bed of the Litho Log in current depth units

Object: LithoBed

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obItem = obLog.LithoBed(0)
Bot = obItem.BottomDepth
```

BSTR ***LithoCode***

Get or set the code for a specific lithology bed

Object: LithoBed

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obItem = obLog.LithoBedAtDepth(10)
obItem.LithoCode = "Sst"
```

BSTR ***TopContact***

Get or set the top contact style for a specific lithology bed

Object: LithoBed

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obItem = obLog.LithoBedAtDepth(10)
obItem.BottomContact = "Sharp"
```

BSTR ***BottomContact***

Get or set the bottom contact style for a specific lithology bed

Object: LithoBed

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obItem = obLog.LithoBedAtDepth(10)
obItem.BottomContact = "Sharp"
```

Float ***Value***

Get or set the "hardness" value (between 0 and 1) for a lithology bed

Object: LithoBed

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obItem = obLog.LithoBedAtDepth(10)
obItem.Value = 0.75
```

Short ***NbOfPatterns***

Retrieve the number of patterns contained in the litho library

Object: LithoDictionary

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
NbPatterns = obDictionaryItem.NbOfPatterns
```

IDispatch ***LithoPattern*** *(VARIANT Pattern)*

Retrieve the litho pattern from the litho dictionary by passing either the code, description or index of the pattern

Object: LithoDictionary

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Set obPatternItem = obDictionaryItem.Lithopattern("Sst")
```

BSTR ***Name***

Get or setthe name of a litho library

Object: LithoDictionary

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Title = obDictionaryItem.Name
```

IsPattern *(BSTR Code)*

Retrive the information whether the specified code belongs to a pattern within the library

Object: LithoDictionary

Return Value: Bool

Parameter:

Code The code refferring to a lito pattern

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Existing = obDictionaryItem.IsPattern("Sst")
```

BSTR ***Description***

Get the description of the litgo pattern

Object: LithoPattern

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Set obPatternItem = obDictionaryItem.Lithopattern("Sst")
Text = obPatternItem.Description
```

bool ***Repeatable***

Retrieve the information whether the lithology pattern is a repeated one or a symbol

Object: LithoPattern

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Set obPatternItem = obDictionaryItem.Lithopattern("Sst")
Repeated = obPatternItem.Repeatable
```

long ***Height***

Get the y extension of the lithology pattern in 1/10 mm

Object: LithoPattern

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Set obPatternItem = obDictionaryItem.Lithopattern("Sst")
ExtY = obPatternItem.Height
```

long ***Width***

Get the x extension of the lithology pattern in 1/10 mm

Object: LithoPattern

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Set obPatternItem = obDictionaryItem.Lithopattern("Sst")
ExtX = obPatternItem.Width
```

BSTR ***Code***

Retrieve the code of a lithology pattern

Object: LithoPattern

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Lithology")
Set obDictionaryItem = obLog.AttachLithoDictionary("C:\Temp\Litho.lth")
Set obPatternItem = obDictionaryItem.Lithopattern("Sand Stone")
Code = obPatternItem.Code
```

Float ***TopDepth***

Get the top depth of a comment box object in current depth units

Object: CommentBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Comments")
Set obItem = CommentBox(0)
Top = obItem.TopDepth
```

Float ***BottomDepth***

Get the bottom depth of a comment box object in current depth units

Object: CommentBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Comments")
Set obItem = CommentBox(0)
Bot = obItem.BottomDepth
```

BSTR ***Text***

Get or set the text within a comment box object

Object: CommentBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Comments")
Set obItem = CommentBox(0)
obItem.Text("Hello World!")
```

*BSTR **AttributeValue**(BSTR AttributeName)*

Retrieve the value from the specified attribute class belonging to the structure object

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
Code = obItem.AttributeValue("Condition")
```

***AttributeValue**(BSTR AttributeName, BSTR AttributeName)*

Set the value for the specified attribute class belonging to the structure object

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
obItem.AttributeValue("Condition", "rough")
```

Float ***Azimuth***

Get or set the azimuth angle for a structure object in degree

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
obItem.Azimuth = 270
```

Float ***Tilt***

Get or set the dip angle (measured from the horizontal) for a structure object in degree

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
obItem.Tilt = 15
```

Short ***Category***

Get or set the code for an attribute named "Type". This property has been left in to be compatible with the Structure Log format in WellCAD v4.1 and older.

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
obItem.Category = 105
```

BSTR ***Description***

Get or set the text for an attribute named "Description". This property has been left in to be compatible with the Structure Log format in WellCAD v4.1 and older.

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
obItem.Description = "Test"
```

Float ***Depth***

Get or set the depth of a structure object in current depth units

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = Structure(0)
Depth = obItem.Depth
```

Float ***Aperture***

Get or set the aperture of a structure object in meter

Object: Structure

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Picks")
Set obItem = StructureAtDepth(10)
obItem.Aperture = 13
```

Float ***BottomDepth***

Get or set the bottom depth of a drill item in an Engineering Log in current depth units

Object: DrillItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = DrillItemAtDepth(10.5)
obItem.BottomDepth = 10.8
```

Float ***Diameter***

Get or set the diameter of a drill item in an Engineering Log

Object: DrillItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = DrillItemAtDepth(10.5)
obItem.Diameter = 100
```

Float ***TopDepth***

Get or set the top depth of an equipment item in the Engineering Log in current depth units

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(2)
obItem.TopDepth = 10
```

Float ***BottomDepth***

Get or set the bottom depth of an equipment item in the Engineering Log in current depth units

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(2)
obItem.BottomDepth = 20
```

Float ***AxisPosition***

Get or set the axis position for a solid item in an Engineering Log in units of the diameter

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(2)
obItem.AxisPosition = 50
```

Float ***ExternalDiameter***

Get or set the external diameter for a hollow or solid item in an Engineering Log in units of the diameter

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(2)
obItem.ExternalDiameter = 100
```

Float ***InternalDiameter***

Get or set the internal diameter for a hollow item in an Engineering Log in units of the diameter

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(2)
obItem.InternalDiameter = 95
```

Float ***InjectionPosition***

Get or set the injection position for a liquid item in an Engineering Log in units of the diameter

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(1)
obItem.InjectionPosition = 100
```

Float ***InjectionDepth***

Get or set the injection depth for a liquid item in an Engineering Log in current depth reference units

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(1)
obItem.InjectionDepth = 10
```

Short ***Type***

Retrieve the type of equipment item used in an Engineering Log (0 = undefined, 1 = Solid Item, 2 = Hollow Item, 3 = Liquid Item)

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(1)
Type = obItem.Type
```

BSTR ***Name***

Retrieve the name of an equiment item in an Engineering Log

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(1)
Name = obItem.Name
```

BSTR ***Description***

Retrieve the description of an equiment item in an Engineering Log

Object: EqpItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Well Layout")
Set obItem = EqpItem(1)
Remark = obItem.Description
```

Float ***TopDepth***

Retrieve the top depth of an interval item in current reference units

Object: IntervalItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
Set obItem = obLog.IntervalItemAtDepth(10.5)
Top = obItem.TopDepth
```

Float ***BottomDepth***

Retrieve the bottom depth of an interval item in current reference units

Object: IntervalItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
Set obItem = obLog.IntervalItemAtDepth(10.5)
Bot = obItem.BottomDepth
```

Float ***Value***

Get the data value of an interval item

Object: IntervalItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
Set obItem = obLog.IntervalItemAtDepth(10.5)
Data = obItem.Value
```

Value *(float Value)*

Set the data value of an interval item

Object: IntervalItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Test")
Set obItem = obLog.IntervalItemAtDepth(10.5)
obItem.Value(100.5)
```

float ***TopDepth***

Retrieve the top depth of a box in the Cross Section Log

Object: CrossBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Caliper")
Set obItem = obLog.CrossBoxAtDepth(10.5)
Top = obItem.TopDepth
```

Float ***BottomDepth***

Retrieve the bottom depth of a box in the Cross Section Log

Object: CrossBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Caliper")
Set obItem = obLog.CrossBoxAtDepth(10.5)
Bot = obItem.BottomDepth
```

Float ***TopDepth***

Retrieve the top depth of a polar & rose box in current depth reference units

Object: SchmitBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
Set obItem = obLog.SchmitBoxAtDepth(10.5)
Top = obItem.TopDepth
```

Float ***BottomDepth***

Retrieve the bottom depth of a polar & rose box in current depth reference units

Object: SchmitBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
Set obItem = obLog.SchmitBoxAtDepth(10.5)
Bot = obItem.BottomDepth
```

BSTR ***Text***

Get or set the description of a polar & rose box

Object: SchmitBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Rose")
Set obItem = obLog.SchmitBoxAtDepth(10.5)
obItem.Text = "Automation Demo"
```

Float ***TopDepth***

Get the top depth of a fossil item in the Coredesc Log.

Object: FossilItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItemAtDepth(10.5)
Top = obItem.TopDepth
```

Float ***BottomDepth***

Get the bottom depth of a fossil item in the Coredesc Log.

Object: FossilItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItemAtDepth(10.5)
Bot = obItem.BottomDepth
```

BSTR ***SymbolCode***

Get or set the symbol code of a fossil item in the Coredesc Log.

Object: FossilItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItemAtDepth(10.5)
obItem.SymbolCode = "ffl"
```

Short ***Dominance***

Get or set the dominance value of a fossil item (1 – 3) in the Coredesc Log.

Object: FossilItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItemAtDepth(10.5)
obItem.Dominance = 1
```

Float ***Abundance***

Get or set the abundance value of a fossil item (1 - 9) in the Coredesc Log.

Object: FossilItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Fossils")
Set obItem = obLog.FossilItemAtDepth(10.5)
obItem.Abundance = 9
```

Float ***TopDepth***

Get the top depth of a stacking pattern item in current reference units.

Object: StackItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
Set obStackItem = obLog.StackItemAtDepth(10.5)
Top = obStackItem.TopDepth
```

Float ***BottomDepth***

Get the bottom depth of a stacking pattern item in current reference units.

Object: StackItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
Set obStackItem = obLog.StackItemAtDepth(10.5)
Bot = obStackItem.BottomDepth
```

Float ***TopWidth***

Set or get the top width of a stacking pattern symbol (value between 0 and 1).

Object: StackItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
Set obStackItem = obLog.StackItemAtDepth(10.5)
obStackItem.TopWidth = 0.2
```

Float ***BottomWidth***

Set or get the bottom width of a stacking pattern symbol (value between 0 and 1).

Object: StackItem

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Coarsening")
Set obStackItem = obLog.StackItemAtDepth(10.5)
obStackItem.BottomWidth = 0.5
```

Float ***Depth***

Set or get the depth of a marker from the Marker Log in current depth units

Object: MarkerBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
Set obItem = obLog.Marker(0)
MarkTop = obItem.Depth
```

BSTR ***Name***

Set or get the depth of a marker from the Marker Log

Object: MarkerBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
Set obItem = obLog.Marker(0)
MarkName = obItem.Name
```

BSTR ***Comment***

Set or get the comment line of a marker from the Marker Log

Object: MarkerBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
Set obItem = obLog.MarkerByName("Reservoir")
MarkComment = obItem.Comment
```

BSTR ***Contact***

Set or get the contact style of a marker from the Marker Log

Object: MarkerBox

Example:

```
Set obWCAD = CreateObject("WellCAD.Application")
obWCAD.ShowWindow()
Set obBorehole = obWCAD.GetBorehole()
Set obLog = obBorehole.Log("Formation Tops")
Set obItem = obLog.MarkerByName("Reservoir")
MarkComment = obItem.Contact
```