

# PutBinaryValue Function

## Description

This function will update the contents of a binary variable.

## Syntax

```
retval = PutBinaryValue(variable, offset, varType [,varSize])
```

## Parameters

The function has the following parameters:

Parameter	Description
variable	A binary structure.
offset	A 1-based offset at which to start the insertion.
varType	Type of variable you want to insert. This is the same as it is for <a href="#">GetValue</a> , <a href="#">LockVariable</a> .
varSize	The size, in bytes, of the data to be inserted. For CHAR, ACHAR, UCHAR, WCHAR or BINARY varTypes.

## Remarks

A buffer must be allocated using the right size or there will be an overflow error.

## See also

[ANSI\\_Uncode\(\)](#), [Unicode\\_ANSI\(\)](#), [GetByteSize\(\)](#), [GetBinaryValue\(\)](#)

## Example

```
/* This example takes the following steps:  
1. Converts an ANSI string to UNICODE  
2. Determines the length of the 2 strings  
3. Creates a buffer for the length of both strings  
4. Places the strings into the buffer  
5. Converts the buffer string to ANSI  
*/  
  
unicodeString = ""  
  
uniCodeString = ANSI_Uncode( "An Example" )  
uniCodeLength = GetByteSize( uniCodeString )  
moreText = ANSI_Uncode( " of the PutBinaryValue routine" )  
moreLength = GetByteSize( moreText )  
buffer = str( \00\, uniLength + moreLength )  
PutBinaryValue( buffer, 1, CHAR, uniCodeString )  
PutBinaryValue( buffer, uniLength + 1, CHAR, moreText )  
* Convert the string back to ANSI  
newString = Unicode_ANSI( buffer )
```