

ifByteArray

Implemented By

- roByteArray

Supported Methods

- WriteFile(path as String) As Boolean
- WriteFile(path as String, start_index as Integer, length as Integer) As Boolean
- ReadFile(path as String) As Boolean
- ReadFile(path as String, start_pos as Integer, length as Integer) As Boolean
- AppendFile(path as String) As Boolean
- AppendFile(path as String, start_pos as Integer, length as Integer) As Boolean
- SetResize(min_size as Integer, auto_resize as Boolean) As Void
- ToHexString() As String
- FromHexString(hexstring as String) As Void
- ToBase64String() As String
- FromBase64String(s as String) As Void
- ToAsciiString() As String
- FromAsciiString(s as String)
- GetSignedByte(index as Integer) As Integer
- GetSignedLong(index as Integer) As Integer
- GetCRC32() as Integer
- GetCRC32(start as Integer, length as Integer) As Integer
- IsLittleEndianCPU() As Boolean

Description of Methods

WriteFile(path as String) As Boolean

Writes the bytes contained in the Byte Array to the specified file.

Returns true if successful.

WriteFile(path as String, start_index as Integer, length as Integer) As Boolean

Writes a subset of the bytes contained in the Byte Array to the specified file. "length" bytes are written, starting at the zero-based start_index.

Returns true if successful.

ReadFile(path as String) As Boolean

Reads the specified file into the Byte Array.

Any data currently in the Byte Array is discarded.

Returns true if successful.

ReadFile(path as String, start_pos as Integer, length as Integer) As Boolean

Reads a section of the file into the Byte Array.

"length" bytes are read, starting at the zero-based start_pos.

Note that in WriteFile, start_index is an index into the Byte Array, but in ReadFile, start_pos is an index into the file.

Any data currently in the Byte Array is discarded.

Returns true if successful.

AppendFile(path as String) As Boolean

Appends the contents of the Byte Array to the specified file.

Returns true if successful.

AppendFile(path as String, start_pos as Integer, length as Integer) As Boolean

Appends a subset of the bytes contained in the Byte Array to the specified file.

"length" bytes are written, starting at the zero-based start_index.

Returns true if successful.

This function is available in firmware 7.5 or later.

SetResize(min_size as Integer, auto_resize as Boolean) As Void

If the size of the Byte Array is less than min_size, expands the Byte Array to min_size.

Also sets the auto-resize attribute of the Byte Array to the specified value.

ToHexString() As String

Returns a hexadecimal string representing the contents of the Byte Array, two digits per byte.

FromHexString(hexstring as String) As Void

Sets the contents of the Byte Array to the specified value.

The string must be an even number of hexadecimal digits.

The string must contain valid hexadecimal digits, or the result is undefined.

Any data currently in the Byte Array is discarded.

ToBase64String() As String

Returns a base-64 string representing the contents of the Byte Array.

FromBase64String(s as String) As Void

Sets the contents of the Byte Array to the specified value.

The string must be a valid base-64 encoding.

Any data currently in the Byte Array is discarded.

ToAsciiString() As String

Returns the contents of the Byte Array as a string.
The contents must be valid UTF-8 (or ASCII subset), or the result is undefined.

FromAsciiString(s as String)

Sets the contents of the Byte Array to the specified string using UTF-8 encoding.
Any data currently in the Byte Array is discarded.

GetSignedByte(index as Integer) As Integer

Returns the signed byte at the specified zero-based index in the Byte Array.
Use `ifArrayGet.GetEntry()` or the `[]` array operator to read an unsigned byte in the Byte Array.

GetSignedLong(index as Integer) As Integer

Returns the signed long (four bytes) starting at the specified zero-based index in the Byte Array.

GetCRC32() as Integer

Calculates a CRC-32 of the contents of the Byte Array.

This function is available in firmware 7.5 or later.

Examples

```
ba = CreateObject("roByteArray")
ba.FromAsciiString("Hello world!")
n = ba.GetCrc32()
print n, "0x" ; StrI(n, 16)
REM 461707669 0x1b851995
```

GetCRC32(start as Integer, length as Integer) As Integer

Calculates a CRC-32 of a subset of the bytes in the Byte Array.

This function is available in firmware 7.5 or later.

IsLittleEndianCPU() As Boolean

Returns true if the CPU architecture is little-endian.