

roCreateFile

ON THIS PAGE

- [ifReadStream](#)
 - [SetReceiveEol\(eol_sequence As String\) As Void](#)
 - [ReadByte\(\) As Integer](#)
 - [ReadByteIfAvailable\(\) As Integer](#)
 - [ReadLine\(\) As String](#)
 - [ReadBlock\(size As Integer\) As String](#)
 - [AtEof\(\) As Boolean](#)
- [ifStreamSend](#)
 - [SetSendEol\(eol_sequence As String\) As Void](#)
 - [SendByte\(byte As Integer\) As Void](#)
 - [SendLine\(string As String\) As Void](#)
 - [SendBlock\(a As Dynamic\) As Void](#)
 - [Flush\(\) As Void](#)
 - [AsyncFlush\(\) As Void](#)
- [ifStreamSeek](#)
 - [SeekAbsolute\(offset As Integer\) As Void](#)
 - [SeekRelative\(offset As Integer\) As Void](#)
 - [SeekToEnd\(\) As Void](#)
 - [CurrentPosition\(\) As Integer](#)

▼ OS8

- [OS8](#)
- [Version 7.1](#)
- [Version 7.0](#)
- [Version 6.2](#)
- [Version 6.1](#)
- [Previous Versions](#)

This object can be used to write a new file or overwrite an existing file.

Object Creation: Creating an *roCreateFile* object opens an existing file or creates a new file. If the file exists, it is truncated to a size of zero.

```
CreateObject("roCreateFile", filename As String)
```

ifReadStream

SetReceiveEol(eol_sequence As String) As Void

Sets the EOL sequence when reading from the stream.

ReadByte() As Integer

Reads a single byte from the stream, blocking if necessary. If the EOF is reached or there is an error condition, then a value less than 0 is returned.

ReadByteIfAvailable() As Integer

Reads a single byte from the stream if one is available. If no bytes are available, it returns immediately. A return value less than 0 indicates either that the EOF has been reached or no byte is available.

ReadLine() As String

Reads until it finds a complete end of the line sequence. If it fails to find the sequence within 4096 bytes, then it returns the 4096 bytes that are found. No data is discarded in this case.

ReadBlock(size As Integer) As String

Reads the specified number of bytes. The number is limited to 65536 bytes. In the event of an EOF or an error, fewer bytes than requested will be returned. Any null bytes in the file will mask any further bytes.

AtEof() As Boolean

Returns True if an attempt has been made to read beyond the end of the file. If the current position is at the end of the file, but no attempt has been made to read beyond it, this method will return False.

ifStreamSend

SetSendEol(eol_sequence As String) As Void

Sets the EOL sequence when writing to the stream. The default value is CR+LF. If you need to set this value to a non-printing character, use the `chr()` [global function](#).

SendByte(byte As Integer) As Void

Writes the specified byte to the stream.

SendLine(string As String) As Void

Writes the specified characters to the stream followed by the current EOL sequence.

SendBlock(a As Dynamic) As Void

Writes the specified characters to the stream. This method can support either a string or an *roByteArray*. If the block is a string, any null bytes will terminate the block.

Flush() As Void

Writes the stream to the file, blocking execution until all data is written out.

AsyncFlush() As Void

Writes the stream to the file within a few seconds. The system will not block execution while writing to the file.

Note

The system performs an asynchronous flush automatically when the object instance is destroyed (for instance, when the the variable containing it is reassigned). The stream is first written to the kernel, which then writes it to the file system when resources are available.

ifStreamSeek

SeekAbsolute(offset As Integer) As Void

Seeks the specified offset. If the offset is beyond the end of the file, then the file will be extended upon the next write and any previously unoccupied space will be filled with null bytes.

SeekRelative(offset As Integer) As Void

Seeks to the specified offset relative to the current position. If the ultimate offset is beyond the end of the file, then the file will be extended as described in `SeekAbsolute()`.

SeekToEnd() As Void

Seeks to the end of the file.

CurrentPosition() As Integer

Retrieves the current position within the file.