# A blue background with white text and symbols Description automatically generated

# DeviceSQL Modbus Master Documentation

### 1. ****BooleanRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlBoolean Value
* **Functions:**
  + public static ModbusMaster.BooleanRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Boolean Register

DECLARE @BooleanRegister dbo.ModbusMaster.BooleanRegister;

-- Parse a Boolean Register from a string

SET @BooleanRegister = dbo.ModbusMaster.BooleanRegister::Parse('1: true');

-- Access Binary Data

SELECT @BooleanRegister.Data;

-- Access Boolean Value

SELECT @BooleanRegister.Value;

### 2. ****BooleanRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.BooleanRegisterArray AddBooleanRegister(ModbusMaster.BooleanRegister booleanRegister)
  + public ModbusMaster.BooleanRegister GetBooleanRegister(SqlInt32 index)
  + public static ModbusMaster.BooleanRegisterArray Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Boolean Register Array

DECLARE @BooleanRegisterArray dbo.ModbusMaster.BooleanRegisterArray;

DECLARE @BooleanRegister dbo.ModbusMaster.BooleanRegister;

-- Parse a Boolean Register Array from a string

SET @BooleanRegisterArray = dbo.ModbusMaster.BooleanRegisterArray::Parse('1:true, 2:false');

-- Add a Boolean Register to the array

SET @BooleanRegister = dbo.ModbusMaster.BooleanRegister::Parse('3:true');

SET @BooleanRegisterArray = @BooleanRegisterArray.AddBooleanRegister(@BooleanRegister);

-- Get a Boolean Register from the array

SELECT @BooleanRegisterArray.GetBooleanRegister(0).Value;

### 3. ****CoilRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlBoolean Value
* **Functions:**
  + public static ModbusMaster.CoilRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Coil Register

DECLARE @CoilRegister dbo.ModbusMaster.CoilRegister;

-- Parse a Coil Register from a string

SET @CoilRegister = dbo.ModbusMaster.CoilRegister::Parse('1: true');

-- Access Binary Data

SELECT @CoilRegister.Data;

-- Access Boolean Value

SELECT @CoilRegister.Value;

### 4. ****CoilRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.CoilRegisterArray AddCoilRegister(ModbusMaster.CoilRegister coilRegister)
  + public ModbusMaster.CoilRegister GetCoilRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use a Coil Register Array

DECLARE @CoilRegisterArray dbo.ModbusMaster.CoilRegisterArray;

DECLARE @CoilRegister dbo.ModbusMaster.CoilRegister;

-- Parse a Coil Register Array from a string

SET @CoilRegisterArray = dbo.ModbusMaster.CoilRegisterArray::Parse('1:true, 2:false');

-- Add a Coil Register to the array

SET @CoilRegister = dbo.ModbusMaster.CoilRegister::Parse('3:true');

SET @CoilRegisterArray = @CoilRegisterArray.AddCoilRegister(@CoilRegister);

-- Get a Coil Register from the array

SELECT @CoilRegisterArray.GetCoilRegister(0).Value;

### 5. ****DiscreteInputRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlBoolean Value
* **Functions:**
  + public static ModbusMaster.DiscreteInputRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Discrete Input Register

DECLARE @DiscreteInputRegister dbo.ModbusMaster.DiscreteInputRegister;

-- Parse a Discrete Input Register from a string

SET @DiscreteInputRegister = dbo.ModbusMaster.DiscreteInputRegister::Parse('1: true');

-- Access Binary Data

SELECT @DiscreteInputRegister.Data;

-- Access Boolean Value

SELECT @DiscreteInputRegister.Value;

### 6. ****DiscreteInputRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.DiscreteInputRegisterArray AddDiscreteInputRegister(ModbusMaster.DiscreteInputRegister discreteInputRegister)
  + public ModbusMaster.DiscreteInputRegister GetDiscreteInputRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use a Discrete Input Register Array

DECLARE @DiscreteInputRegisterArray dbo.ModbusMaster.DiscreteInputRegisterArray;

DECLARE @DiscreteInputRegister dbo.ModbusMaster.DiscreteInputRegister;

-- Parse a Discrete Input Register Array from a string

SET @DiscreteInputRegisterArray = dbo.ModbusMaster.DiscreteInputRegisterArray::Parse('1:true, 2:false');

-- Add a Discrete Input Register to the array

SET @DiscreteInputRegister = dbo.ModbusMaster.DiscreteInputRegister::Parse('3:true');

SET @DiscreteInputRegisterArray = @DiscreteInputRegisterArray.AddDiscreteInputRegister(@DiscreteInputRegister);

-- Get a Discrete Input Register from the array

SELECT @DiscreteInputRegisterArray.GetDiscreteInputRegister(0).Value;

### 7. ****FloatRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlSingle Value
* **Functions:**
  + public static ModbusMaster.FloatRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Float Register

DECLARE @FloatRegister dbo.ModbusMaster.FloatRegister;

-- Parse a Float Register from a string

SET @FloatRegister = dbo.ModbusMaster.FloatRegister::Parse('1: 12.34');

-- Access Binary Data

SELECT @FloatRegister.Data;

-- Access Float Value

SELECT @FloatRegister.Value;

### 8. ****FloatRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.FloatRegisterArray AddFloatRegister(ModbusMaster.FloatRegister floatRegister)
  + public ModbusMaster.FloatRegister GetFloatRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use a Float Register Array

DECLARE @FloatRegisterArray dbo.ModbusMaster.FloatRegisterArray;

DECLARE @FloatRegister dbo.ModbusMaster.FloatRegister;

-- Parse a Float Register Array from a string

SET @FloatRegisterArray = dbo.ModbusMaster.FloatRegisterArray::Parse('1:12.34, 2:56.78');

-- Add a Float Register to the array

SET @FloatRegister = dbo.ModbusMaster.FloatRegister::Parse('3:90.12');

SET @FloatRegisterArray = @FloatRegisterArray.AddFloatRegister(@FloatRegister);

-- Get a Float Register from the array

SELECT @FloatRegisterArray.GetFloatRegister(0).Value;

### 9. ****LongRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlInt64 Value
* **Functions:**
  + public static ModbusMaster.LongRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Long Register

DECLARE @LongRegister dbo.ModbusMaster.LongRegister;

-- Parse a Long Register from a string

SET @LongRegister = dbo.ModbusMaster.LongRegister::Parse('1: 123456');

-- Access Binary Data

SELECT @LongRegister.Data;

-- Access Long Value

SELECT @LongRegister.Value;

### 10. ****LongRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.LongRegisterArray AddLongRegister(ModbusMaster.LongRegister longRegister)
  + public ModbusMaster.LongRegister GetLongRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use a Long Register Array

DECLARE @LongRegisterArray dbo.ModbusMaster.LongRegisterArray;

DECLARE @LongRegister dbo.ModbusMaster.LongRegister;

-- Parse a Long Register Array from a string

SET @LongRegisterArray = dbo.ModbusMaster.LongRegisterArray::Parse('1:123456, 2:654321');

-- Add a Long Register to the array

SET @LongRegister = dbo.ModbusMaster.LongRegister::Parse('3:789012');

SET @LongRegisterArray = @LongRegisterArray.AddLongRegister(@LongRegister);

-- Get a Long Register from the array

SELECT @LongRegisterArray.GetLongRegister(0).Value;

### 11. ****ShortRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlInt16 Value
* **Functions:**
  + public static ModbusMaster.ShortRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Short Register

DECLARE @ShortRegister dbo.ModbusMaster.ShortRegister;

-- Parse a Short Register from a string

SET @ShortRegister = dbo.ModbusMaster.ShortRegister::Parse('1: 123');

-- Access Binary Data

SELECT @ShortRegister.Data;

-- Access Short Value

SELECT @ShortRegister.Value;

### 12. ****ShortRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.ShortRegisterArray AddShortRegister(ModbusMaster.ShortRegister shortRegister)
  + public ModbusMaster.ShortRegister GetShortRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use a Short Register Array

DECLARE @ShortRegisterArray dbo.ModbusMaster.ShortRegisterArray;

DECLARE @ShortRegister dbo.ModbusMaster.ShortRegister;

-- Parse a Short Register Array from a string

SET @ShortRegisterArray = dbo.ModbusMaster.ShortRegisterArray::Parse('1:123, 2:456');

-- Add a Short Register to the array

SET @ShortRegister = dbo.ModbusMaster.ShortRegister::Parse('3:789');

SET @ShortRegisterArray = @ShortRegisterArray.AddShortRegister(@ShortRegister);

-- Get a Short Register from the array

SELECT @ShortRegisterArray.GetShortRegister(0).Value;

### 13. ****StringRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlString Value
* **Functions:**
  + public static ModbusMaster.StringRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a String Register

DECLARE @StringRegister dbo.ModbusMaster.StringRegister;

-- Parse a String Register from a string

SET @StringRegister = dbo.ModbusMaster.StringRegister::Parse('1: "Hello World"');

-- Access Binary Data

SELECT @StringRegister.Data;

-- Access String Value

SELECT @StringRegister.Value;

### 14. ****StringRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.StringRegisterArray AddStringRegister(ModbusMaster.StringRegister stringRegister)
  + public ModbusMaster.StringRegister GetStringRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use a String Register Array

DECLARE @StringRegisterArray dbo.ModbusMaster.StringRegisterArray;

DECLARE @StringRegister dbo.ModbusMaster.StringRegister;

-- Parse a String Register Array from a string

SET @StringRegisterArray = dbo.ModbusMaster.StringRegisterArray::Parse('1:"Hello", 2:"World"');

-- Add a String Register to the array

SET @StringRegister = dbo.ModbusMaster.StringRegister::Parse('3:"SQLCLR"');

SET @StringRegisterArray = @StringRegisterArray.AddStringRegister(@StringRegister);

-- Get a String Register from the array

SELECT @StringRegisterArray.GetStringRegister(0).Value;

### 15. ****ModbusAddress****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlInt32 Value
* **Functions:**
  + public static ModbusMaster.ModbusAddress Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Modbus Address

DECLARE @ModbusAddress dbo.ModbusMaster.ModbusAddress;

-- Parse a Modbus Address from a string

SET @ModbusAddress = dbo.ModbusMaster.ModbusAddress::Parse('1:12345');

-- Access Binary Data

SELECT @ModbusAddress.Data;

-- Access Address Value

SELECT @ModbusAddress.Value;

### Input and Holding Registers + Helper Methods

### 16. ****InputRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlInt32 Value
* **Functions:**
  + public static ModbusMaster.InputRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use an Input Register

DECLARE @InputRegister dbo.ModbusMaster.InputRegister;

-- Parse an Input Register from a string

SET @InputRegister = dbo.ModbusMaster.InputRegister::Parse('1: 3456');

-- Access Binary Data

SELECT @InputRegister.Data;

-- Access the Input Register Value

SELECT @InputRegister.Value;

### 17. ****InputRegisterArray****

#### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.InputRegisterArray AddInputRegister(ModbusMaster.InputRegister inputRegister)
  + public ModbusMaster.InputRegister GetInputRegister(SqlInt32 index)

#### SQL Example:

sql

-- Declare and use an Input Register Array

DECLARE @InputRegisterArray dbo.ModbusMaster.InputRegisterArray;

DECLARE @InputRegister dbo.ModbusMaster.InputRegister;

-- Parse an Input Register Array from a string

SET @InputRegisterArray = dbo.ModbusMaster.InputRegisterArray::Parse('1:3456, 2:7890');

-- Add an Input Register to the array

SET @InputRegister = dbo.ModbusMaster.InputRegister::Parse('3:1234');

SET @InputRegisterArray = @InputRegisterArray.AddInputRegister(@InputRegister);

-- Get an Input Register from the array

SELECT @InputRegisterArray.GetInputRegister(0).Value;

### 18. ****HoldingRegister****

#### SQLCLR Members:

* **Properties:**
  + SqlBinary Data
  + SqlInt32 Value
* **Functions:**
  + public static ModbusMaster.HoldingRegister Parse(SqlString stringToParse)

#### SQL Example:

sql

-- Declare and use a Holding Register

DECLARE @HoldingRegister dbo.ModbusMaster.HoldingRegister;

-- Parse a Holding Register from a string

SET @HoldingRegister = dbo.ModbusMaster.HoldingRegister::Parse('1: 1234');

-- Access Binary Data

SELECT @HoldingRegister.Data;

-- Access the Holding Register Value

SELECT @HoldingRegister.Value;

### 19. ****HoldingRegisterArray****

### SQLCLR Members:

* **Functions:**
  + public ModbusMaster.HoldingRegisterArray AddHoldingRegister(ModbusMaster.HoldingRegister holdingRegister)
  + public ModbusMaster.HoldingRegister GetHoldingRegister(SqlInt32 index)

### SQL Example:

sql

-- Declare and use a Holding Register Array

DECLARE @HoldingRegisterArray dbo.ModbusMaster.HoldingRegisterArray;

DECLARE @HoldingRegister dbo.ModbusMaster.HoldingRegister;

-- Parse a Holding Register Array from a string

SET @HoldingRegisterArray = dbo.ModbusMaster.HoldingRegisterArray::Parse('1:1234, 2:5678');

-- Add a Holding Register to the array

SET @HoldingRegister = dbo.ModbusMaster.HoldingRegister::Parse('3:9012');

SET @HoldingRegisterArray = @HoldingRegisterArray.AddHoldingRegister(@HoldingRegister);

-- Get a Holding Register from the array

SELECT @HoldingRegisterArray.GetHoldingRegister(0).Value;

### Helper Methods for ****InputRegisterArray**** and ****HoldingRegisterArray****

#### 20. ****GetByte****

##### SQL Example:

sql

-- Get a byte from a specific Input Register at a given index and offset

DECLARE @RegisterIndex TINYINT = 1;

DECLARE @Offset INT = 0;

-- Call the GetByte method to retrieve the byte

SELECT dbo.ModbusMaster.InputRegisterArray::GetByte(@RegisterIndex, @Offset) AS ByteValue;

#### 21. ****GetBytes****

##### SQL Example:

sql

-- Get multiple bytes from a specified number of registers

DECLARE @RegisterIndex TINYINT = 1;

DECLARE @RegisterCount INT = 4;

-- Call the GetBytes method to retrieve bytes across multiple registers

SELECT dbo.ModbusMaster.InputRegisterArray::GetBytes(@RegisterIndex, @RegisterCount) AS ByteArray;

#### 22. ****GetShort****

##### SQL Example:

sql

-- Get a short value from a register with optional byte swapping

DECLARE @RegisterIndex TINYINT = 1;

DECLARE @ByteSwap BIT = 1;

-- Call the GetShort method to retrieve the short value

SELECT dbo.ModbusMaster.InputRegisterArray::GetShort(@RegisterIndex, @ByteSwap) AS ShortValue;

#### 23. ****GetFloat****

##### SQL Example:

sql

-- Get a float value from a register with optional byte and word swapping

DECLARE @RegisterIndex TINYINT = 1;

DECLARE @ByteSwap BIT = 1;

DECLARE @WordSwap BIT = 1;

-- Call the GetFloat method to retrieve the float value

SELECT dbo.ModbusMaster.InputRegisterArray::GetFloat(@RegisterIndex, @ByteSwap, @WordSwap) AS FloatValue;

#### 24. ****GetLong****

##### SQL Example:

sql

-- Get a long value from a register with optional byte and word swapping

DECLARE @RegisterIndex TINYINT = 1;

DECLARE @ByteSwap BIT = 1;

DECLARE @WordSwap BIT = 1;

-- Call the GetLong method to retrieve the long value

SELECT dbo.ModbusMaster.InputRegisterArray::GetLong(@RegisterIndex, @ByteSwap, @WordSwap) AS LongValue;

#### 25. ****GetString****

##### SQL Example:

sql

-- Get a string from registers by specifying the index and length

DECLARE @RegisterIndex TINYINT = 1;

DECLARE @Length TINYINT = 10;

-- Call the GetString method to retrieve the string

SELECT dbo.ModbusMaster.InputRegisterArray::GetString(@RegisterIndex, @Length) AS StringValue;

## Main Functions

### 1. ****DisableChannelTracing****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ChannelManager.DisableChannelLogging(SqlString channelName)

#### SQL Example:

sql

-- Disable tracing for a specific channel

DECLARE @ChannelName NVARCHAR(512) = 'Channel.1';

-- Call the DisableChannelLogging method

SELECT dbo.ChannelManager.DisableChannelLogging(@ChannelName);

### 2. ****EnableChannelTracing****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ChannelManager.EnableChannelLogging(SqlString channelName)

#### SQL Example:

sql

-- Enable tracing for a specific channel

DECLARE @ChannelName NVARCHAR(512) = 'Channel.1';

-- Call the EnableChannelLogging method

SELECT dbo.ChannelManager.EnableChannelLogging(@ChannelName);

### 3. ****GetChannels****

#### SQLCLR Members:

* **Properties:**
  + SqlString ChannelName
  + SqlString ChannelType
  + SqlInt32 ReadTimeout
  + SqlInt32 WriteTimeout
  + SqlString ConnectionString
  + SqlBoolean TracingEnabled
* **Method:**
  + public static IEnumerable ChannelManager.GetChannels()

#### SQL Example:

sql

-- Get details about all channels

SELECT \* FROM dbo.ChannelManager.GetChannels();

### 4. ****RegisterSerialPortChannel****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ChannelManager.RegisterSerialPortChannel(SqlString channelName, SqlString portName, SqlInt32 baudRate, SqlByte dataBits, SqlByte parity, SqlByte stopBits, SqlInt32 readTimeout, SqlInt32 writeTimeout)

#### SQL Example:

sql

-- Register a new serial port channel

DECLARE @ChannelName NVARCHAR(512) = 'SerialChannel.1';

DECLARE @PortName NVARCHAR(512) = 'COM1';

DECLARE @BaudRate INT = 9600;

DECLARE @DataBits TINYINT = 8;

DECLARE @Parity TINYINT = 0; -- No parity

DECLARE @StopBits TINYINT = 1;

DECLARE @ReadTimeout INT = 1000;

DECLARE @WriteTimeout INT = 1000;

-- Call the RegisterSerialPortChannel method

SELECT dbo.ChannelManager.RegisterSerialPortChannel(@ChannelName, @PortName, @BaudRate, @DataBits, @Parity, @StopBits, @ReadTimeout, @WriteTimeout);

### 5. ****RegisterTcpChannel****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ChannelManager.RegisterTcpChannel(SqlString channelName, SqlString hostName, SqlInt32 hostPort, SqlInt32 connectAttempts, SqlInt32 connectionRetryDelay, SqlInt32 readTimeout, SqlInt32 writeTimeout)

#### SQL Example:

sql

-- Register a new TCP channel

DECLARE @ChannelName NVARCHAR(512) = 'TcpChannel.1';

DECLARE @HostName NVARCHAR(512) = '192.168.1.1';

DECLARE @HostPort INT = 502;

DECLARE @ConnectAttempts INT = 3;

DECLARE @ConnectionRetryDelay INT = 5000; -- Retry after 5 seconds

DECLARE @ReadTimeout INT = 1000;

DECLARE @WriteTimeout INT = 1000;

-- Call the RegisterTcpChannel method

SELECT dbo.ChannelManager.RegisterTcpChannel(@ChannelName, @HostName, @HostPort, @ConnectAttempts, @ConnectionRetryDelay, @ReadTimeout, @WriteTimeout);

### 6. ****TraceChannels****

#### SQLCLR Members:

* **Properties:**
  + SqlDateTime MessageDateTimeStamp
  + SqlString Name
  + SqlDateTime StartTime
  + SqlString Operation
  + SqlDouble Duration
  + SqlInt32 Sequence
  + SqlInt32 Count
  + SqlString Data
  + SqlString ChannelType
* **Method:**
  + public static IEnumerable ChannelManager.TraceChannels()

#### SQL Example:

sql

-- Trace all channel operations

SELECT \* FROM dbo.ChannelManager.TraceChannels();

### 7. ****UnregisterChannel****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ChannelManager.UnregisterChannel(SqlString channelName)

#### SQL Example:

sql

-- Unregister a specific channel

DECLARE @ChannelName NVARCHAR(512) = 'Channel.1';

-- Call the UnregisterChannel method

SELECT dbo.ChannelManager.UnregisterChannel(@ChannelName);

This documentation provides a complete list of SQLCLR methods and properties from each of the **ChannelManager** files, with SQL examples showing how to use them in a SQL Server environment. Let me know if you'd like any additional details or adjustments! ​​

4o

##### You said:

DeviceManager.zip

Zip Archive

###### ChatGPT said:

ChatGPT

The **DeviceManager** folder contains the following C# files:

1. **GetDevices**
2. **RegisterModbusMaster**
3. **UnregisterDevice**

I'll now extract the SQLCLR methods and properties from these files and document them with corresponding SQL examples. ​​

Based on the extracted content from the **DeviceManager** files, here is the documentation for all **SQLCLR Methods** and **Properties**, along with corresponding SQL examples:

### 1. ****GetDevices****

#### SQLCLR Members:

* **Properties:**
  + SqlString ChannelName
  + SqlString DeviceName
  + SqlString DeviceType
  + SqlString Address
  + SqlInt32 NumberOfRetries
  + SqlInt32 WaitToRetry
  + SqlInt32 RequestWriteDelay
  + SqlInt32 ResponseReadDelay
* **Method:**
  + public static IEnumerable DeviceManager.GetDevices()

#### SQL Example:

sql

-- Retrieve all devices registered with the system

SELECT \* FROM dbo.DeviceManager.GetDevices();

### 2. ****RegisterModbusMaster****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean DeviceManager.RegisterModbusMaster(SqlString channelName, SqlString deviceName, SqlBoolean useMbapHeaders, SqlBoolean useExtendedAddressing, SqlInt32 unitId, SqlInt32 numberOfRetries, SqlInt32 waitToRetry, SqlInt32 requestWriteDelay, SqlInt32 responseReadDelay)

#### SQL Example:

sql

-- Register a new Modbus Master device

DECLARE @ChannelName NVARCHAR(512) = 'ModbusChannel.1';

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @UseMbapHeaders BIT = 1;

DECLARE @UseExtendedAddressing BIT = 0;

DECLARE @UnitId INT = 1;

DECLARE @NumberOfRetries INT = 3;

DECLARE @WaitToRetry INT = 500;

DECLARE @RequestWriteDelay INT = 100;

DECLARE @ResponseReadDelay INT = 200;

-- Call the RegisterModbusMaster method

SELECT dbo.DeviceManager.RegisterModbusMaster(@ChannelName, @DeviceName, @UseMbapHeaders, @UseExtendedAddressing, @UnitId, @NumberOfRetries, @WaitToRetry, @RequestWriteDelay, @ResponseReadDelay);

### 3. ****UnregisterDevice****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean DeviceManager.UnregisterDevice(SqlString deviceName)

#### SQL Example:

sql

-- Unregister a specific device from the system

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

-- Call the UnregisterDevice method

SELECT dbo.DeviceManager.UnregisterDevice(@DeviceName);

This documentation provides the complete list of SQLCLR methods and properties from the **DeviceManager** files, along with SQL examples to demonstrate how to use them in SQL Server. Let me know if you'd like any further information or additional details! ​​

4o

##### You said:

ModbusMaster.zip

Zip Archive

###### ChatGPT said:

ChatGPT

The **ModbusMaster** folder contains the following C# files:

1. **ReadCoils**
2. **ReadDiscreteInputs**
3. **ReadEventArchiveRecords**
4. **ReadFloats**
5. **ReadHistoryArchiveRecord**
6. **ReadHoldings**
7. **ReadInputs**
8. **ReadLongs**
9. **ReadShorts**
10. **ReadString**
11. **WriteBoolean**
12. **WriteFloats**

I will now extract the SQLCLR methods and properties from each of these files and document them with SQL examples. ​​

Here is the detailed documentation for **SQLCLR Methods** and **Properties** from all the files in the **ModbusMaster** folder, along with corresponding SQL examples:

### 1. ****ReadCoils****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.CoilRegisterArray ModbusMaster.ReadCoils(SqlString deviceName, ModbusMaster.ModbusMaster.CoilRegisterArray coilRegisterArray)

#### SQL Example:

sql

-- Read coil registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @CoilRegisterArray dbo.ModbusMaster.CoilRegisterArray;

-- Call the ReadCoils method

SELECT dbo.ModbusMaster.ReadCoils(@DeviceName, @CoilRegisterArray);

### 2. ****ReadDiscreteInputs****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.DiscreteInputRegisterArray ModbusMaster.ReadDiscreteInputs(SqlString deviceName, ModbusMaster.ModbusMaster.DiscreteInputRegisterArray discreteInputRegisterArray)

#### SQL Example:

sql

-- Read discrete input registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @DiscreteInputRegisterArray dbo.ModbusMaster.DiscreteInputRegisterArray;

-- Call the ReadDiscreteInputs method

SELECT dbo.ModbusMaster.ReadDiscreteInputs(@DeviceName, @DiscreteInputRegisterArray);

### 3. ****ReadEventArchiveRecords****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.EventArchiveRecordArray ModbusMaster.ReadEventArchiveRecords(SqlString deviceName, ModbusMaster.ModbusMaster.ModbusAddress eventArchiveAddress, SqlInt32 index)

#### SQL Example:

sql

-- Read event archive records from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @EventArchiveAddress dbo.ModbusMaster.ModbusAddress;

DECLARE @Index INT = 0;

-- Call the ReadEventArchiveRecords method

SELECT dbo.ModbusMaster.ReadEventArchiveRecords(@DeviceName, @EventArchiveAddress, @Index);

### 4. ****ReadFloats****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.FloatRegisterArray ModbusMaster.ReadFloats(SqlString deviceName, ModbusMaster.ModbusMaster.FloatRegisterArray floatRegisterArray)

#### SQL Example:

sql

-- Read float registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @FloatRegisterArray dbo.ModbusMaster.FloatRegisterArray;

-- Call the ReadFloats method

SELECT dbo.ModbusMaster.ReadFloats(@DeviceName, @FloatRegisterArray);

### 5. ****ReadHistoryArchiveRecord****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.HistoryArchiveRecord ModbusMaster.ReadHistoryArchiveRecord(SqlString deviceName, ModbusMaster.ModbusMaster.ModbusAddress historyArchiveAddress, SqlInt32 index, SqlByte recordSize)

#### SQL Example:

sql

-- Read a history archive record from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @HistoryArchiveAddress dbo.ModbusMaster.ModbusAddress;

DECLARE @Index INT = 0;

DECLARE @RecordSize TINYINT = 10;

-- Call the ReadHistoryArchiveRecord method

SELECT dbo.ModbusMaster.ReadHistoryArchiveRecord(@DeviceName, @HistoryArchiveAddress, @Index, @RecordSize);

### 6. ****ReadHoldings****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.HoldingRegisterArray ModbusMaster.ReadHoldings(SqlString deviceName, ModbusMaster.ModbusMaster.HoldingRegisterArray holdingRegisterArray)

#### SQL Example:

sql

-- Read holding registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @HoldingRegisterArray dbo.ModbusMaster.HoldingRegisterArray;

-- Call the ReadHoldings method

SELECT dbo.ModbusMaster.ReadHoldings(@DeviceName, @HoldingRegisterArray);

### 7. ****ReadInputs****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.InputRegisterArray ModbusMaster.ReadInputs(SqlString deviceName, ModbusMaster.ModbusMaster.InputRegisterArray inputRegisterArray)

#### SQL Example:

sql

-- Read input registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @InputRegisterArray dbo.ModbusMaster.InputRegisterArray;

-- Call the ReadInputs method

SELECT dbo.ModbusMaster.ReadInputs(@DeviceName, @InputRegisterArray);

### 8. ****ReadLongs****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.LongRegisterArray ModbusMaster.ReadLongs(SqlString deviceName, ModbusMaster.ModbusMaster.LongRegisterArray longRegisterArray)

#### SQL Example:

sql

-- Read long registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @LongRegisterArray dbo.ModbusMaster.LongRegisterArray;

-- Call the ReadLongs method

SELECT dbo.ModbusMaster.ReadLongs(@DeviceName, @LongRegisterArray);

### 9. ****ReadShorts****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.ShortRegisterArray ModbusMaster.ReadShorts(SqlString deviceName, ModbusMaster.ModbusMaster.ShortRegisterArray shortRegisterArray)

#### SQL Example:

sql

-- Read short registers from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @ShortRegisterArray dbo.ModbusMaster.ShortRegisterArray;

-- Call the ReadShorts method

SELECT dbo.ModbusMaster.ReadShorts(@DeviceName, @ShortRegisterArray);

### 10. ****ReadString****

#### SQLCLR Members:

* **Method:**
  + public static ModbusMaster.ModbusMaster.StringRegister ModbusMaster.ReadString(SqlString deviceName, ModbusMaster.ModbusMaster.StringRegister stringRegister)

#### SQL Example:

sql

-- Read a string register from a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @StringRegister dbo.ModbusMaster.StringRegister;

-- Call the ReadString method

SELECT dbo.ModbusMaster.ReadString(@DeviceName, @StringRegister);

### 11. ****WriteBoolean****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ModbusMaster.WriteBoolean(SqlString deviceName, ModbusMaster.ModbusMaster.BooleanRegister booleanRegister)

#### SQL Example:

sql

-- Write a boolean register to a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @BooleanRegister dbo.ModbusMaster.BooleanRegister;

-- Call the WriteBoolean method

SELECT dbo.ModbusMaster.WriteBoolean(@DeviceName, @BooleanRegister);

### 12. ****WriteFloats****

#### SQLCLR Members:

* **Method:**
  + public static SqlBoolean ModbusMaster.WriteFloats(SqlString deviceName, ModbusMaster.ModbusMaster.FloatRegisterArray floatRegisterArray)

#### SQL Example:

sql

-- Write float registers to a Modbus device

DECLARE @DeviceName NVARCHAR(512) = 'ModbusDevice.1';

DECLARE @FloatRegisterArray dbo.ModbusMaster.FloatRegisterArray;

-- Call the WriteFloats method

SELECT dbo.ModbusMaster.WriteFloats(@DeviceName, @FloatRegisterArray);

This documentation provides the complete list of SQLCLR methods and properties from the **ModbusMaster** files, along with SQL examples showing how to use them in a SQL Server environment. Let me know if you'd like further clarification or additional details! ​​

4o

Top of Form

Bottom of Form

ChatGPT can make mistakes. Check important info.

?