



"Script execute"

1		1
2		1
3	Script execute	2
4		3
5		4
6		5
1	5
2	6
3	6
4	7
5	7
6	7
7	8
7	?	10
1	10

1

Data Logger "Script execute" (, Advanced Serial
 Advanced TCP/IP Data Logger
 PascalScript, C++Script, JScript BasicScript
 Microsoft Scripting Host, Windows,
 Windows.
 "Script execute"
).
 :
 • : PascalScript, C++Script, BasicScript,
 JScript.
 • : , , , ,
 (,), (,
 case, try/finally/except, with), (int, float, bool, char, string,
 enum, variant), (, .).
 • (, TStrings, TFileStream).
 • , ,
 • ,
 • ,

2

Script execute :
 : Windows 2000 SP4 , 32-x 64-x
 - :
 : 5 MB
 :
 (), Advanced Serial Data Logger.

Microsoft Vista :

Program Files,

- 1.
- 2.
3. Windows Vista

Google.

3

Script execute

1. (, Advanced Serial Data Logger),

;

2. ;

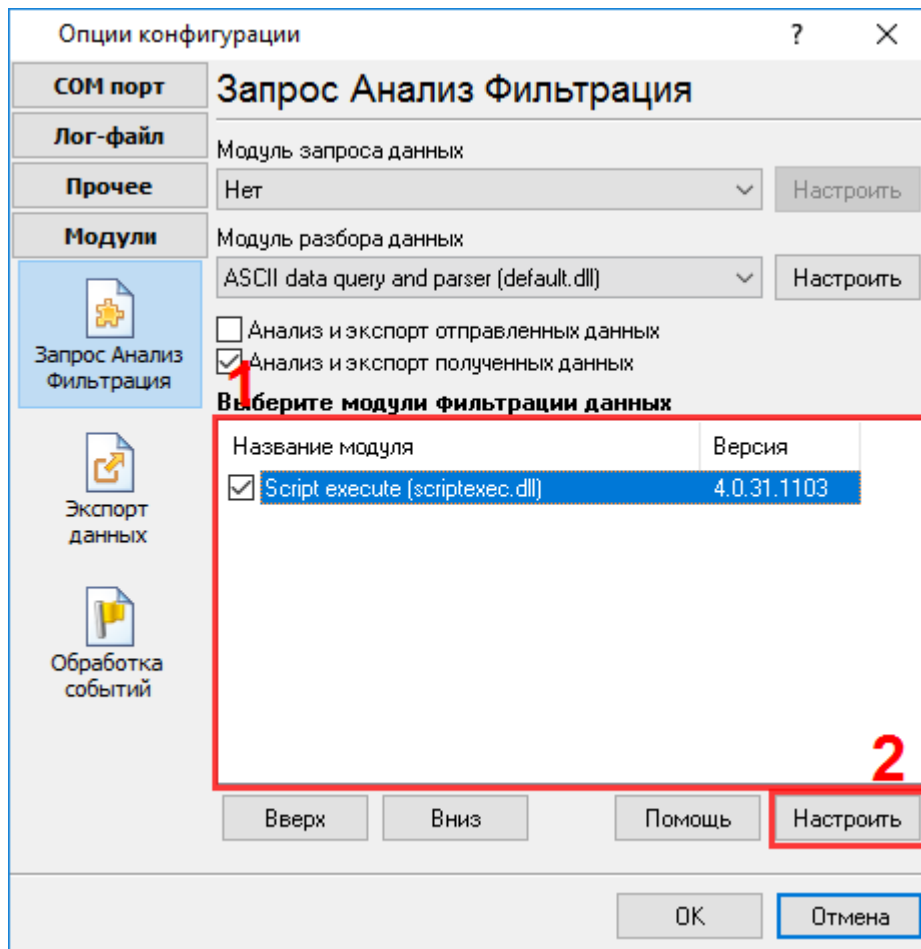
3. ,

Windows;

4. " " ;

5. " " , " "

.1-2.



.1.

4

Plug-in -

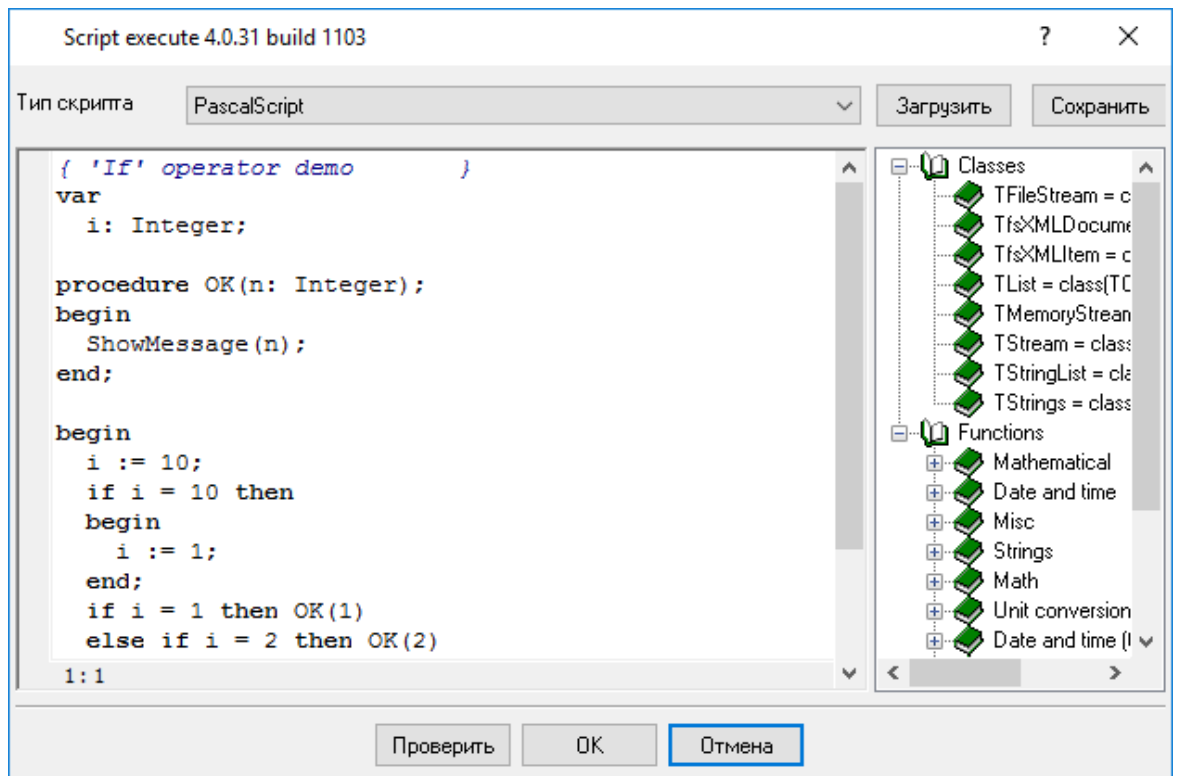
Advanced Serial Data Logger

5

(.1).

"Pascal"

"OK"



.1.

Key	Action
Cursor arrow	Cursor moving
PgUp, PgDn,	Page Up / Page Down
Ctrl+PgUp	Move to the begin of text
Ctrl+PgDn	Move to the end of text

Home	Move to the begin of line
End	Move to the end of line
Enter	Move to the next line
Delete	Delete symbol at right or selected text
Backspace	Delete symbol at left
Ctrl+Y	Delete current line
Ctrl+Z	Undo last change
Shift+	Select the text block
Ctrl+A	Select all text
Ctrl+U	Unindent selected block
Ctrl+I	Indent selected block
Ctrl+C, Ctrl+Insert	Copy to clipboard
Ctrl+V, Shift+Insert	Paste from clipboard
Ctrl+X, Shift+Delete	Cut to clipboard
Ctrl+Shift+< >	Set bookmark
Ctrl+< >	Goto bookmark
Ctrl+F	Search text
F3	Continue search

6

6.1

- sets ('IN' ; "a in
['a'..'c','d']"),
- shortstring
- GOTO.
- C++Script: ; 'break' SWITCH (SWITCH
CASE Pascal); '++' '--'
'++i); '--, '++' '=' 'if(++);

6.2

	Variant
Byte	Integer
Word	
Integer	
Longint	
Cardinal	
Boolean	Boolean
Real	Extended
Single	
Double	
Extended	
TDate	
TTime	
TDateTime	
Char	Char
String	String
Variant	Variant
Pointer	
Array	
C++Script	:
int, long = Integer	
void = Integer	
bool = Boolean	
float = Extended	
JScript	, variants.
BasicScript	(, dim i as Integer), Variant.
Object Pascal Extended	String Integer. Variant

6.3

```

Abs(e: Extended): Extended
ArcTan(X: Extended): Extended
Cos(e: Extended): Extended
Exp(X: Extended): Extended

```



```
Frac(X: Extended): Extended
Int(e: Extended): Integer
Ln(X: Extended): Extended
Pi: Extended
Round(e: Extended): Integer
Sin(e: Extended): Extended
Sqrt(e: Extended): Extended
Tan(X: Extended): Extended
Trunc(e: Extended): Integer
```

6.4

```
Chr(i: Integer): Char
CompareText(s, s1: String): Integer
Copy(s: String; from, count: Integer): String
Delete(var s: String; from, count: Integer)
Insert(s: String; var s2: String; pos: Integer)
Length(s: Variant): Integer
Lowercase(s: String): String
NameCase(s: String): String
Ord(ch: Char): Integer
Pos(substr, s: String): Integer
SetLength(var S: Variant; L: Integer)
Trim(s: String): String
Uppercase(s: String): String
```

6.5

```
Date: TDateTime
DayOfWeek(aDate: TDateTime): Integer
DaysInMonth(nYear, nMonth: Integer): Integer
DecodeDate(Date: TDateTime; var Year, Month, Day: Word)
DecodeTime(Time: TDateTime; var Hour, Min, Sec, MSec: Word)
EncodeDate(Year, Month, Day: Word): TDateTime
EncodeTime(Hour, Min, Sec, MSec: Word): TDateTime
IsLeapYear(Year: Word): Boolean
Now: TDateTime
Time: TDateTime
```

6.6

Other

```
CreateOleObject(ClassName: String): Variant
Dec(var i: Integer; decr: Integer = 1)
Inc(var i: Integer; incr: Integer = 1)
InputBox(ACaption, APrompt, ADefault: string): string
InputQuery(ACaption, APrompt: string; var Value: string): Boolean
MessageDlg(Msg: string; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons; HelpCtx: Longint): Integer
RaiseException(Param: String)
```

```

Random: Extended
Randomize
ShowMessage(Msg: Variant)
ValidDate(cDate: String): Boolean
ValidFloat(cFlt: String): Boolean
ValidInt(cInt: String): Boolean
VarArrayCreate(Bounds: Array; Typ: Integer): Variant
VarType(V: Variant): Integer

```

Conversion

```

DateTimeToStr(e: Extended): String
DateToStr(e: Extended): String
FloatToStr(e: Extended): String
IntToStr(i: Integer): String
StrToDate(s: String): Extended
StrToDateTime(s: String): Extended
StrToFloat(s: String): Extended
StrToInt(s: String): Integer
StrToTime(s: String): Extended
TimeToStr(e: Extended): String
VarToStr(v: Variant): String

```

Formatting

```

Format(Fmt: String; Args: array): String
FormatDateTime(Fmt: String; DateTime: TDateTime): String
FormatFloat(Fmt: String; Value: Extended): String
FormatMaskText(EditMask: string; Value: string): string

```

6.7

```

procedure SetVariable(Name: String; Value: Variant)

```

```

function GetVariable(Name: String):Variant

```

Null.

```

procedure PushVariable(Name: String; Value: Variant)

```

```

function PopVariable(Name: String):Variant

```

```

PushVariable.
Null.

```

```
procedure DeleteVariable(Name: String)

.

procedure DiscardDataPacket

.

function IsVariableDefined(Name: String):boolean

.

function IsVariableStored(Name: String):boolean

.

function SendData(Data: String):boolean

.

).      , Data = #01+'Test'+#02      (      ,

function SendString(Data: String):boolean

.

      #0D      ASCII

function SendByte(Data: Byte):boolean

.

function SendByte(Data: Byte):boolean

.

function SendByte(Data: Byte):boolean

.

function SendDataToDataSource(DataSource, Data: String):boolean

.

      SendString,
      DataSource

.

procedure DbgLog(Msg: String; LogLevel: Byte = 0)

.

      : 0 -      , 1 -      Msg      , 2 -      . LogLevel

.

procedure SendEvent(Name: String)
```

```

procedure SendEventEx(Name: String; Args: array of const)

    : SendEventEx('EVENT-ID', ['ValueName1', 0, 'ValueName2', 'Test'])

procedure RedirectData(TargetConfig: String)

function GetVariableCount:integer

function GetVariableByIndex(AIndex: integer):Variant

function GetVariableNameByIndex(AIndex: integer):string

procedure NewRow()

```

7**?****7.1**

```

" - "
"Plugins"

```

```

( )-

```

```

%s [%s] -

```

```

(%s) -

```

%s. (%s) –

%s. (%s) –

(,).

support@aggsoft.ru.

,"%s"