

"Python plugin"

1		1
2		1
3	Python plugin	1
4		2
5		3
6	API	3
7		5
8		6
9		8
10		8

1

Python 3. Python plugin Python
API.

Python.

2

Python plugin :

: Windows 2000 SP4 ,

32-x 64-x

5 MB

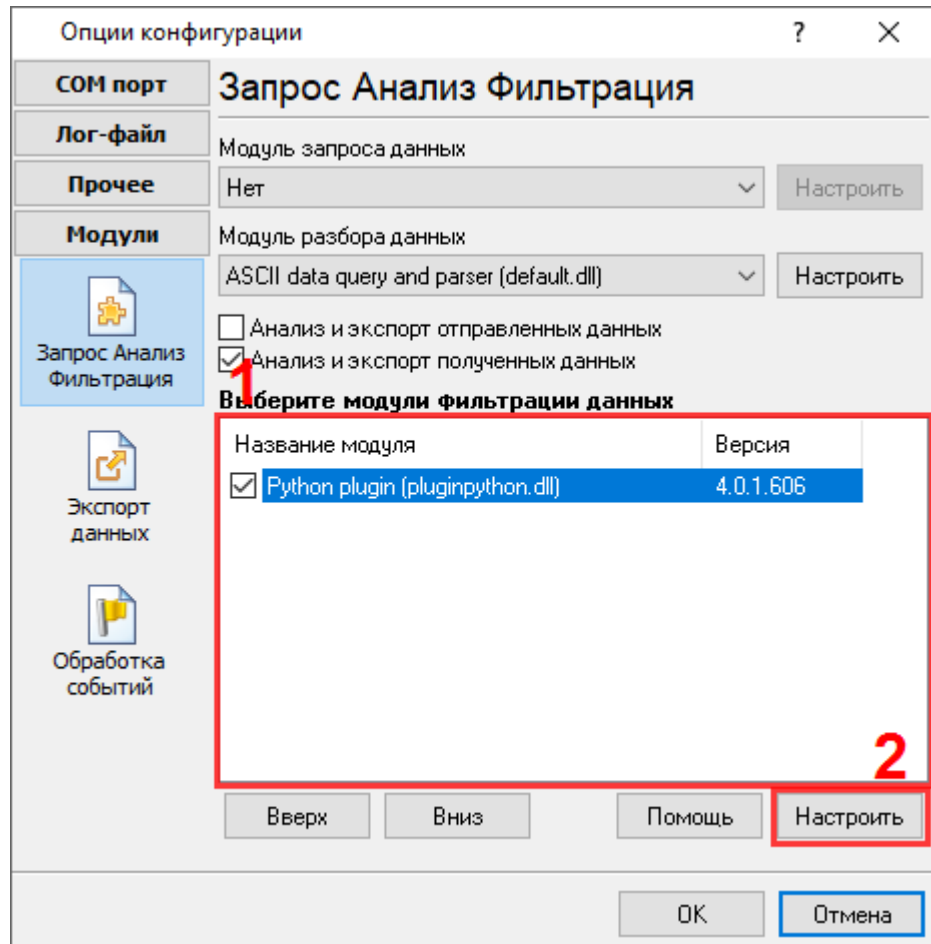
(), Advanced Serial Data Logger.

3

Python plugin

1. (, Advanced Serial Data Logger), ;
2. ;
3. ,
4. Windows;
5. " " ;
" . "

. 1-2.



. 1.

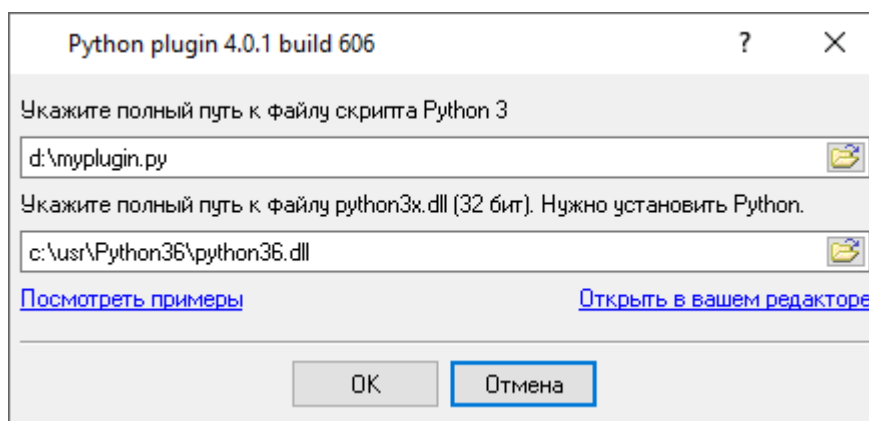
4

Plug-in -

Advanced Serial Data Logger

5

1. [Python 3](#) 32 (x86).
2. ,
3. (. 1).
4. python3x.dll ().



.1.

6 API

```
def FilterData(callparams, datasource, datain, dataout):
```

```
    "Python plugin"
```

```
callparams - _____
```



```
"Python plugin"
```

```
event_params - _____
```

```
datasource - _____
```

```
event - _____
```

```
(None).
```

```
def main():
```

```
:
```

```
def main():
```

```
    print('Loaded')
```

```
main()
```

7

```
Clear() - _____
```

```
Shift(number) - _____
```

```
"number"
```

```
(_____).
```

```
GetBytes() - _____
```

```
"bytes".
```

```
Shift
```

```
GetBytes,
```

```
"bytes"
```

```
:
```

```
>>> buffer.Clear()
```

```
>>> buffer.Shift(100)
```

```
>>> data = buffer.GetBytes()
```

8

,\$NEW_ROW\$.

Clear() -

```
>>> list.Clear()
```

Delete(index) -

index

```
>>> list.Delete(0)
```

CopyFrom(object [, start_index = 0[, end_index = -1]]) -

object.

start_index end_index

end_index -1,

```
>>> list1.CopyFrom(list2, 0, 10)
```

ItemByName(name) -

name ().

```
>>> item = list.ItemByName('VALUE')
```

ItemIndexByName(name) -

name.

```
>>> index = list.ItemIndexByName('VALUE')
```

ItemValueByName(name) -

None. name.

```
>>> value = list.ItemValueByName('VALUE')
```

ItemValueByNameDef(name, default) -

name.

default.

```
>>> value = list.ItemValueByNameDef('VALUE', 5)
```

InsertItem(index, name, value) -

name

value

index.

```
>>> item = list.InsertItem(0, 'VALUE', 5)
```

SetItem(name, value[, canadd = False]) -

value.

canadd

True,

name


```
>>> item = list.SetItem('VALUE', 5)
```

AddItem(name, value) - **name** **value**

```
>>> item = list.AddItem('VALUE', 5)
```

AddItemCopy(item) - **item**

```
>>> item_new = list.AddItemCopy(item_old)
```

FindRow(item, end_idx, is_row_end_sign) - **start_idx** **True** **(**
start_idx) **end_idx -** **,**
is_row_end_sign -
True, **end_idx**

```
from plugin import *
```

```
start_idx = 0
end_idx = CreateVarParam(-1)
is_row_end_sign = CreateVarParam(False)
row_cnt = 0
while datain.FindRow(start_idx, end_idx, is_row_end_sign):
    end_idx2 = end_idx.value
    print('Row: ' + str(row_cnt))
    print('Start index: ' + str(start_idx))
    print('End index: ' + str(end_idx2))
    if is_row_end_sign.value:
        end_idx2 -= 1
    row_cnt += 1
    start_idx = end_idx.value + 1
```

NewRow() -

```
>>> list.NewRow()
```

Count -

```
>>> count = list.Count
```

Items[index] -

```
>>> item = list.Items[0]
```

:

Name - ().

Value - . None.

9

CreateVarParam(value) - value

```
>>> from plugin import *
>>> obj = CreateVarParam(-1)
```

SendData(datasource, buffer) - buffer datasource.
buffer bytes bytearray. datasource
[API.](#) API. ,

```
>>> from plugin import *
>>> SendData(datasource, b'ABCD')
```

GetModuleParameter(name) - name.

```
>>> from plugin import *
>>> value = GetModuleParameter('ModulePath')
```

SetModuleParameter(name, value) - value
name.

```
>>> from plugin import *
>>> value = GetModuleParameter('ModulePath')
>>> SetModuleParameter('LogTitle', 'MyPlugin')
```

10

, API.

"[w]",

ModulePath - (string)

ModuleName - (string)

ApplicationFullName - (string)

