



"Plugin proxy"

1		1
2		1
3	Plugin proxy	1
4		2
5		3
6	API	3
7		7
8	API	7
9		12
10		13
11		14

1

"Plugin proxy" API, API "Python plugin",
 DLL API
 Visual C++ 2015,

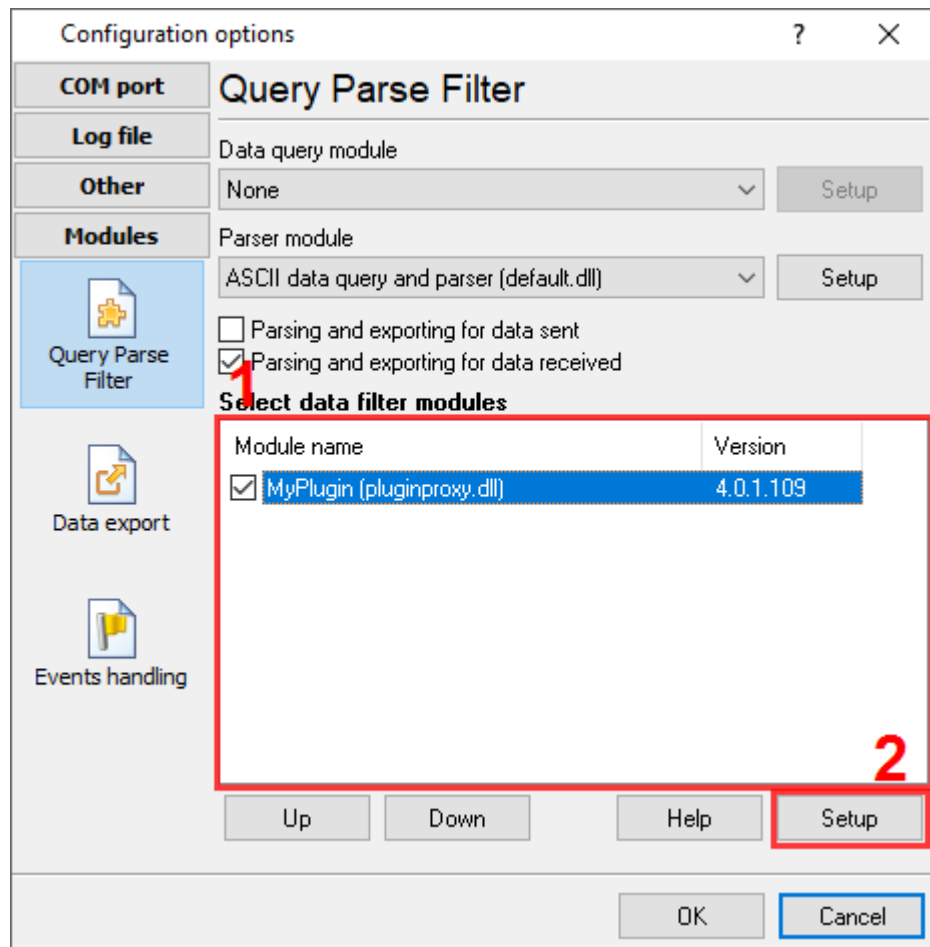
2

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

3

Plugin proxy

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



.1.

4

Plug-in -

Advanced Serial Data Logger

5

```

"Plugin proxy"
, API
.
"
"
.
:
1. DLL
2. pluginproxy.dll (
\plugins\pluginproxy).

```

6 API

```

, , Visual C++ 2015,
.
:
1. API
( aggPluginShowOptionsDialog).
2. DLL
.
3. DLL.
try ... catch.
.

```

aggPluginCreate

```

void __stdcall aggPluginCreate(_di_IaggPluginW Sender)

```

```

Sender - _____ API.
API.

```

```

,
:
1.
2. DLL.
(
).

```

aggPluginLoad.

aggPluginLoad

void __stdcall aggPluginLoad(_di_IaggPluginW Sender, const PPluginCallRec pCallRec, int &Result)

pCallRec - _____

Result -
mrtSuccess,

aggPluginUnload

void __stdcall aggPluginUnload(_di_IaggPluginW Sender, const PPluginCallRec pCallRec, int &Result)

Result -

aggPluginDestroy

void __stdcall aggPluginDestroy(_di_IaggPluginW Sender)

DLL

aggPluginLoadSettingsNow

void __stdcall aggPluginLoadSettingsNow(_di_IaggPluginW Sender)

1. aggPluginShowOptionsDialog

2.

aggPluginSaveSettingsNow

```
void __stdcall aggPluginSaveSettingsNow(_di_IaggPluginW Sender)
```

```
(  
    INI).
```

aggPluginShowOptionsDialog

```
void __stdcall aggPluginShowOptionsDialog(_di_IaggPluginW Sender, const PPluginCallRec pCallRec, bool &
```

```
"  
    "
```

```
Success -  
    TRUE,
```

```
    aggPluginLoadSettingsNow
```

aggPluginEvent

```
void __stdcall aggPluginEvent(_di_IaggPluginW Sender, const PWideChar Event, const PPluginCallRec pCall
```

```
Event -  
    "PORT-CLOSE".
```

```
Result -  
    , mrtSuccess.
```

```
    "          "
```

aggPluginExecuteCustomCommand

```
void __stdcall aggPluginExecuteCustomCommand(_di_IaggPluginW Sender, const PWideChar Command, con
```

```
Command -  
    "MY-COMMAND".
```

```
Result -  
    , mrtSuccess.
```

aggPluginSetParserItems

```
void __stdcall aggPluginSetParserItems(_di_IaggPluginW Sender)
```

```
    ,
    " (mttEventsHandling),
    " (mttFilter), " (mttDataExport),
    aggPluginShowOptionsDialog,
```

aggPluginProcessData

```
void __stdcall aggPluginProcessData(_di_IaggPluginW Sender, const PPluginCallRec pCallRec, int &
```

```
    ,
    " (mttDataQuery), " (mttDataParser) " (mttFilter), "
    " (mttDataExport), " (mttEventsHandling).
    " " ; pCallRec
```

```
Result - , mrtSuccess.
```

aggPluginGetData

```
void __stdcall aggPluginGetData(_di_IaggPluginW Sender, const PPluginCallRec pCallRec, int &Resu
```

```
    ,
    " " ,
    " "
```

aggPluginStateChanged

```
void __stdcall aggPluginStateChanged(_di_IaggPluginW Sender, const DWORD ModuleTypes)
```

```
    ,
    " "
```

```
ModuleTypes - ,
```


"ModuleEnabled",

7

API

```
mrtSuccess = 0.
mrtInvalidParameter = -1.
mrtNotSupported = -2.
mrtException = -3.
mrtNotInitialized = -4.
mrtError = -5.
```

"aggPluginModule.h".

8 API

"aggPluginModule.h"
++

API

"CaggPlugin"

```
mttDataQuery = 0x01.
mttDataParser = 0x02.
mttDataExport = 0x04.
mttEventsHandling = 0x08.
mttLogging = 0x10.
mttFilter = 0x20.
mttDeviceLayer = 0x40.
mttMenuExt = 0x80.
).
```

()

Data Logger Suite.

API.

(

:

```
TDataRecType : unsigned char {
    pdtNone, //
    pdtParamList, //
    pdtBuffer, //
    pdtObject, //
    pdtString, //
    pdtDWORD, //
    pdtCommand, //
```

```

        pParams.
        pData.
        pObject (
        ANSI ASCII,
        dwData.
        dwData (
        ).
        pStr.
```

```

    pdtNotify, // , dwData ( . ).

    pdtParamListObj // ( ).
};

:

pdtCommand_ClientConnect = 0x0101; // ( IP => ).
pdtCommand_ClientDisconnect = 0x0102; // ( )
( IP <=> ).
pdtCommand_TryRestoreConnection = 0x0103; // ( <=
).
pdtCommand_ClientBlock = 0x0104; // ( IP <= ).
pdtCommand_BufferCleanupEnable = $0105; // , ( <=
- ).
pdtCommand_BufferCleanupDisable = $0106; // ( <= -
).
pdtCommand_GetQueueSize = $0201; // ( => ).

struct TDataRec
{
    DWORD dwTag; //
    DWORD dwDataSize; //
    pdtBuffer pdtString.

    TDataRecType bDataType;
    union
    {
        struct
        {
            DWORD dwData;
        };
        struct
        {
            char *pStr;
        };
        struct
        {
            void* pObject;
        };
        struct
        {
            TaggParamsList* pParamsObj;
        };
        struct
        {
            void *pData;
        };
        struct

```

```

    {
        _di_laggParamsListA* pParams;
    };
};

```

API

```

struct TPluginCallRec
{
    DWORD dwSize; // TPluginCallRec
    Byte bVersion; //
    DWORD dwModuleHandle; //
    DWORD dwModuleIndex; //
    DWORD dwDataSource; //
    DWORD dwClientID; //
    , dwDataSource dwClientID
    TModuleTypes aCallModuleType; //
    API.
    TaggParamsList* aParameters_; //
    TDataRec aDataIn; //
    TDataRec aDataOut; //
    unsigned char aClientName[50]; // dwClientID.
    \_di\_laggParamsListA aParameters; //
};

typedef TPluginCallRec* PPluginCallRec;

```

API

0 (S_OK).

HRESULT __stdcall Log(const byte Level, const BSTR str)

str . Level : 0 - , 1
- , 2 -

HRESULT __stdcall GetModuleParameter(const BSTR AName, OleVariant* AResult)

AName.

HRESULT __stdcall SetModuleParameter(const BSTR AName, const OleVariant &Value, bool* AResult)

AName

Value.

HRESULT __stdcall NewData(_di_IaggParamsListW* List)

mttDataParser mttFilter.

HRESULT __stdcall PluginStorageAdd(const BSTR AId, const PVOID AData)

HRESULT __stdcall PluginStorageGet(const BSTR AId, PVOID* AData)

HRESULT __stdcall PluginStorageDel(const BSTR AId, PVOID* AData)

[API.](#)

AId -

AData -

HRESULT __stdcall GetParserItems(OleVariant* AResult)

(BSTR),

CR+LF

:

DESCRIPTION | NAME | DATA_TYPE | DEFAULT_VALUE

DESCRIPTION -

NAME - (, A-Z,

).

DATA_TYPE -

"String". : String, Memo, Bytes, Blob, Boolean, Float, Smallint, Word, Integer, Date, Time, DateTime, DWord, Byte, Shortint, Int64, Currency.

DEFAULT_VALUE -

HRESULT __stdcall GetConfigs(OleVariant* AResult)

(BSTR),

CR+LF

:

NAME<TAB>DATA_SOURCE_ID

NAME -
 <TAB> - ASCII (0x09).
 DATA_SOURCE_ID - HEX.
 DWORD

HRESULT __stdcall SendData(const PPluginCallRec pCallRec, const PBYTE AData, const int ASize, int* AResult)

pCallRec - NULL.

AData -

ASize -

AResult -

>= 0 -

-1 -

-2 -

-3 -

-4 -

-5 -

-6 - (, . .)

-7 - (()).

-8 - (()).

-9 -

-10 -

-11 -

-12 - ().

-13 - (, . .).

-14 -

HRESULT __stdcall GetArgs(const _di_IaggParamsListW* AArgs)

HRESULT __stdcall SendEvent(const BSTR AEventId, const _di_IaggParamsListW AArgs)

LOG-MESSAGE -
 NEW-LOG-FILE -
 LOG-FILE-DELETE -
 NEW-DATA-PACKET -
 ERROR-WRITE-FILE -
 PORT-OPEN - ().
 PORT-CLOSE -
 CONFIG-CHANGE -
 USER-LOGOOF, USER-LOGON -
 STOP-SERVICE -

10

, API
 65
 API aggPluginProcessData,
 "fifobuffer.h".
["CaggPlugin":](#)

CFifoBuffer GetDataSourceBuffer(const PPluginCallRec pCallRec)

CFifoBuffer :

Size -

Data -

void Clear() -

void Shift(const int ASize) - "ASize" ().

int Find(const PBYTE ASign, const byte ASignSize, const unsigned int Offset = 0)

Offset. "ASign" "ASignSize" "-1".

11

```

"aggParamLst.h"                                "CaggParamsList"
++
.
.
.
"$NEW_ROW$".
"CaggParamsList"

Clear() -

Delete(index) -                                index

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

ItemByName(name) -                             name ( ).

ItemIndexByName(name) -                        name.

ItemValueByName(name) -                       name.
                                                Null.

ItemValueByNameDef(name, default) -
name.                                          default.

InsertItem(index, name, value) -               name      value
index.

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

AddItem(name, value) -                         name      value

AddItemCopy(item) -                           item

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

```



```
int nStartIdx = 0;
int nEndIdx = 0;
bool bEndRowSign = false;
while (aData.FindRow(nStartIdx, nEndIdx, bEndRowSign))
{
    //
    // your code here
    //

    nStartIdx = nEndIdx + 1;
}
```

DataRow - `DataRow` class
Count - `Count` property
Items[index] - `Items[index]` property

Name - `Name` property
Value - `Value` property, "Null".