



# **Common Language Extension Interface for script**

# Contents

<b>Common Language Extension Interface for script.....</b>	<b>1</b>
1 Introduction.....	1
1.1 Hierarchy of objects and functions.....	1
1.2 Syntax of different script languages.....	1
1.3 Public variable type.....	2
1.4 Gloal object.....	3
1.5 Notes for java/c#.....	4
1.6 Notes for c# .net 4. x.....	6
1.7 Notes for callback function.....	6
2 Constant definition.....	7
2.1 Sync status.....	7
2.2 Alarm value.....	7
2.3 Object alloc type.....	7
2.4 Object save type.....	7
2.5 Object active command.....	7
2.6 Remotecall result.....	8
2.7 Remotecall source type.....	8
2.8 Attribute type.....	8
2.9 UUID.....	9
2.10 Virtual key code[win32].....	9
2.11 Program run type.....	9
2.12 Module type.....	10
2.13 OS type .....	10
2.14 System object.....	10
3 Struct definition .....	11
3.1 COLOR.....	11
3.2 RECT.....	11
3.3 FONT.....	11
3.4 TIME.....	12
3.5 Active set .....	13
3.6 SRP version.....	13
4 Object system event.....	13
4.1 _OnCreate( self, Event ).....	13
4.2 _OnDestroy(self, Event ).....	14
4.3 _OnCreateChild(self, Event ).....	14
4.4 _OnDestroyChild(self, Event ).....	14
4.5 _OnActivating(self, Event ).....	14
4.6 _OnDeactivating(self, Event ).....	15
4.7 _OnActivate(self, Event ).....	15
4.8 _OnDeactivate(self, Event ).....	15
4.9 _OnActiveChild(self, Event ).....	16
4.10 _OnDeactiveChild(self, Event ).....	16

4.11	<i>_OnParentBeforeChange(self, Event )</i> .....	16
4.12	<i>_OnParentChange(self, Event )</i> .....	17
4.13	<i>_OnStaticChange (self, Event )</i> .....	17
4.14	<i>_OnScriptChange(self, Event )</i> .....	17
4.15	<i>_OnSyncGroupChange(self, Event )</i> .....	17
4.16	<i>_OnChildSyncGroupChange(self, Event )</i> .....	18
4.17	<i>_OnActiveSetChange (self, Event )</i> .....	18
4.18	<i>_OnLoadMask (self, Event )</i> .....	18
4.19	<i>_OnLoadFinish(self, Event )</i> .....	18
4.20	<i>_OnTicket(self, Event ) only triggered to self</i> .....	19
4.21	<i>_OnFrameTicket(self, Event ) only triggered to self</i> .....	19
4.22	<i>_OnIdle(self, Event ) : only triggered to self</i> .....	19
4.23	<i>_OnAppActive(self, Event ) only triggered to self</i> .....	19
4.24	<i>_OnAppDeactive(self, Event ) only triggered to self</i> .....	20
4.25	<i>_OnServiceActive(self, Event ) only triggered to self</i> .....	20
4.26	<i>_OnServiceDeactive(self, Event ) only triggered to self</i> .....	20
4.27	<i>_OnRemoteSend( self, Event )</i> .....	20
4.28	<i>_OnCall ( Event )</i> .....	21
5	<i>Interface function</i> .....	21
5.1	<i>Init CLE</i> .....	21
5.2	<i>Callback and control functions</i> .....	22
5.3	<i>Global function</i> .....	28
5.3.1	<i>OS type and time</i> .....	29
5.3.2	<i>registry query function</i> .....	29
5.3.3	<i>32bit operation</i> .....	29
5.3.4	<i>file find function</i> .....	30
5.3.5	<i>byte order change</i> .....	30
5.3.6	<i>Shell execute command[Windows,do not support C#]</i> .....	30
5.3.7	<i>virtual key status[Windows,do not support C#]</i> .....	30
5.3.8	<i>intger compare</i> .....	32
5.3.9	<i>get UUID</i> .....	32
5.3.10	<i>get current Url</i> .....	32
5.3.11	<i>set program run type [valid at server]</i> .....	33
5.3.12	<i>find char position</i> .....	33
5.3.13	<i>ID to MD5 string</i> .....	33
5.3.14	<i>get system path</i> .....	33
5.3.15	<i>set log file</i> .....	33
5.3.16	<i>authorize</i> .....	33
5.3.17	<i>set locale of cle</i> .....	34
5.3.18	<i>find window handle[win32]</i> .....	34
5.3.19	<i>get script interface index</i> .....	34
5.3.20	<i>get script interface version</i> .....	34
5.3.21	<i>get or set env</i> .....	34
5.3.22	<i>set script interface module</i> .....	35

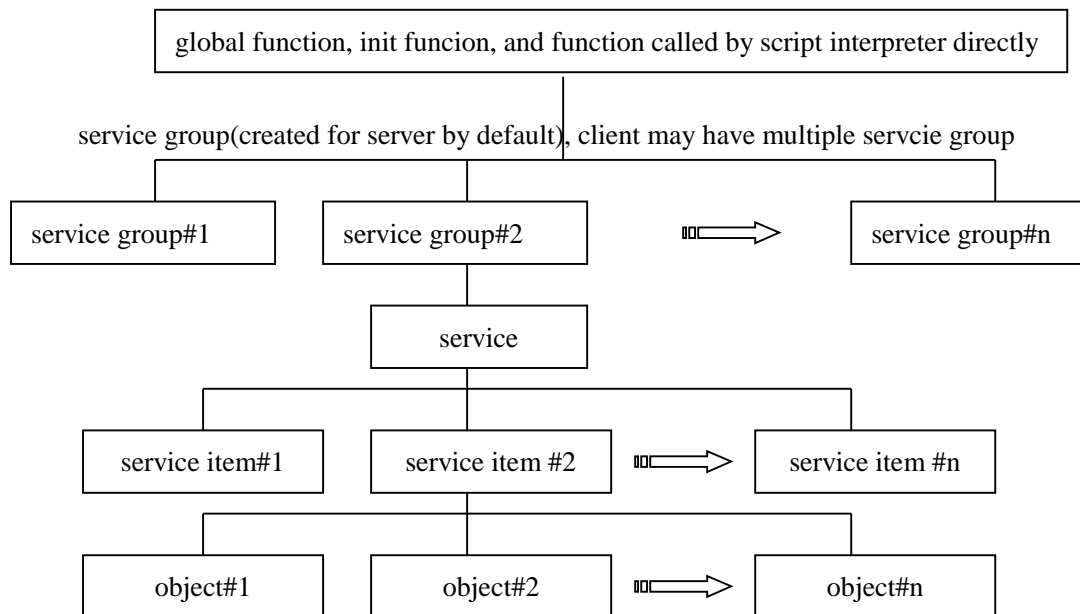
5.3.23	<i>Detach Thread</i> .....	35
5.3.24	<i>Set Operation Path</i> .....	36
5.3.25	<i>String Code Conversion</i> .....	36
5.3.26	<i>Get Core Share Library Handle</i> .....	36
5.3.27	<i>Inject Script Class</i> .....	36
5.4	<i>Service group attribute</i> .....	36
5.5	<i>Service group function</i> .....	37
5.5.1	<i>public functions</i> .....	37
5.5.2	<i>service sync function(valid at client)</i> .....	39
5.5.3	<i>service management function[called at server]</i> .....	40
5.5.4	<i>client connect to server</i> .....	43
5.5.5	<i>trigger hyper connection</i> .....	45
5.5.6	<i>trigger app event</i> .....	45
5.5.7	<i>QueryRecord object</i> .....	45
5.5.8	<i>parapkg object</i> .....	45
5.5.9	<i>binbuf object</i> .....	46
5.5.10	<i>SXML object</i> .....	46
5.5.11	<i>function para object</i> .....	46
5.5.12	<i>CommInterface object</i> .....	46
5.5.13	<i>execute script segments or files</i> .....	46
5.5.14	<i>others function</i> .....	47
5.5.15	<i>RawSocket function</i> .....	49
5.5.16	<i>Timer function</i> .....	51
5.5.17	<i>Restart function</i> .....	52
5.5.18	<i>Http up/download function</i> .....	52
5.5.19	<i>set current Url and service parameter</i> .....	53
5.5.20	<i>script edit[Windows]</i> .....	53
5.5.21	<i>service path</i> .....	54
5.5.22	<i>get doc object registered</i> .....	54
5.5.23	<i>get static data version</i> .....	54
5.5.24	<i>Clipboard[Windows]</i> .....	54
5.5.25	<i>Load service from Url</i> .....	55
5.5.26	<i>Set debug and client port</i> .....	56
5.5.27	<i>set Telnet, Web, and output port</i> .....	56
5.5.28	<i>service register and alloc Cooperator</i> .....	56
5.5.29	<i>WebService object refresh</i> .....	57
5.5.30	<i>get WSDL</i> .....	57
5.5.31	<i>string format convert</i> .....	57
5.5.32	<i>get platform information</i> .....	57
5.5.33	<i>client download callback (used at server service group)</i> ...58	
5.5.34	<i>Dispatch callback</i> .....	58
5.5.35	<i>Raw Object Interface</i> .....	59
5.5.36	<i>Get Last Error</i> .....	60
5.5.37	<i>UnLockGC Log</i> .....	60

5.5.38	<i>_SunLockGC</i> .....	61
5.5.39	<i>Get Path and Local IP</i> .....	61
5.5.40	<i>Get Number of Object created[2.5.1]</i> .....	61
5.5.41	<i>Active Script and Pre-compile Script Segment [2.5.1]</i> .....	61
5.6	<i>Service object</i> .....	61
5.6.1	<i>attribute</i> .....	61
5.6.2	<i>basic function</i> .....	62
5.6.3	<i>client login function</i> .....	64
5.6.4	<i>object operation callback—valid at server [object change, create or delete]</i> 66	
5.6.5	<i>service redirect</i> .....	67
5.6.6	<i>file upload and download function</i> .....	68
5.6.7	<i>create object (base class not exist)</i> .....	69
5.6.8	<i>service macro definition</i> .....	69
5.6.9	<i>user management—valid at server, service object should be get with root user</i> .....	69
5.6.10	<i>execute script</i> .....	70
5.6.11	<i>get peer ip address</i> .....	70
5.6.12	<i>get server ID at client</i> .....	71
5.6.13	<i>force to save service static data (valid at server side)</i> .71	
5.6.14	<i>remove expired data</i> .....	71
5.6.15	<i>static data http download callback—valid at server side</i> .71	
5.6.16	<i>get object string from Lua</i> .....	72
5.6.17	<i>Pack static data</i> .....	72
5.6.18	<i>Xml file</i> .....	72
5.6.19	<i>atomic function</i> .....	74
5.6.20	<i>output service header and skeleton file</i> .....	82
5.6.21	<i>object' s EditLog/Checkpoint[Reserved]</i> .....	82
5.6.22	<i>Authorize</i> .....	83
5.6.23	<i>ShareLibary functions</i> .....	83
5.6.24	<i>Object Group Management</i> .....	83
5.6.25	<i>Get Control Service</i> .....	84
5.6.26	<i>Raw Object Interface</i> .....	84
5.6.27	<i>Get Last Error</i> .....	87
5.6.28	<i>Attribute Get</i> .....	87
5.7	<i>Service item object</i> .....	87
5.7.1	<i>Attribute</i> .....	87
5.7.2	<i>Functon</i> .....	87
5.7.3	<i>Attribute Get or Set</i> .....	88
5.7.4	<i>Callback funcion</i> .....	88
5.8	<i>Object' s predefined attribute and function</i> .....	89
5.9	<i>object defined attributes and functions</i> .....	113
5.9.1	<i>dynamic script language</i> .....	113
5.9.2	<i>static script language[java or c#]</i> .....	114

5.10	<i>ParaPkg object</i> .....	114
5.10.1	<i>attribute</i> .....	114
5.10.2	<i>function</i> .....	114
5.10.3	<i>Attribute Get</i> .....	117
5.10.4	<i>Dict and JSON function</i> .....	117
5.11	<i>Binbuf object</i> .....	117
5.11.1	<i>attribute</i> .....	118
5.11.2	<i>function</i> .....	118
5.11.3	<i>Attribute Get</i> .....	121
5.12	<i>SXML object</i> .....	121
5.12.1	<i>load and save</i> .....	121
5.12.2	<i>read</i> .....	122
5.12.3	<i>change</i> .....	123
5.12.4	<i>duplicate</i> .....	124
5.12.5	<i>Free</i> .....	124
5.13	<i>FunctionPara object</i> .....	124
5.13.1	<i>function</i> .....	124
5.14	<i>CommInterface object</i> .....	125
5.14.1	<i>attribute</i> .....	125
5.14.2	<i>function</i> .....	126
5.14.3	<i>Attribute Get or Set</i> .....	133
5.15	<i>Object garbage collect</i> .....	133

# 1 Introduction

## 1.1 Hierarchy of objects and functions



## 1.2 Syntax of different script languages

Different script may have different syntax. The document defines attributes and functions of objects. But how to write applications depends on specific script language. As example:

for lua:

```
obj = Service._New()
obj.p1 = 123
```

for python:

```
obj = Service._New()
obj.p1 = 123
```

for ruby:

```
obj = Service._New()
obj.p1 = 123
```

In static languages such as java or c#, application uses `_Get` and `_set` functions to access object attributes:

```
_GetInt(String Name);
_GetBool(String Name);
```

```

_GetDouble(String Name);
_GetStr(String Name);
Object _Get(String Name);

_Set(String Name,Object Para);

```

Examples in the document mostly use python language.

Notes of callback functions are used in applications:

For lua/python: callback directly uses script function object

For JAVA/c#: callback function defined in extending class, and uses function's name

For ruby, the callback function is method or proc. It can also be set to nil and then use block, for example,

```
_MsgLoop ( nil ) | true |
```

### 1.3 Public variable type

For C++, lua, python, java, c#, the following types are common, and may be transfered between different languages.

Simple types:

bool type(bool).

integer type(integer)

float type(double)

string type(string)

tuple type(lua table[index start from 1],Python Tuple, ruby array, array,java Object[],c# Object[]).

Tuple only supports one-dimensional. If there are tuples included in tuple, the map may be incorrect.

Mapping table:

to from	lua table	python tuple	ruby array	java Object[]	c# Object[]	Object Property	C/C++ ParaPkg Tuple
lua table		O	O	O	O	O	O
python tuple	O		O	O	O	O	O
ruby array	O	O		O	O	O	O
java Object[]	O	O	O		O	O	O
c# Object[]	O	O	O	O			O
Object Property	Struct table	Struct object	Struct class				



C++	ParaPkg	O	O	O	O	O		
Tuple								

Agreement for return type:

Single return value uses simple type.

For multiple return value:

Lua: return multiple value directly

Python: return tuple

ruby: return array

Java: return Object[]

C#: return object[]

If attribute is struct defined by C/C++, then use array or tuple to set its value, and struct class object for get its value.

#### 1.4 Global object

For each language, there are predefined global variables. By them, you can create other objects, or get predefined constants.

lua:

```
require "libstarcore"
```

```
libstarcore is global variable;
```

python:

```
import libstarpy
```

```
libstarpy is global variable;
```

ruby:

```
if (defined? Libstar_ruby) == nil
```

```
  require "D:\\Work\\starcore\\core\\starcore.ruby\\libruby193\\libstar_ruby.so"
```

```
#   require "libstar_ruby"
```

```
end
```

```
$starruby.XXXX
```

```
$starruby is global variable;
```

java: StarCoreFactory is single instance class.

```
StarCoreFactory starcore= StarCoreFactory.GetFactory();
```

```
starcore is global variable;
```

c#: StarCoreFactory is single instance class.

```
StarCoreFactory starcore= StarCoreFactory.GetFactory();
```

starcore is global variable;.

### 1.5 Notes for java/c#

Java/c# is static languages, which should define classes before compile, and then create instances of the classes.

Among objects defined in StarCore, besides the following three simple types:

StarTimeClass

StarRectClass

StarFontClass

Attributes of instance of other class can be accessed by \_Get. Input parameter is attribute name, and the returned value is Object type. For convenient, cle also provides \_GetInt,GetBool,GetDouble,\_GetStr to return corresponding type of object, and provides \_Getint,\_Getbool,Getdouble function to return the value of simple type, and provides \_Toint,\_Tobool,\_Todouble function to change object to simple type conveniently.

Using \_Set function to set attribute value: \_Set(attribute name, value)

Pre-defined class in starcore:

StarBinBufClass

StarCommInterfaceClass

StarCoreFactory

StarFunctionParaClass

StarObjectClass

StarParaPkgClass

StarQueryRecordClass

StarServiceClass

StarServiceItemClass

StarSrvGroupClass

StarStructClass

StarSXmlClass

StarCoreFactory is class factory. Instance of other class should be created by using it's interface function other than new method. StarCoreFactory is single instance class, which can be obtained by static function GetFacroty, for example:

```
StarCoreFactory starcore= StarCoreFactory.GetFactory();
StarSrvGroupClass SrvGroup = starcore._GetSrvGroup(0);
```

For StarServiceItemClass's function \_ClientToSync, StarCommInterfaceClass's function \_MsgProc and \_WebServerProc,object's system event functions, object's event process functions, \_OnNameValueChange, and \_OnChange, you can define extending class to overwrite these functions. For example, StarCommInterfaceClass defines function \_MsgProc, if you want to overwrite it, you should do as follow:

```
class WebServer_CommInterface extends StarCommInterfaceClass{
```

```
public void _MsgProc(int uMes,Object[] Msg){  
    ...  
}  
public WebServer_CommInterface(StarCommInterfaceClass srcobj){  
    super(srcobj);  
}  
}
```

Using the following format to create instance :

```
WebServer_CommInterface    CommInterface    =    new    WebServer_CommInterface  
(SrvGroup._NewCommInterface());
```

Another method:

```
StarCommInterfaceClass CommInterface = SrvGroup._NewCommInterface();  
CommInterface = CommInterface._Assign( new StarCommInterfaceClass (){  
    public void _MsgProc(int uMes,Object[] Msg){  
        ...  
    }  
})
```

CLE will find the defined functions by reflect mechanism.

For StarObjectClass, besides callback function listed above, you can define functions, which may be called by other languages, or attributes, which can be access by other languages.

These function and attributes should be declared with public. For attributes, their type should be boolean, int, double, String.

For function's parameter and return value type supported may be double, bool, integer, string, StarParaPkgClass, StarBinBufClass and StarObjectClass, and the first parameter must be StarObjectClass. When being called, the first parameter is self object attached by CLE. For example:

```
class MyObjectClass extends StarObjectClass{  
    public int GetNumber( StarObjectClass self,int para )  
    {  
        System.out.println("Remote Call Number "+para);  
        return para + 1;  
    }  
    public MyObjectClass(StarObjectClass srcobj){  
        super(srcobj);  
    }  
}
```

```
StarObjectClass Obj = Service._New();  
Obj = Obj._Assign(new StarObjectClass(){  
    public int GetNumber( StarObjectClass self,int para )
```

```

    {
        System.out.println("Remote Call Number "+para);
        return para + 1;
    }
}
)

```

## 1.6 Notes for c# .net 4.x

For .net4.0, or above, the interface name is “csharp4” or “csharp45” or “csharp451”, and the assembly name is “Star\_csharp4” or “Star\_csharp45” or “Star\_csharp451”. From this version, cle supports dynamic keyword.

StarBinBufClass  
 StarCommInterfaceClass  
 StarCoreFactory  
 StarFunctionParaClass  
 StarObjectClass  
 StarParaPkgClass  
 StarQueryRecordClass  
 StarServiceClass  
 StarServiceItemClass  
 StarSrvGroupClass  
 StarStructClass  
 StarSXmlClass

The above classes override TryGetMember/TrySetMember methods.

StarObjectClass and StarParaPkgClass override TrySetIndex/TryGetIndex methods.

StarObjectClass override TryInvoke/ TryInvokeMember methods

for example:

Service.\_ID equals to Service.\_Get(“\_ID”);

ParaPkg[1] equals to ParaPkg.\_Get(1)

```

dynamic obj = Service._ImportRawContext("lua", "", false, "");
/--call lua function tt, the return contains two integer, which will be packed into parapkg
---*/
dynamic ParaPkg = obj.tt("hello ", "world");
Console.WriteLine("ret from lua : " + ParaPkg._Number+ " " + ParaPkg[0] + " " + ParaPkg[0]);

```

## 1.7 Notes for callback function

For lua or python, callback is lua or python function.

For other scripts, such as java, c#, callback is name of the function.

For ruby, the callback function is method or proc. It can also be set to nil and then use block, for example,

`_MsgLoop ( nil ) | true |`

## *2 Constant definition*

Constant can be accessed through global object and service object

### *2.1 Sync status*

SYNC\_NOT

SYNC

SYNC\_IN

### *2.2 Alarm value*

FAULT\_IND

FAULT\_WARN

FAULT\_NORMAL

FAULT\_CRITICAL

FAULT\_SYSTEM

### *2.3 Object alloc type*

ALLOC\_STATIC

ALLOC\_GLOBAL

ALLOC\_CLIENT

ALLOC\_LOCAL

### *2.4 Object save type*

SAVE\_SAVE

SAVE\_LOCAL

SAVE\_GLOBAL

SAVE\_NONE

### *2.5 Object active command*

ACTIVE\_ALONE

ACTIVE\_FOLLOW

ACTIVE\_ACTIVE

ACTIVE\_DEACTIVE

## *2.6 Remotecall result*

RCALL\_OK  
RCALL\_COMMERROR  
RCALL\_OBJNOTEXIST  
RCALL\_FUNCNOTEXIST  
RCALL\_PARAERROR  
RCALL\_SYSERROR  
RCALL\_INVALIDUSR  
RCALL\_OVERTIME  
RCALL\_UNKNOWN

## *2.7 Remotecall source type*

RCALLSRC\_C  
RCALLSRC\_SCRIPT  
RCALLSRC\_WEBSERVICE

## *2.8 Attribute type*

TYPE\_BOOL  
TYPE\_INT8  
TYPE\_UINT8  
TYPE\_INT16  
TYPE\_UINT16  
TYPE\_INT32  
TYPE\_UINT32  
TYPE\_FLOAT  
TYPE\_LONG  
TYPE\_ULONG  
TYPE\_LONGHEX  
TYPE\_ULONGHEX  
TYPE\_VSTRING  
TYPE\_PTR  
TYPE\_MEMORY  
TYPE\_STRUCT  
TYPE\_COLOR  
TYPE\_RECT  
TYPE\_FONT  
TYPE\_TIME  
TYPE\_CHAR  
TYPE\_UUID  
TYPE\_STATICID

TYPE\_CHARPTR  
TYPE\_PARAPKGPTR  
TYPE\_BINBUFPTR

TYPE\_INT8PTR  
TYPE\_UINT8PTR  
TYPE\_INT16PTR  
TYPE\_UINT16PTR  
TYPE\_INT32PTR  
TYPE\_UINT32PTR  
TYPE\_FLOATPTR  
TYPE\_ULONGPTR  
TYPE\_LONGPTR  
TYPE\_STRUCTPTR  
TYPE\_COLORPTR  
TYPE\_RECTPTR  
TYPE\_FONTPTR  
TYPE\_TIMEPTR  
TYPE\_UUIDPTR  
TYPE\_VOID

TYPE\_OBJPTR  
TYPE\_TABLE

TYPE\_INT64  
TYPE\_DOUBLE  
TYPE\_INT64PTR  
TYPE\_DOUBLEPTR

TYPE\_UWORD  
TYPE\_UWORDPTR

## *2.9 UUID*

INVALID\_UUID

## *2.10 Virtual key code[win32]*

Refer to function “\_GetKeyState”.

## *2.11 Program run type*

VS\_SERVER  
VS\_CLIENT

VS\_DEBUG  
VS\_TOOLS  
VS\_SERVER\_SERVER  
VS\_SERVER\_USER  
VS\_CLIENT\_USER  
VS\_CLIENT\_COOPERATOR  
VS\_CLIENT\_CALLER

### *2.12 Module type*

VSMODULE\_SERVER\_SERVER  
VSMODULE\_SERVER\_USER  
VSMODULE\_CLIENT\_USER  
VSMODULE\_CLIENT\_COOPERATOR  
VSMODULE\_CLIENT\_CALLER  
VSMODULE\_DEBUG

### *2.13 OS type*

VSOS\_WIN32  
VSOS\_LINUX  
VSOS\_ANDROID  
VSOS\_ANDROIDV7A  
VSOS\_ANDROIDX86  
VSOS\_IOS  
VSOS\_WP  
VSOS\_WINRT  
VSOS\_MACOS  
VSOS\_WIN10

### *2.14 System object*

System object is only one, should not create it's instances.

System object can be get by **ServiceGroup.\_SysObject**. It defines the following events:

OnWndAdjust, OnWndResize, OnWndCanBeResize, OnEditSelect, OnSetFocus, OnWinMsg

Whether to create these events depends on application.

```
def SrvGroup_SysObject_OnWndCanBeResize(self, Event ) :  
    print(Event)  
    return True,1 // return 1 means forbid to change manager window size, and 0 means allow.  
SrvGroup._SysObject.OnWndCanBeResize = SrvGroup_SysObject_OnWndCanBeResize
```



### 3 Struct definition

#### 3.1 COLOR

Decimal number.

#### 3.2 RECT

```
python: { 'Value':((INT)Left,(INT)Top,(INT)Right,(INT)Bottom), 'Type':'Rect' }  
lua : { Value = { (INT)Left,(INT)Top,(INT)Right,(INT)Bottom }, Type = "Rect"}  
ruby: { :Value => ((INT)Left,(INT)Top,(INT)Right,(INT)Bottom), :Type => 'Rect' }
```

```
java:  
public class StarRectClass{  
    int left;  
    int top;  
    int right;  
    int bottom;  
};  
  
c#:  
public ref struct StarRectClass{  
    Int32 left;  
    Int32 top;  
    Int32 right;  
    Int32 bottom;  
};
```

#### 3.3 FONT

```
python:      { 'Value':((INT)Height,(INT)Size,(BYTE)CharSet,(BYTE)Style,(String)Name),  
                'Type':'Font' }  
lua: { Value = { (INT)Height,(INT)Size,(BYTE)CharSet,(BYTE)Style,(String)Name }, Type =  
        "Font"}  
ruby: { :Value => { (INT)Height,(INT)Size,(BYTE)CharSet,(BYTE)Style,(String)Name }, :Type  
        => "Font" }
```

```
java:  
class StarFontClass{  
    int Height;  
    int Size;  
    byte CharSet;  
    byte Style;  
    String Name;  
};  
  
c#:  
public ref struct StarFontClass{
```

```
    Int32 Height;  
    Int32 Size;  
    byte CharSet;  
    byte Style;  
    String^ Name;  
};
```

Height is negative.

CharSet = 134 is GB2312\_CHARSET

Size may be 0, which represents default value.

Style may be 0, or the sum of

LOCALFONT\_BOLD : 1

LOCALFONT\_ITALIC : 2

LOCALFONT\_UNDERLINE : 4

LOCALFONT\_STRIKEOUT : 8

.

Name: font name.

For example

1. XX: { 'Value':(-29,0,134,0,"XX\_GB2312"), 'Type': 'Font' }

### 3.4 TIME

Python: { 'Value': (WORD wYear,WORD wMonth,WORD wDay,WORD wHour,WORD wMinute,WORD wSecond,WORD wMilliseconds), 'Type': 'Time' }

lua: { Value = {WORD wYear,WORD wMonth,WORD wDay,WORD wHour,WORD wMinute,WORD wSecond,WORD wMilliseconds} ,Type="Time" }

ruby: { :Value => (WORD wYear,WORD wMonth,WORD wDay,WORD wHour,WORD wMinute,WORD wSecond,WORD wMilliseconds) , :Type => "Time" }

java:StarTimeClass.

```
class StarTimeClass{  
    short wYear;  
    short wMonth;  
    short wDayOfWeek;  
    short wDay;  
    short wHour;  
    short wMinute;  
    short wSecond;  
    short wMilliseconds;  
};  
  
c#:  
public ref struct StarTimeClass  
{  
    unsigned short wYear;  
    unsigned short wMonth;  
    unsigned short wDayOfWeek;  
    unsigned short wDay;  
    unsigned short wHour;  
    unsigned short wMinute;  
    unsigned short wSecond;  
    unsigned short wMilliseconds;  
};
```

```
};
```

### 3.5 Active set

python: (number, number, number,...)

lua: { number, number, number,... }

ruby: { number, number, number,... }

lua uses table, python uses tuple, JAVA/c# uses Object[].

### 3.6 SRP version

( (BYTE)MainVersion,(BYTE)SubVersion,(SHORT )BuildVersion )

lua uses table,python uses tuple,ruby uses array, JAVA/c# uses Object[].

## 4 Object system event

Note: if handler has been defined in C++, then script event functions will not be called.

Event:

lua -> table

python -> dict

ruby -> hash

java/c# -> HashTable

**note : for windows phone, Dictionary<object,object> should be used**

Exmaple:

```
def Object_OnDestroy(self,Event) :
```

```
    Print( self, Event )
```

Return True, 0 #---The first return value is True, means the event has been processed, and followed by return values.

```
Object._OnDestroy = Object_OnDestroy  #Set object event handler
```

```
function Object:_OnDestroy (Event)
```

```
    print( self, Event )
```

return true, 0 --- The first return value is true, means the event has been processed, and followed by return values.

```
end
```

```
public Object[] _OnRemoteSend(Hashtable Ev)
```

```
{
```

```
    return null;  or
```

```
    return new Object[]{true,XX};
```

```
}
```

#### 4.1 \_OnCreate( self, Event )

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self==False instance  
//--OUT none
```

```
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.2 *\_OnDestroy(self, Event )*

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.3 *\_OnCreateChild(self, Event )*

```
//--IN Event['_Arg0'] = ChildObject  
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._Arg0 = ChildObject  
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.4 *\_OnDestroyChild(self, Event )*

```
//--IN Event['_Arg0'] = ChildObject  
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._Arg0 = ChildObject  
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.5 *\_OnActivating(self, Event )*

```
//--IN Event['_Arg0'] = 0 normal activated 1 re-activated for load
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT  number= 0 success, otherwise fail
```

```
//--IN  Event._Arg0 = 0  normal activated 1 re-activated for load
//--IN  Event._DesObject =Object
//--IN  Event._ThisObject =true self ==false instance
//--OUT  number= 0 success, otherwise fail
```

The event is triggered only when object is in sync status.

#### 4.6 *\_OnDeactivating(self, Event )*

```
//--IN Event['_Arg0'] = 0 normal deactivated 1 re-deactivated for load
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT  none
```

```
//--IN  Event._Arg0 = 0  normal deactivated 1 re-deactivated for load
//--IN  Event._DesObject =Object
//--IN  Event._ThisObject =true self ==false instance
//--OUT  none
```

#### 4.7 *\_OnActivate(self, Event )*

```
//--IN Event['_Arg0'] = 0 normal activated 1 re-activated for load
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT  none
```

```
//--IN  Event._Arg0 = 0  normal activated 1 re-activated for load
//--IN  Event._DesObject =Object
//--IN  Event._ThisObject =true self ==false instance
//--OUT  none
```

The event is triggered only when object is in sync status.

#### 4.8 *\_OnDeactivate(self, Event )*

```
//--IN Event['_Arg0'] = 0 normal deactivated 1 re-deactivated for load
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT  none
```

```
//--IN Event._Arg0 = 0 normal deactivated 1 re-deactivated for load
//--IN Event._DesObject =Object
//--IN Event._ThisObject =true self ==false instance
//--OUT none
```

#### 4.9 *\_OnActiveChild(self, Event )*

```
//--IN Event['_Arg0'] = ChildObject
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT none
```

```
//--IN Event._Arg0 = ChildObject
//--IN Event._DesObject =Object
//--IN Event._ThisObject =true self ==false instance
//--OUT none
```

#### 4.10 *\_OnDeactiveChild(self, Event )*

```
//--IN Event['_Arg0'] = ChildObject
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT none
```

```
//--IN Event._Arg0 = ChildObject
//--IN Event._DesObject =Object
//--IN Event._ThisObject =true self ==false instance
//--OUT none
```

#### 4.11 *\_OnParentBeforeChange(self, Event )*

```
//--IN Event['_Arg0'] = NewParentObject
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self ==False instance
//--OUT number== 0 permit change, otherwise not allow and is the error code.
The event is valid for object is under local control.
```

```
//--IN Event._Arg0 = NewParentObject
//--IN Event._SrcObject = nil
//--IN Event._DesObject =Object
//--IN Event._ThisObject =true self ==false instance
//--OUT number== 0 permit change, otherwise not allow and is the error code.
The event is valid for object is under local control.
```

#### 4.12 *\_OnParentChange(self, Event )*

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.13 *\_OnStaticChange (self, Event )*

```
//--IN Event['_Arg0'] = AttributeNamex  
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._Arg0 = AttributeName  
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.14 *\_OnScriptChange(self, Event )*

```
//--IN Event['_Arg0'] = ScriptName  
//--IN Event['_Arg1'] = Operation 0 change 1 create  
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._Arg0 = ScriptName  
//--IN Event._Arg1 = Operation 0 change 1 create  
//--IN Event._DesObject =Object  
//--IN Event._ThisObject =true self ==false instance  
//--OUT none
```

#### 4.15 *\_OnSyncGroupChange(self, Event )*

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT none
```

```
//--IN Event._DesObject =Object
```

```
//--IN   Event._ThisObject =true self ==false instance  
//--OUT   none
```

#### 4. 16 *\_OnChildSyncGroupChange(self, Event )*

```
//--IN Event['_Arg0'] = ChildObject  
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT   none
```

```
//--IN   Event._Arg0 = ChildObject  
//--IN   Event._DesObject =Object  
//--IN   Event._ThisObject =true self ==false instance  
//--OUT   none
```

#### 4. 17 *\_OnActiveSetChange (self, Event )*

```
//--IN Event['_Arg0'] = SysRootItem  
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT   none
```

```
//--IN   Event._Arg0 = SysRootItem  
//--IN   Event._DesObject =Object  
//--IN   Event._ThisObject =true self ==false instance  
//--OUT   none
```

#### 4. 18 *\_OnLoadMask (self, Event )*

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT   True,AttributeName1,..    attribute name list which should be not changed by load
```

```
//--IN   Event._DesObject =Object  
//--IN   Event._ThisObject =true self ==false instance  
//--OUT   true,AttributeName1,..    attribute name list which should be not changed by load
```

#### 4. 19 *\_OnLoadFinish(self, Event )*

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self ==False instance  
//--OUT   none
```

```
//--IN   Event._DesObject =Object
```



```
//--IN    Event._ThisObject =true self ==false instance  
//--OUT    none
```

#### 4.20 *\_OnTicket(self, Event ) only triggered to self*

```
//--IN Event['_Arg0'] = Ticket  
//--IN Event['_DesObject'] =Object  
//--OUT    none
```

```
//--IN    Event._Arg0 = Ticket  
//--IN    Event._DesObject =Object  
//--OUT    none
```

#### 4.21 *\_OnFrameTicket(self, Event ) only triggered to self*

```
//--IN Event['_Arg0'] = Ticket  
//--IN Event['_Arg1'] = FrameTimer , maintained by server  
//--IN Event['_DesObject'] =Object  
//--OUT    none
```

```
//--IN    Event._Arg0 = Ticket  
//--IN    Event._Arg1 = FrameTimer, maintained by server  
//--IN    Event._DesObject = Object  
//--OUT    none
```

#### 4.22 *\_OnIdle(self, Event ) : only triggered to self*

```
//--IN Event['_Arg0'] = Ticket  
//--IN Event['_DesObject'] =Object  
//--OUT    number== 0 wait next cycle , otherwise continue create the s=event
```

```
//--IN    Event._Arg0 = Ticket  
//--IN    Event._DesObject = Object  
//--OUT    number == 0 wait next cycle , otherwise continue create the s=event
```

The event is dispatched to current active service.

#### 4.23 *\_OnAppActive(self, Event ) only triggered to self*

```
//--IN Event['_DesObject'] =Object  
//--OUT    none
```

```
//--IN    Event._DesObject = Object  
//--OUT    none
```

The event is dispatched to current active service.

#### 4.24 *\_OnAppDeactive(self, Event ) only triggered to self*

```
//--IN Event['_DesObject'] =Object  
//--OUT  none
```

```
//--IN  Event._DesObject = Object  
//--OUT  none
```

The event is dispatched to current active service.

#### 4.25 *\_OnServiceActive(self, Event ) only triggered to self*

```
//--IN Event['_DesObject'] =Object  
//--OUT  none
```

```
//--IN  Event._DesObject = Object  
//--OUT  none
```

The event is dispatched to current active service.

#### 4.26 *\_OnServiceDeactive(self, Event ) only triggered to self*

```
//--IN Event['_DesObject'] =Object  
//--OUT  none
```

```
//--IN  Event._DesObject = Object  
//--OUT  none
```

The event is dispatched to current active service.

#### 4.27 *\_OnRemoteSend( self, Event )*

```
//--IN Event['_DesObject'] =Object  
//--IN Event['_ThisObject'] =True self==False instance  
//--IN Event['_Arg0'] = ParaPkg(message)  
//--OUT  none
```

```
//--IN  Event._DesObject =Object  
//--IN  Event._ThisObject =true self==false instance  
//--IN  Event._Arg0 = ParaPkg(message)  
//--OUT  none
```

## 4.28 \_OnCall ( Event )

```
//--IN Event['_DesObject'] =Object
//--IN Event['_ThisObject'] =True self==False instance
//--IN   Event['_Arg0'] = FunctionPara
//--IN   Event['_Arg1'] = FunctionName
//--IN   Event['_Arg2'] = FunctionID
//--OUT   True,RetVal
```

```
//--IN   Event._DesObject =Object
//--IN   Event._ThisObject =true self ==false instance
//--IN   Event._Arg0 = FunctionPara
//--IN   Event._Arg1 = FunctionName
//--IN   Event._Arg2 = FunctionID
//--OUT   true,RetVal
```

## 5 Interface function

### 5.1 Init CLE

#### 1. Integer = \_InitCore(bool ServerFlag,bool ShowMenuFlag,bool ShowClientWndFlag,bool SRPPrintFlag,DebugInterface,DebugPort,ClientInterface,ClientPort);

```
java:      int      _InitCore(boolean      ServerFlag,boolean      ShowMenuFlag,boolean
ShowOutWndFlag,boolean SRPPrintFlag,String  DebugInterface,int   DebugPortNumber,String
ClientInterface,int ClientPortNumber);
```

ServerFlag True : run as server (default), False : run as client

ShowMenuFlag, whether to show menu

ShowClientWndFlag, Whether to show client window, note: it is reserved, no effect.

SRPPrintFlag, whether to print CLE internal information

DebugInterface, interface for debugserver, may be ""

DebugPort, port number for debugserver.

ClientInterface, interface for client, may be "".

ClientPort, port number for client

The function returns zero for success, -1 for failure, 1 for have been init before.

[Note: parameter ShowMenuFlag, ShowOutWndFlag are reserved. Examples:

```
import libstarpy
libstarpy._InitCore
(ServerFlag,ShowMenuFlag,ShowOutWndFlag,SRPPrintFlag,"",DebugPort,"",ClientPort)
```

```
require "libstarcore"
```

```
libstarcore._InitCore
(ServerFlag,ShowMenuFlag,ShowOutWndFlag,SRPPrintFlag,"",DebugPort,"",ClientPort)
]
```

## 2. SrvGroup = \_InitSimpleEx(ClientPort,WebPort,DependService,...);

```
java:      StarSrvGroupClass      _InitSimpleEx(int      ClientPortNumber,int
WebPortNumber,String...DependService);
```

A simplified init method.

Concurrent connections of Webclient is 100, and max upload size is 2MB.

```
[
import libstarpy
SrvGroup= libstarpy._InitSimpleEx(0,0)
```

```
require "libstarcore"
SrvGroup=libstarcore._InitSimpleEx(0,0)
```

## 3. Service = \_InitSimple(ServiceName,Pass,ClientPort,WebPort,DependService,...); Service= \_InitSimple1(ServiceName,ServiceID,Pass,ClientPort,WebPort,DependService,...);

```
java:      StarServiceClass      _InitSimple(String      ServiceName,String      ServicePass,int
ClientPortNumber,int WebPortNumber,String...DependService);
```

```
java: StarServiceClass _InitSimple1(String ServiceName,String ServiceID,String ServicePass,int
ClientPortNumber,int WebPortNumber,String...DependService);
```

A simplified init method. "Pass" is the password of root user.

Concurrent connections of Webclient is 100, and max upload size is 2MB.

```
[
import libstarpy
Service= libstarpy._InitSimple("test","123",0,0)
```

```
require "libstarcore"
Service=libstarcore._InitSimple("test","123",0,0)
```

## 5.2 Callback and control functions

These functions should be called in thread which inits the CLE. If is called in other threads, apps should acquire lock using function \_SRPLock first, and release lock using function \_SRPUnLock after finished.

### 0. \_ModuleExit () : exit function

```
java: void _ModuleExit();
```

Before exit the program, the function should be called to clear resources occupied by CLE.

Note:

For android, because the jni module libstar\_java.so is not unloaded by jvm.

Please don't call \_ModuleExit, and using the following to check the service has been created

before.

```
StarSrvGroupClass SrvGroup = starcore._GetSrvGroup(0);
StarServiceClass Service = SrvGroup._GetService("test","123");
if( Service == null ){ // the service has not been initialized
}else{
}
```

you can call the following function to exit the process.

```
android.os.Process.killProcess(android.os.Process.myPid());
```

### 1. **\_ModuleClear ()**

```
java: void _ModuleClear();
```

The function deletes all CLE objects. After the function is called, application can re-init service or load new service.

### 2. **\_RegMsgCallBack (CallBack):**

```
java: void _RegMsgCallBack(StarCallBackClass Obj,String CallBackName);
```

Register callback function to handle global event from CLE.

For static script language, such as c#, java, the prototype of the function is **\_RegMsgCallBack** (StarCallBackClass Obj,CallBack function name)

```
def CallBack( ServiceGroupID, uMes, wParam, lParam ) :
```

```
...
return True/False,arg1,...    The first argument is True, means the following args is valid.
```

```
function CallBack( ServiceGroupID, uMes, wParam, lParam )
```

```
...
return true/false, arg11,...    The first argument is true, means the following args is valid.
```

java/c#: should extending StarCallBackClass, defining callback function, and registering with callback function name.

```
Object[] CallBack( (int)ServiceGroupID, (int)uMes, Object, Object )
```

```
...
return new Object[] {true/false, arg11,...}    The first argument is true, means the following args is valid.
```

example:

```
class MyStarCallBackClass extends StarCallBackClass{
    public Object[] CallBack( int ServiceGroupID, int uMes, Object wParam, Object lParam )
    {
        return null;
    }
    MyStarCallBackClass(StarCoreFactory
starcore){super(starcore);starcore._RegMsgCallBack(this,"CallBack");}
}

StarCoreFactory starcore= StarCoreFactory.GetFactory();
MyStarCallBackClass CallBack = new MyStarCallBackClass(starcore);
```

### 3. **\_RegMsgCallBack\_P (CallBack):**

for lua, ruby, python, this function is same as \_\_RegMsgCallBack

```
java: void _RegMsgCallBack_P(StarMsgCallBackInterface CallBackProc);
```

```
public interface StarMsgCallBackInterface{
    public Object Invoke( int ServiceGroupID, int uMes, Object wParam, Object lParam);
}
```

Example:

```
starcore._RegMsgCallBack_P(new StarMsgCallBackInterface(){
    public Object Invoke( int ServiceGroupID, int uMes, Object wParam, Object lParam)
    {
        if( uMes == starcore._Getint("MSG_ONTELNETSTRING_PREEEXECUTE")){
        }
        return null;
    }
});
```

```
c#: void _RegMsgCallBack_P(StarMsgCallBackInterface^ CallBackProc);
```

```
public delegate Object StarMsgCallBackInterface(int ServiceGroupID, int uMes, Object
wParam, Object lParam)
```

Example:

```
starcore._RegMsgCallBack_P(delegate (int ServiceGroupID, int uMes, Object wParam, Object
lParam){
    if( uMes == starcore._Getint("MSG_ONTELNETSTRING_PREEEXECUTE")){
    }
    return null;
});
```

uMes :

MSG\_VSDISPMSG :

MSG\_VSDISPLUAMSG :

MSG\_DISPMSG :

MSG\_DISPLUAMSG: displaying information, wParam is content string, lParam is warning level.  
Return value will be ignored.

MSG\_MESSAGEBOX : displaying messagebox, wParam is the caption, lParam is content  
Return value will be ignored.

MSG\_EXIT : request to exit program, wParam is ErrorInfo, lParam is invalid,  
Return value will be ignored.

MSG\_GETWNDHANDLE : get client window handle, wParam is invalid, lParam is invalid,  
Return value: True, handle(uint)

MSG\_SETWNDSize : set client window size, wParam is width, lParam is height,  
Return value will be ignored.

MSG\_GETWNDSize : get client window size, wParam is invalid, lParam is invalid,  
return value: True, width(uint) ,height(uint)

MSG\_CLEARWND : clear client window, wParam is invalid, lParam is invalid,  
Return value will be ignored.

MSG\_HIDEWND :hide client window,wParam is invalid,lParam is invalid,  
Return value will be ignored.

MSG\_SHOWWND :display client window,wParam is invalid,lParam is invalid,  
Return value will be ignored.

MSG\_SETWNBK:set client window background color,wParam is color,lParam is invalid,  
Return value will be ignored.

MSG\_SETFOCUS: set client window focus, wParam is window handle, lParam 0/1, which 0 represents notification, and 1 means the message should be processed.  
Return value will be ignored.

MSG\_ISAPPACTIVE : whether the program is active or not,wParam is invalid,lParam is invalid,  
return value:True,True/False.

MSG\_SETIDLEACTIVE :wParam is True/False,lParam is invalid.  
Return value will be ignored.

MSG\_SETINFOCOLOR :set text color,wParam is (NormalTextColor, ExplaneColor, ObjectNameColor, AttributeTypeColor, NumberColor, ErrorColor),lParam is invalid,  
For java/c#, wPara is int array.  
Return value will be ignored.

MSG\_SETINFOBK: set text background color,wParam is color,lParam is invalid,  
Return value will be ignored.

MSG\_KILLFOCUS: kill client window focus,wParam is window handle. lParam 0/1 ,0 means notification, and 1 means it should be processed.  
Return value will be ignored.

MSG\_ONBEFORESTOPSERVICE : before stop service, wParam is string of service id,lParam is invalid,  
Return value will be ignored.

MSG\_ONSTOPSERVICE: service is stopped, wParam is string of service id, lParam is invalid,  
Return value will be ignored.

MSG\_ONACTIVESERVICE:service is activated,wParam is string of service id, lParam is invalid,  
Return value will be ignored. [note:valid for client service group]

MSG\_SAVESERVICE: whether to save service,wParam is invalid,lParam is invalid  
return value:True,True/False.[note:valid for server service group]

MSG\_HYPERLINK: active hyper connection. wParam is URL,lParam is int which means creating new window or not,  
Return value will be ignored.

MSG\_APPEVENT: trigger app event, wParam EventID, lParam VS\_CHAR \*EventInfo  
Return value will be ignored.

MSG\_SERVERTERM: server close connection,wParam is invalid,lParam is invalid,  
Return value will be ignored. [note:only valid for client service group]

#### Manager window message:

MSG\_ISMANAGERVISIBLE :whether manage window is visible. wParam is invalid,lParam is invalid,  
return value:True,True/False.

MSG\_HIDEMANAGER :hide manager window,wParam is invalid,lParam is invalid,  
Return value will be ignored.

MSG\_SHOWMANAGER :show manager window,wParam is invalid,lParam is invalid,  
Return value will be ignored.

MSG\_SETMANAGERCAPTION:set manager window caption,wParam is caption,lParam is invalid,  
Return value will be ignored.

MSG\_GETMANAGERSIZE :get manager window size,wParam is invalid,lParam is invalid,  
return value:True,width(uint) ,height(uint)

MSG\_GETMANAGERHANDLE :get manager window handle,wParam is invalid,lParam is invalid,  
return value:True,handle(uint)

MSG\_SHOWMANAGERSTATUSMENU:whether to show manager window menu or status bar,wParam is menu status True/False,lParam is status of status bar, True/False,  
Return value will be ignored.

MSG\_SETMANAGERSTYLE :set manager window parameters,wParam is (SystemMenuFlag, MinimizeFlag, MaximizeFlag, ShowBorderFlag, SizeableFlag),lParam is invalid,

for java/c#,wPara is bool array  
 Return value will be ignored.

MSG\_MOVEMANAGER :move window,wParam is (X,Y,nWidth,nHeight),lParam RepaintFlag,  
 Return value will be ignored.

for java/c#,wPara is int array  
 MSG\_GETMANAGERPOS :ge window position,wParam is invalid,lParam is invalid,  
 return value:true,X,Y,nWidth,nHeight

MSG\_SETMANAGERSTATUS: wParam Status lParam 0/--0 normal, 1 minimize, other maximize  
 Return value will be ignored.

MSG\_REDIRECTTOURLREQUEST:wParam is Url string, lParam is WorkDirectory string.  
 Return: false(allow),true(forbid)

MSG\_REDIRECTTOURLINFO: wParam is Url string, lParam is ParaPkg. Return value will be ignored

MSG\_GETURLREQUEST:wParam is invalid,lParam is invalid. Return value:true,url string

MSG\_SETPROGRAMTYPE:wParam Type[VS\_SERVER\_SERVER/Vs\_SERVER\_USER], lParam 0,  
 return value: false(permit),true(forbid)

MSG\_ISWINDOWLESSSITE: wParam 0 lParam 0. Return: false(yes),true(not)

/\*-----telnet string, add 2.50.0\*/

MSG\_ONTELNETSTRING /\*--wParam \*Buf lParam \*ScriptInterface, return: true(not  
 run),false(run)\*/

/\* if lParam is "CTRL", then wParam is key char of CTRL+A...Z, except H/I/J/M \*/

/\*-----telnet string pre processing before executed, add 2.52.0\*/

MSG\_ONTELNETSTRING\_PREEXECUTE /\*--wParam \*Buf lParam \*ScriptInterface,  
 return: string [for lua,ruby,python]

Return object[] {true/false,string} [for java or c#, the first bool result is ignored]\*/

If return result is null, then the script continue to be executed.

If return result is empty string "",then the script will not be executed.

You can change script string in the callback and return the changed script to cle.

#### 4. **bool = \_SRPDispatch** (bool WaitFalg) : message dispatch function

```
java: boolean _SRPDispatch(boolean WaitFalg);
```

One message will be processed for each call. If return value is True, then there is a message being processed, otherwise the message queue is empty.

#### 5. **\_SRPIdle** () : Notify CLE to create IDLE system event.

```
java: boolean _SRPIdle();
```

#### 6. **\_MsgLoop** ( bool ExitFunc = None / LoopTimes = times) : message loop function.

```
java: boolean _MsgLoop(StarCallBackClass Obj,Object ExitFuncName_LoopTimes );
    public boolean ExitFunc()
    {
        return true/false;
    }
```

For static language, such as c#, java, its prototype is **\_MsgLoop** ( StarCallBackClass Obj,bool ExitFunc = None / LoopTimes = times) :

If the ExitFunc returns False, then the loop continues, or else the loop breaks.

If Looptimes is not 0, the loop is ended after the corresponding times. Each loop will process all the message in the queue.

#### 7. **\_MsgLoop\_P** (CallBack):

for lua, ruby, python, this function is same as \_MsgLoop

```
java: boolean _MsgLoop_P(StarMsgLoopInterface CallBackProc );
```



```

public interface StarMsgLoopInterface{
    public boolean Invoke( );
}
c#: bool _MsgLoop_P(StarMsgLoopInterface CallBackProc );

public delegate bool StarMsgLoopInterface();

```

C++ code:

```

while(1){
    if(PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE)){
        if(!GetMessage(&msg, NULL, 0, 0))
            break;
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }else{
        if( SRPDispatch( VS_FALSE ) == VS_FALSE )
            SRPIde procedure
    }
}

```

```

function ExitFunc()
    return true/false;
end
libstarcore._MsgLoop(ExitFunc)

```

```

def ExitFunc()
    return true/false;
libstarpy._MsgLoop(ExitFunc)

```

```

class MyStarCallBackClass extends StarCallBackClass{
    MyStarCallBackClass(StarCoreFactory starcore){super(starcore);}
    public boolean ExitProc()
    {
        if(StarCore._KeyPress() == 27){
            return true;
        }
        return false;
    }
}

StarCoreFactory starcore= StarCoreFactory.GetFactory();
MyStarCallBackClass CallBack = new MyStarCallBackClass(starcore);

starcore._MsgLoop(CallBack,"ExitProc");

```

```

class MyStarCallBackClass : StarCallBackClass{    c#
    public MyStarCallBackClass(StarCoreFactory starcore):base(starcore){ }
    public Boolean ExitProc()
    {
        if(StarCore._KeyPress() == 27){
            return true;
        }
        return false;
    }
}

StarCoreFactory starcore= StarCoreFactory.GetFactory();
MyStarCallBackClass CallBack = new MyStarCallBackClass(starcore);

starcore._MsgLoop(CallBack,"ExitProc");

```

## 8. bool = \_WinMsgDispatch () : dispatch win32 message

```
java: boolean _WinMsgDispatch();
```

Returns FALSE means WM\_QUIT message is detected.

9. **code = \_KeyPress () :** return key code, for example: ESC=27

```
java: int _KeyPress();
```

10. **\_SRPLock ()/\_SRPUnLock**, lock and unlock

```
java: void _SRPLock();
```

```
java: void _SRPUnLock();
```

11. **\_RegDispatchRequest\_P()**

The CallBack is called in the thread context different from main thread.

It must not call SRPDispatch function to dispatch message.

It should send message to main thread, and return. Then wait main thread to dispatch message

Lua:

```
libstarcore._RegDispatchRequest_P(function() xxxx end)
```

Python:

```
def xxxxx():
```

```
    Xxx
```

```
libstarpy. _RegDispatchRequest_P(xxxxx)
```

**Ruby : this function is not supported. For ruby always run in main thread context.**

Java:

```
starcore._RegDispatchRequest_P(new StarDispatchRequestCallBackInterface(){
    public void Invoke( )
    {
        System.out.println("DISpatch Request from java");
    }
});
```

CS:

```
starcore._RegDispatchRequest_P(=>{
    Console.WriteLine("DISpatch Request from cs");
});
```

### 5.3 Global function

1. **ServiceGroup = \_GetSrvGroup(ServiceGroupID or Service name);**

```
java: StarSrvGroupClass _GetSrvGroup(Object ServiceName_GroupID);
```

Get service group object. For server service group, ServiceGroupID is fixed set to 0, and for client service group, the function must be called after the service group has been created .

2. **ServiceGroup = \_CreateSrvGroup (ServiceGroupID,int Type);**

```
java: StarSrvGroupClass _CreateSrvGroup(int ServiceGroupID,int Type );
```

Create service group and return the service group object. Type should be set to VS\_CLIENT\_USER, which means to create client service group.

If ServiceGroupID==0, the id will be allocated by CLE.

If the service group has existed, None/null will be returned.

**DeleteSrvGroup (ServiceGroupID);**

```
java: void _DeleteSrvGroup(int ServiceGroupID);
```

Delete service group. Group 0 can not be deleted.

**3. SrvGroupInfo ();**

```
java: void _SrvGroupInfo();
```

Print current service information

**4. ServiceGroupID = \_FirstSrvGroup ();**

```
java: int _FirstSrvGroup();
```

Return ID of the first service group. Return 0xFFFFFFFF means no more service group.

**5. ServiceGroupID = \_NextSrvGroup ();**

```
java: int _NextSrvGroup();
```

Return ID of next service group. Return 0xFFFFFFFF means no more service group

*5.3.1 OS type and time*

**1. Type=\_GetOsType();**

```
java: int _GetOsType();
```

System type is 0 Win32, 1 Linux.

**2. \_Time();**

```
java: StarTimeClass _Time();
```

Get current system time.

*5.3.2 registry query function*

**1. Str = \_GetRegStr (SubKey, ValueName, char \*DefaultValue);**

```
java: String _GetRegStr(String SubKey,String ValueName,String DefaultValue);
```

The format of subkey is as "Software\\SRPLab\\SRPServer"

When the SubKey does not exist, DefaultValue will be returned.

**2. Int = \_GetRegInt (SubKey, ValueName, int DefaultValue);**

```
java: int _GetRegInt(String SubKey,String ValueName,int DefaultValue);
```

The format of subkey is as "Software\\SRPLab\\SRPServer"

When the SubKey does not exist, DefaultValue will be returned.

*5.3.3 32bit operation*

**1. int = \_shl32(a,b); a << b**

```
java: int _shl32(int a,int b);
```

**2. int = \_shr32(a,b); a >> b**

```
java: int _shr32(int a,int b);
```

**3. int = \_and32(a,b); a & b**

```
java: int _and32(int a,int b);
```

4. **int = \_or32(a,b);    a | b**

```
java: int _or32(int a,int b);
```

5. **int = \_xor32(a,b);    a ^ b**

```
java: int _xor32(int a,int b);
```

### 5.3.4 file find function

1. **Exist,Handle,FileName,IsDir(true/false) = \_FindFirstFile (FileName);**

```
java: Object[] _FindFirstFile(String FileName);
```

2. **Exist, FileName,IsDir(true/false)= \_FindNextFile (Handle);**

```
java: Object[] _FindNextFile(Object Handle);
```

3. **\_FindClose (Handle)**

```
java: void _FindClose(Object Handle);
```

\_FindClose should be called after \_FindFirstFile function.

### 5.3.5 byte order change

1. **\_htonl(long)**

```
java: int _htonl(int a);
```

2. **\_htons(short)**

```
java: short _htons(short a);
```

3. **\_ntohl(long)**

```
java: int _ntohl(int a);
```

4. **\_ntohs(short)**

```
java: short _ntohs(short a);
```

### 5.3.6 Shell execute command[Windows, do not support C#]

1. **\_ShellExecute (int hWnd, String ShellOperation, String ShellFile, String Parameters, String Directory, int ShowCmd)**

```
java: _ShellExecute(int hWnd, String ShellOperation, String ShellFile, String Parameters, String Directory, int ShowCmd);
```

C# does not support this function.

### 5.3.7 virtual key status[Windows, do not support C#]

1. **bool = \_GetKeyState (Key) :**

```
java: boolean _GetKeyState(int Key);
```

If the key is pressed, then True will be returned. Key defines as follows:

```
VS_LBUTTON    0x01
VS_RBUTTON    0x02
VS_MBUTTON    0x04
```

```
VS_ESCAPE     0x1B
VS_BACKSPACE  0x08
VS_TAB        0x09
```

VS_ENTER	0x0D
VS_SPACE	0x20
VS_SHIFT	0x10
VS_CTRL	0x11
VS_ALT	0x12
VS_LWIN	0x5B
VS_RWIN	0x5C
VS_APPS	0x5D
VS_PAUSE	0x13
VS_CAPSLOCK	0x14
VS_NUMLOCK	0x90
VS_SCROLLLOCK	0x91
VS_PGUP	0x21
VS_PGDN	0x22
VS_HOME	0x24
VS_END	0x23
VS_INSERT	0x2D
VS_DELETE	0x2E
VS_LEFT	0x25
VS_UP	0x26
VS_RIGHT	0x27
VS_DOWN	0x28
VS_0	0x30
VS_1	0x31
VS_2	0x32
VS_3	0x33
VS_4	0x34
VS_5	0x35
VS_6	0x36
VS_7	0x37
VS_8	0x38
VS_9	0x39
VS_A	0x41
VS_B	0x42
VS_C	0x43
VS_D	0x44
VS_E	0x45
VS_F	0x46
VS_G	0x47
VS_H	0x48
VS_I	0x49
VS_J	0x4A
VS_K	0x4B
VS_L	0x4C
VS_M	0x4D
VS_N	0x4E
VS_O	0x4F
VS_P	0x50
VS_Q	0x51
VS_R	0x52
VS_S	0x53
VS_T	0x54
VS_U	0x55
VS_V	0x56
VS_W	0x57
VS_X	0x58
VS_Y	0x59
VS_Z	0x5A
VS_GRAVE	0xC0
VS_MINUS	0xBD
VS_EQUALS	0xBB

VS_BACKSLASH	0xDC
VS_LBRACKET	0xDB
VS_RBRACKET	0xDD
VS_SEMICOLON	0xBA
VS_APOSTROPHE	0xDE
VS_COMMA	0xBC
VS_PERIOD	0xBE
VS_SLASH	0xBF

VS_NUMPAD0	0x60
VS_NUMPAD1	0x61
VS_NUMPAD2	0x62
VS_NUMPAD3	0x63
VS_NUMPAD4	0x64
VS_NUMPAD5	0x65
VS_NUMPAD6	0x66
VS_NUMPAD7	0x67
VS_NUMPAD8	0x68
VS_NUMPAD9	0x69

VS_MULTIPLY	0x6A
VS_DIVIDE	0x6F
VS_ADD	0x6B
VS_SUBTRACT	0x6D
VS_DECIMAL	0x6E

VS_F1	0x70
VS_F2	0x71
VS_F3	0x72
VS_F4	0x73
VS_F5	0x74
VS_F6	0x75
VS_F7	0x76
VS_F8	0x77
VS_F9	0x78
VS_F10	0x79
VS_F11	0x7A
VS_F12	0x7B

### 5.3.8 *intger compare*

#### 1. **int = \_IntComp(a,b); -1: a<b 0 a==b 1 a>b**

```
java: int _IntComp(int a,int b);
```

#### 1. **int = \_UIntComp(a,b); -1: a<b 0 a==b 1 a>b**

```
java: int _UIntComp(int a,int b);
```

### 5.3.9 *get UUID*

#### 1. **Uuid=\_UuidCreate()**

```
java: String _UuidCreate();
```

### 5.3.10 *get current Url*

#### 1. **Url=\_GetUrl ()**

```
java: String _GetUrl();
```

Example: for <http://127.0.0.1/test/test.html>, the return value is `http:\\127.0.0.1\\test`

#### 2. **Url=\_GetRootUrl ()**

```
java: String _GetRootUrl();
```

Example: for <http://127.0.0.1/test/test.html>, the return value is `http:\\127.0.0.1`

### 5.3.11 set program run type [valid at server]

If type is different from current, current service will be unloaded first.

#### 1. **\_SetProgramType (Type) [Reserved]**

```
java: void _SetProgramType(int Type);
```

Type takes value from: VS\_SERVER\_SERVER/VS\_SERVER\_USER

#### 2. **Type= \_GetProgramType()**

```
java: int _GetProgramType();
```

### 5.3.12 find char position

#### 1. **Int=\_strchr (string,char)**

```
java: int _strchr(String Str,String Ch);
```

#### 2. **Int=\_strrchr(string,char)**

```
java: int _strrchr(String Str,String Ch);
```

For lua, Return index starts from 1, and -1 means not exist.

For other script, Return index starts from 0, and -1 means not exist.

Do not distinguish '/' and '\\'

### 5.3.13 ID to MD5 string

#### 1. **MD5Str=\_IDToMD5 (IDStr)**

```
java: String _IDToMD5(String IDStr);
```

### 5.3.14 get system path

#### 1. **Path=\_GetSysPath()**

```
java: String _GetSysPath();
```

On Windows, the return is X:\windows\system32

On Linux, the return is /usr/lib

### 5.3.15 set log file

#### 1. **\_SetLogFile (FileName,bool LogAll)**

```
java: void _SetLogFile(String FileName,boolean LogAll);
```

If FileName is "", current log will be canceled.

LogAll:true, log all print information, otherwise only log warning and error information.

### 5.3.16 authorize

#### 1. **string=\_GetSystemRegCode()**

```
java: String _GetSystemRegCode();
```

Get local serial number

**Note: From Version 2.0, this function should not be used.**

#### 2. **bool=\_SetRegisterCode(CodeStr,Single)**

```
java: boolean _SetRegisterCode(String CodeString,boolean Single);
```

Note: From Version 2.0, Single will be ignored.

The function should be called after starcore init and before creating any service.

require "libstarcore"

libstarcore.\_InitCore(true,true,false,true,"",0,"",0); // should init starcore before call

\_PreAuthorize

libstarcore.\_SetRegisterCode(XXXX,false);

**3. bool= \_IsRegistered ()**

java: boolean \_IsRegistered();

Whether CLE is registered.

**4. bool \_PreAuthorize(string ServiceName,string ServiceID,string CodeStr,bool Single)**

java: boolean \_PreAuthorize(String ServiceName,String ServiceID,String CodeStr,boolean Single);

Note: From Version 2.0, ServiceName,ServiceID,and Single will be ignored.

This function is depreciated

*5.3.17 set locale of cle*

**1. void \_SetLocale(String Lang);**

void \_SetLocale(String Lang);

**2. string \_GetLocale();**

String \_GetLocale();

for windows:

Lang should be "utf8" or "ansi",or ".XXX", XXX is number of codepage, such as ".950"

*5.3.18 find window handle[win32]*

**3. long hWnd= \_FindWindow(String Caption);**

long \_FindWindow(String Caption);

Find window handle.

*5.3.19 get script interface index*

**4. int \_GetScriptIndex(String Interface);**

int \_GetScriptIndex(String Interface);

*5.3.20 get script interface version*

**5. int[] \_Version();**

int[] \_Version();

Return interface share library version of caller script;

*5.3.21 get or set env*

**1. boolean \_SetEnv(String Name,String Value)**

Set enviornment value for current process



## 2. String \_GetEnv(String Name)

Get environment value from current process

### 5.3.22 set script interface module

#### 1. Boolean = \_SetScript(String ScriptInterface,String Module, String Para)

Module can be input ""

For java:

Para might be used to set jvm share library and class path, "jvm=XXXX", "java.class.path=XXX", "java.library.path=XXX", or both, separated by ",". For example:

```
_SetScript("java","", "jvm=C:\\Program Files\\Java\\jdk1.6.0_21\\jre\\bin\\server\\jvm.dll,
java.class.path=.;c:\\srplab\\libs\\starcore.jar")
```

**if java.library.path is set, star\_java.dll/libstar\_java.so should be in the path.**

for python:

```
-S : (Py_NoSiteFlag)
-d : (Py_DebugFlag)
-s : (Py_NoUserSiteDirectory)
-v : (Py_VerboseFlag)
-i : (Py_InspectFlag)
-B : (Py_DontWriteBytecodeFlag)
```

For example: "-S -v"

for ruby:

```
-m/-M: ruby share library name with full path
-v/-V : version of ruby, default is "1.9.3". Note: this parameter is only valid on windows desktop
```

For example

```
_SetScript("ruby","D:\\Work\\starcore\\core\\starcore.ruby\\libruby193\\libstar_ruby.so", "-v
1.9.3")
```

for other languages:

Para will be ignored

### 5.3.23 Detach Thread

#### 1. public void \_DetachCurrentThread();

### 5.3.24 Set Operation Path

#### 1. **public String \_SetCoreOperationPath(String Path);**

This function returns old operation path.

### 5.3.25 String Code Conversion

#### 1. **boolean \_iconv(String FromLocale,String ToLocale,StarBinBufClass InBuf,StarBinBufClass OutBuf)**

### 5.3.26 Get Core Share Library Handle

#### 1. **long \_CoreHandle()**

If the libstarcore.dll/so is loaded from script, apps can use this function to get the handle of the share library.

Note: This function always return 0 for lua, and returns 0 on windows phone 8.

### 5.3.27 Inject Script Class

#### 1. **void \_InjectClass(String ClassName,Class cl)**

**This function is valid for java and csharp. If java or csharp class is loaded by a class loader, this function can be used to inject the class to cle.**

## 5.4 Service group attribute

#### 1. **SysObject=ServiceGroup.\_SysObject**

get system defined object

#### 2. **Int=ServiceGroup.\_EnvStartType**

service start type

Return value lists below :

0 start from directory contains multiple files.

1 start from single file package.

#### 3. **ParaPkg=ServiceGroup.\_EnvPara**

current service parameter

#### 4. **ParaPkg =ServiceGroup.\_EnvInputPara**

current service input parameter

#### 5. **Url =ServiceGroup.\_EnvParentUrl**

parent service Url

#### 6. **SysDocClass=ServiceGroup.\_SysDocClass**

get system doc class

#### 7. **ServiceGroup.\_ProgramType**

refer to macro definitions of “program run type”.

## 8. ServiceGroup. \_IsRootService

If the service is dynamic service which is loaded by import function, the return value is False, or else is true.

## 5.5 Service group function

### 5.5.1 public functions

#### 9. Service= \_GetService( username, userpassword ) \_GetServiceEx( ServiceName,username, userpassword );

java: StarServiceClass \_GetService( String username, String userpassword )

java: StarServiceClass \_GetServiceEx( String ServiceName,String username, String userpassword )

Get current service object, the input parameter is user name and password.

User name and password will be ignored at client service group.

#### 10. \_MessageBox(Caption,Info) :

java: void \_MessageBox(String Caption,String Info)

Display message box, it is processed by host application.

#### 11. \_Print (...) /\_PrintError(AlarmLevel,string):

java: void \_Print(String Args)

java: void \_PrintError(int AlarmLevel,String Args)

For static language, only support \_Print(String)/\_PrintError(AlarmLevel,string).

#### 12. \_ID();

java: int \_ID()

Returns service group id

#### 13. \_IsServer();

java: boolean \_IsServer()

Returns True, means is server service group.

#### 14. \_IsClient();

java: boolean \_IsClient()

Returns True, means is client service group.

#### 15. \_IsDebug();

java: boolean \_IsDebug()

Returns True, means is debug service group.

#### 16. \_IsServerClient();

java: boolean \_IsServerClient()

Returns True, means is server client service group.

#### 17. \_IsObject(Object);

java: boolean \_IsObject(Object Args)

Returns True or False

#### 18. \_IsParaPkg (ParaPkg);

java: boolean \_IsParaPkg(Object Args)

Returns True or False

#### 19. \_IsQueryRecord (QueryRecord);

java: boolean \_IsQueryRecord(Object Args)

### Returns True or False

20. **\_IsBinBuf (BinBuf);**

```
java: boolean _IsBinBuf(Object Args)
```

## Returns True or False

21. \_IsSXml(SXml);

```
java: boolean _IsSXml(Object Args)
```

### Returns True or False

**22. \_IsFunctionPara(FunctionPara);**

```
java: boolean _IsFunctionPara(Object Args)
```

### Returns True or False

**23. \_IsCommInterface(CommInterface);**

```
java: boolean _IsCommInterface(Object Args)
```

## Returns True or False

24. `_TickCount ();`

```
java: int _TickCount()
```

If high precision is supported, then it will return high precision timer.

**25.        \_MD5(Str);**

```
java: String _MD5(String Args)
```

Returns MD5 of the string.

```
26.      Bool      =      _SetDataServerAddr      (DirectConnect,DataServerInterface,
      DataServerName, DataServerPort, LocalDataServerInterface,LocalDataServerPort);
```

```
java: boolean _SetDataServerAddr(boolean DirectConnect,String DataServerInterface, String
DataServerName,      int      DataServerPort,      String      LocalDataServerInterface,int
LocalDataServerPort)
```

Valid at server service group;

DirectConnect: ==True, when DataServerPort is valid, client will connect to it directly, otherwise, will be relayed by server.

DataSourceInterface: DataSource interface, may be “““

DataSetName: DataSet Url.

DataServerPort: DataServer port number

LocalDataServerInterface: Local DataServer interface, may be'''

LocalDataServerPort: Local DataServer port, 0 means invalid.

```
27.      _SetServerPara      (MaxClientNumber,
      MaxDataServerConnectionNumber,DataServerOverTime);
```

```
java: void _SetServerPara(int MaxClientNumber,int MaxDataServerConnectionNumber,int
DataServerOverTime)
```

Set server parameter

28. **\_QueryStatistic(ClientID);**

```
java: Object[] _QuyeryStatistic(int ClientID)
```

Get statistic info of CLE when ClientID is 0, or else returns statistic of special client. Return value is

```
int ClientConnectionNumber;
```

```
int    DebugConnectionNumber;
int    ServerConnectionNumber;
int    DataConnectionNumber;
int    ReceiveMsgItemNumber;
int    ReceiveMsgItemBytes;
int    SendMsgItemNumber;
int    SendMsgItemBytes;
int    SysSendQueueOccupyRate;
int    ObjSendQueueOccupyRate;
```

```
int    PeerDelayTicket;
```

**29. int = \_Hash(String);**

```
java: int _Hash(String Args)
```

Get string hash value.

**30. \_GetModulePath ();**

```
java: String _GetModulePath()
```

Get module path.

**31. \_GetServicePath ();**

```
java: String _GetServicePath()
```

Get service default path

**32. \_SetServicePath (Path);**

```
java: void _SetServicePath(String Args)
```

Set service default path

**33. \_GetCurrentPath ();**

```
java: String _GetCurrentPath()
```

Get current path.

**34. \_SetCurrentPath (Path);**

```
java: void _SetCurrentPath(String Args)
```

Set current path.

**35. Bool= \_ServicePathIsSet();**

```
java: boolean _ServicePathIsSet()
```

Service default path is set or not.

**36. \_GetSRPTempPath ();**

```
java: String _GetSRPTempPath()
```

Get temporary path

**37. \_GetSRPConfigPath ();**

```
java: String _GetSRPConfigPath()
```

Get path of config file.

### 5.5.2 service sync function(valid at client)

**1. Bool = \_IsInSync ();**

```
java: boolean _IsInSync()
```

Returns True means the service is in sync procedure.

**2. Bool = \_IsServiceSync ();**

```
java: boolean _IsServiceSync()
```

Returns True means the service is in sync status.

**3. Bool = \_WaitServiceSync(WaitTimeMs);**

```
java: boolean _WaitServiceSync(int WaitTimeMs)
```

Wait service becomes synchronous. If success, return value is True . WaitTimeMs is the maximum time (ms) to wait. Zero means wait forever.

*5.5.3 service management function[called at server]*

**1. Bool = ImportServiceEx (ServiceID,LoadRunModule=True);**

```
java: boolean _ImportServiceEx(String ServiceID,boolean LoadRunModule)
```

Import a service,ServiceID is UUID of the service,for example

:"66efc37a-a16c-442b-bc2a-3df5b03b3294". Server should not load or create any service before the function is called. ServiceID should be registered at registry under key SOFTWARE\SRPLab\SRPServer.

**2. Bool = \_ImportServiceWithPath (ServicePath,ServiceName,LoadRunModule=True);**

```
java: boolean _ImportServiceWithPath(String ServicePath,String ServiceName,boolean LoadRunModule)
```

Import a service, input parameters are service path and service name. Server should not load or create any service before the function is called. Under service path, directory which name is same with the service name must exist.

ServicePath is local path or network path, such as:<http://www.XXX.com>. If it equals to "http://srplab", then service is located at default srplab website.

If service name contains ".", then it is dynamic service, the function is same as ImportDynaService.

If service name starts with char "@", then the service file locates at local disk.

**3. Bool = \_ImportService (ServiceName,LoadRunModule=True);**

```
java: boolean _ImportService(String ServiceName,boolean LoadRunModule)
```

Import a service, input parameter is service name. Server should not load or create any service before the function is called. ServiceName must be registered at registry under key SOFTWARE\SRPLab\SRPServer.

If service name contains ".", then it is dynamic service, the function is same as ImportDynaService.

If service name starts with char "@", then the service file locates at local disk.

**4. servicename = \_ImportDynaService (Url);**

```
java: String _ImportDynaService(String Url)
```

Import dynamic service, which may be local file or network file. Return value is service name.

If Url starts with char "@", then the service file locates at local disk.

**5. Bool = \_ImportServiceFromXmlBuf (Buf, LoadRunModule=True);**

```
java: boolean _ImportServiceFromXmlBuf(String Buf,boolean LoadRunModule)
```

Import service from XML buffer. Server should not load or create any service before the function is called.

**6. Service= \_CreateService/\_CreateServiceEx (ServicePath,ServiceName, RootPass, FrameInterval, NetPkgSize, UploadPkgSize, DownloadPkgSize , DataUpPkgSize, DataDownPkgSize ,ServiceID=“”);**

```
java: StarServiceClass _CreateService(String ServicePath,String ServiceName,String RootPass,int FrameInterval,int NetPkgSize,int UploadPkgSize,int DownloadPkgSize,int DataUpPkgSize,int DataDownPkgSize,String ServiceID)
```

```
java: StarServiceClass _CreateServiceEx(String ServicePath,String ServiceName,String RootPass,int FrameInterval,int NetPkgSize,int UploadPkgSize,int DownloadPkgSize,int DataUpPkgSize,int DataDownPkgSize,String ServiceID)
```

Server should not load or create any service before the function is called. RootPass is password of root user. FrameInterval is interval(10ms) between frame of the service. NetPkgSize is size of network package, which default is 10240(bytes); UploadPkgSize is size of data uploaded per frame by client, unit is byte and default is 2048;DownloadPkgSize is size of data downloaded per frame by client, unit is byte, default is 2048. Range of FrameIntervale is [2,100], NetPkgSize is [1024,100\*1024]; UploadPkgSize/ DataUpPkgSize is [1024,100\*1024]; DownloadPkgSize/DataDownPkgSize is [1024,100\*1024].

DataUpPkgSize, DataDownPkgSize is valid only for independent data port.

ServicePath is path of new service, may be “”. In this case, default path is used.

For example, suppose the new created service will be saved at directory:

C:AAA\BBB\service name\service files, then ServicePath should be set to “C:AAA\BBB”

CreateServiceEx: if BIN type files exist in the service path, then they are not deleted. Service should use ClearStatic function to clear static data.

CreateService: if BIN type files exist in the service path, then they will be deleted.

**7. Service = \_LoadServiceEx (ServiceID,UserName,UserPass, LoadRunModule=True);**

```
java: StarServiceClass _LoadServiceEx(String ServiceID,String UserName,String UserPass,boolean LoadRunModule)
```

Load a service. Server should not load or create any service before the function is called. ServiceID should be registered in registry under key SOFTWARE\SRPLab\SRPServer.

**8. Service = \_LoadServiceWithPath (ServicePath,ServiceName,UserName,UserPass,LoadRunModule=True);**

```
java: StarServiceClass _LoadServiceWithPath(String ServicePath,String ServiceName,String UserName,String UserPass,boolean LoadRunModule)
```

Server should not load or create any service before the function is called.

**9. Service = \_LoadService (ServiceName,UserName,UserPass,LoadRunModule=True);**

```
java: StarServiceClass _LoadService(String ServiceName,String UserName,String UserPass,boolean LoadRunModule)
```

Server should not load or create any service before the function is called.

**10. \_ClearService ();**

```
void _ClearService()
```

For server service group, service will be unloaded.

For client service group, connection will be closed.

#### 11. **\_ClearServiceEx();**

java: void \_ClearServiceEx()

For server service group, service will be unloaded.

For client service group, connection will be closed.

The function will be processed for all service groups.

#### 12. **Bool = \_ExportServiceHeader(ServiceName,Path="");**

java: boolean \_ExportServiceHeader(String ServiceName,String Path)

Export service header files for C++. Path may be "", in this case, output will be written to default service path.

#### 13. **Bool = \_XmlToService (SXml,DataPath,SegmentName,PrintFunc=None);**

java: boolean \_XmlToService(StarSXmlClass SXml,String DataPath,String SegmentName,String PrintFuncName)

Another function for creating service, which uses xml file as input. Its syntax is described in other documents.

#### 14. **Bool = \_XmlToService\_P (SXml,DataPath,SegmentName,CallBackProc);**

for lua, ruby, python, this function is same as \_XmlToService

java: boolean \_XmlToService\_P(StarSXmlClass SXml,String DataPath,String SegmentName,StarSrvGroupInfoInterface CallBackProc)

```
public interface StarSrvGroupInfoInterface{
    public void Invoke(Object SrvGroup,String Info);
}
```

c#: bool \_XmlToService\_P(StarSXmlClass SXml,String DataPath,String SegmentName,StarSrvGroupInfoInterface CallBackProc);

public delegate void StarSrvGroupInfoInterface(Object SrvGroup,String Info);

#### 15. **Bool = \_XmlToServiceEx (FileName,PrintFunc=None);**

java: boolean \_XmlToServiceEx(String FileName,String PrintFuncName)

Another function for creating service, which uses xml file as input. Its syntax is described in other documents.

def InfoFunc(Info) :

print( Info)

java/c#:

public void PrintFuncName(String Info)

```
{
}
```

#### 16. **Bool = \_XmlToServiceEx\_P(FileName,CallBackProc);**

for lua, ruby, python, this function is same as \_XmlToServiceEx

java: boolean \_XmlToServiceEx\_P(String FileName,StarSrvGroupInfoInterface CallBackProc)



```

public interface StarSrvGroupInfoInterface{
    public void Invoke(Object SrvGroup,String Info);
}

c#:    bool    _XmlToServiceEx_P(String    FileName, StarSrvGroupInfoInterface
CallBackProc);

public delegate void StarSrvGroupInfoInterface(Object SrvGroup, String Info);

```

#### 5.5.4 client connect to server

1. **\_Connect (ServerInterface,ServerName,PortNumber,RetrySecond, Username, UserPassword, ParaPkg=None, ServiceGroup\_ClientConnectProc=None);**

```

java:  int  _Connect(String  ServerInterface,String  ServerName,int  PortNumber,int
RetrySecond,String  Username,String  UserPassword,StarParaPkgClass  ParaPkg,String
ServiceGroup_ClientConnectProc)

```

2. **\_Connect\_P (ServerInterface,ServerName,PortNumber,RetrySecond, Username, UserPassword, ParaPkg=None, StarSrvGroupConnectInterface);**

for lua, ruby, python, this function is same as \_\_Connect

```

java:  int  _Connect_P(String  ServerInterface,String  ServerName,int  PortNumber,int
RetrySecond,String          Username,String          UserPassword,StarParaPkgClass
ParaPkg,StarSrvGroupConnectInterface CallBackProc)

```

```

public interface StarSrvGroupConnectInterface{
    public void Invoke(Object SrvGroup,int  uMes,int  ConnectionID,int
LinkInterfaceStatus,String ServerName,int PortNumber);
}

```

```

c#:  int  _Connect_P(String  ServerInterface, String  ServerName, int  PortNumber, int
RetrySecond, String          Username, String          UserPassword, StarParaPkgClass
ParaPkg, StarSrvGroupConnectInterface CallBackProc);

```

```

public delegate void StarSrvGroupConnectInterface(Object SrvGroup, int  uMes, int
ConnectionID, int  LinkInterfaceStatus, String ServerName, int  PortNumber);

```

3. **ServiceGroup.\_ConnectEx (ServiceName,RetrySecond, Username,UserPassword, ParaPkg=None, ServiceGroup\_ClientConnectProc=None);**

```

java:  int  _ConnectEx(String  ServiceName,int  RetrySecond,String  Username,String
UserPassword,StarParaPkgClass ParaPkg,String ServiceGroup_ClientConnectProc)

```

4. **ServiceGroup.\_ConnectEx\_P (ServiceName,RetrySecond, Username,UserPassword, ParaPkg, StarSrvGroupConnectInterface);**

for lua, ruby, python, this function is same as \_\_ConnectEx

```

java:  int  _Connect_P(String  ServiceName,int  RetrySecond,String  Username,String
UserPassword,StarParaPkgClass ParaPkg,StarSrvGroupConnectInterface CallBackProc)

```

```
c#: int _Connect_P(String ServiceName,int RetrySecond,String UserName,String
UserPassword,StarParaPkgClass ParaPkg,StarSrvGroupConnectInterface
CallBackProc);
```

Valid at client service group. Client connects to server. Return value is ConnectionID, and zero means failure. Prototype of the function is:

```
def ServiceGroup_ClientConnectProc(self,uMes, ConnectionID,
LinkInterfaceStatus,ServerName, PortNumber ) :
```

```
function ServiceGroup:ClientConnectProc(uMes, ConnectionID,
LinkInterfaceStatus,ServerName, PortNumber )
```

```
JAVA/c#:
public void ClientConnectProc(int uMes,int ConnectionID,int
LinkInterfaceStatus,String ServerName,int PortNumber)
{
}
```

uMes:

- 0 //---after the event ,connection has been setup successfully, and will be followed by service initialization message.
- 1 //---if ConnectionID is zero, indicates the connection is released and no callback will be created latter. Otherwise, CLE will retry to connect.
- 2 //---service initialization is failed at client side.
- 3 //---service initialization is successful at client side.
- 4 //---service finish sync at client side.
- 5 //---connection is closed.

Interface is string, which format is :

“Host=interface address;if=share library file name(include extension);site=share library download address, ftp/http address ;para=parameter string”

for example:" host=192.168.0.1;if=SRPTcpLinkInterface1.dll;site=<ftp://127.0.0.1>"

1. **ConnectionID = \_SConnect (ServerInterface,ServerName,PortNumber, UserName, UserPassword,ParaPkg = nil/None);**

```
java: int _SConnect(String ServerInterface,String ServerName,int PortNumber,String
UserName,String UserPassword,StarParaPkgClass ParaPkg)
```

2. **ConnectionID =\_SConnectEx(ServiceName,UserName, UserPassword, ParaPkg = nil/None);**

```
java: int _SConnectEx(String ServiceName,String UserName,String
UserPassword,StarParaPkgClass ParaPkg)
```

Setup connection to server, wait, and returns ID of the connection after service is loaded.

Interface is string, which format is :

“Host=interface address;if=share lib file name(include extension);site=share lib download address, ftp/http address ;para=parameter string”

for example:" host=192.168.0.1;if=SRPTcpLinkInterface1.dll;site=<ftp://127.0.0.1>"

### 3. **Service =\_Connect2 (ServerInterface,ServerName,PortNumber, UserName, UserPassword,SysRootItemName,ParaPkg = None);**

```
java: StarServiceClass _Connect2(String ServerInterface,String ServerName,int
PortNumber,String UserName,String UserPassword,String
SysRootItemName,StarParaPkgClass ParaPkg)
```

### 4. **Service =\_ConnectEx2(ServiceName,UserName, UserPassword, SysRootItemName,ParaPkg = None);**

```
java: StarServiceClass _Connect2Ex(String ServiceName,String UserName,String
UserPassword,String SysRootItemName,StarParaPkgClass ParaPkg)
```

Setup connection to server, and wait service becomes sync. The return value is service object.

### 5. **\_Disconnect ();**

```
java: void _Disconnect()
```

Close connection.

### 6. **Bool =\_IsConnect ();**

```
java: boolean _IsConnect()
```

Valid at client service group, which is used to determine whether the connection is established.

## 5.5.5 *trigger hyper connection*

### 1. **\_HyperLink (URL,bool = CreateNewWindow);**

```
java: void _HyperLink(String URLArg,boolean CreateNewWindow)
```

Valid at client and server, trigger a new hyper connection. This function will send a callback message to host application.

## 5.5.6 *trigger app event*

### 1. **\_AppEvent(VS\_ULONG EventID,VS\_CHAR \*EventInfo);**

```
java: void _AppEvent(int EventID,String EventInfo)
```

Valid at client and server, create callback message:MSG\_APPEVENT.

## 5.5.7 *QueryRecord object*

### 1. **\_NewQueryRecord ();**

```
java: StarQueryRecordClass _NewQueryRecord()
```

## 5.5.8 *parapkg object*

**1. \_NewParaPkg (Arg1,Arg2,...);**

```
java: StarParaPkgClass _NewParaPkg(Object...Args)
```

**5.5.9 binbuf object****1. \_NewBinBuf();**

```
java: StarBinBufClass _NewBinBuf()
```

**5.5.10 SXML object****1. XML = \_NewSXml()**

```
java: StarSXmlClass _NewSXml()
```

**5.5.11 function para object****1. FuncPara = \_NewFunctionPara()**

```
java: StarFunctionParaClass _NewFunctionPara()
```

**5.5.12 CommInterface object****1. CommInterface = \_NewCommInterface()**

```
java: StarCommInterfaceClass _NewCommInterface()
```

**5.5.13 execute script segments or files**

**Note: The execution of script may be not in the same thread with the caller**

**1. bool = \_RunScript(ScriptInterface,ScriptBuf,ModuleName);**

```
java: boolean _RunScript(String ScriptInterface,String ScriptBuf,String ModuleName)
```

Static script language including java/c# does not support this function.

ModuleName is name of the module, which may be NULL. This Parameter is only valid for lua and python.

**2. bool = \_RunScriptEx(ScriptInterface,BinBuf,ModuleName);**

```
java: boolean _RunScriptEx(String ScriptInterface,StarBinBufClass BinBuf,String ModuleName)
```

BinBuf is created by \_NewBinBuf

Static script language including java/c# does not support this function.

ModuleName is name of the module, which may be NULL. This Parameter is only valid for lua and python.

**3. bool = \_DoFile(ScriptInterface,FileName)/ \_DoFileEx(ScriptInterface,FileName,ModuleName);**

```
java: boolean _DoFile(String ScriptInterface,String FileName)
```

```
java: boolean _DoFileEx(String ScriptInterface,String FileName,String ModuleName)
```

If ScriptInterface is "", default is lua. It can take value from lua,python,ruby,java,c#, or others registered script languages.

ModuleName is name of the module, may be "" or NULL. This Parameter is only valid for lua

and python and java and csharp. For java and csharp, ModuleName is the init class name. **For example, the ModuleName is “com.srplab.www.test”.**

ModuleName should not set to “cmd”, case insensitive.

#### 5.5.14 others function

How to process such functions depends on environment. The corresponding command is send to host program by CLE.

##### 1. **\_IsDefaultServer ();**

```
java: boolean _IsDefaultServer()
```

Whether the server is default server, returns bool.

##### 2. **\_IsWindowVisible ();**

```
java: boolean _IsWindowVisible()
```

Whether manager window is visible, returns bool.

##### 3. **\_HideWindow ();**

```
java: void _HideWindow()
```

Hide manager window

##### 4. **\_ShowWindow ();**

```
java: void _ShowWindow()
```

Show manager window.

##### 5. **\_SetCaption(Caption);**

```
java: void _SetCaption(String Caption)
```

Set manager window caption, which is a string.

When create and load a service, the caption of manager window is set automatically. Therefore, the function should be called after service is created or loaded.

##### 6. **\_ExitVSSystem (ErrorInfo);**

```
java: void _ExitVSSystem(String ErrorInfo)
```

exit program.

##### 7. **\_StartVSService (ServiceID);The function will be provided in the future.**

```
java: void _StartVSService(String ServiceID)
```

Start a service

ServiceID is the service ID to be started.

##### 8. **\_IsAppActive();**

```
java: boolean _IsAppActive()
```

Whether application is active, returns bool

##### 9. **\_SetIdleActive(True/False);**

```
java: void _SetIdleActive(boolean Args)
```

##### 10. **\_GetVersion ();**

```
java: Object[] _GetVersion()
```

Get cle version, return version table.

##### 11. **\_GetVersionInfo ();**

```
java: String _GetVersionInfo()
```

Get cle version information, returns string.

12. **\_SetColor(Text,Explane,ObjName,AttrType,Number,Error);**  
java: void \_SetColor(int Text,int Explane,int ObjName,int AttrType,int NumberColor,int ErrorColor)  
set text color
13. **\_SetBkColor(BkColor);**  
java: void \_SetBkColor(int BkColor)  
set text background color.
14. **\_SetClientBkColor (BkColor);**  
java: void \_SetClientBkColor(int BkColor)  
set client window background color
15. **\_ShowStatusMenu (showmenuflag,showstatusflag);**  
java: void \_ShowStatusMenu(boolean showmenuflag,boolean showstatusflag)  
Whether to show menu or status bar, input is bool.
16. **\_ClearClientWnd ();**  
java: void \_ClearClientWnd()  
clear client window
17. **\_HideClientWnd ();**  
java: void \_HideClientWnd()  
Hide client window.
18. **\_ShowClientWnd ();**  
java: void \_ShowClientWnd()  
Show client window.
19. **\_SetClientSize (Width,Height);**  
void \_SetClientSize(int Width,int Height)  
Set client window width and height.
20. **Width,Height =\_GetClientSize ();**  
java: Object[] \_GetClientSize()  
Get client window width and height.
21. **KernelAllocSize , DataAllocSize , AppAllocSize , ScriptMemoryUsed =\_MemorySize ();**  
java: Object[] \_MemorySize()  
KernelAllocSize: memory of core.  
DataAllocSize: memory of static data  
AppAllocSize: memory of application.  
ScriptMemoryUsed: memory of lua.
22. **\_SetWindowStyle (bool :SystemMenuFlag, MinimizeFlag, MaximizeFlag, ShowBorderFlag, SizeableFlag);**  
java: void \_SetWindowStyle(boolean SystemMenuFlag,boolean MinimizeFlag,boolean MaximizeFlag,boolean ShowBorderFlag,boolean SizeableFlag)  
Set window parameter.
23. **\_MoveWindow (X, Y, nWidth, nHeight, bool RepaintFlag);**  
java: void \_MoveWindow(int X,int Y,int nWidth,int nHeight,boolean RepaintFlag)  
Move window, If nWidth,nHeight equal to 0, then current window size remains unchanged.
24. **X, Y, nWidth, nHeight =\_GetWindowPos ();**

```
java: Object[] _GetWindowPos()
```

get client position.

## 25. **\_SetWindowStatus (int Status);**

```
java: void _SetWindowStatus(int Status)
```

Status //--0 normal, 1 minimize, other maximize

### 5.5.15 RawSocket function

If RawSocket fails to setup connection, it will not retry; Application should retry to setup the connection again.

#### 1. **int =\_SetupSocketServer (Interface,PortNumber,ServiceGroup\_AcceptFunc);**

```
java: int _SetupSocketServer(String LInterface,int PortNumber,String AcceptFunc)
```

Set up a SOCKET server. If returns 0, then means failure, or else is the ID of the connection.

Interface is link-layer interface, may be ""

```
def ServiceGroup_AcceptFunc( self,ConnectionID, IPAddr, IPPort, MachineID ) :
```

```
    return ClientFunc    #-- should return the function which processes the connection.
```

```
function ServiceGroup:AcceptFunc( ConnectionID, IPAddr, IPPort, MachineID )
```

```
    return ClientFunc    -- should return the function to processes the connection.
```

```
end
```

```
java/c#:
```

```
public String AcceptFunc(int ConnectionID, String IPAddr, int IPPort, int MachineID)
{
}
}
```

MachineID is the state machine ID allocated for the client.

#### 2. **int =\_SetupSocketServer \_P(Interface,PortNumber,CallBackProc);**

for lua, ruby, python, this function is same as \_SetupSocketServer

```
java:          int          _SetupSocketServer_P(String          LInterface,int
PortNumber,StarSrvGroupAcceptFuncInterface CallBackProc)
```

```
public interface StarSrvGroupAcceptFuncInterface{
    public Object Invoke(Object SrvGroup,int ConnectionID, String IPAddr, int IPPort, int
MachineID);
}
```

the returned object should be the instance of interface:

```
public interface StarSrvGroupClientFuncInterface{
    public void Invoke(Object SrvGroup,int uMes,int MachineID, int
LinkInterfaceStatus,Object Para1, Object Para2);
}
```

```
c#:          int          _SetupSocketServer_P(String          LInterface, int
```

```
PortNumber, StarSrvGroupAcceptFuncInterface CallBackProc);
```

```
public delegate object StarSrvGroupAcceptFuncInterface(object SrvGroup, int
ConnectionID, string IPAddr, int IPPort, int MachineID);
```

the returned object should be the instance of delegate:

```
public delegate void StarSrvGroupClientFuncInterface(object SrvGroup, int uMes,
int MachineID, int LinkInterfaceStatus, object Para1, object Para2)
```

### 3. **int**     **=\_SetupSocketClient**     (**ServerInterface**     **,ServerName,**     **PortNumber,** **ServiceGroup\_ClientFunc);**

```
java: int _SetupSocketClient(String ServerInterface,String ServerName,int PortNumber,String
ClientFunc)
```

Setup a SOCKET client, which return the request ID which can be used to close the connection. The ID will be invalid after the callback function is called.

Interface is link-layer interface, may be “”

```
def ServiceGroup_ClientFunc( self,uMes,MachineID, LinkInterfaceStatus,Para1, Para2 ) :
    uMes == 1 : VS_SOCKET_ONCONNECT, Para1 is ipaddress, Para2 is port
    uMes == 2 : connection is closed
    uMes == 3 : fail to setup connection
    uMes == 4 : receive one package ,Para1 is ParaPkg.
```

```
function ServiceGroup:ClientFunc( uMes,MachineID, LinkInterfaceStatus,Para1, Para2 )
    uMes == 1 : VS_SOCKET_ONCONNECT, Para1 is ipaddress, Para2 is port
    uMes == 2 : connection is closed
    uMes == 3 : fail to setup connection
    uMes == 4 : receive one package ,Para1 is ParaPkg
    return
end
```

java/c#:

```
public void ClientFunc( int uMes,int MachineID, int LinkInterfaceStatus,Object Para1, Object
Para2 ){
}
```

LinkInterfaceStatus =0 normal   1 download, 2 error.

### 4. **int** **=\_SetupSocketClient\_P** (**ServerInterface** **,ServerName,** **PortNumber,** **CallBackProc);**

for lua, ruby, python, this function is same as \_SetupSocketClient

```
java:     int     _SetupSocketClient_P(String     ServerInterface,String     ServerName,int
PortNumber,StarSrvGroupClientFuncInterface CallBackProc)
```

```
public interface StarSrvGroupClientFuncInterface{
    public void Invoke(Object SrvGroup,int uMes,int MachineID, int
LinkInterfaceStatus,Object Para1, Object Para2);
}
```



```
c#: int _SetupSocketClient_P(String^ ServerInterface, String^ ServerName, Int32
PortNumber, StarSrvGroupClientFuncInterface^ CallBackProc);
```

```
public delegate void StarSrvGroupClientFuncInterface(object SrvGroup, int uMes,
int MachineID, int LinkInterfaceStatus, object Para1, object Para2)
```

##### 5. **\_CloseSocketConnect (ConnectionID/MachineID);**

```
java: void _CloseSocketConnect(int ConnectionID_MachineID)
```

Close connection. Input may be connection ID, or request ID, or client MachineID.

##### 6. **Bool =\_SocketSend (MachineID,ParaPkg,bool assure);**

```
java: boolean _SocketSend(int MachineID,StarParaPkgClass ParaPkg,boolean Assure)
```

send one package.

#### 5.5.16 Timer function

##### 1. **\_SetTimer (Ticket, ServiceGroup\_Func, int Arg1, int Arg2 )**

```
java: int _SetTimer(int Ticket,String TimerFunc,int Arg1,int Arg2)
```

Function is function which should be defined in service group.

Ticket is the interval which uint is 10ms;

calling format:

```
a = ServiceGroup._SetTimer(100,Func, 0,0 )
```

Parameter should be number, and the return value is TimerID

```
def ServiceGroup_Func(self,TimerID,Arg1,Arg2) :
```

self is service group object

```
calling format:a = ServiceGroup:_SetTimer(100,Func, 0 )
```

Parameter should be number, and the return value is TimerID

```
function ServiceGroup:Func(TimerID,Arg1,Arg2)
```

end

```
java: public void Func(int TimerID,int Arg1,int Arg2)
```

```
{
```

```
    return;
```

```
}
```

##### 2. **\_SetTimer\_P (Ticket, int Arg1, int Arg2, CallBackProc )**

for lua, ruby, python, this function is same as \_SetTimer

```
java: int _SetTimer_P(int Ticket,int Arg1,int Arg2,StarSrvGroupTimerInterface
CallBackProc)
```

```
public interface StarSrvGroupTimerInterface{
```

```
    public void Invoke(Object SrvGroup,int TimerID,int Arg1,int Arg2);
```

```
}
```

```
c#: int _SetTimer_P(int Ticket,int Arg1,int Arg2,StarSrvGroupTimerInterface CallBackProc);
```

```
public delegate void StarSrvGroupTimerInterface(Object SrvGroup,int TimerID,int Arg1,int Arg2);
```

### 3. **\_KillTimer (TimerID)**

```
java: void _KillTimer(int TimerID)
```

TimerID is the ID of the timer.

#### 5.5.17 Restart function

### 1. **\_ProgramRestart();**

```
java: void _ProgramRestart()
```

The function is valid only when cle is started by manager program.

#### 5.5.18 Http up/download function

### 1. **\_HttpDownload (ServerUrl,ClientPath,FileName);**

```
java: void _HttpDownload(String ServerUrl,String ClientPath,String FileName)
```

FileName is the file name. The download process is asynchronious and the file downloaded will be saved at local.

```
SrvGroup._HttpDownload("http://www.srplab.com/Files","e:","srrrlicht_index.htm")
```

### 2. **\_HttpDownloadAbort ();**

```
java: void _HttpDownloadAbort()
```

Cancel all http/ftp downloads.

### 3. **\_RegWebDownFunction (ServiceGroup\_Func);**

```
java: void _RegWebDownFunction(String DownFunc)
```

The prototype is :

```
def ServiceGroup_Func (self,uMes,FileName,MaxSize,CurSize)
```

```
function ServiceGroup:Func (uMes,FileName,MaxSize,CurSize)
```

```
end
```

```
java:
```

```
public void DownFunc(int uMes,String FileName,long MaxSize,long CurSize)
{
    return;
}
```

uMes:

- 0 //---Start download
- 1 //---download process
- 2 //---finish
- 3 //---error

#### 4. **\_RegWebDownFunction\_P(CallBackProc);**

for lua, ruby, python, this function is same as \_RegWebDownFunction

```
java: void _RegWebDownFunction_P(StarSrvGroupWebDownInterface CallBackProc)
```

```
public interface StarSrvGroupWebDownInterface{
    public void Invoke(Object SrvGroup,int uMes,String FileName,long MaxSize,long
CurSize);
}
```

```
c#: void _RegWebDownFunction_P(StarSrvGroupWebDownInterface CallBackProc);
```

```
public delegate void StarSrvGroupWebDownInterface(object SrvGroup, int uMes, string
FileName, long MaxSize, long CurSize);
```

### 5.5.19 set current Url and service parameter

#### 1. **\_SetEnvCurrentUrl (Url)**

```
java: void _SetEnvCurrentUrl(String Args)
```

Set current service Url

#### 2. **\_SetEnvPara (ParaPkg)**

```
java: void _SetEnvPara(StarParaPkgClass ParaPkg)
```

set current service parameter

### 5.5.20 script edit[Windows]

#### 1. **Bool=\_OpenLuaEdit (Module,Config,bool CloseEnable)**

```
java: boolean _OpenLuaEdit(String Module,int Config,boolean CloseEnable)
```

Open editor which is compiled to single share library SRPLuaEdit.DLL. The host application is responsible for windows message loop.

Config is the combination of the following value:

```
#define SRPLUAEDITMODULECONFIG_SCRIPTCONSOLE 0x00000001
```

```
#define SRPLUAEDITMODULECONFIG_PROJECT 0x00000002
```

```
#define SRPLUAEDITMODULECONFIG_SRPDOC 0x00000004
```

Module is reserved.

#### 2. **\_LuaEditDisp (Info)**

```
java: void _LuaEditDisp(String Info)
```

Display information in editor.

#### 3. **\_LuaEditHelp (Type,HelpFile)**

```
java: void _LuaEditHelp(int Type,String HelpFile)
```

Type=0 help info

Display help in editor. Input is help file name which is located on local disk.

=1 Examples project, at this time, the input is help string.

#### 4. **\_CloseLuaEdit ()**

```
java: void _CloseLuaEdit()
```

Close editor.

### 5.5.21 *service path*

#### 1. **\_InsertSearchPath (String Path);**

```
java: void _InsertSearchPath(String Path)
```

Set service search path.

#### 2. **\_ClearSearchPath ();**

```
java: void _ClearSearchPath()
```

Clear service search path.

#### 3. **Path=\_FirstSearchPath (QueryRecord);**

```
java: String _FirstSearchPath(StarQueryRecordClass QueryRecord)
```

get first search path.

#### 4. **Path=\_NextSearchPath (QueryRecord);**

```
java: String _NextSearchPath(StarQueryRecordClass QueryRecord)
```

get next search path.

### 5.5.22 *get doc object registered*

#### 1. **Object,DocName=\_FirstDoc( QueryRecord )**

```
java: Object[] _FirstDoc(StarQueryRecordClass QueryRecord)
```

Get first doc object registered.

#### 2. **Object,DocName = \_NextDoc( QueryRecord )**

```
java: Object[] _NextDoc(StarQueryRecordClass QueryRecord)
```

Get next doc object registered.

#### 3. **\_RegisterDoc(Object,DocName)**

```
java: void _RegisterDoc(Object ObjectArg,String DocName)
```

Register Doc object.

#### 4. **\_UnRegisterDoc(Object)**

```
java: void _UnRegisterDoc(Object ObjectArg)
```

Unregister Doc object.

### 5.5.23 *get static data version*

#### 1. **Version=\_GetStaticVersion (BinBuf/FileName)**

```
String _GetStaticVersion(Object Arg)
```

Compute static data version, input is file name or binary buffer.

### 5.5.24 *Clipboard[Windows]*

#### 1. **\_ToClipboard(String)**

```
java: void _ToClipboard(String Arg)
```

Copy string to clipboard.

## 2. **String=\_FromClipboard()**

```
java: String _FromClipboard()
```

Get string from clipboard.

### 5.5.25 Load service from Url

#### 1. **int Result=\_RunFromUrl(Url,Type,bool WaitFlag)**

```
java: int _RunFromUrl(String Url,int Type,boolean WaitFlag)
```

**Note: This function should not be called in normal case.**

Type:

VS\_RUMFROMURL\_NORESTART     0

VS\_RUMFROMURL\_RESTART       1

VS\_RUMFROMURL\_WAITRESTART   2

Result:

SRPLOADPROCESS\_OK           0

SRPLOADPROCESS\_BUSY        -1

SRPLOADPROCESS\_DISABLE     -2

SRPLOADPROCESS\_FAIL        -3

After Url may be attach parameter such as"?parameter";

hostip=XXX: Redirect host IP, which is valid for files on http or ftp server. In this case, the url uses this ip address other than returned by DNS.

depend=#depended service 1,# depended service 2, depended service 3; '#' is in front of the depended service name, which should be downloaded from starcore website, otherwise downloaded from parent url ["depend=" without spaces]

script=lua/python/..; [ "script=" without spaces]

The last string is command argument string, service can read it from EnvInputPara for examples:

<http://XXX/XX?depend=AAA,#bbb;para1=111>

<http://XXX/XX?depend=AAA,#bbb;script=python;para1=111>

<http://XXX/XX?para1=111>

depend,script,and command string are seperated by ";;".

If use FTP, the user name and password is input as follow:

<ftp://XXX?USER=XXX;PASS=XXX/XXX/XXX>

<ftp://XXX:21?USER=XXX;PASS=XXX/XXX/XXX>

If hostip is set, then when download, url will be replaces by hostip. For example:

<ftp://XXX?USER=XXX;PASS=XXX/XXX/XXX?hostip=127.0.0.1>,

<ftp://XXX:21?USER=XXX;PASS=XXX/XXX/XXX?hostip=127.0.0.1>,

Download will be started from <ftp://127.0.0.1> 和 <ftp://127.0.0.1:21>

If Url starts with char “@”, then the service file locates at local disk.

#### 5.5.26 Set debug and client port

##### Valid for service group 0

###### 1. Bool= \_SetClientPort (“Interface”,Portnumber)

```
java: boolean _SetClientPort(String LInterface,int Portnumber)
```

Interface may be “““

###### 2. Bool= \_SetDebugPort (“Interface”,Portnumber)

```
java: boolean _SetDebugPort(String LInterface,int Portnumber)
```

Interface may be “““

Interface is string, which format is :

“Host=interface address;if=share lib file name(include extension);site=share library download address, ftp/http address ;para=parameter string”

for example:" host=192.168.0.1;if=SRPTcpLinkInterface1.dll;site=<ftp://127.0.0.1>"

#### 5.5.27 set Telnet, Web, and output port

##### Valid for service group 0

###### 1. Bool= \_SetTelnetPort (Portnumber)

```
java: boolean _SetTelnetPort(int Portnumber)
```

###### 2. Bool= \_SetOutputPort (Host,Portnumber)

```
java: boolean _SetOutputPort(String Host,int Portnumber)
```

Using syslog to receive information

Output is coded as UTF-8 format.

###### 3. Bool= \_SetWebServerPort (Host,Portnumber, ConnectionNumber, PostSize)

```
java: boolean _SetWebServerPort(String Host,int Portnumber,int ConnectionNumber,int PostSize)
```

Host is Url, which may be set to None

ConnectionNumber: the maximum connection number

PostSize : permit upload size, unit is Kbytes.

#### 5.5.28 service register and alloc Cooperator

###### 1. Bool= \_RegisterServer (ServiceName)

```
java: boolean _RegisterServer(String ServiceName)
```

###### 2. \_AllocCooperator (ServiceName)

```
java: void _AllocCooperator(String ServiceName)
```

###### 3. \_FreeCooperator(ServiceName)

```
java: void _FreeCooperator(String ServiceName)
```

###### 4. Interface,Host,Port= \_GetServerUrlInfo ()

```
java: Object[] _GetServerUrlInfo()
```

Local function to get parameters for client to connect

### 5.5.29 *WebService object refresh*

#### 1. **\_WebServiceRefresh()**

```
void _WebServiceRefresh()
```

### 5.5.30 *get WSDL*

#### 1. **bool =\_GetWSDL(int WSDLVersion, string Host, BinBuf)**

```
java: boolean _GetWSDL(int WSDLVersion,String Host,StarBinBufClass BinBuf)
```

WSDLVersion: reserved, current is version 1.1.

Host: If equals to nil, then uses default host.

Mapping type:

If object sets WebServiceFlag, then it acts as a PortType mapped to WSDL. And the functions defined in object act as Operation.

### 5.5.31 *string format convert*

#### 1. **String= \_ToAnsi(Lang,String)**

#### 2. **String= \_FromAnsi(Lang,String)**

for windows:

Lang should be "utf8" or "ansi", or ".XXX", XXX is number of codepage, such as ".950"

note:

java and c# does not support this function

### 5.5.32 *get platform information*

#### 1. **EnvTag= \_GetConfigEnvTag()**

```
java: String _GetConfigEnvTag()
```

Get environment Tag which is set through VS\_STARCONFIGEX at cle initial process.

Three types are predefined:

"" , "nolooop", "activex"

#### 2. **bool debugserver port result,client port config result,telnet port config result,WebServer port config result = \_GetConfigResult()**

```
java: Object[] _GetConfigResult()
```

#### 3. **\_GetConfig (SXml)**

Get config parameter.

```
java: void _GetConfig(StarSXmlClass SXml)
```

#### 4. **Host=\_GetConfigHost()**

```
java: String _GetConfigHost()
```

Get config Host and Web port number.

### 5.5.33 client download callback (used at server service group)

#### 1. **RefValue=\_RegFileReqCallBack(SrvGroup.Function);**

```
java: long _RegFileReqCallBack(String Function)
```

#### 2. **RefValue=\_RegFileReqCallBack\_P(CallBackProc);**

for lua, ruby, python, this function is same as \_RegFileReqCallBack

```
java: long _RegFileReqCallBack_P(StarSrvGroupFileReqInterface CallBackProc)
```

```
public interface StarSrvGroupFileReqInterface{
    public boolean Invoke(Object SrvGroup,int ClientID,int ClientPrivateTag,int
uMsg,boolean DataFile,int DataSize,String FileName_ObjectID,int UniqueDataUnitID,String
Version);
}
```

```
c#:          long          SrvGroup_RegFileReqCallBack_P(object
selfobj, StarSrvGroupFileReqInterface CallBackProc);
```

```
public delegate bool StarSrvGroupFileReqInterface(object SrvGroup,int
ClientID,int ClientPrivateTag,int uMsg, Boolean DataFile,int DataSize, String
FileName_ObjectID,int UniqueDataUnitID, String Version);
```

#### 3. **\_UnRegFileReqCallBack(RefValue);**

```
java: void _UnRegFileReqCallBack(long RefValue)
```

Function prototype:

```
def ServiceGroup_CallbackProc( self, ClientID, ClientPrivateTag, uMsg, DataFile,
DataSize, FileName Or ObjectID, UniqueDataUnitID,Version )
```

```
java:
    public boolean Function(int ClientID,int ClientPrivateTag,int uMsg,boolean DataFile,int
DataSize,String FileName_ObjectID,int UniqueDataUnitID,String Version)
    {
        return true/false;
    }
```

If DataFile = true, then is FileName

If DataFile = false, then is ObjectID.

UniqueDataUnitID represents APPID of static data, and Version represents the version of static data.

uMsg takes value:

0 : start to download

4 : start to upload

if returns true, data can be downloaded or uploaded, or else can not be downloaded or uploaded.

### 5.5.34 Dispatch callback

#### 1. **void \_RegDispatchCallBack(SrvGroup\_Function)**



```
java: void _RegDispatchCallBack(String Function)
```

If input is None, then previous callback will be canceled.

```
def SrvGroup_Function(self)
```

```
java:
```

```
    public void Function()
    {
        return;
    }
```

Should only register on service group 0, and only once.

## 2. **bool \_RegDispatchCallBack\_P(CallBackProc)**

for lua, ruby, python, this function is same as \_RegDispatchCallBack

```
java: void _RegDispatchCallBack_P(StarSrvGroupDispatchInterface CallBackProc)
```

```
public interface StarSrvGroupDispatchInterface{
    public void Invoke(Object SrvGroup);
}
```

```
c#: void _RegDispatchCallBack_P(StarSrvGroupDispatchInterface^ CallBackProc);
```

```
public delegate void StarSrvGroupDispatchInterface(Object SrvGroup);
```

### 5.5.35 Raw Object Interface

Raw object supports lua, python, ruby, java, and c#

## 1. **boolean \_InitRaw(String ScriptInterface,StarServiceClass Service)**

This function is used to init script interface for raw function.

## 2. **boolean \_LoadRawModule(String ScriptInterface,String ModuleName,String FileOrString, boolean IsString )**

Load Raw Module, which may be lua module, python module, ruby module, java class library, csharp class library. for version 2.0, the above four scripts are supported.

ModuleName : if has been loaded, the function does nothing. Module name may be input null for global name space.

RunString : may be file or string buf, which will be executed.

IsString : true for string, false for file.

ErrorInfo : may be null;

**for lua :**

if RunString == true, then load and execute lua code.

if RunString == false, then load and execute lua file.

**for python/ruby :**

If IsString == false, and FileOrString is valid, then load python file as module.

If `IsString == true/false`, and `FileOrString` is invalid, using `python import` function to load module

If `IsString == true`, and `FileOrString` is valid, then load python code as module.

**for java :**

IsString will be ignored.

If FileOrString is valid, then take FileOrString as class library file. Or else, take ModuleName as class library file.

**for csharp :**

IsString will be ignored.

If `ModuleName` is valid and `FileOrString` is invalid, then load Assembly by name.

If FileOrString is valid, then load Assembly by file.

```
3. boolean _LoadRawModuleEx(String ScriptInterface,String
   ModuleName,StarObjectClass object )
```

This function is valid for csharp, object should be raw assembly object.

```
4. boolean _DefScriptRawType(String ScriptInterface,String ModuleName,String
FileOrString, boolean IsString )
```

The function is reserved.

```
5. int _RegScriptRawType(String ScriptInterface,String TypeGroupName,int
   GroupIndex,String TypeName)
```

The function is reserved.

```
6. int _GetScriptRawType(String ScriptInterface,String TypeGroupName,String
   TypeName)
```

The function is reserved.

## 7. String[] \_GetScriptRawTypeEx(int RawType)

The function is reserved.

### 5.5.36 Get Last Error

## 1. int \_GetLastError()

## 2. String\_GetLastErrorInfo()

### 5.5.37 UnLockGC Log

### 1. void \_LogObjectFreeByUnLock(boolean Flag)

If `Flag == true` and `object` is freed by `UnLockGC/DelRefEx` or script, a message is output to

console.

### 5.5.38 \_SUnLockGC

#### 1. void \_SUnLockGC()

This function is provided for check all cle object's with ReleaseOwnerEx record and trigger \_SUnLockGC function of the corresponding script interface.

### 5.5.39 Get Path and Local IP

#### 1. String \_GetCorePath()

#### 2. String \_GetUserPath()

#### 3. String \_GetLocalIP()

Note: This function always returns "127.0.0.1" for android

#### 4. String[] \_GetLocalIPEx()

Note: This function always returns the address of localhost for android.

### 5.5.40 Get Number of Object created[2.5.1]

#### 1. int \_GetObjectNum()

### 5.5.41 Active Script and Pre-compile Script Segment [2.5.1]

#### 1. Bool[2] = \_ActiveScriptInterface(String ScriptInterfaceName)

Bool[0]: true for success

Bool[1]: true if the script supports online compile.

#### 2. Object[2] = \_PreCompile(String ScriptInterfaceName,String ScriptSegment)

Compile the scriptsegment to check there is syntax error or not.

Object[0]: boolean value, true for success

If Object[0] is false, then,

Object[1] is string value, if the length of Object[1] is zero, the scriptsegment is not finish, more script sentence needed. Or else, Object[1] returns the compiler error info.

For example,

\_PreCompile("python","print(123)")

## 5.6 Service object

### 5.6.1 attribute

#### 1. Service.\_Path

service path.

**2. Service.\_Name**

service name.

**3. Service.\_FrameTimeInterval**

Get interval between frames. The return is number which unit is 10ms.

**4. Service.\_ID**

ID of the service

**5. Service.\_FrameTicket**

Frame ticket of the service.

**6. Service.\_ServiceGroup**

Service group which service belongs to.

*5.6.2 basic function*

**1. \_DeactiveAll ()**

java: void \_DeactiveAll()

deactive all objects.

**2. \_GetObject (ObjectName )**

java: StarObjectClass \_GetObject(String ObjectName)

get object, input is object name

**3. \_GetObjectEx (ObjectID );**

java: StarObjectClass \_GetObjectEx(String ObjectID)

ObjectID is the ID of the object.

Calling format : a = Service.\_GetObjectEx ("66efc37a-a16c-442b-bc2a-3df5b03b3294")

**4. \_GetObjectEx2 (ServiceName,ObjectName);**

java: StarObjectClass \_GetObjectEx2(String ServiceName,String ObjectName)

Get object of special service.

**5. \_GetSysRootItem(SysRootItemName)**

java: StarServiceItemClass \_GetSysRootItem(String SysRootItemName)

Get service item.If service item does not active, the returns None.

**6. \_Exit ()**

java: void \_Exit()

exit the service.

**7. \_Save (Path=None)**

java: void \_Save(String Path)

Save service

**8. \_IsChange()**

java: boolean \_IsChange()

whether the service is changed

**9. \_IsActive ()**

java: boolean \_IsActive()

whether the service is active.

**10. \_QueryFirstFromSDT()**

java: StarObjectClass \_QueryFirstFromSDT()

Query first object from service table

**11.     \_QueryNextFromSDT()**

java: StarObjectClass \_QueryNextFromSDT()

Query next object from service table

**12.     \_PrintInfo ()**

java: void \_PrintInfo()

Print service information.

**13.     Bool         =\_CreateSysRootItem(SysRootItemName,         DependSysRootItem, SysRootItemID=None,SysRootItemIDEx=None)/SysRootItem=\_CreateSysRootItemEx(SysRootItemName, DependSysRootItem,SysRootItemID=None, SysRootItemIDEx=None)**

java:         boolean         \_CreateSysRootItem(String         SysRootItemName,String DependSysRootItem,String SysRootItemID,String SysRootItemIDEx)

java:     StarServiceItemClass     \_CreateSysRootItemEx(String     SysRootItemName,String DependSysRootItem,String SysRootItemID,String SysRootItemIDEx)

Create service item,SysRootItemID is UUID string, SysRootItemIDEx is UUID string which may be "",DependSysRootItem is the depended service item which may be "" if not exist.

**14.     \_ActiveSysRootItem(SysRootItemName)**

java: void \_ActiveSysRootItem(String SysRootItemName)

Activate service item, no return value.

**15.     \_ActiveAllSysRootItem()**

java: void \_ActiveAllSysRootItem()

Activate all service items, no return value, valid at server service group.

**16.     \_DeactiveSysRootItem (SysRootItemName)**

java: void \_DeactiveSysRootItem(String SysRootItemName)

deactivate service item, no return value.

**17.     \_ActiveCSysRootItem(ClientID,SysRootItemName)**

java: void \_ActiveCSysRootItem(int ClientID,String SysRootItemName)

activate service item at client service group, no return value, should be called at server service group.

**18.     \_DeactiveCSysRootItem (ClientID,SysRootItemName)**

java: void \_DeactiveCSysRootItem(int ClientID,String SysRootItemName)

deactivate service item at client side, no return value, valid at server service group.

**19.     Name = \_QueryFirstSysRootItem ()**

java: String \_QueryFirstSysRootItem()

query first service item name

**20.     Name = \_QueryNextSysRootItem ()**

java: String \_QueryNextSysRootItem()

query next service item name

**21.     Name,ID,Continue = \_QueryFirstDepend (QueryRecord)**

java: Object[] \_QueryFirstDepend()

query first service depended.

**22.     Name,ID,Continue =\_QueryNextDepend (QueryRecord)**

java: Object[] \_QueryNextDepend()

query next service depended.

**23. bool = \_IsOsSupport (ProgramRunType,OsType )**

```
java: boolean _IsOsSupport(int ProgramRunType,int OsType)
```

Returns true if the corresponding OSType share library of modules of the service and depended service exists.

**24. bool = \_SetClientObject (ClientID,Object)**

```
java: boolean _SetClientObject(int ClientID,StarObjectClass Obj)
```

Called at server, each client should only exist one. ClientID can not be set to 0.

**25. Object = \_GetClientObject()**

```
java: StarObjectClass _GetClientObject()
```

Gets client object, called at client side.

**26. \_SetPrivateTag ( ClientPrivateTag)**

```
java: void _SetPrivateTag(int ClientPrivateTag)
```

Valid at client side. The tag is used for server to check the validity of remote call, global object creating, changing or deleting operations.

**27. value=Service.\_GetOPPermission()**

```
java: int _GetOPPermission()
```

Get operation permission. valid at client side.

*5.6.3 client login function***1. \_RegMachineFunction(Service\_ClientRegisterProc);**

```
java: void _RegMachineFunction(String FunctionName)
```

Valid at server service group. The callback function is registered to process client login, which format is:

```
def Service_ClientRegisterProc( self,int uMes,int ClientID,char SrcServiceID,
SrcServiceAdd,int SrcServicePort,ParaPkg,UserName,UserPassword) :
```

```
function Service:ClientRegisterProc( int uMes,int ClientID,char SrcServiceID,
SrcServiceAdd,int SrcServicePort,ParaPkg,UserName,UserPassword):
```

java/c#:

```
public void ClientRegisterProc(int uMes,int ClientID,String SrcServiceID, String
SrcServiceAdd,int SrcServicePort,StarParaPkgClass ParaPkg, String UserName,
String UserPassword)
```

No result needed return from the callback. If client is permit to login, then the callback should call \_AcceptClient(ClientID,True,True). If not, should call \_AcceptClient(ClientID,False,False). In addition, \_AcceptClient may be called sometime after the callback function.

For Login, uMes = 1

SrcServiceAdd and SrcServicePort are source service address and port which are invalid other than in service redirect process. ParaPkg is login parameter.

For Logout, uMes = 3,

SrcServiceID, SrcServiceAdd, SrcServicePort, ParaPkg will be set to None

For LoadServiceOk, uMes = 2,

SrcServiceID, SrcServiceAdd, SrcServicePort, ParaPkg will be set to None

## 2. **\_RegMachineFunction\_P(CallBackProc);**

for lua, ruby, python, this function is same as \_RegMachineFunction

java: void \_RegMachineFunction\_P(StarServiceMachineInterface CallBackProc)

```
public interface StarServiceMachineInterface{
    public void Invoke(Object Service,int uMes,int ClientID,String SrcServiceID, String
SrcServiceAdd,int SrcServicePort,Object ParaPkg, String UserName, String
UserPassword);
}
```

c#: void \_RegMachineFunction\_P(StarServiceMachineInterface CallBackProc);

```
public delegate void StarServiceMachineInterface(Object Service,int uMes, int
ClientID,String SrcServiceID, String SrcServiceAdd, int SrcServicePort,Object
ParaPkg, String UserName, String UserPassword);
```

## 3. **Bool =\_AcceptClient (ClientID,True/False,bool ReSyncFlag,char \*TermOldScript, char \*InitNewScript, ClientPrivateTag ,ClientOPPermission, ClientUploadMaxSize);**

java: boolean \_AcceptClient(int ClientID,boolean Result,boolean ReSyncFlag,String TermOldScript,String InitNewScript,int ClientPrivateTag,int ClientOPPermission,int ClientUploadMaxSize)

Valid at server which is used to accept or reject client login, and is called in ClientRegisterProc function or sometime later. Login procedure is finished after the function is called. If ReSyncFlag == True, then client will be forced to resynchronize for the same service, otherwise not.

TermOldScript: is the lua script called before service is load, should use carefully. If no active service exists at client side, then it will be ignored. Before running the script, CLE creates a lua global variable “\_gService” which represents current service.

InitNewScript: is the lua script called before service is load Before running the script, CLE creates a lua global variable “\_gService” represents current service.

TermOldScript + InitNewScript: total length should < 32Kbytes.

If is redirected from the same service, then only InitNewScript is valid.

When InitNewScript is executed, the service is not synchronized, therefore do not operate objects in the service.

ClientOPPermission takes value from list:

```
#define VSCLIENTOP_CREATE ((VS_ULONG)0x00000001)
```

```
#define VSCLIENTOP_DELETE ((VS_ULONG)0x00000002)
```

```
#define VSCLIENTOP_CHANGE ((VS_ULONG)0x00000004)
```

ClientUploadMaxSize: maximum size permitted for upload at client side, which unit is bytes.

Client side uses SetStaticData to upload data.

**4. IPStr,OsType,RunType =\_GetClientInfo (ClientID);**

```
java: Object[] _GetClientInfo(int ClientID)
```

Valid at server, returns client informaton

**5. \_GetClientNumber ();**

```
java: int _GetClientNumber()
```

returns client number.

**6. \_PrintClientInfo ();**

```
java: void _PrintClientInfo()
```

prints all clients information.

**7. \_DelClient (ClientID);**

```
java: void _DelClient(int ClientID)
```

Valid at client and server. The connection will be closed after the function call.

*5.6.4 object operation call back—vaild at server [object change, create or delete]***1. \_RegClientOpFunction(Service\_Func);**

```
java: void _RegClientOpFunction(String FunctionName)
```

The function prototype is:

```
def Service_Func (self,Op,ClientID,ClientPrivateTag,Object,AttrIndex,ClassID) :
```

```
    return True
```

```
function Service:Func (Op,ClientID,ClientPrivateTag,Object, AttrIndex,ClassID)
```

```
    return true
```

```
end
```

```
java/c#:
```

```
public boolean OpFunction(int Op,int ClientID,int ClientPrivateTag,StarObjectClass Obj,int
AttrIndex,String ClassID)
```

ClientPrivateTag: is used to determine whether the client is legal.

If the function returns False, then the operation is denied, otherwise is permitted

```
///---for delete Op = 0; Object is valid
```

```
///---for change Op = 1; Object is valid
```

```
///---for create Op = 2; Object is the parent object,ClassID is id of the class.
```

**2. \_RegClientOpFunction\_P(CallBackProc);**

for lua, ruby, python, this function is same as \_RegClientOpFunction

```
java: void _RegClientOpFunction_P(StarServiceClientOpInterface CallBackProc)
```

```
public interface StarServiceClientOpInterface{
```

```
    public boolean Invoke(Object Service,int Op,int ClientID,int ClientPrivateTag,Object
Obj,int AttrIndex,String ClassID);
```

```
}
```



```

c#: void _RegClientOpFunction_P(StarServiceClientOpInterface CallBackProc);

public delegate bool StarServiceClientOpInterface(Object Service,int Op,int
ClientID,int ClientPrivateTag, Object Obj,int AttrIndex,String ClassID);

```

### 5.6.5 service redirect

#### 1. **\_Redirect (ClientID, DesServerInterface,DesServerName, DesServerPortNumber,ParaPackage, Service\_ClientRedirectProc=None);**

```

java: void _Redirect(int ClientID,String DesServerInterface,String DesServerName,int
DesServerPortNumber,StarParaPkgClass ParaPackage,String RedirectProcName)

```

DesServerInterface may be ""

Valid at server, direct client to connect to other server, no return value.

```

def          Service_ClientRedirectProc(          self,uMsg,          ClientID,
DesServerName,DesServerPortNumber) :

```

```

function          Service:ClientRedirectProc(          uMsg,          ClientID,
DesServerName,DesServerPortNumber)

```

```

java/c#:
public void ClientRedirectProc( int uMsg, int ClientID, String DesServerName,int
DesServerPortNumber)

```

uMsg = 0 : redirect succeed. Server will close the connection after the callback by cle.

uMsg = 1 : redirect failed.

call examples:

```
_Redirect(ClientID,DesServerName,DesServerPortNumber)
```

```
_Redirect(ClientID,DesServerName,DesServerPortNumber, ClientRedirectProc)
```

```
_Redirect(ClientID,DesServerName,DesServerPortNumber,ParaPackage, ClientRedirectProc)
```

#### 2. **\_Redirect\_P (ClientID, DesServerInterface,DesServerName, DesServerPortNumber,ParaPackage, CallBackProc);**

for lua, ruby, python, this function is same as \_Redirect

```

java: public void _Redirect_P(int ClientID,String DesServerInterface,String
DesServerName,int DesServerPortNumber,StarParaPkgClass
ParaPackage,StarServiceRedirectInterface CallBackProc)

```

```

public interface StarServiceRedirectInterface{
    public void Invoke(Object Service,int uMsg, int ClientID, String DesServerName,int
DesServerPortNumber);
}

```

```

c#: void _Redirect_P(int ClientID, String DesServerInterface, String
DesServerName, int DesServerPortNumber, StarParaPkgClass
ParaPackage, StarServiceRedirectInterface CallBackProc);

```

```
public delegate void StarServiceRedirectInterface(Object Service,int uMsg, int
ClientID, String DesServerName,int DesServerPortNumber);
```

#### 5.6.6 file upload and download function

##### 1. **\_Download (ServerPath,ClientPath,FileName); [reserved]**

```
java: void _Download(String ServerPath,String ClientPath,String FileName)
```

Download file from server, valid at client or debug.

##### 2. **\_HttpDownload (ServerUrl, ClientPath,FileName);**

```
java: void _HttpDownload(String ServerUrl,String ClientPath,String FileName)
```

Service.\_HttpDownload("http://www.srplab.com/Files","e:","srrlicht\_index.htm")

##### 3. **\_HttpDownloadAbort ();**

```
java: void _HttpDownloadAbort()
```

SrvGroup: \_HttpDownloadAbort ()

Cancel all http/ftp download process.

##### 4. **\_Upload (ServerPath,ClientPath,FileName); [reserved]**

```
java: void _Upload(String ServerPath,String ClientPath,String FileName)
```

Upload file to server, valid at client or debug.

##### 5. **long RefValue=\_RegFileCallBack(Service\_Function);**

```
java: long _RegFileCallBack(String FunctionName)
```

valid at client or debug. should only register one.

##### 6. **long RefValue=\_RegFileCallBack\_P(CallBackProc);**

for lua, ruby, python, this function is same as \_RegFileCallBack

```
java: long _RegFileCallBack_P(StarServiceFileCallBackInterface CallBackProc)
```

```
public interface StarServiceFileCallBackInterface{
    public void Invoke(Object Service,int uMsg, boolean DataFile,int
ReceiveOrSendOffset,int DataSize, Object FileName_Object, int UniqueDataUnitID,String
Version);
}
```

```
c#: int _RegFileCallBack_P(StarServiceFileCallBackInterface CallBackProc);
```

```
public delegate void StarServiceFileCallBackInterface(Object Service,int uMsg,
bool DataFile,int ReceiveOrSendOffset,int DataSize, Object FileName_Object, int
UniqueDataUnitID,String Version);
```

##### 7. **\_UnRegFileCallBack(RefValue);**

```
java: void _UnRegFileCallBack(long RefValue)
```

Callback function prototype

```
def Service_DownloadProc( self, uMsg, DataFile,ReceiveOrSendOffset,DataSize,
FileName Or Object, UniqueDataUnitID,Version )
```

```
function Service: DownUpProc( uMsg, DataFile, ReceiveOrSendOffset, DataSize,
FileName Or Object, UniqueDataUnitID, Version )
```

```
java/c#:
public void DownUpProc( int uMsg, boolean DataFile, int ReceiveOrSendOffset, int
DataSize, Object FileName_Object, int UniqueDataUnitID, String Version )
```

If DataFile = true, then variable FileName\_Object is FileName

If DataFile = false, then variable FileName\_Object is ID, UniqueDataUnitID represents APPID of static data, and Version represents version of static data.

uMsg takes value:

0 : start download, 1:download progress,2:finish,3:error

4 : start upload, 5:upload progress,6:finish,7:error

### 5.6.7 create object (base class not exist)

#### 1. **\_New/\_NewEx (ObjectID,QueueAttrName,ParentObject)**

```
java: StarObjectClass _New(Object...Args)
```

```
java: StarObjectClass _NewEx(Object...Args)
```

refer to Object.\_New

#### 2. **\_NewGlobal/\_NewGlobalEx(ObjectID,ClientID, QueueAttrName,ParentObject)**

```
java: StarObjectClass _NewGlobal(Object...Args)
```

```
java: StarObjectClass _NewGlobalEx(Object...Args)
```

refer to Object.\_NewGlobal

#### 3. **\_NewClient/\_NewClientEx(ObjectID,ClientID, QueueAttrName,ParentObject)**

```
java: StarObjectClass _NewClient(Object...Args)
```

```
java: StarObjectClass _NewClientEx(Object...Args)
```

refer to Object.\_NewClient

### 5.6.8 service macro definition

#### 1. **\_PrintMacro (ServiceName.MacroName);**

```
java: void _PrintMacro(String ServiceName_MacroName)
```

Print macro definition. If MacroName does not exist, then print macro list of the service.

### 5.6.9 user management—valid at server, service object should be get with root user

#### 1. **Bool = \_CreateUser (UserName,UserPass, ReadWriteOrExecute);**

```
java: boolean _CreateUser(String UserName,String UserPass,int ReadWriteOrExecute)
```

If user does not exist, then create new one, or else change its information.

ReadWriteOrExecute may be combination of the following values:

READ 0x01

WRITE 0x02

EXECUTE 0x04

EXPORTXML 0x08

## 2. **\_DeleteUser(Username);**

```
java: void _DeleteUser(String Username)
```

## 3. **UserName, ReadWriteOrExecute= \_FirstUser(QueryRecord);**

```
java: Object[] _FirstUser(StarQueryRecordClass QueryRecord)
```

QueryRecord is allocated using function ServiceGroup.\_NewQueryRecord

## 4. **UserName, ReadWriteOrExecute= \_NextUser(QueryRecord);**

```
java: Object[] _NextUser(StarQueryRecordClass QueryRecord)
```

If user does not exist, the function returns None.

### 5.6.10 execute script

**Note: The execution of script may be not in the same thread with the caller**

## 1. **True/false, ErrorInfo =\_RunScript(ScriptInterface,ScriptBuf, ModuleName, WorkDirectory);**

```
java: Object[] _RunScript(String ScriptInterface,String ScriptBuf,String ModuleName,String WorkDirectory)
```

Executes script, WorkDirectory is work directory, may be "".

Static language does not support the function, such as java/c#.

ModuleName is name of the module, may be NULL. This Parameter is only valid for lua and python.

## 2. **True/false, ErrorInfo =\_RunScriptEx(ScriptInterface,BinBuf, ModuleName, WorkDirectory);**

```
java: Object[] _RunScriptEx(String ScriptInterface,StarBinBufClass BinBuf,String ModuleName,String WorkDirectory)
```

Executes script, WorkDirectory is work directory, may be "".

BinBuf should be allocated by \_NewBinBuf.

Static language does not support this function, such as java/c#.

ModuleName is name of the module, which may be "". This Parameter is only valid for lua and python.

## 3. **True/false, ErrorInfo = \_DoFile(ScriptInterface,FileName, WorkDirectory)/\_DoFileEx(ScriptInterface,FileName, WorkDirectory,ModuleName);**

```
java: Object[] _DoFile(String ScriptInterface,String FileName,String WorkDirectory)
```

```
java: Object[] _DoFileEx(String ScriptInterface,String FileName,String WorkDirectory,ModuleName)
```

If ScriptInterface is "", then default is lua. ScriptInterface may be lua,python,ruby,java,c# or other script language registered.

ModuleName is name of the module,may be "" or NULL. This Parameter is only valid for lua and python and java and csharp. For java and csharp, ModuleName is the init class name. **For example, the ModuleName is "com.srplab.www.test".**

ModuleName should not set to "cmd", case insensitive.

### 5.6.11 get peer ip address

**1. string IP =\_GetPeerIP (ClientID)**

```
java: String _GetPeerIP(int ClientID)
```

Called at client side, the input ClientID will be ignored.

Called at server side, ClientID indicates which client.

*5.6.12 get server ID at client***1. ServerID =\_GetServerID()**

```
java: int _GetServerID()
```

Valid at client side, which is used to get server ID for the communication with server.

*5.6.13 force to save service static data (valid at server side)***1. \_ForceToSaveStatic ()**

```
java: void _ForceToSaveStatic()
```

When service exits, then service static data is forced to save.

*5.6.14 remove expired data***1. \_ClearStatic (Days)**

```
java: void _ClearStatic(int Days)
```

*5.6.15 static data http download callback—valid at server side***1. \_RegServerWebDownFunction (Service\_Func);**

```
java: void _RegServerWebDownFunction(String FuncName)
```

function prototype is:

```
def Service_Func (self,uMes,FileName,MaxSize,CurSize)
```

```
function Service:Func (uMes,FileName,MaxSize,CurSize)
```

```
end
```

```
java/c#:
```

```
public void Func (int uMes,String FileName,long MaxSize,long CurSize)
{ }
```

uMes:

```
#define VSFILE_ONDOWNSTART      0    //---start download
#define VSFILE_ONDOWNPROGRESS  1    //---download progress
#define VSFILE_ONDOWNFINISH    2    //---finish
#define VSFILE_ONDOWNERROR     3    //---error
```

**2. \_RegServerWebDownFunction\_P (CallBackProc);**

for lua, ruby, python, this function is same as \_RegServerWebDownFunction

```

java: void _RegServerWebDownFunction_P(StarServiceWebDownInterface CallBackProc)

public interface StarServiceWebDownInterface{
    public void Invoke(Object Service,int uMes,String FileName,long MaxSize,long
CurSize);
}

c#: void _RegServerWebDownFunction_P(StarServiceWebDownInterface CallBackProc);

public delegate void StarServiceWebDownInterface(Object Service,int uMes,String
FileName,long MaxSize,long CurSize);

```

#### 5.6.16 get object string from Lua

##### 1. \_GetObjectFromLua(Str)

```
java: StarObjectClass _GetObjectFromLua(String Str)
```

The format of String is as “Service.DriveClass”.

#### 5.6.17 Pack static data

##### 1. \_PackStaticData()

```
java: void _PackStaticData()
```

Removes redundant space of static data file.

#### 5.6.18 Xml file

##### 1. bool = \_XmlToSysRootItem (SXml,DataPath, SegmentName,PrintFunc=None)

```
java: boolean _XmlToSysRootItem(StarSXmlClass SXml,String DataPath,String
SegmentName,String PrintFuncName)
```

def PrintFunc(Info) :

Print( Info)

##### 2. bool = \_XmlToSysRootItem\_P(SXml,DataPath, SegmentName,CallBackProc)

for lua, ruby, python, this function is same as \_XmlToSysRootItem

```
java: boolean _XmlToSysRootItem_P(StarSXmlClass SXml,String DataPath,String
SegmentName,StarServiceInfoInterface CallBackProc)
```

```

public interface StarServiceInfoInterface{
    public void Invoke(Object Service,String Info);
}

```

```

c#: bool _XmlToSysRootItem_P(StarSXml Class SXml, String DataPath, String
SegmentName, StarServiceInfoInterface CallBackProc);

```

```
public delegate void StarServiceInfoInterface(Object Service, String Info);
```

3. **bool = \_XmlToObject (SXml, Object/SystemRootItem, QueueName, DataPath, SegmentName, PrintFunc=None)**

```
java: boolean _XmlToObject(StarSXmlClass SXml, Object Obj, String QueueName, String DataPath, String SegmentName, String PrintFuncName)
```

4. **bool = \_XmlToObject\_P (SXml, Object/SystemRootItem, QueueName, DataPath, SegmentName, CallbackProc)**

for lua, ruby, python, this function is same as \_XmlToObject

```
java: boolean _XmlToObject_P(StarSXmlClass SXml, Object Obj, String QueueName, String DataPath, String SegmentName, StarServiceInfoInterface CallBackProc)
```

```
public interface StarServiceInfoInterface{
    public void Invoke(Object Service, String Info);
}
```

```
c#: bool _XmlToObjcet_P(StarSXmlClass SXml, Object Obj, String QueueName, String DataPath, String SegmentName, StarServiceInfoInterface CallBackProc);
```

```
public delegate void StarServiceInfoInterface(Object Service, String Info);
```

5. **bool = \_ServiceToXml (SXml, PassWord, DataPath, bool CFunctionFlag, bool OutputObjectID, PrintFunc=None)**

```
java: boolean _ServiceToXml(StarSXmlClass SXml, String PassWord, String DataPath, boolean CFunctionFlag, boolean OutputObjectID, String PrintFuncName)
```

6. **bool = \_ServiceToXml\_P (SXml, PassWord, DataPath, bool CFunctionFlag, bool OutputObjectID, CallbackProc)**

for lua, ruby, python, this function is same as \_ServiceToXml

```
java: boolean _ServiceToXml_P(StarSXmlClass SXml, String PassWord, String DataPath, boolean CFunctionFlag, boolean OutputObjectID, StarServiceInfoInterface CallBackProc)
```

```
public interface StarServiceInfoInterface{
    public void Invoke(Object Service, String Info);
}
```

```
c#: bool _ServiceToXml_P(StarSXmlClass SXml, String PassWord, String DataPath, bool CFunctionFlag, bool OutputObjectID, StarServiceInfoInterface CallBackProc);
```

```
public delegate void StarServiceInfoInterface(Object Service, String Info);
```

7. **bool = \_SysRootItemToXml (SXml, SysRootItemName, DataPath, bool CFunctionFlag, bool OutputObjectID, PrintFunc=None)**

```
java: boolean _SysRootItemToXml(StarSXmlClass SXml, String SysRootItemName, String DataPath, boolean CFunctionFlag, boolean OutputObjectID, String PrintFuncName)
```

8. **bool = \_SysRootItemToXml\_P (SXml, SysRootItemName, DataPath, bool CFunctionFlag, bool OutputObjectID, CallbackProc)**

for lua, ruby, python, this function is same as \_SysRootItemToXml

```
java: boolean _SysRootItemToXml_P(StarSXmlClass SXml, String SysRootItemName, String
```

```
DataPath,boolean      CFunctionFlag,boolean      OutputObjectID,StarServiceInfoInterface
CallBackProc)
```

```
public interface StarServiceInfoInterface{
    public void Invoke(Object Service,String Info);
}
```

```
c#: bool _SysRootItemToXml_P(StarSXmlClass SXml, String SysRootItemName, String
DataPath, bool CFunctionFlag, bool OutputObjectID, StarServiceInfoInterface
CallBackProc);
```

```
public delegate void StarServiceInfoInterface(Object Service, String Info);
```

9. **bool = \_ObjectToXml (SXml,Object,DataPath, bool CfunctionFlag, bool OutputObjectID ,PrintFunc=None)**

```
java: boolean _ObjectToXml(StarSXmlClass SXml,Object Obj,String DataPath,boolean
CfunctionFlag,boolean OutputObjectID,String PrintFuncName)
```

```
function Service:PrintFunc(Info)
end
```

```
java/c#:
public void PrintFunc(String Info)
{
}
```

10. **bool = \_ObjectToXml\_P (SXml,Object,DataPath, bool CfunctionFlag, bool OutputObjectID ,CallBackProc)**

for lua, ruby, python, this function is same as \_ObjectToXml

```
java: boolean _ObjectToXml_P(StarSXmlClass SXml,Object Obj,String DataPath,boolean
CfunctionFlag,boolean OutputObjectID,StarServiceInfoInterface CallBackProc)
```

```
public interface StarServiceInfoInterface{
    public void Invoke(Object Service,String Info);
}
```

```
c#: bool _ObjectToXml_P(StarSXmlClass SXml, Object Obj, String DataPath, bool
CfunctionFlag, bool OutputObjectID, StarServiceInfoInterface CallBackProc);
```

```
public delegate void StarServiceInfoInterface(Object Service, String Info);
```

### 5.6.19 atomic function

1. **AtomicDepend= \_CreateAtomicDepend (DependServiceName)**

```
java: long _CreateAtomicDepend(String DependServiceName)
```

2. **AtomicMacro= \_CreateAtomicMacro (MacroName,MacroType)**

```
java: long _CreateAtomicMacro(String MacroName,int MacroType)
```



MacroType = 0 Integer MacroType = 1 float MacroType = 2 string

3. **AtomicMacroItem= \_CreateAtomicMacroItem (MacroObject, MacroItemName,string MacroItemValue)**

```
java: long _CreateAtomicMacroItem(long MacroObject,String MacroItemName,String MacroItemValue)
```

4. **AtomicModule= \_CreateAtomicModule (ModuleName,ModuleType,ObjectID)**

```
java: long _CreateAtomicModule(String ModuleName,int ModuleType,String ObjectID)
```

5. **AtomicModule= \_CreateAtomicEditModule (ModuleName,ObjectID)**

```
java: long _CreateAtomicEditModule(String ModuleName,String ObjectID)
```

6. **AtomicStruct= \_CreateAtomicStruct (StructName,StructCaption,ObjectID)**

```
java: long _CreateAtomicStruct(String StructName,String StructCaption,String ObjectID)
```

7. **AtomicSysRootItem= \_CreateAtomicSysRootItem (SysRootItemName,DependSysRootItem)**

```
java: long _CreateAtomicSysRootItem(String SysRootItemName,String DependSysRootItem)
```

8. **AtomicObject= \_CreateAtomicObject (ParentAtomicObject, ParentQueueName, AtomicClassObject, ObjectName,ObjectID )**

```
java: long _CreateAtomicObject(long ParentAtomicObject,String ParentQueueName,long AtomicClassObject,String ObjectName,String ObjectID )
```

If ParentObject is service item, then ParentQueueName may be nil.

9. **AtomicAttribute= \_CreateAtomicAttachAttribute (AtomicObject, AttributeName, Caption, int Type, int StaticID, int SyncFlag, int CreateFlag, int NotifyFlag, int EditType, int EditControl, int EditReadOnly, string Default,string Desc)**

```
java: long _CreateAtomicAttachAttribute(long AtomicObject,String AttributeName,String Caption,int Type,int StaticID,int SyncFlag,int CreateFlag,int NotifyFlag,int EditType,int EditControl,int EditReadOnly,String Default,String Desc)
```

Attribute type supported is:

TYPE\_BOOL :

TYPE\_INT8 :

TYPE\_UINT8 :

TYPE\_INT16 :

TYPE\_UINT16 :

TYPE\_INT32 :

TYPE\_UINT32 :

TYPE\_FLOAT :

TYPE\_LONG :

TYPE\_ULONG :

TYPE\_CHAR :

TYPE\_COLOR :

TYPE\_RECT :

TYPE\_FONT :

TYPE\_TIME :

TYPE\_STRUCT :

a. "SyncFlag",TYPE\_UINT8,default is 0 note: 0 global, 1 local

b. "CreateFlag",TYPE\_UINT8,default is 0 note: 0 do not need, 1 need

- c. "NotifyFlag",TYPE\_UINT8,default is 0 note:1 notify before change, 2 notify after change, or both.
- d. "StaticID",TYPE\_ULONG,default is 0
- e. "EditType", TYPE\_UINT8,default is 0 note:0 normal edit, 1 combobox, 3 button ,4 check box, 5 can not edit, 6 mask edit
- f. "EditControl", TYPE\_UINT8,default is 0 note:0 none, 1 display callback
- g. "EditReadOnly", TYPE\_UINT8,default is 0 note:0can edit, 1 readonly
- h. "ComboBox",string,default is "", or indicates the macro name.

**10. AtomicAttribute= \_CreateAtomicAttribute (AtomicObject, AttributeName, Caption, Type, StaticID, SyncFlag, CreateFlag, NotifyFlag, EditType, EditControl, EditReadOnly, Default,Desc)**

```
java: long _CreateAtomicAttribute(long AtomicObject,String AttributeName,String Caption,int Type,int StaticID,int SyncFlag,int CreateFlag,int NotifyFlag,int EditType,int EditControl,int EditReadOnly,String Default,String Desc)
```

Attribute type supported is:

```
TYPE_BOOL :
TYPE_INT8 :
TYPE_UINT8 :
TYPE_INT16 :
TYPE_UINT16 :
TYPE_INT32 :
TYPE_UINT32 :
TYPE_FLOAT :
TYPE_LONG :
TYPE_ULONG :
TYPE_LONGHEX :
TYPE_ULONGHEX :
TYPE_VSTRING :
TYPE_PTR :
TYPE_STRUCT :
TYPE_CHAR :
TYPE_COLOR :
TYPE_RECT :
TYPE_FONT :
TYPE_TIME :
TYPE_UUID :
TYPE_STATICID :
```

**11. AtomicAttribute= \_CreateAtomicFuncRetAttribute (AtomicObject, int Type,Desc)**

```
java: long _CreateAtomicFuncRetAttribute(long AtomicObject,int Type,String Desc)
```

Return value type supported is:

```
TYPE_BOOL :
TYPE_INT8 :
```

TYPE\_UINT8 :

TYPE\_INT16 :

TYPE\_UINT16 :

TYPE\_INT32 :

TYPE\_UINT32 :

TYPE\_FLOAT :

TYPE\_LONG :

TYPE\_ULONG :

TYPE\_CHARPTR :

TYPE\_PARAPKGPTR :

TYPE\_PTR :

TYPE\_VOID :

TYPE\_INT8PTR :

TYPE\_UINT8PTR :

TYPE\_INT16PTR :

TYPE\_UINT16PTR :

TYPE\_INT32PTR :

TYPE\_UINT32PTR :

TYPE\_FLOATPTR :

TYPE\_LONGPTR :

TYPE\_ULONGPTR :

TYPE\_STRUCTPTR :

TYPE\_COLORPTR :

TYPE\_RECTPTR :

TYPE\_FONTPTR :

TYPE\_TIMEPTR :

TYPE\_UUIDPTR :

TYPE\_OBJPTR :

**12. AtomicAttribute= \_CreateAtomicFuncParaAttribute (AtomicObject, AttributeName, AttributeCaption ,Type,Desc)**

```
java: long _CreateAtomicFuncParaAttribute(long AtomicObject,String AttributeName,String  
AttributeCaption,int Type,String Desc)
```

Parameter type supported is:

TYPE\_BOOL :

TYPE\_INT8 :

TYPE\_UINT8 :

TYPE\_INT16 :

TYPE\_UINT16 :

TYPE\_INT32 :

TYPE\_UINT32 :

TYPE\_FLOAT :

TYPE\_LONG :  
TYPE\_ULONG :  
TYPE\_COLOR :  
TYPE\_RECT :  
TYPE\_FONT :  
TYPE\_TIME :  
TYPE\_UUID :  
TYPE\_PARAPKGPTR :

TYPE\_PTR :  
TYPE\_VOID :  
TYPE\_CHARPTR :

TYPE\_INT8PTR :  
TYPE\_UINT8PTR :  
TYPE\_INT16PTR :  
TYPE\_UINT16PTR :  
TYPE\_INT32PTR :  
TYPE\_UINT32PTR :  
TYPE\_FLOATPTR :  
TYPE\_LONGPTR :  
TYPE\_ULONGPTR :  
TYPE\_STRUCTPTR :  
TYPE\_COLORPTR :  
TYPE\_RECTPTR :  
TYPE\_FONTPTR :  
TYPE\_TIMEPTR :  
TYPE\_UUIDPTR :  
TYPE\_OBJPTR :

**13. AtomicAttribute= \_CreateAtomicStructAttribute** (AtomicObject, AttributeName, Caption, Type, Desc)

```
java: long _CreateAtomicStructAttribute(long AtomicObject,String AttributeName,String  
Caption,int Type,String Desc)
```

Attribute type of struct supported is:

TYPE\_BOOL :  
TYPE\_INT8 :  
TYPE\_UINT8 :  
TYPE\_INT16 :  
TYPE\_UINT16 :  
TYPE\_INT32 :  
TYPE\_UINT32 :  
TYPE\_FLOAT :  
TYPE\_LONG :  
TYPE\_ULONG :

TYPE\_CHAR :  
 TYPE\_COLOR :  
 TYPE\_RECT :  
 TYPE\_FONT :  
 TYPE\_TIME :  
 TYPE\_UUID :  
 TYPE\_MEMORY(Reserved) :

**14. Bool= \_SetAtomicAttributeLength (AtomicObject,Length)**

java: boolean \_SetAtomicAttributeLength(long AtomicObject,int Length)

used for char array.

**15. Bool= \_SetAtomicAttributeStruct (AtomicObject,AtomicStruct)**

java: boolean \_SetAtomicAttributeStruct(long AtomicObject,int AtomicStruct)

Set the attribute of the corresponding struct or class.

**16. Bool= \_SetAtomicAttributeCombobox (AtomicObject,MacroName)**

java: boolean \_SetAtomicAttributeCombobox(long AtomicObject,String MacroName)

Set attribute macro.

**17. Bool= \_SetAtomicAttributeSyncFlag (AtomicObject,int SyncFlag)**

java: boolean \_SetAtomicAttributeSyncFlag(long AtomicObject,int SyncFlag)

Set attribute sync : type ==0 sync attribute, ==1 local attribute.

**18. Bool= \_SetAtomicObjectSyncGroup (AtomicObject,int SyncGroup)**

java: boolean \_SetAtomicObjectSyncGroup(long AtomicObject,int SyncGroup)

set object syncgroup.

**19. Bool= \_SetAtomicObjectAttribute (AtomicObject,SysEvent(bool), int SpecialEvent,int ActiveCmd, int SaveFlag)**

java: boolean \_SetAtomicObjectAttribute(long AtomicObject,boolean SysEvent,int SpecialEvent,int ActiveCmd, int SaveFlag)

set object attribute

i. "SysEvent", TYPE\_BOOL,default is false

j. "SpecialEvent", TYPE\_UINT8,default is 0

#define VSSYSEVENT\_PROCESS\_TICKET 0x0001 //---10msTicket event

#define VSSYSEVENT\_PROCESS\_FRAMETICKET 0x0002 //---service frame pulse event.

#define VSSYSEVENT\_PROCESS\_IDLE 0x0004 //---program Idle event

#define VSSYSEVENT\_PROCESS\_APPACTIVE 0x0008

#define VSSYSEVENT\_PROCESS\_APPDEACTIVE 0x0010

#define VSSYSEVENT\_PROCESS\_SERVICEACTIVE 0x0020

#define VSSYSEVENT\_PROCESS\_SERVICEDEACTIVE 0x0040

k. "ActiveCmd", TYPE\_UINT8,default is 0

#define VACTIVE\_ALONE 0 //---Activated or deactivated through command.

#define VACTIVE\_FOLLOW 1 //---Activated or deactivated with parent object.

If parent object is service item, then the object will be activated automatically

//--The following two commands is dynamic, can not be saved.

1. "SaveFlag", TYPE\_UINT8, default is list below.

```
#define VSSTATIC_SAVE          0    ///---save global and static object.
#define VSSTATIC_CLIENTSAVE    1    ///---Saved at client, for client dynamic object
    and global dynamic object
#define VSSTATIC_NONE          2    ///---do not save, local object
```

**20. AtomicScript= \_CreateAtomicScript (AtomicObject, ScriptName,ObjectID, Desc, ScriptBuf )**

```
java: long _CreateAtomicScript(long AtomicObject,String ScriptName,String ObjectID,String
Desc,String ScriptBuf)
```

create object's script.

**21. AtomicFunction= \_CreateAtomicFunction (AtomicObject, FunctionName,ObjectID, Desc, CantOvl(bool), CallBack(bool),StdCallFlag(bool), GlobalFunctionFlag(bool))**

**AtomicFunction,ErrorInfo= \_CreateAtomicFunctionEx (AtomicObject, FunctionName,ObjectID, Desc, CantOvl(bool), CallBack(bool),Type(int) ,StdCallFlag(bool), GlobalFunctionFlag(bool))**

```
java: long _CreateAtomicFunction(long AtomicObject,String FunctionName,String
ObjectID,String Desc,boolean CantOvl,boolean CallBack,boolean StdCallFlag,boolean
GlobalFunctionFlag)
```

```
java: Object[] _CreateAtomicFunctionEx(long AtomicObject,String FunctionName,String
ObjectID,String Desc,boolean CantOvl,boolean CallBack,String Type,boolean
StdCallFlag,boolean GlobalFunctionFlag)
```

create object's function

Type is function prototype. For example: "VS\_CHAR \*GetBackImg(VS\_INT32 IndexX,VS\_INT32 IndexY)"

**22. AtomicFunction= \_CreateAtomicLuaFunction(AtomicObject, LuaFunctionName,ObjectID, Desc)**

```
java: long _CreateAtomicLuaFunction(long AtomicObject,String LuaFunctionName,String
ObjectID,String Desc)
```

create lua function

**23. AtomicFunction= \_CreateAtomicOvlFunction (AtomicObject, FunctionName, OriginFunctionName,ObjectID, Desc, CantOvl(bool))**

```
java: long _CreateAtomicOvlFunction(long AtomicObject,String FunctionName,String
OriginFunctionName,String ObjectID,String Desc,boolean CantOvl)
```

create object's overloading function,OriginFunctionName is the function defined in parent class.

**24. AtomicInEvent= \_CreateAtomicInEvent (AtomicObject, InEventName,ObjectID , OutEventName)**

```
java: long _CreateAtomicInEvent(long AtomicObject,String InEventName,String
ObjectID,String OutEventName)
```

create object's event,OutEventName is the output event defined in parent class

**25. AtomicOutEvent= \_CreateAtomicOutEvent (AtomicObject, OutEventName,ObjectID, Desc, DynamicFlag(bool))**

```
java: long _CreateAtomicOutEvent(long AtomicObject,String OutEventName,String
ObjectID,String Desc,boolean DynamicFlag)
```

create object's output event

## 26. AtomicSysRootItem= \_GetAtomicSysRootItem(SysRootItemName)

```
java: long _GetAtomicSysRootItem(String SysRootItemName)
```

gets service item atomic object.

## 27. AtomicObject= \_GetAtomicObjectEx (ParentAtomicObject, ObjectName)

```
java: long _GetAtomicObjectEx(long ParentAtomicObject,String ObjectName)
```

gest atomic object

## 28. ID= \_GetAtomicID (AtomicObject)

```
java: String _GetAtomicID(long AtomicObject)
```

gets ID of atomic object

## 29. AtomicObject= \_ObjectToAtomic (Object)

```
java: long _ObjectToAtomic(StarObjectClass Obj)
```

Converts normal object to atomic object

## 30. Object= \_AtomicToObject(AtomicObject)

```
java: StarObjectClass _AtomicToObject(long AtomicObject)
```

Converts atomic object to normal object

## 31. Create simple object, struct, object attribute and function

AtomicObject,ErrorInfo

=

\_CreateAtomicObjectSimple(SysRootItemName,ObjectName,Attribute,**ObjectID**)

```
java: Object[] _CreateAtomicObjectSimple(String SysRootItemName,String
ObjectName,String Attribute,String ObjectID)
```

If SysRootItemName does not exist, then create a new one.

Attribute is string of the attribute, such as "VS\_CHAR aaa;VS\_INT32 bbb;" it may be set to nil;

Local attribute should add prefix "local", such as "local VS\_CHAR aaa;VS\_INT32 bbb;"

AtomicStruct,ErrorInfo = \_CreateAtomicStructSimple(StructName,Attribute,**ObjectID**)

```
java: Object[] _CreateAtomicStructSimple(String StructName,String Attribute,String
ObjectID)
```

Attribute is string of the attribute, such as "VS\_CHAR aaa;VS\_INT32 bbb;"

AtomicObject,ErrorInfo = \_CreateAtomicObjectAttributeSimple(AtomicObject,Attribut)

```
java: Object[] _CreateAtomicObjectAttributeSimple(long AtomicObject,String Attribut)
```

Attribute is string of the attribute, such as "VS\_CHAR aaa;VS\_INT32 bbb;" Local attribute should add prefix "local", such as "local VS\_CHAR aaa;VS\_INT32 bbb;"

AtomicFunction,ErrorInfo

=

\_CreateAtomicFunctionSimple(AtomicObject,FunctionName,Attribute,**ObjectID**,StdCallFlag(**bool**),GlobalFunctionFlag(**bool**))

```
java: Object[] _CreateAtomicFunctionSimple(long AtomicObject,String FunctionName,String
Attribute,String ObjectID,boolean StdCallFlag,boolean GlobalFunctionFlag)
```

Attribute is the function prototype. For example: "VS\_CHAR \*GetBackImg(VS\_INT32 IndexX,VS\_INT32 IndexY)"

Attribute may be signature. Details refer to ScriptCall function of c++ interface.

For example:

Attribute: "VS\_INT32;VS\_FLOAT;" may be use string "if"

Attribute: "VS\_INT32 Func(VS\_FLOAT)" may be use string "(f)i"

## 32. attach sharelib

bool = \_AtomicAttach(long atomicobject, ShareLibName )

java: boolean \_AtomicAttach(long AtomicObject,String ShareLibName)

The function should be called after all function definition has been created.

\_AtomicAttach(atomicobject,"user32.dll");

### 5.6.20 output service header and skeleton file

#### 1. bool,errorInfo= \_ExportModule (XmlCfgFile)

java: Object[] \_ExportModule(String XmlCfgFile)

The format of XmlCfgFile is as follow:

```
<?xml version="1.0" standalone="no" encoding="utf-8" ?>
<ExportModuleInfo ExportModuleDir=D:\Work\VS_NEW\Examples>
  <SRPFSEngineBasicBCeditModule>
    <DriveClass/>
  </SRPFSEngineBasicBCeditModule>
  <SRPFSEngineBCModule>
    <FileToolClass/>
  </SRPFSEngineBCModule>
  <SRPFSEngineBasicModule>
    <DriveClass/>
    <DirectoryClass/>
    <FileClass/>
  </SRPFSEngineBasicModule>
</ExportModuleInfo>
```

### 5.6.21 object's EditLog/Checkpoint *[Reserved]*

With the checkpoint mechanism to provide high reliability.

#### 1. \_SetLog (Object/SysRootItem,true/false)

java: void \_SetLog(Object Arg,boolean BeginOrEnd)

Set the object or service item to start to log, then any changes of object attribute, child object created or deleted, child object attribute change will be recorded in log file, which can be recover by ApplyLog.

#### 2. \_SetLogFile(FileName)

java: void \_SetLogFile(String FileName)



Set Log file name which default path is service directory.

**3. FileName = \_GetLogFile()**

```
java: String _GetLogFile()
```

Get log file name.

**4. \_ClearLog()**

```
java: void _ClearLog()
```

Clear Log content.

**5. bool = \_ApplyLog()**

```
java: boolean _ApplyLog()
```

Recover.

### 5.6.22 Authorize

**1. bool= \_IsServiceRegistered ()**

```
java: boolean _IsServiceRegistered()
```

Whether service is registered. Do not call the function frequently.

**2. void= \_CheckPassword(true/false)**

```
java: boolean _CheckPassword(boolean Flag)
```

If check password is set to false, then when \_GetService of ServiceGroupObject is called, the cle will not check user password.

### 5.6.23 ShareLibrary functions

**1. String \_FisrtShareLib()**

return first share library name loaded.

**2. String \_NextShareLib()**

return first share library name loaded.

**3. long \_GetShareLib(String ShareLibName)**

return handle of share library.

**4. void \_FreeShareLib(long ShareLibHandle)**

free share library loaded before.

### 5.6.24 Object Group Management

**1. int \_NewGroup()**

Create an object group

**2. void \_FreeGroup(GroupID)**

```
java: void _FreeGroup(int GroupID)
```

Delete an object group

**3. int \_GroupAdd(GroupID,object)**

```
java: int _GroupAdd(int GroupID,StarObjectClass object)
```

Add an object to group, the return value is RefID;

**4. object \_GroupGet (GroupID,RefID)**

```
java: StarObjectClass _GroupGet(int GroupID,int RefID)
```

Get object in the group by RefID;

**5. void \_GroupRemove(GroupID,RefID)**

```
java: void _GroupRemove(int GroupID,int RefID)
```

Remove object from group

**6. void \_GroupRemoveEx(GroupID,object)**

```
java: void _GroupRemoveEx(int GroupID,StarObjectClass object)
```

Remove object from group

**7. void \_GroupClear(GroupID,FreeObjectFlag)**

```
java: void _GroupClear(int GroupID,boolean FreeObject)
```

Clear the group, if FreeObject == true, then objects in the group will be freed.

### 5.6.25 Get Control Service

**1. Service = \_GetControlService()**

```
java: StarServiceClass _GetControlService()
```

This function may be return nil/None,null;

### 5.6.26 Raw Object Interface

**1. boolean \_LoadRawModule(String ScriptInterface,String ModuleName,String FileOrString,boolean IsString)**

The function is same as the one in ServiceGroupClass.

**2. boolean \_LoadRawModuleEx(String ScriptInterface,String ModuleName,StarObjectClass object )**

This function is valid for csharp, object should be raw assembly object.

**3. Object \_NewScriptRawType(int RawType)**

The function is reserved.

**4. StarObjectClass \_NewRawProxy(String ScriptInterface,Object AttachObject,String AttachFunction,String ProxyInfo,int ProxyType)**

AttachObject: is cle object or instance of script class, which contains functions callback from the proxy.

AttachFunction: is callback function name

ProxyInfo : proxy info, varies with script interface

ProxyType : 1, the callback function run in main thread, which inits CLE. Others are reserved, set to 0;

**for c/c++ :**

The function is invalid.

**for lua :**

The function returns a callable lua userdata, which can be called from lua script.

ProyInfo: will be ignored

AttachObject may be input lua function for c# delegate, or function table for java interface. In this case, AttachFunction will be ignored. For example:

```
close_cb = {  
    }  
function close_cb.windowClosing(ev)  
    print("close")  
    frame:setVisible(false)  
    frame:dispose()  
end  
  
function close_cb.windowActivated(ev)  
    print("act")  
end  
jproxy = Service:_NewRawProxy("java",close_cb,"","java.awt.event.WindowListener",0);
```

or

```
exit_cb = {  
    actionPerformed = function(ev)  
        print("exit")  
        frame:setVisible(false)  
        frame:dispose()  
    end  
}  
jproxyb = Service:_NewRawProxy("java",exit_cb,"","java.awt.event.ActionListener",0);
```

**for python :**

The function returns a callable python object. which can be called from python script.

ProyInfo: will be ignored

AttachObject may be python function, in this case, AttachFunction will be ignored.

**for ruby :**

The function returns a method. which can be called from ruby script.

ProyInfo: will be ignored

AttachObject may be ruby function, in this case, AttachFunction will be ignored.

**for java :**

The function returns a proxy of java interface.

ProyInfo: java interfaces, separated with ','

AttachFunction will be ignored

**for csharp :**

The function returns a proxy of c# delegate.

ProyInfo: c# delegate name

note : for wp8, only delegates : void XXX(object arg) and void XXX(object arg1,object arg2) are supported

## 5. String \_CreateRawProxyCode(String ScriptInterface,String NewPackageName,String Imports,String NewClassName,String BaseClass,String Methods,String Interface)

This function is used to create code of extend class, valid for java and c#.

### for java :

NewPackageName : is package name

Imports : are libraries to be import, separated by “,” , may be NULL

NewClassName : new class name to be created

BaseClass : name of base class, may be NULL

Methods : public methods to be override, may be NULL

Interface : name of interfaces, separated by “,” , may be NULL

for example:

```
code
Service:_CreateRawProxyCode("java","", "testextend.*", "ExtendBaseClass", "testextend/BaseClasses", "getstr, getmyclass, getstres, getmyclasses, getmyobjectes, getmyobject", "testextend/ICallBack");
```

### for csharp :

NewPackageName : is namespace

Imports : is libraries to be import, separated by “,” , may be NULL

NewClassName : new class name to be created

BaseClass : name of base class,, may be NULL

Methods : public methods to be override, may be NULL

Interface : name of interfaces, separated by “,” , may be NULL

for example:

```
code
Service:_CreateRawProxyCode("csharp","", "testextend", "ExtendBaseClass", "testextend.BaseClasses", "getstr, getmyclass, getstres, getmyclasses, getmyobjectes, getmyobject", "testextend.ICallBack");
```

### for python/ruby :

NewPackageName : will be ignored

Imports : is libraries to be import, separated by “,” , may be NULL

NewClassName : new class name to be created

BaseClass : name of base class,

Methods : methods to be override, should use fullname with parameter and seperatedby ‘;’

Interface : will be ignored

for example:

```
code
service._CreateRawProxyCode("python","", "", "ExtendBaseClass", "BaseClass", "__init__(self);get
str(self,val)", "");
```

#### 6. **StarObjectClass \_ImportRawContext(String ScriptInterface,String ContextName,boolean IsClass,String ContextInfo)**

The function is the combination of two functions :

```
Obj = Service._New()
```

```
Obj._AttachRawContext(ScriptInterface,ContextName,IsClass,ContextInfo)
```

#### 7. **StarObjectClass \_ImportRawObject(Object RawObject,boolean IsClass)**

If the rawobject has been imported, then the function returns old cle object. Otherwise, The function alloc a new cle object, and assign raw object to it. In this case, it is the combination of two functions :

```
Obj = Service._New()
```

```
Obj._AttachRawObject(RawObject,IsClass)
```

### 5.6.27 Get Last Error

#### 1. **int \_GetLastError()**

#### 2. **String \_GetLastErrorInfo()**

### 5.6.28 Attribute Get

#### 1. **value = \_Get (Name)**

## 5.7 Service item object

### 5.7.1 Attribute

#### 1. **SrvItem.\_Name**

Service item name

#### 2. **SrvItem.\_Service**

The corresponding service of the service item

### 5.7.2 Functon

#### 1. **BOOL=\_IsSync () [valid at client]**

```
java: boolean _IsSync()
```

Whether service item is sync.

#### 2. **BOOL=\_WaitSync () [valid at client]**

```
java: boolean _WaitSync()
```

Waiting service item becomes sync. Return False indicates failed, and True indicates success.

3. **int \_GetGroupSyncStatus(GroupIndex)**

```
java: int _GetGroupSyncStatus(int GroupIndex)
```

Get syncgroup status.

4. Active set table=\_GetActiveSet ()

```
java: Object[] _GetActiveSet()
```

Get active set of servie item. Return None for not existing.

5. **\_SetActiveSet (active set)**

```
java: void _SetActiveSet(Object...Arg)
```

Set service item active set.

6. **\_SetClientActiveSet (ClientID, active set)**

```
java: void _SetClientActiveSet(int ClientID,Object...Arg)
```

Set active set of service item for client side. If the service item is not active at client, then it will be activated automaticly.

ClientID can be obtained from Object.\_ClientID.

7. **\_QueryFirstGroupObject(GroupIndex)**

```
java: StarObjectClass _QueryFirstGroupObject(int GroupIndex)
```

Get first object in the special group of the service item

GroupIndex should not be set to zero;

8. **\_QueryNextGroupObject()**

```
java: StarObjectClass _QueryNextGroupObject()
```

Get next object in the special group of the service item

### 5.7.3 Attribute Get or Set

1. **value = \_Get (Name)**

2. **\_Set (Name,value)**

Name should be “\_OnClientToSync”

### 5.7.4 Callback function

1. **\_OnClientToSync(ClientID,SyncGroupIndex)**

When client becomes sync, the callback function will be called at server side.

```
def SrvItem_OnClientToSync( self,ClientID,SyncGroupIndex ) :
```

```
SrvItem._OnClientToSync = SrvItem_OnClientToSync
```

```
function SrvItem:_OnClientToSync( ClientID,SyncGroupIndex )
```

```
end
```

```
java/c#:
```

```
public void _OnClientToSync(int ClientID,int SyncGroupIndex )  
{  
}  
}
```

**2. \_RegClientToSync(CallBack)**

```
java: void _RegClientToSync(String FunctionName)
```

**3. \_RegClientToSync\_P(CallBack)**

for lua, ruby, python, this function is same as \_RegClientToSync

```
java: void _RegClientToSync_P(StarServiceItemClientToSyncInterface CallBackProc)
```

```
public interface StarServiceItemClientToSyncInterface{
    public void Invoke(Object ServiceItem,int ClientID,int SyncGroupIndex);
}
```

```
c#: void _RegClientToSync_P(StarServiceItemClientToSyncInterface CallBackProc);
```

```
public delegate void StarServiceItemClientToSyncInterface(Object ServiceItem, int
ClientID, int SyncGroupIndex);
```

*5.8 Object' s predefined attribute and function***1. Object.\_Service**

The corresponding service of the object

**2. Object.\_Parent**

Parent object may be service item object, which can be distinguished by function SrvGroup.\_IsObject.If the function returns True, then it is normal object, or else is service item object

**3. Object.\_Index**

Get queue attribute name of parent object, which the object belongs to

**4. Object.\_Order**

Get order of the object in queue of parent object. Returns 0 for no order. Order starts from 1.

**5. Object.\_Class**

Get class of the object

**6. Object.\_ThisService**

Is the object of this service[Returns True is this service,False other service]

**7. Object.\_ActiveService**

Is the object of active service[Returns True is True active service,False, not active service]

**8. Object.\_ClassID**

Get class ID of the object, is string.

**9. Object.\_ID**

Get object ID,is string.

**10. Object.\_Next**

Get next object in the same queue of the parent

**11. Object.\_Prev**

Get previous object in the same queue of the parent

**12. Object.\_NextEx**

- Get next object with same name
13.     **Object.\_PrevEx**  
Get previous object with same name.
14.     **Object.\_Name**  
object name[read and write]
15.     **Object.\_ClientID**  
Object's ClientID(the value represents client machine ID at server side)
16.     **Object.\_SysRootItem**  
The service item of the object belongs to
17.     **Object.\_SyncStatus**  
Object's sync status
18.     **Object.\_SyncGroup**  
Object syncgroup, can be read or write  
For write:  
For global dynamic object, static object, or client dynamic object, the function is only valid at server. Object's syncgroup starts from 1.  
Object must belong to one service item, otherwise the function takes no effect.  
If parent object syncgroup has been set, then child object syncgroup can not be set , which is to avoid sync problem if they are different.
19.     **Bool = Object.\_IsLocalControl**  
Return True if object is under local control, otherwise is under server control.  
For static object, global dynamic object or client object, the function returns True at server side.
20.     **Bool = Object.\_IsRemoteCreate**  
If returns true, then the object is created by local side, or else is created by remote side.
21.     **Bool = Object.\_IDInParent[read or write]**  
Object tag in parent object, 0 is reserved.
22.     **Object.\_Layer**  
Object layer of class, start from 1. If the layer is 1, then the object has no parent class.
23.     **Object.\_RemoteID**  
Client ID of remotecall, valid at server. The value is set when client starts remotecall. The remotecall function can use this value to determine which client initiates the remotecall.
24.     **Object.\_RemoteSourceTag**  
Source tag, is used to determine the source type.
25.     **Object.\_RemotePrivateTag**  
Client private tag, which is used to determine the legality of the client.
26.     **Object.\_RemoteCallID**  
Remotecall ID, which identify a remotecall.
27.     **Bool = Object.\_IsRemoteCall**  
Whether object is in remotecall process.
28.     **Bool = Object.\_RemoteCallName**  
Remotecall function name, used in response.
29.     **Object.\_AllocType**  
Object alloc type



**30. Object.\_SaveFlag**

can be read or write. Takes value from 0,1,2, corresponding to

SAVE\_NONE

SAVE\_LOCAL

SAVE\_GLOBAL

**31. Object.\_EditMode**

Can be read or write, the value is True or False, the value should be set when the object is edited.

**32. Object.\_WebServiceFlag**

The object is published for webservice. Returns true indicates published [read or write]

**33. Object.\_Super**

Only Used for call function of super class

**34. bool Object.\_ReturnRawFlag[r/w]**

\_ReturnRawFlag is valid for raw object which wraps lua and python and ruby object. In normal case, lua table and python tuple and ruby array is tried to be converted to parapg. But if ReturnRawFlag == true, then, lua table and python tuple and ruby array will be wrapped with CLE object.

**Function****1. \_V/\_F()**

print object attributes or lua functions

For Java/c#, should use \_V/\_F(null).

**2. \_V/\_F (FuncName)**

```
java: void _V(String Name)
```

```
java: void _F(String Name)
```

print description information of attribute or lua function

**3. \_E ()**

```
java: void _E()
```

print output event name and attribute.

**4. \_S()/\_S(ScriptName)**

```
java: void _S(String Name)
```

print script list or script contents

**5. \_NV()**

```
java: void _NV()
```

print object's name value paires.

**6. \_GetChild (ObjectName )**

```
java: StarObjectClass _GetChild(String Name)
```

Get child object

**7. \_GetChildByID (AttributeName,int IDInParent )**

```
java: StarObjectClass _GetChildByID(String AttributeName,int IDInParent)
```

Get child object based on IDInParent from special queue.

**8. \_FirstInst ( QueryRecord)**

```
java: StarObjectClass _FirstInst(StarQueryRecordClass QueryRecord)
```

get first instance.

**9. \_NextInst (QueryRecord)**

java: StarObjectClass \_NextInst(StarQueryRecordClass QueryRecord)

get next instance

**10. \_QueryClose (QueryRecord)**

java: void \_QueryClose(StarQueryRecordClass QueryRecord)

close query.

**11. Child,long Context =\_FirstActiveChild ()**

java: Object[] \_FirstActiveChild()

get first active child.

**12. Child, int Context =\_NextActiveChild (long Context)**

java: Object[] \_NextActiveChild(long Context)

get next active child.

**13. bool = \_IsInactiveSet ()**

java: boolean \_IsInactiveSet()

Whether object is in active set of it's service item.

**14. bool=\_IsInst ( Object )**

java: boolean \_IsInst(StarObjectClass Obj )

Input object is instance of the object or not.

**15. bool=\_IsDirectInst ( Object )**

java: boolean \_IsDirectInst(StarObjectClass Obj )

Input object is direct instance of the object or not.

**16. bool=\_IsChild (Object )**

java: boolean \_IsChild(StarObjectClass Obj )

Input object is child of the object.

**17. bool=\_IsThisClient ()**

java: boolean \_IsThisClient()

Object belongs to this client or not, valid at client side.

**18. \_SetPrivateValue (int ClassLayer,int Index,Value )**

java: void \_SetPrivateValue(int ClassLayer,int Index,Object Value )

Set object private value

**Note : Value is int for 32 bit version, and int64 for 64 bit version**

**19. Value =\_GetPrivateValue (ClassLayer,Index )**

java: Object \_GetPrivateValue(int ClassLayer,int Index )

Get object private value

**Note : Value is int for 32 bit version, and int64 for 64 bit version**

**20. \_InsertToSDT ( )**

java: void \_InsertToSDT( )

Insert object to service table

**21. \_DelFromSDT( )**

java: void \_DelFromSDT( )

Delete object from service table.

**22. Object = \_QueryFirstInstFromSDT (QueryRecord )**

java: StarObjectClass \_QueryFirstInstFromSDT(QueryRecord )

Query first instance of object from service table.

**23. Object = \_QueryNextInstFromSDT (QueryRecord )**

java: StarObjectClass \_QueryNextInstFromSDT(QueryRecord )

Query next instance of object from service table.

**24. \_ChangeParent (ParentObject,QueueAttrName )**

java: void \_ChangeParent(StarObjectClass ParentObject,String QueueAttrName )

Change parent object. **ParentObject may be object or service item. If it is service item, then QueueAttrName may be omitted.**

**Calling format:**

**Object: \_ChangeParent (ParentObject)**

**Object: \_ChangeParent (ParentObject, "QueueAttrName");**

**For global static,dynamic, or client object, the function is invalid at debug server side.**

**When change parent object, the object status will be reset. If object is active then the following event will be generated:**

ONDEACTIVATING,ONDEACTIVATE,ONDESTORY,ONCREATE,ONACTIVATING,ONACTIVATE.

**25. bool=\_ActiveCmd (int Cmd )**

java: boolean \_ActiveCmd(int Cmd )

Called at server, return value is bool type.

**26. Cmd = \_GetActiveCmd ()**

java: int \_GetActiveCmd()

Called at server, return object's active command

**27. bool=\_ActiveClient(int ClientID )**

java: boolean \_ActiveClient(int ClientID )

Called at server, active object at client

If ClientID equals to 0, then object is activated on all client.

**28. \_DeactiveClient (int ClientID )**

java: void \_DeactiveClient(int ClientID)

Called at server, deactive object at client,no return value.

If ClientID equals to 0, then the object is deactivated on all client.

**29. \_Active ( )**

java: boolean \_Active( )

activate object,no return value.

**30. \_Deactive ( )**

java: void \_Deactive( )

deactivate object,no return value.

**31. Bool = \_IsActive ()**

java: boolean \_IsActive()

whether object is active.

**32. Object=\_QueryFirstActiveInst (QueryRecord)**

java: StarObjectClass \_QueryFirstActiveInst()

get first active instance of object from current active service.

**33. Object=\_QueryNextActiveInst (QueryRecord)**

java: StarObjectClass \_QueryNextActiveInst()

get next active instance of object from current active service.

### 34. **long RefValue = \_RegEventFunction(InObject,EventName, Object\_Function)**

java: long \_RegEventFunction(StarObjectClass InObject,String EventName,String Object\_Function)

Register object's event process function, Function Object\_Function processes InObject.EventName event.

```
def Object_Function ( self,Event,Args...) :
    Print(self, Event )
    return True,... (...values)
```

```
function Object:Function ( Event,Args...)
    print(self, Event )
    return true,... (...values)
end
```

java/c#:

```
public Object[] _OnObjectEvent(StarObjectClass self, Hashtable Event, Args)
{
    return null; or Object[] {true,...};
}
```

Parameter event contains four values : \_SrcObject,\_DesObject,\_EventID, \_ServiceGroupID.  
Args depends on event definition of the object.

### 35. **long RefValue = \_RegEventFunction\_P(InObject,EventName, CallBackProc)**

for lua, ruby, python, this function is same as \_RegEventFunction

java: long \_RegEventFunction\_P(StarObjectClass InObject,String EventName,StarObjectEventProcInterface CallBackProc)

```
public interface StarObjectEventProcInterface{
    public Object[] Invoke(Object CleObject,Object CleEvent,Object[] EventParas);
}
```

c#: long \_RegEventFunction\_P(StarObjectClass InObject,String EventName,StarObjectEventProcInterface CallBackProc);

```
public delegate object[] StarObjectEventProcInterface(Object CleObject, Object CleEvent, object[] EventParas);
```

### 36. **\_UnRegEventFunction(InObject,EventName, long RefValue)**

java: void \_UnRegEventFunction(StarObjectClass InObject,String EventName, long RefValue)

Unregister object event process function.

### 37. **ReturnValue = \_ProcessEvent(EventName,...)**

java: Object[] \_ProcessEvent(String EventName,Object...Args)

Create object event,...is args.

Types of argments and return value may be string, integer,bool,or float.

### 38. **\_PostProcessEvent(EventName,...)**

java: void \_PostProcessEvent(String EventName,Object...Args)

Create object event,...is args, no return value.

Types of arguments may be string, integer,bool,or float.

### 39. **string EventID = \_EventID(EventName)**

```
java: String _EventID(String EventName)
```

Get ID of the event.

### 40. **TimerID= \_SetTimer (Ticket, Object. Func, Arg1, Arg2 )**

```
java: int _SetTimer(int Ticket,String Object_Func,int Arg1,int Arg2 )
```

Function should be defined within the object

Ticket is interval which unit is 10ms;

Calling format : a = Object.SetTimer(100,Func, 0,0 )

Args should be number, returns TimerID

```
def Object_Func(self,TimerID,Arg1,Arg2) :
```

```
function Object:Func(TimerID,Arg1,Arg2)
```

```
end
```

```
java/c#:
```

```
public void TimerFunc(StarObjectClass self,int TimerID,int Arg1,int Arg2)
{
}
```

### 41. **TimerID= \_SetTimer\_P (Ticket, Arg1, Arg2, CallBackProc)**

for lua, ruby, python, this function is same as \_SetTimer

```
java: int _SetTimer_P(int Ticket,int Arg1,int Arg2, StarObjectTimerInterface CallBackProc )
```

```
public interface StarObjectTimerInterface{
    public void Invoke(Object CleObject,int TimerID,int Arg1,int Arg2);
}
```

```
c#: int _SetTimer_P(int Ticket,int Arg1,int Arg2, StarObjectTimerInterface CallBackProc );
```

```
public delegate void StarObjectTimerInterface(Object CleObject, int TimerID, int
Arg1, int Arg2);
```

### 42. **\_KillTimer (TimerID)**

```
java: void _KillTimer(int TimerID)
```

TimerID is the ID of the timer.

### 43. **\_New(QueueAttrName,ParentObject, ObjectName,InitScript)/\_NewEx (ObjectID,QueueAttrName,ParentObject, ObjectName,InitScript)**

```
java: StarObjectClass _New(Object...Args)
```

```
java: StarObjectClass _NewEx(Object...Args)
```

ParentObject is parent object,QueueAttrName is queue name of the parent object. ParentObject may be object or service item. If it is service item, then QueueAttrName may be "".

Calling format:

```
Object._New("QueueAttrName",ParentObject)
```

```
Object._New(ParentObject)
```

Object.\_New() : no parent object

If there is no parent object, then the object is created under context of the active service. If the active service is changed, then the object will be freed by CLE.

InitScript is lua script used to assign attribute or perform object's function. Each sentence is separated by ';'.

For example :Object.\_New(“”,”attr1=value; attr2=value2; func();”)  
';' should not be omitted.

syntax format:

sentences are separated by ';'.

for each sentence,

If the first char is '\$', then the following '\$O' will be replaced by Object before executed.

For example:”\$O:\_Active();” is translated to Object:\_Active() .

If not, then the following rule takes effect.

If contains '=' , then translate to Object. XXXX

otherwise translate to Object: XXXX

\$O is not case sensitive.

#### 44. **\_NewGlobal(ClientID, QueueAttrName,ParentObject, ObjectName,InitScript)/\_NewGlobalEx(ObjectID,ClientID, QueueAttrName,ParentObject, ObjectName,InitScript)**

```
java: StarObjectClass _NewGlobal(Object...Args)
```

```
java: StarObjectClass _NewGlobalEx(Object...Args)
```

ParentObject is parent object, QueueAttrName is queue name of parent object. ClientID is ID of the client. ClientID should be set to 0 at client side and set to ID of client at server side. If the object is not created for special client, Client ID should also be set to 0. ParentObject may be object or service item. If it is service item, then QueueAttrName may be “”.

ClientID may be omitted.

Calling format:

Object.\_NewGlobal(“QueueAttrName”,ParentObject)

Object.\_NewGlobal(ParentObject)

Object.\_NewGlobal(ClientID,”QueueAttrName”,ParentObject)

Object.\_NewGlobal(ClientID,ParentObject)

The function is invalid at debug.

InitScript is lua script used to assign attribute or perform object's function. Each sentence is separated by ';'.

For example :Object.\_New(“”,”attr1=value; attr2=value2; func();”)  
';' should not be omitted.

syntax format:

sentences are separated by ';'.

for each sentence,

If the first char is '\$', then the following '\$O' will be replaced by Object before executed.

For example:”\$O:\_Active();” is translated to Object:\_Active() .

If not, then the following rule takes effect.

If contains '=' , then translate to Object. XXXX

otherwise translate to Object: XXXX

\$O\$ is not case sensitive.

```

45.      _NewClient(ClientID, QueueAttrName,ParentObject,
      ObjectName,InitScript))/_NewClientEx(ObjectID,ClientID,
      QueueAttrName,ParentObject, ObjectName,InitScript)

```

```
java: StarObjectClass _NewClient(Object...Args)
```

```
java: StarObjectClass _NewClientEx(Object...Args)
```

ParentObject is parent object, QueueAttrName is queue name of parent object. ClientID is the ID of the client. ClientID should be set to 0 at client side and set to ID of client at server side. ParentObject may be object or service item. If it is service item, then QueueAttrName may be "".

ClientID may be omitted.

Calling format:

```
Object._NewClient("QueueAttrName",ParentObject)
```

Object.\_NewClient (ParentObject)

```
Object._NewClient (ClientID,"QueueAttrName",ParentObject)
```

Object.\_NewClient (ClientID,ParentObject)

The function is invalid at debug.

InitScript is lua script used to assign attribute or perform object's function. Each sentence is seperated by ';'.

For example `:Object._New(““,”attr1=value; attr2=value2; func();”)`

’;’ should not be omitted.

syntax format:

sentences are seperated by ';

for each sentence,

If the first char is '\$', then the following '\$O' will be replaced by Object before executed.

For example: "\$\$O:\_Active();" is translated to `Object:_Active()`.

If not, then the following rule takes effect.

If contains '=', then translate to Object. XXXX

otherwise translate to Object: XXXX

\$O\$ is not case sensitive.

**46.**`_Change(“AttrName”,...)`

```
java: void _Change(String AttrName,Object Arg)
```

Chane object attribute at local

#### 47. MarkChange(“AttrName”)

```
java: void MarkChange(String AttrName)
```

For attribute change, it will be synchronized to client at next service frame. [for global object, static object, and client object], valid at server.

48.      **bool = WaitMalloc()**

```
java: boolean _WaitMalloc()
```

Called at client side to wait server to confirm the allocation of global or client object. If the return value is false, then the object is not permitted by server and has been freed or the

connection to server is closed.

#### 49. **\_Copy(SrcObject)**

```
java: void _Copy(StarObjectClass SrcObject)
```

The function does not copy pointer and local attribute.

The two objects should belong to the same class.

#### 50. **\_Free()**

```
java: void _Free()
```

Free object. For global, static or client object, the function is invalid at debug server side.

At client side, the object is not freed intermidately because the request will be sent to server.

The script object must not be used after the function is called.

#### 51. **\_Dispose()**

```
java: void _Dispose()
```

This function clears the script context. If the object is created by other scripts, this function does not free the object. This function is for speed up the gc procedures.

The script object must not be used after the function is called.

#### 52. **\_DeferFree()**

```
java: void _DeferFree()
```

Free object. For global, static or client object, the function is invalid at debug.

The object is freed at next Ticket. At client side, it is same as FreeObject.

#### 53. **Bool=\_IsInFree()**

```
java: boolean _IsInFree()
```

Returns True if object is being freed.

#### 54. **long RefValue =\_RegFileCallBack(Object.\_RegFileCallBack)**

```
java: long _RegFileCallBack(String Object_RegFileCallBack)
```

The callback function should be defined within the object. The prototype is as follows:

```
def function (self, uMsg, DataFile, ReceiveOrSendOffset,
DataSize, FileName Or Object, UniqueDataUnitID,Version ) :
```

```
Object:_RegFileCallBack( function (self, uMsg, DataFile, ReceiveOrSendOffset,
DataSize, FileName Or Object, UniqueDataUnitID,Version ) ... end )
```

```
java/c#:
```

```
public void Obj_RegFileCallBack(StarObjectClass self,int uMsg, boolean
DataFile,int ReceiveOrSendOffset,int DataSize, Object FileName_Object, int
UniqueDataUnitID,String Version )
{
}
```

#### 55. **long RefValue=\_RegFileCallBack\_P(CallBackProc);**

for lua, ruby, python, this function is same as \_RegFileCallBack

```
java: long _RegFileCallBack_P(StarObjectFileCallBackInterface CallBackProc)
```

```
public interface StarObjectFileCallBackInterface{
    public void Invoke(Object CleObject,int uMsg, boolean DataFile,int
ReceiveOrSendOffset,int DataSize, Object FileName_Object, int UniqueDataUnitID,String
Version);
```



```

}

c#: long _RegFileCallBack_P(StarObjectFileCallBackInterface CallBackProc);

public delegate void StarObjectFileCallBackInterface(Object CleObject,int uMsg,
bool DataFile,int ReceiveOrSendOffset,int DataSize, Object FileName_Object, int
UniqueDataUnitID,String Version);

```

#### 56. **\_UnRegFileCallBack(RefValue)**

```
java: void _UnRegFileCallBack(long RefValue)
```

#### 57. **RetVal = \_Call(FuncName,...)**

```

java: Object _Call(String FuncName,Object...Args)
java/c#: boolean _Callbool(String FuncName,Object...Args)
java/c#: int _Callint(String FuncName,Object...Args)
java/c#: double _Calldoule(String FuncName,Object...Args)
java/c#: StarObjectClass _Callobject(String FuncName,Object...Args)

```

#### 58. **RetVal = \_SyncCall(FuncName,...)**

```

java: Object _SyncCall(String FuncName,Object...Args)
java/c#: boolean _SyncCallbool(String FuncName,Object...Args)
java/c#: int _SyncCallint(String FuncName,Object...Args)
java/c#: double _SyncCalldoule(String FuncName,Object...Args)
java/c#: StarObjectClass _SyncCallobject(String FuncName,Object...Args)

```

The called function runs in main thread, which inits cle.

#### 59. **void = \_RemoteCall(ClientID, FuncName,...) / \_RemoteCallEx(ExcludeClientID, FuncName,...)**

```

java: void _RemoteCall(int ClientID,String FuncName,Object...Args)
java: void _RemoteCallEx(int ExcludeClientID,String FuncName,Object...Args)

```

ClientID will be ignored at debug and client side; At server side, if ClientID equals 0, then the call will be proceed by all the client, or else the corresponding client.

FuncName is the function name. “...” is input arguments, which may be number,bool,parapkg,binbuf,or string.

Calling format:

```
Object._RemoteCall(“FuncName”,Arg1,Arg2..)
```

```
Object._RemoteCall(ClientID,”FuncName”,Arg1,Arg2,...)
```

```
Object._RemoteCallEx(ExcludeClientID,FuncName,...)
```

\_RemoteCallEx indicates all client except ExcludeClientID, and only valid at server.

```
Object._RemoteCallEx(ExcludeClientID,”FuncName”,Arg1,Arg2,...)
```

#### 60. **RetCode, RetVal1,RetVal,... =\_SRemoteCall(WaitTime(ms),ClientID,FuncName,...) valid at client**

```
java: Object[] _SRemoteCall(int WaitTime,int ClientID,String FuncName,Object...Args)
```

FuncName is the function name. “...” is input arguments, which may be number,bool,parapkg,binbuf,or string.

RetCode = 0 for succeed

Calling format:

Object.\_SRemoteCall(0,0,"FuncName",Arg1,Arg2..)

WaitTime: unit is ms, if equals to zero, then will wait for event.

#### 61. **\_\_ARemoteCall(WaitTime(ms),ClientID,CallBackFunc, FuncName, (int)Para,...)**

```
java: void __ARemoteCall(int WaitTime,int ClientID,String CallBackFunc,String
FuncName,int Para,Object...Args)
```

FuncName is the function name. "..." is input arguments, which may be number,bool,parapkg,binbuf,or string.

Calling format:

```
Object._ARemoteCall(0,0,CallBackFunc,"FuncName",Para,Arg1,Arg2..)
```

```
def CallBackFunc( self, RetCode,ServiceGroupID,Para, RetValue) :
```

```
function Object:CallBackFunc( RetCode,ServiceGroupID, Para, RetVal1)
```

```
end
```

```
java/c#:
public void ARemoteCallBack(StarObjectClass self,int RetCode,int ServiceGroupID,int
Para,Object RetValue)
{
}
```

WaitTime: unit is ms, if equals to zero, then will wait forever.

#### 62. **\_\_ARemoteCall\_P(WaitTime(ms),ClientID,CallBackFunc, FuncName, (int)Para,...)**

for lua, ruby, python, this function is same as \_\_ARemoteCall

```
java: void __ARemoteCall_P(int WaitTime,int ClientID,StarObjectARemoteCallInterface
CallBackProc,String FuncName,int Para,Object...Args)
```

```
public interface StarObjectARemoteCallInterface{
    public void Invoke(Object CleObject,int RetCode,int ServiceGroupID,int Para,Object
RetVal);
}
```

```
c#: void __ARemoteCall_P(int WaitTime,int ClientID,StarObjectARemoteCallInterface
CallBackProc,String FuncName,int Para,Object...Args);
```

```
public delegate void StarObjectARemoteCallInterface(Object CleObject, int
RetCode, int ServiceGroupID, int Para, Object RetValue);
```

#### 63. **Value=\_GetRemoteAttach(ParaName)**

```
java: Object _GetRemoteAttach(String ParaName)
```

Get attach parameters of the request, which is used for VSRCALLSRC\_WEBSERVICE, and defines as follow ( C format ):

```
struct StructOfVSRemoteCallRequestAttach_WebService{
    struct StructOfSRPComm_HttpOnRequest *HttpRequest;
    class ClassOfSRPSXMLInterface *SoapInfo;
    VS_CHAR *OperationName;
```

```

struct{
    VS_ULONG MimeDataSize;
    VS_INT8  *MimeDataBuf;
}MimeData;  ->  map to BinBuf
};

```

#### 64.     \_SetDeferRspFlag ()

```
java: void _SetDeferRspFlag()
```

The response to client will be delayed. The function should be executed at the function being called.

#### 65.     \_SetRetCode ( int RetCode )

```
java: void _SetRetCode( int RetCode )
```

Set return code for response. the function should be executed at the function being called.

#### 66.     \_SetRemoteRspAttach ( RemoteAttach ) **[Reserved]**

```
java: void _SetRemoteRspAttach( Object...RemoteAttach)
```

The function is valid for VSRCALLSRC\_WEBSERVICE:

```

struct StructOfVSRemoteCallResponseAttach_WebService{
    class ClassOfSRPSXMLInterface *SoapInfo;
    struct{
        VS_ULONG MimeDataSize;
        VS_INT8  *MimeDataBuf;
    }MimeData;
    VS_CHAR  *MimeContentType;
};

```

SoapInfo: is used to define Envelop,Head,and Body,Operation.

for example:

Object:\_SetRemoteRspAttach(SoapInf\_XMLInterface,MiniData\_BinBuf,

MimeContentType)

java/c# should use Hashtable.

#### 67.     \_RemoteCallRsp (int ClientID, int RemoteCallID, String RemoteCallName,int RemoteSourceTag,int RetCode, Object[] RemoteAttach,int RetType, Object RetValue)

```
java: void _RemoteCallRsp(int ClientID,int RemoteCallID,String RemoteCallName,int RemoteSourceTag,int RetCode,Object[] RemoteAttach,int RetType,Object RetValue)
```

If the remotecall response is set to be deferred, then the function is used to send response to client some times later.

ClientID: get from **Object.\_RemoteID** in the function being called.

RemoteCallID: get from **Object.\_RemoteCallID** in the function being called.

RemoteSourceTag: get from **Object.\_RemoteSourceTag** in the function being called.

**RetValue:** supported types including number,bool,parapkg,binbuf,string and object.

RemoteAttach: if not exist,should set to None

string ,Service.TYPE\_CHARPTR

parapkg,Service.TYPE\_PARAPKG

object,Service.TYPE\_OBJPTR

#### 68.     Bool = \_FillSoapRspHeader (SXml)

```
java: boolean _FillSoapRspHeader(StarSXmlClass SXml)
```

fill SOAP response header, the contents is:

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="<http://schemas.xmlsoap.org/soap/envelope/>"

xmlns:SOAP-ENC="<http://schemas.xmlsoap.org/soap/encoding/>"

xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"

xmlns:xsd="<http://www.w3.org/2001/XMLSchema>"

xmlns:ns1="urn:starcore-XXXXX">

</SOAP-ENV:Envelope>

#### 69. Bool = \_CreateFunc(FuncName,"FuncScript")

```
java: boolean _CreateFunc(String FuncName,String FuncScript)
```

Valid at client and server. The is used to create lua script function.

FunctionScript format:

script starts by prefix '@', follows by script interface name with space, and then the script body.

if '@' does not exist, default is lua script.

script interface length should be less than 15 bytes.

note: only support lua in current version.

#### 70. Bool = \_CreateFuncEx(FuncName,"ScriptFileName")

```
java: boolean _CreateFuncEx(String FuncName,String ScriptFileName)
```

Valid at client and server, which is used to create lua script function from file.

FunctionScript format:

script starts by prefix '@', follows by script interface name with a space, and then the script body.

if '@' does not exist, default is lua script.

script interface length should be less than 15 bytes.

note: only support lua in current version.

#### 71. \_DelFunc(FuncName)

```
java: void _DelFunc(String FuncName)
```

Valid at client and server, which is used to delete object's script.

#### 72. Bool = \_SaveToFile(FileName,char \*PassWord,int SaveFlag, Boolean SaveNameValue)

```
java: boolean _SaveToFile(String FileName,String PassWord,int SaveFlag,boolean SaveNameValue)
```

save object to file.

The function can not save static object and client object.

#### 73. bool = \_LoadFromFile(FileName, char \*Password, bool LoadAsLocal ,bool LoadNameValue,bool UpdateFlag, bool StaticDataUseFile)

bool = \_LoadFromBuf(BinBuf, char \*Password, bool LoadAsLocal ,bool LoadNameValue,bool UpdateFlag, bool StaticDataUseFile)

```
java: boolean _LoadFromFile(String FileName, String Password, boolean LoadAsLocal,boolean LoadNameValue,boolean UpdateFlag, boolean StaticDataUseFile)
```

```
java: boolean _LoadFromBuf(StarBinBufClass BinBuf,String Password,boolean LoadAsLocal,boolean LoadNameValue,boolean UpdateFlag,boolean StaticDataUseFile)
```

Restore object from file or binbuf. If UpdateFlag is true, ID of object will be updated.

**74.     \_DeferLoadFromFile(FileName, char \*Password, bool LoadAsLocal ,bool LoadNameValue,bool UpdateFlag, bool StaticDataUseFile)**

```
java: void     _DeferLoadFromFile(String     FileName,String     Password,boolean  
LoadAsLocal,boolean LoadNameValue,boolean UpdateFlag,boolean StaticDataUseFile)
```

Restore object from file or binbuf. If UpdateFlag is true, ID of object is updated.

for example:

```
VDisk._LoadFromFile("d:\\Disk.Obj", "123",True, True,True)
```

Restore object will lead to object being deactivated,and then reactivated. If is called in object's activate event, DeferLoadFromFile function should be used, which will restore object at next Ticket(10ms).

when handles the activate event, application should check the event paramater LParam. If it equals to 1, then should not continue to call this function.

**75.     \_ResetLoad()**

```
java: void _ResetLoad()
```

clear all child objects being loaded

//Name Value functions should be called at server and client.If they are called at server, the changes will be synchronized to client automaticly.If they are called at client, the changes take effect at local.

**76.     Bool =\_SetNameInt("Name",int,bool LocalChange)**

```
java: boolean _SetNameInt(String Name,int Value,boolean LocalChange)
```

Set integer name value.

**77.     Int =\_GetNameInt("Name",DefaultInt)**

```
java: int _GetNameInt(String Name,int DefaultInt)
```

Get integer name value.

**78.     Bool =\_SetNameFloat("Name",float,bool LocalChange)**

```
java: boolean _SetNameFloat(String Name,double Value,boolean LocalChange)
```

Set float name value.

**79.     float =\_GetNameFloat("Name",DefaultFloat)**

```
java: double _GetNameFloat(String Name,double DefaultFloat)
```

Get float name value.

**80.     Bool =\_SetNameStr("Name",Str,bool LocalChange)**

```
java: boolean _SetNameStr(String Name,String Str,boolean LocalChange)
```

Set string name value.

**81.     Str =\_GetNameStr("Name" ,DefaultStr)**

```
java: String _GetNameStr(String Name,String DefaultStr)
```

Get string name value.

**82.     Bool =\_SetNameTime("Name",Time,bool LocalChange)**

```
java: boolean _SetNameTime(String Name,StarTimeClass Tm,boolean LocalChange)
```

Set datetime name value.

**83.     Time =\_GetNameTime("Name",DefaultTime)**

```
java: StarTimeClass _GetNameTime(String Name,StarTimeClass DefaultTm)
```

Get datetime name value.

#### 84. **\_FreeNameValue("Name")**

```
java: void _FreeNameValue(String Name)
```

free name value, **Object.\_FreeNameValue()** means free all name value.

#### 85. **Type = \_GetNameValueType("Name")**

```
java: int _GetNameValueType(String Name)
```

Get type of name value.

#### 86. **name value change call back**

```
def Object_OnNameValueChange(self, Name,NameHashValue) :
```

```
Object._OnNameValueChange = Object_OnNameValueChange
```

```
function Object:_OnNameValueChange(Name,NameHashValue)
```

```
end
```

```
function _OnNameValueChange(self,Name,NameHashValue)
```

```
end
```

```
java/c#:
```

```
public void _OnNameValueChange(StarObjectClass self,String Name,int NameHashValue)
{
}
```

1. **void \_RegOnNameValueChange(String FuncName)**

2. **void \_RegOnNameValueChange\_P(StarObjectOnNameValueChangeInterface  
CallBackProc)**

java:

```
public interface StarObjectOnNameValueChangeInterface{
    public void Invoke(Object CleObject,String Name,int NameHashValue);
}
```

c#:

```
public delegate void StarObjectOnNameValueChangeInterface(Object CleObject,String  
Name,int NameHashValue)
```

#### 87. **simple format to set name value(name is started by '\_\_\_',three '\_'), LocalChange is False as default.**

```
Object.___Name = Value
```

```
Value = Object. ___Name
```

**88. attribute change callback**

```
def Object_OnChange(self,AttributeName) :
```

```
Object_OnChange = Object_OnChange
```

```
function Object:_OnChange(AttributeName)
```

```
end
```

```
function _OnChange (self, AttributeName)
```

```
end
```

```
java/c#:
```

```
public void _OnChange(StarObjectClass self,String Name)
```

```
{
```

```
}
```

**1. void \_RegOnChange(String FuncName)**

**2. void \_RegOnChange\_P(StarObjectOnChangeInterface CallBackProc)**

java:

```
public interface StarObjectOnChangeInterface{
```

```
    public void Invoke(Object CleObject,String AttributeName);
```

```
}
```

c#:

```
public delegate void StarObjectOnChangeInterface(Object CleObject,String AttributeName)
```

**object static data management.****89. bool =\_CanSetStaticData(Size)**

```
java: boolean _CanSetStaticData(int Size)
```

whether setting object static data is permitted, valid at client.

**90. String DataVersion =\_SetStaticData(StaticAttributeName,BinBuf)**

```
java: String _SetStaticData(String StaticAttributeName,StarBinBufClass BinBuf)
```

Update object static data. If succeed, the function returns static data version. Which should be set to the corresponding attribute of object. If fails, the function returns None.

StaticAttributeName: is attribute name of static data of object

BinBuf: binary buffer.

**91. DataVersion =\_SetStaticDataEx(StaticAttributeName, DataSize,Offset, FileName)[Reserved]**

```
java: String _SetStaticDataEx(String StaticAttributeName,int DataSize,int Offset,String FileName)
```

StaticAttributeName is attribute name of static data of object

**92.      DataVersion =\_GetStaticData(StaticAttributeName,BinBuf,DataVersion, (BOOL) AutoDownload)**

```
java: String _GetStaticData(String StaticAttributeName,StarBinBufClass BinBuf,String
DataVersion,boolean AutoDownload)
```

StaticAttributeName is attribute name of static data of object

BinBuf: binary buffer.

DataVersion may be "", which means not care about version.

If fails, the function returns None.

**93.      bool =\_WaitGetStaticData (StaticAttributeName,Object\_CallBack, bool WaitFlag)**

```
java: boolean _WaitGetStaticData(String StaticAttributeName,String
Object_CallBack,boolean WaitFlag)
```

If returns False, then the static data is not downloaded or error occurred in download process.

WaitFlag=False, query whether to wait upload or download.

**94.      bool =\_WaitGetStaticData\_P(StaticAttributeName, bool WaitFlag,CallBackProc)**

for lua, ruby, python, this function is same as \_\_WaitGetStaticData

```
java: boolean _WaitGetStaticData_P(String StaticAttributeName,boolean
WaitFlag,StarObjectGetStaticDataInterface CallBackProc)
```

```
public interface StarObjectGetStaticDataInterface{
    public int Invoke(Object CleObject,int uMes,int CurSize,int MaxSize);
}
```

```
c#: bool _WaitGetStaticData_P(String StaticAttributeName,bool
WaitFlag,StarObjectGetStaticDataInterface CallBackProc);
```

```
public delegate int StarObjectGetStaticDataInterface(Object CleObject,int
uMes,int CurSize,int MaxSize);
```

**95.      bool =\_WaitSetStaticData (StaticAttributeName, Object\_CallBack, bool WaitFlag)**

```
java: boolean _WaitSetStaticData(String StaticAttributeName,String
Object_CallBack,boolean WaitFlag)
```

If returns False, then the static data is not downloaded or error occurred in download process.

WaitFlag=False, query whether to wait upload or download.

```
def Object_CallBack(self,uMes,CurSize,MaxSize)
```

```
    return 0,1
```

```
function Object:CallBack(uMes,CurSize,MaxSize)
```

```
    return 0,1
```

```
end
```

```
java/c#:
```

```
public int CallBack(StarObjectClass self,int uMes, int CurSize, int MaxSize)
```



```
{
    return 0,1;
}
```

If returns 0, then the process is continue. Otherwise the process is canceled. The return value takes effect in VSFILE\_ONDOWNPROGRESS/ VSFILE\_ONUPPROGRESS process.

#### 96. **bool = \_WaitSetStaticData\_P(StaticAttributeName, bool WaitFlag, CallBackProc)**

for lua, ruby, python, this function is same as \_\_WaitSetStaticData

```
java:      boolean      _WaitSetStaticData_P(String      StaticAttributeName,boolean
WaitFlag,StarObjectSetStaticDataInterface CallBackProc)
```

```
public interface StarObjectSetStaticDataInterface{
    public int Invoke(Object CleObject,int uMes,int CurSize,int MaxSize);
}
```

```
c#:      bool      _WaitSetStaticData_P(String      StaticAttributeName,bool
WaitFlag,StarObjectSetStaticDataInterface CallBackProc);
```

```
public delegate int StarObjectSetStaticDataInterface(Object CleObject, int
uMes, int CurSize, int MaxSize);
```

#### 97. **bool=\_SaveToLuaFunc (FileName,Funcname)**

```
java: boolean _SaveToLuaFunc(String FileName,String Funcname)
```

The function can be used to create the lua script of object attribute value.

#### 98. **\_Init (“attr1=value; attr2=value2; func();”)**

```
java: void _Init(String Arg)
```

**Each sentence should be followed by';'.**

syntax format:

sentences are seperated by ';' ;

for each sentence,

If the first char is'\$', then the following '\$O' will be replaced by Object before executed.

For example:”\$O:\_Active();” is translated to Object:\_Active() .

If not, then the following rule takes effect.

If contains '=' , then translates to Object. XXXX

otherwise translate to Object: XXXX

for example:

**“attr1=value; attr2=value2; func();”**

\$O is not case sensitive.

#### 99. **bool=\_RemoteSend(ClientID, ParaPkg)**

```
java: boolean _RemoteSend(int ClientID,StarParaPkgClass ParaPkg)
```

At server side, if ClientID equals to 0, then the message is sent to all client, and the object must be global object; If ClientID does not equal to 0, then the message is sent to the corresponding client;

At client side, the function means sending message to server, and ClientID will be ignored.

At receive side, application may define event handler Object\_OnRemoteSend(self,Event) to process the message.

- 100.     **int RefValue= \_WaitEvent(InObject,EventName, Object.Function,bool AutoDelete)**  
**Object: \_UnWaitEvent(InObject,EventName, RefValue)**  
**Para1,...= \_GetEventPara(int Event)****

**Only valid for lua.**

Wait object event, when InObject.EventName event is triggered, the callback function will be called.

```
function Object:Function (SrcObject,int Event)
    print(self, Event )
    return
end
```

- 101.     **int \_GetSourceScript();****

```
int _GetSourceScript();
```

Get script interface index which creates the object

- 102.     **Object = \_DefinedClass (AttributeName);****

```
StarObjectClass _DefinedClass(String AttributeName);
```

Which object defines the attribute or function

- 103.     **StarObjectClass                         =                         \_IsFunctionDefined(FuncName, IncludeRawOrDefaultFunction);****

```
StarObjectClass _IsFunctionDefined(String FuncName,bool IncludeRawOrDefaultFunction);
```

If there are lua function or other script function registered with the object, then the function returns true.

IncludeRawOrDefaultFunction: == true, then the function of raw script object or registered using RegLuaFunc with FuncName == null is taken into account.

The return value is object or class which defines the function.

- 104.     **self = \_FromTuple(Index,Tuple);****

```
StarObjectClass _FromTuple(int Index,Object[] tuple)
```

Set object's attributes from tuple. Index is the first attribute index.

- 105.     **tuple = \_ToTuple(Index,Num);****

```
Object[] _ToTuple(int Index,int Num)
```

Get tuple from object's attributes. Index is first attribute index. Num is the number of attributes to tuple. If Num < 0, then all attributes will be changed to tuple from start index.

- 106.     **boolean \_AttachRawContext(String ScriptInterface,String ContextName,boolean IsClass,String ContextInfo)****

Associate an object with a script class or object.

**for c/c++ :**

ContextName is share library name, IsClass and ContextInfo will be ignored.

**for lua :**

ContextName is table or userdata.

if ContextName == "{", then create a new lua table.

if IsClass == true, then the lua value can be called to create instance.

ContextInfo will be ignored.

**for python :**

ContextName is module, dict, tuple, list, class.

"{}": dict "[ ]Size" list "()Size" tuple

If ContextName is invalid, then the object attach python global name space.

if IsClass == true, then the python value should be a class.

ContextInfo will be ignored.

**for ruby :**

ContextName is module, hash, array, class.

"{}": hash "[ ]Size" array

If ContextName is invalid, then the object attach ruby global name space.

if IsClass == true, then the ruby value should be a class.

ContextInfo will be ignored.

**for java :**

ContextName must be valid.

ContextName is class name.

ContextInfo will be ignored.

**for csharp :**

ContextName must be valid.

ContextName is class name.

ContextInfo will be ignored.

**107. void \_DetachRawContext(boolean CallUnLockGC)**

if CallUnLockGC == true, then \_UnLockGC will be called autotically.

**108. String \_GetRawContextType()**

Get Type name of raw script object.

**109. StarObjectClass \_AttachRawObject(Object NewObject,boolean IsClass)**

Associated cle object with raw script object. If succeed, the function returns the object self. Or else, returns null.

**110. StarObjectClass \_AssignRawObject(StarObjectClass RawObject)**

copy raw object info from RawObject, and return self

**111. Object \_GetRawObject()**

Get raw object associated with cle object

**112.     boolean \_HasRawContext()**

**113.     StarObjectClass         \_NewRawProxyEx(String         ScriptInterface,String  
          AttachFunction,String ProyInfo)**

**114.     StarParaPkgClass \_GetInitPara()**

The function is valid in ONCREATE event

**115.     boolean \_Equals(StarObjectClass Which)**

**116.     boolean \_SetScriptRawType(int RawType)**

The function is reserved

**117.     int \_GetScriptRawType()**

The function is reserved

**118.     int \_GetRefEx()**

Get object reference count.

**119.     String \_GetRefInfo()**

object is referenced by scripts, return value is script name string, separated by “,”

if object is SLockGC by the script, then there has a “\*” prefix before the script name.

for example:

“lua,\*python”

**120.     boolean \_IsValid()**

**121.     int \_GetLastError()**

**122.     String \_GetLastErrorInfo()**

**123.     void \_RegSysEventProc(String EventName,String FuncName)**

Register object's sys event function, the eventname takes value from  
\_OnTicket,\_OnFrameTicket,\_OnIdle,\_OnAppActive,\_OnAppDeactive,\_OnServiceActive,\_OnServiceDeactive,\_OnCreate,\_OnDestroy,\_OnCreateChild,\_OnDestroyChild,\_OnActivating,\_OnDeactivating,\_OnActivate,\_OnDeactivate,\_OnActiveChild,\_OnDeactiveChild,\_OnParentBeforeChange,\_OnParentChange,\_OnStaticChange,\_OnScriptChange,\_OnSyncGroupChange,\_OnChildSyncGroupChange,\_OnActiveSetChange,\_OnLoadMask,\_OnLoadFinish,\_OnRemoteSend,\_OnCall

**124.     void \_RegSysEventProc\_P(String EventName,StarObjectSysEventProcInterface  
          CallBackProc)**

for lua, ruby, python, this function is same as \_RegSysEventProc

java:   void    \_RegSysEventProc\_P(String   EventName,StarObjectSysEventProcInterface  
          CallBackProc)

```

public interface StarObjectSysEventProcInterface{
    public Object[] Invoke(Object CleObject,Object CleEvent);
}

c#: void _RegSysEventProc_P(String EventName,StarObjectSysEventProcInterface
CallBackProc);

public delegate object[] StarObjectSysEventProcInterface(Object CleObject, Object
CleEvent);

```

**125. void \_RegScriptProc\_P(String ScriptName,StarObjectScriptProcInterface CallBackProc)**

define new script function, support lua,ruby,python,java,c#.

```

public interface StarObjectScriptProcInterface{
    public Object Invoke(Object CleObject,Object[] Paras);
}

c#:
public delegate object[] StarObjectScriptProcInterface(Object CleObject, object[]
Paras);

```

**126. Iterator \_Iterator()**

**c#: IEnumerator \_Iterator();**

Support lua/python/java/c#.

**127. boolean \_ReleaseOwnerEx()**

This function should be used when cle object is create by one language, and assign script function, and will not used in this language and output to other languages

**128. boolean \_IsSLock()**

**129. Object \_R(Object AttributeNameOrIndex)**

Get attribute's raw value. Object should be raw object;

**130. Attribute Get or Set**

**value = \_Get (Name)**

**\_Set (Name,value)**

**131. Attribute Get or Set Callback**

From v2.5.0, cle supports two default callbacks for attributes "GetValueCallBack" and "SetValueCallBack". The SetValueCallBack is only valid for object self, and GetValueCallBack takes effect for object self and it's instance.

For an object, only one GetValueCallBack and one SetValueCallBack should be defined for each script language.

```
public Object[] GetValueCallBack(StarObjectClass self, String a)
public boolean SetValueCallBack(StarObjectClass self, String a, Object Val)
```

for example,

java:

```
StarObjectClass c = obj._New()._Assign( new StarObjectClass(){
    public Object[] GetValueCallBack(StarObjectClass self, String a)
    {
        System.out.println(a);
        return new Object[]{true,"aaaaaaa"};
    }
    public boolean SetValueCallBack(StarObjectClass self, String a, Object Val)
    {
        System.out.println(a);
        System.out.println(Val);
        return true;
    }
});
```

python:

```
def c_SetValueCallBack(self,name,val) :
    print(name,val)
    return False
c.SetValueCallBack = c_SetValueCallBack

def c_GetValueCallBack(self,name) :
    print(name)
    return (False,"aaaaaaa");
c.GetValueCallBack = c_GetValueCallBack
```

for java and c#, if \_RegScriptProc\_P is used to define the above two function. The prototype should be as follow,

```
public Object GetValueCallBack(Object CleObject, Object Paras[])
public boolean SetValueCallBack(Object CleObject, Object Paras[])
```

for example,

```
static object GetValueCallBack(Object CLEObject,object[] Paras)
{
    Console.WriteLine(Paras[0]);
    return new Object[]{ false,"aaaaaaa"};
```

```

}
static object SetValueCallBack(Object CLEObject,object[] Paras)
{
    Console.WriteLine(Paras[0]);
    Console.WriteLine(Paras[1]);
    return false;
}

c._RegScriptProc_P(      "GetValueCallBack",      new      StarObjectScriptProcInterface
(testgetset.GetValueCallBack));
c._RegScriptProc_P(      "SetValueCallBack",      new      StarObjectScriptProcInterface
(testgetset.SetValueCallBack));

```

### 132. RawToParaPkg

java/c# : StarParaPkgClass \_RawToParaPkg()

## 5.9 object defined attributes and functions

### 5.9.1 dynamic script language

#### 1. set or get object attribute

For global pointer attribute, should only get, which will return the first object in the queue.

The change will take effect at local. If it is needed to synchronize to client, should use function

Object.\_Change("AttrName",value)

struct attribute should be changed with tuple or array. For example:

attribute P1 of Object is struct, which include two variables X and Y.

print( Object.P1.X,Object.P1.Y );    lua/python/ruby

Object.P1={2,3}    lua

Object.P1=(2,3)    python

Object.P1=[2,3]    ruby

#### 2. call object's function

Call lua functions of object directly using function name.

calling format:

Object.FuncName(...).

FuncName is function name, may be define in object' class. The function is lua function, which prototype is int FunctionName( void \*LuaState ).

### 5.9.2 static script language[java or c#]

#### 1. set or get object attribute

Using \_Get or \_Set method, for example:

Value = Object.\_Get(attribute name)

Object.\_Set(attribute name, Value);

#### 2. call object function

Object.\_Call(function name, args, ...)

### 5.10 ParaPkg object

Support bool, integer, float, string , time, object, and binbuf.

#### 5.10.1 attribute

##### 1. ParaPkg.\_Number

The number of values in the package.

#### 5.10.2 function

##### 1. int \_T(Index)

java: int \_T(int Index)

Get type of value at Index, which defines as follow:

SRPPARATYPE_INVALID	0	//--invalid
SRPPARATYPE_INT	1	//--integer
SRPPARATYPE_FLOAT	2	//--float
SRPPARATYPE_BIN	3	
SRPPARATYPE_CHARPTR	4	//--string
SRPPARATYPE_TIME	5	//--time
SRPPARATYPE_BOOL	6	//--bool
SRPPARATYPE_OBJECT	7	
SRPPARATYPE_PARAPKG	8	
SRPPARATYPE_INT64	9	

For C#, please use \_TT(Index)

##### 2. StarParaPkgClass \_Clear()

java: StarParaPkgClass \_Clear()

Clear all values and return self.

##### 3. StarParaPkgClass \_Build(...)

java: StarParaPkgClass \_Build(Object...tuple)

Same as \_FromTuple

##### 4. bool = \_Exchange(DesIndex,SrcIndex)

java: boolean \_Exchange(int DesIndex,int SrcIndex)

Exchange position of two values.



**5. void \_Del(Index)**

```
java: void _Del(int Index)
```

Delete a value.

For lua and python and ruby, app can use Para[Index] to get or set values.

**6. Bool = \_AppendFrom(SrcParaPkg)**

```
java: boolean _AppendFrom(StarParaPkgClass SrcParaPkg)
```

Duplicate values from SrcParapkg to ParaPkg.

**7. \_GetUUID(Index)**

```
java: String _GetUUID(int Index)
```

Get UUID value of binbuf, which returns a string.

**8. int = \_GetHash(Index)**

```
java: int _GetHash(int Index)
```

Get hash code of the value.

**9. BOOL = \_SaveToFile(Index,FileName)**

```
java: boolean _SaveToFile(int Index,String FileName)
```

Save binbuf to disk file. The value at Index must be binbuf

**10. BOOL = \_LoadFromFile(Index,FileName)**

```
java: boolean _LoadFromFile(int Index,String FileName)
```

Load binbuf from disk file.

**11. BOOL = \_CopyBin(Index, SrcParaPkg,SrcIndex)**

```
java: boolean _CopyBin(int Index,StarParaPkgClass SrcParaPkg,int SrcIndex)
```

Duplicate binbuf from parapkg. The value at SrcIndex must be binbuf.

**12. Value = \_Get (Index)**

```
java: Object _Get(int Index)
    public int _Getint(int Index);
    public boolean _Getbool(int Index);
    public double _Getdouble(int Index);
    public String _GetStr(int Index);
    public StarBinBufClass _GetBinBuf(int Index);
    public StarObjectClass _Getobject(int Index);
    public StarParaPkgClass _GetParaPkg(int Index)
```

Get value from parapkg. For lua and python and ruby, app can use ParaPkg[Index].

**13. StarParaPkgClass \_Set(Index,Value)**

```
java: StarParaPkgClass _Set(int Index,Object Value)
```

Set value to parapkg and return self, For lua and python and ruby, app can use ParaPkg[Index].

**14. Value = \_GetTime (Index)**

```
java: StarTimeClass _GetTime(int Index)
```

Get time value from parapkg, For lua, app can use ParaPkg[Index].

**15. bool = \_SetTime (Index,Value)**

```
java: boolean _SetTime(int Index,StarTimeClass Value)
```

Set time value to parapkg, For lua, app can use ParaPkg[Index].

**16. \_SetChangeFlag (Index)**

```
java: void _SetChangeFlag(int Index)
```

Set field change flag.

**17.        \_SetChangeFlagEx ()**

java: void \_SetChangeFlagEx()

Set all fields change flag.

**18.        \_ClearChangeFlag (Index)**

java: void \_ClearChangeFlag(int Index)

Clear field change flag.

**19.        \_ClearChangeFlagEx ()**

java: void \_ClearChangeFlagEx()

Clear all fields change flag.

**20.        Bool = \_IsChangeFlag (Index)**

java: boolean \_IsChangeFlag(int Index)

whether the field is changed

**21.        Bool = \_IsChangeFlagEx()**

java: boolean \_IsChangeFlagEx()

whether there is any field changed

**22.        Bool = \_SaveChangeToBuf(BinBuf)**

java: boolean \_SaveChangeToBuf(StarBinBufClass BinBuf)

Pack changed fields to binbuf.

**23.        Bool = \_SaveChangeToBufEx(BinBuf)**

java: boolean \_SaveChangeToBufEx(StarBinBufClass BinBuf)

Pack all fields to binbuf.

**24.        Bool = \_LoadChangeFromBuf (BinBuf)**

java: boolean \_LoadChangeFromBuf(StarBinBufClass BinBuf)

Restore fields from binbuf.

**25.        self = \_FromTuple(Value1,...)**

java: StarParaPkgClass \_FromTuple(Object...tuple)

The returned value is self. The function will clear existing items

for **python and ruby**, this function can handle dict or hashtable(for ruby). The key must be string or integer. As result, the IsDict is set to true.

**26.        tuple = \_ToTuple ()**

java: Object[] \_ToTuple()

Restore table, tuple, or array. If error occurs, the function returns nil/NULL/None.

for **python and ruby**, IsDict == true, then this function returns dict for python and hashtable for ruby. In this case, the key must be string or integer.

**27.        void \_SetScriptRawType(int RawType)**

the function is reserved

**28.        int \_GetScriptRawType()**

the function is reserved

**29.        Iterator \_Iterator()**

**c#: IEnumerator \_Iterator();**

Support lua/python/java/c#.

for ruby, this function returns the parapkg self.

**30. void \_Free()**

The script object must not be used after the function is called.

**31. void \_Dispose()**

This function is same as \_Free.

**32. void \_ReleaseOwner()**

This function is mainly for java and c#, which will decrease the effect of performance degrade of GC operation.

If the instance is returned to other language and do not used in future, this function can be called.

The script object must not be used after the function is called.

note : for ruby, parapg supports “each”, for example,

```
for val in pa
  puts( val )
end
```

*5. 10. 3 Attribute Get***1. value = \_Get (Name/int)**

if Name is int value, then the function is same as \_Get(Index)

*5. 10. 4 Dict and JSON function***1. StarParaPkgClass = \_AsDict(bool IsDict)**

The parapg is interpreted as dict. when mapping to raw value, it will be mapped to dictionary. This function returns self.

**2. bool = \_IsDict()****3. index = \_FindDict(string key)**

Return -1 means not found.

**4. index = \_FindDictEx(int key)**

Return -1 means not found.

**5. bool = \_FromJSon (string jsonstring)****6. string = \_ToJSon ()**

The parapg must be a dict.

*5. 11 Binbuf object*

Types supported:

```
#define SRPBINTYPE_BOOL ((VS_UINT8)'z')
#define SRPBINTYPE_BYTE ((VS_UINT8)'b') //map to int
#define SRPBINTYPE_UBYTE ((VS_UINT8)'B') // map to int
#define SRPBINTYPE_CHAR ((VS_UINT8)'c') // map to int
#define SRPBINTYPE_SHORT ((VS_UINT8)'h') // map to int
```

```

#define SRPBINTYPE_USHORT ((VS_UINT8)'H')    // map to int
#define SRPBINTYPE_INT    ((VS_UINT8)'i')    // map to int
#define SRPBINTYPE_UINT   ((VS_UINT8)'T')    // map to int
#define SRPBINTYPE_LONG   ((VS_UINT8)'l')    // map to int
#define SRPBINTYPE_ULONG  ((VS_UINT8)'L')    // map to int
#define SRPBINTYPE_FLOAT  ((VS_UINT8)'d')     //double
#define SRPBINTYPE_STRING ((VS_UINT8)'s')     //with the end of 0
#define SRPBINTYPE_STRINGEX ((VS_UINT8)'S')   //without 0 at end
#define SRPBINTYPE_UNICODE ((VS_UINT8)'u')    // with the end of 0
#define SRPBINTYPE_UNICODEEX ((VS_UINT8)'U')  //without 0 at end
#define SRPBINTYPE_UTF8    ((VS_UINT8)'t')    // with the end of 0
#define SRPBINTYPE_UTF8EX  ((VS_UINT8)'T')    // without 0 at end
#define SRPBINTYPE_BIN     ((VS_UINT8)'r')    //binary

```

Input uses string with single char, for example :”o”, “b”,etc.

### 5. 11. 1 attribute

1. **BinBuf.\_Size**  
size of the memory allocated
2. **BinBuf.\_Offset**  
size of data
3. **BinBuf.\_Buf**  
buffer address
4. **BinBuf.\_Name[r/w]**  
buffer’s alias name

### 5. 11. 2 function

1. **\_Init (BufSize)**  
java: void \_Init(int BufSize)  
alloc memory with BufSize, the function clears existed data.
2. **\_Clear ()**  
java: void \_Clear()  
Clear existed data
3. **\_ClearEx (Offset,Length)**  
java: void \_ClearEx(int Offset,int Length)  
Clear data from Offset and Length size.
4. **Length=\_Set (Offset,Length,Type,Object Value)**  
java: int \_Set(int Offset,int Length,String Type,Object Value)  
Returns Length; if the function returns 0, then there is error occured. Length is reserved.
5. **object Value = \_Get (Offset,Length,Type)**  
java: Object \_Get(int Offset,int Length,String Type)  
If fails, returns None. Length is reserved.
6. **BOOL =\_SaveToFile(FileName,TxtFileFlag = true/false)**

- java: boolean \_SaveToFile(String FileName,boolean TxtFileFlag)  
Save binay data to diak file. If **TxtFileFlag=true**, then save as text file.
7. **BOOL = \_LoadFromFile(FileName,TxtFileFlag = true/false)**  
java: boolean \_LoadFromFile(String FileName,boolean TxtFileFlag)  
Load binary data from disk file. if **TxtFileFlag=true**, then load as text file.
8. **BOOL = \_SetOffset (Offset)**  
java: boolean \_SetOffset(int Offset)  
Set size of data.
9. **BOOL = \_Expand (NewBufSize)**  
java: boolean \_Expand(int NewBufSize)  
expand buffer size.
10. **BOOL = \_Fill (Offset,Length,Char)**  
java: boolean \_Fill(int Offset,int Length,String Val)  
fill with char
11. **BOOL = \_PackObject(Object)/\_UnPackObject(Object)**  
java: boolean \_PackObject(StarObjectClass Obj)  
java: boolean \_UnPackObject(StarObjectClass Obj)  
pack object into binbuf
12. **BOOL = \_ToUTF8()**  
java: boolean \_ToUTF8()  
Change data coding to UTF8. The buffer should be a string. If the string is ended with 0, then the converted string is ended with 0. Otherwise, without 0.
13. **BOOL = \_ToAnsi ()**  
java: boolean \_ToAnsi()  
Change data coding to Ansi, The buffer should be a string. If the string is ended with 0, then the converted string is ended with 0. Otherwise, without 0.
14. **\_InsertStr(Offset,String)**  
java: void \_InsertStr(int Offset,String Val)
15. **int = \_FindStr(Offset,String)**  
java: int \_FindStr(int Offset,String Val)  
If not find, returns a value less than 0
16. **int = \_FindStri(Offset,String)**  
java: int \_FindStri(int Offset,String Val)  
If not find, returns a value less than 0
17. **\_Print (..)**  
java: void \_Print(String Arg)  
print string to the tail of the buf.
18. **long Handle = \_OpenFile(FileName,String Mode)**  
java: long \_OpenFile(String FileName,String Mode)
19. **Size = \_GetFileSize(long Handle)**  
java: int \_GetFileSize(long Handle)
20. **Length = \_ReadFile(long Handle,Offset,Length)**  
java: int \_ReadFile(long Handle,int Offset,int Length)
21. **Length = \_WriteFile(long Handle,Offset,Length)**

```
java: int _WriteFile(long Handle,int Offset,int Length)
```

## 22. **\_CloseFile(long Handle)**

```
java: void _CloseFile(long Handle)
```

## 23. **bool=\_IsLightBuf()**

```
java: boolean _IsLightBuf()
```

## 24. **BOOL =\_AnsiToUnicode (code,BytesPerChar)**

```
java: boolean _AnsiToUnicode(String code,int BytesPerChar)
```

Change data coding to unicode. The buffer should be a string. If the string is ended with 0, then the converted string is ended with 0. Otherwise, without 0. Code will be ignored on windows, which may be take values from : UCS2 UCS4 UTF-16 UTF-32 UTF-16BE UTF-16LE UTF-32BE UTF-32LE, BytesPerChar should be set to 2 on windows.

## 25. **BOOL =\_UnicodeToAnsi (code,BytesPerChar)**

```
java: boolean _UnicodeToAnsi(String code,int BytesPerChar)
```

Change data coding to ansi, The buffer should be a string. If the string is ended with 0, then the converted string is ended with 0. Otherwise, without 0. Code will be ignored on windows, which may be take values from: UCS2 UCS4 UTF-16 UTF-32 UTF-16BE UTF-16LE UTF-32BE UTF-32LE, BytesPerChar should be set to 2 on windows.

## 26. **lstring=\_Read(Offset,Length)/java/c#:int Length=\_Read(byte[] buf,int Offset,int Length)**

```
java: int _Read(byte[] Buf,int Offset,int Length)
```

## 27. **Length =\_Write(Offset, lstring)/java/c#:int Length=\_Write(int Offset,byte[] buf,int Length)**

```
java: int _Write(int Offset,byte[] Buf,int Length)
```

## 28. **\_Read2:java/c#**

```
public int _Read2(short[] Buf,int Offset,int Length,int byte0index,int byte1index)
```

Length is the number of elements. byte0index and byte1index may be 0 or 1.

## 29. **\_Write2:java/c#**

```
public int _Write2(int Offset,short[] Buf,int Length,int byte0index,int byte1index)
```

Length is the number of elements. byte0index and byte1index may be 0 or 1.

## 30. **\_Read4:java/c#**

```
public int _Read4(int[] Buf,int Offset,int Length,int byte0index,int byte1index,int byte2index,int byte3index)
```

Length is the number of elements. byte0index and byte1index, byte2index, byte3index may be 0,1,2,or 3.

## 31. **\_Write4:java/c#**

```
public int _Write4(int Offset,int[] Buf,int Length,int byte0index,int byte1index,int byte2index,int byte3index)
```

Length is the number of elements. byte0index and byte1index, byte2index, byte3index may be 0,1,2,or 3.

## 32. **Length =\_WriteFromMemoryFile (Service,Offset, FileName)**

```
java: int _WriteFromMemoryFile(StarServiceClass Service,int Offset,String FileName)
```

Load from Memory Files of the service

## 33. **String=\_GetMD5 ()**

```
java: String _GetMD5()
```

Get MD5

**34. Int = \_GetHash()**

```
java: int _GetHash()
```

Get Hash Value.

**35. void \_Free()**

The script object must not be used after the function is called.

**36. void \_Dispose()**

This function is same as \_Free.

**33. void \_ReleaseOwner()**

This function is mainly for java and c#, which will decrease the effect of performance degrade of GC operation.

If the instance is returned to other language and do not used in future, this function can be called.

The script object must not be used after the function is called.

**34. Byte[]/string \_ToBuf() v2.5.1**

For java and c# this function returns bytes array. For lua/ruby/python, this function return string buf.

**35. BinBuf = \_ToBuf(Byte[]/string) v2.5.1**

For java and c#, input parameter is bytes array. For lua/ruby/python, input parameter is string buf.

### 5.11.3 Attribute Get

**1. value = \_Get (Name/int)**

if Name is int value, then the function is same as \_Get(Offset,Length,Type)

### 5.12 SXML object

For Element,Text,Attribute,Comment, function returns integer, if does not exist, function returns 0 or NULL.

#### 5.12.1 load and save

**1. Bool,ErrorInfo=\_LoadFromFile(FileName)**

```
java: Object[] _LoadFromFile(String FileName)
```

**2. Bool,ErrorInfo=\_LoadFromBuf(BinBuf)**

```
java: Object[] _LoadFromBuf(StarBinBufClass BinBuf)
```

**3. bool,ErrorInfo=\_LoadFromBufEx(String) String UTF-8 format**

```
java: Object[] _LoadFromBufEx(String UtfArg)
```

**4. bool=\_SaveToFile(FileName)**

```
java: boolean _SaveToFile(String FileName)
```

**5. bool=\_SaveToBuf(BinBuf)**

```
java: boolean _SaveToBuf(StarBinBufClass BinBuf)
```

## 5. 12. 2 read

1. **string=\_GetStandalone()**

```
java: String _GetStandalone()
```

2. **string=\_GetVersion()**

```
java: String _GetVersion()
```

3. **string=\_GetEncoding()**

```
java: String _GetEncoding()
```

4. **long Element=\_FindElement(string Value)**

```
java: long _FindElement(String Value)
```

support format such as "a.b.c.d"

5. **long Element=\_FindElementEx(long ParentElement, string Value)**

```
java: long _FindElementEx(long ParentElement,String Value)
```

6. **long Element=\_FirstElement(long ParentElement)**

```
java: long _FirstElement(long ParentElement)
```

ParentElement may be 0

7. **Element=\_NextElement(long Element)**

```
java: long _NextElement(long Element)
```

8. **Element=\_ParentElement(Element)**

```
java: long _ParentElement(long Element)
```

9. **string Value=\_GetElement(Element)**

```
java: String _GetElement(long Element)
```

10. **string Value=\_GetElementEx(Element) a.b.c.d format**

```
java: String _GetElementEx(long Element)
```

11. **Bool,nsName,nsValue=\_GetNs(Element)**

```
java: Object[] _GetNs(long Element)
```

12. **sting nsValue=\_GetNsValue(Element,nsName)**

```
java: String _GetNsValue(long Element,String nsName)
```

13. **long Attribute=\_FindAttribute(Element,Name)**

```
java: long _FindAttribute(long Element,String Name)
```

14. **long Attribute=\_FirstAttribute(Element)**

```
java: long _FirstAttribute(long Element)
```

15. **long Attribute=\_NextAttribute(Attribute)**

```
java: long _NextAttribute(long Attribute)
```

16. **string Name=\_GetAttributeName(Attribute)**

```
java: String _GetAttributeName(long Attribute)
```

17. **string Value=\_GetAttributeValue(Attribute)**

```
java: String _GetAttributeValue(long Attribute)
```

18. **string=\_GetSingleText(Element)**

```
java: String _GetSingleText(long Element)
```

19. **long Text=\_FirstText(Element)**

```
java: long _FirstText(long Element)
```

20. **long Text=\_NextText(Text)**

```
java: long _NextText(long Text)
```



**21. string Value=\_GetText(Text)**

```
java: String _GetText(long Text)
```

**5. 12. 3 change****1. \_SetDeclaration(Version,Encoding,Standalone)**

```
java: void _SetDeclaration(String Version,String Encoding,String Standalone)
```

**2. \_RemoveDeclaration()**

```
java: void _RemoveDeclaration()
```

**3. Element=\_InsertElementBefore(ParentElement,Element,Value)**

```
java: long _InsertElementBefore(long ParentElement,long Element,String Value)
```

ParentElement/Element may be 0

**4. Element=\_InsertElementAfter(ParentElement,Element,Value)**

```
java: long _InsertElementAfter(long ParentElement,long Element,String Value)
```

ParentElement/Element may be 0

**5. \_RemoveElement(Element)**

```
java: void _RemoveElement(long Element)
```

**6. \_SetElement(Element,Value)**

```
java: void _SetElement(long Element,String Value)
```

**7. \_SetNs (Element,nsName,nsValue)**

```
java: void _SetNs(long Element,String nsName,String nsValue)
```

**8. long Text=\_InsertTextBefore(ParentElement,long Text,Value,bool CDataFlag)**

```
java: long _InsertTextBefore(long ParentElement,long Text,String Value,boolean CDataFlag)
```

ParentElement/Text may be 0

**9. long Text=\_InsertTextAfter(ParentElement, long Text,Value,bool CDataFlag)**

```
java: long _InsertTextAfter(long ParentElement,long Text,String Value,boolean CDataFlag)
```

ParentElement/Text may be 0

**10. \_RemoveText(long Text)**

```
java: void _RemoveText(long Text)
```

**11. \_SetText(Text,Value)**

```
java: void _SetText(long Text,String Value,boolean CDataFlag)
```

**12. long Comment=\_InsertCommentBefore(ParentElement, long Comment,String Value)**

```
java: long _InsertCommentBefore(long ParentElement,long Comment,String Value)
```

ParentElement/Comment may be 0

**13. long Comment=\_InsertCommentAfter(ParentElement, long Comment,Value)**

```
java: long _InsertCommentAfter(long ParentElement,long Comment,String Value)
```

ParentElement/Comment may be 0

**14. \_RemoveComment(Comment)**

```
java: void _RemoveComment(long Comment)
```

**15. \_SetComment(Comment,Value)**

```
java: void _SetComment(long Comment,String Value)
```

**16. \_SetAttribute(Element,Name,Value)**

```
java: void _SetAttribute(long Element,String Name,String Value)
```

**17.     \_RemoveAttribute(Element,Name)**

```
java: void _RemoveAttribute(long Element,String Name)
```

**5. 12. 4 duplicate****1.   Element=\_CopyElementBefore(ParentElement,Element,SrcElement)**

```
java: long _CopyElementBefore(long ParentElement,long Element,long SrcElement)
```

ParentElement/Element may be set to 0,SrcElement may belongs to another SXml object.

**2.   Element=\_CopyElementAfter(ParentElement,Element, SrcElement)**

```
java: long _CopyElementAfter(long ParentElement,long Element,long SrcElement)
```

ParentElement/Element may be set to 0,SrcElement may belongs to another SXml object.

**3.   bool=\_CopyChild(DesElement, SrcElement)**

```
java: boolean _CopyChild(long DesElement, long SrcElement)
```

SrcElement may belongs to another SXml object.

**4.   bool=\_Dup(SrcSXML)**

```
java: boolean _Dup(StarSXmlClass SrcSXML)
```

**5. 12. 5 Free****1.   void \_Free()**

The script object must not be used after the function is called.

**2.   void \_Dispose()**

This function is same as \_Free.

**3.   void \_ReleaseOwner()**

This function is mainly for java and c#, which will decrease the effect of performance degrade of GC operation.

If the instance is returned to other language and do not used in future, this function can be called.

The script object must not be used after the function is called.

**5. 13 FunctionPara object****5. 13. 1 function****1.   Number=\_GetNumber()/\_Number()**

```
java: int _GetNumber()/_Number()
```

get number of parameter

**2.   Value=\_GetValue (Index)/\_Get(Index)**

```
java: Object _GetValue(int Index)
```

get value of parameter. For lua/python/ruby, can use [index]

**3.   \_Clear ()**

```
java: void _Clear()
```

clear all parameters

**4.   bool=\_SetValue (Index,Value)/\_Set(Index,Value)**

```
java: boolean _SetValue(int Index,Object Value)
```

set parameter value. For lua/python/ruby, can use [index]=Value

#### 5. **int = \_Type (Index)**

```
java: int _Type(int Index)
```

get parameter type

#### 6. **Result= \_Call (Object,FunctionName)**

```
java: Object _Call(StarObjectClass Obj ,String FunctionName)
```

The object's function must be created with function “\_CreateAtomicFunctionSimple”/”\_CreateAtomicFunction”/”\_CreateAtomicFunctionEx”.

#### 7. **void \_Free()**

The script object must not be used after the function is called.

#### 8. **void \_Dispose()**

This function is same as \_Free.

#### 9. **void \_ReleaseOwner()**

This function is mainly for java and c#, which will decrease the effect of performance degrade of GC operation.

If the instance is returned to other language and do not used in future, this function can be called.

The script object must not be used after the function is called.

### 5.14 *CommInterface object*

Running in the thread context of cle.

#### 5.14.1 *attribute*

##### 1. **CommInterface. \_MsgProc**

message callback:

```
function CommInterface_MsgProc(self,uMes,Msg)

    return;
end
```

java/c#:

```
public void _MsgProc(int uMes,Object[] Msg)
{
}
}
```

uMes's value and Msg definition

CommInterface.TCP\_ONCLOSE,Msg:{ ConnectionID }

CommInterface.TCP\_ONCONNECT,Msg:{

ServerConnectionID,ConnectionID,

BinBuf\_LocalSockAddr, BinBuf\_PeerSockAddr, Result }

CommInterface.TCP\_ONREAD,Msg:{ ConnectionID }

CommInterface.TCP\_ONWRITE,Msg:{ ConnectionID }

```

CommInterface.UDP_ONREAD,Msg:{ ConnectionID}
CommInterface.UDP_ONWRITE,Msg:{ ConnectionID}
CommInterface.HTTP_ONSTART,Msg:{ ConnectionID, (int64)FileSize, (char*)ResponseHeader}
CommInterface.HTTP_ONREAD,Msg:{ ConnectionID}
CommInterface.HTTP_ONWRITE,Msg:{ ConnectionID}
CommInterface.HTTP_ONFINISH,Msg:{ ConnectionID,Bin_ResidualData}
CommInterface.HTTP_ONPEERFINISH,Msg:{ ConnectionID}; valid for post
CommInterface.HTTP_ONERROR,Msg:{ ConnectionID, (char *)ErrorInfo}
CommInterface.HTTP_ONREQUEST,Msg:{ ConnectionID, BinBuf_PeerSockAddr, RequestType,
BoundaryNumber,      ServiceGroupIndexOrName      ,(int64)FileSize,      (char*)FileName,
(char*)ContentType,      (char*)Cookie,      BinBuf_BoundaryInfo,      (char*)RequestHeader,
BinBuf_RequestBody}
CommInterface.TIMER,Msg:{ TimerID}
BinBuf_RequestBody: is a light-weight buffer.

```

## 2. CommInterface. \_WebServerProc

```

WebServer callback:
function CommInterface_WebServerProc(self,uMes,Msg)

    return true/false(Result), true/false(continue)
end

```

```

java/c#:
public Object[] _WebServerProc(int uMes,Object[] Msg)
{
}

```

uMes's value and Msg definition

```

CommInterface.HTTP_ONREQUEST,Msg:{ ConnectionID,      PeerSockAddr,      RequestType,
BoundaryNumber, ServiceGroupIndexOrName ,FileSize, FileName, ContentType, Cookie,
BoundaryInfo, RequestHeader, RequestBody}
RequestType : CommInterface.HTTPREQUEST_GET / CommInterface.HTTPREQUEST_POST
CommInterface.HTTP_ONREAD,Msg:{ ConnectionID}
CommInterface.HTTP_ONWRITE,Msg:{ ConnectionID}

```

```

CommInterface.HTTP_ONFINISH,Msg:{ ConnectionID}
CommInterface.HTTP_ONPEERFINISH,Msg:{ ConnectionID}; valid for post
BinBuf_RequestBody: is a light-weight buffer.

```

### 5. 14. 2 function

#### 1. void \_RegMsgProc(CallBackProc)

```
java: void _RegMsgProc(String FuncName)
```

#### 2. void \_RegMsgProc\_P(CallBackProc)

for lua, ruby, python, this function is same as \_RegMsgProc

```

java: void _RegMsgProc_P(StarCommMsgInterface CallBackProc)

public interface StarCommMsgInterface{
    public void Invoke(Object CommInterface,int uMes,Object[] Msg);
}
c#: void _RegMsgProc_P(StarCommMsgInterface CallBackProc);

public delegate void StarCommMsgInterface(object CommInterface, int uMes, object[] Msg);

```

### 3. void \_RegWebServerProc(CallBackProc)

```
java: void _RegWebServerProc(String FuncName)
```

### 4. void \_RegWebServerProc\_P(CallBackProc)

for lua, ruby, python, this function is same as \_RegWebServerProc

```
java: void _RegWebServerProc_P(StarCommWebServerInterface CallBackProc)
```

```

public interface StarCommWebServerInterface{
    public Object[] Invoke(Object CommInterface,int uMes,Object[] Msg);
}
c#: void _RegWebServerProc_P(StarCommWebServerInterface CallBackProc);

```

```

public delegate object[] StarCommWebServerInterface(object CommInterface, int uMes, object[] Msg);

```

### 5. ConnectionID=\_TCPSetupServer(BufferPkgNum, LocalServerName, PortNumber)

```
java: int _TCPSetupServer(int BufferPkgNum,String LocalServerName,int PortNumber)
```

If fails,it returns 0.

### 6. ConnectionID=\_TCPSetupClient(BufferPkgNum, ServerName, PortNumber)

```
java: int _TCPSetupClient(int BufferPkgNum,String ServerName,int PortNumber)
```

If fails,it returns 0.

### 7. Size=\_TCPSend (ConnectionID,BinBuf,Offset,bool MoreData)

```
java: int _TCPSend(int ConnectionID,StarBinBufClass BinBuf,int Offset,boolean MoreData)
```

It returns the size of data has been sent. If the data is not sent completely, then application should wait message to continue send.

### 8. Size=\_TCPRecv (ConnectionID,BinBuf,Offset)

```
java: int _TCPRecv(int ConnectionID,StarBinBufClass BinBuf,int Offset)
```

return the size of data received

### 9. Size=\_TCPRecvLine (ConnectionID,BinBuf)

```
java: int _TCPRecvLine(int ConnectionID,StarBinBufClass BinBuf)
```

return the size of data received

### 10. \_TCPRelease (ConnectionID)

```
java: void _TCPRelease(int ConnectionID)
```

### 11. ConnectionID=\_UDPSetupServer(BufferPkgNum, LocalServerName, PortNumber)

```
java: int _UDPSetupServer(int BufferPkgNum,String LocalServerName,int PortNumber)
```

If fails,it returns 0.

### 12. ConnectionID=\_UDPSetupClient(BufferPkgNum)

```
java: int _UDPSendClient(int BufferPkgNum)
```

If fails,it returns 0.

**13. Size=\_UDPSend (ConnectionID,BinBuf,BinBuf\_IP)**

```
java: int _UDPSend(int ConnectionID,StarBinBufClass BinBuf,StarBinBufClass BinBuf_IP)
```

It returns the size of data has been sent. If the data is not sent completely, then application should wait message to continue send. BinBuf\_IP is filled with destination address, using function \_UDPSetSockAddr.

**14. Size=\_UDPRecv (ConnectionID,BinBuf,BinBuf\_IP)**

```
java: int _UDPRecv(int ConnectionID,StarBinBufClass BinBuf,StarBinBufClass BinBuf_IP)
```

return the size of data received

**15. \_UDPRelease (ConnectionID)**

```
java: void _UDPRelease(int ConnectionID)
```

**16. Bool=\_UDPSetSockAddr (Name,Port,BinBuf\_IP)**

```
java: boolean _UDPSetSockAddr(String Name,int Port,StarBinBufClass BinBuf_IP)
```

**17. IpString=\_GetIP (BinBuf\_IP)**

```
java: String _GetIP(StarBinBufClass BinBuf_IP)
```

**18. IpPort=\_GetPort (BinBuf\_IP)**

```
java: int _GetPort(StarBinBufClass BinBuf_IP)
```

**19. ConnectionID=\_HttpDownLoad (Url,FileName)**

```
java: int _HttpDownLoad(String Url,String FileName)
```

If fails,it returns 0.

if Url contains the request file, then FileName must be null.

**20. ConnectionID=\_HttpUpLoad (Url,FileName, FileSize,string ContentType, Boolean MultiPartFlag, string SaveFileName)**

```
java: int _HttpUpLoad(String Url,String FileName,long FileSize,String ContentType,boolean MultiPartFlag,String SaveFileName)
```

If fails,it returns 0.

SaveFileName : for download, if not NULL, is used to set the content\_type based on file ext

: for upload, if not NULL, is used to set Content-Disposition for multipart-data

it can be filename or tag&filename

: if it is filename, then [[Content-Disposition: form-data; name="file\";

filename=\"%s\"\\r\\n]]

/\* : if it is tag&filename, then [[Content-Disposition: form-data; name=\"%s\";

filename=\"%s\"\\r\\n]]

**21. ConnectionID=\_HttpDownLoadEx (Url,FileName, string RequestHeader)**

```
java: int _HttpDownLoadEx(String Url,String FileName,String RequestHeader)
```

If fails,it returns 0.

**22. ConnectionID=\_HttpUpLoadEx (Url,FileName, FileSize, string RequestHeader)**

```
java: int _HttpUpLoadEx(String Url,String FileName,long FileSize,String RequestHeader)
```

If fails,it returns 0.

**23. Size=\_HttpSend (ConnectionID,BinBuf,Offset,bool MoreData)**

```
java: int _HttpSend(int ConnectionID,StarBinBufClass BinBuf,int Offset,boolean MoreData)
```

It returns the size of data has been sent. If the data is not sent completely, then application should

wait message to continue send.

**24. Size=\_HttpRecv (ConnectionID,BinBuf,Offset)**

```
java: int _HttpRecv(int ConnectionID,StarBinBufClass BinBuf,int Offset)
```

return the size of data received

**25. \_HttpRelease (ConnectionID)**

```
java: void _HttpRelease(int ConnectionID)
```

**26. ConnectionID=\_HttpServer (LocalServerName, PortNumber,MaxPostSize)**

```
java: int _HttpServer(String LocalServerName,int PortNumber,int MaxPostSize)
```

If fails,it returns 0.

**27. RspString=\_FormatRspHeader (\*RspInfo,\*ServerInfo,\*Connection,\*ContentType, int64 ContentLength)**

```
java: String _FormatRspHeader(String RspInfo,String ServerInfo,String LConnection,String ContentType,long ContentLength)
```

```
/*FormatRspHeader("200 OK","Microsoft-IIS/5.1","Close","text/html",268);*/
```

**28. string ParaValue=\_ParsePara(Info,ParaName);**

```
java: String _ParsePara(String Info,String ParaName)
```

```
//---Para=XXX&Para2=XXX
```

**29. RspCode,RspInfo=\_GetResponseCode (BinBuf);**

```
java: Object[] _GetResponseCode(StarBinBufClass BinBuf)
```

**30. Str=\_GetResponseStr (BinBuf,string Title);**

```
java: String _GetResponseStr(StarBinBufClass BinBuf,String Title)
```

```
/*-- Title
```

```
    Date:
```

```
    Connection:
```

```
    Content-Type:
```

```
    Content-Length:
```

```
    and so on
```

**31. Lenght=\_GetResponseLength (BinBuf);**

```
java: int _GetResponseLength(StarBinBufClass BinBuf)
```

**Content- Length:**

**32. bool=\_GetResponseBody(BinBuf,BodyBinBuf);**

```
java: boolean _GetResponseBody(StarBinBufClass BinBuf,StarBinBufClass BodyBinBuf)
```

**33. ConnectionID=\_HttpLocalRequest (int RequestType, FileName, string ContentType, string Cookie, BinBuf);**

```
java: int _HttpLocalRequest(int RequestType,String FileName,String ContentType,String Cookie,StarBinBufClass BinBuf)
```

//---BinBuf:Request Body, which may be nil.

**34. ConnectionID=\_HttpLocalRequestEx (HtmlPlainText);**

```
java: int _HttpLocalRequestEx(String HtmlPlainText)
```

//---Input is complete Http request. In callback \_MsgProc, using HTTP\_READ message to get the response. The response is complete http response.

**35. TimerID=\_SetupTimer(Interval,NumberOfValid)**

```
java: int _SetupTimer(int Interval,int NumberOfValid)
```

If fails,it returns 0. Interval unit is 10ms.

### 36. **\_KillTimer (Timer)**

```
java: void _KillTimer(int Timer)
```

### 37. **\_WebServerRelease (ConnectionID)**

```
java: void _WebServerRelease(int ConnectionID)
```

Used by HttpServer to close connection.

### 38. **RspString=\_FormatRspHeaderEx**

(\*RspInfo,\*ServerInfo,\*Connection,\*ContentType, ContentLength,ExtendInfo)

```
java: String _FormatRspHeaderEx(String RspInfo,String ServerInfo,String
LConnection,String ContentType,long ContentLength,String ExtendInfo)
```

```
/*FormatRspHeaderEx("200 OK","Microsoft-IIS/5.1","Close","text/html",268, "Cookie:
aa=aa");*/
```

### 39. **RspString= \_HttpGetHeaderItem (string\_Header, ItemIndex, ItemName)**

```
java: String _HttpGetHeaderItem(String string_Header,int ItemIndex,String ItemName)
```

ItemName may be "",ItemIndex starts from 0

### 40. **RspString= \_HttpGetHeaderSubItem (string\_Item, SubItemIndex, SubItemName)**

```
java: String _HttpGetHeaderSubItem(String string_Item,int SubItemIndex,String
SubItemName)
```

SubItemName may be "",SubItemIndex starts from 0

SubItem is seperated by ";". Each SubItem has the format of name=value.

### 41. **RspString= \_HttpGetNVValue (Buf, Name)**

```
java: String _HttpGetNVValue(String Buf,String Name)
```

SubItem is seperated by ";". Each SubItem has the format of name=value.

### 42. **RspString=\_TimeToHttpTime (local time)**

```
java: String _TimeToHttpTime(StarTimeClass Tm)
```

### 43. **local time=\_HttpTimeToTime (String)**

```
java: StarTimeClass _HttpTimeToTime(String Stm)
```

### 44. **\_HttpSetCookie(string Domain,Path,Cookie,bool\_secure)**

```
java: void _HttpSetCookie(String Domain,String Path,String Cookie,boolean bool_secure)
```

### 45. **\_HttpClearCookie(Domain,Path,Cookie)**

```
java: void _HttpClearCookie(String Domain,String Path,String Cookie)
```

### 46. **String= \_HttpGetMediaType(FileName)**

```
java: String _HttpGetMediaType(String FileName)
```

### 47. **\_HttpSetMaxPostSize(ConnectionID,Size)**

```
java: void _HttpSetMaxPostSize(int ConnectionID,int Size)
```

The unit of Size is KBytes

### 48. **int PartLength,int PartOffset,string PartHeader = \_HttpGetMultiPart ( binbuf\_RequestBody, Index,int BoundaryNumber, binbuf\_Boundary)**

```
java: Object[] _HttpGetMultiPart(StarBinBufClass RequestBody,int Index,int
BoundaryNumber,StarBinBufClass Boundary)
```

Parse parameters in the request.

return value : (int64) RetMul ti PartBodySi ze, (int) PartSi ze, (string) PartStri ng

### 49. **bool= \_IsTCPConnect(ConnectionID)**

```
java: boolean _IsTCPConnect(int ConnectionID)
```



If 0 is returned for send or receive function, the function may be called to determine the connection is active or not. If the connection has been closed, it returns 0.

**50. bool= \_IsHttpConnect(ConnectionID)**

```
java: boolean _IsHttpConnect(int ConnectionID)
```

If 0 is returned for send or receive function, the function may be called to determine the connection is active or not. If the connection has been closed, it returns 0.

**51. bool= \_FileDownload (Url,LocalFileName,bool WaitFlag, Comm. Callback)**

```
java: boolean _FileDownload(String Url,String LocalFileName,boolean WaitFlag,String
Comm_Callback)
```

```
java: public void Comm_Callback(int uMes,String FileName,long MaxSize,long CurSize)
{
}
```

http/ftp download, for example:

```
comm: _FileDownload(http://127.0.0.1/index.html,"d:/index.html",true,nil)
```

```
comm: _FileDownload(http://127.0.0.1/index.html,"d:/index.html",true,comm.callback)
```

```
function comm:CallBack(uMes,FileName,MaxSize,CurSize)
```

```
end
```

uMes takes value from:

- 0 //---star download or upload
- 1 //---download process
- 2 //---finish
- 3 //---error
- 5 //---upload process

**52. bool= \_FileDownload\_P (Url,LocalFileName,bool WaitFlag, Comm. Callback)**

for lua, ruby, python, this function is same as \_FileDownload

```
java: boolean _FileDownload_P(String Url,String LocalFileName,boolean
WaitFlag,StarCommCallbackInterface CallBackProc)
```

```
public interface StarCommCallbackInterface{
    public void Invoke(Object CommInterface,int uMes,String FileName,long MaxSize,long
CurSize);
}
```

```
c#: bool _FileDownload_P(String Url, String LocalFileName, bool
WaitFlag, StarCommCallbackInterface CallBackProc);
```

```
public delegate void StarCommCallbackInterface(Object CommInterface, int
uMes, String FileName, long MaxSize, long CurSize);
```

**53. bool= \_BufDownload(Url,BinBuf,WaitFlag,Comm.Callback)**

```
java: boolean _BufDownload(String Url,StarBinBufClass BinBuf,boolean WaitFlag,String
Comm_Callback)
```

http/ftp download, download data into buf

**54. bool= \_BufDownload\_P(Url,BinBuf,WaitFlag,Comm.Callback)**

for lua, ruby, python, this function is same as \_BufDownload

```
java:    boolean    _BufDownLoad_P(String    Url,StarBinBufClass    BinBuf,boolean
WaitFlag,StarCommCallbackInterface CallBackProc)
```

```
c#:      bool      _BufDownLoad_P(String      Url , StarBinBufClass      BinBuf , bool
WaitFlag , StarCommCallbackInterface CallBackProc);
```

```
public delegate void StarCommCallbackInterface(Object CommInterface, int
```

**55. bool=\_FileUpload (Url,LocalFileName, RemoteFileName,RetBinBuf, MultiPartFlag,ContentType,WaitFlag, Comm. Callback)**

```
java:    boolean    _FileUpload(String    Url,String    LocalFileName,String
RemoteFileName,StarBinBufClass    RetBinBuf,boolean    MultiPartFlag,String
ContentType,boolean WaitFlag,String Comm_Callback)
```

http/ftp upload,RetBinBuf may be set to nil,for example:

```
comm:_FileUpload(http://127.0.0.1/index.html,"d:/index.html",
"index.html" ,None,false,"",true,nil)
```

```
comm:_FileUpload(http://127.0.0.1/index.html,"d:/index.html",
"index.html" ,None,false,"",true,comm.callback)
```

default Content-Type: application/octet-stream

**56. bool=\_FileUpload\_P (Url,LocalFileName, RemoteFileName,RetBinBuf, MultiPartFlag,ContentType,WaitFlag, Comm. Callback)**

for lua, ruby, python, this function is same as \_FileUpload

```
java:    boolean    _FileUpload_P(String    Url,String    LocalFileName,String
RemoteFileName,StarBinBufClass    RetBinBuf,boolean    MultiPartFlag,String
ContentType,boolean WaitFlag,StarCommCallbackInterface CallBackProc)
```

```
c#:      bool      _FileUpload_P(String      Url , String      LocalFileName , String
RemoteFileName , StarBinBufClass      RetBinBuf , bool      MultiPartFlag , String
ContentType , bool WaitFlag , StarCommCallbackInterface CallBackProc)
```

**57. bool=\_BufUpload (Url,BinBuf, RemoteFileName,RetBinBuf, MultiPartFlag,ContentType,WaitFlag, Comm. Callback)**

```
java:    boolean    _BufUpload(String    Url,StarBinBufClass    BinBuf,String
RemoteFileName,StarBinBufClass    RetBinBuf,boolean    MultiPartFlag,String
ContentType,boolean WaitFlag,String Comm_Callback)
```

http/ftp upload,RetBinBuf may be set to nil.

**58. bool=\_BufUpload\_P (Url,BinBuf, RemoteFileName,RetBinBuf, MultiPartFlag,ContentType,WaitFlag, Comm. Callback)**

for lua, ruby, python, this function is same as \_BufUpload

```
java:    boolean    _BufUpload_P(String    Url,StarBinBufClass    BinBuf,String
RemoteFileName,StarBinBufClass    RetBinBuf,boolean    MultiPartFlag,String
ContentType,boolean WaitFlag,StarCommCallbackInterface CallBackProc)
```

```
c#:      bool      _BufUpload_P(String      Url , StarBinBufClass      BinBuf , String
RemoteFileName , StarBinBufClass      RetBinBuf , bool      MultiPartFlag , String
ContentType , bool WaitFlag , StarCommCallbackInterface CallBackProc)
```

**59. void \_Free()**

The script object must not be used after the function is called.

**60. void \_Dispose()**

This function is same as \_Free.

**61. void \_ReleaseOwner()**

This function is mainly for java and c#, which will decrease the effect of performance degrade of GC operation.

If the instance is returned to other language and do not used in future, this function can be called.

The script object must not be used after the function is called.

### 5.14.3 Attribute Get or Set

**1. value = \_Get (Name)****2. \_Set (Name,value)**

The Name should be “\_MsgProc” or “\_WebServerProc”

### 5.15 Object garbage collect

Service group and service object will not being garbage collected.

If the object is create by script language, then the object is freed automaticlly when the corresponding script memory is garbage collected. If the object is created by cle, then the object is not freed when the corresponding script memory is garbage collected.

**1. Object. \_LockGC()/\_UnLockGC()**

```
java: void _LockGC()
```

```
java: void _UnLockGC()
```

After lock, the object is not freed when the corresponding script memory is garbage collected.

**2. Object. \_SLockGC/\_SUnlock()/\_IsSLock**

```
java: void _SLockGC/_SUnlock()
```

Strong lock, after the function is called, the corresponding script memory will not be garbage collected.

Defining object's event handler, or functions corresponding to the script language, will prevent garbage collection of objects. CLE will call \_SLockGC() automatically.

**3. Object. \_ReleaseOwnerEx**

This function should be used when cle object is create by one language, and assign script function, which will not be used in this language and output to other languages

**4. void \_SUnlockGC()**

This function is provided for check all cle object's with ReleaseOwnerEx record and trigger \_SUnLockGC function of the corresponding script interface.