



# MMS

## 客户端开发包接口 API 使用说明书

版本：V2.0

深圳天勺电力软件有限公司

深圳 中国

## 1. 开发包简介

该开发包根据已经正式发行的 IEC 61850 最新版本开发而成，包含了标准中规定的客户端所有服务。

为了最大限度地简化使用，完全封装了 IEC 61850 中的诸多细节。用户利用它作开发的时候无需了解底层细节，只需要调用接口函数完成相应的功能开放，具有方便、快捷的优点。

本开发包主要适用于 IEC61850 后台及通信管理机等产品上。

## 2. 接口描述

### 2.1 关联函数

函数声明	<b>BOOL Connect(const char *svrname, const char *IPAddr);</b>
功能	同指定 IP 地址的 61850 服务器建立关联
参数	svrname 服务器名字，在客户端是服务器的唯一标识 IPAddr 服务器的 IP 地址
返回值	关联成功返回真，否则返回假
示例	Connect（"SZTS", "192.168.0.10"）;/*"SZTS"标识要连接的服务器名字， "192.168.0.10"是服务器的 IP 地址*/

函数声明	<b>BOOL AutoConnect(const char *svrname, const char *IPAddr);</b>
功能	同指定 IP 地址的 61850 服务器建立关联，并自动完成所有的读目录服务
参数	svrname 服务器名字，在客户端是服务器的唯一标识 IPAddr 服务器的 IP 地址
返回值	关联成功返回真，否则返回假
示例	AutoConnect（"SZTS", "192.168.0.10"）;/*"SZTS"标识要连接的服务器名字， "192.168.0.10"是服务器的 IP 地址*/

函数声明	<b>BOOL SaveToFile(const char *svrname, const char *fileName);</b>
功能	将使用服务器动态建立的数据结构保存到文件中
参数	svrname 服务器名字，在客户端是服务器的唯一标识 fileName 是要保存的文件全名
返回值	关联成功返回真，否则返回假
示例	SaveToFile（"SZTS", "SZTS.txt"）;/*"SZTS"标识要连接的服务器名字， "SZTS.txt"是保存的文件*/

函数声明	<b>BOOL DetectConnect(const char *svrname, const char *IPAddr, const char *fileName);</b>
功能	在服务器配置没有发生变化时，只进行关联，利用 SaveToFile 函数保存的文件建立数据结构
参数	svrname 服务器名字，在客户端是服务器的唯一标识 IPAddr 服务器的 IP 地址 fileName 是保存的结构文件
返回值	关联成功返回真，否则返回假
示例	DetectConnect (“SZTS”, “192.168.0.10”, “SZTS.txt”);

函数声明	<b>BOOL Disconnect(const char *svrname);</b>
功能	断开关联
参数	svrname 是建立连接时的服务器名
返回值	成功时返回真，失败时返回假
示例	Disconnect(“SZTS”);

函数声明	<b>int GetServerStatus(const char *svrname);</b>
功能	取得服务器的连接状态
参数	svrname 是建立连接时的服务器名
返回值	如果关联已经异常中止，返回-2；如果是断开连接状态返回-1；如果连接正常返回 100。
示例	int status = GetServerStatus(“SZTS”);/*返回客户端和 “SZTS” 服务器的连接状态*/

函数声明	<b>BOOL ExitServer(const char *svrname, bool flag);</b>
功能	当同某一个服务器的连接不再需要时，调用该函数释放此连接中使用的所有资源
参数	svrname 是建立连接的服务器名 Flag 是一个标志：flag 为真，先断开连接，然后才释放资源，如果断开连接失败，该函数也返回失败；flag 为假时，只是尝试断开连接，无论结果怎样，都会释放资源并返回真。
返回值	成功时返回真，失败时返回假
示例	ExitServer(“SZTS”, true);/*先安全断开连接，然后释放资源*/

## 2.2 目录函数

函数声明	<b>BOOL GetSvrDirLD(const char *svrname);</b>
功能	读服务器目录中的逻辑设备（LD）
参数	svrname 是关联时给服务器命的名字
返回值	成功时返回真，否则返回假
示例	GetSvrDirLD(“SZTS”);/*读”SZTS”服务器下的 LD*/

函数声明	<b>BOOL GetLDDir(const char *svrname, const char *LDName);</b>
功能	读 LD 目录中的逻辑节点 (LN)
参数	svrname 是关联时给服务器命的名字 LDName 是一个 LD 设备的名字
返回值	成功时返回真, 失败时返回假
示例	GetLDDir("SZTS", "LD0");/*"LD0"是服务器目录中的 LD 名字*/

函数声明	<b>BOOL GetDirectory(const char *svrname, const char *ptref);</b>
功能	读逻辑节点 (LN) 或数据 (DO) 的目录
参数	svrname 是关联时给服务器命的名字 ptref 是 LN 或 DO 的路径 (如"LD1/LLN0", "LD1/LLN0\$CF\$Mod")
返回值	成功时返回真, 失败时返回假
示例	GetDirectory("SZTS", "LD1/LLN0");

函数声明	<b>BOOL GetDSName(const char *svrname, const char *LDName);</b>
功能	读取逻辑设备下的数据集名
参数	svrname 是关联时给服务器命的名字 LDName 是一个 LD 设备的名字
返回值	成功时返回真, 失败时返回假
示例	GetDSName("SZTS", "LD0");

函数声明	<b>GetDSDir(const char *svrname, const char *DSRef);</b>
功能	读数据集 (DS) 目录
参数	svrname 是关联时给服务器命的名字 DSRef 是数据集路径 (如"LD0/LLN0\$ds01")
返回值	成功时返回真, 失败时返回假
示例	GetDSDir("SZTS", "LD0/LLN0\$ds01");/*读取该数据集下的数据条目*/

## 2.3 读写函数

函数声明	<b>BOOL GetValues(const char *svrname, const char *ptref);</b>
功能	读 LN、FC 或 DO 的值, 该函数包括了读所有数据值服务和读数据值服务
参数	svrname 是关联时给服务器命的名字 ptref 是要读的某个点的路径, 如 LD0/LLN0\$CF\$Mod\$ctlModel
返回值	成功时返回真, 失败时返回假
示例	GetValues("SZTS", "LD0/LLN0\$CF\$Mod\$ctlModel");

函数声明	<b>BOOL GetMulDOValues(const char svrname, int count, const char** refArray)</b>
功能	一次读取多个数据的值
参数	svrname 是关联时给服务器命的名字 count 是数组 refArray 中元素的数目

	refArray 是一个存放路径的数组
返回值	成功时返回真，失败时返回假
示例	refArray = {"LD0/LLN0\$ST\$Mod", "LD0/LLN0\$CF\$Mod"}; GetValues("SZTS", 2, refArray);

函数声明	<b>BOOL SetDOValues(const char *svrname, const char *doref)</b>
功能	设置数据或数据属性的值
参数	svrname 是服务器名 doref 是要设置的数据或数据属性路径
返回值	成功时返回真，失败时返回假
示例	SetDOValues ("SZTS", "LD0/LLN0\$CF\$Mod") ;

注：使用该函数之前要先调用 ChangeDataValues 函数修改数据属性的值，下面所有的设置值函数遵循相同的原理。

函数声明	<b>BOOL Control(const char *svrname, const char *ref, int cv)</b>
功能	控制函数，可以进行带值选择、操作或取消功能
参数	svrname 是服务器名 ref 是控制路径 cv 是控制值
返回值	成功时返回真，失败时返回假
示例	Control ("SZTS", "LD0/LLN0\$CO\$Pos\$Oper", 1) ;/*控值闭合*/

函数声明	<b>int Asy_GetValues(const char *svrname, const char *ref);</b>
功能	异步读值函数
参数	svrname 是服务器名 ref 是路径
返回值	成功时返回一个 id 号，在回调函数中通过这个 id 号识别请求；失败时返回 0
示例	Asy_GetValues ("SZTS", "LD0/LLN0\$CO\$Pos\$Oper", 1) ;/*异步读值*/

函数声明	<b>int Asy_SetValues(const char *svrname, const char *ref);</b>
功能	异步设置数据值函数
参数	svrname 是服务器名 ref 是路径
返回值	成功时返回一个 id 号，在回调函数中通过这个 id 号识别请求；失败时返回 0
示例	Asy_SetValues ("SZTS", "LD0/LLN0\$CO\$Pos\$Oper", 1) ;/*异步读值*/

函数声明	<b>int IECControl(const char *svrname, struct _ctrlStruc *ctlPtr, int flag);</b>
功能	控制函数，可以进行带值选择、操作或取消功能
参数	svrname 是服务器名 ctlPtr 是控制信息 flag 为 1 时标记是同步控制，为 0 标记是异步控制
返回值	同步操作时请求成功返回 1，异步操作时请求成功返回 id 号

示例	
	<pre> struct _ctrlStruc {     char CntrlObj[256];//存放内容长度不能超过255     int ctlVal;     char operTm[35];/*时间格式如：1970.01.01-23:00:00.0，设置为空字符串时等同于 Operate*/     struct     {         int orCat;         char orIdent[129];//存放内容长度不能超过129     }origin;     int ctlNum;     int Test;//bool类型，值为0或1     char Check[3];//枚举类型可能的值为{"00", "01", "10", "11"} }; </pre>

## 2.4 数据集函数

函数声明	<b>BOOL GetDSValues(const char *svrname, const char *dsref);</b>
功能	读取数据集的值
参数	svrname 是关联时给服务器命的名字 dsref 是数据集路径
返回值	成功时返回真，失败时返回假
示例	GetDSValues(“SZTS”, “LD0/LLN0\$ds01”);/*成功时会刷新数据集中所有数据项的值*/

函数声明	<b>BOOL CreateDataSet(const char *svrname, int count, const char **members);</b>
功能	创建数据集
参数	svrname 是关联时给服务器命的名字 count 是 members 包含的字符串数目 members 中第一条是数据集路径，之后是数据集条目，如 char members[3][100]={“LD0/LLN0\$ds02”, “LD0/LLN0\$CF\$Mod”, “LD0/LPHD\$ST\$stVal”}
返回值	成功时返回真，失败时返回假
示例	CreateDataSet(“SZTS”, 3, members);

函数声明	<b>BOOL DeleteDataSet(const char *svrname, const char *dsref);</b>
功能	删除数据集
参数	svrname 是关联时给服务器命的名字

	dsref 是要删除的数据集路径
返回值	成功时返回真，失败时返回假
示例	DeleteDataSet (“SZTS”, “LD0/LLN0\$ds02”);

函数声明	<b>BOOL SetDSValues(const char *svrname, const char *dsref);</b>
功能	设置数据集的值
参数	svrname 是服务器名 dsref 是要设置的数据集路径
返回值	成功时返回真，失败时返回假
示例	SetDSValues(“SZTS”, “LD0/LLN0\$ds02”);

注：使用该函数之前要先调用 ChangeDataValues 函数修改数据集条目的值，由于涉及数据项比较多，使用时需要慎重，目前多数厂家的服务器不支持此功能。

函数声明	<b>int Asy_GetDSValues(const char *svrname, const char *dsref);</b>
功能	异步读取数据集的值
参数	svrname 是关联时给服务器命的名字 dsref 是数据集路径
返回值	成功时返回一个 id 号，在回调函数中通过这个 id 号识别请求；失败时返回 0
示例	

函数声明	<b>int Asy_SetDSValues(const char *svrname, const char *dsref);</b>
功能	异步设置数据集的值
参数	svrname 是服务器名 dsref 是要设置的数据集路径
返回值	成功时返回一个 id 号，在回调函数中通过这个 id 号识别请求；失败时返回 0
示例	

## 2.5 文件操作函数

函数声明	<b>BOOL GetSvrDirFile(const char *svrname);</b> //读文件目录
功能	读服务器的文件目录
参数	svrname 是服务器名
返回值	成功时返回真，失败时返回假
示例	GetSvrDirFile(“SZTS”);

函数声明	<b>BOOL GetFile(const char *svrname, const char *fileName);</b>
功能	读文件，将服务器中的文件读到客户端
参数	svrname 是服务器名 fileName 是文件全名，如”/file/config.cid”
返回值	成功时返回真，失败时返回假
示例	GetFile(“SZTS”, “file/config.cid”);

函数声明	<b>BOOL SetFile(const char *svrname,</b>
------	--

	<b>const char *dstFile, const char *srcFile);</b>
功能	设置文件，将客户端的某个文件写到服务器中
参数	svrname 是服务器名 dstFile 是客户端源文件名 srcFile 是设置后文件在服务器端的名字
返回值	成功时返回真，失败时返回假
示例	SetFile("SZTS", "x1.cid", "x2.cid");/*将客户端的 x1.cid 文件写到服务器，并且该文件在服务器中的名字是 x2.cid*/

函数声明	<b>BOOL FileDelete(const char *svrname, const char *fileName);</b>
功能	文件删除函数
参数	svrname 是服务器名; fileName 是要在服务器中删除的文件名;
返回值	成功时返回真，失败时返回假
示例	FileDelete("SZTS", "x2.cid");/*在服务器中删除 x2.cid 文件*/

函数声明	<b>int Asy_GetSvrDirFile(const char *svrname);//读文件目录</b>
功能	异步读服务器的文件目录
参数	svrname 是服务器名
返回值	成功时返回一个 id 号，在回调函数中通过这个 id 号识别请求；失败时返回 0
示例	

## 2.6 日志操作

函数声明	<b>BOOL FQueryLogByTime(const char *logName, const char *svrname, const char *ref, const char *starttime, const char *endtime);</b>
功能	读取 strattime 和 endtime 之间的日志，将日志存放在名为 logName 的文件中
参数	logName 是生成的日志文件全名 svrname 是服务器名; ref 是日志路径; starttime 是起始时间; endtime 是终止时间
返回值	成功时返回真，失败时返回假
示例	

注：如果调用该函数之前 logName 文件已经存在，函数会删除之，下面几个函数遵循相同的原理。

函数声明	<b>BOOL FQueryLogAfter(const char *logName,</b>
------	---



	<pre> const char *svrname, const char *ref, const char *entrytime, const char *entryID); </pre>
功能	读取 entryID 和 entrytime 之后的日志
参数	logName 是生成的日志文件全名; svrname 是服务器名; ref 是日志路径; entrytime 是日志产生时间; entryID 是日志的八位位串编号
返回值	成功时返回真, 失败时返回假
示例	

函数声明	<pre> BOOL QueryLogByTime(const char *svrname, const char *ref, const char *starttime, const char *endtime); </pre>
功能	读取 strattime 和 endtime 之间的日志, 将日志存放在 svrname_journal.txt 文件中
参数	含义与 FQueryLogByTime 函数中相同
返回值	成功时返回真, 失败时返回假
示例	

函数声明	<pre> BOOL QueryLogAfter(const char *svrname, const char *ref, const char *entrytime, const char *entryID); </pre>
功能	读取 entryID 和 entrytime 之后的日志, 将日志存放在 svrname_journal.txt 中
参数	含义与 FqueryLogAfter 函数中的相同
返回值	成功时返回真, 失败时返回假
示例	

函数声明	<pre> int Asy_QueryLogByTime(const char *logName, const char *svrname, const char *ref, const char *starttime, const char *endtime); </pre>
功能	与函数 FQueryLogByTime 对应的异步函数
参数	logName 是生成的日志文件全名 svrname 是服务器名; ref 是日志路径; starttime 是起始时间; endtime 是终止时间
返回值	成功时返回一个 id 号, 在回调函数中通过这个 id 号识别请求; 失败时返回 0

示例	
函数声明	<b>int Asy_QueryLogAfter(const char *logName, const char *svrname, const char *ref, const char *entrytime, const char *entryID);</b>
功能	与函数 FQueryLogAfter 对应的异步函数
参数	logName 是生成的日志文件全名; svrname 是服务器名; ref 是日志路径; entrytime 是日志产生时间; entryID 是日志的八位位串编号
返回值	成功时返回一个 id 号, 在回调函数中通过这个 id 号识别请求; 失败时返回 0
示例	

## 2.7 校时函数

函数声明	<b>BOOL StartTimeCheck(const char *IPAddr, const int period);</b>
功能	启动校时函数, 如果校时已经启动返回假
参数	IPAddr 时间服务器的地址; period 是校时周期 (单位是 ms)
返回值	成功时返回真, 失败时返回假
示例	StartTimeCheck("192.168.0.22", 2000);/*时间服务器 IP 是"192.168.0.22", 校时周期是 2000ms*/

注: 如果校时已经启动, 调用该函数时返回假。

函数声明	<b>void StopTimeCheck(void);</b>
功能	停止校时
参数	无
返回值	无
示例	

## 2.8 回调函数

/\*报告回调函数, 报告内容存放在结构体中\*/

**Typedef void (\*Svr\_RptStruc)(struct rptInfo \*rpt);**

/\*

\*控制时相关报告响应的回调函数, 参数 svrname 指示数据的来源,

\*参数 flag 为 0 指示正响应，此时参数 ctrlPtr 的 CntrlObj 变量指示出被控对象的路径（其它变量无效），

\*参数 flag 为 1 指示负响应，此时参数 errPtr 中记录了错误信息

\*/

```
Typedef void (*Svr_CtrReport)(const char *svrname, int flag, struct _ctrlStruc *ctrlPtr,
struct _lstApplErr *errPtr);
```

/\*异常中止和自动重连回调，网络中断后，动态库在 15 秒内检测出中断并利用异常中止回调给出提示，如果异常中止返回一个非 0 值，动态库将以返回值为周期自动重连，直至成功恢复关联，然后利用回调函数提示重连成功\*/

```
typedef unsigned int (*Conn_Abort)(const char *server_name);
```

/\*自动重连成功的回调函数\*/

```
typedef void (*T_callback_reconn)(const char *server_name);
```

/\*

\*异步请求回调函数

\*svrname指示服务器名,reqid指示请求的id,

\*result指示请求结果,当result为0时表示请求成功

\*/

```
typedef void (*T_callback_res)(const char *svrname, int reqid, int result);
```

这些回调函数的注册函数分别为：

```
void __declspec(dllexport) Reg_Svr_RptStruc(Svr_RptStruc rptStruc);//注册报告回调
```

```
void __declspec(dllexport) Reg_Svr_Ctr(Svr_CtrReport ctrErr); /*注册控制时报告回调*/
```

```
void __declspec(dllexport) Reg_Conn_Abort(Conn_Abort abort);//注册异常中止回调
```

```
void __declspec(dllexport) Reg_Conn_Restart(T_callback_reconn reconn);//注册自动重新连接回调
```

```
void __declspec(dllexport) Reg_Svr_Res(T_callback_res res);//带确认请求的响应
```

## 2.9 数据结构浏览

函数声明	<b>int GetChildCount(const char *svrname, const char *reference);</b>
功能	取得路径 reference 下节点数目，失败返回-1
参数	Svrname 是服务器名； reference 可以是路径，或"LD","DS","NpDS","FILE"；

	为路径时返回该路径下层节点数； 为”LD”时返回 LD 数目； 为”DS”时返回永久性数据集数目； 为”NpDS”时返回非永久性数据集数目； 为”FILE”时返回文件数目
返回值	成功时返回真，失败时返回假
示例	int count=GetChildCount(“SZTS”, “LD”);/*返回 LD 数目*/

函数声明	<b>BOOL GetChildName(char *dest,                   const int len,                   const char *svrname,                   const char *reference,                   const int pos);</b>
功能	取得 reference 下第 pos 个节点的名字并放在 dest 中
参数	dest 是目的缓冲区； len 是 dest 长度； svrname、reference 含义同上； pos 是节点位置，从 0 开始
返回值	成功时返回真，失败时返回假
示例	GetChildName(dest, 100, “SZTS”, “LD0/LLN0”, 3);/*成功后 dest 中保存 LN 下第三个功能约束的名字*/

函数声明	<b>BOOL GetChildReference (char *dest,                           const int len,                           const char *svrname,                           const char *reference,                           const int pos);</b>
功能	取得 reference 下第 pos 个节点的路径放在 dest 中
参数	含义同上
返回值	成功时返回真，失败时返回假
示例	<b>GetChildReference (dest, 100, “SZTS”, “LD0/LLN0”, 3);/*成功后 dest 中保存 LN 下第三个功能约束的路径*/</b>

函数声明	<b>BOOL GetValueType(char *dest,                   const int len,                   const char* svrname,                   const char* reference);</b>
功能	取出某一个数据属性值的类型
参数	dest 是目标缓存区，存放类型； len 是 dest 的长度； svrname 是服务器名； reference 是数据属性的路径，如”LD0/LLN0\$ST\$Mod\$stVal”
返回值	成功时返回真，失败时返回假

示例	char dest[100]; GetValueType(dest, 100, "SZTS", "LD0/LLN0\$ST\$Mod\$stVal");/* 函数成功后将在 dest 中记录该数据属性的值类型*/
----	---

函数声明	<b>BOOL GetStrValues(char *dest,                   const int len,                   const char *svrname,                   const char *reference);</b>
功能	取出某一个数据属性用字符串表示的值
参数	dest 是目标缓冲区，存放数据值； len 是 dest 长度； svrname 是服务器名； reference 是数据属性的路径，如"LD0/LLN0\$ST\$Mod\$stVal"
返回值	成功时返回真，失败时返回假
示例	char dest[100]; GetStrValues(dest, 100, "SZTS", "LD0/LLN0\$ST\$Mod\$stVal");/* 成功后将数据属性的值放在 dest 中*/

函数声明	<b>BOOL GetFormatValues(char *dest,                       const int len,                       const char *svrname,                       const char *reference)</b>
功能	取出某一个数据属性用 ASN.1 格式的值
参数	dest 是目标缓冲区，存放数据值； len 是 dest 长度； svrname 是服务器名； reference 是数据属性的路径，如"LD0/LLN0\$ST\$Mod\$stVal"
返回值	成功时返回真，失败时返回假
示例	char dest[100]; GetStrValues(dest, 100, "SZTS", "LD0/LLN0\$ST\$Mod\$stVal");/* 成功后将数据属性的值放在 dest 中*/

函数声明	<b>BOOL ChangeDataValues(const char *svrname,                       const char *reference,                       const char *src);</b>
功能	修改某一个数据属性的值，在设置数据值之前调用此函数
参数	svrname 是服务器名字 reference 是要修改值的数据属性路径 src 是用字符串表示的数据值
返回值	成功时返回真，失败时返回假
示例	ChangeDataValues("SZTS", "LD0/LLN0\$CF\$Mod\$ctlModel", "4");

函数声明	<b>BOOL GetDSReference(char *dest,                           const int len,                           const char *svrname,                           const int pos);</b>
功能	取得第 pos 个永久性数据集路径
参数	含义同上
返回值	成功时返回真，失败时返回假
示例	GetDSReference(dest, 100, "SZTS", 2);/*取得数据集路径放在 dest 中*/

函数声明	<b>BOOL GetNpDSReference(char *dest,                           const int len,                           const char *svrname,                           const int pos);</b>
功能	取得第 pos 个非永久性数据集路径
参数	含义同上
返回值	成功时返回真，失败时返回假
示例	

函数声明	<b>int GetDSMemberCount(const char *svrname,                           const char *dsref);</b>
功能	取得路径 dsref 所指示数据集下条目的数量
参数	svrname 服务器名; dsref 是数据集路径
返回值	成功时返回真，失败时返回假
示例	

函数声明	<b>BOOL GetDSMember(char *dest,                           const int len,                           const char *svrname,                           const char *dsref,                           const int pos);</b>
功能	取得数据集第 pos 个条目的路径
参数	dsref 是数据集路径
返回值	成功时返回真，失败时返回假
示例	

函数声明	<b>BOOL GetFileName(char *dest,                           const int len, const char *svrname, const int pos);</b>
功能	获取第 pos 个文件名
参数	同上
返回值	成功时返回真，失败时返回假
示例	