

Label mx 通用条码标签设计系统 SDK开发文档

外部程序通过《Label mx 通用条码标签设计系统》开放的函数接口，传递变量数据来修改标签文件里的条码、文字等内容，快速开发出带有条码标签打印功能的程序。

SDK 实现的原理：

- 1、用 Label mx 通用版设计好标签模板文件 (*.lax) 。
- 2、外部程序通过开放的函数接口，传递变量数据来修改模板文件里的条码、文字内容。
- 3、打印的过程中不弹出 Label mx 的设计环境。

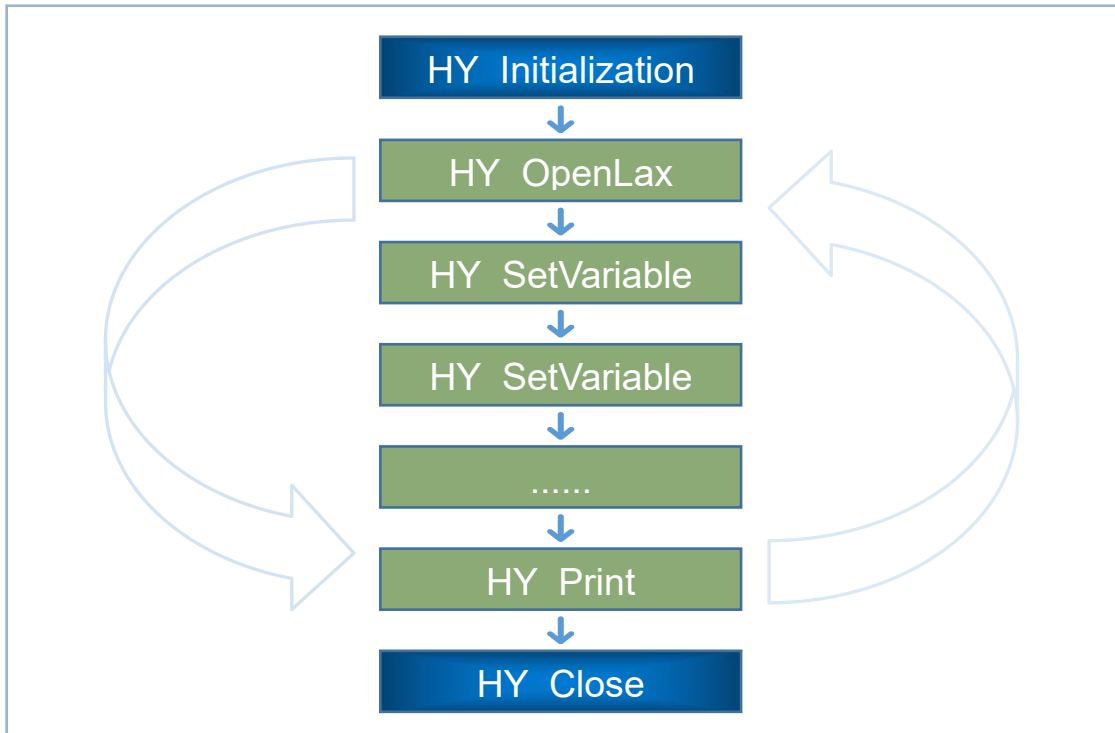
SDK 授权模式：

- 一、SDK 为独立的 DLL 文件；
- 二、支持语言：VB、VB.NET、C++、C#、Java、易语言、Delphi 等；
- 三、支持架构：C/S 程序，最后编译程序需 EXE；
- 四、SDK 一次授权终身免费使用和免费同步升级；
- 五、授权绑定的程序不需要注册授权，用户端免费使用；
- 六、《SDK 授权工具》的功能是 DLL 和开发的程序做正版绑定。
- 七、《SDK 授权工具》不限授权程序数量和绑定次数。

SDK 函数列表：

HY_Initialization	HY_SetTrim
HY_Close	HY_Print , HY_PrintEx , HY_IsPrinting
HY_OpenLax , HY_OpenLaxPw	HY_PrintSet
HY_SaveLax	HY_PrintPic
HY_GetSize_w , HY_GetSize_h , HY_SetSize	HY_Pic_bin
HY_GetTop , HY_SetTop	HY_FreeMen
HY_GetLeft , HY_SetLeft	HY_GetRGB
HY_GetPicWidth , HY_GetPicHigh	HY_SetColorTXT
HY_SetVariable	HY_SetColor1D
HY_SetVariable2	HY_SetColor2D
HY_SetVariable_UTF8	HY_SetBarcodeErrCheck
HY_SetVariable_Array	HY_GetRunDir
HY_SetVariable_Array_UTF8	HY_GetRunName
HY_SetDataSource_TXT	HY_IsReg
HY_SetDataSource_Excel	HY_MsgBoxC
HY_SetDataSource_Access	HY_SetErrTip
HY_SetDataSource_Sqlite	HY_GetVariableName
HY_SetDataSource_SQLServer	HY_GetVariableNameEx
HY_SetDataSource_MySQL	HY_GetPicData
HY_SetDataSource_DBF	HY_GetPicData_UTF8
HY_SetDataSource_Oracle	HY_SerPicDisplay
HY_GetBarType , HY_GetBarCheckCode	HY_SetPicFont

函数执行过程



举例:

```
HY_Initialization (0) //只运行一次  
HY_OpenLax ( "demo.lax" )  
HY_SetVariable ( "barcode", "6901234567892", " ", " " )  
HY_SetVariable ( "txt", "20181018001", "批号: ", " " )  
HY_Print (-2, "-1", true, false)  
//上面代码可以反复使用  
HY_Close() //只运行一次
```

- 注意:
1. 如果文字、条码设置了流水号属性，每次传递的是起始值；
 2. 图片变量传递的是“图片路径”；
 3. 不传递变量的图形按本身属性打印；
 4. 为了函数传递参数方便，可以给图形设置别名，具体方法：菜单“图层”-“设置图层别名”，或按下快捷键 F10，如果多个图形设置相同的别名，可以实现传值批量修改。起别名推荐英文字母或字母数字的形式。



函数详细说明

函数名称	<code>int HY_Initialization (int Language)</code>
函数说明	说明：初始化函数，返回 0 成功，-1 失败 参数：Language 为 显示语言 （0=简体中文，1=繁体，2=英文） 注意：本函数只在整个项目程序“开始”的时候运行 1 次，不能多次使用
函数名称	<code>void HY_Close()</code>
函数说明	说明：关闭函数 注意：本函数只在整个项目程序“结束”的时候运行 1 次，不能多次使用

注意：如果没有进行 HY_Initialization（）初始化工作，以下的函数调用会出错。

函数名称	<code>bool HY_OpenLax(string lax_files)</code>
函数说明	说明： 打开 标签模板文件 参数：lax_file 为 Label mx 设计保存的标签文件，默认后缀 lax，如："d:\test\123.lax"
函数名称	<code>bool HY_OpenLaxPw(string lax_files,string password)</code>
函数说明	说明： 打开带密码的 模板文件 参数：lax_file 为 Label mx 设计保存的标签文件，默认后缀 lax，如："d:\test\123.lax" 参数：password 为标签密码
函数名称	<code>bool HY_SaveLax(string lax_files)</code>
函数说明	说明： 保存 标签模板文件 参数：lax_file 为保存文件名，如果给 "" 空路径，则覆盖保存原打开的文件

函数名称	<code>double HY_GetSize_w()</code>
函数说明	说明：获取标签模板文件的宽度属性（mm 毫米单位） 参数：无
函数名称	<code>double HY_GetSize_h()</code>
函数说明	说明：获取标签模板文件的高度属性（mm 毫米单位） 参数：无
函数名称	<code>void HY_SetSize(double Size_w, double Size_h)</code>
函数说明	说明：设置标签模板的宽度、高度属性 参数：Size_w 为标签宽度值，Size_h 为标签高度值（mm 毫米单位）

函数名称	<code>double HY_GetTop(string PicName)</code>
函数说明	说明: 获取 图形的顶边坐标值 //0.01mm 单位 参数: PicName 为标签文件里对应的图形名称, 可以使用别名
函数名称	<code>bool HY_SetTop(string PicName, double TopY)</code>
函数说明	说明: 修改 图形的顶边坐标值 参数: PicName 为标签文件里对应的图形名称, 可以使用别名 参数: TopY 顶边值 //0.01mm 单位, 举例 11.85mm 传入值是 1185
函数名称	<code>double HY_GetLeft(string PicName)</code>
函数说明	说明: 获取 图形的左边坐标 //0.01mm 单位 参数: PicName 为标签文件里对应的图形名称, 可以使用别名
函数名称	<code>bool HY_SetLeft(string PicName, double LeftX)</code>
函数说明	说明: 修改 图形的左边坐标 参数: PicName 为标签文件里对应的图形名称, 可以使用别名 参数: LeftX 左边值 //0.01mm 单位, 举例 11.85mm 传入值是 1185

函数名称	<code>double HY_GetPicWidth(string PicName)</code>
函数说明	说明: 获取图形的宽度尺寸, 0.01mm 毫米单位 参数: PicName 为标签文件里对应的图形名称, 可以使用别名
函数名称	<code>double HY_GetPicHigh(string PicName)</code>
函数说明	说明: 获取图形的高度尺寸, 0.01mm 毫米单位 参数: PicName 为标签文件里对应的图形名称, 可以使用别名

函数名称	<code>string HY_GetVariableName ()</code>
函数说明	说明: 获取可传递数据的图形组 (文字、段落、条码、二维码、图片) 格式: 图层名;别名;图层名:null 外部使用自行分割(冒号和分号), null 代表没有别名
函数名称	<code>string HY_GetVariableNameEx ()</code>
函数说明	说明: 获取带有“别名”的图形组, 可用于管理别名接口 格式: 别名 1;别名 2;别名 3 外部使用自行分割(;分号), 枚举出有别名的全部图形。



函数名称	<code>string HY_GetPicData(string PicName)</code>
函数说明	<p>说明：获取图形的默认值（文字、段落、条码、二维码、图形路径） 返回“Error”代表 PicName 无效</p> <p>参数：PicName 为标签文件里对应的图形名称，可以使用别名</p>
函数名称	<code>void HY_GetPicData_UTF8(string PicName, *int P_Variable_data *int LpSize)</code>
函数说明	<p>说明：获取图形数据 UTF8 编码格式；</p> <p>参数：PicName 为标签文件里对应的图形名称，可以使用别名</p> <p>参数：P_Variable_data 参考 内存地址指针；HY_FreeMen 释放</p> <p>参数：LpSize 参考 内存地址长度</p>

函数名称	<code>bool HY_SetVariable(string PicName, string Variable_data, string FirstData, string EndData)</code>
函数说明	<p>说明：设置图形变量数据，注意：传递数据不改变原“存在方式”，如连接了数据库，即使传递了数据，打印还是读取数据库；而“流水号”方式传递数据是修改起始号码</p> <p>参数：PicName 为标签文件里对应的图形名称，可以使用别名</p> <p>参数：Variable_data 传递的变量数据内容；传递参数为“\DEL”，则表示原标签数据跳过为不打印，传递“-65535”保持不变</p> <p>参数：FirstData 附加字符的首部内容，文字和条码有效且勾选了“附加字符”属性</p> <p>参数：EndData 附加字符的尾部内容，传递“-65535”保持不变</p> <p>实例：<code>HY_SetVariable (“T文字1”, “2013001”, “a”, “c”)</code> <code>HY_SetVariable (“test”, “2013001”, “”, “”) // test 为别名</code> <code>HY_SetVariable (“barcode”, “\DEL”, “”, “”) // barcode 为别名</code></p>
函数名称	<code>bool HY_SetVariable_UTF8(string PicName, string Variable_data, string FirstData, string EndData)</code>
函数说明	<p>说明：设置图形变量数据，注意：传递数据不改变原“存在方式”，如连接了数据库，即使传递了数据，打印还是读取数据库；而“流水号”方式传递数据是修改“起始号码”，类型 string 全部为 UTF8 编码格式</p> <p>参数：PicName 为标签文件里对应的图形名称，可以使用别名</p> <p>参数：Variable_data 传递的变量数据内容；传递参数为“\DEL”，则表示原标签数据跳过为不打印，传递“-65535”保持不变</p> <p>参数：FirstData 附加字符的首部内容，文字和条码有效且勾选了“附加字符”属性</p> <p>参数：EndData 附加字符的尾部内容，传递“-65535”保持不变</p> <p>实例：<code>HY_SetVariable (“T文字1”, “2013001”, “a”, “c”)</code> <code>HY_SetVariable (“test”, “2013001”, “”, “”) // test 为别名</code> <code>HY_SetVariable (“barcode”, “\DEL”, “”, “”) // barcode 为别名</code></p>

函数名称	int HY_SetVariable_Array(string DataTxtArray)
函数说明	<p>说明：设置变量数据组，一次性传递多个标签数据，返回值 0 失败，大于 0 是标签数</p> <p>参数：DataTxtArray 传递的数据文本组，格式说明如下： 图层名 1{d}数据 1 {n} 图层名 2{d}数据 2 {n} {#} {d} 图层名和数据 分割符 {n} 同一个标签的多个数据 分割符 {#} 多个标签 分隔符 \DEL 不打印</p> <p>举例：HY_SetVariable_Array ("txt1{d}恒佑科技{n}txt2{d}00001{n}barcode{d}6901234567892{#}txt1{d}恒佑条码{n}txt2{d}00002{n}barcode{d}\DEL")</p>
函数名称	int HY_SetVariable_Array_UTF8(string DataTxtArray)
函数说明	<p>说明：设置变量数据组，一次性传递多个标签数据，返回值 0 失败，大于 0 是标签数 类型 string 全部为 UTF8 编码格式</p> <p>参数：DataTxtArray 传递的数据文本组，格式说明如下： 图层名 1{d}数据 1 {n} 图层名 2{d}数据 2 {n} {#} {d} 图层名和数据 分割符 {n} 同一个标签的多个数据 分割符 {#} 多个标签 分隔符 \DEL 不打印</p> <p>举例：HY_SetVariable_Array_UTF8 ("txt1{d}恒佑科技{n}txt2{d}00001{n}barcode{d}6901234567892{#}txt1{d}恒佑条码{n}txt2{d}00002{n}barcode{d}\DEL")</p>
函数名称	bool HY_SetVariable2(string PicName, string Variable_data, string FirstData, string EndData, int Fcolor_R, int Fcolor_G, int Fcolor_B, int Bcolor_R, int Bcolor_G, int Bcolor_B)
函数说明	<p>说明：设置图形变量数据，和 HY_SetVariable 的区别是多了前景背景色参数</p> <p>参数：PicName 为标签文件里对应的图形名称，可以使用别名</p> <p>参数：Variable_data 传递的变量数据内容；如果传递参数为 "\DEL"，则表示原标签数据跳过为不打印，传递 "-65535" 保持不变</p> <p>参数：FirstData 附加字符的首部内容，文字和条码有效且勾选了“附加字符”属性</p> <p>参数：EndData 附加字符的尾部内容，传递 "-65535" 保持不变</p> <p>参数：Fcolor_R, Fcolor_G, Fcolor_B 为前景色颜色，Bcolor_R, Bcolor_G, Bcolor_B 为背景色颜色，传递-1 不修改颜色</p> <p>实例：HY_SetVariable2 (" T文字 1","2013001","a","c", -1, -1, -1, -1, -1, -1) HY_SetVariable2 ("test","2013001", "", "", -1, -1, -1, -1, -1, -1) // test 为别名 HY_SetVariable2 ("barcode", "\DEL", "", "") // barcode 为别名 HY_SetVariable2 ("test","2013001", "", "", 100, 50, 50, 80, 80, 80)</p>



函数名称	<code>bool HY_SetDataSource_TXT (string PicName, string TxtFiles, string DataRange)</code>
函数说明	<p>说明：设置数据源 TXT 的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：TxtFiles 文件路径，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条</p>
函数名称	<code>bool HY_SetDataSource_Excel (string PicName, string ExcelFiles, string DataRange)</code>
函数说明	<p>说明：设置数据源 Excel 文件的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：ExcelFiles 文件路径，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from [sheet1\$] where ID=1001”</p>
函数名称	<code>bool HY_SetDataSource_Access (string PicName, string MDBFiles, string MDBPassword, string DataRange)</code>
函数说明	<p>说明：设置数据源 Access 文件的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：MDBFiles 数据库文件路径，给空表示不修改</p> <p>参数：MDBPassword 数据库密码，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from table1 where ID=1001”</p>
函数名称	<code>bool HY_SetDataSource_Sqlite (string PicName, string SqliteFiles, string DataRange)</code>
函数说明	<p>说明：设置数据源 Sqlite 的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：SqliteFiles 数据库文件路径，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from SqliteFiles where ID=1001”</p>
函数名称	<code>bool HY_SetDataSource_DBF (string PicName, string DbFiles, int iDrive, string DataRange)</code>
函数说明	<p>说明：设置数据源 Access 文件的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：DbFiles 数据库文件路径，给空表示不修改</p> <p>参数：iDrive 连接驱动，传值 0~5，六个连接驱动文本</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from table1 where ID=1001”</p>



数名称	bool HY_SetDataSource_SQLServer(string PicName, string IP, string Database, string Username, string Password, string DataRange)
函数说明	<p>说明：设置数据源 SQLServer 的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：IP 数据库的服务器地址，给空表示不修改</p> <p>参数：Database 数据库名称，给空表示不修改</p> <p>参数：Username 用户名，给空表示不修改</p> <p>参数：Password 密码，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from table1 where ID=1001”</p>
函数名称	bool HY_SetDataSource_MySQL (string PicName, string IP, string Username, string Password, string Database, string Rort, string DataRange)
函数说明	<p>说明：设置数据源 MySQL 的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：IP 数据库的服务器地址，给空表示不修改</p> <p>参数：Username 用户名，给空表示不修改</p> <p>参数：Password 密码，给空表示不修改</p> <p>参数：Database 数据库名称，给空表示不修改</p> <p>参数：Rort 端口号，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from table1 where ID=1001”</p>
函数名称	bool HY_SetDataSource_Oracle (string PicName,int DataVer, string IP, string Database, string Username, string Password, string DataRange)
函数说明	<p>说明：设置数据源 Oracle 的连接参数</p> <p>参数：PicName 为标签文件里对应的“数据源”名称，可以使用别名</p> <p>参数：DataVer 数据库版本，给空表示不修改；0= 8.05, 1= 8i, 2= 9i, 3=10g, 4=11g</p> <p>参数：IP 数据库的服务器地址，给空表示不修改</p> <p>参数：Database 数据库名称，给空表示不修改</p> <p>参数：Username 用户名，给空表示不修改</p> <p>参数：Password 密码，给空表示不修改</p> <p>参数：DataRange 记录范围，空 "" 为全部记录；“12-100” 第几条到第几条；“sql SQL 语句” 如：“sql select * from table1 where ID=1001”</p>



函数名称	<code>void HY_SetTrim(int Mode)</code>
函数说明	说明：数据库记录中空格的处理方式，只对文本、条码读取数据库时有效 参数：Mode 处理方式，0=不处理、1=删首空、2=删尾空、3=删首尾空、4=删全部空
函数名称	<code>void HY_SetPicFont(string PicName,string Name, int Size, bool Bold, bool Italic, bool UnderLine, bool Strikethrough)</code>
函数说明	说明：设置图形的字体属性，只对文本、段落、条码、二维码有效 参数：PicName 为标签文件里对应的图形名称，可以使用别名 其他参数分别是：字体名称、字体大小、是否粗体、是否倾斜，下划线、删除线。
函数名称	<code>int HY_SerPicDisplay(string PicName,bool Cbool)</code>
函数说明	说明：设置图形的隐藏状态，成功返回 0，失败返回-1 代表 PicName 无效 参数：Cbool 是否隐藏 true 为隐藏，false 为不隐藏

打印函数:

函数名称	int HY_Print(int PrintSetup, string PrinterName, bool PageWindowsShow, bool PringWinShow)
函数说明	<p>说明: 打印函数, 成功 返回 0, 打印错误或打印中止返回 -1</p> <p>参数: PrintSetup 打印设置 可为 0</p> <p>参数: PrinterName 打印机名 可为空 ""</p> <p>参数: PageWindowsShow 是否显示页面范围设置窗口 true 为显示, false 为不显示</p> <p>参数: PringWinShow 是否显示打印进度状态窗口 true 为显示, false 为不显示</p> <p>-----</p> <p>详细说明:</p> <p>PrintSetup 传值范围 (0、-1、-2、-3、-4、大于 0 的整数页数)</p> <p>0 默认标签文件的打印设置和打印数量 (如果连接了数据库, 打印数量设置无效, 为数据库连接的数量)</p> <p>-1 默认标签文件的打印设置, 打印 1 个标签。</p> <p>-2 弹出“打印设置”窗口。</p> <p>-3 弹出“打印预览”窗口。</p> <p>-4 弹出“打印预览”窗口, 禁止预览窗口上的“打印设置”按钮。</p> <p>大于 0 的整数 设置打印数量。</p> <p>PrinterName 传值范围 (“”、“-1”、“-2”、“打印机名称”)</p> <p>“”或“-1” 弹出选择打印机窗口</p> <p>“-2” 不弹出选择打印机窗口, 采用默认打印机打印</p> <p>“打印机名” 打印到指定的打印机, 如“TSC 243E”。</p> <p>PageWindowsShow 是否显示页面范围设置窗口 true 为显示, false 不显示 (下左图)</p> <p>PringWinShow 是否显示打印进度状态窗口 true 为显示, false 不显示 (下右图)</p> <div style="display: flex; justify-content: space-around; margin-top: 20px;">   </div>

函数名称	bool HY_IsPrinting()
函数说明	<p>说明: 返回打印函数 HY_Print 的打印状态, true 为正在打印, false 空闲状态, 建议打开模板文件 HY_OpenLax 之前判断一下 (任何操作数据或打印函数之前都可以判断)。</p>

函数名称	int HY_PrintEx(int PrintSetup, string PrinterName, string PageNumber, bool PringWinShow)
函数说明	<p>说明: 打印函数, 成功 返回 0, 打印错误或打印中止返回 -1</p> <p>参数: PrintSetup 打印设置 可为 0</p> <p>参数: PrinterName 打印机名 可为空 ""</p> <p>参数: PageNumber 不能为空 "" 和 "0" 值; "-1" 表示全部页码。页码范围分割符为逗号, "-"代表多少到多少页(如:1,3,5-12)</p> <p>参数: PringWinShow 是否显示打印进度状态窗口 true 为显示, false 为不显示</p> <p>-----</p> <p>详细说明:</p> <p>PrintSetup 传值范围 (0、-1、-2、-3、-4、大于0的整数页数)</p> <p>0 默认标签文件的打印设置和打印数量 (如果连接了数据库, 打印数量设置无效, 为数据库连接的数量)</p> <p>-1 默认标签文件的打印设置, 打印 1 个标签。</p> <p>-2 弹出“打印设置”窗口。</p> <p>-3 弹出“打印预览”窗口。</p> <p>-4 弹出“打印预览”窗口, 禁止预览窗口上的“打印设置”按钮。</p> <p>大于0的整数 设置打印数量。</p> <p>PrinterName 传值范围 (“”、“-1”、“-2”、“打印机名称”)</p> <p>""或“-1” 弹出选择打印机窗口</p> <p>“-2” 不弹出选择打印机窗口, 采用默认打印机打印</p> <p>“打印机名” 打印到指定的打印机, 如“TSC 243E”。</p> <p>PageNumber 传递打印页面范围 不能为空和“0”值 (如:1, 3, 5-12)</p> <p>PringWinShow 是否显示打印进度状态窗口 true 为显示, false 不显示</p>

函数名称	void HY_PrintSet(double PageLeft, double PageTop, double Spacing_H, double Spacing_V, int SingleCopy, int StartPosition, int Orientation, int Quantity)
函数说明	<p>说明: 打印设置; 打印函数 HY_Print 之前操作</p> <p>参数: PageLeft 页面左边留空距离, -1024 表示默认不改, mm 单位</p> <p>参数: PageTop 页面顶边留空距离, -1024 表示默认不改, mm 单位</p> <p>参数: Spacing_H 单个标签的水平距离, -1024 表示默认不改, mm 单位</p> <p>参数: Spacing_V 单个标签的垂直距离, -1024 表示默认不改, mm 单位</p> <p>参数: SingleCopy 单条复制份数, 1 或-1024 表示默认不改</p> <p>参数: StartPosition 打印起始位置, 1 或-1024 表示默认不改</p> <p>参数: Orientation 打印排列方向, 0 是横向 1 是纵向, -1024 表示默认不改</p> <p>参数: Quantity 标签数量, -1024 表示默认不改</p>

函数名称	<code>void HY_SetBarcodeErrCheck (bool Check)</code>
函数说明	<p>说明：设置条码打印过程中是否关闭错误校正窗口，默认 false 即发现条码错误会提示 参数：Check 是否关闭，true 真为关闭提示，关闭后发现校验码错误会自动修复，发现长度或格式不对，打印空白并跳过去继续打印</p>  

函数名称	<code>int HY_PrintPic(string Out_directory,string Out_FirstName, int Out_DPI_H, int Out_DPI_V, int Out_Color, bool Out_LenAlign)</code>
函数说明	<p>说明：打印到图片；成功返回生成的图片数量，失败返回 -1。 参数：Out_directory 输出目录 参数：Out_FirstName 输出文件名前缀，可为空 "" 参数：Out_DPI_H 输出的横向分辨率 参数：Out_DPI_V 输出的纵向分辨率 参数：Out_Color 输出位深度 (1、4、8、24、32) 参数：Out_LenAlign 文件名长度是否对齐，true 为对齐，false 不对齐</p>
函数名称	<code>int HY_Pic_bin(int* Out_bin, int Out_DPI_H, int Out_DPI_V, int Out_Color, int Out_RecNO)</code>
函数说明	<p>说明：生成内存图片；成功返回内存长度，开发者通过内存地址和长度读取图片数据；失败返回 -1。 参数：Out_bin 参考 内存地址指针；HY_FreeMen 释放 参数：Out_DPI_H 输出的横向分辨率 参数：Out_DPI_V 输出的纵向分辨率 参数：Out_Color 输出位深度 (1、4、8、24、32) 参数：Out_RecNO 记录号，1 为第一个标签图片，2 为第二个.....</p>

函数名称	void HY_FreeMen (int P_bin)
函数说明	说明: 释放内存 参数: P_bin 内存地址, HY_GetVariable_UTF8 和 HY_Pic_bin 参考获取的内存地址, 使用后要释放, 切记

设置彩码函数:

函数名称	void HY_SetColorTXT (string PicName, bool Color_Enable, int Color_Mode, string Color_Style, string Color_Group)
函数说明	说明: 设置文本彩码 参数: PicName 为标签文件里对应的 文字 图形名称, 可以使用别名 参数: Color_Enable 是否启用彩码功能, true 启用, false 不启用 参数: Color_Mode 颜色排列, 0 顺序, 1 随机, 2 系统随机 参数: Color_Style 字符范围, -65535 全部, 2-5 是字符范围 (第 2 到第 5 个字符) 参数: Color_Group 颜色值组, 逗号分隔的颜色值, 可以用 HY_GetRGB 计算
函数名称	void HY_SetColor1D (string PicName, bool Color_Enable, int Color_Mode, string Color_Style, string Color_Group)
函数说明	说明: 设置一维码彩色字符 参数: PicName 为标签文件里对应的 一维条码 图形名称, 可以使用别名 参数: Color_Enable 是否启用彩码功能, true 启用, false 不启用 参数: Color_Mode 颜色排列, 0 顺序, 1 随机, 2 系统随机 参数: Color_Style 字符范围, -65535 全部, 2-5 是字符范围 (第 2 到第 5 个字符) 参数: Color_Group 颜色值组, 逗号分隔的颜色值, 可以用 HY_GetRGB 计算
函数名称	void HY_SetColor2D (string PicName, bool Color_Enable, int Color_Mode, string Color_Style, string Color_Group, int, Color_Shape)
函数说明	说明: 设置二维码 QRCODE 彩码 参数: PicName 为标签文件里对应的 二维码 图形名称, 可以使用别名 参数: Color_Enable 是否启用彩码功能, true 启用, false 不启用 参数: Color_Mode 颜色排列, 0 顺序, 1 随机, 2 系统随机 参数: Color_Style 样式值, 二维码是 1-8 个样式值 参数: Color_Group 颜色值组, 逗号分隔的颜色值, 可以用 HY_GetRGB 计算 参数: Color_Shape 码点形状, 0 方形 1 圆形 2 菱形 3 圆角矩形
函数名称	Int HY_GetRGB (int R, int G, int B)
函数说明	说明: 返回计算 RGB 颜色值 参数: R 红; G 绿; B 蓝 (取值范围 0~255)

其他函数：

函数名称	void HY_SetErrTip (bool Enable)
函数说明	说明：设置是否关闭程序的错误提示框 参数：Enable 是否关闭，真 true 不提示，false 假 提示，推荐开发的时候设置提示，发布的时候设置不提示

函数名称	string GetRunDir ()
函数说明	说明：取运行目录 参数：无，返回 DLL 文件运行的目录，尾部不带 “\” ,如： “d:\Pub\system”

函数名称	string HY_GetRunName ()
函数说明	说明：取 DLL 全路径文件名，如： “ c:\Labelmx_sdk.dll” 参数：无

函数名称	bool HY_IsReg ()
函数说明	说明：检查 DLL 是否绑定授权，真 true 已授权，false 假 未授权 参数：无

函数名称	void HY_MsgBoxC (string Caption)
函数说明	说明：设置提示框的标题，默认是 “Label mx” 参数：Caption 标题文本，比如改成开发的程序标题： “XX 证卡打印系统”

函数名称	string HY_GetBarCheckCode (string Tbarcode,int CheckN)
函数说明	说明：计算数据校验码； 参数：Tbarcode 待计算的数据； 参数：CheckN 校验码算法 ' CheckN = 0 ' MOD 10 标准 ' CheckN = 1 ' MOD 10_MSI ' CheckN = 2 ' MOD 10_UNISON ' CheckN = 3 ' MOD 11 ' CheckN = 4 ' MOD 11,2 ' CheckN = 5 ' MOD 11,10 ' CheckN = 6 ' MSI 10,10 双校验码 ' CheckN = 7 ' MSI 11,10 双校验码 ' CheckN = 8 ' MOD 36 ' CheckN = 9 ' MOD 43 ' CheckN = 10 ' MOD 34(I,O)



函数名称	int HY_GetBarType(string PicName)
函数说明	<p>说明：获取条码、二维码的类型，对应文本可以参考下面的常量值。</p> <p>参数：PicName 为标签文件里对应的条码、二维码的图形名称，可以使用别名</p> <pre>' "Aztec Code" = 106 ' "PDF 417" = 100 ' "MicroPDF 417" = 104 ' "QR Code" = 101 ' "Data Matrix" = 102 ' "MaxiCode" = 103 ' "GS1 QR Code" = 110 ' "GS1 Data Matrix" = 111 ' "EAN-13/JAN-13" = 24 ' "EAN-8/JAN-8" = 23 ' "UPC-A" = 25 ' "UPC-E" = 26 ' "UPC-E Number system 1" = 27 ' "UPC_Supp2" = 28 ' "UPC_Supp5" = 29 ' "Code 128 A" = 15 ' "Code 128 B" = 16 ' "Code 128 C" = 17 ' "Code 128 Auto" = 33 ' "UCC/EAN(GS1)-128 A " = 30 ' "UCC/EAN(GS1)-128 B" = 31 ' "UCC/EAN(GS1)-128 C" = 32 ' "UCC/EAN(GS1)-128 Auto" = 34 ' "UCC/EAN(GS1)-128 Mulit" = 43 ' "GS1 DataBar Omnidirectional" = 50 ' "GS1 DataBar Limited" = 51 ' "GS1 DataBar Expanded" = 52 ' "Codabar (NW-7)" = 22 ' "Code 39" = 13 ' "Code 39 Extended" = 14 ' "Code 93" = 18 ' "Code 93 Extended" = 19 ' "Interleaved 2of5" = 10 ' "Industrial 2of5 " = 11 ' "Matrix 25 (矩阵 25 码)" = 12 ' "EAN-14" = 38 ' "ITF-14" = 36 ' "SCC-14 (ITF)" = 39 ' "SCC-14 (EAN)" = 40 ' "SSCC-18" = 37 ' "Code 11 (USD-8)" = 46 ' "LOGMARS" = 48 ' "Telepen" = 47 ' "Code MSI Plessey" = 20 ' "Code PostNet" = 21 ' "药品监管码" = 44 ' "电子监管码" = 35 ' "邮件条码" = 45</pre>

