PlateDSP™ 车牌识别系统

第 6 版　SDK 开发手册

深圳市普利得软件开发有限公司
SHENZHEN PLATEDSP SOFTWARE DEVELOPMENT CO., LTD.

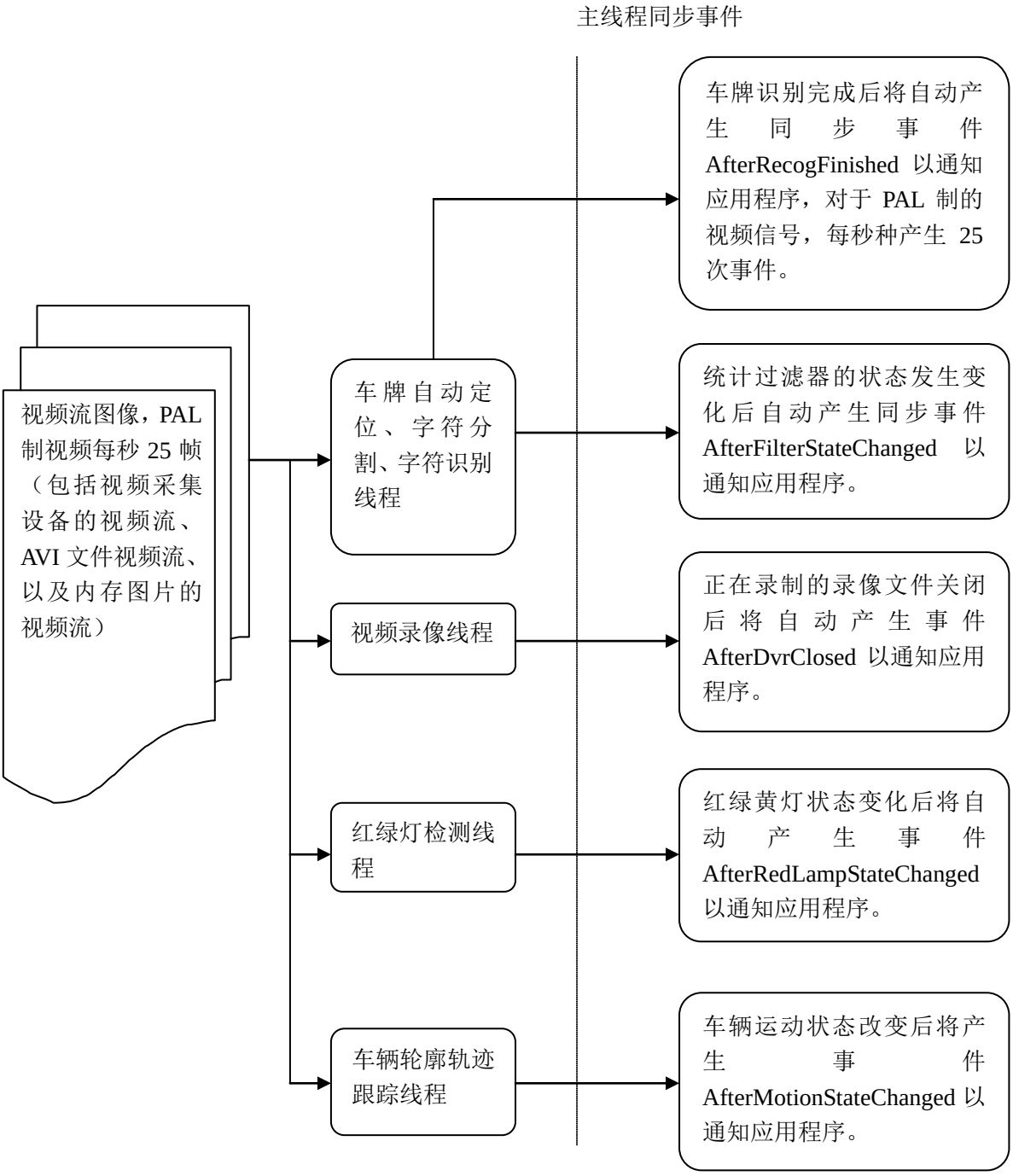
网页：www.PlateDSP.com
电话：+86 755 86219206,　86219209
地址：广东省深圳市南山区深南大道 10128 号南山软件园东塔楼 1801 室

最后修订日期：2014 年 2 月 18 日

目录

开发接口 API

“PlateDSP™ 车牌识别系统 V6”有超过 100 个函数，分为以下几个功能模块：录像文件回放模块、视频录像模块、识别核心模块、统计模块、视频管理模块、应用程序再加密、车辆轮廓运动轨迹视频跟踪、红绿灯状态视频检测、杂项。以下将按功能模块分类进行说明。



创建及销毁

DLL/SO C HDSP DSPAPI dspCreate(const char* pGuiName);
DLL/SO C++ bool Open(const char* pGuiName);
说明 创建一个 DSP 结构（C++中的类），并返回结构的指针。该指针将被其它函数引用，每次创建的结构都是独立的，可以并行运行及操作。pGuiName 指出使用 gui 的类型，可以使用以下一些字符串：

NULL	表示使用 windows 操作系统，并且回调的事件是与主线程同步的。必须使用 dspResetImageDisplayWindow 设置有效的窗口句柄 HWND。
“gtk”	表示使用 linux 操作系统中的 gtk2.0，并且回调的事件是与主线程同步的。必须使用 dspResetImageDisplayWindow 设置有效的窗口句柄 GtkDrawingArea*。
“qt”	表示使用 linux 操作系统，并且回调的事件是与主线程同步的。必须使用 dspResetImageDisplayWindow 设置有效的窗口句柄 QWidget*。
”async”	表示回调的事件是由其它线程异步调用的。不可以设置窗口句柄。

成功返回非 0 值，失败返回 NULL。

C++中的 Open 函数同时已经封装了 dspSetSyncEventCallback 函数，不需要再单独调用它设置回调了，C++类中使用重载事件函数的方式来回调函数。
安全 要求在主线程中调用。

DLL/SO C int DSPAPI dspDestroy(HDSP hdsp);
DLL/SO C++ void Close(void);
说明 销毁已创建的结构/类。hdsp 是由 dspCreate 返回的指针。
安全 要求在主线程中调用。

DLL/SO C int DSPAPI dspSetSyncEventCallback(HDSP hdsp,void* pObj,PLATEDSP_SYNC_EVET Proc);
说明 设置回调函数。hdsp 是由 dspCreate 返回的指针。pObj 指向用户定义的一个数据指针，该指针在回调 Proc 时将自动传回给用户，以帮助用户识别数据来源。成功返回 1，失败返回 0。
安全 要求在主线程中调用。

DLL/SO C int DSPAPI dspResetImageDisplayWindow(HDSP hdsp,HWND hwnd);
DLL/SO C++ int ResetImageDisplayWindow(HWND hwnd);
说明 设置图像显示以及消息传递的窗口句柄。成功返回 1，失败返回 0。
安全 要求在主线程中调用。

录像文件回放模块

OCX	VC6	long getAviCurrentPosition();
	Delphi7	function getAviCurrentPosition: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviGetCurrentPosition(HDSP hdsp);
DLL/SO	C++	int AviGetCurrentPosition(void);
说明		读取当前回放的录像文件的位置。以帧为单位。成功返回 0-N，失败返回-1。

OCX	VC6	void setAviCurrentPosition(long Pos);
	Delphi7	procedure setAviCurrentPosition(Pos: Integer); safecall;
DLL/SO	C	int DSPAPI dspAviSetCurrentPosition (HDSP hdsp,int Pos);
DLL/SO	C++	int AviSetCurrentPosition (int Pos);
说明		重新设置录像文件的回放位置。Pos 指定新的位置。以帧为单位。成功返回 1，失败返回 0 或-1。

OCX	VC6	long getAviDuration();
	Delphi7	function getAviDuration: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviGetDuration (HDSP hdsp);
DLL/SO	C++	int AviGetDuration (void);
说明		读取当前回放的录像文件的总长度。以帧为单位。成功返回 0-N，失败返回 -1。

OCX	VC6	long AviFrameStep(long Frames);
	Delphi7	function AviFrameStep(Frames: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspAviSetFrameStep (HDSP hdsp,int Frames);
DLL/SO	C++	int AviSetFrameStep (int Frames);
说明		暂停回放当前录像文件，并在当前位置的基础上前进或后退 Frames 指定的步长。Frames 的值为正，则前进，为负则后退。以帧为单位。成功返回 1，失败返回 0 或-1。

OCX	VC6	long AviIsFinished();
	Delphi7	function AviIsFinished: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviIsFinished (HDSP hdsp);
DLL/SO	C++	int AviIsFinished (void);
说明		读取当前回放的录像文件是否回放完成的状态。已完成返回 1，未完成返回 0，失败返回-1。

OCX	VC6	long AviPause();
	Delphi7	function AviPause: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviPause (HDSP hdsp);
DLL/SO	C++	int AviPause(void);
说明		暂停回放当前录像文件。成功返回 1，失败返回 0 或-1。

OCX	VC6	long AviStart(LPCTSTR aviFileName);
	BCB6	long __fastcall AviStart(BSTR aviFileName);
	Delphi7	function AviStart(const aviFileName: WideString): Integer; safecall;
DLL/SO	C	int DSPAPI dspAviStart (HDSP hdsp,const wchar_t* FileName);
DLL/SO	C++	int AviStart(const wchar_t* FileName);
说明		打开由 aviFileName 指定的视频流媒体文件并回放。是否在控件窗口中显示图像可提前调用 setImageDisplayEnabled 设置。默认为显示图像。是否在回

		放过程中对视频进行识别，可调用 setRecogEnableCount 函数来决定。不管是否对图像识别，在回放的过程中，每帧图像都将触发产生 AfterRecogFinished 事件，在该事件中，可安全地读取识别结果，也可抓取图片。成功返回 1，失败返回 0 或-1。 当 aviFileName 为空指针，或字符串长度为 0 时，调用该函数可继续运行当前已暂停或停止的回放过程。 某些版本的 Windows 操作系统没有安装 Microsoft 的 MPEG4 编解码器，无法回放该格式的录像文件，可安装光盘目录 MPEG4Codec 中的 wmpcdcs8.exe
OCX	VC6	long AviStop();
	Delphi7	function AviStop: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviStop (HDSP hdsp);
DLL/SO	C++	int AviStop(void);
说明		停止回放当前录像文件。成功返回 1，失败返回 0 或-1。

视频录像模块

OCX	VC6	long getDvrBufferFrameNum();
	Delphi7	function getDvrBufferFrameNum: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetBufferFrameNum (HDSP hdsp);
DLL/SO	C++	int DvrGetBufferFrameNum (void);
说明		读取录像机的当前缓冲区大小。单位为帧。返回值 0-N。当返回值为 0 时，表示无缓冲区，即录像方式为立即方式，不需要缓冲。当返回值大于 0 时，表示当前录像方式为缓冲方式录像。默认值为 0。

OCX	VC6	void setDvrBufferFrameNum(long FrameNum, long bHalf);
	Delphi7	procedure setDvrBufferFrameNum(FrameNum: Integer; bHalf: Integer); safecall;
说明		设置录像机的当前缓冲区大小以及图像的比例。FrameNum 指定缓冲区的大小，单位为帧。当 FrameNum 为 0 时，则使用立即方式录像，当大于 0 时为缓冲方式录像。bHalf 指定是否使用一半宽度录像。当为 0 时，使用原始图像宽度录像，当为非 0 值时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度录像。无返回值。

DLL/SO	C	int DSPAPI dspDvrSetBufferFrameNum (HDSP hdsp,int FrameNum);
DLL/SO	C++	int DvrSetBufferFrameNum (int FrameNum);
说明		设置录像机的当前缓冲区大小。FrameNum 指定缓冲区的大小，单位为帧。当 FrameNum 为 0 时，则使用立即方式录像，当大于 0 时为缓冲方式录像。成功返回 1，失败返回 0。

DLL/SO	C	int DSPAPI dspDvrSetUseHalfX (HDSP hdsp,int Params);
DLL/SO	C++	int DvrSetUseHalfX (int Params);
说明		设置录像机的图像的比例。Params 指定是否使用一半宽度录像。当为 0 时，使用原始图像宽度录像，当为 DSP_HALFY=0x02 值时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度录像。成功返回 1，失败返回 0。

OCX	VC6	long DvrCompressDlg();
	Delphi7	function DvrCompressDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrCompressDlg (HDSP hdsp);
DLL/SO	C++	int DvrCompressDlg (void);
说明		显示录像压缩格式对话框。选择成功返回非 0 值，该值为压缩器代码。失败返回 0。

OCX	VC6	long getDvrCompressor();
	Delphi7	function getDvrCompressor: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetCompressor (HDSP hdsp);
DLL/SO	C++	int DvrGetCompressor (void);
说明		读取当前录像压缩格式代码。返回值大于 0，表示为压缩器代码。-1 表示还没有设置压缩器（自动模式，系统将自动选择合适的压缩器），失败返回 0。默认值为-1。对于自动模式，将以下面的先后顺序自动寻找已安装的压缩器： ◆ Microsoft MPEG-4 Video Codec V3 (MPEG4 编解码器) ◆ Microsoft MPEG-4 Video Codec V2 (MPEG4 编解码器) ◆ Microsoft MPEG-4 Video Codec V1 (MPEG4 编解码器) ◆ DivX Codec (MPEG4 编解码器) Indeo? Video 5.10

OCX	VC6	void setDvrCompressor(long Compressor);
	Delphi7	procedure setDvrCompressor(Compressor: Integer); safecall;
DLL/SO	C	int DSPAPI dspDvrSetCompressor (HDSP hdsp,int Compressor);
DLL/SO	C++	int DvrSetCompressor (int Compressor);
说明		设置当前录像压缩格式代码。代码不能等于 0。大于 0，表示压缩器代码。-1 表示由系统自动选择合适的压缩器。某些版本的 Windows 操作系统没有安装 Microsoft 的 MPEG4 编解码器，无法进行该格式的压缩，可安装光盘目录 MPEG4Codec 中的 wmpcdcs8.exe
OCX	VC6	CString getDvrCompressorDes();
	BCB6	BSTR __fastcall getDvrCompressorDes(void);
	Delphi7	function getDvrCompressorDes: WideString; safecall;
说明		读取当前录像压缩格式名称。返回压缩器的名称字符串。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
DLL/SO	C	int DSPAPI dspDvrGetCompressorDes (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int DvrGetCompressorDes (wchar_t* pBuff,int BuffNum);
说明		读取当前录像压缩格式名称。成功返回 1 失败返回 0。 pBuff 指定目的地址，BuffNum 指定了 pBuff 的字符数。
OCX	VC6	long getDvrCurrentPosition();
	Delphi7	function getDvrCurrentPosition: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetCurrentPosition (HDSP hdsp);
DLL/SO	C++	int DvrGetCurrentPosition (void);
说明		读取当前录像机中正在录制的文件的当前位置。单位为帧。成功返回 0-N，失败返回-1。把该值与识别结果关联保存在数据库中，可利于车牌文本信息到录像资料的自动定位。
OCX	VC6	long getDvrFrameStep();
	Delphi7	function getDvrFrameStep: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetFrameStep (HDSP hdsp);
DLL/SO	C++	int DvrGetFrameStep (void);
说明		读取当前录像机的步长。单位为帧。1 表示每帧图像都必须录制；2 表示隔一帧录制一帧，依此类推。成功返回 1-N，失败返回 0 或-1。默认值为 1。
OCX	VC6	void setDvrFrameStep(long Frames);
	Delphi7	procedure setDvrFrameStep(Frames: Integer); safecall;
DLL/SO	C	int DSPAPI dspDvrSetFrameStep (HDSP hdsp,int Step);
DLL/SO	C++	int DvrSetFrameStep (int Step);
说明		设置当前录像机的步长。单位为帧。1 表示每帧图像都必须录制；2 表示隔一帧录制一帧，依此类推。在缓冲方式下，可以暂时设置为-1 的步长来暂停缓冲或立即录像，当需要缓冲或立即录像时再恢复到原来的值。
OCX	VC6	long DvrStart(LPCTSTR aviFileName);
	BCB6	long __fastcall DvrStart(BSTR aviFileName);
	Delphi7	function DvrStart(const aviFileName: WideString): Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrStart (HDSP hdsp,const wchar_t* FileName);
DLL/SO	C++	int DvrStart (const wchar_t* FileName);

说明		指定当前的录像文件为 aviFileName，（扩展名应该为.avi）。如果当前的录像方式为立即方式，则系统立即对视频流（图像采集设备输出的视频流、录像文件回放过程中输出的视频流 或 内存识别时用户传递过来的视频流）进行录像。如果当前的录像方式为缓冲方式，则系统只是设置一个录像文件名，并立即启动图像的缓冲功能，此时，图像将以先进先出的方式在给定大小的缓冲区内存储，超过缓冲区大小时，最旧的一帧图像被清除。在缓冲方式已经启动时，可以再次调用该函数重新给定文件名，而不影响正在缓存的图像。成功返回 1，失败返回 0 或-1。
OCX	VC6	long DvrStop(long bWaitFinished);
	Delphi7	function DvrStop(bWaitFinished: Integer): Integer; safecall;
说明		如果当前的录像方式为立即方式，则关闭正在录制的文件。如果当前的录像方式为缓冲方式，则停止缓冲，并把当前缓冲区中的图像压缩录制到由 DvrStart 函数设置的 AVI 文件中。bWaitFinished 指定等待的方式，0 表示关闭录像文件，但不等待就返回；1 表示关闭录像文件，并等待关闭完成才返回（由于压缩过程在多任务系统中执行，建议 bWaitFinished 设置为 0，不等待返回，这样可以提高 CPU 的使用效率）。不管是否等待，录像文件关闭完成后，将触发 AfterDvrClosed 事件。成功返回 1，失败返回 0 或-1。
OCX	VC6	long DvrStopEx(LPCTSTR aviFileName, long bWaitFinished, long bNotClearBuff);
	BCB6	long __fastcall DvrStopEx (BSTR aviFileName, long bWaitFinished, long bNotClearBuff);
	Delphi7	function DvrStopEx(const aviFileName: WideString; bWaitFinished: Integer; bNotClearBuff: Integer): Integer; safecall;
说明		如果当前的录像方式为立即方式，则关闭正在录制的文件，aviFileName 以及 bNotClearBuff 两个参数将被忽略。 如果当前的录像方式为缓冲方式，则停止缓冲，并把当前缓冲区中的图像压缩录制到由 aviFileName 指定的 AVI 文件中。bNotClearBuff 指明是否清除缓冲区中的数据，0 表示清除；1 表示不清除。 bWaitFinished 指定等待的方式，0 表示关闭录像文件，但不等待就返回；1 表示关闭录像文件，并等待关闭完成才返回（由于压缩过程在多任务系统中执行，建议 bWaitFinished 设置为 0，不等待返回，这样可以提高 CPU 的使用效率）。 不管是否等待，录像文件关闭完成后，将触发 AfterDvrClosed 事件。成功返回 1，失败返回 0 或-1。
DLL/SO	C	int DSPAPI dspDvrStop (HDSP hdsp,const wchar_t* FileName,int Params);
DLL/SO	C++	int DvrStop (const wchar_t* FileName,int Params);
说明		Params 指定了等待以及清除缓冲的模式，可以为 DSP_WAIT_FINISHED=0x04 或 DSP_NOT_CLEAR_BUFF=0x10。 如果当前的录像方式为立即方式，则关闭正在录制的文件，FileName 以及 DSP_NOT_CLEAR_BUFF 两个参数将被忽略。 如果当前的录像方式为缓冲方式，则停止缓冲，并把当前缓冲区中的图像压缩录制到由 FileName 指定的 AVI 文件中。 如果 Params 指定了 DSP_NOT_CLEAR_BUFF 位为 1，则不会清除缓冲区中的数据，否则清除。 如果 Params 指定 DSP_WAIT_FINISHED 位为 0 表示关闭录像文件，但不等待就返回；为 1 表示关闭录像文件，并等待关闭完成才返回（由于压缩过程在多任务系统中执行，建议设置为 0，不等待返回，这样可以提高 CPU 的使用效率）。

不管是否等待，录像文件关闭完成后，将触发 AfterDvrClosed 事件。成功返回 1，失败返回 0 或-1。

OCX VC6 long DvrImageCopy(long* pDesBuf, long BufSize, long Num, long bCircumgyrate90);

Delphi7 function DvrImageCopy (var pDesBuf: Integer; BufSize: Integer; Num: Integer; bCircumgyrate90: Integer): Integer; safecall;

DLL/SO C int DSPAPI dspDvrImageCopy (HDSP hdsp,void* pDesBuf,int BufSize,int Num,int bCircumgyrate90);

DLL/SO C++ int DvrImageCopy (void* pDesBuf,int BufSize,int Num,int bCircumgyrate90);

说明

只有在录像方式为缓冲方式时有效，该函数可方便冲红灯抓拍应用中抓取过程图像，一般采用 Num 等于 3，以抓取斑马线附近的三张图像的合成图片。将录像器中当前缓存的图像以 BMP 文件格式保存到由 pDesBuf 指定地址，BufSize 指定大小的内存中。成功返回已抓取图片的内存流的大小。失败返回 0 或-1。当 pDesBuf 为空指针（C++中的 NULL）或 BufSize 为 0 时，不复制数据，只返回所需内存的大小，单位为字节（Byte），当 pDesBuf 不为空指针时，BufSize 小于所需内存空间大小时，将引起调用失败。bCircumgyrate90 为 0 表示不旋转，如果为 1 表示将图像旋转 90 度，当摄像机旋转 90 度安装时，可以使用红灯更清楚。

Num 值	说明
-N	从缓冲器中取出第 N 张旧的一幅图片
-1	从缓冲器中取出最旧的一幅图片
0	从缓冲器的中间位置取出一幅图片
1	从缓冲器中取出最新的一幅图片
N（大于 1）	从缓冲器中等距离取出 N 幅图片，并从左到右拼成一张图片

为了方便一些解释性开发平台的使用，现增加一种由识别控件内部自动管理缓存的方法，该方法要求 BufSize 恒等于 4(即 long 整型变量的字节数)，pDesBuf 指向一个 long 整型数，该整型数数值范围从 0 到 15（表示内部 A 组缓冲内存块的编号）。

示例

```
//vc6，先取得所需内存流的大小，再申请内存块，再复制图像
int ImgSize = m_dsp.DvrImageCopy(NULL,0,3,0);
BYTE* pStream = new BYTE[ImgSize];
if( ImgSize==m_dsp.DvrImgaeCopy( (long*)pStream,ImgSize,3,0 ) )
{
      //成功
      ImageStreamSaveExNoWait (L “c:\\abc.jpg”, (long*)pStream);
}
delete[] pStream;
//end example

以上代码也可如下实现(2007/5/9 起提供的新方法)：
long BuffImageBlock = 0;    //使用 A 组 #0 内存块
if( m_dsp.DvrImageCopy( & BuffImageBlock, 4, 3, 0 ) > 0 )
{    //成功
      ImageStreamSaveExNoWait ( L “c:\\abc.jpg”, & BuffImageBlock );
}
```

OCX VC6 void setDvrTitle(long x,long y,LPCTSTR Title);

BCB6 void __fastcall setDvrTitle(long x,long y,BSTR Title);

Delphi7 procedure setDvrTitle(x: Integer; y: Integer; const Title: WideString); safecall;

DLL/SO C int DSPAPI dspDvrSetTitle (HDSP hdsp,int x,int y,const wchar_t* Msg);

DLL/SO C++ int DvrSetTitle (int x,int y,const wchar_t* Msg);

说明	设置当前录像机的叠加字符信息。x 指明水平方向的坐标，y 指明垂直方向的坐标，单位为象素。当 x 或 y 为负数时，清除以前设置的所有坐标处的叠加信息。Title 或 Msg 指明叠加的字符，空串表示在指定的坐标处不叠加字符（清除以前设置的对应坐标处的叠加信息）。可以设置多个不同坐标的叠加信息。为了在每帧图像上叠加实时变化的信息，可以在 AfterRecogFinished 事件中设置所要叠加的字符串。
	
	该函数还可以设置字体等信息，例如：
	setDvrTitle(0,0,L"<font:name=黑体,size=32,color=hf08080,bkcolor=h800080>");
	setDvrTitle(0,0,L"<font:size=32>");
	setDvrTitle(0,0,L"<font:name=宋体,size=24>");
	setDvrTitle(0,0,L"<font:bkcolor=0x80000000>"); // alpha=0x80
	中间没有空格。

识别核心模块

OCX	VC6	long getCarColor ();
	Delphi7	function getCarColor: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetCarColor (HDSP hdsp);
DLL/SO	C++	int GetCarColor (void);
说明		返回车身颜色的识别结果。0 表示无定义，1 表示蓝色，2 表示黑色，3 表示白色，4 表示黄色，5 表示红色，6 表示绿色。-1 表示失败。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	CString getColorName (long Color);
	BCB6	BSTR __fastcall getColorName(long Color);
	Delphi7	function getColorName(Color: Integer): WideString; safecall;
说明		说明： 返回颜色代码 Color 的字符串。返回值为：“蓝”，“黑”，“白”，“黄”，“红”，“绿”，“？”。“？”表示未知颜色。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。

DLL/SO	C	int DSPAPI dspGetColorName (HDSP hdsp,int Color,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int GetColorName (int Color,wchar_t* pBuff,int BuffNum);
说明		说明： 返回颜色代码 Color 的字符串。返回值为：“蓝”，“黑”，“白”，“黄”，“红”，“绿”，“？”。“？”表示未知颜色。 pBuff 指定目的地址，BuffNum 指定字符空间大小。

OCX	VC6	long getRecogCarColorEnable ();
	Delphi7	function getRecogCarColorEnable: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetRecogCarColorEnabled (HDSP hdsp);
DLL/SO	C++	int GetRecogCarColorEnabled (void);
说明		返回是否自动识别车身颜色。1 表示自动识别；0 表示不识别。默认值为 0。

OCX	VC6	long setRecogCarColorEnable (long bEnabled);
	Delphi7	function getRecogCarColorEnable(bEnabled: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspSetRecogCarColorEnabled (HDSP hdsp,int bEnabled);
DLL/SO	C++	int SetRecogCarColorEnabled (int bEnabled);
说明		设置并返回是否自动识别车身颜色。1 表示自动识别；0 表示不识别。默认值为 0。

OCX	VC6	long getImageByField();
	Delphi7	function getImageByField: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageGetByField (HDSP hdsp);
DLL/SO	C++	int ImageGetByField (void);
说明		返回是否自动以场的方式处理（识别及录像）图像（视频设备、回放的录

像文件、图片文件 或 内存流位图)。1 表示以场的方式处理；0 表示按原始图像的格式处理。默认值为 1。

OCX	VC6	void setImageByField(long bEnabled);
	Delphi7	procedure setImageByField (bEnabled: Integer); safecall;
DLL/SO	C	int DSPAPI dspImageSetByField (HDSP hdsp,int Params);
DLL/SO	C++	int ImageSetByField (int Params);
说明		设置是否自动以场的方式处理（识别及录像）图像（视频设备、回放的录像文件、图片文件 或 内存流位图)。1 表示以场的方式处理；0 表示按原始图像的格式处理。默认值为 1。

OCX	VC6	long setImageCompressQuality(long Quality);
	Delphi7	function setImageCompressQuality(Quality: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspImageSetCompressQuality (HDSP hdsp,int Quality);
DLL/SO	C++	int ImageSetCompressQuality (int Quality);
说明		设置 JPG 图片的压缩质量。1 最差，100 最好。图像质量越好，图片的文件大小越大。成功返回设置之前的值，失败返回 0。

OCX	VC6	long getImageDisplayEnabled();
	Delphi7	function getImageDisplayEnabled: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageGetDisplayEnabled (HDSP hdsp);
DLL/SO	C++	int ImageGetDisplayEnabled (void);
说明		返回是否显示图像（视频设备、回放的录像文件、图片文件 或 内存流位图)。默认值为 1。

OCX	VC6	void setImageDisplayEnabled (long bEnabled);
	Delphi7	procedure setImageDisplayEnabled(bEnabled: Integer); safecall;
DLL/SO	C	int DSPAPI dspImageSetDisplayEnabled (HDSP hdsp,int Params);
DLL/SO	C++	int ImageSetDisplayEnabled (int Params);
说明		设置是否显示图像（视频设备、回放的录像文件、图片文件 或 内存流位图 。

bEnabled/ Params 取值	含义
0	不显示图像
DSP_DISPLAY_CAR_IMAGE = 0x01	显示输入的图像
DSP_DISPLAY_PLATE_IMAGE = 0x02	显示车牌图像

OCX	VC6	long ImageGetIdentity();
	Delphi7	function ImageGetIdentity: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageGetIdentity (HDSP hdsp);
DLL/SO	C++	int ImageGetIdentity (void);
说明		读取当前图片的识别编号。成功返回非 0 值；否则返回 0。 该函数主要用于在事件中判断是否是同一张图片。 该值自动顺序递增。 2011/5/16 新增
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long ImageIsBest();
	Delphi7	function ImageIsBest: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageIsBest (HDSP hdsp);
DLL/SO	C++	int ImageIsBest (void);

说明		根据识别结果判断当前图象是否最好的图片。是返回 1。否则返回 0。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
示例		请参考 C:\Program Files\PlateDSP.V6\V3Examples\BCB6\Test\main.cpp

OCX	VC6	long ImagePlateIsBest ();
	Delphi7	function ImagePlateIsBest: Integer; safecall;
DLL/SO	C	int DSPAPI dspImagePlateIsBest (HDSP hdsp);
DLL/SO	C++	int ImagePlateIsBest (void);
说明		根据识别结果判断当前的车牌图象是否最好的。是返回 1。否则返回 0。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long ImageStreamCopy(long* pDesBuf, long BufSize, long bHalf);
	Delphi7	function ImageStreamCopy(var pDesBuf: Integer; BufSize: Integer; bHalf: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspImageStreamCopy (HDSP hdsp,void* pBuf,int BufSize,int Params);
DLL/SO	C++	int ImageStreamCopy (void* pBuf,int BufSize,int Params);
说明		读取当前图像帧到由 pDesBuf 指定地址，BufSize 指定大小的内存中。bHalf / Params 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。成功返回已抓取图片的内存流的大小。失败返回 0 或-1。当 pDesBuf 为空指针（C++中的 NULL）或 BufSize 为 0 时，不复制数据，只返回所需内存的大小，单位为字节（Byte），当 pDesBuf 不为空指针时，BufSize 小于所需内存空间大小时，将引起调用失败。如果已调用 setImageStreamTitle 设置了叠加的字符串，则图片中将叠加字符。为了方便一些解释性开发平台的使用，现增加一种由识别控件内部自动管理缓存的方法，该方法要求 BufSize 恒等于 4(即 long 整型变量的字节数)，pDesBuf 指向一个 long 整型数，该整型数数值范围从 0 到 15（表示内部 A 组缓冲内存块的编号）。 当 bHalf / Params 指定了 DSP_TRANS_TO_JPG=0x200 时，将压缩图像到用户定义的内存，当不使用 DSP_WAIT_FINISHED 参数时将产生 AfterCompressedJpgData 事件。

示例		//vc6，先取得所需内存流的大小，再申请内存块，再复制图像
		int ImgSize = m_dsp.ImageStreamCopy(NULL,0,0);
		BYTE* pStream = new BYTE[ImgSize];
		if(ImgSize==m_dsp.ImgaeStreamCopy((long*)pStream,ImgSize,0))
		{
		  //成功
		  ImageStreamSaveExNoWait (L “c:\\abc.jpg”, (long*)pStream);
		}
		delete[] pStream;
		//end example
		
		以上代码也可如下实现(2007/5/9 起提供的新方法)：
		long BuffImageBlock = 1; //使用 A 组 #1 内存块
		if(m_dsp.ImageStreamCopy(& BuffImageBlock, 4, 0) > 0)
		{ //成功
		  ImageStreamSaveExNoWait (L “c:\\abc.jpg”, & BuffImageBlock);

}

OCX	VC6	long ImageStreamPlateCopy(long* pDesBuf, long BufSize);
	Delphi7	function ImageStreamPlateCopy(var pDesBuf: Integer; BufSize: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspImageStreamPlateCopy (HDSP hdsp,void* pBuf,int BufSize,int Params);
DLL/SO	C++	int ImageStreamPlateCopy (void* pBuf,int BufSize,int Params);
说明		读取已成功识别的当前图像帧中的车牌牌照图像到由 pDesBuf 指定地址，BufSize 指定大小的内存中。成功返回已图片的内存流的大小。失败返回 0 或-1。当 pDesBuf 为空指针（C++中的 NULL）或 BufSize 为 0 时，不复制数据，只返回所需内存的大小，单位为字节（Byte），当 pDesBuf 不为空指针时，BufSize 小于所需内存空间大小将引起调用失败。为了方便一些解释性开发平台的使用，现增加一种由识别控件内部自动管理缓存的方法，该方法要求 BufSize 恒等于 4(即 long 整型变量的字节数)，pDesBuf 指向一个 long 整型数，该整型数数值范围从 0 到 15（表示内部 A 组缓冲内存块的编号 。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

示例		//vc6，先取得所需内存流的大小，再申请内存块，再拷图
		int ImgSize = m_dsp. ImageStreamPlateCopy (NULL,0);
		BYTE* pStream = new BYTE[ImgSize];
		if(ImgSize==m_dsp. ImageStreamPlateCopy ((long*)pStream,ImgSize))
		{
		  //成功
		  ImageStreamSaveExNoWait (L “c:\\abc.jpg”, (long*)pStream);
		}
		delete[] pStream;
		//end example
		
		以上代码也可如下实现(2007/5/9 起提供的新方法)：
		long BuffImageBlock = 2; //使用 A 组 #2 内存块
		if(m_dsp.ImageStreamPlateCopy(& BuffImageBlock, 4) > 0)
		{ //成功
		  ImageStreamSaveExNoWait (L “c:\\abc.jpg”, & BuffImageBlock);
		}

OCX	VC6	long ImageStreamPlateSave(LPCTSTR FileName);
	BCB6	long __fastcall ImageStreamPlateSave(BSTR FileName);
	Delphi7	function ImageStreamPlateSave(const FileName: WideString): Integer; safecall;
说明		保存已成功识别的当前图像帧中的车牌牌照图像到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。成功返回 1。失败返回 0 或-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long ImageStreamPlateSaveNoWait(LPCTSTR FileName);
	BCB6	long __fastcall ImageStreamPlateSaveNoWait (BSTR FileName);
	Delphi7	function ImageStreamPlateSaveNoWait (const FileName: WideString): Integer; safecall;
说明		使用线程，保存已成功识别的当前图像帧中的车牌牌照图像到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。图片保存还没有完成就将返回，这样可以提高 CPU 的利用率。成

		功返回 1。失败返回 0 或-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
DLL/SO	C	int DSPAPI dspImageStreamPlateSave (HDSP hdsp,const wchar_t* FileName,int Params);
DLL/SO	C++	int ImageStreamPlateSave (const wchar_t* FileName,int Params);
说明		使用线程，保存已成功识别的当前图像帧中的车牌牌照图像到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。图片保存还没有完成就将返回，这样可以提高 CPU 的利用率。Params 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。Prams 如果指定了位 DSP_WAIT_FINISHED=0x04 为 1，则等待保存结束再返回。 成功返回 1。失败返回 0 或-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
OCX	VC6	long ImageStreamSave(LPCTSTR FileName, long bHalf);
	BCB6	long __fastcall ImageStreamSave(BSTR FileName, long bHalf);
	Delphi7	function ImageStreamSave(const FileName: WideString; bHalf: Integer): Integer; safecall;
说明		保存当前图像帧到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。bHalf 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。成功返回 1。失败返回 0 或-1。如果已调用 setImageStreamTitle 设置了叠加的字符串，则图片中将叠加字符。
OCX	VC6	long ImageStreamSaveNoWait(LPCTSTR FileName, long bHalf);
	BCB6	long __fastcall ImageStreamSaveNoWait (BSTR FileName, long bHalf);
	Delphi7	function ImageStreamSaveNoWait (const FileName: WideString; bHalf: Integer): Integer; safecall;
说明		使用线程，保存当前图像帧到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。bHalf 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。图片保存还没有完成就将返回，这样可以提高 CPU 的利用率。成功返回 1。失败返回 0 或-1。如果已调用 setImageStreamTitle 设置了叠加的字符串，则图片中将叠加字符。
DLL/SO	C	int DSPAPI dspImageStreamSave (HDSP hdsp,const wchar_t* FileName,int Params);
DLL/SO	C++	int ImageStreamSave (const wchar_t* FileName,int Params);
说明		保存当前图像帧到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。

Params 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。
Prams 如果指定了位 DSP_WAIT_FINISHED=0x04 为 1，则等待保存结束再返回。
成功返回 1。失败返回 0 或-1。
如果已调用 setImageStreamTitle 设置了叠加的字符串，则图片中将叠加字符。

OCX	VC6	long ImageStreamSaveEx (LPCTSTR FileName, long * pBmpStream);
	BCB6	long __fastcall ImageStreamSaveEx(BSTR FileName, long * pBmpStream);
	Delphi7	function ImageStreamSaveEx(const FileName: WideString; var pBmpStream: Integer): Integer; safecall;
说明		保存由 pBmpStream 指定的 BMP 格式的内存位图流，到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。成功返回 1。失败返回 0 或-1。可以使用该函数保存由 ImageStreamCopy 或 ImageStreamPlateCopy 复制出来的内存位图到 JPG 文件中。为了方便一些解释性开发平台的使用，现增加一种由识别控件内部自动管理缓存的方法，该方法要求 pBmpStream 指向一个 long 整型数，该整型数数值范围从 0 到 7（表示内部缓冲内存块的编号）。

OCX	VC6	long ImageStreamSaveExNoWait (LPCTSTR FileName, long * pBmpStream);
	BCB6	long __fastcall ImageStreamSaveExNoWait (BSTR FileName, long * pBmpStream);
	Delphi7	function ImageStreamSaveExNoWait (const FileName: WideString; var pBmpStream: Integer): Integer; safecall;
说明		使用线程，保存由 pBmpStream 指定的 BMP 格式的内存位图流，到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。图片保存还没有完成就将返回，这样可以提高 CPU 的利用率。成功返回 1。失败返回 0 或-1。可以使用该函数保存由 ImageStreamCopy 或 ImageStreamPlateCopy 复制出来的内存位图到 JPG 文件中。为了方便一些解释性开发平台的使用，现增加一种由识别控件内部自动管理缓存的方法，该方法要求 pBmpStream 指向一个 long 整型数，该整型数数值范围从 0 到 7（表示内部缓冲内存块的编号）。

DLL/SO	C	int DSPAPI dspImageStreamSaveEx (HDSP hdsp,const wchar_t* FileName,const void* pStream,int Params);
DLL/SO	C++	int ImageStreamSaveEx (const wchar_t* FileName,const void* pStream,int Params);
说明		保存由 pStream 指定的 BMP 格式的内存位图流，到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG 图片格式。 Params 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。 Prams 如果指定了位 DSP_WAIT_FINISHED=0x04 为 1，则等待保存结束再返回。 成功返回 1。失败返回 0 或-1。 可以使用该函数保存由 ImageStreamCopy 或 ImageStreamPlateCopy 复制出来的内存位图到 JPG 文件中。为了方便一些解释性开发平台的使用，现增加

一种由识别控件内部自动管理缓存的方法，该方法要求 pStream 指向一个 long 整型数，该整型数数值范围从 0 到 7（表示内部缓冲内存块的编号）。

OCX	VC6	void setImageStreamTitle(long x,long y,LPCTSTR Title);
	BCB6	void __fastcall setImageStreamTitle(long x,long y,BSTR Title);
	Delphi7	procedure setImageStreamTitle(x: Integer; y: Integer; const Title: WideString); safecall;
DLL/SO	C	int DSPAPI dspImageStreamSetTitle (HDSP hdsp,int x,int y,const wchar_t* Msg);
DLL/SO	C++	int ImageStreamSetTitle (int x,int y,const wchar_t* Msg);

说明		设置当前图片的叠加字符信息。x 指明水平方向的坐标，y 指明垂直方向的坐标，单位为象素。当 x 或 y 为负数时，清除以前设置的所有坐标处的叠加信息。Title 可 Msg 指明叠加的字符，空串表示在指定的坐标处不叠加字符（清除以前设置的对应坐标处的叠加信息）。可以设置多个不同坐标的叠加信息。
		该函数还可以设置字体等信息，例如：
		setImageStreamTitle(0,0,L"<font:name= 黑 体 ,size=32,color=hf08080,bkcolor=h800080>");
		setImageStreamTitle (0,0,L"<font:size=32>");
		setImageStreamTitle (0,0,L"<font:name=宋体,size=24>");
		setImageStreamTitle (0,0,L"<font:bkcolor=0x80000000>"); //alpha=0x80
		中间没有空格。

OCX	VC6	long BufferImageStream (long Id);
	Delphi7	function BufferImageStream (Id: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspBufferImageStream (HDSP hdsp,unsigned int Id);
DLL/SO	C++	int BufferImageStream (unsigned int Id);
说明		缓冲当前图像帧到由 Id 指定序号的 B 组内存块中，成功返回 1，失败返回 0。Id 数值范围从 0 到 255（表示内部 B 组缓冲内存块的编号）。该函数类似于 ImageStreamCopy，但不进行图像格式转换，效率高。

OCX	VC6	long BufferCopy (long* pDesBuf, long BufSize, long Id, long Params);
	Delphi7	function BufferCopy (var pDesBuf: Integer; BufSize: Integer; Id: Integer; Params: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspBufferCopy (HDSP hdsp,void* pDesBuf,int BufSize,unsigned int Id,int Params);
DLL/SO	C++	int BufferCopy (void* pDesBuf,int BufSize,unsigned int Id,int Params);
说明		转换（如果必要转换为 RGB 格式）并复制由 Id 指定编号（0-255）的 B 组内存块的图像到由 pDesBuf 指定地址，BufSize 指定大小的内存中。 Params 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。成功返回已抓取图片的内存流的大小。失败返回 0 或-1。当 pDesBuf 为空指针（C++中的 NULL）或 BufSize 为 0 时，不复制数据，只返回所需内存的大小，单位为字节（Byte），当 pDesBuf 不为空指针时，BufSize 小于所需内存空间大小时，将引起调用失败。如果已调用 setImageStreamTitle 设置了叠加的字符串，则图片中将叠加字符。为了方便一些解释性开发平台的使用，增加一种由识别控件内部自动管理缓存的方法，该方法要求 BufSize 恒等于 4(即 long 整型变量的字节数)，pDesBuf 指向一个 long 整型数，该整型数数值范围从 0 到 15（表示内部 A 组缓冲内存块的编号）。当 Params 指定的 DSP_BUFF_TO_BUFF=0x80 位为 1 时，pDesBuf 指向的整数范围从 0 到 255

（B 组内存块）。
当 bHalf / Params 指定了 DSP_TRANS_TO_JPG=0x200 时，将压缩图像到用户定义的内存，当不使用 DSP_WAIT_FINISHED 参数时将产生 AfterCompressedJpgData 事件。

OCX VC6 long BufferSave (long Id, long Params, LPCTSTR FileName ,long bWaitFinished);
BCB6 long __fastcall BufferSave (long Id, long Params, BSTR FileName ,long bWaitFinished);
Delphi7 function BufferSave (Id: Integer; Params: Integer; const FileName: WideString; bWaitFinished: Integer): Integer; safecall;
DLL/SO C int DSPAPI dspBufferSave (HDSP hdsp,unsigned int Id,int Params,const wchar_t* FileName,int bWaitFinished);
DLL/SO C++ int BufferSave (unsigned int Id,int Params,const wchar_t* FileName,int bWaitFinished);
说明
保存由 Id 指定序号（0-255）的 B 组内存图，到由 FileName 指定的图片文件中，文件的格式由文件名扩展名指定，可以是.BMP/.JPG/图片格式。
Params 指定是否使用原图的一半宽度，当为 0 时，使用原始图像宽度抓图，当 DSP_HALFX=0x01 位为 1 时，且原始图像的宽度大于 384 点宽，则以原始图像一半的宽度抓图。当 DSP_AUTO_ZOOM_Y=0x100 位为 1 时，且图像的高度可以拉伸时，则图像高度自动加倍。
bWaitFinished 指定是否等待保存结束再返回。
成功返回 1。失败返回 0 或-1。
可以使用该函数保存由 BufferImageStream / BufferCopy / ImageStreamCopy 或 ImageStreamPlateCopy 复制出来的内存位图到 JPG 文件中。该函数类似于 ImageStreamSaveEx。

OCX VC6 long WaitRecogFinished (long Tick);
Delphi7 function WaitRecogFinished (Tick: Integer): Integer; safecall;
DLL/SO C int DSPAPI dspWaitRecogFinished (HDSP hdsp,unsigned int Tick);
DLL/SO C++ int WaitRecogFinished (unsigned int Tick);
说明 以消息循环的方式等待当前的识别过程结束。成功返回 1,失败返回 0。

Tick	说明
0	查询是否结束。已结束返回 1，否则返回 0.
(-1)	等待并查询是否结束。已结束返回 1.
其它 N	最多等待 N 毫秒并查询是否结束。已结束返回 1. 否则返回 0.

OCX VC6 long WaitFileSaved ((long Tick);
Delphi7 function WaitFileSaved (Tick: Integer): Integer; safecall;
DLL/SO C int DSPAPI dspWaitFileSaved (HDSP hdsp,unsigned int Tick);
DLL/SO C++ int WaitFileSaved (unsigned int Tick);
说明 以消息循环的方式等待线程保存的文件过程结束。成功返回 1,失败返回 0。

Tick	说明
0	查询是否结束。已结束返回 1，否则返回 0.
(-1)	等待并查询是否结束。已结束返回 1.
其它 N	最多等待 N 毫秒并查询是否结束。已结束返回 1. 否则返回 0.

OCX VC6 long getPlateBrightness();
Delphi7 function getPlateBrightness: Integer; safecall;
DLL/SO C int DSPAPI dspGetPlateBrightness (HDSP hdsp);

DLL/SO	C++	int GetPlateBrightness (void);
说明		读取已成功识别的车牌的亮度值，成功返回 1-255 的值，255 为最高值。当没有识别结果时，返回当前视频的亮度值的负数，以帮助自动调节图像亮度，返回值为-1 到-255, -255 为最高值(该功能目前只支持某些高清图像采集设备)。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long getPlateColor();
	Delphi7	function getPlateColor: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateColor (HDSP hdsp);
DLL/SO	C++	int GetPlateColor (void);
说明		读取已成功识别的车牌的颜色代码。0 表示无定义，1 表示蓝色，2 表示黑色，3 表示白色，4 表示黄色，5 表示红色，6 表示绿色。失败返回-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	CString getPlateColorName();
	BCB6	BSTR __fastcall getPlateColorName(void);
	Delphi7	function getPlateColorName: WideString; safecall;
说明		读取已成功识别的车牌的颜色字符串。返回值为：“蓝”，“黑”，“白”，“黄”，“红”，“绿”，“？”。“？”表示未知颜色。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long getPlateCount();
	Delphi7	function getPlateCount: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateCount (HDSP hdsp);
DLL/SO	C++	int GetPlateCount (void);
说明		读取已成功识别的车牌的计数值，出现的次数越多，计数值越高。当设置为允许统计时，返回值表示统计队列中该车牌的计数值，禁止统计时返回值为 1。失败返回-1。 建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long getPlateDir();
	Delphi7	function getPlateDir: Integer; safecall;
说明		读取车辆运动方向识别结果，正数表示车牌从上到下运动，负数表示车牌从下到上运动。绝对值越高表示方向识别可靠性越高。0 值表示无法识别方向。调用 setStatEnabled 函数启动统计功能可使方向的识别更准确。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long getPlateLocatedRect(long* pLeft, long* pTop, long* pRight, long* pBottom);
	Delphi7	function getPlateLocatedRect(var pLeft: Integer; var pTop: Integer; var pRight: Integer; var pBottom: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateLocatedRect (HDSP hdsp,long* pLeft,long* pTop,long* pRight,long* pBottom);

DLL/SO	C++	int GetPlateLocatedRect (long* pLeft,long* pTop,long* pRight,long* pBottom);
说明		读取已成功识别的车牌的图像在原始图像中的位置。pLeft, pTop, pRight, pBottom 为 long*型指针，分别指向左，上，右，下坐标的变量。单位为像素点。失败返回-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

示例		//vc6
		long Left,Top,Right,Bottom;
		if(getPlateLocatedRect(&Left,&Top,&Right,&Bottom)==1)
		{
		  //成功
		}

OCX	VC6	CString getPlateNumber();
	BCB6	BSTR __fastcall getPlateNumber(void);
	Delphi7	function getPlateNumber: WideString; safecall
说明		读取已成功识别的车牌号码字符串。失败返回空串。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
DLL/SO	C	int DSPAPI dspGetPlateNumber (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int GetPlateNumber (wchar_t* pBuff,int BuffNum);
说明		读取已成功识别的车牌号码字符串。 pBuff 指定目的地址，BuffNum 指定目的地的字符空间数。 成功返回 1，失败返回 0。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
OCX	VC6	long getPlateReliability();
	Delphi7	function getPlateReliability: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateReliability (HDSP hdsp);
DLL/SO	C++	int GetPlateReliability (void);
说明		读取已成功识别的车牌置信度。返回值 0-100。100 表示最可信。失败返回-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
OCX	VC6	long getPlateReliabilityByChar(long Index);
	Delphi7	function getPlateReliabilityByChar(Index: Integer): Integer; safecall;
说明		读取已成功识别的车牌号码中由 Index 指定的字符的置信度。Index 为 0 表示第 1 个字符，依此类推。返回值 0-100。100 表示最可信。失败返回-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
DLL/SO	C	int DSPAPI dspGetPlateReliabilityByChar (HDSP hdsp,unsigned char* pBuff,int BuffNum);
DLL/SO	C++	int GetPlateReliabilityByChar (unsigned char* pBuff,int BuffNum);
说明		读取已成功识别的车牌号码中每个字符的置信度（0-100）。 pBuff 指定目的地，BuffNum 指定目的的字节数。

		成功返回字符的个数，失败返回 0。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
OCX	VC6	long getPlateSpeed (long* pSpeedX,long* pSpeedY);
	Delphi7	function getPlateSpeed (var pSpeedX: Integer; var pSpeedY: Integer): Integer; safecall;
说明		读取已成功识别的车辆的运动速度（即视频测速功能）。单位为：像素点/每分钟。成功返回 1，失败返回 0 或-1。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
DLL/SO	C	int DSPAPI dspGetPlateSpeed (HDSP hdsp,int * px,int * py);
DLL/SO	C++	int GetPlateSpeed (int * px,int * py);
说明		读取已成功识别的车辆的运动速度（即视频测速功能）。单位为：像素点/每分钟。 px,py 指定的地址中将分别得到 X 和 Y 方向的速度值。 返回值为正数表示车牌从上到下运动，负数表示车牌从下到上运动。绝对值越高表示方向识别可靠性越高。0 值表示无法识别方向。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
OCX	VC6	long getPlateTypeId();
	Delphi7	function getPlateTypeId: Integer; safecall;
说明		读取已成功识别的车牌类型代码。成功时返回非 0 值,。失败返回 0 或-1。 返回值请参考 getRecogCfgUseTemplate。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
OCX	VC6	CString getPlateTypeName();
	BCB6	BSTR __fastcall getPlateTypeName(void);
	Delphi7	function getPlateTypeName: WideString; safecall;
说明		读取已成功识别的车牌类型。成功时返回有：“民用车牌(92 式)”、“民用货车尾牌(双行)”、“民用车牌(2002 个性化)”、“警车车牌(*警)”、“武警车牌(WJ*)”、“军用车牌(2004 式)”。失败返回空串。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。
DLL/SO	C	int DSPAPI dspGetPlateTypeName (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int GetPlateTypeName (wchar_t* pBuff,int BuffNum);
说明		读取已成功识别的车牌类型。成功时返回有：“民用车牌(92 式)”、“民用货车尾牌(双行)”、“民用车牌(2002 个性化)”、“警车车牌(*警)”、“武警车牌(WJ*)”、“军用车牌(2004 式)”。 pBuff 指定目的地址，BuffNum 指定目的地的字符空间大小。 成功返回类型编号，失败返回 0。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX VC6 long getPlateTypeIdEx();
Delphi7 function getPlateTypeIdEx: Integer; safecall;
说明 读取已成功识别的营运车辆标记的代码。成功时返回非 0 值。失败返回 0 或-1。

代码	意义
1	深圳出租

安全 建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX VC6 CString getPlateTypeName Ex();
BCB6 BSTR __fastcall getPlateTypeNameEx(void);
Delphi7 function getPlateTypeNameEx: WideString; safecall;
说明 读取已成功识别的营运车辆标记类型。成功时返回有：“深圳出租”。失败返回空串。
由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
安全 建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX VC6 long getRecogCfgMinReliability();
Delphi7 function getRecogCfgMinReliability: Integer; safecall;
DLL/SO C int DSPAPI dspRecogCfgGetMinReliability (HDSP hdsp);
DLL/SO C++ int RecogCfgGetMinReliability (void);
说明 读取识别配置参数“最低置信度”。返回值 0-100。100 表示最要求 100%准确才可输出识别结果(实际上不会有 100%准确的车牌识别结果)。默认值为 75。

OCX VC6 void setRecogCfgMinReliability(long Reliability);
Delphi7 procedure setRecogCfgMinReliability(Reliability: Integer); safecall;
DLL/SO C int DSPAPI dspRecogCfgSetMinReliability (HDSP hdsp,int Params);
DLL/SO C++ int RecogCfgSetMinReliability (int Params);
说明 设置识别配置参数“最低置信度”。100 表示要求 100%准确才可输出识别结果。

OCX VC6 long getRecogCfgPlateMaxHeight();
Delphi7 function getRecogCfgPlateMaxHeight: Integer; safecall;
说明 读取识别配置参数“车牌最大高度”。返回值 0-N。默认值为 60 点高。

OCX VC6 void setRecogCfgPlateMaxHeight(long Height);
Delphi7 procedure setRecogCfgPlateMaxHeight(Height: Integer); safecall;
说明 设置识别配置参数“车牌最大高度”。用户一般不应设置该值。

OCX VC6 long getRecogCfgPlateMaxWidth();
Delphi7 function getRecogCfgPlateMaxWidth: Integer; safecall;
说明 读取识别配置参数“车牌最大宽度”。返回值 0-N。默认值为 220 点宽。

OCX VC6 void setRecogCfgPlateMaxWidth(long Width);
Delphi7 procedure setRecogCfgPlateMaxWidth(Width: Integer); safecall;
说明 设置识别配置参数“车牌最大宽度”。用户一般不应设置该值。

OCX	VC6	long getRecogCfgPlateMinHeight();
	Delphi7	function getRecogCfgPlateMinHeight: Integer; safecall;
说明		读取识别配置参数“车牌最小高度”。返回值 0-N。默认值为 20 点高。
OCX	VC6	void setRecogCfgPlateMinHeight(long Height);
	Delphi7	procedure setRecogCfgPlateMinHeight(Height: Integer); safecall;
说明		设置识别配置参数“车牌最小高度”。用户一般不应设置该值。
OCX	VC6	long getRecogCfgPlateMinWidth();
	Delphi7	function getRecogCfgPlateMinWidth: Integer; safecall;
说明		读取识别配置参数“车牌最小宽度”。返回值 0-N。默认值为 80 点宽。
OCX	VC6	void setRecogCfgPlateMinWidth(long Width);
	Delphi7	procedure setRecogCfgPlateMinWidth(Width: Integer); safecall;
说明		设置识别配置参数“车牌最小宽度”。用户一般不应设置该值。
DLL/SO	C	int DSPAPI dspRecogCfgSetPlateRange (HDSP hdsp,long MinWidth,long MinHeight,long MaxWidth,long MaxHeight);
DLL/SO	C++	int RecogCfgSetPlateRange (long MinWidth,long MinHeight,long MaxWidth,long MaxHeight);
说明		设置识别配置参数,以指定车牌的最小值以及最大值。 用户一般不应设置该值。
DLL/SO	C	int DSPAPI dspRecogCfgGetPlateRange (HDSP hdsp,long* pMinWidth,long* pMinHeight,long* pMaxWidth,long* pMaxHeight);
DLL/SO	C++	int RecogCfgGetPlateRange (long* pMinWidth,long* pMinHeight,long* pMaxWidth,long* pMaxHeight);
说明		读取识别配置参数,以得到车牌的最小值以及最大值。
OCX	VC6	CString getRecogCfgProvince();
	BCB6	BSTR __fastcall getRecogCfgProvince(void);
	Delphi7	function getRecogCfgProvince: WideString; safecall;
说明		读取识别配置参数“车牌号码前辍”字符串。没有设置返回空串。默认值为空串。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
DLL/SO	C	int DSPAPI dspRecogCfgGetProvince (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int RecogCfgGetProvince (wchar_t* pBuff,int BuffNum);
说明		读取识别配置参数“车牌号码前辍”字符串。 pBuff 指定目的地址，BuffNum 指定目的地的字符空间数。 成功返回 1，失败返回 0。
OCX	VC6	void setRecogCfgProvince(LPCTSTR Province);
	BCB6	void __fastcall setRecogCfgProvince(BSTR Province);
	Delphi7	procedure setRecogCfgProvince(const Province: WideString); safecall;
DLL/SO	C	int DSPAPI dspRecogCfgSetProvince (HDSP hdsp,const wchar_t* Province);
DLL/SO	C++	int RecogCfgSetProvince (const wchar_t* Province);
说明		设置识别配置参数“车牌号码前辍”字符串。可输入本地区省、市代码，如“京 A”、“京”等，也可以不输入任何字符。当输入了地区代码后，系统

可以在字符模糊的情况下替换相应字符，变相提高识别率，这种方式可大大降低误识率，减少人工修改的机率，具有极高的实用价值。默认值为空串。也可以在 RecogParamDlg 函数显示的对话框中设置。

OCX	VC6	void getRecogCfgRange(long* pLeft, long* pTop, long* pRight, long* pBottom);
	Delphi7	procedure getRecogCfgRange(var pLeft: Integer; var pTop: Integer; var pRight: Integer; var pBottom: Integer); safecall;
DLL/SO	C	int DSPAPI dspRecogCfgGetImageRange (HDSP hdsp,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C++	int RecogCfgGetImageRange (long* pLeft,long* pTop,long* pRight,long* pBottom);
说明		读取识别配置参数“图像识别范围”。单位为%。取值 0-100。默认值为左边=0；上边=0；右边=100；下边=100。pLeft, pTop, pRight, pBottom 为 long* 型指针，分别指向左，上，右，下边在图中的坐标的百分比。
示例		//vc6 long Left,Top,Right,Bottom; getRecogCfgRange (&Left,&Top,&Right,&Bottom);

OCX	VC6	long setRecogCfgRange(long Left, long Top, long Right, long Bottom);
	Delphi7	function setRecogCfgRange(Left: Integer; Top: Integer; Right: Integer; Bottom: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogCfgSetImageRange (HDSP hdsp,long Left,long Top,long Right,long Bottom);
DLL/SO	C++	int RecogCfgSetImageRange (long Left,long Top,long Right,long Bottom);
说明		设置识别配置参数“图像识别范围”。单位为%。取值 0-100。用户一般不需要设置该参数。也可以在 RecogParamDlg 函数显示的对话框中设置。

OCX	VC6	long getRecogCfgUseTemplate();
	Delphi7	function getRecogCfgUseTemplate: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogCfgGetUseTemplate (HDSP hdsp);
DLL/SO	C++	int RecogCfgGetUseTemplate (void);
说明		读取识别配置参数“车牌模板”。返回值中从低位 B0 到高位 B31 分别表示：

位	车牌模板	为 0	为 1
B0	民用车牌(92 式)	禁止识别	允许识别
B1	民用货车尾牌(双行)	禁止识别	允许识别
B2	民用车牌(2002 个性化)	禁止识别	允许识别
B3	警车车牌(*警)	禁止识别	允许识别
B4	武警车牌(WJ*)	禁止识别	允许识别
B5	军用车牌(2004 式)	禁止识别	允许识别

默认值为 61（十进制）= 3D（十六进制） = 111101（二进制）。
请参考头文件 platedsp_def.h 中的定义。

OCX	VC6	void setRecogCfgUseTemplate(long Templates);
	Delphi7	procedure setRecogCfgUseTemplate(Templates: Integer); safecall;
DLL/SO	C	int DSPAPI dspRecogCfgSetUseTemplate (HDSP hdsp,int Params);
DLL/SO	C++	int RecogCfgSetUseTemplate(int Params);
说明		设置车牌模版的区域，种类等。

PLATE_TYPE_ID	设置车牌种类
PLATE_TYPE_ID_OR_DEFAULT	设置指定的车牌种类，再加上默认的

	种类
PLATE_TYPE_ID_TAXI	废弃了的功能
PLATE_TYPE_CONTAINER	设置车牌区域（国家/地区） 只有海外版的授权才有效。

例如：

gDSP.RecogCfgSetUseTemplate(PLATE_TYPE_CONTAINER | PLATE_CONTAINER_ID_HK);

gDSP.RecogCfgSetUseTemplate(PLATE_TYPE_ID_OR_DEFAULT);

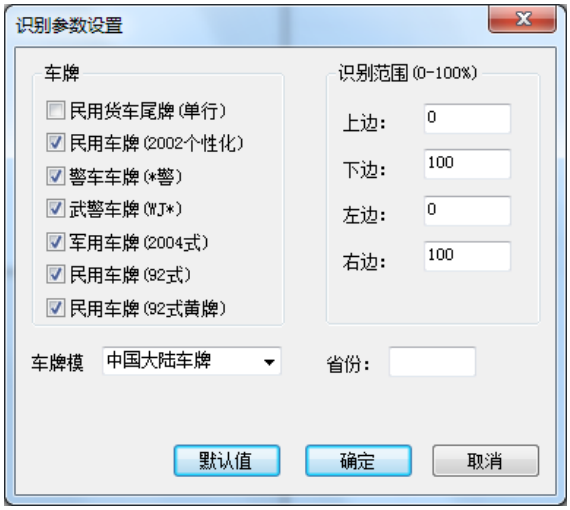
gDSP.RecogCfgSetUseTemplate(PLATE_TYPE_ID_NORMAL5);

请参考头文件 platedsp_def.h 中的定义。

OCX	VC6	long getRecogEnableCount();
	Delphi7	function getRecogEnableCount: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogGetEnableCount (HDSP hdsp);
DLL/SO	C++	int RecogGetEnableCount (void);
说明		读取“识别允许”设置值。返回值为-1 表示连续识别；返回值为正数 N，表示识别 N 帧图像后自动停止，计数值自动递减，直到 0 停止；返回值为 0 时表示停止识别，为 0 时也可对图像抓取，但不再识别。

OCX	VC6	void setRecogEnableCount(long Count);
	Delphi7	procedure setRecogEnableCount(Count: Integer); safecall;
DLL/SO	C	int DSPAPI dspRecogSetEnableCount (HDSP hdsp,int Count);
DLL/SO	C++	int RecogSetEnableCount (int Count);
说明		设置“识别允许”参数。当 Count 为-1 表示连续识别；Count 值为正数 N，表示识别 N 帧图像后自动停止，计数值自动递减，直到 0 停止；Count 值为 0 时表示停止识别，为 0 时也可对图像抓取，但不再识别。

OCX	VC6	long RecogParamDlg();
	Delphi7	function RecogParamDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogParamDlg (HDSP hdsp);
DLL/SO	C++	int RecogParamDlg (void);
说明		显示识别参数设置对话框。如下所示：



OCX	VC6	long RecogStartWithFile(LPCTSTR FileName);

	BCB6	long __fastcall RecogStartWithFile(BSTR FileName);
	Delphi7	function RecogStartWithFile(const FileName: WideString): Integer; safecall;
说明		对 FileName 指定的文件进行识别。FileName 的扩展名可以是.BMP/.JPG 图片格式。由于识别过程在另一个线程中完成，所以，函数返回后，识别过程不一定已经结束，此时读取识别结果是不安全的，应该在 AfterRecogFinished 事件中读取识别结果。函数调用成功返回 1；失败返回 0 或-1。失败的原因可能是图片格式错误 或者 上一次识别还没有结束。是否在控件窗口中显示该图片可提前调用 setImageDisplayEnabled 设置。默认为显示图片。对于 BMP 文件，只支持 RGB15、RGB16、RGB24、RGB32 四种常见格式，支持从上到下 或 从下到上的 BMP 格式。
OCX	VC6	long RecogStartWithFileWait(LPCTSTR FileName);
	BCB6	long __fastcall RecogStartWithFileWait (BSTR FileName);
	Delphi7	function RecogStartWithFileWait (const FileName: WideString): Integer; safecall;
说明		对 FileName 指定的文件进行识别。FileName 的扩展名可以是.BMP/.JPG 图片格式。函数将等待识别完成后才返回，返回后可以安全地读取识别结果，函数调用成功返回 1；失败返回 0 或-1。失败的原因可能是图片格式错误 或者 上一次识别还没有结束。是否在控件窗口中显示该图片可提前调用 setImageDisplayEnabled 设置。默认为显示图片。对于 BMP 文件，只支持 RGB15、RGB16、RGB24、RGB32 四种常见格式，支持从上到下 或 从下到上的 BMP 格式。由于该函数 CPU 使用率不高，不建议使用。
DLL/SO	C	int DSPAPI dspRecogStartWithFile (HDSP hdsp,const wchar_t* FileName,int Params);
DLL/SO	C++	int RecogStartWithFile (const wchar_t* FileName,int Params);
说明		对 FileName 指定的文件进行识别。FileName 的扩展名可以是.BMP/.JPG 图片格式。 如果 Params 指定了 DSP_WAIT_FINISHED=0x04 位为 1，则等待识别结束后再返回。 如果 Params 指定了 DSP_WAIT_FINISHED=0x04 位为 0，则不等待就返回，由于识别过程在另一个线程中完成，所以，函数返回后，识别过程不一定已经结束，此时读取识别结果是不安全的，应该在 AfterRecogFinished 事件中读取识别结果。 函数调用成功返回 1；失败返回 0 或-1。失败的原因可能是图片格式错误 或者 上一次识别还没有结束。是否在控件窗口中显示该图片可提前调用 setImageDisplayEnabled 设置。默认为显示图片。对于 BMP 文件，只支持 RGB15、RGB16、RGB24、RGB32 四种常见格式，支持从上到下 或 从下到上的 BMP 格式。
OCX	VC6	long RecogStartWithMem(long* pImgData);
	Delphi7	function RecogStartWithMem(var pImgData: Integer): Integer; safecall;
说明		对 pImgData 指向的内存中的图像数据进行识别。第 1 次调用该函数前应调用 setRecogWithBitmapHeader4Mem 或 SetRecogImageFormat4Mem 函数，以指定图像的各种参数。由于识别过程在另一个线程中完成，所以，函数返回后，识别过程不一定已经结束，此时读取识别结果是不安全的，应该在 AfterRecogFinished 事件中读取识别结果。是否在控件窗口中显示该图像可提前调用 setImageDisplayEnabled 设置。默认为显示图像。函数调用成功返回 1；失败返回 0 或-1。失败的原因可能是图片格式错误 或者 上一次识别还没有结束。调用该函数时，可以使用视频流识别方法（快速识别）以及文件识别方法（精确识别，稍慢），具体使用方法参见

setRecogWithBitmapHeader4Mem 函数。

OCX	VC6	long RecogStartWithMemEx(long* pData,long Size,long Params);
	Delphi7	function RecogStartWithMemEx(var pData: Integer; Size: Integer; Params: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogStartWithMem (HDSP hdsp,void* pData,unsigned int Size,int Params);
DLL/SO	C++	int RecogStartWithMem (void* pData,unsigned int Size,int Params);
说明		对 pData 指向的内存中的图像数据进行识别。第 1 次调用该函数前应调用 setRecogWithBitmapHeader4Mem 或 SetRecogImageFormat4Mem 函数，以指定图像的各种参数。Size 指定图像数据大小，如果是动态大小的数据（如：JPG 图像数据），就必须指定大小，否则可以传递 0 值 。如果 Params 指定了 DSP_WAIT_FINISHED=0x04 位为 1，则等待识别结束后再返回；否则，立即返回，由于识别过程在另一个线程中完成，所以，函数返回后，识别过程不一定已经结束，此时读取识别结果是不安全的，应该在 AfterRecogFinished 事件中读取识别结果。是否在控件窗口中显示该图像可提前调用 setImageDisplayEnabled 设置。默认为显示图像。函数调用成功返回 1；失败返回 0 或-1。失败的原因可能是图片格式错误 或者 上一次识别还没有结束。调用该函数时，可以使用视频流识别方法（快速识别）以及文件识别方法（精确识别，稍慢），具体使用方法参见 setRecogWithBitmapHeader4Mem 函数。

OCX	VC6	long RecogStartWithMemWait(long* pImgData);
	Delphi7	function RecogStartWithMemWait(var pImgData: Integer): Integer; safecall;
说明		对 pImgData 指向的内存中的图像数据进行识别。该数据为按 BMP 格式排列的位图数据，第 1 次调用该函数前应调用 setRecogWithBitmapHeader4Mem 函数，以指定图像的各种参数。函数将等待识别完成后才返回，返回后可以安全地读取识别结果。是否在控件窗口中显示该图像可提前调用 setImageDisplayEnabled 设置。默认为显示图像。函数调用成功返回 1；失败返回 0 或-1。失败的原因可能是图片格式错误 或者 上一次识别还没有结束。调用该函数时，可以使用视频流识别方法（快速识别）以及文件识别方法（精确识别，稍慢），具体使用方法参见 setRecogWithBitmapHeader4Mem 函数。由于该函数 CPU 使用率不高，不建议使用。

OCX	VC6	long RecogTrainDlg();
	Delphi7	function RecogTrainDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogTrainDlg (HDSP hdsp);
DLL/SO	C++	int RecogTrainDlg (void);
说明		由于 V6 版使用了新的快速人工智能字符识别算法，但其中的训练算法却非常慢（一次训练可能要几十个小时），使用也非常复杂，用户操作起来有困难。从 V6 版开始，我们不再提供字符训练的用户界面。如果您确实需要训练字符，请把包含车牌字符的图片或录像文件传给我们，我们将为您效劳。

OCX	VC6	void setRecogWithBitmapHeader4Mem(long* pBitmapInfoHeader, long bFastRecog);
	Delphi7	procedure setRecogWithBitmapHeader4Mem(var pBitmapInfoHeader: Integer; bFastRecog: Integer); safecall;
说明		设置 RecogStartWithMem 内存图像识别方法的图片格式及识别方法。图片格式由 pBitmapInfoHeader 指定，由于标准的 OCX 支持的指针数据类型有限，虽然这里的 pBitmapInfoHeader 为 long*型指针，但实际应为 BITMAPINFOHEADER* 型指针。bfastRecog 指定的位

		DSP_FAST_RECOG=0x01 为 0 时指定使用文件识别方法（精确识别，稍慢）；为 1 时指定使用视频流识别方法（快速识别）。支持 RGB15、RGB16、RGB24、RGB32 四种常见 BMP 格式，支持从上到下 或 从下到上的 BMP 格式。支持 YUY2,UYVY,I420,YV12,Y42B。支持 BAYER GB/GR/BG/RG。支持 JPG。
OCX	VC6	long SetRecogImageFormat4Mem(long Format,long Width,long Height,long Params);
	Delphi7	function SetRecogImageFormat4Mem (Fomat: Integer;Width: Integer;Height : Integer; Params: Integer) : Integer; safecall;
DLL/SO	C	int DSPAPI dspSetRecogImageFormat4Mem (HDSP hdsp,unsigned int Format,unsigned int Width,int Height,int Params);
DLL/SO	C++	int SetRecogImageFormat4Mem (unsigned int Format,unsigned int Width,int Height,int Params);
说明		设置 RecogStartWithMem 等内存图像识别方法的图片格式及识别方法。图片格式由 Format 指定，可以是 platedsp_def.h 中定义的： DSP_VIDEO_FORMAT_RGB_8, DSP_VIDEO_FORMAT_RGB_X555, DSP_VIDEO_FORMAT_RGB_565,DSP_VIDEO_FORMAT_RGB_888, DSP_VIDEO_FORMAT_RGB_X888,DSP_VIDEO_FORMAT_UYVY, DSP_VIDEO_FORMAT_YUY2,DSP_VIDEO_FORMAT_I420, DSP_VIDEO_FORMAT_YV12,DSP_VIDEO_FORMAT_Y42B, DSP_VIDEO_FORMAT_JPEG,DSP_VIDEO_FORMAT_BAYER_GR, DSP_VIDEO_FORMAT_BAYER_GB,DSP_VIDEO_FORMAT_BAYER_RG, DSP_VIDEO_FORMAT_BAYER_BG 如果 Params 指定的位 DSP_FAST_RECOG=0x01 为 0 时指定使用文件识别方法（精确识别，稍慢）；为 1 时指定使用视频流识别方法（快速识别）。

统计模块

系统初始化时默认关闭了统计功能，要想打开统计功能，请调用 setStatEnabled 函数。V3-V6 版还内置了“统计过滤器”，以方便用户对运动车牌的识别结果的处理。

OCX	VC6	long StatClear();
	Delphi7	function StatClear: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatClear (HDSP hdsp);
DLL/SO	C++	int StatClear (void);
说明		清除统计队列中的所有统计数据。
安全		建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX	VC6	long getStatColorUsed();
	Delphi7	function getStatColorUsed: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatGetColorUsed (HDSP hdsp);
DLL/SO	C++	int StatGetColorUsed (void);
说明		返回统计方法中是否允许对车牌颜色进行对比，1=允许；0=禁止。默认值为 1。

OCX	VC6	void setStatColorUsed(long bUsed);
	Delphi7	procedure setStatColorUsed(bUsed: Integer); safecall;
DLL/SO	C	int DSPAPI dspStatSetColorUsed (HDSP hdsp,int Params);
DLL/SO	C++	int StatSetColorUsed (int Params);
说明		设置统计方法中是否允许对车牌颜色进行对比，bUsed/Params 为 1=允许；为 0=禁止。

OCX	VC6	long getStatEnabled();
	Delphi7	function getStatEnabled: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatGetEnabled (HDSP hdsp);
DLL/SO	C++	int StatGetEnabled (void);
说明		返回是否启动统计功能。1=允许；0=禁止。默认值为 0。

OCX	VC6	void setStatEnabled(long bEnabled);
	Delphi7	procedure setStatEnabled(bEnabled: Integer); safecall;
DLL/SO	C	int DSPAPI dspStatSetEnabled (HDSP hdsp,int Params);
DLL/SO	C++	int StatSetEnabled (int Params);
说明		设置是否启动统计功能。bEnabled/Params 为 1=允许；为 0=禁止。 允许统计后，内部的运动车辆检测相关功能将自动关闭。

OCX	VC6	long getStatMaxTime();
	Delphi7	function getStatMaxTime: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatGetMaxTime (HDSP hdsp);
DLL/SO	C++	int StatGetMaxTime (void);
说明		返回进行统计的最长时间，单位为毫秒。默认值为 1000。

OCX	VC6	void setStatMaxTime(long Time);
	Delphi7	procedure setStatMaxTime(Time: Integer); safecall;
DLL/SO	C	int DSPAPI dspStatSetMaxTime (HDSP hdsp,int Tick);
DLL/SO	C++	int StatSetMaxTime (int Tick);

说明　　设置进行统计的最长时间，单位为毫秒。超过设定时间的旧的记录将被自动清除出队列。默认值为 1000。

OCX　VC6　long StatPasteBestRecord(long bUse);
　　Delphi7　function StatPasteBestRecord(bUse: Integer): Integer; safecall;
说明　　设置读取当前识别结果的方法。

版本	bUse 高 16	bUse 低 16	含义
	指定目标编号（0-3）	0	指定读取当前最新的识别结果。在每次产生 AfterRecogFinished 事件时，系统自动恢复设置为 0。
		1	指定从“统计队列”中读取识别结果。
V3 版新增		2	指定从“统计过滤器”中读取识别结果。
V3.5 版新增	最大目标数（1-4）	3	指定系统可跟踪的最大目标数（1-4 个），默认为 1

受该函数的影响的函数有：getPlateColor，getPlateColorName，getPlateNumber，getPlateReliability，getPlateReliabilityByChar，getPlateTypeName，getPlateTypeId，getPlateTypeNameEx，getPlateTypeIdEx。

安全　　建议只在 AfterRecogFinished 或 AfterFilterStateChanged 事件中调用，否则可能有不可预知的任务冲突。

OCX　VC6　long StatSetMaxTargetNum (long Num);
　　Delphi7　function StatSetMaxTargetNum (Num: Integer): Integer; safecall;
DLL/SO　C　int DSPAPI dspStatSetMaxTargetNum (HDSP hdsp,int Num);
DLL/SO　C++　int StatSetMaxTargetNum (int Num);
说明　　设置最大目标数。
　　相当于调用 StatPasteBestRecord（（Num<<16）|3）。

OCX　VC6　long StatSetCurrentTarget (long Type, long Id);
　　Delphi7　function StatSetCurrentTarget (Type: Integer; Id: Integer): Integer; safecall;
DLL/SO　C　int DSPAPI dspStatSetCurrentTarget (HDSP hdsp,int Type,int Id);
DLL/SO　C++　int StatSetCurrentTarget (int Type,int Id);
说明　　设置读取当前识别结果的方法。

Id	Type	含义
指定目标编号（0-3）	0	指定读取当前最新的识别结果。在每次产生 AfterRecogFinished 事件时，系统自动恢复设置为 0。
	1	指定从“统计队列”中读取识别结果。
	2	指定从“统计过滤器”中读取识别结果。

相当于调用 StatPasteBestRecord（（Id<<16）| Type ）

视频管理模块

OCX	VC6	long getVideoBrightness();
	Delphi7	function getVideoBrightness: Integer; safecall;
说明		返回当前视频设备的亮度设置值。成功时返回值在 0-100 之间。返回为-1 表示失败。默认值为 50。
OCX	VC6	void setVideoBrightness(long Brightness);
	Delphi7	procedure setVideoBrightness(Brightness: Integer); safecall;
说明		设置当前视频设备的亮度值。必须在视频设备已经成功打开的条件下才会成功。成功时返回 1，失败返回 0 或-1。设置范围 1-100。 对于高清相机：当相机支持自动增益时（如大恒高清相机），0 为自动增益。传递负数将自动取正后直接传到相机，而不再按 1-100 归一化。
OCX	VC6	long getVideoCaptureSize(long* pWidth, long* pHeight);
	Delphi7	function getVideoCaptureSize(var pWidth: Integer; var pHeight: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoGetCaptureSize (HDSP hdsp,int* pWidth,int* pHeight);
DLL/SO	C++	int VideoGetCaptureSize (int* pWidth,int* pHeight);
说明		返回当前视频采集设备的采集分辨率。pWidth 为长整型指针，指向接收宽度数据的长整数；pHeight 为长整型指针，指向接收高度数据的长整数。成功时返回 1，失败返回 0 或-1。
OCX	VC6	long setVideoCaptureSize(long Width, long Height);
	Delphi7	function setVideoCaptureSize(Width: Integer; Height: Integer): Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoSetCaptureSize (HDSP hdsp,int Width,int Height);
DLL/SO	C++	int VideoSetCaptureSize (int Width,int Height);
说明		设置当前视频采集设备的采集分辨率。Width 为要设定的宽度值；Height 为要设定的高度值。成功时返回 1，失败返回 0 或-1。必须在调用 setVideoDeviceIndex 函数之后，在调用 setVideoConnected 函数之前设置才有效。
OCX	VC6	long getVideoConnected();
	Delphi7	function getVideoConnected: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoGetConnected (HDSP hdsp);
DLL/SO	C++	int VideoGetConnected (void);
说明		返回当前的视频采集设备是否已经打开。返回 1，表示已经打开；返回 0 表示没有打开。
OCX	VC6	void setVideoConnected(long bConnected);
	Delphi7	procedure setVideoConnected(bConnected: Integer); safecall;
DLL/SO	C	int DSPAPI dspVideoSetConnected (HDSP hdsp,int Params);
DLL/SO	C++	int VideoSetConnected (int Params);
说明		调用该函数首先将关闭已打开的视频设备（视频采集卡 或 AVI 录像文件），然后打开或关闭当前的视频采集设备。bConnected/Params 为 1，表示打开设备；为 0 表示关闭。是否在控件窗口中显示该图像可提前调用 setImageDisplayEnabled 设置。默认为显示图像。
OCX	VC6	long getVideoDeviceIndex();
	Delphi7	function getVideoDeviceIndex: Integer; safecall;

DLL/SO C int DSPAPI dspVideoGetDeviceIndex (HDSP hdsp);
DLL/SO C++ int VideoGetDeviceIndex (void);
说明 返回当前的视频采集设备编号。0 表示第 1 个设备，1 表示第 2 个设备，依此类推。

OCX VC6 void setVideoDeviceIndex(long DeviceIndex);
Delphi7 procedure setVideoDeviceIndex(DeviceIndex: Integer); safecall;
DLL/SO C int DSPAPI dspVideoSetDeviceIndex (HDSP hdsp,int Index);;
DLL/SO C++ int VideoSetDeviceIndex (int Index);
说明 设置当前的视频采集设备编号。0 表示第 1 个设备，1 表示第 2 个设备，依此类推。调用该函数后，系统将自动调用已设置的该编号的视频设备的参数值。必须在调用 setVideoConnected 函数之前设置编号才有效。
如果加上 DSP_JUST_HI_VIDEO_INDEX=1000，将只打开高清相机。
DeviceIndex 允许的最大值受您购买的产品类型（4 路/8 路/12 路）限制。

OCX VC6 CString getVideoDeviceName();
BCB6 BSTR __fastcall getVideoDeviceName(void);
Delphi7 function getVideoDeviceName: WideString; safecall;
说明 返回当前已经打开的视频采集设备的名称字符串。
由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。

DLL/SO C int DSPAPI dspVideoGetDeviceName (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO C++ int VideoGetDeviceName (wchar_t* pBuff,int BuffNum);
说明 返回当前已经打开的视频采集设备的名称字符串。
pBuff 指定目的地址，BuffNum 指定目的字符空间大小。
成功返回 1，失败返回 0.

OCX VC6 long VideoDisplayDlg();
Delphi7 function VideoDisplayDlg: Integer; safecall;
DLL/SO C int DSPAPI dspVideoDisplayDlg (HDSP hdsp);
DLL/SO C++ int VideoDisplayDlg (void);
说明 显示当前已经打开的视频采集设备的显示设置对话框。

OCX VC6 long getVideoDisplayFormat();
Delphi7 function getVideoDisplayFormat: Integer; safecall;
DLL/SO C int DSPAPI dspVideoGetDisplayFormat (HDSP hdsp);
DLL/SO C++ int VideoGetDisplayFormat (void);
说明 返回当前视频的标准制式。

OCX VC6 void setVideoDisplayFormat(long Format);
Delphi7 procedure setVideoDisplayFormat(Format: Integer); safecall;
DLL/SO C int DSPAPI dspVideoSetDisplayFormat (HDSP hdsp,int Params);
DLL/SO C++ int VideoSetDisplayFormat (int Params);
说明 设置当前视频的标准制式。必须在调用 setVideoDeviceIndex 函数之后，在调用 setVideoConnected 函数之前设置才有效。以下为视频制式列表：

制式	十六进制值	制式	十六进制值
NTSC_M	0x00000001	PAL_B	0x00000010
NTSC_M_J	0x00000002	PAL_D	0x00000020
NTSC_433	0x00000004	PAL_H	0x00000080

		PAL_I	0x00000100
		PAL_M	0x00000200
		PAL_N	0x00000400
		PAL_60	0x00000800

OCX	VC6	long VideoFormatDlg();
	Delphi7	function VideoFormatDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoFormatDlg (HDSP hdsp);
DLL/SO	C++	int VideoFormatDlg (void);
说明		显示当前已经打开的视频采集设备的格式设置对话框。

OCX	VC6	long getVideoSource();
	Delphi7	function getVideoSource: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoGetSource (HDSP hdsp);
DLL/SO	C++	int VideoGetSource (void);
说明		返回当前视频设备的输入端子编号。成功返回 0-N。失败返回-1

OCX	VC6	void setVideoSource(long Source);
	Delphi7	procedure setVideoSource(Source: Integer); safecall;
DLL/SO	C	int DSPAPI dspVideoSetSource (HDSP hdsp,int Source);
DLL/SO	C++	int VideoSetSource (int Source);
说明		设置当前视频设备的输入端子编号。

OCX	VC6	long VideoSourceDlg();
	Delphi7	function VideoSourceDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoSourceDlg (HDSP hdsp);
DLL/SO	C++	int VideoSourceDlg (void);
说明		显示当前已经打开的视频采集设备的通道设置对话框。

OCX	VC6	long setVideoOtherParams (long Type, long Param);
	Delphi7	function setVideoOtherParams(Type: Integer; Param: Integer); safecall;
DLL/SO	C	int DSPAPI dspVideoSetOtherParams (HDSP hdsp,int Type,int Params);
DLL/SO	C++	int VideoSetOtherParams (int Type,int Params);
说明		设置视频设备的其它参数。成功返回 1。失败返回 0 或-1。

Type	Param
DSP_VIDEO_CRYOSC=0 晶体频率	28/35 （MHz）
DSP_VIDEO_BRIGHTNESS=1 亮度	0-100 (%)
DSP_VIDEO_CONTRAST=2 对比度	0-100 (%)
DSP_VIDEO_HUE=3 色调	0-100 (%)
DSP_VIDEO_SATURATION=4 饱和度	0-100 (%)
DSP_VIDEO_SHUTTER=5 设置快门速度	(us 微秒) 当相机支持自动快门时（如大恒高清相机），0 为自动快门
DSP_FLASH_SOURCE=6 强制闪光	1=闪光
DSP_FLASH_WIDTH=7 闪光脉冲宽度	(us 微秒)

DSP_FLASH_DELAY=8 闪光到采集图像延时	(us 微秒)
DSP_VIDEO_RUN=9 暂停/运行视频设备	0=暂停；1=运行
DSP_VIDEO_ADC_BIT=10 ADC 位数控制	参考具体相机
DSP_VIDEO_DLL_HANDLE=11 相应 DLL 的 HINSTANCE	用户可据此调用一些特定的函数，参考具体相机 SDK
DSP_VIDEO_DEVICE_HANDLE=13 相应视频设备的 HANDLE	
DSP_VIDEO_ANGLE90=12 旋转 90 度	0=不旋转； 90=顺时针； -90=逆时针；
DSP_VIDEO_WHITE_BALANCE_ON_OFF=14 相机的自动白平衡功能	0=禁止； 1=允许
DSP_VIDEO_JPG_QUALITY=15 直接传输 JPG 图像	0=不允许； 1-100=JPG 压缩质量；
DSP_VIDEO_ANTI_FLICKER=16 抗闪烁	0=关闭； 1=允许
DSP_VIDEO_AUTO_GAIN_MIN=17 软件自动光圈最小值	0-100（%）
DSP_VIDEO_AUTO_GAIN_MAX=18 软件自动光圈最大值	0-100（%）
DSP_VIDEO_AUTO_SHUTTER_MIN=19 软件自动快门最小值	单位 uS
DSP_VIDEO_AUTO_SHUTTER_MAX=20 软件自动快门最大值	单位 uS
DSP_VIDEO_AUTO_SHUTTER=21 软件自动快门	0=关闭； 1=允许
DSP_VIDEO_AUTO_BRIGHTNESS=22 软件自动光圈	0=关闭； 1=允许
DSP_VIDEO_WHITE_BALANCE_R=23 软件自动白平衡 R 值比例	100-
DSP_VIDEO_WHITE_BALANCE_G=24 软件自动白平衡 G 值比例	100-
DSP_VIDEO_WHITE_BALANCE_B=25 软件自动白平衡 B 值比例	100-
DSP_VIDEO_CAMERA_WHITE_BALANCE_ON_OFF=26 相机自动白平衡	0=关闭； 1=允许
DSP_VIDEO_BAYER_TO_YUV=27 当显卡支持 YV12 或 I420 时转换为 YUV 否则转换为 RGB	0=关闭（默认值） 1=允许
DSP_VIDEO_TRIG_CAPTURE=28 软触发抓拍图像	闪光灯定义
DSP_VIDEO_SET_IP_ADDRESS=29 设置网络相机 IP 地址	如 192.1.1.1 = 0xC0010101
DSP_VIDEO_SET_NET_PORT0=30 设置网络相机网络端口号	如 8886,如果设置为 0，则使用系统默认值
DSP_VIDEO_SET_NET_PORT1=31 设置网络相机网络端口号	如 8888,如果设置为 0，则

	使用系统默认值
DSP_VIDEO_SET_DEV_CHANNEL=32 设置视频设备的通道号	如 1，如果设置为 0，则使用系统默认值
DSP_VIDEO_AUTO_LOAD_CONFIG=36 设置打开视频设备时是否使用已保存的参数自动设置	1=自动设置，0=不设置。系统默认值为 1

2012/8/15 新增功能

OCX VC6 long setVideoOtherParamsStr (long Type, LPCTSTR Param);

Delphi7 function setVideoOtherParamsStr(Type: Integer; const Param: WideStringer); safecall;

DLL/SO C int DSPAPI dspVideoSetOtherParamsStr (HDSP hdsp,int Type,const wchar_t* Params);

DLL/SO C++ int VideoSetOtherParamsStr (int Type,const wchar_t* Params);

说明 设置视频设备的其它参数。成功返回 1。失败返回 0 或-1。

Type	Param
DSP_VIDEO_SET_USERNAME_STR=33 设置访问相机的用户名	如 L"admin", 如果设置为 NULL，则使用系统默认值
DSP_VIDEO_SET_PASSWORD_STR=34 设置访问相机的用户密码	如 L"12345", 如果设置为 NULL，则使用系统默认值
DSP_VIDEO_SET_IP_ADDRESS_STR=35 设置访问相机的 IP 地址	如 L"192.168.1.10"
DSP_VIDEO_SET_STREAM_NAME_STR=37 设置访问视频流的名称	如: L"c:\\abc.avi" L"RTSP://…"

DLL/SO C int DSPAPI dspVideoGetOtherParams (HDSP hdsp,int Type,long * Ret);

DLL/SO C++ int VideoGetOtherParams (int Type,long * Ret);

说明 读取视频设备的其它参数。成功返回 1。失败返回 0 或-1。

参见 VideoSetOtherParams。

DLL/SO C int DSPAPI dspVideoGetDeviceHandle (HDSP hdsp,const char* DllName,HVID* DeviceHandle);

DLL/SO C++ int VideoGetDeviceHandle (const char* DllName,HVID* DeviceHandle);

说明 读取视频设备的句柄。

DllName 指定 Dll 名，或直接传 NULL 值 。

DeviceHandle 为指向接收句柄的指针。

成功返回 1。失败返回 0 或-1。

DLL/SO C void* DSPAPI dspVideoGetDllFunction (HDSP hdsp,const char* DllName,const char* FunctionName);

DLL/SO C++ void* VideoGetDllFunction (const char* DllName,const char* FunctionName);

说明 读取视频开发包 DLL、SO 中函数的指针。

DllName 指定 Dll 名，或直接传 NULL 值 。

FunctionName 为函数名称。

成功返回指针。失败返回 NULL。

OCX VC6 long VideoReadIO(long Type,long* pBuf,long BufSize);

Delphi7 function VideoReadIO(Type: Integer;var pBuf: Integer;BufSize: Integer): Integer; safecall;

DLL/SO C int dspVideoReadIO(int Type,void* pBuf,int BufSize);

DLL/SO C++ int VideoReadIO (int Type,void* pBuf,int BufSize);

说明		读取视频设备支持的 IO 状态或数据。 2011/5/16 新增 DSP_VIDEO_READ_CURRENT_FRAME 表示读取当前闪光帧的数据 DSP_VIDEO_IO_RS232_0 表示读取RS232接口的数据 DSP_VIDEO_IO_RS232_1 DSP_VIDEO_IO_RS232_2 DSP_VIDEO_IO_RS232_3 DSP_VIDEO_IO_STATUS 表示读取 I/O 接口的状态
示例 1		BYTE Buf[16]; int ReadSize = VideoReadIO(DSP_VIDEO_IO_RS232_0,Buf,sizeof(Buf));
示例 2		BYTE Buf[16]; int ReadSize = VideoReadIO(DSP_VIDEO_READ_CURRENT_FRAME | DSP_VIDEO_IO_RS232_0, Buf,sizeof(Buf));
OCX	VC6	long VideoWriteIO(long Type,long* pData,long Size);
	Delphi7	function VideoWriteIO (Type: Integer;var pData: Integer; Size: Integer): Integer; safecall;
DLL/SO	C	int dspVideoWriteIO (int Type,const void* pData,int Size);
DLL/SO	C++	int VideoWriteIO (int Type,const void* pData,int Size);
说明		设置或写入视频设备支持的 IO 状态或数据。 2011/5/16 新增

示例		int Status;
		if(VideoWriteO(DSP_VIDEO_IO_STATUS,&Status,sizeof(Status))
		{
		}

应用程序再加密

OCX	VC6	long LicenseDataRead(long FileID, long* pDes, long BufSize);
	Delphi7	function LicenseDataRead(FileID: Integer; var pDes: Integer; BufSize: Integer): Integer; safecall;
说明		从正式版软件的加密狗中读取用户设置的授权信息。成功返回读取的字节数。失败返回 0 或-1。FileID 指定文件的 ID 编号，编号为 0-4（0 由识别软件开发商保留使用，用户最好不要使用该编号）。pDes 指向目标数据存取缓冲区。BufSize 指定缓冲区的大小，单位为字节。调用前应调用 LicensePasswordInput 设置并通过用户密码。详细使用请参考示例中的 License 目录中的源代码。
DLL/SO	C	int DSPAPI dspLicenseDataRead (HDSP hdsp,int FileID,const void* Password,int PasswordLen,void* pBuf,int BufSize);
DLL/SO	C++	int LicenseDataRead (int FileID,const void* Password,int PasswordLen,void* pBuf,int BufSize);
说明		从正式版软件的加密狗中读取用户设置的授权信息。成功返回读取的字节数。失败返回 0 或-1。FileID 指定文件的 ID 编号，编号为 0-4（0 由识别软件开发商保留使用，用户最好不要使用该编号）。 Password 指向用户密码,PasswordLen 为密码字节长度，总字符数不应超过 64 字节。 pDes 指向目标数据存取缓冲区。BufSize 指定缓冲区的大小，单位为字节。调用前应调用 LicensePasswordInput 设置并通过用户密码。 成功返回读取的字节数。失败返回 0 或-1。
OCX	VC6	long LicenseDataWrite(long FileID, long* pSrc, long Bytes);
	Delphi7	function LicenseDataWrite(FileID: Integer; var pSrc: Integer; Bytes: Integer): Integer; safecall;
说明		写用户的授权信息数据到正式版软件的加密狗中。成功返回已写入的字节数。失败返回 0 或-1。FileID 指定文件的 ID 编号，编号为 0-4（0 由识别软件开发商保留使用，用户最好不要使用该编号）。pSrc 指向用户数据存取缓冲区。Bytes 指定要写入的字节数，该值不应大于 64，否则最多写入 64 字节。调用前应调用 LicensePasswordInput 设置并通过用户密码。详细使用请参考示例中的 License 目录中的源代码。
DLL/SO	C	int DSPAPI dspLicenseDataWrite (HDSP hdsp,int FileID,const void* Password,int PasswordLen,const void* pSrc,int Bytes);
DLL/SO	C++	int LicenseDataWrite (int FileID,const void* Password,int PasswordLen,const void* pSrc,int Bytes);
说明		写用户的授权信息数据到正式版软件的加密狗中。成功返回已写入的字节数。失败返回 0 或-1。FileID 指定文件的 ID 编号，编号为 0-4（0 由识别软件开发商保留使用，用户最好不要使用该编号）。 Password 指向用户密码,PasswordLen 为密码字节长度，总字符数不应超过 64 字节。 pSrc 指向用户数据存取缓冲区。Bytes 指定要写入的字节数，该值不应大于 64，否则最多写入 64 字节。调用前应调用 LicensePasswordInput 设置并通过用户密码。 成功返回已写入的字节数。失败返回 0 或-1。

OCX	VC6	long LicensePasswordInput(LPCTSTR Password);
	BCB6	long __fastcall LicensePasswordInput(BSTR Password);
	Delphi7	function LicensePasswordInput(const Password: WideString): Integer; safecall;
说明		输入用户密码。Password 指明密码，总字符数不应超过 64 字节。返回值为一个与加密狗唯一对应的数值（该值与密码及加密狗唯一对应，应用该值与用户数据运算可大大提高加密强度）。用户密码默认为：12345678
DLL/SO	C	int DSPAPI dspLicenseGetUserSerialID (HDSP hdsp,const void* Password,int PasswordLen);
DLL/SO	C++	int LicenseGetUserSerialID (const void* Password,int PasswordLen);
说明		Password 指向用户密码,PasswordLen 为密码字节长度。 返回值为一个与加密狗唯一对应的数值（该值与密码及加密狗唯一对应，应用该值与用户数据运算可大大提高加密强度）。
OCX	VC6	long LicensePasswordChange(long FileID, LPCTSTR NewPassword);
	BCB6	long __fastcall LicensePasswordChange(long FileID, BSTR NewPassword);
	Delphi7	function LicensePasswordChange(FileID: Integer; const NewPassword: WideString): Integer; safecall;
说明		更改用户密码。FileID 指定文件的 ID 编号，编号为 0-4（0 由识别软件开发商保留使用，用户最好不要使用该编号）。NewPassword 指定新的用户密码。成功时返回非 0 值，失败返回 0。调用前应调用 LicensePasswordInput 设置并通过用户密码。详细使用请参考示例中的 License 目录中的源代码。
DLL/SO	C	int DSPAPI dspLicensePasswordChange (HDSP hdsp,int FileID,const void* OldPassword,int OldPasswordLen,const void* NewPassword,int NewPasswordLen);
DLL/SO	C++	int LicensePasswordChange (int FileID,const void* OldPassword,int OldPasswordLen,const void* NewPassword,int NewPasswordLen);
说明		更改用户密码。FileID 指定文件的 ID 编号，编号为 0-4（0 由识别软件开发商保留使用，用户最好不要使用该编号）。 OldPassword 指向用户旧密码, OldPasswordLen 为旧密码字节长度。 NewPassword 指定新的用户密码。成功时返回非 0 值，失败返回 0。调用前应调用 LicensePasswordInput 设置并通过用户密码。详细使用请参考示例中的 License 目录中的源代码。 成功返回值大于 0，失败返回 0.

红绿灯信号检测

2011/12/9 新增功能

OCX	VC6	int RedLampDetectSetEnabled(int Params);
DLL/SO	C	int DSPAPI dspRedLampDetectSetEnabled(HDSP hdsp,int Params);
DLL/SO	C++	int RedLampDetectSetEnabled(int Params);
说明		设置禁止或允许红绿灯检测。Params=0 为禁止；Params=1 为允许。 返回值为本次设置已前的旧状态值。
OCX	VC6	int RedLampDetectGetNum();
DLL/SO	C	int DSPAPI dspRedLampDetectGetNum(HDSP hdsp);
DLL/SO	C++	int RedLampDetectGetNum();
说明		读取已设置的灯数。返回值从 0 到 DSP_MAX_RED_LAMP_NUM=8
OCX	VC6	int RedLampDetectSetNum(int Num);
DLL/SO	C	int DSPAPI dspRedLampDetectSetNum(HDSP hdsp,int Num);
DLL/SO	C++	int RedLampDetectSetNum(int Num);
说明		设置检测的灯数。Num 取值从 0 到 DSP_MAX_RED_LAMP_NUM=8 成功返回 1，失败返回 0
OCX	VC6	int RedLampDetectSetDisplayResultEnabled(int Params);
DLL/SO	C	int DSPAPI dspRedLampDetectSetDisplayResultEnabled(HDSP hdsp,int Params);
DLL/SO	C++	int RedLampDetectSetDisplayResultEnabled(int Params);
说明		设置禁止或允许显示检测的结果。Params=0 为禁止；Params=1 为允许。
OCX	VC6	int RedLampDetectSetDisplayRangeEnabled(long Params);
DLL/SO	C	int DSPAPI dspRedLampDetectSetDisplayRangeEnabled(HDSP hdsp,long Params);
DLL/SO	C++	int RedLampDetectSetDisplayRangeEnabled(long Params);
说明		设置禁止或允许显示检测范围。Params=0 为禁止；Params=1 为允许。
OCX	VC6	int RedLampDetectSetMinDots(int DotNum);
DLL/SO	C	int DSPAPI dspRedLampDetectSetMinDots(HDSP hdsp,int DotNum);
DLL/SO	C++	int RedLampDetectSetMinDots(int DotNum);
说明		设置检测的灯的 X/Y 方向最小的像素点（设置灵敏度）。默认值为 4。
OCX	VC6	int RedLampDetectGetMinDots();
DLL/SO	C	int DSPAPI dspRedLampDetectGetMinDots(HDSP hdsp);
DLL/SO	C++	int RedLampDetectGetMinDots();
说明		读取检测最小的像素点（设置灵敏度）。默认值为 4。
OCX	VC6	int RedLampDetectSetRecogRange(int LampIndex,long Left,long Top,long Right,long Bottom);
DLL/SO	C	int DSPAPI dspRedLampDetectSetRecogRange(HDSP hdsp,int LampIndex,long Left,long Top,long Right,long Bottom);
DLL/SO	C++	int RedLampDetectSetRecogRange(int LampIndex,long Left,long Top,long Right,long Bottom);
说明		设置检测范围。LampIndex 表示灯的编号（从 0 到 7）。 Left 表示左边像素 X 坐标值；Top 表示上边像素 Y 坐标值；Right 表示右边

像素 X 坐标值；Bottom 表示下边像素 Y 坐标值。
对于某些红黄绿一体灯（表示一个灯的不同状态的），可以化定为一个灯。

OCX	VC6	int RedLampDetectGetRecogRange(int LampIndex,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C	int DSPAPI dspRedLampDetectGetRecogRange(HDSP hdsp,int LampIndex,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C++	int RedLampDetectGetRecogRange(int LampIndex,long* pLeft,long* pTop,long* pRight,long* pBottom);
说明		读取已设置的灯的检测范围。LampIndex 表示灯的编号（从 0 到 7）。 在 pLeft,pTop,pRight,pBottom 指向的 long 类型变量中将返回读取的值。 成功返回 1，失败返回 0

OCX	VC6	int RedLampDetectGetLocate(int LampIndex,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C	int DSPAPI dspRedLampDetectGetLocate(HDSP hdsp,int LampIndex,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C++	int RedLampDetectGetLocate(int LampIndex,long* pLeft,long* pTop,long* pRight,long* pBottom);
说明		读取检测到的灯的坐标。LampIndex 表示灯的编号（从 0 到 7）。 在 pLeft,pTop,pRight,pBottom 指向的 long 类型变量中将返回读取的值。 成功返回 1，失败返回 0
安全		只可以在 AfterRedLampStateChanged 事件中调用。

OCX	VC6	int RedLampDetectCfgLoad();
DLL/SO	C	int DSPAPI dspRedLampDetectCfgLoad(HDSP hdsp);
DLL/SO	C++	int RedLampDetectCfgLoad();
说明		从注册表中读取已保存的设置信息。

OCX	VC6	int DSPAPI dspRedLampDetectCfgSave();
DLL/SO	C	int DSPAPI dspRedLampDetectCfgSave(HDSP hdsp);
DLL/SO	C++	int DSPAPI dspRedLampDetectCfgSave();
说明		保存设置信息到注册表中。

车辆轮廓运动轨迹视频跟踪

2011/12/9 新增功能

“车辆轮廓运动轨迹视频跟踪”仅适用于车流密度不是很高的场所，如果车流太密，可能无法有效检测。对于大型车辆较多的地方，检测效果也不一定理想。

OCX VC6 int MotionSetRunMode(int RunMode);
DLL/SO C int DSPAPI dspMotionSetRunMode(HDSP hdsp,int RunMode);
DLL/SO C++ int MotionSetRunMode(int RunMode);
说明 设置检测的工作模式。

RunMode 值：	解释：
DSP_MOTION_RUN_MODE_NONE = 0	禁止检测
DSP_MOTION_RUN_MODE_CROSSROADS = 1	十字路口模式
DSP_MOTION_RUN_MODE_LANE = 2	卡口模式
DSP_MOTION_RUN_MODE_PARKING = 3	停车场模式

返回值为设置之前的工作模式。

OCX VC6 int MotionSetDisplayResultEnabled(int Params);
DLL/SO C int DSPAPI dspMotionSetDisplayResultEnabled(HDSP hdsp,int Params);
DLL/SO C++ int MotionSetDisplayResultEnabled(int Params);
说明 设置禁止或允许显示检测的结果。Params=0 为禁止；Params=1 为允许。

OCX VC6 int MotionSetDisplayLaneEnabled(int Params);
DLL/SO C int DSPAPI dspMotionSetDisplayLaneEnabled(HDSP hdsp,int Params);
DLL/SO C++ int MotionSetDisplayLaneEnabled(int Params);
说明 设置禁止或允许显示通道定义的范围。Params=0 为禁止；Params=1 为允许。

OCX VC6 int MotionGetLaneNum();
DLL/SO C int DSPAPI dspMotionGetLaneNum(HDSP hdsp);
DLL/SO C++ int MotionGetLaneNum();
说明 读取已设置的通道数。

OCX VC6 int MotionSetLaneNum(int Num);
DLL/SO C int DSPAPI dspMotionSetLaneNum(HDSP hdsp,int Num);
DLL/SO C++ int MotionSetLaneNum(int Num);
说明 设置检测的通道数。Num 取值从 0 到 DSP_MAX_MOTION_LANE_NUM=4

OCX VC6 int MotionSetLaneRangeY(int Top,int Bottom);
DLL/SO C int DSPAPI dspMotionSetLaneRangeY(HDSP hdsp,int Top,int Bottom);
DLL/SO C++ int MotionSetLaneRangeY(int Top,int Bottom);
说明 设置检测范围的上边及下边。

OCX VC6 int MotionGetLaneRangeY(long* pTop,long * pBottom);
DLL/SO C int DSPAPI dspMotionGetLaneRangeY(HDSP hdsp,long* pTop,long * pBottom);
DLL/SO C++ int MotionGetLaneRangeY(long* pTop,long * pBottom);
说明 读取检测范围的上边及下边。
在 pTop,pBottom 指向的 long 类型变量中将返回读取的值。
成功返回 1，失败返回 0

OCX	VC6	int MotionSetLaneRangeX(int LaneIndex,int TopLeft,int TopRight,int BottomLeft,int BottomRight);
DLL/SO	C	int DSPAPI dspMotionSetLaneRangeX(HDSP hdsp,int LaneIndex,int TopLeft,int TopRight,int BottomLeft,int BottomRight);
DLL/SO	C++	int MotionSetLaneRangeX(int LaneIndex,int TopLeft,int TopRight,int BottomLeft,int BottomRight);
说明		设置由 LaneIndex（取值 0 到 3）指定的通道的 X 坐标值。 TopLeft 指定左上角的 X 坐标；TopRight 指定右上角的 X 坐标；BottomLeft 指定左下角的 X 坐标；BottomRight 指定右下角的 X 坐标。X 坐标值可以为负数。

OCX	VC6	int MotionGetLaneRangeX(int LaneIndex,long* pTopLeft,long* pTopRight,long* pBottomLeft,long* pBottomRight);
DLL/SO	C	int DSPAPI dspMotionGetLaneRangeX(HDSP hdsp,int LaneIndex,long* pTopLeft,long* pTopRight,long* pBottomLeft,long* pBottomRight);
DLL/SO	C++	int MotionGetLaneRangeX(int LaneIndex,long* pTopLeft,long* pTopRight,long* pBottomLeft,long* pBottomRight);
说明		读取由 LaneIndex（取值 0 到 3）指定的通道的 X 坐标值。 在 pTopLeft, pTopRight, pBottomLeft, pBottomRight 指向的 long 类型变量中将返回读取的值。 成功返回 1，失败返回 0

OCX	VC6	int MotionGetCurrentLocatedRect(int MotionTarget,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C	int DSPAPI dspMotionGetCurrentLocatedRect(HDSP hdsp,int MotionTarget,long* pLeft,long* pTop,long* pRight,long* pBottom);
DLL/SO	C++	int MotionGetCurrentLocatedRect(int MotionTarget,long* pLeft,long* pTop,long* pRight,long* pBottom);
说明		读取检测到的目标定位结果。MotionTarget 指定目标编号（取值 0-7）， 在 pLeft, pTop, pRight, pBottom 指向的 long 类型变量中将返回读取的值。 成功返回 1，失败返回 0
安全		只可以在 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionGetIdentity(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetIdentity(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetIdentity(int MotionTarget);
说明		读取目标（由 MotionTarget 指定）的识别编号（不等于 0,自动顺序递增）。 成功返回非 0 值，失败返回 0
安全		只可以在 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionGetIdentityByPlateTarget(int Target);
DLL/SO	C	int DSPAPI dspMotionGetIdentityByPlateTarget(HDSP hdsp,int Target);
DLL/SO	C++	int MotionGetIdentityByPlateTarget(int Target);
说明		读取与车牌识别目标（由 Target 指定）绑定的运动目标的识别编号。 成功返回非 0 值，失败返回 0
安全		只可以在 AfterFilterStateChanged 或 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionGetCurrentLaneIndex(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetCurrentLaneIndex(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetCurrentLaneIndex(int MotionTarget);
说明		读取目标（由 MotionTarget 指定）当前的车道编号。

安全		只可以在 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionGetResultType(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetResultType(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetResultType(int MotionTarget);
说明		读取目标（由 MotionTarget 指定）当前的运动类型。 返回值为： DSP_MOTION_UNKNOW = 0 DSP_MOTION_UP_TO_DOWN = 0x1 DSP_MOTION_DOWN_TO_UP = 0x2
安全		只可以在 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionGetSpeedX(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetSpeedX(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetSpeedX(int MotionTarget);
说明		读取目标（由 MotionTarget 指定）当前的 X 方向的速度。单位为：像素/分钟。
安全		只可以在 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionGetSpeedY(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetSpeedY(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetSpeedY(int MotionTarget);
说明		读取目标（由 MotionTarget 指定）当前的 Y 方向的速度。单位为：像素/分钟。
安全		只可以在 AfterMotionStateChanged 事件中调用。

OCX	VC6	int MotionCfgLoad();
DLL/SO	C	int DSPAPI dspMotionCfgLoad(HDSP hdsp);
DLL/SO	C++	int MotionCfgLoad();
说明		从注册表中读取已保存的设置信息。

OCX	VC6	int MotionCfgSave();
DLL/SO	C	int DSPAPI dspMotionCfgSave(HDSP hdsp);
DLL/SO	C++	int MotionCfgSave();
说明		保存设置信息到注册表中。

杂项

OCX	VC6	long AboutDlg();
	Delphi7	function AboutDlg: Integer; safecall;
说明		显示如下的 About 对话框。

OCX	VC6	long getMsgInfoEnabled();
	Delphi7	function getMsgInfoEnabled: Integer; safecall;
DLL/SO	C	int DSPAPI dspMsgInfoGetEnabled (HDSP hdsp);
DLL/SO	C++	int MsgInfoGetEnabled (void);
说明		返回是否显示识别系统的状态栏。1 表示显示，0 表示不显示。默认为 1。

OCX	VC6	void setMsgInfoEnabled(long bEnabled);
	Delphi7	procedure setMsgInfoEnabled(bEnabled: Integer); safecall;
DLL/SO	C	int DSPAPI dspMsgInfoSetEnabled (HDSP hdsp,int Params);
DLL/SO	C++	int MsgInfoSetEnabled (int Params);
说明		设置是否显示识别系统的状态栏。bEnabled 为 1 表示显示，为 0 表示不显示。调用该函数可以将视频窗口的位置及大小信息立即刷新。应用该特性，可用来更新放大或缩小的窗口信息。

OCX	VC6	long MsgInfoDisplay(LPCTSTR Msg);
	BCB6	long __fastcall MsgInfoDisplay(BSTR Msg);
	Delphi7	function MsgInfoDisplay(const Msg: WideString) : Integer; safecall;
DLL/SO	C	int DSPAPI dspMsgInfoDisplay (HDSP hdsp,const wchar_t* Msg);
DLL/SO	C++	int MsgInfoDisplay (const wchar_t* Msg);
说明		在状态栏上显示用户指定的 Msg 信息。返回 1 表示成功，0 表示失败。

OCX	VC6	CString getMsgLogoImageFile();
	BCB6	BSTR __fastcall getMsgLogoImageFile(void);
	Delphi7	function getMsgLogoImageFile: WideString; safecall;
说明		返回当前用户图标文件名。默认为空串。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。

DLL/SO	C	int DSPAPI dspMsgLogoImageGetFile (HDSP hdsp,wchar_t* pBuff,int BuffNum);

DLL/SO	C++	int MsgLogoImageGetFile (wchar_t* pBuff,int BuffNum);
说明		读取当前用户图标文件名。 pBuff 指向目的地址，BuffNum 指定目的地的字符空间大小。 返回复制的空间大小。
OCX	VC6	void setMsgLogoImageFile(LPCTSTR ImageFile);
	BCB6	void __fastcall setMsgLogoImageFile(BSTR ImageFile);
	Delphi7	procedure setMsgLogoImageFile(const ImageFile: WideString); safecall;
DLL/SO	C	int DSPAPI dspMsgLogoImageSetFile (HDSP hdsp,const wchar_t* FileName);
DLL/SO	C++	int MsgLogoImageSetFile (const wchar_t* FileName);
说明		设置当前用户图标文件名 ImageFile 指定的图片文件（文件的扩展名可以是.BMP/.JPG 图片格式）。
OCX	VC6	long MsgLogoImageRefresh();
	Delphi7	function MsgLogoImageRefresh: Integer; safecall;
DLL/SO	C	int DSPAPI dspMsgLogoImageRefresh (HDSP hdsp);
DLL/SO	C++	int MsgLogoImageRefresh (void);
说明		显示当前的用户图标文件。可以使用该组函数显示用户指定的图片。
OCX	VC6	CString getPathOfCharLib();
	BCB6	BSTR __fastcall getPathOfCharLib(void);
	Delphi7	function getPathOfCharLib: WideString; safecall;
说明		返回识别控件字库文件 CharLib.ini 所在的目录，也就是识别控件 PlateDSP.V6.OCX 的目录。 由于某些 C++编译器（如 BCB6：Borland C++ Builder）对 OCX / COM 组件返回的 BSTR 字符串无法正确释放，从而可能产生内存泄漏，解决方法请参考第52页的“BSTR 字符串与内存泄漏”。
OCX	VC6	void setRedBoxDisplayEnabled (long bEnabled);
	Delphi7	procedure setRedBoxDisplayEnabled (bEnabled: Integer); safecall;
DLL/SO	C	int DSPAPI dspSetRedBoxDisplayEnabled (HDSP hdsp,int Params);
DLL/SO	C++	int SetRedBoxDisplayEnabled (int Params);
说明		设置是否显示红色的车牌家位框。bEnabled/Params 为 1 表示显示，为 0 表示不显示。
OCX	VC6	long DoEvent();
	Delphi7	function DoEvent: Integer; safecall;
DLL/SO	C	int DSPAPI dspDoEvent (HDSP hdsp);
DLL/SO	C++	int DoEvent (void);
说明		系统消息循环。用户如果要在主线程中循环等待时，可以调节器用该函数，以保证消息的循环，以免引起系统阻塞。
DLL/SO	C	int DSPAPI dspMsgEnabledFPS (HDSP hdsp,int bEnabled);
DLL/SO	C++	int MsgEnabledFPS (int bEnabled);
说明		允许或禁止显示帧率信息。bEnabled 为 1 表示显示，为 0 表示不显示。

控件事件

OCX VC6 void AfterDvrClosed();
Delphi7 procedure AfterDvrClosed;
DLL/SO C++ virtual void AfterDvrClosed(void);
说明 当录制视频流的文件完成关闭后，将产生该事件，以通知用户进行相关的处理。该事件是任务安全的，可进行 GDI 操作。

OCX VC6 void AfterRecogFinished (long PlateNum);
Delphi7 procedure AfterRecogFinished(PlateNum: Integer);
DLL/SO C++ virtual void AfterRecogFinished(int PlateNum);
说明 图像抓拍 或 识别过程结束后，将产生该事件，以通知用户进行相关的处理。该事件是任务安全的，可进行 GDI 操作。识别结果的读取应该在该事件中进行，否则可能产生不可预知的错误。只有正式版的软件才会产生该事件。对于 PAL 制的视频流，该事件每秒将产生 25 次。

PlateNum 值	Def.h 文件中的宏定义	含义
0	DSP_RECOG_NO_RESULT	已经允许识别，无识别结果
1	DSP_RECOG_HAS_RESULT	已经允许识别，有 1 个识别结果
-1	DSP_JUST_CAPTURE	已经禁止了识别，但可以抓拍图片
-8000	DSP_MOTION_DETECT	检测到了运动车辆
-8001	DSP_FOUND_CHARS	发现了一些字符。运用该标记可以有效提高抓拍率

调用 setStatEnabled 函数设置为 1 后，内部的运动车辆检测相关功能将自动关闭，即：不会有 DSP_MOTION_DETECT 及 DSP_FOUND_CHARS 通知。

OCX VC6 void AfterFilterStateChanged (long evType);
Delphi7 procedure AfterFilterStateChanged (evType: Integer);
DLL/SO C++ virtual void AfterFilterStateChanged(int Event, unsigned int Target);
说明 图像抓拍 或 识别过程结束后，如果统计过滤器的状态发生变化将产生该事件，以通知用户进行相关的处理。该事件是任务安全的，可进行 GDI 操作。识别结果的读取在该事件中进行是安全的。只有正式版的软件才会产生该事件。

evType 高 16 位 Target	evType 低 16 位 Event	Def.h 文件中的宏定义	含义
表示目标的编号（0-3）	10	DSP_FILTER_OUT_VIEW	运动的车牌可能已出界了
	11	DSP_FILTER_BEFORE_CLEAR	统计过滤器清空之前的通知
	12	DSP_FILTER_TIMER_OVER	本次统计已超

			时
	13	DSP_FILTER_NEW_PLATE	发现了一个不同的新车牌
	14	DSP_FILTER_LIKE_PLATE	发现了一个近拟的车牌
	15	DSP_FILTER_SAME_PLATE	发现了一个相同的车牌
触发条件值	20	DSP_FILTER_FLASH	当前图像是闪光拍摄的

OCX VC6 void AfterImageSizeChanged (long Width,long Height);
Delphi7 procedure AfterImageSizeChanged(Width: Integer; Height: Integer);
DLL/SO C++ virtual void AfterImageSizeChanged(unsigned int Width,unsigned int Height);
说明 当输入的图像大小改变时，将产生该事件。Width 为新的图像的宽度，Height 为高度。

OCX VC6 void AfterRedLampStateChanged(long LampColor,long LampIndex);
Delphi7 procedure AfterRedLampStateChanged (LampColor: Integer; LampIndex: Integer);
DLL/SO C++ virtual void AfterRedLampStateChanged (int LampColor, int LampIndex);
说明 当灯的颜色发生变化时，将产生该事件。LampColor 为灯的颜色，LampIndex 为灯的编号。

OCX VC6 void AfterMotionStateChanged(long Event,long MotionTarget);
Delphi7 procedure AfterMotionStateChanged (Event: Integer; MotionTarget: Integer);
DLL/SO C++ virtual void AfterMotionStateChanged (int Event, int MotionTarget);
说明 当检测到运动的车辆时，将产生该事件。MotionTarget 为运动车辆编号（从 0 到 7），Event 为事件类型。

Event 值：	解释：
DSP_MOTION_OUT_VIEW = 50	运动出界
DSP_MOTION_BEFORE_CLEAR= 51	即将清空
DSP_MOTION_TIMER_OVER = 52	超时
DSP_MOTION_NEW = 53	新目标
DSP_MOTION_LIKE = 54	相近/相同目标
DSP_MOTION_LANE_CHANGED = 55	变道
DSP_MOTION_CROSS_CENTER_Y = 56	通过中间线

OCX VC6 void AfterGetJpgSourceData(Variant* pData,long ImageSize);
Delphi7 procedure AfterGetJpgSourceData(pData: OleVariant; ImageSize: Integer);
DLL/SO C++ virtual void AfterGetJpgSourceData(const void* pData,unsigned int ImageSize);
说明 当收到 JPG/MJPG 格式的图像视频流时，将产生该事件，以方便用户处理（如网络传输）。pData 返回 JPG 数据指针，ImageSize 返回实际大小。

OCX VC6 void AfterCompressedJpgData (Variant* pData,long ImageSize);
Delphi7 procedure AfterCompressedJpgData(pData: OleVariant; Size: Integer);
DLL/SO C++ virtual void AfterCompressedJpgData (const void* pData,unsigned int ImageSize);
说明 当用户使用不等待的方式保存图像为 JPG 格式到内存时将产生该事件。如果用户指定的缓冲内存指针有效，并且空间足够时，系统将自动复制数据到用户指定的内存，并从 pData 返回用户定义的内存指针，以及 JPG 图像大小 ImageSize。如果指针无效或空间不足，则不会复制数据，pData 返回

已压缩的 JPG 数据指针，ImageSize 返回实际大小，以方便用户复制。

DLL/SO 示例： C++

```
void CMainFrame::OnFileCapturejpgtomemory()
{
    // TODO: Add your command handler code here
    gDSP.ImageStreamCopy( NULL, 0, DSP_TRANS_TO_JPG );
}

void CmyDSP::AfterCompressedJpgData(const void* pData,unsigned int Size)
{
    FILE* fp = fopen( "c:\\aaa.jpg","wb");
    if( fp != NULL )
    {
        fwrite( pData, Size, 1, fp );
        fclose(fp);
    }
}
```

OCX 示例： VC6

```
void CTestDlg::OnButton1()
{
    // TODO: Add your control notification handler code here
    m_dsp.ImageStreamCopy(NULL,0, DSP_TRANS_TO_JPG);
}
void CTestDlg::OnAfterCompressedJpgDataPlatedsp2v1(VARIANT FAR*
pData, long ImageSize)
{
    // TODO: Add your control notification handler code here
    void* pImage = pData->byref;
    FILE* fp = fopen( "c:\\aaa.jpg","wb");
    if( fp != NULL )
    {
        fwrite( pImage, ImageSize, 1, fp );
        fclose(fp);
    }
}
```

OCX 示例： C# 2005

```
private void Capture_Click(object sender, EventArgs e)
{
    const int DSP_TRANS_TO_JPG = 0x200;
    int temp_point = 0;
    m_dsp.ImageStreamCopy(ref temp_point, 0, DSP_TRANS_TO_JPG);
}

private void m_dsp_AfterCompressedJpgData(object sender,
AxPlateDSPV2._IPlateDSP2VEvents_AfterCompressedJpgDataEvent e)
{
    IntPtr pJpgData = Marshal.ReadIntPtr(e.pData, 0);
    FileStream fs = new FileStream("c:\\csharp.jpg",FileMode.CreateNew );
    BinaryWriter bw = new BinaryWriter(fs);
    for (int i = 0; i < e.imageSize; i++)
    {
        bw.Write(Marshal.ReadByte(pJpgData, i));
    }
    bw.Close();
    fs.Close();
}
```

OCX 示例：

VB.net 2005

```
Private Sub Capture_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Capture.Click
    Const DSP_TRANS_TO_JPG = &H200
    m_dsp.ImageStreamCopy(vbNullString, 0, DSP_TRANS_TO_JPG)
End Sub

Private Sub m_dsp_AfterCompressedJpgData(ByVal sender As System.Object, ByVal e As AxPlateDSPV2._IPlateDSP2VEvents_AfterCompressedJpgDataEvent) Handles m_dsp.AfterCompressedJpgData
    Dim file As New System.IO.FileStream("c:\vb.jpg", System.IO.FileMode.Create)
    Dim file_write As New System.IO.BinaryWriter(file, System.Text.Encoding.Unicode)
    Dim abyte As Byte
    Dim i As Integer
    For i = 0 To e.imageSize
        abyte = System.Runtime.InteropServices.Marshal.ReadByte( System.Runtime.InteropServices.Marshal.ReadIntPtr(e.pData, 0), i)
        file_write.Write(abyte)
    Next
    file_write.Close()
    file_write.Close()
End Sub
```

OCX 示例：

BCB6

```
void __fastcall TForm1::CaptureClick(TObject *Sender)
{
    m_dsp->ImageStreamCopy(NULL,0, DSP_TRANS_TO_JPG);
}

void __fastcall TForm1::m_dspAfterCompressedJpgData(TObject *Sender,
      Variant *pData, long ImageSize)
{
    void* pImage = pData->byref;
    FILE* fp = fopen( "c:\\aaa.jpg","wb");
    if( fp != NULL )
    {
        fwrite( pImage, ImageSize, 1, fp );
        fclose(fp);
    }
}
```

OCX 示例：

Delphi7

```
procedure TForm1.CaptureClick(Sender: TObject);
var
  tmp_nil: Integer;
  const DSP_TRANS_TO_JPG = $200;
begin
  m_dsp.ImageStreamCopy(tmp_nil,0,DSP_TRANS_TO_JPG);
end;

procedure TForm1.m_dspAfterCompressedJpgData(ASender: TObject;
  var pData: OleVariant; ImageSize: Integer);
var
  stream: TMemoryStream;
  pImage: pbyte;
begin
  pImage := TVarData(pData).VPointer;
```

```
    stream := TMemoryStream.Create;
    stream.Write(pImage^,ImageSize);
    stream.Position := 0;
    stream.SaveToFile('c:\delphi.jpg');
    stream.Destroy;
end;
```

机动车号牌图像自动识别技术规范 API

使用时动态库 aPlate.dll 以及 PlateDSP.dll 必须在系统目录或应用程序目录下。请参考 GA/T 833-2009《机动车号牌图像自动识别技术规范》。
在实际的应用中，许多程序会要求更多更快更复杂的功能，所以我们并不建议您使用这一规范去设计应用程序。

软件初始化函数
函数名称：bool SoftWareInit(void)。
功能说明：将识别软件调入内存并初始化。
函数类型：布尔型，其中 true 表示软件初始化成功，false 表示软件初始化不成功。
函数入口参数：无。

图像识别函数
函数名称：int ImagePlateNum(char *filenam,byte *pnum,int size,int &pc,int &pv,int &pt,byte *ps)。
功能说明：对文件名为 filenam 的图像文件进行号牌识别。
函数类型：整数，其中 0 表示识别成功，1 表示识别不成功，2 表示识别不完整。
函数入口参数：size 表示用于存放识别号牌号码的缓冲区大小，取值为 255 ~258，其中 255 表示仅识别一个号牌号码，256 表示需识别二个号牌号码，257 表示需识别三个号牌号码，258 表示需识别四个号牌号码；filenam 表示需要识别的机动车号牌图像文件名。
函数出口参数：pnum 表示存放号牌号码的缓冲区，按照识别准确率高低存放相应结果，相互间以“%”分隔；pc 表示存放号牌颜色的变量，取值如下：

1	2	3	4	5	6	7	8	…
第 1 个号牌				第 2 个号牌				

第 1 位为 1,表示蓝色；第 2 位为 1,表示黄色；第 3 位为 1,表示白色；第 4 位为 1,表示黑色；第 1 ~4 位全 0,表示其他。

pv 表示存放号牌结构的变量，取值如下：

1	2	3	4	5	6	7	8	…
第 1 个号牌				第 2 个号牌				

第 1 位为 1,表示单排；第 2 位为 1,表示武警；第 3 位为 1,表示警用；第 4 位为 1,表示双排；第 1 ~4 位全 0,表示其他。

pt 表示存放识别时间的变量，以毫秒为单位；ps 表示存放识别可信度的缓冲区，以字符串形式存放，对应识别字符间以“,”分隔，多个识别结果间以“%”分隔。

软件失效函数
函数名称：void SoftwareFree(void)。
功能说明：将识别软件从内存中清除。
函数类型：无。
函数入口参数：无。
函数出口参数：无。

BSTR 字符串与内存泄漏

BSTR 字符串与内存泄漏经常会出现在一些应用软件中。我们知道：如果调用 OCX/COM 组件中返回值的类型为 BSTR 的函数，最终必须主动调用 SysFreeString 来释放。

对于 Borland 公司的 C++编译器（如 BCB6：Borland C++ Builder），getPlateNumber 函数在 PlateDSPV2_TLB.h 中的原型申明为：

```
BSTR __fastcall getPlateNumber(void)
{
   BSTR pVal = 0;
   OLECHECK(this->getPlateNumber((BSTR*)&pVal));
   return pVal;
}
```

由于其返回值类型为 BSTR，使用时应小心，否则容易产生内存泄漏。可以如下处理：

```
WideString Get_OCX_BSTR(BSTR bstr)
{    //利用 WideString 自动释放 bstr
   WideString tmp;
   tmp.Attach(bstr);
   return tmp;
}
AnsiString PlateNumber = Get_OCX_BSTR( dsp->getPlateNumber() );          //取车牌号码
```

或使用如下的 C++通用方法：

```
void Copy_OCX_BSTR(wchar_t* Buff, int BufNum, BSTR bstr)
{
     wcsncpy( Buff, bstr, BufNum );
     ::SysFreeString(bstr);    //调用 Windows API 函数释放
}
wchar_t PlateNumberBuf[20];
Copy_OCX_BSTR( PlateNumberBuf, 20, dsp->getPlateNumber() );          //取车牌号码
```

对于 Microsoft 公司的 C++编译器（如 VC6），getPlateNumber 函数在 platedsp2v.cpp 中的原型申明为：

```
CString CPlateDSP2V::getPlateNumber()
{
     CString result;
     InvokeHelper(0x2f, DISPATCH_METHOD, VT_BSTR, (void*)&result, NULL);
     return result;
}
```

其返回值为 CString 类型，用户可以直接调用，不用担心无法正确释放的问题。

如：

```
CString PlateNumber = m_dsp. getPlateNumber();
```

对于其它编程语言（如 VB，DELPHI 等）都可直接调用，不用担心无法正确释放的问题。

应用软件分发

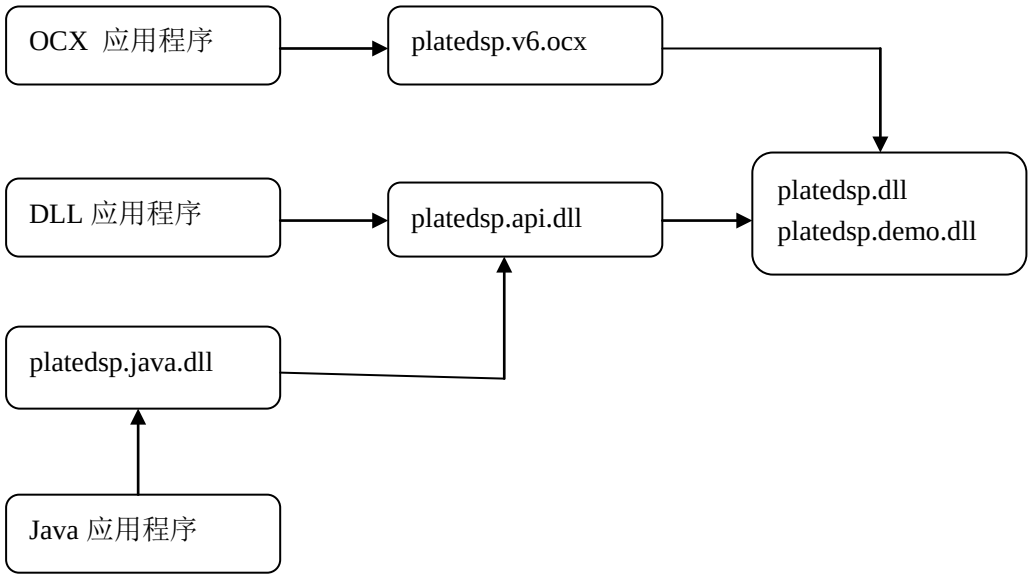
除了安装加密狗的驱动及 DirectX9.0 以外，识别软件的最小安装如下：

文件	安装目录	说明
platedsp.api.dll	System 系统目录	V6 接口 DLL。
platedsp.dll	System 系统目录	V6 正式版核心 DLL。
platedsp.demo.dll	System 系统目录	V6 演示版核心 DLL，正式版调用失败后，也会自动调用该 DLL。
platedsp.v6.ocx	安装目录	V2/V3/V3.5/V5/V6 版 OCX 外壳，应该按 OCX 的方式注册。
GdiPlus.dll	System 系统目录	Windows XP 系统包含的 GDI+库文件，提供图片文件处理。安装在应用程序目录或 Windows 系统目录。Windows XP 及以上系统可不安装
platedsp.video.axis.dll	System 系统目录	可选文件。 AXIS 摄像机支持库。
avcodec-55.dll avfilter-3.dll avutil-52.dll swresample-0.dll swscale-2.dll platedsp.avi.ffmpeg.dll	System 系统目录	可选文件。 ffmpeg 支持库，以支持视频文件，RTSP 网络视频流。

Linux 下请参考安装包中的 readme-on-linux.html 文件。

Android 下请参考安装包中的 readme-on-android.html 文件。

应用软件开发商在制作自己的软件安装包时，可以按上述要求把识别软件核心一起打包分发，但这种方式不利于识别软件的单独升级安装。



HTML 简单测试

PlateDSP 车牌识别组件也可应用于网页文件中，请将以下内容保存在 V2Test.htm 文件中，并在 IE 中打开。

```
<HTML>
<H1>PlateDSP 车牌识别软件 V3 测试网页</H1><p>
本网页可以测试您的系统中是否安装了 PlateDSP 车牌识别软件 V3 版，
如果您的 IE 中的安全设置禁止了调用 Active 控件，下面将显示空白；
如果系统中已经安装了识别控件并允许在 IE 中调用，则下面将显示一个正常的识别控件的界面,
并且以下的按钮可调出相应的对话框。
<p>
<INPUT id=button1 type=button value=参数设置  name=button1>
<INPUT id=button2 type=button value=字符训练  name=button2>
<INPUT id=button3 type=button value=关于  name=button3>
<p>

<HR><center><P>

<OBJECT
 classid="clsid:BD1BBBA2-89CA-4434-BDB7-63B34681E935"
 width=320 height=240 border=1 bordercolor="#0000FF" align=center hspace=0 vspace=0
 id=platedsp
>

</OBJECT>
<SCRIPT LANGUAGE=vbscript>
Sub button1_onclick
platedsp.RecogParamDlg()
End Sub

Sub button2_onclick
platedsp.RecogTrainDlg()
End Sub

Sub button3_onclick
platedsp.AboutDlg()
End Sub
</SCRIPT>

</HTML>
```

附录A 索引