PlateDSP™

Vehicle License Plate Recognition System

Version 6 SDK Manual

SHENZHEN PLATEDSP SOFTWARE DEVELOPMENT CO., LTD.

Website:www.PlateDSP.com

Tel:(+86) 0755-86219206,86219209

Add: Room 1801, East Building, Nanshan Software Park, NO. 10128, Shennan Road, Nanshan District, Shenzhen City

The final revision date 2/18 2014.

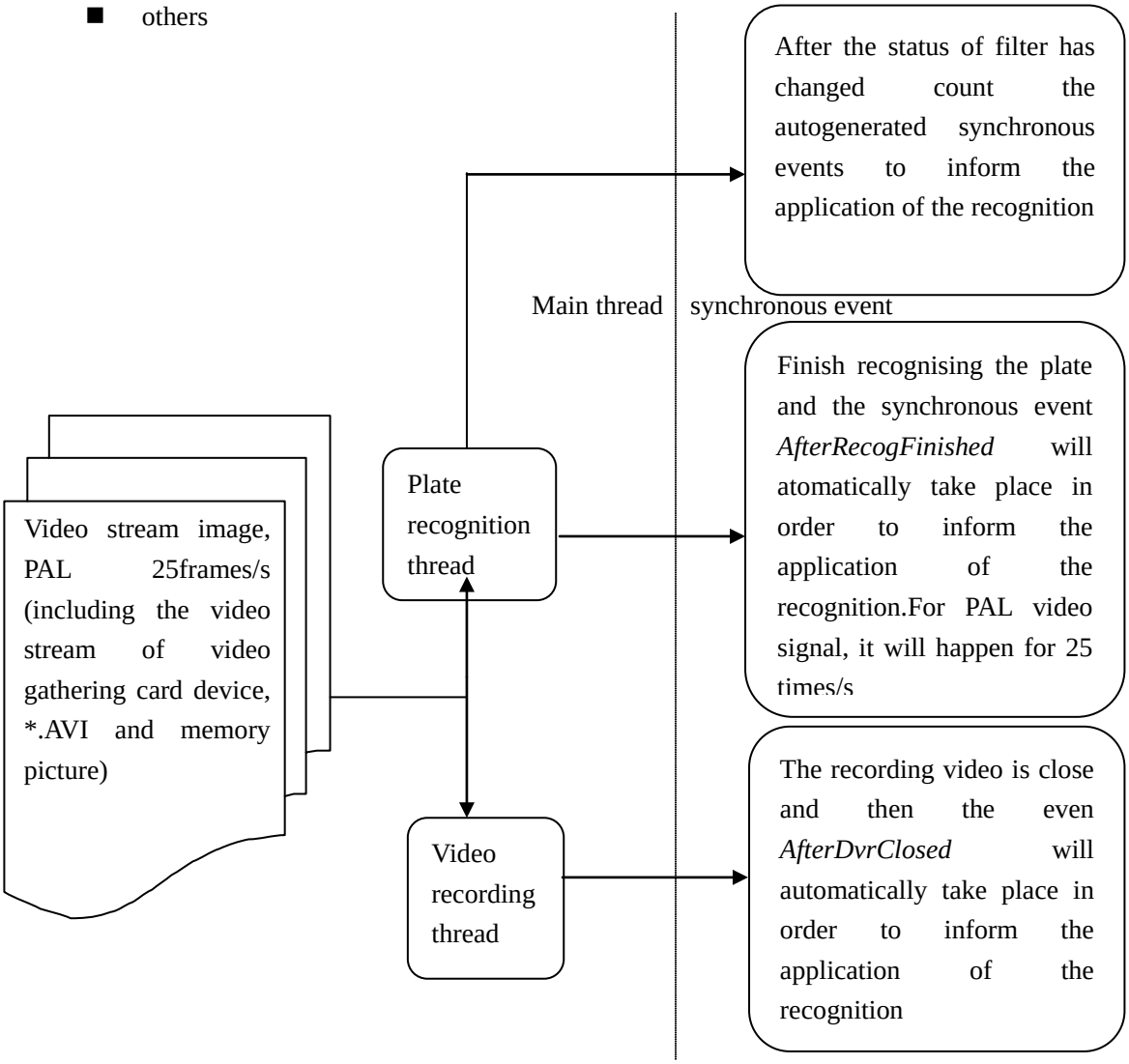
Catalog

Aplication Software Dvelopment API 2
Video Record Replaying Module 4
Video Recording Module 6
Recognition Core Module 11
Statistics Module 30
Video Management Module 32
Application Re-encryption 38
Others 45
Control Event 47
BSTR String and Memory Leakiness 52
The Distripution of the Application software 54
HTML Simple Test 55
Appendix A:Index 56

Aplication Software Dvelopment API

PlateDSP ™ Vehicle License Plate Recognition System V6 contains nearly 100 functions, and it is divided into the following functional modules:

- video record replaying module
- video recording module
- recognition core module
- statistics module
- video management module
- application re-encryption
- Vehicle outline trajectoey vedio tracking
- Tractic light state vedio detection
- others



Next, we will have some classified explaination according to the functional module

Created and destroyed

DLL/SO C HDSP DSPAPI dspCreate(const char* pGuiName);
DLL/SO C++ bool Open(const char* pGuiName);
Note Creat a DSP structure (in c++ class),and returns a pointer of the structure.The pointer will be reference by other function,each created structure is independent,can be operation and operation. pGuiName pointed out the type of use GUI,can use the following some string:

NULL	Use Windows operating system, and the callback event will be sync to the main thread. You must call the dspResetImageDisplayWindow function to set a valid HWND handle.
“gtk”	Use linux operating system and gtk2, and the callback event will be sync to main thread. You must call the dspResetImageDisplayWindow fuction to set a valid GtkDrawingArea* handle.
“qt”	Use linux operating system and QT4, and the callback event will be sync to the main thread. You must call the dspResetImageDisplayWindow fuction to set a valid QWidget* handle.
“async”	The call back event is by other threads of asynchronous calls, you can’t set the window handle.

Successful return non-zero, failure returns NULL.
In c++,the Open function has enclosed the function of SetSyncEventCallback at the same time,don’t need to call it alone to set up a callback,the c++class use the way of heavy event function to callback function.
safty Requires to call in the main thread.

DLL/SO C int DSPAPI dspDestroy(HDSP hdsp);
DLL/SO C++ void Close(void);
note Destroy the structure/class already created.HDSP is the pointer returned by dspCreate.
safty Require to call in the main thread.

DLL/SO C int DSPAPI dspSetSyncEventCallback(HDSP hdsp,void* pObj,PLATEDSP_SYNC_EVET Proc);
Note Set up callback function, HDSP is the pointer returned by dspCreate. pObj point to user defined data pointer,this pointer will be returned to the user when callback Proc,in order to help users identify data sources.Success returns 1,failure returns 0.
Safty Requires to call in the main thread.

DLL/SO C int DSPAPI dspResetImageDisplayWindow(HDSP hdsp,HWND hwnd);
DLL/SO C++ int ResetImageDisplayWindow(HWND hwnd);
Note Set up the image display and the message window handle. hwnd point to a valid window handle.
Success returns 1,failure returns 0
safty Requires to call in the main thread

Video Record Replaying Module

OCX	VC6	long getAviCurrentPosition();
	Delphi7	function getAviCurrentPosition: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviGetCurrentPosition(HDSP hdsp);
DLL/SO	C++	int AviGetCurrentPosition(void);
Note		Read the location of the current replaying video record. Take a frame as a unit. If successful, the result wil1 return 0-N, else return -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);
Note		Reset the location of replaying video record, Pos point the new location. Take a frame as a unit.If successful, it wil1 return 1, else return 0 or-1.

OCX	VC6	long getAviDuration();
	Delphi7	function getAviDuration: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviGetDuration(HDSP hdsp);
DLL/SO	C++	int AviGetDuration(void);
Note		Read the length of the current replaying video record. Take a frame as a unit.If successful, the result wil1 return 0-N, else return -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);
Note		Pause the current replaying video record and forward or go back for given frames based on the current location. If the value of Frames is plus, then forward, if the vulue is minus, then go back. Take a frame as a unit.If successful, it wil1 return 1, else return 0 or-1.

OCX	VC6	long AviIsFinished();
	Delphi7	function AviIsFinished: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviIsFinished(HDSP hdsp);
DLL/SO	C++	int AviIsFinished(void);
Note		Read the status of the current replaying video record.If have finished, it will return 1 and not have finished, it will return 0, else return -1.

OCX	VC6	long AviPause();
	Delphi7	function AviPause: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviPause(HDSP hdsp);
DLL/SO	C++	int AviPause(void);
Note		pause the current replaying video record .If successful, it will return 1, else return 0 or -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);
Note		Open the video Streanming Media assigned by AviFileName.Whether to show

		the picture in the control window or not, you can call the setting of setImageDisplayEnabled in advance. Default is showing the picture.Whether to recognise the video in the process of replaying or not, you can call the function setRecogEnableCount to decide. No matter whether to recognise the picture or not, in the process of replaying, every frame will trigger the AfterRecogFinished event, in the event, you can read safely the recognition, and you can cut the picture. Successful, it will return 1, otherwise it will return 0 or -1.If aviFileName is Null or the length of characters is 0, you can call the function to continue the current process of replaying which has paused or stopped. Some Windows operating system do not install the Microsoft MPEG4 decoder, so they can not replay this kind of video record, but you can install the wmpcdcs8.exe in the CD-ROM directory MPEG4Codec.
OCX	VC6	long AviStop();
	Delphi7	function AviStop: Integer; safecall;
DLL/SO	C	int DSPAPI dspAviStop(HDSP hdsp);
DLL/SO	C++	int AviStop(void);
Note		Stop the current replaying video record. If successful, it will return 1, else return 0 or -1.

Video Recording Module

OCX	VC6	long getDvrBufferFrameNum();
	Delphi7	function getDvrBufferFrameNum: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetBufferFrameNum (HDSP hdsp);
DLL/SO	C++	int DvrGetBufferFrameNum (void);
Note		Read the size of current buffer of the video recorder. Take a frame for a unit.When it returns 0, it means that there is no buffer, that's to say, the video record is immediate, it needn't buffer.When the return value is over 0, it means the method of video recording need the buffer. Default is 0.

OCX	VC6	void setDvrBufferFrameNum(long FrameNum, long bHalf);
	Delphi7	procedure setDvrBufferFrameNum(FrameNum: Integer; bHalf: Integer); safecall;
Note		Set the size of buffer and the proportion of picture of the current video recorder. FrameNum gives the size of buffer, the unit is a frame. When the value of FrameNum is 0, it will use the immediate method to record video; else it will use the buffer. The value bHalf appoints whether to use the half size of the video record or not .When the value is 0, it will use the primal size to record.When not and the size of primal picture is over 384, it will record by half of primal picture. It does not have any return.

DLL/SO	C	int DSPAPI dspDvrSetBufferFrameNum (HDSP hdsp,int FrameNum);
DLL/SO	C++	int DvrSetBufferFrameNum (int FrameNum);
Note		Set the size of buffer FrameNum gives the size of buffer, the unit is a frame. When the value of FrameNum is 0, it will use the immediate method to record video; else it will use the buffer. If successful, it will return 1, else return 0.

DLL/SO	C	int DSPAPI dspDvrSetUseHalfX (HDSP hdsp,int Params);
DLL/SO	C++	int DvrSetUseHalfX (int Params);
Note		Set the proportion of picture of the current video recorder. The value bHalf appoints whether to use the half size of the video record or not .When the value is 0, it will use the primal size to record.When not and the size of primal picture is over 384, it will record by half of primal picture. If successful, it will return 1, else return 0.

OCX	VC6	long DvrCompressDlg();
	Delphi7	function DvrCompressDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrCompressDlg (HDSP hdsp);
DLL/SO	C++	int DvrCompressDlg (void);
Note		Show the dialog of the compressed video record.If your selection is successful, it will return non-0 value, this value is the codes of compressor.If unsuccessful, it will return 0.

OCX	VC6	long getDvrCompressor();
	Delphi7	function getDvrCompressor: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetCompressor (HDSP hdsp);
DLL/SO	C++	int DvrGetCompressor (void);
Note		◆ Read the codes of current compression video record.The return is over 0, it means that the codes of compressor and -1 means that you do not set the compressor (When you choose Automatic mode, the system will automatically choose the right compressor).If unsuccessful, it will return 0. Default is -1. For the automatical mode, the system will automatically find the compressors which

have been installed by the following order:

- Microsoft MPEG-4 Video Codec V3 (MPEG4 Encoder and Decoder)
- Microsoft MPEG-4 Video Codec V2 (MPEG4 Encoder and Decoder)
- Microsoft MPEG-4 Video Codec V1 (MPEG4 Encoder and Decoder)
- DivX Codec(MPEG4 Encoder and Decoder)

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);
Note		Set the code of the current compressed video record.The code can not be 0.If it is over 0, that means the code of the compressor.-1 means that the compressor is chosen automatically by the system.Some Windows operating systems have not installed the Microsoft MPEG4 Encoder and Decoder and this kind of format can not be compressed, but you can install wmpcdcs8.exe of the MPEG4 Code in the CD-ROM directory.

OCX	VC6	CString getDvrCompressorDes();
	BCB6	BSTR __fastcall getDvrCompressorDes(void);
	Delphi7	function getDvrCompressorDes: WideString; safecall;
Note		Read the format's name of the current video record and return the string of the compressor's name. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM, so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

DLL/SO	C	int DSPAPI dspDvrGetCompressorDes (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int DvrGetCompressorDes (wchar_t* pBuff,int BuffNum);
Note		read the format's name of the current video record and return the string of the compressor's name. If successful, it will return 1, else return 0. pBuff assigning the goal address, BuffNum assigning the string numble of the pBuff.

OCX	VC6	long getDvrCurrentPosition();
	Delphi7	function getDvrCurrentPosition: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetCurrentPosition (HDSP hdsp);
DLL/SO	C++	int DvrGetCurrentPosition (void);
Note		Read the current position of the file in the video recorder. Take a frame for a unit. If successful, it will return a value (from 0 to N), unsuccessful, it will return -1.The relation of the value and the recognition is saved in the database, which can help locate the text messages of the plate to the video record information.

OCX	VC6	long getDvrFrameStep();
	Delphi7	function getDvrFrameStep: Integer; safecall;
DLL/SO	C	int DSPAPI dspDvrGetFrameStep (HDSP hdsp);
DLL/SO	C++	int DvrGetFrameStep (void);
Note		Read the step of the current video recorder .Take a frame for a unit .1 means that every frame of the picture must be recorded, 2 means that frames is recorded at every two frames, and so forth .Successful ,it will return 1-N, else return 0 or -1. Default is 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);
Note		Set the step of the current video recorder. Take a frame for a unit. 1 means that every frame of the picture must be recorded, 2 means that it is recorded at every two frames, and so forth. Under the buffer way, you can set its value for -1 to pause the buffer or immediate records, when you need buffer or immediate video, you can resume the value.
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);
Note		Set the name of the current video record as aviFileName (extension is .avi).If the current video mode is immediate; the system will immediately record the video stream. If the current video mode is buffer, the system will only set a file name for it and immediately start the buffer, at this time, the picture will be saved in the given buffer area by the way of FIFO.If it brings buffer overflow, the oldest frame will be deleted.When the buffer mode is activated, it will call the function again and rename it, but it will not affect the picture in the buffer. Successful, it will return 1; else return 0 or -1.
OCX	VC6	long DvrStop(long bWaitFinished);
	Delphi7	function DvrStop(bWaitFinished: Integer): Integer; safecall;
Note		if the current video mode is immediate, the file recording will be closed. If the mode is buffer, the buffering will stop and the system will record the compressed picture in the buffer to *.AVI set by the function of DvrStart. The parameter bWaitFinished points out the way of waiting. 0 means closing the record, but if it does not wait, it will return. 1 means closing the record and the system does not return until it has finished closing (because the process of compressing is executed in the multi-tasking system, we suggest you shoule set bWaitFinished to 0, which can improve the efficiency of CPU). No matter whether it waits or not, the event of AfterDvrClosed will be activated after the video file finishs closing. Successful, it will return 1; else return 0 or -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;
Note		If the video mode is immediate, the file recording will be closed, and the two parameters (aviFileName and bNotClearBuff) will be ignored.If the current video mode is buffer, the buffering will stop and the system will record the compressed picture in the buffer to *.AVI appointed by DvrStart.The parameter bNotClearBuff points out whether to clear up the data in the buffer or not, 0 means clearing up and 1 means saving. The parameter bWaitFinished points out the way of waiting, 0 means closing the record, but it does not wait and directly returns, 1 means closing the record and it does not return until the system has finished closing(because the process of compressing is executed in the multi-tasking system,we suggest you shoule set bWaitFinished to 0, this way can improve the efficiency of CPU).No matter whether it waits or not, the event of AfterDvrClosed will be activated after the video file finishs closing.Successful,it will return 1, else return 0 or -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);

Note Params assigning the pattern of waiting and clear the mode of buffering,can be DSP_WAIT_FINISHED=0x04 or DSP_NOT_CLEAR_BUFF=0x10.
If the video mode is immediate, the file recording will be closed, and the two parameters (FileName and DSP_NOT_CLEAR_BUFF) will be ignored.
If the current video mode is buffer, the buffering will stop and the system will record the compressed picture in the buffer to *.AVI appointed by DvrStart.
If Params assigning DSP_NOT_CLEAR_BUFF is 1,not delete data in the buffer,or else it will delete.
If Params assigning DSP_WAIT_FINISHED is 0, means closing the record, but it does not wait and directly returns, 1 means closing the record and it does not return until the system has finished closing(because the process of compressing is executed in the multi-tasking system,we suggest you shoule set bWaitFinished to 0, this way can improve the efficiency of CPU).No matter whether it waits or not, the event of AfterDvrClosed will be activated after the video file finishs closing.Successful,it will return 1, else return 0 or -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);

Note Only in the video and the buffer mode is valid, the function can help capture the cource picture in the application of run the red light capture. Generally, Num is set to 3 to capture the picture composed by three images near the zebra crossing.Then it will keep the current picure in the buffer as a *.bmp into the address assigned by pDesBuf, BufSize points out the size. If successful, it will return the memory stream of the captured picture,else return 0 or -1.When pDesBuf is NULL or BufSize is 0, it will not copy the data and only return the size, the unit is a Byte.When pDesBuf is not NULL and BufSize is less than the memory that it requires, it will lead to an unsuccessful calling .The parameter bCircumgyrate90 is 0, it means it does not rotate.If its value is 1,it means that it rotates by 90 degrees.When the camera is rotated by 90 degrees at the time of installation, it will make the red light more clear.

Num	explaination
-N	take out an old picture of No. N from the buffer
-1	take out the oldest picture from the buffer
0	take out a picture from the middle of buffer
1	take out the newest picture from the buffer
N（>1）	take out N equidistant pictures from the buffer and compose them into one picture from left to right.

In order to help the use of interpretative development platform, now we add a new method that a recognition control can manage the buffer automatically on its own, the method requires the BufSize is always 4 (the bytes of the long), pDesBuf is pointed to a variable of long, its number ranges from 0 to 15(it shows the number of the internal buffer memory block)

Sample

```
// vc6
// first obtains the size required by memory stream and applys for memory blocks
// and then copys the image.
int ImgSize = m_dsp.DvrImageCopy(NULL,0,3,0);
BYTE* pStream = new BYTE[ImgSize];
if( ImgSize==m_dsp.DvrImgaeCopy( (long*)pStream,ImgSize,3,0 ) )
{//success
    ImageStreamSaveExNoWait (L “c:\\abc.jpg”, (long*)pStream);
```

```
}
delete[] pStream;
//end example

// The following can replace the codes above:
long BuffImageBlock = 0;     //use #0 memory block
if ( m_dsp.DvrImageCopy( & BuffImageBlock, 4, 3, 0 ) > 0 )
{     //sucessful
      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);
Note		Set the overlapped characters information of the current camera. The x points out the value of coordinate in the horizontal direction, y points out the value of coordinate in the vertical direction, the unit is a pixel.When the x an y are negative, the function will clear up all the overlapped information in the location of coordinates set before. The Title or Msg points out the overlapped characters, an empty string means there are not any overlapped characters in the given coordinates(clear up the overlapped information in the relevant coordinates set before).You can set many different coordinates . In order to overlap the real-time mutative information on the every frame, you can set the string in the event of AfterRecogFinished.

This function can set the typeface,for example:
setDvrTitle(0,0,L"<font:name=BLACK,size=32,color=hf08080,bkcolor=h800080>");
setDvrTitle(0,0,L"<font:size=32>");
setDvrTitle(0,0,L"<font:name=Arial,size=24>");
setDvrTitle(0,0,L"<font:color=0xff8080,bkcolor=0x8000ff>");
No blank in the middle.

Recognition Core Module

OCX	VC6	long getCarColor ();
	Delphi7	function getCarColor: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetCarColor (HDSP hdsp);
DLL/SO	C++	int GetCarColor (void);
Note		Return the recognition result of the vehicle color. 0 represents no meaning, 1 represents blue, 2 represents black, 3 represents white, 4 represents yellow, 5 represents red,6 represents green. If unsuccessful, it will return -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX	VC6	CString getColorName (long Color);
	BCB6	BSTR __fastcall getColorName(long Color);
	Delphi7	function getColorName(Color: Integer): WideString; safecall;
Note		Return the string of the color name. The '?' represents unkown color. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

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);
Note		Return the string of the color name.The '?' represents unkown color.pBuff assigning the goal address, BuffNum assigning the size of the string.

OCX	VC6	long getRecogCarColorEnable ();
	Delphi7	function getRecogCarColorEnable: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetRecogCarColorEnabled (HDSP hdsp);
DLL/SO	C++	int GetRecogCarColorEnabled (void);
Note		Return the vehicle color, 1 reprsent auto-recoginition;0 reprsent no recognition.Default value is 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);
Note		Set and return the vehicle color, 1 reprsent auto-recoginition; 0 reprsent no recogniyion.Default value is 0.

OCX	VC6	long getImageByField();
	Delphi7	function getImageByField: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageGetByField (HDSP hdsp);
DLL/SO	C++	int ImageGetByField (void);
Note		Return whether the system automatically processes (identifies or records) the image (video device, replaying video record, picture or memory stream bitmap)

by using the scene or not. 1 means it processes them by using the scene, 0 means it processes them according to the format of the primitive image. Default is 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);
Note		Return whether the system automatically processes (identifies or records) the image (video device, replaying video record, picture or memory stream bitmap) by using the scene or not. 1 means it processes them by using the scene, 0 means it processes them according to the format of the primitive image. Default is 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);
Note		Set the quality of the compressed picture (*.jpg). 1means it is the worst, 100 means it is the best.The better the quality of the picture is, the larger the memory space the picture needed.Successful, it will return the value set before, else it will return 0.

OCX	VC6	long getImageDisplayEnabled();
	Delphi7	function getImageDisplayEnabled: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageGetDisplayEnabled (HDSP hdsp);
DLL/SO	C++	int ImageGetDisplayEnabled (void);
Note		Return whether the system shows the image (video device, replaying video record, and picture or memory stream bitmap) or not. Default value is 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);
Note		Set whether the system shows the image (video device, replaying video record, and picture or memory stream bitmap) or not.

bEnabled/ Params value	meaning
0	not show the image
DSP_DISPLAY_CAR_IMAGE = 0x01	show the inputed image
DSP_DISPLAY_PLATE_IMAGE = 0x02	show the picture of the plate

OCX	VC6	long ImageGetIdentity();
	Delphi7	function ImageGetIdentity: Integer; safecall;
DLL/SO	C	int DSPAPI dspImageGetIdentity (HDSP hdsp);
DLL/SO	C++	int ImageGetIdentity (void);
Note		Read the current image identity number, successful,it will return non-0, else, it will return 0. This function is mainly used to judge whether it is the same picture in the event. Automatic increasing order. 2011/5/16 newly added
safty		Suggest that it should be only called in the event of AfterRecogFinished or AfterFilterStateChanged,otherwise,there maybe an unpredictable task conflict.

OCX	VC6	long ImageIsBest();
	Delphi7	function ImageIsBest: Integer; safecall;

DLL/SO	C	int DSPAPI dspImageIsBest (HDSP hdsp);
DLL/SO	C++	int ImageIsBest (void);
Note		Judge whether the current picture is the best or not according to the result of recognition. Yes, it will return 1, No, it will return 0.
Safty		Suggest that it should be only called in the events of AfterRecogFinished or AfterFilterStateChanged, otherwise, there may be an unpredictable task conflict.
Sample		Please refer to the file (C:/Program /Files/ PlateDSP.V5 /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);
Note		Judge whether the current picture is the best or not according to the result of recognition. Yes, it will return 1, No, it will return 0.
Safty		Suggest that it should be only called in the events of AfterRecogFinished or AfterFilterStateChanged, otherwise, there may be an unpredictable task conflict.

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);
Note		Read the address of the current frame given by pDesBuf, BufSize points out the size in the memory. The bHalf point out whether to use the half of the width of the old picture, when the value is 0, it will use the width of the old picture to capture pictures.When the value is 1 and the width of the old picture is over 384, it will use the half of it to capture pictures.when DSP-AUTO-ZOOM-Y=0x100 is 1,and can stretch the hight of the picture,then the hight of the picture auto-increasing.If successful, it will return the size of the memory stream of the captured pictures, unsuccessful, it will return 0 or -1. When pDesBuf is NULL or BufSize is 0, it will not copy the data and only return the size, the unit is a Byte.When pDesBuf is not NULL and BufSize is less than the memory that it requires, it will lead to the unsuccessful calling. If it has called the overlapped strings given by setImageStreamTitle, the picture will overlap the characters. In order to help the use of interpretative development platform, now we increase a new method that a recognition control can manage the buffer automatically on its own, the method requires that the BufSize is always 4 (the bytes of the “long”), pDesBuf points to a variable of long, its number ranges from 0 to 15(it shows the number of the internal buffer memory block). When bHalf/Params designate DSP-TRANS-TO JPG=0x200,compress the image to user defined memory.when don’t use DSP-WAIT-FINISHED parameter,AfterCompressedJpgData event will emerge.

Sample		//vc6
		//firstly gets the size of memory stream, and applys for the memory blocks,
		//and then copys the image
		int ImgSize = m_dsp.ImageStreamCopy(NULL,0,0);
		BYTE* pStream = new BYTE [ImgSize];
		if(ImgSize==m_dsp.ImgaeStreamCopy((long*)pStream,ImgSize,0))
		{//successful
		ImageStreamSaveExNoWait (L “c://abc.jpg”, (long*) pStream);
		}
		delete[] pStream;
		//end example

```
//The following can also replace the codes above:
long BuffImageBlock = 1;     //use #1 memory blocks
if( m_dsp.ImageStreamCopy( & BuffImageBlock, 4, 0 ) > 0 )
{     //successful
      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);
Note		Read the plate picture of the current image frames which have been identified successfully given by pDesBuf, BufSize points out the size of memory.If successful, it will return the size of memory stream, else return 0 or -1.When pDesBuf is NULL or BufSize is 0, it will not copy the data and only return the size, the unit is a Byte. When pDesBuf is not NULL and BufSize is less than the memory that it requires, it will lead to the unsuccessful calling. In order to help the use of interpretative development platform, now we add a new method that a recognition control can manage the buffer automatically on its own, the method requires the BufSize is always 4 (the bytes of the long), pDesBuf points to a variable of long, its number ranges from 0 to 15(it shows the number of the Internal buffer memory block).
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

Sample		//vc6
		//firstly get the size of memory stream, and apply for the memory blocks,
		//and then copy the image
		BYTE* pStream = new BYTE [ImgSize];
		If (ImgSize==m_dsp. ImageStreamPlateCopy ((long*)pStream,ImgSize))
		{ //successful
		ImageStreamSaveExNoWait (L “c://abc.jpg”, (long*) pStream);
		}
		delete[] pStream;
		//end example
		
		//The following can also replace the codes above:
		long BuffImageBlock = 2; //use #2 memory blocks
		if (m_dsp.ImageStreamPlateCopy(& BuffImageBlock, 4) > 0)
		{ //successful
		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;
Note		Keep the plate picture of the current image frames which have been identified successfully into the picture file assigned by FileName.The format of file is appointed by file name and extension and it can be the picture format (such as*.BMP/*.JPG/*).If successful, it will return 1, else return 0 or -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished or AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX	VC6	long ImageStreamPlateSaveNoWait(LPCTSTR FileName);
	BCB6	long __fastcall ImageStreamPlateSaveNoWait (BSTR FileName);

	Delphi7	function ImageStreamPlateSaveNoWait (const FileName: WideString): Integer; safecall;
Note		Use the thread and Keep the plate picture of the current image frames which have been identified successfully into the picture file assigned by FileName, The format of file is appointed by file name and extension, it can be the picture format (such as *.BMP/*.JPG/*).It returns before the pictures have not fininshed saving, this way can improve the utilization of CPU. If successful, it will return 1, else return 0 or -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
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);
Note		Use the thread and Keep the plate picture of the current image frames which have been identified successfully into the picture file assigned by FileName, The format of file is appointed by file name and extension, it can be the picture format such as BMP/.JPG. It returns before the pictures have not fininshed saving, this way can improve the utilization of CPU. Params assigining whether to use the half width of the old picture or not, when the value is 0,it will use primal size to capture, When DSP-HALFX=0x01 is 1 and the width of the old picture is over 384, it will use the half of the old picture to capture.when DSP-AUTO-ZOOM-Y=0x100 is 1,and the hight of the picture can strecth,then the hight of the picture auto-increasing. If Prams assigining DSP_WAIT_FINISHED=0x04 is 1,wait for the saving before return. If successful, it will return 1, else return 0 or -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
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;
Note		Keep the current image frames into the picture file given by FileName, the format of file is appointed by file name and Extension, it can be the picture format such as BMP/*.JPG. The parameter bHalf appoints whether the system uses the half size of the video record or not .When the value is 0, it will use the primal size to record. When the value is 1 and the size of primal picture is over 384, it will record by half of primal picture. when DSP-AUTO-ZOOM-Y=0x100 is 1,and the hight of the picture can strecth,then the hight of the picture auto-increasing. If successful, it will return 1, else return 0 or -1. If it has called the overlapped strings assigned by setImageStreamTitle, the picture will overlap the characters.
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;
Note		Use the thread and keep the current image frames into the picture file given by FileName, the format of file is appointed by file name and Extension,it can be the picture format (such as *.BMP/*.JPG/*). The parameter bHalf appoints whether the system uses the half size of the video record or not .When the value is 0, it will use the primal size to record.When the value is 1 and the size of primal picture is over 384, it will record by half of primal picture. when DSP-AUTO-ZOOM-Y=0x100 is 1,and the hight of the picture can strecth,then the hight of the picture auto-increasing. It returns before the pictures have not

		fininshed saving, this way can improve the utilization of CPU.If successful, it will return 1, else return 0 or -1. If it has called the overlapped strings assigned by setImageStreamTitle, the picture will overlap the characters.
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);
Note		Use the thread and keep the current image frames into the picture file given by FileName, the format of file is appointed by file name and Extension,it can be the picture format such as.BMP/.JPG. The Parameter bHalf appoints whether the system uses the half size of the video record or not. When the value is 0, it will use the primal size to record. When the value is 1 and the size of primal picture is over 384, it will record by half of primal picture. If Prams assigning DSP_WAIT_FINISHED=0x04 is 1,the wait for saving before return. If successful, it will return 1, else return 0 or -1. If it has called the overlapped strings assigned by setImageStreamTitle, the picture will overlap the characters.
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;
Note		Keep the memory bitmap stream of *.bmp given by pBmpStream into the picture file given by FileName, the format of file is appointed by file name and Extension,it can be the picture format such as .BMP/*.JPG/. If successful, it will return 1, else return 0 or -1.You can use the function to keep the memory bitmap copied by ImageStreamCopy or ImageStreamPlateCopy into a *.jpg. In order to help the use of interpretative development platform,now we add a new method that a recognition control can manage the buffer automatically on its own,the method requires pBmpStream is long, its number ranges from 0 to 7(it shows the number of the Internal buffer memory block)
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;
Note		Use the thread and Keep the memory bitmap stream of *.bmp given by pBmpStream into the picture file given by FileName, The format of file is appointed by file name and Extension,it can be the picture format (such as *.BMP/*.JPG/*). It returns before the pictures have not fininshed saving, this way can improve the utilization of the CPU. If successful, it will return 1, else return 0 or -1. You can use the function to keep the memory bitmap copied by ImageStreamCopy or ImageStreamPlateCopy into a *.jpg. In order to help the use of interpretative development platform,now we add a new method that a recognition control can manage the buffer automatically on its own,the method requires pBmpStream is long, its number ranges from 0 to 7(it shows the number of the Internal buffer memory block)
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);
Note		Use the thread and Keep the memory bitmap stream of *.bmp given by

pBmpStream into the picture file given by FileName, The format of file is appointed by file name and Extension,it can be the picture format which can be supported by the programming interface of.BMP/.JPG.
If Params assigning DSP_HALFX=0x01is 0 save the primal size to record. .When DSP-HALFX=0x01 is 1 and the size of primal picture is over 384, it will record by half of primal picture.
When DSP-AUTO-ZOOM-Y=0x100 is 1,and the hight of the picture can strecth,then the hight of the picture auto-increasing.
If Prams assigining DSP_WAIT_FINISHED=0x04 is 1,wait for the saving before return. If successful, it will return 1, else return 0 or -1.
You can use the function to keep the memory bitmap copied by ImageStreamCopy or ImageStreamPlateCopy into a *.jpg.
In order to help the use of interpretative development platform,now we add a new method that a recognition control can manage the buffer automatically on its own,the method requires pBmpStream is long, its number ranges from 0 to 7(it shows the number of the Internal buffer memory block).

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);

Note		set the overlapped character information of the current picture. X points out the value of coordinate in the horizontal direction, y points out the value of coordinate in the vertical direction,the unit is a pixel.When the x or y is negative,it will clear up all the overlapped information in the location of coordinates set before.The Title and Msg points out the overlapped characters,empty string means there are not overlapped characters in the given coordinates(clear up the overlapped information in the relevant coordinates set before).You can set much different overlapped information of the coordinates.
		This function can set the typeface,for example:
		setImageStreamTitle(0,0,L"<font:name=BLACK, size=32, color=hf08080, bkcolor=h800080>");
		setImageStreamTitle (0,0,L"<font:size=32>");
		setImageStreamTitle (0,0,L"<font:name=Arial, size=24>");
		setImageStreamTitle (0,0,L"<font:color=0xff8080, bkcolor=0x8000ff>");
		No blank in the middle.

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);
Note		Buffer the image frame to the memory module of group B from id.If success.return 1,else 0.The range of id is from 0 to 255(the memory module NO.of group B).The function just like ImageStreamCopy,but it can't transfer the image type,with high efficiency.

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);
Note		Transfer and copy the assigning Id (0-255）. module image to the pDesBuf goal

address. BufSize assigning the size of the memory.
Params assigning DSP_HALFX=0x01is 0,capture with the primate width of the image,When the value is 1 and the width of the old picture is over 384, it will use the half of it to capture pictures. When DSP-AUTO-ZOOM-Y=0x100 is 1,and the hight of the picture can strecth,then the hight of the picture auto-increasing.If successful, it will return the size of the memory stream of the captured pictures, unsuccessful, it will return 0 or -1. When pDesBuf is NULL or BufSize is 0, it will not copy the data and only return the size, the unit is a Byte.When pDesBuf is not NULL and BufSize is less than the memory that it requires, it will lead to the unsuccessful calling. If it has called the overlapped strings given by setImageStreamTitle, the picture will overlap the characters. In order to help the use of interpretative development platform, now we increase a new method that a recognition control can manage the buffer automatically on its own, the method requires that the BufSize is always 4 (the bytes of the "long"), pDesBuf points to a variable of long, its number ranges from 0 to 15(it shows the number of the internal group A buffer memory block).when Params assigning DSP_BUFF_TO_BUFF=0X80 is 1,the integer range of pDesBuf point to is from 1-255(group B memory block).
When bHalf/Params assigning DSP_TRANS_TO_JPG=0x200,will compress the image to user-defined memory,when don't use DSP_WAIT_FINISHED paremeter,it will emerge AfterCompressedJpgData event.

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);
Note		Save the id assigning Id (0-255). group B memory image to the FileName assigning file,the type is assigned by the name of the file,it can be .BMP or .JPG Params assigning DSP_HALFX=0x01is 0,capture with the primate width of the image,When the value is 1 and the width of the old picture is over 384, it will use the half of it to capture pictures. When DSP-AUTO-ZOOM-Y=0x100 is 1,and the hight of the picture can strecth,then the hight of the picture auto-increasing.bWaitFinished assigining whether wait for the return untile save or not. If successful, it will return 1,unsuccessful, it will return 0 or -1. Use the function saving the memory image copyed by BufferImageStream / BufferCopy / ImageStreamCopy or ImageStreamPlateCopy to the JPG file.The function is similar as 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);
Note		Waiting for the recognition by message circle.If success,return 1.else 0.

Tick	note
0	Inquiry wether it over.If over,retun 1,else 0.
(-1)	Waiting and inquiry wether it over or not.If over,return 1.
Other N	Waiting N ms and inquiry wether it over or not. If over,retun 1,else 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);
Note		Waiting for saving the file by message circle. If success,return 1.else 0

Tick	note
0	Inquiry wether it over.If over,retun 1,else 0.
(-1)	Waiting and inquiry wether it over or not.If over,return 1..
Other N	Waiting N ms and inquiry wether it over or not. If over,retun 1,else 0.

OCX	VC6	long getPlateBrightness();
	Delphi7	function getPlateBrightness: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateBrightness (HDSP hdsp);
DLL/SO	C++	int GetPlateBrightness (void);
Note		Read the value of the brightness of the plate which has been identified successfully, If successful, it will return a value ranging from 1 to 255, and the highest is 255.When there is no any recognition, it will return a negative in order help auto-adjust the brightess of the image, and its return ranges from -1 to -255, the lowest is -255(Nowadays this function can only support some high definition capture device).
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX	VC6	long getPlateColor();
	Delphi7	function getPlateColor: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateColor (HDSP hdsp);
DLL/SO	C++	int GetPlateColor (void);
Note		Read the color codes of the plate which has been successfully identifed. 0 represents no meaning, 1represents blue, 2 represents black, 3 represents white, 4 represents yellow, 5 represents red,6 reprsent green. If unsuccessful, it will return -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX	VC6	CString getPlateColorName();
	BCB6	BSTR __fastcall getPlateColorName(void);
	Delphi7	function getPlateColorName: WideString; safecall;
Note		Read the color string of the plate which has been successfully identifed. The value of return can be blue, black, white, yellow,green or '?'. The '?' represents unkown color. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX	VC6	long getPlateCount();
	Delphi7	function getPlateCount: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateCount (HDSP hdsp);
DLL/SO	C++	int GetPlateCount (void);
Note		Read the count of the plates which have been successfully identified, the more

		the times of recognition are, the higher the count is. When you set they can be cumulatived, the return represents the count of the plate in the statistic queue, and when they can be counted, the system will return 1, else return -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
OCX	VC6	long getPlateDir();
	Delphi7	function getPlateDir: Integer; safecall;
Note		Read the direction of the moving vehicle and recognise the result. A positive number represents the plate is moving from the top to down and a negative number represents the opposite. The bigger absolute value is, the higher the reliability of recognition is. 0 means it can not be idenfication. You can call the setStatEnabled and start the counting function to recognise more accurately.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
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);
Note		Read the location of the plate, which have been successfully identified, in the old image. The “pLeft”, “pTop”, “pRight”, “pBottom” are the pointers of long,and respectively represent these variables of”Left”, “ Top”, “Right” , “Bottom” .The unit is a Pixel. Unsuccessful, it will return -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce..

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

OCX	VC6	CString getPlateNumber();
	BCB6	BSTR __fastcall getPlateNumber(void);
	Delphi7	function getPlateNumber: WideString; safecall
Note		Read the string of the plate numbers which have been successfully identified.Unsuccessful, it will return empty string. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
DLL/SO	C	int DSPAPI dspGetPlateNumber (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int GetPlateNumber (wchar_t* pBuff,int BuffNum);
Note		Read the string of the plate numbers which have been successfully identified.pBuff assigning the goal address,BuffNum assigning the numble of the goal string place. If success,return 1.else 0.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and

		AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
OCX	VC6	long getPlateReliability();
	Delphi7	function getPlateReliability: Integer; safecall;
DLL/SO	C	int DSPAPI dspGetPlateReliability (HDSP hdsp);
DLL/SO	C++	int GetPlateReliability (void);
Note		Read the confidence level of the plate which has been successfully identified.The return ranges from 0 to 100.100 represents the highest.Unsuccessful, it will return 0.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwis, unpredictable tasks conflict may take palce.
OCX	VC6	long getPlateReliabilityByChar(long Index);
	Delphi7	function getPlateReliabilityByChar(Index: Integer): Integer; safecall;
Note		Read the confidence level of the characters given by Index in the plate numbers which have been successfully identified.Index is 0, it means the first character and the rest can be deduced by analogy. The return ranges from 0 to 100.100 represents the highest.Unsuccessful, it will return 0.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.
DLL/SO	C	int DSPAPI dspGetPlateReliabilityByChar (HDSP hdsp,unsigned char* pBuff,int BuffNum);
DLL/SO	C++	int GetPlateReliabilityByChar (unsigned char* pBuff,int BuffNum);
Note		Read the confidence level of each character in the plate which has been successfully identified.(0-100) pBuffassigning the goal address ,BuffNum assigning the goal address of the byte. If success,return the numble of the string,else 0.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged
OCX	VC6	long getPlateSpeed (long* pSpeedX,long* pSpeedY);
	Delphi7	function getPlateSpeed (var pSpeedX: Integer; var pSpeedY: Integer): Integer; safecall;
Note		Read speed of the plate which has been successfully identified. The unit is the pixels/m.Successful, it will return 1, else return 0 or -1.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

DLL/SO	C	int DSPAPI dspGetPlateSpeed (HDSP hdsp,int * px,int * py);
DLL/SO	C++	int GetPlateSpeed (int * px,int * py);
Note		Read speed of the plate which has been successfully identified. The unit is the pixels/m. In the px,py assigning address,get the speed of X direction and Y direction. If the return value is positive numble,that is the plate move up to down,else down to up.The higher of the absolute value,the more reliable.0 reprsent can't recognize the direcrion.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX	VC6	long getPlateTypeId();
	Delphi7	function getPlateTypeId: Integer; safecall;
Note		Read the type code of the plate which has been successfully identified. Successful, it will return non - 0, else it will return 0 or -1. Please refer to the

return of getRecogCfgUseTemplate.

Safty Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX VC6 CString getPlateTypeName();

BCB6 BSTR __fastcall getPlateTypeName(void);

Delphi7 function getPlateTypeName: WideString; safecall;

Note Read the type code of the plate which has been successfully identified.Successful,the return will be Civil License Plate (92 edition), Civil Freight Vehicle Rear License Plate(double rows), Civil License Plate(2002 individuation), Police Cars License Plate(*police), Armed Police License Plate(WJ*, 2007edition), Military License Plate(2004 edtion),else it wll return an empty string. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

Safty Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

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

DLL/SO C++ int GetPlateTypeName (wchar_t* pBuff,int BuffNum);

Note Read the type code of the plate which has been successfully identified.Successful,the return will be Civil License Plate (92 edition), Civil Freight Vehicle Rear License Plate(double rows), Civil License Plate(2002 individuation), Police Cars License Plate(*police), Armed Police License Plate(WJ*), Military License Plate(2004 edtion).

pBuff assigning the goal address, BuffNum assigning the size of the string space.

If success,return type of the numble,else 0.

Safty Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX VC6 long getPlateTypeIdEx();

Delphi7 function getPlateTypeIdEx: Integer; safecall;

Note Read the recognition mark code of operating vehicle which has been successfully recognition.Successful, it will return non-0, else return 0 or -1.

code	meaning
1	Shenzhen rented

Safty Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX VC6 CString getPlateTypeName Ex();

BCB6 BSTR __fastcall getPlateTypeNameEx(void);

Delphi7 function getPlateTypeNameEx: WideString; safecall;

Note Read the mark type of the operation vehicle which has been successfully identified. Successful,it will return" Shenzhen rented",else it will return empty string.Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM, so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

Safty Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable tasks conflict may take palce.

OCX VC6 long getRecogCfgMinReliability();

Delphi7 function getRecogCfgMinReliability: Integer; safecall;

DLL/SO	C	int DSPAPI dspRecogCfgGetMinReliability (HDSP hdsp);
DLL/SO	C++	int RecogCfgGetMinReliability (void);
Note		Read the configuration parameter --lowest confidence level.Its return ranges 0 to 100. 100 means that it will not output the indentification result unless its accuracy is up to 100% (In fact, there is no 100% accurate plate recognition result). Default is 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);
Note		Set the configuration parameter -- lowest confidence level. 100 means it will not output the indentification result unless its accuracy is up to 100%.
OCX	VC6	long getRecogCfgPlateMaxHeight();
	Delphi7	function getRecogCfgPlateMaxHeight: Integer; safecall;
Note		Read the recognition configuration parameter--MaxHeight.The return ranges from 0 to N. Default is 60.
OCX	VC6	void setRecogCfgPlateMaxHeight(long Height);
	Delphi7	procedure setRecogCfgPlateMaxHeight(Height: Integer); safecall;
Note		Set the recognition configuration parameter--MaxHeight.In general, the user does not set this value.
OCX	VC6	long getRecogCfgPlateMaxWidth();
	Delphi7	function getRecogCfgPlateMaxWidth: Integer; safecall;
Note		Read the recognition configuration parameter -- MaxWidth.The return ranges from 0 to N.Default is 220.
OCX	VC6	void setRecogCfgPlateMaxWidth(long Width);
	Delphi7	procedure setRecogCfgPlateMaxWidth(Width: Integer); safecall;
Note		Set the recognition configuration parameter--MaxWidth.In general, the user does not set the value.
OCX	VC6	long getRecogCfgPlateMinHeight();
	Delphi7	function getRecogCfgPlateMinHeight: Integer; safecall;
Note		Read the recognition configuration parameter -- MinHeight.The return ranges from 0 to N.Default is 20.
OCX	VC6	void setRecogCfgPlateMinHeight(long Height);
	Delphi7	procedure setRecogCfgPlateMinHeight(Height: Integer); safecall;
Note		Set the recognition configuration parameter--MinHeight.In general, the user does not set the value.
OCX	VC6	long getRecogCfgPlateMinWidth();
	Delphi7	function getRecogCfgPlateMinWidth: Integer; safecall;
Note		Read the recognition configuration parameter--MinWidth.The return ranges from 0 to N. Default is 80.
OCX	VC6	void setRecogCfgPlateMinWidth(long Width);
	Delphi7	procedure setRecogCfgPlateMinWidth(Width: Integer); safecall;
Note		Set the recognition configuration parameter--MinWidth.In general, the user does not set the value.

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);
Note		Set the recognition configuration parameter to assigning min and max value. the user does not set the value.
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);
Note		Set the recognition configuration parameter to get the min and max value.
OCX	VC6	CString getRecogCfgProvince();
	BCB6	BSTR __fastcall getRecogCfgProvince(void);
	Delphi7	function getRecogCfgProvince: WideString; safecall;
Note		Read the string of the recognition configuration parameter --Pre-number.It does not return an empty string. Default is an empty string. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.
DLL/SO	C	int DSPAPI dspRecogCfgGetProvince (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int RecogCfgGetProvince (wchar_t* pBuff,int BuffNum);
Note		Read the string of the recognition configuration parameter –‘re-numble of the plate’ string. pBuff assigning the goal address, BuffNum assigning the numble of the string space. If success,return 1,else 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);
Note		Set the string of the recognition configuration parameter--Pre-number. You can input the codes of province and the city, for example,”Jing A” and “jing” and you can enter none. When you input the area code, the system can replace the relevant character when the character is not clear, it can improve the effect of recognition in a disguised form, which can greatly reduce the opportunity of misrecognition and reduce the opportunity of manual modification. The system is very practical.Default is an empty string.You can also set in the dialog of the RecogParamDlg.
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);
Note		Read the recognition configuration parameter -- Range of the recognition .The

unit is %, the value rangs from 0 to 100. The default: pLeft: 0, pTop: 0, pRight: 100, pBottom: 100, pLeft, pTop, pRight and pBottom are the pointer of long ,they respectively means the percent of the coordinates of left ,top, right and bottom in the image.

Sample
```
//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);`

Note Set the recognition configuration parameter -- range of image recognition .The unit is %. The value ranges from 0 to 100. The user does not set the parameter. You can also set in the dialog of the RecogParamDlg.

OCX VC6 `long getRecogCfgUseTemplate();`

Delphi7 `function getRecogCfgUseTemplate: Integer; safecall;`

DLL/SO C `int DSPAPI dspRecogCfgGetUseTemplate (HDSP hdsp);`

DLL/SO C++ `int RecogCfgGetUseTemplate (void);`

Note Read the recognition configuration parameters -- platetemplate .Return is a value from low bit B0 to high bit B31:

bit	License plate template	0	1
B0	Civil License Plate (92 edition)	FR	PR
B1	Civil Freight Vehicle Rear License Plate(double rows)	FR	PR
B2	Civil License Plate(2002 individuation)	FR	PR
B3	Police Cars License Plate(*police)	FR	PR
B4	Armed Police License Plate(WJ*)	FR	PR
B5	Military License Plate(2004 edtion)	FR	PR

Default is 61 (D) = 3D (HEX) = 111101 (B).

Note: FR means Forbid Recognizing and PR means Permit Recognising

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);`

Note Set the rigion and type of the license plate temple.

PLATE-TYPE-ID	Set the type of the license plate
PLATE-TYPE-ID-OR-DEFAULT	Set the designated license plate type,and the default type.
PLATE-TYPE-ID-TAXI	Abandoned function
PLATE-TYPE-CONTAINER	Set the rigion of the license plate(country/rigion) it valid only have the overseas authorize.

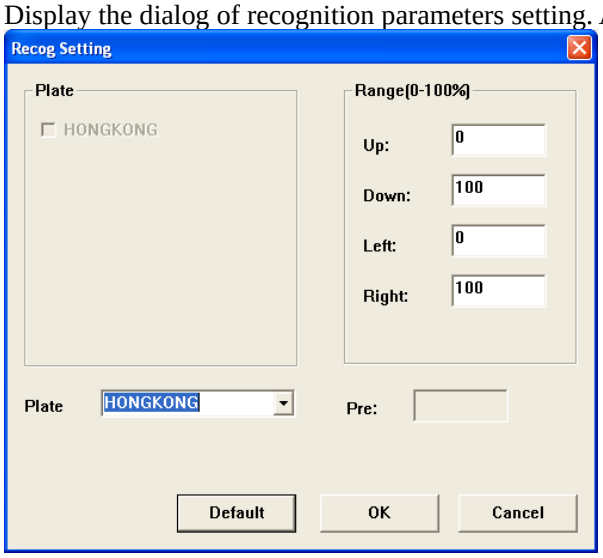
Sample:
```
gDSP.RecogCfgSetUseTemplate(PLATE_TYPE_CONTAINER | PLATE_CONTAINER_ID_HK);
gDSP.RecogCfgSetUseTemplate(PLATE_TYPE_ID_OR_DEFAULT);
gDSP.RecogCfgSetUseTemplate(PLATE_TYPE_ID_NORMAL5);
```

please refer to the defination in platedsp-def.h

OCX	VC6	long getRecogEnableCount();
	Delphi7	function getRecogEnableCount: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogGetEnableCount (HDSP hdsp);
DLL/SO	C++	int RecogGetEnableCount (void);
Note		Read the value of recognition permission.If the value is -1, it means continuous recognition, if it is a positive number N, it means that the system will automatically stop after recognising N frames of image and the count will decrease progressively till 0. If it is 0, it will stop recognising, at then time, it can also capture the image, but not recognise anything.

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);
Note		Set the parameter - recognition permission. If the count is -1, it means continuous recognition, if it is a positive number N, it means that the system will automatically stop after recognising N frames image and the count will decrease progressively till 0, if it is 0, it will stop recognising, at this time, it can also capture the image, but not recognise anything.

OCX	VC6	long RecogParamDlg();
	Delphi7	function RecogParamDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogParamDlg (HDSP hdsp);
DLL/SO	C++	int RecogParamDlg (void);
Note		Display the dialog of recognition parameters setting. As follows:



OCX	VC6	long RecogStartWithFile(LPCTSTR FileName);
	BCB6	long __fastcall RecogStartWithFile(BSTR FileName);
	Delphi7	function RecogStartWithFile(const FileName: WideString): Integer; safecall;
Note		Recognise the file which is assigned by FileName.The extention of FileName can be *.BMP/*.JPG/*. Because the recognition is compeleted in another thread, after the function returns, the recognising may not have finished, at this time, it is not safe to read the recognition, you shoule do it in the event of AfterRecogFinished.When the function calls successfully, it will return 1, else return 0 or -1 and the cause may be the wrong picture's format or may be that the last recognition has not fininshed.Whether to show the picture in the control window, you can call the setImageDisplayEnabled in advance. Default is to

		show the picture. For *.BMP, it only supports four common formats of --RGB15、RGB16、RGB24、RGB32 and the format from top to bottom or whereas. Because of the high utility rate of CPU,it is not be recommended.
OCX	VC6	long RecogStartWithFileWait(LPCTSTR FileName);
	BCB6	long __fastcall RecogStartWithFileWait (BSTR FileName);
	Delphi7	function RecogStartWithFileWait (const FileName: WideString): Integer; safecall;
Note		Recognise the file which is assigned by fileName. The extention of FileName can be *.BMP/*.JPG/*.function will return until finished recognise,when returned,you can read the recognition results safely,When the function calls successfully, it will return 1, else will return 0 or -1 and the cause may be the wrong picture's format or may be that the last recognition has not fininshed. Whether to show the picture in the control window, you can call the setImageDisplayEnabled in advance. Default is to show the picture. For *.BMP, it only supports four common formats of --RGB15、RGB16、RGB24、RGB32 and the format from top to bottom or whereas. Because of the high utility rate ofCPU,it is not be recommended.
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);
Note		Recognise the file which is assigned by fileName. The extention of FileName can be *.BMP/*.JPG/*. If Params assigning DSP_WAIT_FINISHED=0x04 is 1,then return after recognition. If Params assigning DSP_WAIT_FINISHED=0x04 is 0,then return and don't need to wait . Because the recognition is compeleted in another thread, after the function returns, the recognising may not have finished,at the time, it is not safe to read the recognition, you should read the result in the event of AfterRecogFinished. When the function calls successfully, it will return 1, else will return 0 or -1 and the cause may be the wrong picture's format or may be that the last recognition has not fininshed.Whether to show the picture in the control window, you can call the setImageDisplayEnabled in advance.Default is to show the picture. For *.BMP, it only supports four common formats of --RGB15、RGB16、RGB24、RGB32 and the format from top to bottom or whereas. Because of the high utility rate of CPU,it is not be recommended.
OCX	VC6	long RecogStartWithMem(long* pImgData);
	Delphi7	function RecogStartWithMem(var pImgData: Integer): Integer; safecall;
Note		Recognise the image data in the memory which is assigned by pImgData. before the system calls firstly the function, you should call the function setRecogWithBitmapHeader4Mem or setRecogImageFormat4Mem to point all kinds of parameters, at the time, it is not safe to read the recognition, you should read the result in the event of AfterRecogFinished. Whether to show the picture in the control window, you can call the setImageDisplayEnabled in advance.Default is to show picture. When the function calls successfully, it will return 1, else will return 0 or -1 and the cause may be the wrong picture's format or may be that the last recognition has not fininshed. When calling the function, you can use the video stream identication (fast) or file recognition (accurater but slower). Please refer to the specific use of setRecogWithBitmapHeader4Mem. Because of the high utility rate ofCPU,it is not be recommended.
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);
Note		Recognise the image data in the memory which is assigned by pImgData. The data is composed by numbers of *.BMP, before the system calls firstly the function, you should call the function setRecogWithBitmapHeader4Mem or SetRecogImageFormate4Mem to point all kinds of parameters, size specify the size of the image data,if it is the dynamic data,(such as :JPG image data),you must specify the size,otherwise return 0. If Params assigning DSP_WAIT_FINISHED=0x04 is 1,then return after recognition.otherwise,return immediately.at the time, it is not safe to read the recognition, you should read the result in the event of AfterRecogFinished. Whether to show the picture in the control window, you can call the setImageDisplayEnabled in advance.Default is to show picture. When the function calls successfully, it will return 1, else will return 0 or -1 and the cause may be the wrong picture's format or may be that the last recognition has not fininshed. When calling the function, you can use the video stream identication (fast) or file recognition (accurater but slower). Please refer to the specific use of setRecogWithBitmapHeader4Mem.
OCX	VC6	long RecogStartWithMemWait(long* pImgData);
	Delphi7	function RecogStartWithMemWait(var pImgData: Integer): Integer; safecall;
Note		Recognise the image data in memory which is assigned by pImgData.this data is bitmap data arrange by*.BMP format. before the system calls firstly the function, you should call the function setRecogWithBitmapHeader4Mem to point all kinds of parameters, Because the recognition is compeleted in another thread, after the function returns, the recognising may not have finished, at this time, it is not safe to read the recognition, you shoule do it in the event of AfterRecogFinished.When the function calls successfully, it will return 1, else return 0 or -1 and the cause may be the wrong picture's format or may be that the last recognition has not fininshed.Whether to show the picture in the control window, you can call the setImageDisplayEnabled in advance. Default is to show the picture. For *.BMP, it only supports four common formats of --RGB15、RGB16、RGB24、RGB32 and the format from top to bottom or whereas. Because of the high utility rate ofCPU,it is not be recommended.
OCX	VC6	long RecogTrainDlg();
	Delphi7	function RecogTrainDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspRecogTrainDlg (HDSP hdsp);
DLL/SO	C++	int RecogTrainDlg (void);
Note		Because the version V6 introduces a new fast artificiao intelligence algorithm, but the algorithm of character trainning is very slow and the trainning may take dozens of hours for a time, and its process is very complicated and difficult for the users. So begining from the version V6, our company will not provide the interface of character trainning. But if you need it indeed, please send us the pictures or videos with the vehicle license plate charactors and we will be glad to offer you our services.
OCX	VC6	void setRecogWithBitmapHeader4Mem(long* pBitmapInfoHeader, long bFastRecog);
	Delphi7	procedure setRecogWithBitmapHeader4Mem(var pBitmapInfoHeader: Integer; bFastRecog: Integer); safecall;
Note		Set the picture format and method of momory image recognition of RecogStartWithMem. The format is appointed by pBitmapInfoHeader, owing to the limited data type supported by standard OCX pointer, the

		pBitmapInfoHeader is a long, but in fact, it is a BITMAPINFOHEADER.When bfastRecog is 0, it is file recognition (accurater but slower), 1 is video stream recognition (fast). For BMP, it only supports four common formats of-- RGB15、RGB16、RGB24、RGB32 and the format from top to bottom or whereas. Support YUY2,UYVY,I420,YV12,Y42B.Support BAYER GB/GR/BG/RG.Support 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);
Note		Set the method of recogniti and the type of the memory image.The type of the image can be assigned by Format,can definit in the platedsp_def.h: 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 When Params assigning DSP_FAST_RECOG=0x01 is 0, it is file recognition (accurater but slower), 1 is video stream recognition (fast).

Statistics Module

When the System is initializing, the statistics function has been closed by default.In order to open it, please call setStatEnabled, V3 to V6 also incorporate a statistical filter to help users process the recognition of mving plate.

OCX	VC6	long StatClear();
	Delphi7	function StatClear: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatClear (HDSP hdsp);
DLL/SO	C++	int StatClear (void);
Note		Clear the statistical data in the statistical queue.
Safty		Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable task conflict may take palce.

OCX	VC6	long getStatColorUsed();
	Delphi7	function getStatColorUsed: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatGetColorUsed (HDSP hdsp);
DLL/SO	C++	int StatGetColorUsed (void);
Note		Return in the the statistical method whether to allow to contrast the plate color, 1 is Enable and 0 is Disable.Default is 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);
Note		Set in the the statistical method whether to allow to contrast the plate color or not. For the bUsed/Params, 1 is Enable and 0 is Disable.

OCX	VC6	long getStatEnabled();
	Delphi7	function getStatEnabled: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatGetEnabled (HDSP hdsp);
DLL/SO	C++	int StatGetEnabled (void);
Note		Return whether to start Stating or not. 1 is allowed, 0 is forbidden. Default is 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);
Note		Set whether to start Stating or not. For the bEnabled/Params, 1 is allowed, 0 is forbidden. After allowing Stating, the relevant function of moving vehicle inspection built-in will be closed automatically.

OCX	VC6	long getStatMaxTime();
	Delphi7	function getStatMaxTime: Integer; safecall;
DLL/SO	C	int DSPAPI dspStatGetMaxTime (HDSP hdsp);
DLL/SO	C++	int StatGetMaxTime (void);
Note		Return the max time of Stating, the unit is a millsecond. Default is 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);

Note Set the max time of Stating, the unit is a millsecond.The old record which is beyond the given time will be cleared automatically from the queue.Default is 1000.

OCX VC6 long StatPasteBestRecord(long bUse);

Delphi7 function StatPasteBestRecord(bUse: Integer): Integer; safecall;

Note Set the method which can be used to read the current recognition.

edition	bUse high 16	bUse low 16	meaning
	the number of objects (0 to 3)	0	Read the new recognition result.When the event AfterRecogFinished happens, the system will automatically set it to be 0.
		1	Read the recognition result in the statistical queue.
New function of V3		2	Read the recognition result in the statistical filter.
New function of V3.5	Most objects(1-4)	3	The largest number (1 to 4) of objects which can be followed by our system. Default is 1.

Other functions affected::getPlateColor,getPlateColorName,getPlateNumber,getPlateReliability, getPlateReliabilityByChar,getPlateTypeName,getPlateTypeId,getPlateTypeNameEx,getPlateTypeIdEx.

Safty Suggest that you should call it only in the event of AfterRecogFinished and AfterFilterStateChanged, otherwise, unpredictable task conflict may take palce.

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);

Note Set the max-goal.

Like transfer 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);

Note Set the method which can be used to read the current recognition.

Id	Type	meaning
numble （0-3）	0	Read the new recognition result.When the event AfterRecogFinished happens, the system will automatically set it to be 0.
	1	Read the recognition result in the statistical queue.
	2	Read the recognition result in the statistical filter

Like transfer StatPasteBestRecord（ （Id<<16） | Type ）

Video Management Module

OCX	VC6	long getVideoBrightness();
	Delphi7	function getVideoBrightness: Integer; safecall;
Note		Return the value of the current video device light. Successful, it will be 0 to 100, else it will return -1. Default is 50.
OCX	VC6	void setVideoBrightness(long Brightness);
	Delphi7	procedure setVideoBrightness(Brightness: Integer); safecall;
Note		Set the value of the current video device light.It is unsuccessful unless you have opened successfully the video device. Successful, it will return 1, else return 0 or -1.Set the range is from 1 to 100.High-camera:when the camera support automatic gain(Daheng),0 is automatic gain.Transmis the positive value to the camera,no need normalization from 1 to 100.
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);
Note		Return the capturing Pixel of the current video device, pWidth and pHeight is long pointers.They point respectively a long which receives the width and height of a data. Successful, it will return 1, else return 0 or -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);
Note		Set the capturing Pixel of the current video device, pWidth and pHeight are the width and height of a data. Successful, it will return 1, else return 0 or -1. It is not valid unless you set it before calling setVideoConnected and after calling setVideoDeviceIndex.
OCX	VC6	long getVideoConnected();
	Delphi7	function getVideoConnected: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoGetConnected (HDSP hdsp);
DLL/SO	C++	int VideoGetConnected (void);
Note		Rerurn whether the current video capturing device is open or not. Return 1, it means it is open, return 0, it means it is not open.
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);
Note		Call the function and close the video device which has been open (video capture card or AVI video record), and then open or close the current video capturing device. The parameter bConnected is 1, it indicates opening the device, 0 means closing it .Whether to disply the image in the control window or not, you can call setImageDisplayEnabled and set it. Default is to display the image.
OCX	VC6	long getVideoDeviceIndex();
	Delphi7	function getVideoDeviceIndex: Integer; safecall;

DLL/SO	C	int DSPAPI dspVideoGetDeviceIndex (HDSP hdsp);
DLL/SO	C++	int VideoGetDeviceIndex (void);
Note		Return the serial numbers of the current video capturing device. 0 means the first device, 1 means the second device, the rest can be deduced by analogy

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);
Note		Set the serial numbers of the current video capturing device. 0 means the first device, 1 means the second device, the rest can be deduced by analogy. After calling the function, the system will automatically call the parameters, which's serial numbers. It is not valid unless you set the number before calling setVideoConnected. if plus DSP_JUST_HI_VIDEO_INDEX=1000,will only open high-camera. The max-value of DeviceIndex is limited by the type of your product (4 channels/8channels/12channels).

OCX	VC6	CString getVideoDeviceName();
	BCB6	BSTR __fastcall getVideoDeviceName(void);
	Delphi7	function getVideoDeviceName: WideString; safecall;
Note		Return the string of the name of the current video capturing device which has been open. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

DLL/SO	C	int DSPAPI dspVideoGetDeviceName (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int VideoGetDeviceName (wchar_t* pBuff,int BuffNum);
Note		Return the string of the name of the current video capturing device which has been open.pBuff assigning the goal address,BuffNum assigning the size of the string space.If success,return 1.else 0.

OCX	VC6	long VideoDisplayDlg();
	Delphi7	function VideoDisplayDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoDisplayDlg (HDSP hdsp);
DLL/SO	C++	int VideoDisplayDlg (void);
Note		Display the dialog of the configuration of the current video capturing device.

OCX	VC6	long getVideoDisplayFormat();
	Delphi7	function getVideoDisplayFormat: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoGetDisplayFormat (HDSP hdsp);
DLL/SO	C++	int VideoGetDisplayFormat (void);
Note		Rerurn the standard mode of the current video.

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);
Note		Set the standard mode of the current video.It is not valid unless you set it before calling setVideoConnected and afer calling setVideoDeviceIndex.The video mode list as follow:

mode	Hex	Mode	Hex
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);
Note		Display the dialog of seting the format of the current video device

OCX	VC6	long getVideoSource();
	Delphi7	function getVideoSource: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoGetSource (HDSP hdsp);
DLL/SO	C++	int VideoGetSource (void);
Note		Return the code of the current video input terminals. Successful, it will return 0 – N, else return -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);
Note		Set the code of current video input terminals.

OCX	VC6	long VideoSourceDlg();
	Delphi7	function VideoSourceDlg: Integer; safecall;
DLL/SO	C	int DSPAPI dspVideoSourceDlg (HDSP hdsp);
DLL/SO	C++	int VideoSourceDlg (void);
Note		Display the dialog of seting the channel of the current video capturing device.

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);
Note		Set the other parameters of video device.successful, it will return 1, else it will return 0 or -1.

Type	Param
DSP_VIDEO_CRYOSC=0 crystal frequency	28/35 （MHz）
DSP_VIDEO_BRIGHTNESS=1 Brightness	0-100 (%)
DSP_VIDEO_CONTRAST=2 Contrastion	0-100 (%)
DSP_VIDEO_HUE=3 Hue	0-100 (%)
DSP_VIDEO_SATURATION=4 saturation	0-100 (%)
DSP_VIDEO_SHUTTER=5 set the speed of shutter	(us) When the camera support automatic shutter(Daheng),0 is automatic shutter
DSP_FLASH_SOURCE=6 compulsive flash	1=flash

DSP_FLASH_WIDTH=7 flash pulse width	(us)
DSP_FLASH_DELAY=8 delay from flashing to capturing image	(us)
DSP_VIDEO_RUN=9 pause / run video device	0=pause;1=run
DSP_VIDEO_ADC_BIT=10 ADC figure control	Reference to camera
DSP_VIDEO_DLL_HANDLE=11 Corresponding HINSTANCE of DLL	User can transfer function to camera SDK
DSP_VIDEO_DEVICE_HANDLE=13 Corresponding video equipment of HANDLE	
DSP_VIDEO_ANGLE90=12 Revolve 90 degree	0=no revolve; 90=clockwise; -90=anti-clockwise;
DSP_VIDEO_WHITE_BALANCE_ON_OFF =14 Automatic white balance	0=forbidden; 1=permission
DSP_VIDEO_JPG_QUALITY=15 Transmis JPG image	0=forbidden; 1-100=JPG compress;
DSP_VIDEO_ANTI_FLICKER=16 Anti-flash	0=close; 1=permission
DSP_VIDEO_AUTO_GAIN_MIN=17 min value of the sutomatic aperture	0-100（%）
DSP_VIDEO_AUTO_GAIN_MAX=18 max value of the automatic aperture	0-100（%）
DSP_VIDEO_AUTO_SHUTTER_MIN=19 Min value of the automatic shutter	uS
DSP_VIDEO_AUTO_SHUTTER_MAX=20 Max value of the automatic shutter	uS
DSP_VIDEO_AUTO_SHUTTER=21 the automatic shutter	0=close; 1=permission
DSP_VIDEO_AUTO_BRIGHTNESS=22 the automatic aperture	0=close; 1= permission
DSP_VIDEO_WHITE_BALANCE_R=23 the proportion of white balance R	100-
DSP_VIDEO_WHITE_BALANCE_G=24 the proportion of white balance G	100-
DSP_VIDEO_WHITE_BALANCE_B=25 the proportion of white balance B	100-
DSP_VIDEO_CAMERA_WHITE_BALANC E_ON_OFF=26 automatic white balance	0=close 1=permission
DSP_VIDEO_BAYER_TO_YUV=27 When video card support YV12or I420 convert to YUV,else RGB.	0=close;(default value) 1=permission
DSP_VIDEO_TRIG_CAPTURE=28 Soft trigger capture image	Flash default
DSP_VIDEO_SET_IP_ADDRESS=29 Set Net camera IP address	192.1.1.1 = 0xC0010101
DSP_VIDEO_SET_NET_PORT0=30 Set net camera port#0	0 = default
DSP_VIDEO_SET_NET_PORT1=31 Set net camera port#1	0 = default
DSP_VIDEO_SET_DEV_CHANNEL=32 Set video channel	0 = default

DSP_VIDEO_AUTO_LOAD_CONFIG=36 Set auto-load video config data from register	1=yes(default)，0=no

2012/8/15 addtion new function

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);

Note Set the other parameters of video device.successful, it will return 1, else it will return 0 or -1

Type	Param
DSP_VIDEO_SET_USERNAME_STR=33 Set camera user name	like L"admin" NULL=default
DSP_VIDEO_SET_PASSWORD_STR=34 Set camerta password	like L"12345" NULL=default
DSP_VIDEO_SET_IP_ADDRESS_STR=35 Set camera IP address	like L"192.168.1.10"
DSP_VIDEO_SET_STREAM_NAME_STR=37 Set video stream name	like :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);

Note Read the other parameters of video device. successful, it will return 1, else it will return 0 or -1.

Refer to VideoSetOtherParams.

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

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

Note Read the handle of the video device.

DllName assigning the name of Dll,or transmis NULL.

DeviceHandle is the indicator of receiving handle.

successful, it will return 1, else it will return 0 or -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);

Note Read the indicator in the video package DLL or SO.

DllName assigning the name of Dll,or transmis NULL.

FunctionName is the name of function

Successful,rerurn indicator,else 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);

note Read IO status or data, vedio equipment support of.

2011/5/16 newly added

DSP_VIDEO_READ_CURRENT_FRAME:represent read the current data in the flash frame.

DSP_VIDEO_IO_RS232_0 :represent read the data of RS232 interface.

		DSP_VIDEO_IO_RS232_1 DSP_VIDEO_IO_RS232_2 DSP_VIDEO_IO_RS232_3 DSP_VIDEO_IO_STATUS :represent read the state of I/O interface.
Example 1		BYTE Buf[16]; int ReadSize = VideoReadIO(DSP_VIDEO_IO_RS232_0,Buf,sizeof(Buf));
Example 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);
note		Set or write IO state or data that vedio equipment support of. 2011/5/16 newly edded

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

Application Re-encryption

OCX	VC6	long LicenseDataRead(long FileID, long* pDes, long BufSize);
	Delphi7	function LicenseDataRead(FileID: Integer; var pDes: Integer; BufSize: Integer): Integer; safecall;
Note		Read the authorization information set by the user in the official software softdog. Successful, it will return the number of Byte, else return 0 or -1. FileID gives the ID of the file, ID is from 0 to 4(0 is ready for recognition software developer and users had better not to use it). The parameter pDes points the objective data access buffer area. BufSize gives the length of buffer area, its unit is a byte. Before calling it, you must firstly call LicensePasswordInput and pass the user password. Please refer to the specific use of source codes in the directory of License in the example.
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);
Note		Read the authorization information set by the user in the official software softdog. Successful, it will return the number of Byte, else return 0 or -1. FileID gives the ID of the file, ID is from 0 to 4(0 is ready for recognition software developer and users had better not to use it). Password means the user's password,PasswordLen is the length of the password bite,the total length can't over 64 bite. pDes point to the goal data buffer.BufSize assigning the size of the buffer,the unit is bite.Transfer LicensePasswordInput.Set and through the password. If success,return the numble of the bite,else 0 or -1.
OCX	VC6	long LicenseDataWrite(long FileID, long* pSrc, long Bytes);
	Delphi7	function LicenseDataWrite(FileID: Integer; var pSrc: Integer; Bytes: Integer): Integer; safecall;
Note		Write the user's authorization information into the softdog. Successful, it will return the number of input bytes, else it will return 0 or -1. FileID gives the ID of the file, the ID is from 0 to 4(0 is ready for recognition software developer and users had better not to use it). The parameter pSrc points the objective data access buffer area.Bytes gives the number of byte needed to write into. The value should not be over 64, else you can write 64 bytes at most. Before calling it, you must firstly call LicensePasswordInput and pass the user password. Please refer to the specific use of source codes in the directory of License in the example.
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);
Note		Write the user's authorization information into the softdog. Successful, it will return the number of input bytes, else it will return 0 or -1. FileID gives the ID of the file, the ID is from 0 to 4(0 is ready for recognition software developer and users had better not to use it). Password means the user's password,PasswordLen is the length of the password bite,the total length can't over 64 bite. pSrc point to the goal data buffer.Bytes gives the number of byte needed to write

		into. The value should not be over 64.call LicensePasswordInput. Set and through the password. If success,return the numble of the bite written,else 0 or -1.
OCX	VC6	long LicensePasswordInput(LPCTSTR Password);
	BCB6	long __fastcall LicensePasswordInput(BSTR Password);
	Delphi7	function LicensePasswordInput(const Password: WideString): Integer; safecall;
Note		Input the user password. Password is the password, the total number of bytes should not be over 64.The return is a number which is the one for softdog (The value , the password and the softdog are only corresponding ,using the value and user data operation can greatly impove the intensity of encryption).The default password is 12345678.
DLL/SO	C	int DSPAPI dspLicenseGetUserSerialID (HDSP hdsp,const void* Password,int PasswordLen);
DLL/SO	C++	int LicenseGetUserSerialID (const void* Password,int PasswordLen);
Note		Password means the user's password,PasswordLen is the length of the password bite.The return value is the only one corresponded with softdog.(using the value and user data operation can greatly impove the intensity of encryption).
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;
Note		Modify the user password. FileID gives the ID of the file, the ID is from 0 to 4(0 is ready for recognition software developer and the users had better not to use it). NewPassword appointed new user password. Successful, it will return non-0 value, else return 0. Before calling it, you must firstly call LicensePasswordInput and pass the user password. Please refer to the specific use of source codes in the directory of License in the example.
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);
Note		Modify the user password. FileID gives the ID of the file, the ID is from 0 to 4(0 is ready for recognition software developer and the users had better not to use it). OldPassword is the old password, OldPasswordLen is the length of the old password. NewPassword assigning the new password.If success ,return non-0,else 0.Before calling it,you must firstly call LicensePasswordInput and pass the password. Please refer to the specific use of source codes in the directory of License in the example. If success,return the value bigger than 0.else 0.

Traffic light signal detection

2011/12/09 new function

OCX	VC6	int RedLampDetectSetEnabled(int Params);
DLL/SO	C	int DSPAPI dspRedLampDetectSetEnabled(HDSP hdsp,int Params);
DLL/SO	C++	int RedLampDetectSetEnabled(int Params);
note		Set up Disable or allow traffic light detection.Params=0 as Disable Params=1as allow. The return value is the old state value before set.

OCX	VC6	int RedLampDetectGetNum();
DLL/SO	C	int DSPAPI dspRedLampDetectGetNum(HDSP hdsp);
DLL/SO	C++	int RedLampDetectGetNum();
note		Read the number of the lamp that has been set. Return value from 0 to 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);
note		set the number of the detected light. Num value from 0 to DSP_MAX_RED_LAMP_NUM = 8 success returns 1,failure returns 0.

OCX	VC6	int RedLampDetectSetDisplayResultEnabled(int Params);
DLL/SO	C	int DSPAPI dspRedLampDetectSetDisplayResultEnabled(HDSP hdsp,int Params);
DLL/SO	C++	int RedLampDetectSetDisplayResultEnabled(int Params);
note		set up Disable or allowed to display the results of detection. Params = 0 as Disable; Params = 1 as allow

OCX	VC6	int RedLampDetectSetDisplayRangeEnabled(long Params);
DLL/SO	C	int DSPAPI dspRedLampDetectSetDisplayRangeEnabled(HDSP hdsp,long Params);
DLL/SO	C++	int RedLampDetectSetDisplayRangeEnabled(long Params);
note		set up Disable or allow display detection range. Params = 0 as Disable; Params = 1 as allow.

OCX	VC6	int RedLampDetectSetMinDots(int DotNum);
DLL/SO	C	int DSPAPI dspRedLampDetectSetMinDots(HDSP hdsp,int DotNum);
DLL/SO	C++	int RedLampDetectSetMinDots(int DotNum);
note		set the X/Y direction smallest pixel of the detected light (set sensitivity). The default value is 4.

OCX	VC6	int RedLampDetectGetMinDots();
DLL/SO	C	int DSPAPI dspRedLampDetectGetMinDots(HDSP hdsp);
DLL/SO	C++	int RedLampDetectGetMinDots();
note		read the detected smallest pixel. (set sensitivity). The default value is 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);
note		set up test range. LampIndex is the Numbers of the light (from 0 to 7). Left is the

		value of Left pixel X coordinates; Top is above pixel Y coordinates; Right is the Right pixel X coordinates; The Bottom is bottom pixel Y coordinates. For some lamp have red yellow green color(say different states of one lamp), can be designated as one lamp.
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);
note		Read the detection range of the lamp that has been set. LampIndex means the Numbers of the light(from 0 to 7). Return the read value in pLeft, pTop, pRight, pBottom point to a variable of type long.
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);
note		Read the lamp coordinates that have detected . LampIndex is the light of the Numbers (from 0 to 7). Return the value read In pLeft, pTop, pRight, pBottom point to a variable of type long. Successful returns 1, failure returns 0.
safty		may only be calls in AfterRedLampStateChanged incident.
OCX	VC6	int RedLampDetectCfgLoad();
DLL/SO	C	int DSPAPI dspRedLampDetectCfgLoad(HDSP hdsp);
DLL/SO	C++	int RedLampDetectCfgLoad();
note		Read the saved information from the registry.
OCX	VC6	int DSPAPI dspRedLampDetectCfgSave();
DLL/SO	C	int DSPAPI dspRedLampDetectCfgSave(HDSP hdsp);
DLL/SO	C++	int DSPAPI dspRedLampDetectCfgSave();
note		Save the Settings to the registry.

Vehicle outline trajectory video tracking

2011/12/9 new function

"Vehicle outline trajectory video tracking" applies only to the traffic flow density is not very high places, if traffic is too dense, it may not be effective detection. For some places have much large vehicles, detection effect also not ideal.

OCX	VC6	int MotionSetRunMode(int RunMode);
DLL/SO	C	int DSPAPI dspMotionSetRunMode(HDSP hdsp,int RunMode);
DLL/SO	C++	int MotionSetRunMode(int RunMode);
note		Set up test work model.

RunMode value:	note:
DSP_MOTION_RUN_MODE_NONE = 0	Banned detection
DSP_MOTION_RUN_MODE_CROSSROADS = 1	Crossroads mode
DSP_MOTION_RUN_MODE_LANE = 2	Baynoet mode
DSP_MOTION_RUN_MODE_PARKING = 3	Parking mode

The return value is the old work model before set.

OCX	VC6	int MotionSetDisplayResultEnabled(int Params);
DLL/SO	C	int DSPAPI dspMotionSetDisplayResultEnabled(HDSP hdsp,int Params);
DLL/SO	C++	int MotionSetDisplayResultEnabled(int Params);
note		Set up Disable or allowed to display the results of detection. Params = 0 as Disable; Params = 1 as Enable.

OCX	VC6	int MotionSetDisplayLaneEnabled(int Params);
DLL/SO	C	int DSPAPI dspMotionSetDisplayLaneEnabled(HDSP hdsp,int Params);
DLL/SO	C++	int MotionSetDisplayLaneEnabled(int Params);
note		set up Disable or Enable to display the scope of the channel definition. Params = 0 as Disable; Params = 1 as Enable.

OCX	VC6	int MotionGetLaneNum();
DLL/SO	C	int DSPAPI dspMotionGetLaneNum(HDSP hdsp);
DLL/SO	C++	int MotionGetLaneNum();
note		read the number of the channels that has been set.

OCX	VC6	int MotionSetLaneNum(int Num);
DLL/SO	C	int DSPAPI dspMotionSetLaneNum(HDSP hdsp,int Num);
DLL/SO	C++	int MotionSetLaneNum(int Num);
note		set the channel number that has been detected. Num value from 0 to 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);
note		set the top and bottom.of the detection range.

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);
note		Read the top and bottom of the detection range. return the read value in pTop, pBottom point to a variable of type long. Successful returns 1, failure returns 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);
note		set up the X coordinate value of the channel specofied by LaneIndex (value 0 to 3). TopLeft specify the upper left corner of the X coordinate; TopRight specify the top right corner of the X coordinate; BottomLeft specify the lower left corner of the X coordinate; BottomRight specify the lower right corner of the X. the value of X coordinates can be negative.

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);
note		Read the X coordinates value of the channel specified by LaneIndex(value from 0 to 3). Return the read value in long type variable which specified by pTopLeft, pTopRight, pBottomLeft, pBottomRight .Successful returns 1, failure returns 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);
Note		Read the results of target location that has been detected. MotionTarget designated target Numbers (value 0-7), Return the read value in long type variabl which specified by In pLeft, pTop, pRight, pBottom. Successful returns 1, failure returns 0.
Safty		may only be call in AfterMotionStateChanged incident.

OCX	VC6	int MotionGetIdentity(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetIdentity(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetIdentity(int MotionTarget);
note		Read the target (designated by MotionTarget) identification Numbers (not equal to 0, automatic increasing order). Successful returns non-zero value, failure returns 0.
safty		may only be calls in AfterMotionStateChanged incident.

OCX	VC6	int MotionGetIdentityByPlateTarget(int Target);
DLL/SO	C	int DSPAPI dspMotionGetIdentityByPlateTarget(HDSP hdsp,int Target);
DLL/SO	C++	int MotionGetIdentityByPlateTarget(int Target);
note		Read the movement of the target identification numbers which (designated by the Target) binding with license plate identification target. Successful returns non-zero value, failure returns 0
safty		may only be calls in AfterFilterStateChanged or AfterMotionStateChanged incident.

OCX	VC6	int MotionGetCurrentLaneIndex(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetCurrentLaneIndex(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetCurrentLaneIndex(int MotionTarget);

note		Read the(designated by MotionTarget) current lane Numbers of the target.
safty		may only be calls AfterMotionStateChanged incident..
OCX	VC6	int MotionGetResultType(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetResultType(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetResultType(int MotionTarget);
note		Read the (designated by MotionTarget) current moving types of the target. The return value is: DSP_MOTION_UNKNOW = 0 DSP_MOTION_UP_TO_DOWN = 0x1 DSP_MOTION_DOWN_TO_UP = 0x2
safty		may only be calls in AfterMotionStateChanged incident.
OCX	VC6	int MotionGetSpeedX(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetSpeedX(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetSpeedX(int MotionTarget);
note		Read the current(designated by MotionTarget) moving speed of the object in X direction. The unit is: pixel/minutes.
safty		may only be calls in AfterMotionStateChanged incident.
OCX	VC6	int MotionGetSpeedY(int MotionTarget);
DLL/SO	C	int DSPAPI dspMotionGetSpeedY(HDSP hdsp,int MotionTarget);
DLL/SO	C++	int MotionGetSpeedY(int MotionTarget);
note		Read the current (designated by MotionTarget) moving speed of the object in Y direction. The unit is: pixel/minutes.
safty		may only be calls in AfterMotionStateChanged incident.
OCX	VC6	int MotionCfgLoad();
DLL/SO	C	int DSPAPI dspMotionCfgLoad(HDSP hdsp);
DLL/SO	C++	int MotionCfgLoad();
note		read the settings that have been saved in the registry.
OCX	VC6	int MotionCfgSave();
DLL/SO	C	int DSPAPI dspMotionCfgSave(HDSP hdsp);
DLL/SO	C++	int MotionCfgSave();
note		Save the Setting information to registry

Others

OCX	VC6	long AboutDlg();
	Delphi7	function AboutDlg: Integer; safecall;
Note		Display the dialog of About, as follows:

OCX	VC6	long getMsgInfoEnabled();
	Delphi7	function getMsgInfoEnabled: Integer; safecall;
DLL/SO	C	int DSPAPI dspMsgInfoGetEnabled (HDSP hdsp);
DLL/SO	C++	int MsgInfoGetEnabled (void);
Note		Return whether to display the status bar of the recognition system. 1 means displaying, 0 means not displaying. Default is 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);
Note		Set whether to display the status bar of the recognition system. The parameter bEnabled is1, it means displaying, 0 means the opposite.If you call the function, the location and size of the video window will be flushed immediately. The characteristic can update, enlarge or reduce the information in the window.

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);
Note		Msg appointed by the user is showed in the status bar. 1 means success, o means failure.

OCX	VC6	CString getMsgLogoImageFile();
	BCB6	BSTR __fastcall getMsgLogoImageFile(void);
	Delphi7	function getMsgLogoImageFile: WideString; safecall;
Note		Return the current user's the icon filename. Default is an empty string. Some C++ compilers (such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM,so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

DLL/SO	C	int DSPAPI dspMsgLogoImageGetFile (HDSP hdsp,wchar_t* pBuff,int BuffNum);
DLL/SO	C++	int MsgLogoImageGetFile (wchar_t* pBuff,int BuffNum);
Note		Read the filename of the icon. pBuff point to the goal address,BuffNum assigning the size of string space. Return the size of the copy space.

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);
Note		Set the current user icon file which is appointed by ImageFile (The extension can be .BMP/.JPG)

OCX	VC6	long MsgLogoImageRefresh();
	Delphi7	function MsgLogoImageRefresh: Integer; safecall;
DLL/SO	C	int DSPAPI dspMsgLogoImageRefresh (HDSP hdsp);
DLL/SO	C++	int MsgLogoImageRefresh (void);
Note		Display the current user's icon file.The group of functions can be used to display the picture appointed by the user.

OCX	VC6	CString getPathOfCharLib();
	BCB6	BSTR __fastcall getPathOfCharLib(void);
	Delphi7	function getPathOfCharLib: WideString; safecall;
Note		Return the directory which the recognition control character library CharLib.ini is kept in and it is also the directory of the recognition control PlateDSP.V6.OCX. Some C++ compilers(such as BCB6: Borland C + + Builder) can not release correctly the BSTR string which is returned by the components of OCX/COM, so memory leakage may take place. Please refer to the solution of BSTR String and Memory Leakiness of Page 52.

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);
Note		Set whether to display the locating frame of the red vehicle license plate or not. The parameter bEnabled is 1, it means displaying, 0 means not displying

OCX	VC6	long DoEvent();
	Delphi7	function DoEvent: Integer; safecall;
DLL/SO	C	int DSPAPI dspDoEvent (HDSP hdsp);
DLL/SO	C++	int DoEvent (void);
Note		System message circle.If user need to wait when the main route is circling,can transfer this function to ensure the message circle.

DLL/SO	C	int DSPAPI dspMsgEnabledFPS (HDSP hdsp,int bEnabled);
DLL/SO	C++	int MsgEnabledFPS (int bEnabled);
Note		Permission or forbidden displaying the frame message.If bEnabled is 1,that is display,else no.

Control Event

OCX VC6 void AfterDvrClosed();
Delphi7 procedure AfterDvrClosed;
DLL/SO C++ virtual void AfterDvrClosed(void);
Note When having finished recording the video stream file and closing it, there is an event in order to inform the user of doing something relevant with it.The event is for task safe and can carry out a certain operation about GDI.

OCX VC6 void AfterRecogFinished (long PlateNum);
Delphi7 procedure AfterRecogFinished(PlateNum: Integer);
DLL/SO C++ virtual void AfterRecogFinished(int PlateNum);
Note Having finished capturing or recognising the image, there is an event in order to inform the user of doing something relevant with it. The event is for task safe and can carry out GDI. Reading the recognition should be in the event, otherwise there may be some unforeseen mistakes. The event takes place only in the official edition. For the video stream of PAL, the event will happen by 25 times/s

PlateNum	Def.h	meaning
0	DSP_RECOG_NO_RESULT	have permitted recognising, there is no recognition result
1	DSP_RECOG_HAS_RESULT	have permitted recognising, there is a recognition result
-1	DSP_JUST_CAPTURE	have forbidden recognising, but can capture image
-8000	DSP_MOTION_DETECT	detect there is a moving vehicle
-8001	DSP_FOUND_CHARS	Find some characters. Using the marker , we can improve the rate of capturing image

After calling *setStatEnabled* and the corresponding value is 1, the relevant function, which is built in moving vehicle inspection, will be automatically closed. Namely, there will be not any notice of DSP_MOTION_DETECT or DSP_FOUND_CHARS.

OCX VC6 void AfterFilterStateChanged (long evType);
Delphi7 procedure AfterFilterStateChanged (evType: Integer);
DLL/SO C++ virtual void AfterFilterStateChanged(int Event, unsigned int Target);
Note Having finished capturing or recognising the image. If the status of the counting filter changes, the event will be activated in order to inform the user of doing something relevant with it. The even is for task safe and can carry out some operation about GDI. It is safe to read the recognition in the event. The event takes place only in the official edition.

evType high 16 bits	evTypelow 16 bits	Def.h	meaning
Object number(0-3)	10	DSP_FILTER_OUT_VIEW	the moving vehicle license plate may have been out-of-bound

			s
	11	DSP_FILTER_BEFORE_CLEAR	count the notice in the filter before clearing up
	12	DSP_FILTER_TIMER_OVER	this Statistic is overtime
	13	DSP_FILTER_NEW_PLATE	find a new different license plate
	14	DSP_FILTER_LIKE_PLATE	find a similar license plate
	15	DSP_FILTER_SAME_PLATE	find the same license plate
Value of trigger condition	20	DSP_FILTER_FLASH	the current image is a flash photography

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 When the input image size changes，will emerge this event,Width is the width of the new image,Height is hight.

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 When the color of the lamp changes，will emerge this event,LampColor is the color of lamp,LampIndex is the number.

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 when detected the moving vehicles,will emerge this event.MotionTarget is the number of the moving vehicles,（from 0 to 7），Event is the type of the event.

Event:	note:
DSP_MOTION_OUT_VIEW = 50	Moving out-of-bounds
DSP_MOTION_BEFORE_CLEAR= 51	Will clearing up
DSP_MOTION_TIMER_OVER = 52	overtime
DSP_MOTION_NEW = 53	New object
DSP_MOTION_LIKE = 54	Similar/same object
DSP_MOTION_LANE_CHANGED = 55	Changing lines
DSP_MOTION_CROSS_CENTER_Y = 56	Through the middle line

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 When receiving JPG/MJPG format of streaming vedio,will emerge this event,convenient users to deal with,（such as Network transmission）.pData returns JPG data pointer，ImageSize returns actual size.

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);
note		When users use no-waitting way to save image for JPG format to memory will emerge the event,if the user specified the pointer of buffer memory is effective,and have enough space,the system will auto-copy the data to the user specified memory,and return the user defined memory pointer and the size of JPG image (ImageSize) from pData,if the pointer is invalid or the space is not enough,it will not copy the data,pData returns compressed JPG data pointer,ImageSize returns actual size,convenient users to copy.

DLL/SO example: 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 example: 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 example: 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 example: 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 example: 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 example: 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;
```

BSTR String and Memory Leakiness

BSTR string and memory leakiness often takes place in some application softwares. As we all know, if the type of the return of the OCX/COM is BSTR function, at last, we must initiativly call *SysFreeString* to release it.

For the C++ complier of Borland (for example BCB6:Borland C++ Builder),The prototype of getPlateNumber in PlateDSPV2_TLB.h:

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

Because the type of its return is BSTR, you should use it carefully, otherwise there may be memory leakiness.You can do with it like this:

```
WideString Get_OCX_BSTR (BSTR bstr)
{    //use WideString to automatically release bstr
   WideString tmp;
   tmp.Attach (bstr);
   return tmp;
}
AnsiString PlateNumber = Get_OCX_BSTR (dsp->getPlateNumber ());
 // get the number of a plate
```

Or use C++ universal method as below:

```
void Copy_OCX_BSTR(wchar_t* Buff, int BufNum, BSTR bstr)
{
     wcsncpy ( Buff, bstr, BufNum );
     ::SysFreeString (bstr);    //call Windows API to release
}
wchar_t PlateNumberBuf[20];
Copy_OCX_BSTR (PlateNumberBuf, 20, dsp->getPlateNumber ());
 // get the number of a plate
```

For the C++ complier of Microsoft (for example VC6), the prototype of getPlateNumber in the platedsp2v.cpp :

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

The return is CString and the user can call it directly not to worry about releasing incorrectly.

For example:

CString PlateNumber = m_dsp. getPlateNumber ();

For other programming languages (for example VB, DELPHI and so on), all of them can directly call it not to worry about releasing incorrectly.

The Distripution of the Application software

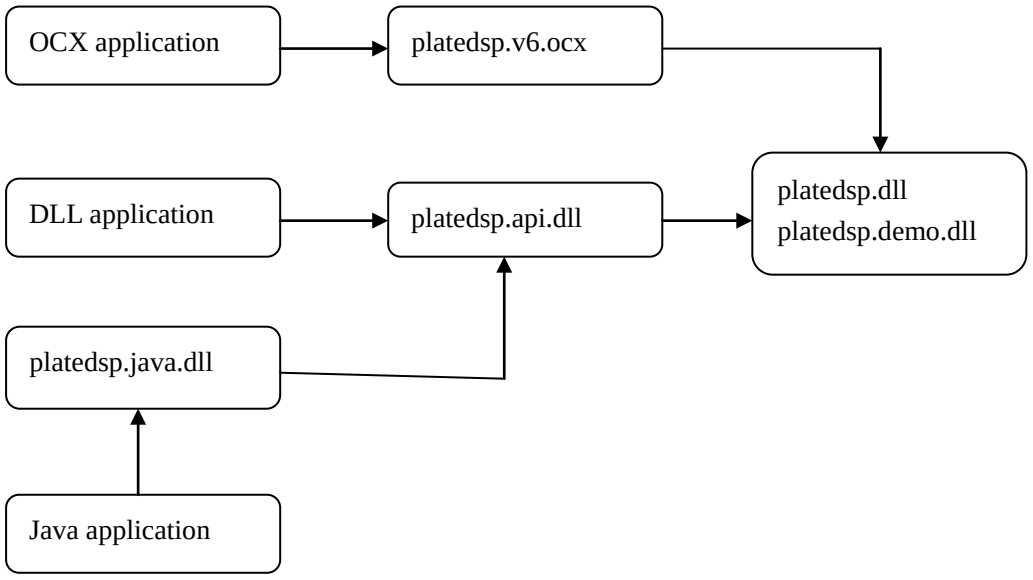
Except for installing softdog driver and DirectX9.0, the easiest installation of the recognition software as follows:

File	Installation directory	note
platedsp.api.dll	system directory	DLL of the V6 API interface.
platedsp.dll	system directory	DLL of the V6 official edition core.
platedsp.demo.dll	system directory	DLL of the V6 demostration edition core when the official edition is called unsuccessfully.it can call DLL automaticly
platedsp.v6.ocx	setup directory	OCX shell of the V2/V3/V3.5/V5/V6, you should enroll by the way of OCX.
GdiPlus.dll	system directory	GDI+library file included by Windows XP system,it can process picture and install in the directory of application or Windows system.Windows XP and above system needn't install it..
platedsp.video.axis.dll	system directory	Optional file. AXIS Camera support library.
avcodec-55.dll avfilter-3.dll avutil-52.dll swresample-0.dll swscale-2.dll platedsp.avi.ffmpeg.dll	system directory	Optional files. ffmpeg support library to support video files, RTSP network video streams.

Linux:please refer the file of readme-on-linux.html in installation package.

Android:please refer the file of readme-on-android.html in installation package.

When the application software developers are developing their own software installation package, they can pack them with recognition software core as above,but it doesn't favor for the single update of the software.



HTML Simple Test

PlateDSP vehicle license plate recognition component can also be applied in web page, please save the content below into V2Test.htm and open it by the IE.

```
<HTML>
<H1>PlateDSP plate recognition software V3 test web page </H1><p>
The web page can test whether PlateDSP plate recognition software V3 is installed in your system or not.
If calling the Active control is forbidden in your IE, the following will display blank.
If recognition control has been installed in your system and can be called in IE, a natural interface of recognition control will appear in the fllowing
And the following button can call the relevant dialog.
<p>
<INPUT id=button1 type=button value=parameter configuration name=button1>
<INPUT id=button2 type=button value=character trainning name=button2>
<INPUT id=button3 type=button value=about 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>>
```

Appendix A:Index

AboutDlg 45
AfterCompressedJpgData 48
AfterDvrClosed 47
AfterFilterStateChanged 47
AfterGetJpgSourceData 48
AfterImageSizeChanged 48
AfterMotionStateChanged 48
AfterRecogFinished 47
AfterRedLampStateChanged 48
AviFrameStep 4
AviIsFinished 4
AviPause 4
AviStart 4
AviStop 5
BufferCopy 17
BufferImageStream 17
BufferSave 18
Close 3
DoEvent 46
dspBufferCopy 17
dspBufferImageStream 17
dspBufferSave 18
dspCreate 3
dspDestroy 3
dspDoEvent 46
dspDvrCompressDlg 6
dspDvrGetBufferFrameNum 6
dspDvrGetCompressor 6
dspDvrGetCompressorDes 7
dspDvrGetCurrentPosition 7
dspDvrGetFrameStep 7
dspDvrImageCopy 9
dspDvrSetBufferFrameNum 6
dspDvrSetCompressor 7
dspDvrSetFrameStep 8
dspDvrSetTitle 10
dspDvrSetUseHalfX 6
dspDvrStart 8
dspDvrStop 8
dspGetCarColor 11
dspGetColorName 11
dspGetPlateBrightness 19
dspGetPlateColor 19
dspGetPlateCount 19
dspGetPlateLocatedRect 20
dspGetPlateNumber 20
dspGetPlateReliability 21
dspGetPlateReliabilityByChar 21
dspGetPlateSpeed 21
dspGetPlateTypeName 22
dspGetRecogCarColorEnabled 11
dspImageGetByField 11
dspImageGetDisplayEnabled 12
dspImageGetIdentity 12
dspImageIsBest 13
dspImagePlateIsBest 13
dspImageSetByField 12
dspImageSetCompressQuality 12
dspImageSetDisplayEnabled 12
dspImageStreamCopy 13
dspImageStreamPlateCopy 14
dspImageStreamPlateSave 15
dspImageStreamSave 16
dspImageStreamSaveEx 16
dspImageStreamSetTitle 17
dspLicenseDataRead 38
dspLicenseDataWrite 38
dspLicenseGetUserSerialID 39
dspLicensePasswordChange 39
dspMotionCfgLoad 44
dspMotionCfgSave 44
dspMotionGetCurrentLaneIndex 43
dspMotionGetCurrentLocatedRect 43
dspMotionGetIdentity 43
dspMotionGetIdentityByPlateTarget 43
dspMotionGetLaneNum 42
dspMotionGetLaneRangeY 42
dspMotionGetResultType 44

dspMotionGetSpeedX 44
dspMotionGetSpeedY 44
dspMotionSetDisplayLaneEnabled 42
dspMotionSetDisplayResultEnabled 42
dspMotionSetLaneNum 42
dspMotionSetLaneRangeX 43
dspMotionSetLaneRangeY 42
dspMotionSetRunMode 42
dspMsgEnabledFPS 46
dspMsgInfoDisplay 45
dspMsgInfoGetEnabled 45
dspMsgInfoSetEnabled 45
dspMsgLogoImageGetFile 46
dspMsgLogoImageRefresh 46
dspMsgLogoImageSetFile 46
dspRecogCfgGetImageRange 24
dspRecogCfgGetMinReliability 23
dspRecogCfgGetPlateRange 24
dspRecogCfgGetProvince 24
dspRecogCfgGetUseTemplate 25
dspRecogCfgSetImageRange 25
dspRecogCfgSetMinReliability 23
dspRecogCfgSetPlateRange 24
dspRecogCfgSetProvince 24
dspRecogGetEnableCount 26
dspRecogParamDlg 26
dspRecogSetEnableCount 26
dspRecogStartWithFile 27
dspRecogStartWithMem 28
dspRecogTrainDlg 28
dspRedLampDetectCfgLoad 41
dspRedLampDetectCfgSave 41
dspRedLampDetectGetLocate 41
dspRedLampDetectGetMinDots 40
dspRedLampDetectGetNum 40
dspRedLampDetectGetRecogRange 41
dspRedLampDetectSetDisplayRangeEnabled 40
dspRedLampDetectSetDisplayResultEnabled 40
dspRedLampDetectSetEnabled 40
dspRedLampDetectSetMinDots 40
dspRedLampDetectSetNum 40
dspRedLampDetectSetRecogRange 40
dspResetImageDisplayWindow 3
dspSetRecogCarColorEnabled 11
dspSetRecogImageFormat4Mem 29
dspSetRedBoxDisplayEnabled 46
dspSetSyncEventCallback 3
dspStatClear 30
dspStatGetColorUsed 30
dspStatGetEnabled 30
dspStatGetMaxTime 30
dspStatSetColorUsed 30
dspStatSetCurrentTarget 31
dspStatSetEnabled 30
dspStatSetMaxTargetNum 31
dspStatSetMaxTime 30
dspVideoDisplayDlg 33
dspVideoFormatDlg 34
dspVideoGetCaptureSize 32
dspVideoGetConnected 32
dspVideoGetDeviceHandle 36
dspVideoGetDeviceIndex 33
dspVideoGetDeviceName 33
dspVideoGetDisplayFormat 33
dspVideoGetDllFunction 36
dspVideoGetOtherParams 36
dspVideoGetSource 34
dspVideoReadIO 36
dspVideoSetCaptureSize 32
dspVideoSetConnected 32
dspVideoSetDeviceIndex 33
dspVideoSetDisplayFormat 33
dspVideoSetOtherParams 34
dspVideoSetOtherParamsStr 36
dspVideoSetSource 34
dspVideoSourceDlg 34
dspVideoWriteIO 37
dspWaitFileSaved 19
dspWaitRecogFinished 18
DvrCompressDlg 6
DvrCompressDlg 6
DvrGetBufferFrameNum 6
DvrGetCompressor 6
DvrGetCompressorDes 7
DvrGetCurrentPosition 7
DvrGetFrameStep 7
DvrImageCopy 9
DvrSetBufferFrameNum 6

DvrSetCompressor 7
DvrSetFrameStep 8
DvrSetTitle 10
DvrSetUseHalfX 6
DvrStart 8
DvrStop 9
DvrStop 8
DvrStopEx 8
getAviCurrentPosition 4
getAviDuration 4
GetCarColor 11
getCarColor 11
GetColorName 11
getColorName 11
getDvrBufferFrameNum 6
getDvrCompressor 6
getDvrCompressorDes 7
getDvrCurrentPosition 7
getDvrFrameStep 7
getImageByField 11
getImageDisplayEnabled 12
getMsgInfoEnabled 45
getMsgLogoImageFile 45
getPathOfCharLib 46
GetPlateBrightness 19
getPlateBrightness 19
GetPlateColor 19
getPlateColor 19
getPlateColor 31
getPlateColorName 19, 31
GetPlateCount 19
getPlateCount 19
getPlateDir 20
GetPlateLocatedRect 20
getPlateLocatedRect 20
GetPlateNumber 20
getPlateNumber 20
getPlateNumber 31
GetPlateReliability 21
getPlateReliability 21
getPlateReliability 31
GetPlateReliabilityByChar 21
getPlateReliabilityByChar 21
getPlateReliabilityByChar 31
GetPlateSpeed 21
getPlateSpeed 21
getPlateTypeId 21, 31
getPlateTypeIdEx 22, 31
GetPlateTypeName 22
getPlateTypeName 22
getPlateTypeName 31
getPlateTypeNameEx 22, 31
getRecogCarColorEnable 11
GetRecogCarColorEnabled 11
getRecogCfgMinReliability 22
getRecogCfgPlateMaxHeight 23
getRecogCfgPlateMaxWidth 23
getRecogCfgPlateMinHeight 23
getRecogCfgPlateMinWidth 23
getRecogCfgProvince 24
getRecogCfgRange 24
getRecogCfgUseTemplate 25
getRecogEnableCount 26
getStatColorUsed 30
getStatEnabled 30
getStatMaxTime 30
getVideoBrightness 32
getVideoCaptureSize 32
getVideoConnected 32
getVideoDeviceIndex 32
getVideoDeviceName 33
getVideoDisplayFormat 33
getVideoSource 34
identifying the direction of movement 20
ImageGetByField 11
ImageGetDisplayEnabled 12
ImageGetIdentity 12
ImageIsBest 13
ImageIsBest 12
ImagePlateIsBest 13
ImageSetByField 12
ImageSetCompressQuality 12
ImageSetDisplayEnabled 12
ImageStreamCopy 13
ImageStreamCopy 13
ImageStreamPlateCopy 14
ImageStreamPlateCopy 14
ImageStreamPlateSave 15

ImageStreamPlateSave 14
ImageStreamPlateSaveNoWait 14
ImageStreamSave 16
ImageStreamSave 15
ImageStreamSaveEx 16
ImageStreamSaveEx 16
ImageStreamSaveExNoWait 16
ImageStreamSaveNoWait 15
ImageStreamSetTitle 17
LicenseDataRead 38
LicenseDataRead 38
LicenseDataWrite 38
LicenseDataWrite 38
LicenseGetUserSerialID 39
LicensePasswordChange 39
LicensePasswordChange 39
LicensePasswordInput 39
MotionCfgLoad 44
MotionCfgSave 44
MotionGetCurrentLaneIndex 43
MotionGetIdentity 43
MotionGetIdentityByPlateTarget 43
MotionGetLaneNum 42
MotionGetLaneRangeX 43
MotionGetLaneRangeY 42
MotionGetResultType 44
MotionGetSpeedX 44
MotionGetSpeedY 44
MotionSetDisplayLaneEnabled 42
MotionSetDisplayResultEnabled 42
MotionSetLaneNum 42
MotionSetLaneRangeX 43
MotionSetLaneRangeY 42
MotionSetRunMode 42
MsgEnabledFPS 46
MsgInfoDisplay 45
MsgInfoGetEnabled 45
MsgInfoSetEnabled 45
MsgLogoImageGetFile 46
MsgLogoImageRefresh 46
MsgLogoImageRefresh 46
MsgLogoImageSetFile 46
Open 3
RecogCfgGetImageRange 24
RecogCfgGetMinReliability 23
RecogCfgGetPlateRange 24
RecogCfgGetProvince 24
RecogCfgGetUseTemplate 25
RecogCfgSetImageRange 25
RecogCfgSetMinReliability 23
RecogCfgSetPlateRange 24
RecogCfgSetProvince 24
RecogGetEnableCount 26
RecogParamDlg 26
RecogParamDlg 26
RecogSetEnableCount 26
RecogStartWithFile 27
RecogStartWithFile 26
RecogStartWithFileWait 27
RecogStartWithMem 28
RecogStartWithMem 27
RecogStartWithMemWait 28
RecogTrainDlg 28
RecogTrainDlg 28
RedLampDetectCfgLoad 41
RedLampDetectGetLocate 41
RedLampDetectGetMinDots 40
RedLampDetectGetNum 40
RedLampDetectGetRecogRange 41
RedLampDetectSetDisplayRangeEnabled 40
RedLampDetectSetDisplayResultEnabled 40
RedLampDetectSetEnabled 40
RedLampDetectSetMinDots 40
RedLampDetectSetNum 40
RedLampDetectSetRecogRange 40
ResetImageDisplayWindow 3
setAviCurrentPosition 4
setDvrBufferFrameNum 6
setDvrCompressor 7
setDvrFrameStep 7
setDvrTitle 10
setImageByField 12
setImageCompressQuality 12
setImageDisplayEnabled 12
setImageStreamTitle 17
setMsgInfoEnabled 45
setMsgLogoImageFile 46
setRecogCarColorEnable 11

SetRecogCarColorEnabled 11
setRecogCfgMinReliability 23
setRecogCfgPlateMaxHeight..... 23
setRecogCfgPlateMaxWidth..... 23
setRecogCfgPlateMinHeight 23
setRecogCfgPlateMinWidth..... 23
setRecogCfgProvince..... 24
setRecogCfgRange..... 25
setRecogCfgUseTemplate..... 25
setRecogEnableCount 26
SetRecogImageFormat4Mem 29
setRecogWithBitmapHeader4Mem 28, 29
SetRedBoxDisplayEnabled..... 46
setRedBoxDisplayEnabled..... 46
setStatColorUsed..... 30
setStatEnabled..... 30
setStatMaxTime 30
setVideoBrightness..... 32
setVideoCaptureSize 32
setVideoConnected..... 32
setVideoDeviceIndex 33
setVideoDisplayFormat..... 33
setVideoOtherParams..... 34
setVideoOtherParamsStr 36
setVideoSource 34
StatClear..... 30
StatClear..... 30
StatGetColorUsed 30
StatGetEnabled..... 30
StatGetMaxTime 30
StatPasteBestRecord 31
StatSetColorUsed..... 30
StatSetCurrentTarget..... 31
StatSetEnabled..... 30
StatSetMaxTargetNum 31
StatSetMaxTime 31
VideoDisplayDlg 33
VideoDisplayDlg 33
VideoFormatDlg 34
VideoFormatDlg 34
VideoGetCaptureSize 32
VideoGetConnected..... 32
VideoGetDeviceHandle 36
VideoGetDeviceIndex 33
VideoGetDeviceName 33
VideoGetDisplayFormat 33
VideoGetDllFunction 36
VideoGetOtherParams 36
VideoGetSource..... 34
VideoReadIO 36, 37
VideoSetCaptureSize 32
VideoSetConnected 32
VideoSetDeviceIndex 33
VideoSetDisplayFormat..... 33
VideoSetOtherParams..... 34
VideoSetOtherParamsStr 36
VideoSetSource 34
VideoSourceDlg 34
VideoSourceDlg 34
VideoWriteIO 37
WaitFileSaved..... 19
WaitRecogFinished..... 18