

文通文档影像分类SDK

文通文档影像分类SDK，是北京文通科技有限公司研制开发的文档影像分类系统，提供强大的识别API开发接口，帮助用户将文档类影像进行自动分类，为数字资源和影像平台的建立提供了方便，快捷，稳定的技术手段。

第一部分：许可协议

1、著作权声明

本产品软件著作权属于中华人民共和国北京文通信息技术有限公司（以下简称文通公司），本系统及使用说明书的著作权属于文通公司，受《中华人民共和国著作权法》、《计算机软件保护条例》以及国际版权公约和其他知识产权条约的保护。除以下授权之条款外，非经著作权人或其全权代理人之书面授权，任何与《协议书》不符的、以任何方式、任何文字所做局部或全部之复制、重制或转载、修改的行为均被法律明确禁止，违者将根据中华人民共和国有关法律承担民事及刑事责任。

2、授权

- 1) 您在使用期间作为合法用户应该持有《信誉卡》。此为获得售后服务、技术支持、升级服务以及进行法律追索必要的资格证明。
- 2) 您可在《使用许可协议》规定的范围内使用本系统。
- 3) “使用”指您利用本系统提供的功能对本系统的数据进行访问。

3、授权限制

- 1) 禁止利用本系统从事任何违反中华人民共和国现行法律、法规的活动。
- 2) 本系统作为一个整体，禁止您在《使用许可协议》许可范围以外使用。

地址：北京市海淀区北四环西路9号银谷大厦16层1609室
电话：(010) 62800286/62800250
网址：www.wintone.com.cn

禁止使用的范围包括但不限于：

- (A) 将本系统的部分或全部嵌入其他计算机软件系统；
- (B) 将本系统的任何内容用于BBS、INTERNET等多使用者的环境；
- 3) 禁止您将本系统的全部或部分用于销售、出租、转让、出借、散布、公开展示或从事其他损害著作权人权利的行为，不得将本系统的复制品在未得到著作权人或其代理人同意的前提下分发给任何第三方。
- 4) 禁止您对本系统进行任何更改、反编译(decompile)、软件还原工程(reverse-engineer the software)、反汇编(disassemble)、或任何更改原始程序设计系统上的锁定与解除锁定。
- 5) 禁止非法盗拷或使用任何《使用许可协议》许可之外的存储媒介进行部分或整体内容的复制与贩售。
- 6) 禁止遮盖、移动或删除本系统著作权所有者的图形或文字说明。
- 7) 如果您未遵守《协议书》的条款，在不作任何通知的情况下，著作权人或其代理人有权终止授权。一旦发生此情况，您必须立即终止使用本系统并销毁所有副本。

4、有限担保

在本系统交付给您60天内，如发现软件载体有任何非人为的损坏，请在此期间内凭《信誉卡》与文通公司联系，可获得修理或更换。由于人为损坏、操作不当或使用错误所引起的故障，不承担任何责任。

5、对用户的赔偿

在法律所允许的范围内，您因使用或不能使用本系统所发生的任何直接的或间接的损失（包括但不限于营业利润的损失、营业中断、商业信息的遗失或其它金钱上的损失），文通公司均不承担赔偿责任。

文通公司为本系统唯一的制作、发行、维护及进行法律追索的代理人。

未经本协议书明确授予的权利均予保留。

文通公司保留对《协议书》的解释权。《协议书》受中国法律保护 and 管辖，

发生法律纠纷时选择北京地区的法院作为管辖法院。

第二部分：系统最小配置

操作系统： Windows 2000、Windows XP、Windows 2003操作系统；

CPU： 586以上；

内存： 256M以上。

第三部分：系统文件概述

1. BIN目录

包括识别核心所需要的各种数据文件，其中内含模板编辑器 TemplateEditor.exe 文件。

2. SDK目录

generalClassifier.dll 该文件是本SDK直接调用的DLL，供用户在程序中调用。

3. Demo目录

包含用VC写的调用实例程序。

4. Doc目录

包含模板编辑器使用手册和本开发指南。

5. Mods

包含三种已经制作好的常用模板，可供参考。

第五部分 系统调用方法

1) InitClassifier

2) FreeClassifier

3) ClassifyImageFile

4) GetTemplateName

函数详细说明如下：

根据模板指导识别总共有4个函数接口，调用过程分为四步进行：加载模板和识别核心；对指定图像分类；获取分类结果；释放模板及核心。可参考附录2的VC调用实例程序。

1) InitClassifier

【声明】 `int InitClassifier(const wchar_t* lpTemplates, const wchar_t* lpFilters)`

【功能】 加载分类用模板和识别核心；

【参数】 `lpTemplates` 加载指定模板集(绝对路径)；

`lpFilters` 加载模板集对应的分类器信息配置文件(本版传NULL)；

【返回值】 0表示成功，其它值为错误代码，详见附录1错误代码。

2) FreeClassifier

【声明】 `void FreeClassifier();`

【功能】 释放识别核心和模板内存空间

【参数】 无；

【返回值】 无。

3) ClassifyImageFile

【声明】 `int ClassifyImageFile(const wchar_t* lpImageName, int templateIndex);`

【功能】 对指定图像影像进行分类；

地址：北京市海淀区北四环西路9号银谷大厦16层1609室
电话：(010) 62800286/62800250
网址：www.wintone.com.cn

【参数】 lpImageName 输入待分类图像名(绝对路径)，支持JPG、TIF、BMP等常用格式；

templateIndex 指定分类所需的模板集(本版传0即可)；

【返回值】 0表示成功返回，其它值为错误代码，见附录1错误代码。

4) GetTemplateName

【声明】 int GetTemplateName(wchar_t* lpTemplateName,
int nTemplateNameLen) ;

【功能】 获得分类结果并通过模板名称返回；

【参数】 lpTemplateName 返回分类得到的模板名称，需要在调用前申请内存(不小于256字节)；

nTemplateNameLen 指定所申请字符串内存空间的大小；

【返回值】 0表示成功，其它值为错误代码，见附录1错误代码。

附录

附录 1 错误代码

#define TW_BANK_NONE	0x0000	//没有错误
#define TW_BANK_LOAD_TEMP_FAILED	0x0001	//装载模板失败
#define TW_BANK_LOAD_IMG_FAILED	0x0002	//读取图像失败
#define TW_BANK_LOAD_ENGINE_FAILED	0x0003	//装载识别引擎失败
#define TW_BANK_DETECT_LINE_FAILED	0x0004	//框线检测失败
#define TW_BANK_UNKNOW	0x0011	//未知错误
#define TW_BANK_UNKNOW_CALL_FUN_FAIL	0x0012	//函数调用失败
#define TW_BANK_UNKNOW_CALL_ORDER	0x0013	//函数调用顺序错误

附录 2 调用实例

```
#include "stdafx.h"
#include "windows.h"

//所需接口声明
int (WINAPI *g_lpInitClassifier)(const wchar_t* lpTemplates, const wchar_t* lpFilters);
void (WINAPI *g_lpFreeClassifier)();
int (WINAPI *g_lpClassifyImageFile)(const wchar_t* lpImageName, int templateIndex);
int (WINAPI *g_lpGetTemplateName)(wchar_t* lpTemplateName, int nTemplateNameLen);
//
int main(int argc, char* argv[])
{
    HINSTANCE hInst = ::LoadLibrary("GeneralClassifier.dll");
    if (hInst == NULL)
        return -1;
    //
    (FARPROC &)g_lpInitClassifier = ::GetProcAddress(hInst, "InitClassifier");
    (FARPROC &)g_lpFreeClassifier = ::GetProcAddress(hInst, "FreeClassifier");
    (FARPROC &)g_lpClassifyImageFile = ::GetProcAddress(hInst, "ClassifyImageFile");
    (FARPROC &)g_lpGetTemplateName = ::GetProcAddress(hInst, "GetTemplateName");

    if (g_lpInitClassifier == NULL || g_lpFreeClassifier == NULL ||
        g_lpClassifyImageFile == NULL || g_lpGetTemplateName == NULL)
        return -2;
```



```
//Step 1: 初始化识别核心
if (0 != g_lpInitClassifier(L"c:\\modlist.mod", NULL))
    return -3;

//Step 2: 影像分类
if (0 != g_lpClassifyImageFile(L"c:\\temp1.jpg", 0))
    return -4;

wchar_t szModName[MAX_PATH] = {0};
if (0 != g_lpGetTemplateName(szModName, MAX_PATH))
    return -5;

//Step 3: 释放内存
g_lpFreeClassifier();

::FreeLibrary(hInst);
return 0;
}
```