

光栅输入设备的 VC++ 编程

高晓兴 王志鹏 (东北大学计算中心 110006)

摘要:本文介绍了光栅输入设备与应用软件之间连接的TWAIN标准的基本原理及结构,简要说明了在图片输入过程中与TWAIN相关软件的工作过程与状态。也根据作者的实践给出了实现光栅输入设备的图象输入及与TWAIN相关软件的调用的VC++程序,对读者理解和实现通过TWAIN编程实现图象输入有一定的参考价值。

关键词:光栅输入设备 TWAIN 扫描仪 数码相机 VC++ 接口

1 概述

光栅输入设备是计算机图象输入的主要输入设备,通过长期的发展通过光栅输入设备向计算机输入图象已经形成了统一的标准,这一标准就是TWAIN, TWAIN标准是一种软硬件的标准,最初的是由一些图形工业的软硬件公司确立的。其目的就是提供一个开放的、多平台的解决方案,以连接光栅输入设备(如扫描仪、数码相机)和应用软件。历经近十年的发展,最新的TWAIN标准是1.9版,是由包括Canon、HP、Intel等11家成员公司共同制定的。1998年5月微软正式宣布Windows98和WinNT 5.0(也就是Windows2000)将包含TWAIN源管理软件。简单的说, TWAIN就是一个软件接口标准。它定义了一种标准的软件协议和API,以支持图片源和应用程序之间的数据传输。支持TWAIN标准的主要设备就是扫描仪和现在很时髦的数字相机,而且TWAIN的最新版本已经可以支持图片数据库、声音流等逻辑设备了。

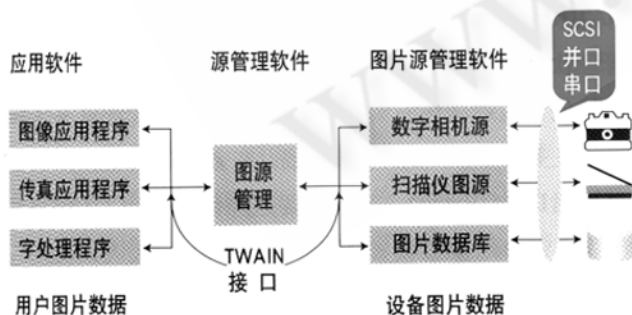


图 1

TWAIN 的实现由 3 大部分组成(如图 1):

(1)应用软件,当然是指采用了TWAIN接口的软件,比如Photoshop、WORD。

(2)源管理软件,这部分软件包含在TWAIN开发工具包中,它负责管理应用软件和图片源之间的交互通WY,通常会与应用软件或设备驱动软件一起免费提供给最终用户。

(3)图片源软件,用于控制图片获取设备,通常这部分软件由设备开发商按照TWAIN标准开发,一般会包含在设备驱动软件中。

对于大多数程序开发人员来说,使用TWAIN标准可以使应用程序更好的兼容光栅输入设备,减轻程序编制的工作量,提高通用性。作为应用程序的开发者,主要是在源管理软件的管理、帮助下对图片源软件进行编程。

TWAIN标准的源管理软件和图片源管理软件都含有自己的用户界面,应用程序可以直接应用。如Photoshop在扫描过图片时,它的扫描仪选择对话框就是包含在源管理软件和图片源管理软件中。

TWAIN定义的操作有140多种,限于篇幅笔者不可能都涉及到,且每种操作的细节设置很多。灵活掌握这些内容足以完成一般的编程任务,具体说明可以参考TWAIN白皮书。TWAIN白皮书可以在TWAIN的网站免费下载,同时还有许多相关资料。在编程前请先下载TWAIN Toolkit软件包。其中有包含了预定义数据结构的头文件,相关例程和可以帮助调试的工具软件。TWAIN标准组织的网站是<http://www.twain.org>

2 编程实现

现在来看看程序编制的过程。简单的说,需要先打开源管理软件,显示图片源选择对话框,根据用户的输入,打开用户选择的图片源,显示图片源的用户界面。然后由用户与图片源管理软件直接交互,来确定所需图片的属

性,比如大小、颜色数等。属性确定后图片源按照用户指定的属性生成图片数据,并传回应用程序。这就是应用 TWAIN 标准的工作过程,而且可以不显示源管理软件和图片源管理软件的用户界面直接在程序中设定图片属性或用自己的程序界面来完成图片属性的采集操作。笔者自己开发的应用程序因为对图片的质量要求很低,所以就没有显示图片源对话框而是直接设定了参数。

如图 2 所示为获得图片整个过程的 7 种状态:

- (1) 为初始状态。
- (2) 为源管理软件调入状态。
- (3) 为源管理软件打开状态。在这个状态显示图片源选择对话框。
- (4) 为打开数据源状态。
- (5) 为数据源打开状态。在这个状态显示数据源的用户对话框。
- (6) 传输准备状态。
- (7) 为数据传输状态。

应用程序通过 TWAIN 接口控制整个会话在 7 种状态间转换。

下面笔者简要介绍利用 VC++ 的编程方法。

TWAIN 的编程接口看起来很简单,因为实际上只需要两个函数,不同的操作是通过向这两个函数传递不同的参数实现的。这两个函数就是 DSM_Entry() 和 DS_Entry()。DSM_Entry() 操作源管理软件,DS_Entry() 操作数据源软件。

·第二个标识结构用于说明函数操作对象的信息,一般是源管理软件或数据源软件。

·DG 即数据组标识,一般是 DG_CONTROL 或 DG_IMAGE,分别用于控制 TWAIN 的会话和图片操作。

·DAT 用于标识传送的数据类型,具体类型预定义可以参考 TWAIN 的白皮书。

·MSG 用于标识应用程序或源管理软件所要求的操作,比如 MSG_GET 或 MSG_SET。

·pData 为指向具体数据的指针。

如从应用程序操作源管理软件,使用函数 DSM_Entry() 并设置函数的 pDest 参数为 NULL;

如从应用程序通过源管理软件操作数据源软件,使用函数 DSM_Entry() 并设置参数 pDest 为要操作的数据源的标识;

如从源管理软件操作数据源,则直接使用 DS_Entry() 函数;

函数所要执行的操作主要是通过 DG, DAT, MSG 三个参数描述的。

下面依照会话的 7 个状态来说明前述两个函数的用法。其实就是如何设置参数以完成相应的操作。因限于篇幅略去了函数返回值的判读,读者在实际编程时请参阅 TWAIN 白皮书。

(1) 从状态 1 到状态 2。目的是调入源管理软件,并取得 DSM_Entry() 的地址。TWAIN 的源管理软件是标准的动态连接库,因此使用标准的动态连接库的调入方法就可以。

(2) 状态 2 至状态 3。打开源管理软件。将 DSM_Entry() 的参数 DG/DAT/MSG 分别设置为 DG_CONTROL / DAT_PARENT / MSG_OPENDSM,调用函数即可打开源管理软件。Porigin 需指向一个 TW_IDENTITY 结构,其中的包含应用程序的信息,函数调用后,源管理软件会填写 TW_IDENTITY 结构的 Id 域。在会话的其他部分当需要调用 DSM_Entry() 时还需要这些信息,所以应该将其值保存。Pdest 参数设置为 NULL。PData 参数应设置为应用程序的主窗口的句柄。笔者在此附上函数的调用例程。

```
TW_UINT16 rc;
rc = (*pDSM_Entry) (&AppID,
    NULL,
    DG_CONTROL,
    DAT_PARENT,
    MSG_OPENDSM,
```

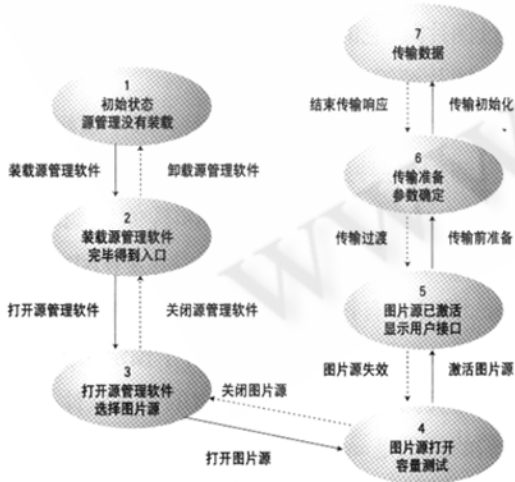


图 2

理解这两个函数参数的意义对于实际编程是非常重要的,参数的具体说明如下:

·首先是一个标识结构,包含调用函数的应用程序的信息。

(TW_MEMREF) &hWnd);

(3) 状态3。选择数据源。参数DG/DAT/MSG分别设置为DG_CONTROL / DAT_IDENTITY / MSG_USERSELECT, Porigin为前述的应用程序描述结构, Pdest设置为NULL, pData也需指向一个TW_IDENTITY结构。函数调用后, 此结构会带回数据源的信息。

(4) 状态3至4。打开数据源。参数DG/DAT/MSG分别设置为DG_CONTROL / DAT_IDENTITY / MSG_OPENS, Porigin为前述的应用程序描述结构, Pdest设置为NULL, pData需指向前一步返回的数据源描述TW_IDENTITY结构。pData参数也可以不这样设置, 这时程序直接打开系统缺省的数据源。其实一般情况下一台计算机只有一个数据源, 那么可以跳过上一步而直接打开缺省的, 也就是唯一的数据源。TWAIN也提供了浏览数据源列表的功能, 可以让程序选择, 具体的编程就要看程序的需要了。

(5) 状态4。获取和设置设备属性。参数DG/DAT/MSG分别设置为DG_CONTROL / DAT_CAPABILITY / MSG_GET, 用于获取设备属性, 设置为DG_CONTROL / DAT_CAPABILITY / MSG_SET, 用于设置设备属性。Porigin为前述的应用程序描述结构, Pdest设置为前述描述数据源的TW_IDENTITY结构, pData指向TW_CAPABILITY结构。当执行获取信息时TW_CAPABILITY结构保存结果, 当执行设置操作时, 该结构应预先初始化为既定值。

(6) 状态4到5。确定图片的属性。参数DG/DAT/MSG分别设置为DG_CONTROL / DAT_USERINTERFACE / MSG_ENABLEDS, Porigin为前述的应用程序描述结构, Pdest设置为前述数据源的描述结构, pData指向TW_USERINTERFACE结构。在这一步会显示数据源的用户界面, 让用户来决定所需的图片的属性。

(7) 状态5到6。所需数据准备完成。这一步不需要发出操作, 而是等待数据源的准备就绪的消息。

(8) 状态6到状态7。发动传送过程。这一步一般要完成两个操作。第一, DG/DAT/MSG分别设置为DG_IMAGE / DAT_IMAGEINFO / MSG_GET, Porigin为前述的应用程序描述结构, Pdest为前述数据源的描述结构, pData指向TW_IMAGEINFO结构。这个操作将取回待传送的图片的信息, 并保存在TW_IMAGEINFO结构, 应用程序可以根据此信息决定后面的操作。第二, DG/DAT/MSG分别设置为DG_IMAGE / DAT_IMAGENA-

TIVEXFER / MSG_GET, Porigin为前述的应用程序描述结构, Pdest为前述数据源的描述结构, Pdata指向一个Windows标准的32位句柄, 数据源将使之指向一个数据源建立的设备无关位图(DIB), 也就是所需的图片。TWAIN共提供了3种传输模式, 可以在第5步, 即状态4选择, 选择了不同的传输模式, 则这一步的操作也要做相应的改动。

(9) 状态7到5。结束传送。DG/DAT/MSG分别设置为DG_CONTROL / DAT_PENDINGXFERS / MSG_ENDXFER, Porigin为前述的应用程序描述结构, Pdest为前述数据源的描述结构, Pdata指向TW_PENDINGXFERS结构, 其中保存了有关传送的信息。根据不同的情况, 会话有可能结束, 或返回至状态5。

(10) 状态5到状态1。结束会话。这一步包含3个操作。第一, 从状态5到状态4, DG/DAT/MSG分别设置为DG_CONTROL / DAT_USERINTERFACE / MSG_DISABLED, Porigin为前述的应用程序描述结构, Pdest为前述数据源的描述结构, Pdata指向TW_USERINTERFACE结构。第二, 从状态4到状态3, DG/DAT/MSG分别设置为DG_CONTROL / DAT_IDENTITY / MSG_CLOSED, Porigin为前述的应用程序描述结构, Pdest为NULL, Pdata指向前述数据源的描述结构, 这一操作将关闭数据源。第三, 从状态3到状态2, DG/DAT/MSG分别设置为DG_CONTROL / DAT_PARENT / MSG_CLOSED, Porigin为前述的应用程序描述结构, Pdest为NULL, Pdata指向数据源的父窗口的句柄, 这一操作将关闭源管理软件。第四, 从状态2到状态1, 释放TWAIN动态连接库。

3 结束语

因为应用程序的需求是各不相同的, 不需要拘泥于上述的步骤, 读者可以自行查阅资料。在笔者浏览TWAIN的网站时, 同时发现了支持TWAIN的控件, 但是需要注册, 否则有使用时间限制。在实践过程中可以直接编程应用也可通过VC作成控件、动态连接库供其他程序调用。■

参考资料

- 1 《WIN32程序员手册》Microsoft Corporation著 1995年, 清华大学出版社出版
- 2 《Visual C++技术内幕》David J. Kruglinski著 1999年, 清华大学出版社出版