

DataSharer 使用说明

林靖宇 linjy02@hotmail.com

DataSharer 通过共享数据块实现进程间数据共享。共享数据块由名称和序号标识，名称加上前缀"Global\\"表示允许跨网络访问。

此模块分为 C++版（DataSharer_x64.dll）和 C#版（DataSharerCs.dll，并需要 DataSharer_x64.dll）。使用方法如下：

- （1）创建 DataSharer 对象。
- （2）用 OpenSharedBlock()创建共享数据块，设置名称、序号和大小。数据块大小应能容纳数据信息和数据内容。用 GetBlockName()和 GetBlockSN()可以查询数据块名称和序号。
- （3）用 WriteData()写入一般数据及其信息，用 ReadData()读取一般数据及其信息。
- （4）用 WriteImage()写入图像数据及其信息，用 ReadImage()读取图像数据及其信息。
- （5）共享数据块创建后不能改变大小。如果需要增大，则应再调用 OpenSharedBlock()并改变数据块名称或序号，原共享数据块不使用后自动释放。

1. C++编程

用 C++编程时注意设置项目属性，包括引用包含文件 DataSharer.hpp 和库文件 DataSharer_x64.lib，设置系统路径指向 DataSharer_x64.dll。

DataSharer 的 C++成员函数的说明见表 1。

表 1 DataSharer 的 C++函数说明

函数原型	说明
int OpenSharedBlock(const char* BlockName, INT32 snBlock, INT32 szBlock)	功能：创建共享数据块，并释放之前创建的共享数据块。 输入：BlockName - 数据块名称，最长 255 字节。snBlock - 数据块序号。szBlock - 数据块字节数，应能容纳数据信息和数据内容。 返回值：成功则返回共享内存大小，否则返回 0。 说明：数据块名称和序号用于标识共享数据块。如果存在同名数据块并且字节数小于 szBlock 则失败。
int WriteData(BYTE* infoSrc, INT32 szInfo, BYTE* dataSrc, INT32 szData, INT32 type);	功能：将数据及其信息写入共享数据块。 输入：infoSrc[szInfo] - 被写入的数据信息。dataSrc[szData] - 被写入的数据内容。type - 数据类型，0 是一般数据，1 是图像 返回值：写入的字节数（包括信息和数据），0 表示写入失败。 说明：先写信息再写数据。infoSrc 为 NULL 或 szInfo 为 0 则无数据信息，只写入数据。dataSrc 为 NULL 或 szData 为 0 则写入失败。

int ReadData(BYTE* dest, INT32 szDest, INT32& byteInfo, INT32& byteData, INT32& type, double& secStamp, INT32& updated);	<p>功能：从共享数据块读取数据及其信息。</p> <p>输入：dest[szDest] - 用于保存数据及其信息的缓冲区。</p> <p>输出：dest - 读出的数据及其信息，信息在前数据在后。byteInfo - 数据信息的字节数。byteData - 数据内容的字节数。数据内容紧跟数据信息。type - 数据类型。0 是一般数据，1 是图像。secStamp - 数据更新的时戳（秒）。updated - 1 表示上次调用 Read 之后数据已经更新。</p> <p>返回值：读出的字节数（包括信息和数据），0 表示读取失败。</p> <p>说明：dest 为 NULL 或 szDest 为 0 则仅读取数据块信息，返回值是可读取的数据字节数。</p>
int WriteImage(BYTE* imgSrc, INT32 szImgSrc, INT32 Width, INT32 Height, INT32 Channel, double secPerFrame);	<p>功能：将图像及其信息写入共享数据块。</p> <p>输入：imgSrc[szSrcImg] - 被写入的图像。Width, Height, Channel - 图像的宽、高和颜色通道。secPerFrame - 传送图像序列时两幅图像之间的时间间隔（秒）。</p> <p>返回值：图像数据字节数，0 表示写入失败。</p> <p>说明：图像数据字节数为 Channel×Width×Height，其中 Channel×Width 应为 4 的倍数。</p>
int ReadImage(BYTE* imgDest, INT32 szImgDest, Infolmage& infolmg, double& secStamp, INT32& updated);	<p>功能：从共享数据块读取图像及其信息。</p> <p>输入：imgDest[szImgDest] - 用于保存图像的缓冲区。</p> <p>输出：imgDest - 读出的图像数据。infolmg - 图像信息。secStamp - 数据更新的时戳（秒）。updated - 1 表示上次调用 Read 之后数据已经更新。</p> <p>返回值：读出的图像数据字节数，0 表示失败。</p> <p>说明：imgDest 为 NULL 或 szImgDest 为 0 则仅读取图像信息，返回值是可读取的字节数。headerData.typeData 应为 1，否则读取失败。</p>
const char* GetBlockName(); int GetBlockSN(); int GetBlockBytes();	<p>功能：读取共享数据块的名称、序号和大小。NULL 和-1 表示未成功调用 Openxxx()。</p>

2. C#函数说明

用 C#编程时注意引用 DataSharerCs.dll，并设置系统路径指向 DataSharer_x64.dll。DataSharerCs.dll 包含 C#类 DataSharer，命名空间为 Vision。

DataSharer 的 C#成员函数的说明见表 2。

表 2 DataSharer 的 C#函数说明

函数原型	说明
int OpenSharedBlock(string BlockName, int snBlock, int szBlock)	<p>功能：创建共享数据块，并释放之前创建的共享数据块。</p> <p>输入：BlockName - 数据块名称，最长 255 字节。snBlock - 数据块序号。szBlock - 数据块字节数，应能容纳数据信息和数据内容。</p> <p>返回值：成功则返回共享内存大小，否则返回 0。</p> <p>说明：数据块名称和序号用于标识共享数据块。如果存在同名数据块并且字节数小于 szBlock 则失败。</p>

int WriteData(IntPtr infoSrc, int szInfo, IntPtr dataSrc, int szData, int type)	<p>功能：将数据及其信息写入共享数据块。</p> <p>输入：infoSrc, szInfo - 被写入的数据信息及其字节数。dataSrc, szData - 被写入的数据内容及其字节数。type - 数据类型，0 是一般数据，1 是图像。</p> <p>返回值：写入的字节数（包括信息和数据），0 表示写入失败。</p> <p>说明：先写信息再写数据。infoSrc 为 IntPtr.Zero 或 szInfo 为 0 则无数据信息，只写入数据。dataSrc 为 IntPtr.Zero 或 szData 为 0 则写入失败。</p>
int ReadData(IntPtr dest, int szDest, out int byteInfo, out int byteData, out int type, out double secStamp, out Int32 updated)	<p>功能：从共享数据块读取数据及其信息。</p> <p>输入：dest, szDest - 用于保存数据及其信息的缓冲区及其大小，可以用 Marshal 的 AllocHGlobal()或 AllocCoTaskMem()方法获取。</p> <p>输出：dest - 读出的数据及其信息，信息在前数据在后。用 Marshal 的方法可以访问。用 IntPtr.Add(dest,byteInfo)可以获得数据指针。byteInfo - 数据信息的字节数。byteData - 数据内容的字节数。数据内容紧跟数据信息。type - 数据类型，0 是一般数据，1 是图像。secStamp - 数据更新的时戳（秒）。updated - 1 表示上次调用 Read 之后数据已经更新。</p> <p>返回值：读出的字节数（包括信息和数据），0 表示读取失败。</p> <p>说明：dest 为 IntPtr.Zero 或 szDest 为 0 则仅读取数据块信息，返回值是可读取的数据字节数。</p>
int WriteImage(Bitmap imgSrc, double secPerFrame)	<p>功能：将图像及其信息写入共享数据块。</p> <p>输入：imgSrc - 被写入的图像。secPerFrame - 传送图像序列时两幅图像之间的时间间隔（秒）。</p> <p>返回值：图像数据字节数，0 表示写入失败。</p> <p>说明：图像数据字节数为 Channel×Width×Height，其中 Channel×Width 应为 4 的倍数。</p>
Bitmap ReadImage(out double secPerFrame, out double secStamp, out Int32 updated)	<p>功能：从共享数据块读取图像及其信息。</p> <p>输出：byteBlock - 共享数据块大小。secPerFrame - 两次写入图像的时间间隔。secStamp - 数据时戳。updated - 1 表示上次调用 Read 之后数据已经更新。</p> <p>返回值：读出的图像，null 表示失败。</p>
String GetBlockName () int GetBlockSN() int GetBlockBytes()	<p>功能：读取共享数据块的名称、序号和大小。NULL 和-1 表示未成功调用 Openxxx()。</p>
static PixelFormat ChannelToPixelFormat(int channel)	<p>功能：将颜色通道数转换为 PixelFormat 类型。</p>