第六章

# 6.1 General functions

### HRESULT load ([in] BSTR fileName, [out, retval] int\* result)

Load a workspace (.fws) or a scan (.fls). If a scan is loaded, its load state will be TRUE and it will be number 0.

Return Values:0, 11, 12, 13, 25

### HRESULT getNumScans ([out, retval] int\* numScans)

Get the number of scans in the current workspace. Scans are numbered starting with 0.

Return Values:Number of scans, -1 in case of error

### HRESULT getScanNumRows ([in] int scan,[out, retval] int\* numRows)

Get the number of rows in the given scan. The scan number can be in the range 0 to numScans-1.

Return Values:Number of rows in this scan, -1 in case of error

### HRESULT getScanNumCols ([in] int scan,[out, retval] int\* numCols)

Get the number of columns in the given scan. The scan number can be in the range 0 to numScans-1.

Return Values:Number of columns in this scan, -1 in case of error

### HRESULT getScanOrientation ([in] int scan,[out] double\* x, [out] double\* y, [out] double\* z,[out] double\* angle, [out, retval] int\* result)

用轴角表示法获取给定扫描方向。其结果是在模型的全球笛卡尔坐标系中。扫描数范围是0到numscans-1。有关扫描点全局位置计算的扫描方向使用情况

### HRESULT getScanPoint ([in] int scan, [in] int row, [in] int col, [out] double\* x, [out] double\* y, [out] double\* z, [out] int\* color, [out, retval] int\* result)

在给定的行/列在给定的扫描得到的点。结果是在扫描器的局部笛卡尔坐标系中。返回的颜色值取决于所选择的反射模式：

0。灰度扫描点：扫描仪的原始反射值。在

一个激光扫描器这是范围在0到2047。

颜色：在范围0到255的颜色的明度。

1。对于灰色扫描点：映射的反射值在0（黑色）范围内

255（白色）。

颜色：在范围0到255的颜色的明度。

2。对于灰色扫描点：映射的反射值在0（黑色）范围内

255（白色）作为RGB灰度值。

颜色：RGB颜色，每通道8位（位0-7：红色，绿色，位位8-15：

16-23：蓝色）。

默认反射模式为1。

扫描次数范围是0到numscans-1。

有关扫描全球位置计算的信息，请参阅第8.4章。

点。

注意！由于COM的开销，这种方法是不是最好的方式读取完整的扫描。使用getxyzscanpoints或getxyzscanpoints2来阅读单个调用中的多个扫描点。例如，您可以读取扫描点。用这两种方法按列。

返回值：0, 26

# 6.2 自动化相关数据的功能

本章描述的功能仅为FARO Focus3D扫描已使用Focus3D自动化接口自动化数据更增强。

关于自动化接口的信息，请参考自动化接口手册。

获得综合自动化数据存储在扫描数据流，FARO开放接口提供访问例程。通过这些访问例程，可以读出每个自动化消息和每个自动触发信号。扫描操作完成后，自动化数据和扫描数据访问就发生了。

自动化数据保存在两个列表中，一个用于通过CAN总线发送的自动化消息，另一个用于通过I/O接口发出的触发信号。每个列表是按时间顺序排序。

获取自动化数据和扫描数据之间的对应关系，每个扫描点和

每个自动化消息或触发器信号都有单独的时间戳。对于时间

与第三方设备同步，绝对自动化时间可以由

带有can消息的自动化应用程序。

在大多数情况下，您将以以下方式使用SDK：

1。有一个事件（触发信号或自动消息），您希望识别对应于该事件的扫描点。在这种情况下，首先必须获得事件的自动化时间，然后使用检索的自动化时间标识扫描点的行和列。

2。在已知行和列上有扫描点，您需要确定其自动化时间，以便找到与外部数据对应的时间。在这种情况下，您可以通过给定的行和列检索给定扫描点的自动化时间。

# 7.1 Connecting to the scanner

### Property BSTR ScannerIP (Read/Write)

The IP address of the scanner扫描仪的ip地址

### HRESULT connect ([out, retval] int\* result)

Tries to connect to the scanner with the given ScannerIP. Changing ScannerIP

doesn’t automatically start the connection, so always first set ScannerIP, then

call connect.

### Property bool Connected (Read only)

It returns, whether FARO LS SDK is connected to a scanner or not

# 7.2 Scanner parameters扫描仪参数

## 7.2.1. Spherical scan mode球面扫描模式

扫描仪默认扫描模式，仪器整体与镜头都会旋转。

## 7.2.2. Helical scan mode螺旋扫描模式

在螺旋扫描模式下，一种可选的操作模式，扫描器在二维仿形模式下工作，在旋转坐标轴上使用旋转镜轴。

保持锁定。扫描仪捕获扫描点数据的连续流，并且通常沿着轨道或道路移动。通过将捕获的扫描点数据与定位信息同步，以螺旋线（螺旋）的形式生成纵向轮廓。

螺旋模有螺旋和螺旋两种模式。这两个子模式在控制和通信信号中彼此不同。

螺旋子模在记录螺旋扫描之前必须先启用。使一个螺旋子模式以及设置扫描参数可以通过扫描仪的用户界面或与FARO LS SDK。

当启用螺旋模式时，水平旋转轴保持锁定，扫描器不会水平转动。当以螺旋模式切换扫描仪时，它将保持这种模式的知识，当扫描仪再次打开时，水平旋转轴仍然处于锁定状态。

在这两种螺旋模式下，不支持彩色扫描的捕获，也不需要设置水平扫描角度。除了标准的扫描参数分辨率和

质量，要记录的扫描线的数目可以设置。当这个号码到达时，扫描仪将停止扫描。也可以选择当扫描线达到一定数量时，扫描数据流被自动分割成多个文件。

当已启用螺旋模式并设置参数时，在开始螺旋扫描之前必须开始螺旋扫描操作。这是可以做到的

通过按下启动按钮扫描仪的用户界面，

按下扫描仪上的启动/停止按钮，

通过执行开始扫描的法鲁LS SDK函数或

发送自动化CAN报文（两，TTL和模态）。

## 7.2.3. 扫描模式特性

### Property ScanMode ScanMode (Read/Write)

扫描方式可以有四个值：

stationarygrey

stationarygrey采取正常的球形灰度扫描。

stationarycolor

stationarycolor用颜色记录球面扫描。这个功能是

只可与FARO Focus3D X330和x130扫描仪。它不会工作

FARO光子，FARO LS扫描仪或Focus3D X30。

helicalgrey

helicalgrey采取扫描使用TTL接口螺旋模式。

helicalcangrey

helicalcangrey采取扫描螺旋模式采用CAN通信。请注意，此功能仅可用于FARO Focus3D

扫描仪将不能与FARO光子或FARO LS扫描仪

，SDK的行为在helicalgrey和helicalcangrey模式相比stationarygrey / stationarycolor模式存在着一定的差异：

numcols必须由用户设置并不会自动调整

当改变其他参数如分辨率时。

horizontalanglemin和horizontalanglemax始终被设置为0°。

HorizontalAngleMax的horizontalanglemin和原值会

丢失，必须重新设置回stationarygrey或stationarycolor模式。

在启动startscan后，镜头会旋转，但扫描点的记录不会启动

直到通过扫描仪的自动化接口发送一个触发信号

### Property int SplitAfterLines (Read/Write)

当达到此参数定义的行或列的数量时，将扫描数据流分割为多个文件。这主要与螺旋扫描记录有关。

### Property double VerticalAngleMax (Read/Write)

读取/写入扫描区域的垂直起始角度。

### Property int Resolution (Read/Write)

读取/写入扫描分辨率。

### Property int MeasurementRate (Read/Write)

测量速率或速度用于记录扫描

可能值：1, 2, 4，8。

1 =每秒122点

2 =每秒244点

4 =每秒488点

8 =每秒976点

### Property int NoiseCompression (Read/Write) （噪声压缩）

运用噪声压缩记录的扫描。这样可以减少扫描和由此影响的分辨率中的噪声，

不是所有的组合与MeasurementRate将通过扫描仪接受。为支持的组合见7.3章。

可能值：1, 2, 4。

1 = no noise compression.

2 = reduces number of points by 4.

4 = reduces number of points by 16.

### Property int NumCols (Read/Write)

读/写的扫描列数。

在扫描模式stationarygrey numcols会自动进行调整，每一次，当分辨率，horizontalanglemin，或horizontalanglemax已经改变时，这个参数也改变，最后，ScanMode HelicalGrey改变分辨率或扫描区域会对numcols无影响。

### Property int NumRows (Read Only)

返回扫描行数

### Property int ScanFileNumber (Read/Write)

读/写扫描最后扫描文件数。每次新扫描前都会自动增加。

### Property BSTR ScanBaseName (Read/Write)

全扫描文件的名称将是“scanbasename”+ scanfilenumber +”。FLS”。这个

扫描将保存在扫描仪驱动程序的扫描文件夹中。

### Property StorageMode StorageMode (Read/Write)

此模式定义保存记录扫描的位置。

可能的值：

SMLocal：扫描存储扫描（对Focus3D可移动SD卡或

激光扫描器的内部硬盘光子）。

SMRemote：在远程计算机上存储扫描（计算机SDK上运行）。

SMAuto：如果连接到扫描仪，扫描将被存储到远程计算机。如果远程计算机没有连接，扫描将在扫描仪上本地存储。

smundefined：存储方式不明确（如未连接扫描仪）

### Property BSTR RemoteScanStoragePath (Read/Write)

这个定义的文件夹的远程计算机上保存扫描（有关如果StorageMode将SMRemote）。

### HRESULT syncParam ([out, retval] int \*result)

所有的扫描仪参数由Faroe LS SDK本地存储。通过调用syncparam他们得到与扫描同步

# 7.4 Scanning

### HRESULT startScan ([out, retval] int\* result)

在球面扫描模式下，这个函数开始扫描。

在螺旋扫描模式，这将启动扫描。镜子将开始转动。

但是扫描数据不会被记录，除非触发信号通过扫描器的螺旋TTL接口发送，如果使用螺旋TTL模式。

在螺can模式，相应的可以留言或记录（见第7.4.1）将开始扫描数据记录。如果是helicalcangrey模式，recordScan和pausescan可用。

一定要连接和运行syncparam来传递参数到扫描仪。

返回值（参见第8.1章）：

0, 1, 3，4

startscan是异步的，这意味着即使它返回值0（OK），

由于某些原因（如SD卡已满），扫描仪可能仍不能开始扫描。在这种情况下，您将通过错误处理获取扫描器的相关错误代码（参见第7.5章获取更多信息）

### HRESULT stopScan ([out, retval] int\* result)

用于停止扫描

### Property int ScanProgress

它返回当前扫描进度百分比。（0到100）

### HRESULT shutdown ([out, retval] int\* result)

这就关闭了扫描仪。在将扫描仪从电源断开之前，一定要关闭扫描仪（通过调用这个函数或按下扫描仪上的按钮）。

返回值0, 3

## 7.4.1. Special Functions for the Helical CAN Mode

### HRESULT recordScan ([out,retval] int\* result)

在螺旋扫描模式下开始扫描数据记录。

返回值：0, 3

### HRESULT pauseScan ([out, retval] int\* resul

暂停扫描数据记录在扫描螺旋can模式。用recordScan从新开始记录数据。

返回值：0, 3

### HRESULTinquireRecordingStatus([out] HelicalRecordingStatus\* status, [out,retval] int\* result)

从螺旋扫描中检索记录状态。可能的值是：

Hrsunknown-SDK不知道状况。

hrspaused -扫描不被记录

hrsrecording -扫描正在被记录

返回值（参见第8.1章）：

0, 2, 3，4

## 7.5 Error Handling

如果一个faro.ls.sdk接口函数返回一个错误代码，你可以从扫描仪获取更详细的异常代码，这将给你更多信息为什么接口函数失败

### Property int NumberExceptions

返回当前的异常数。

### HRESULT clearExceptions ([out, retval] int\* result)

清除扫描仪上的所有异常。这个功能时自动调用。

返回值（参见第8.1章）：0, 1, 3

## 7.6 Connection Point \_IScanCtrlSDKEvents

### HRESULT scanCompleted(void)

这个函数将被调用，当扫描已完成。只有当程序运行Windows消息循环（如在每个Windows应用程序中）时，此回调才有效。控制台应用程序必须运行Windows的消息循环syncparam手动轮询功能。一个例子可以发现在第8.3.1.4。

## 7.7 Scanner Turning

扫描仪可以在没有实际扫描的情况下被转为任意水平角度。该转换将异步处理，所以应用程序必须定期检查所请求的位置是否已由扫描仪到达。

### HRESULT moveToHorizontalAngle ([in] double angle,[out, retval] int\* result)

这个函数使转动开始，输入的角度用度数表示。

### HRESULT requestScannerAngles ([out, retval] int\* result)

这将向扫描仪发送查询请求来检索当前的角度。调查将进行异步处理，当答复回来时，属性receivedscannerangles被设置为true。

返回值（参见第8.1章）：0, 3

### Property bool ReceivedScannerAngles

它返回扫描器角度的最后一次查询的状态。它的值会被赋为true如果最后的查询被回复。

### Property double HorizontalAngle

返回水平角，来自最近且同时被回复的请求，它的值用度数表示。请注意：如果你使用Focus3D FARO的螺旋适配器，你把扫描仪调到89°或91°，以便用适配器的固定销锁定扫描仪。

## 7.8 Remote Access to Scans (only Focus3D)

访问Focus3D扫描（扫描存储在插入sd卡）必须启用远程访问。启用远程访问后，访问网络驱动器的所有常用方法都被允许。

**Property RemoteScanAccessStatus RemoteScanAccess (Read/Write)**

remotescanaccess可以有三个不同的值：

rsaenabled，rsadisabled或rsaunknown

通过获取这个值，您可以确定远程访问的当前状态。

如果SDK不知道当前状态返回将rsaunknown。

将该值设置为rsaenable或rsadisable将启用或禁用远程访问。设置为rsaunknown不会改变现状。

设置状态将等到扫描仪具有正确的状态。

**8.3.1.3. Garbage Collector**

在C #中我们不必在乎销毁对象，这一切将由垃圾回收器进行。不幸的是，我们无法确定确切的时间，一个对象将被删除，但有时候这是必要的，例如，如果我们与FARO LS open加载工作区必须设置可以删除在FARO LS open之外的文件。

因此，我们可以手动调用垃圾收集器来删除所有未使用的对象。它必须用下面的方式使用：

...

iQLibIf libRef = new iQLibIfClass();

...

// delete reference

libRef = null;

// run garbage collector

GC.Collect();

GC.WaitForPendingFinalizers();

8.2 Using FARO LS with Side-By-Side

当从COM对象使用功能时，有两个不同的版本号应该区分哪一个。

首先，有接口的版本号。不同的版本号表示接口以某种方式改变，例如它提供的函数数量，或者在参数的个数中。您编译的应用程序非常依赖于这个版本号，因为在大多数情况下，如果接口发生变化，它将不能正确地使用COM对象。至少需要重新编译，在一些你甚至需要改变你的源代码。第二，实现的版本号。例如，表示bug修复在实现中，它会有不同的实现版本号，但是界面还是一样的。接口的版本号和实现的版本号组合描述FARO LS程序集的版本。一般来说，您的应用程序需要一个特定的接口，但它希望使用最高版本号的实现。

虽然COM的注册机制中支持这种行为。接口，最好的支持是由微软并排提供的。建议只有并排使用FARO LS。并排的使用是有点棘手，所以简短的教程将在本章提供。安装FARO有将文件夹“C创建后：\ Windows \ WinSxS命名“x86\_faro。ls\_1d23f5635ba800ab…”。这个文件夹包含所有需要的DLL FARO LS。WinSxS文件夹还包含一个文件夹命名为清单。在这里你会找到一个程序集清单文件命名为“x86\_faro。ls\_1d23f5635ba800ab的…清单”。它只是一个XML文件。包含一些信息，如版本号的法鲁LS装配。现在不同版本的法鲁LS安装在WinSxS不同的文件夹，所以有没有问题的几个版本并行安装。在许多情况下，新版本与旧版本相同的接口。它完全兼容旧版本，它只包含一些bug修复，在法鲁开放的情况下，就可以打开一个新的扫描现场版。在这种情况下，应用程序应该使用新的法鲁LS版。因此，在含有一些重定向采取新的“政策”WinSxS文件夹版本。此重定向自动工作，没有什么你必须做的。您所要做的就是为可执行文件编写应用程序清单文件。一个应用程序清单是将可执行文件与程序集清单连接在一起的XML文件。WinSxS文件夹。如果你的可执行文件的名称为“MyApplication .exe”，清单的名字必须“MyApplication。exe。清单”，它必须在同一文件夹中的可执行文件。应该是这样的：

<?xml version="1.0" encoding="utf-8"?>

<assembly xmlns='urn:schemas-microsoft-com:asm.v1' manifestVersion='1.0'>

<dependency>

<dependentAssembly>

<assemblyIdentity type='win32' name='FARO.LS' version='1.1.0.0'

processorArchitecture='x86' publicKeyToken="1d23f5635ba800ab"/> </dependentAssembly>

</dependency>

</assembl>

有一个入门版本= '1.1.0.0必须匹配的法鲁LS的版本号

程序集清单。清单的版本号连接到接口的版本号。您将在文件中找到正确的版本号的清单。

例如你法鲁LS文件夹.faro.ls.exe的清单。

8.2.1. C++ with Visual Studio

如果你使用C++在Visual Studio 2005，然后Visual Studio可以为你写清单文件。打开项目的属性对话框并更改链接器清单文件中的选项，生成清单为yes，默认情况下清单文件应该是正确的。然后添加额外的体现依赖FARO LS的依赖（见图片）。插入法鲁LS程序集清单的版本号。然后设置允许隔离为yes。

作为一种替代的附加清单依赖期权可以通过这个pragma命令直接进入源代码集的依赖：

#pragma comment(linker, "\"/manifestdependency:type='win32' name='FARO.LS'

version='1.1.0.0' processorArchitecture='x86'

publicKeyToken='1d23f5635ba800ab'\"")

（请编辑版本字符串），您将在文件中找到正确的版本号。

example.faro.ls.exe.manifest你法鲁LS文件夹）

清单文件可以嵌入到可执行文件的资源中，尽管它与外部清单文件一样有效。但它的方便，都在执行。清单将被嵌入的mt.exe工具和一些参数，但在视觉这一切都可以通过点击几次：打开清单工具-输入和输出属性，并将嵌入清单设置为“是”（参见图像）。其他属性应该是默认改正。设置完所有的这些特性，你应该清洁解决方案和重建，否则VisualStudio无法正确更新清单文件。可以肯定的检查生成的清单文件。

8.2.5. Using FARO LS SDK and FARO Open within C++ Plug-in DLLs

FARO LS SDK和FARO打开对象使用免注册COM创建。对于COM层解决对象ID的DLL提供的执行顺序

需要显示信息。当使用FARO组件在执行清单的依赖在C++和Visual Studio的描述就足够了。然而这种方法要求FARO组件被安装在目标机器上。

由于静态依赖性，组件丢失时应用程序无法加载。

当开发一个插件应用程序的插件DLL应取决于法鲁成分而不是可执行的应用程序。在这种情况下，一些额外的工作

待做。在引擎盖下，Windows进程显示并将信息转换为激活上下文。在加载插件DLL的激活上下文包括法鲁LS的关系创造了积极的电话到DllMain的初始化时间。这意味着，当FARO组件创建在你的插件DLL的DllMain有什么特殊的事要做。

但是，当在DLL导出的函数中创建对象时，需要激活适当的激活上下文，以便加载程序能够正确地解析COM对象创建。要做到这一点我们建议存储激活上下文在DllMain由出口后激活功能。例在第8.3.2.3说明这种方法。其他的替代方案概述在MSDN。我们鼓励感兴趣的人。

读取器在“Win32和com”中查找“孤立应用程序和并排组件”

开发/管理和管理/应用安装和服务。

注意：当测试你编译的插件进行适当的激活上下文管理必须确保既不是FARO.LS.SDK.dll也不是iqopen.dll是在您的系统上注册。否则，您将不会注意到来自错误激活上下文的任何负载故障。