Spatial Analyzer SDK

The Spatial Analyzer SDK provides a means by which to write custom applications that utilize Measurement Plan functionality within Visual C++ and VB.NET. Once the SDK engine has been added to the development environment, Measurement Plan Steps can be executed within the programming language.

Adding the SDK to the Visual Studio Application

C++:

1. Create a new C++/MFC project within Visual Studio

New Project		2 🛛
Project types: Visual C++ 		Iemplates: III IIII
		MFC ActiveX Control MFC Application My Templates
Smart Devi Win32 Other Languag	ce 🛛	Search Online Templates
 Visual Basic Windows Smart Device Database Starter Kits Robotics 		
A project for creati	ng an application th	at uses the Microsoft Foundation Class Library
<u>N</u> ame:	ExampleApp	
Location:	C:\Temp	Browse
Solution Name: ExampleApp		Create directory for solution
	Add to Source Control	
		OK Cancel

2. For illustration purposes, let's assume a dialog application is created.

MFC Application Wizard - ExampleApp 🛛 🔹 🔀						
Application Type						
Overview Application Type Compound Document Support Document Template Strings Database Support User Interface Features Advanced Features Generated Classes	Application type: Single document Multiple documents Dialog based Use HTML dialog Multiple top-level documents Document/View architecture support Resource language: English (United States)	Project style: Windows Explorer MFC standard Use of MFC: Use MFC in a shared DLL Use MFC in a static library				
	< Previous Ne	ext > Finish Cancel				

Note that "Dialog based" has been indicated in the above dialog.

MFC Application Wizard - ExampleApp				
Advanced	d Features			
Overview Application Type Compound Document Support Document Template Strings Database Support User Interface Features Advanced Features Generated Classes	Advanced features: Number of files on recent file list: Context-sensitive Help Image: Context-sensitive Help WinHelp Eormat (unsupported) Image: HTML Help format Printing and print preview Printing and print preview Automation ActiveX controls MAPI (Messaging API) Image: Mindows sockets Active Accessibility Common Control Manifest			
	< Previous Next > Finish Cancel			

Make sure that "Automation" has been checked.

3. Now add the SDK to the application. From the Solution Explorer, right mouse click on the ExampleApp within the tree and select Add >> Class.



4. Select MFC Class From TypeLib, then press the Add button.

Add Class - ExampleApp		? 🛛	
Categories:	Templates:		
Visual C++ CLR ATL MFC C++ C++ Spart Davise	Visual Studio installed templates ^{MEC} Class ^{MEC} Class From ActiveX Control ^{MEC} Class From TypeLib ^{MEC} CDBC Consumer ^{MEC} Class From TypeLib ^{MEC} CDBC Consumer		
	Search Online Templates		
Adds a Microsoft Foundation Class Libra	y class based on a type library		
Name;			
Location:		Browse	
		Add Cancel	

5. Select the "SpatialAnalyzerSDK" from the "Available type libraries" drop-down. Then press the ">>" button to add ISpatialAnalyzerSDK to the Generated Classes section.

Add Class From Typelib Wizard - ExampleApp					
MAA FC	Welcome to the Add Class From Typelib Wizard				
Add class from:	Available type libraries:				
	SpatialAnalyzerSDK<1.0>	*			
Location:					
C: \Program Files (x86))\New River Kinematics\SpatialAnalyzer 2009.05.12\SpatialAnalyzerSDK.exe				
ISpatialAnalyzerSDK	CSpatialAnalyzerSDK				
Cla <u>s</u> s:	File:				
CSpatialAnalyzerSDK	CSpatialAnalyzerSDK.h				
	Finish Can	cel			

- 6. Select Finish.
- 7. "CSpatialAnalyzerSDK.h" will automatically be added to your project and contains all the SDK interfaces available.

VB.NET

1. Create a new Visual Basic Project within Studio:

New Project					? 🔀
Project types: Visual C++ Other Languages Visual Basic Windows Bart Device		Implates: Visual Studio installed templates Windows Application Console Application Windows Control Library Web Control Library Windows Service		trol Library ice	
Otabase Starter Kits Robotics Web Visual C# Visual J# Other Project Types		My Templates	J Crystal Repor	ts Application	
A project for creating an application with		n a Windows user interface			
<u>N</u> ame:	ExampleVBApp				
Location:	C:\Temp			~	Browse
Solution Name: ExampleVBApp			Create directory for sol	ution	
			Add to Source Control		
				ОК	Cancel

2. From the Project menu, select "Add Reference"

	Add Windows Form
•	Add User Control
1	Add Component
1	Add Module
23	Add Class
8	Add New Item Ctrl+Shift+A
:::	Add Existing Item Shift+Alt+A
	Exclude From Project
P	Show All Files
	Visio UML 🔹 🕨
	Add Reference
	Add Web Reference
	Set as StartUp Project
c	ExampleVBApp Properties Alt+F7

3. From the Add Reference dialog, select the COM tab followed by the "SpatialAnalyzerSDK" component name. Press OK.

Add Reference			
	VET COM Projects Browse	Recent	Path 🔊
	Component Name -		
	ShagitShellExt 1.0 Type Library	1.0	C:\Program Files (x86)\Tech5i
	SoftwareUpdate	1.0	C:\Program Files (x86)(Apple
	Sonic HTTP Client 2.0 Type Li	1.0	C:\Program Files (x86)(Commi
	SonicLicenseManager 2.0 Typ	2.0	C:\Program Files (x86)(Commi
	SpatialAnalyzerSDK	1.0	C: (Program Files (xob) (New R
	Sql Server Projects Extensibili	1.0	C:\Program Files (x86)(Commi
	STCG4Control 1.0 Type Library	1.0	
	STClient 1.0 Type Library	1.0	C:(WINDOWS(SyswOw64)so
	Stillssvr 1.0 Type Library	1.0	C:\Program Files (xob)(Commi
	StreetsOkAddin 1.0 Type Lib	1.0	C:\Program Files (x86)(Micros
	StreetsOkShim 1.0 Type Library	1.0	C:\Program Files (X86)(Imicros
	STSUPICITUTIVE LIDrary	1.0	C:\PROGRA~2\MICROS~4\OI
	SwBroker 1.0 Type Library	1.0	C:\WINDOWS\SysWOW64\Ac
	System	2.0	c:\WINDOWS\Microsoft.NET\F
	System Monitor Control	3.6	C:)WINDOWS)system32)syst
			OK Cancel

4. Double left mouse click on the windows form from within the designer. Studio should automatically display the following method:

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

5. Modify the class code to appear as follows:

Public Class Form1
REM Declare the object as the SA SDK Interface
Private NRKSdk As SpatialAnalyzerSDK.SpatialAnalyzerSDK
REM Declare an enumerated type that defines the various return codes from the SA SDK Interface
Enum MPStatus
SdkError = -1
Undone = 0
InProgress = 1
DoneSuccess = 2
DoneFatalError = 3
DoneMinorError = 4
CurrentTask = 5
End Enum
REM Allocate the SDK Interface object
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
NRKSdk = New SpatialAnalyzerSDK.SpatialAnalyzerSDK
End Sub
End Class

6. Compile and run the application.

SDK Overview

The SDK consists of set of function calls that allow the specification and execution of MP Steps. The client must first instruct the SDK engine to connect to Spatial Analyzer before performing any MP Step operations. The following example illustrates initiating the connection with Spatial Analyzer (in Visual Basic):

```
If (NRKSdk.Connect("localhost") = False) Then
MsgBox("Failed to connect with automation server!")
```

End If

To execute the MP Step that will "Construct a Point in Working Coordinates" the syntax would appear as follows (shown in Visual Basic):

NRKSdk.SetStep("Construct a Point in Working Coordinates") NRKSdk.SetVectorArg("Working Coordinates", 10.1234, 20.2345, 30.5678) NRKSdk.SetPointNameArg("Point Name", "", "TestGrp", "TestPt") Dim bSendStatus As Boolean bSendStatus = NRKSdk.ExecuteStep()

The string names specified for the step name and its corresponding arguments need to **exactly** match the SA MP Text strings in English.

The return status of executing any MP can be checked as follows (shown in Visual Basic):

If (bSendStatus) Then

Dim rCode As Long

NRKSdk.GetMPStepResult(rCode)

If (rCode <> MPStatus.DoneSuccess) Then

REM Do some error handling here

If (rCode = MPStatus.SdkError) Then

Dim msgArray As Object

REM Display text returned from execution attempt to user...

If (NRKSdk.GetMPStepMessages(msgArray)) Then

Dim idx

Dim msg As String

msg = ""

For idx = LBound(msgArray) To UBound(msgArray)

```
msg = msg + msgArray(idx) + Chr(10) + Chr(13)
```

Next

MsgBox(msg)

End If

End If

End If

End If

If an MP contains a "Result Only" argument, then the result argument can be retrieved via calling the appropriate Get function call. For example, if the MP step successfully executed and a vector result argument needed to be extracted, the following code could be used (shown in Visual Basic):

REM This section illustrates getting return values from the successfully sent MP Step



End If

SDK Engine

Whenever a client application is executing, the Spatial Analyzer SDK engine will automatically start and appear minimized on the TaskBar. Below shows the engine in its maximized state:



Spatial Analyzer

Spatial Analyzer also supports the generation of sample code through the MP Editor. Setup an MP Step with the necessary arguments and press the code generation button to display sample code in either Visual C++ or Visual Basic.

MP: SDKTester.mp					×
Eile <u>S</u> tep <u>P</u> assword					
Step List			Calast	ad Stan	~
🖹 🔳 + 🛆 🗸 🗉 🥥	Selected Step				
	Comment:				
[1] Construct a Point in Working Coordinates					
		Argun	nents		
		Description	Method	Value	
	Point Name	istes	Dialog Entry Enter Values	B::TestGrp::TestPt0	3
Code Generator	Thermany coordinate	aloo			
Button					
					-
< >>	1				×
]]2]				-
SDK Code Viewer					
Language					
	O Visual C+	+ (🔿 Visual Basic		
	• + 16441 0 ·				
NrkSdk.SetStepf"Construct a	Point in Work	ing Coordinates'');			
NrkSdk.SetPointNa	meArg("Point	Name", "B", "TestGi	rp'', ''TestPt0'');		
NrkSdk.SetVectorA	rg("Working C	Coordinates", 15.000	000, 30.000000, 45.00000	00);	
NrkSdk.ExecuteStep()					
NickSidk, SatStap("Construct S	aboro'''):				
Niksuk.setstept Constructis Niksdk SetObjectN	priere j, ame∆ra("Sobi	ere Name'' "MuSphe	ere'').		
NrkSdk.SetVectorA	ral"Sphere Ce	enter fin working coor	rdinates)''. 45.000000. 55.	000000, 65.000C	
NrkSdk.SetDouble4	vrg("Sphere R	adius'', 15.000000);	, ,,		
NrkSdk.ExecuteStep()					
				~	
		ОК			

Generated code will automatically be placed on the Clipboard for pasting into a client application that is being developed.

Examples

Sample client applications are available for Visual C++ and Visual Basic. The Visual C++ sample contains a helper class called **SDKHelper**. The SDKHelper class provides convenience functions when accessing Arrays from a Visual C++ client application. When using the new Spatial Analyzer code generation feature, the code generator will utilize this helper function in any code generated to Set/Get array data. Please refer to the Visual C++ sample client application source code for more details.