There is a folder in the SA install directory called SAInstrumentSDK.  In that directory, there is a file called SDKInstDefs.xml.

Its default contents serve as a guide to fill in the blanks, and to instruct SA not to show the new instrument anywhere.  Those default contents are:

<sdkInstruments>

 <PhotoGramm>

 <exeName>

 YourExe.exe

 </exeName>

 <uiName>

 yourNameToShowInUI

 </uiName>

 <logoName>

 yourLogo.png

 </logoName>

 <graphicName>

 yourInstrumentGraphic.png

 </graphicName>

 </PhotoGramm>

 <PhotoGramm2>

 <exeName>

 YourExe.exe

 </exeName>

 <uiName>

 yourNameToShowInUI

 </uiName>

 <logoName>

 yourLogo.png

 </logoName>

 <graphicName>

 yourInstrumentGraphic.png

 </graphicName>

 </PhotoGramm2>

</sdkInstruments>

In the above state of the xml, there is no mention of your SAInstrumentSDK instrument anywhere in SA.  However, to make sdk interfaces show up in SA,

1. In the SAInstSDKUserSample project, find the call to sa.Logon(int, std::string). The first argument to that method is the SA instrument index. 212 is the index for the <PhotoGramm> interface, and 213 is for <PhotoGramm2>. We’ll use the first on in this example, so set the first argument to 212. The second argument is to ensure a unique settings file name, and does not matter currently.
2. Build the sdk sample app and copy the resulting SAInstSDKUserSample.exe, and the SAInstrumentSDK64uvc19.dll, to the SA install directory's **\x64** subdirectory.
3. Make a copy of your SDKInstDefs.xml outside the SA install directory, so windows will not complain about an attempt to edit it. Change the exeName, uiName, logoName, and graphicName in the xml file as follows (these may already be as shown in your deployment):

<sdkInstruments>

 <PhotoGramm>

 <exeName>

 SAInstSDKUserSample.exe

 </exeName>

 <uiName>

 InstSDK Test

 </uiName>

 <logoName>

 gsi logo.png

 </logoName>

 <graphicName>

 Nikon Z9.png

 </graphicName>

 </PhotoGramm>

 <PhotoGramm2>

 <exeName>

 YourExe.exe

 </exeName>

 <uiName>

 yourNameToShowInUI

 </uiName>

 <logoName>

 yourLogo.png

 </logoName>

 <graphicName>

 yourInstrumentGraphic.png

 </graphicName>

 </PhotoGramm2>

</sdkInstruments>

Copy this over the SDKInstDefs.xml in your SA install directory’s “SAInstrumentSDK” subfolder. And from the “XML for SAInstrumentSDK folder” in the sdk, also copy the “Nikon Z9.png” and “gsi logo.png” to the same “SAInstrumentSDK” subfolder. You’ll need to provide administrator privilege for all that.

Then, run SA and you’ll see the following in the add instrument dialog:



and hit the [Add Instrument] button above to add your instrument to SA. You can now run it from SA as well:



You can also just click the “blue running man” to run the interface directly.

Just try out the buttons in the sample app, and go to the button handler methods in the sample app to see how the code works for different data types, etc.
Also note the setup of the Docking button in the sample. This is described in the Readme (Documentation folder in SDK), and code is heavily commented in the sample.

Note that the screen shots shown are all without the newer SA Ribbon Bar interface. They are shown in the original SA interface mode.