

SVFunctions Guide

Document Information

Software Version:	4.0.1.3
Creation Date:	17 September, 2020
Last Edit Date:	07 December, 2020
Version:	0.4

Table of Contents

1.	Scope	3
2.	Summary	3
3.	Using SVFunctions	3
4.	Intellisense	4
5.	SVFunctions - Examples	
5.1	l. Output	5
5.2	2 ShutdownWindows	6
5.3	3. Send	8
_5. 4	4. Close	9
5.5	. Open	
5.6	. ProjectPath	10
5.7	. ToLower	

1. Scope

This document describes the ADISRA SmartView System Function Libraries, explain how they work and the steps to follow to use them.

2. Summary

System Function Libraries (SVFunctions) are functions that the ADISRA SmartView makes available to the user and they are grouped into different libraries. All these functions can be used in any script (i.e., service document, trigger, screens, etc.).

3. Using SVFunctions

To access the SVFunctions available, expand the node "System Function Libraries" on the navigation tree and open one of the documents.



Each of the documents under "System Functions Library" contains the functions signature. The user will not be able to see the function's implementation, but they will learn how to use them. The function signature shows the parameters expected for each of the functions and the function's return type. If further function information is needed, please expand the regions (+) and there will be a summary for each of the functions and a summary for each of the parameters. Additionally, the Help file can be checked for further information and examples.

The next chapter will provide some SVFunctions usage examples:

4. Intellisense

The Intellisense is an ADISRA SmartView feature that autocompletes the script codes. It can be used for adding tags into different fields/scripts (when @ sign is written into a script, it shows all the application tags available) and it can also be used to add the System Function Libraries.

All of the System Function Libraries starts with an SV prefix, so as soon as the user types "SV" in the script box, all of the libraries will be listed and can be selected.



After selecting the library (i.e. SVApplication), the user will have access to the functions after typing a period (.).





5.1. Output

How to use a function to log a message in the "Engineering Log" will be described in this example.

SVApplications.Output()

Configure the "mouse up" script in a button object as shown in the image below:

Graphics1 X
1 SVApplications.Output("Configured Message");
Line: 1
Mouse Up 📄 Mouse Down Mouse While Mouse Right Up Mouse Right Down Mouse Double Click
Mode View 💽 🚬 🗖 Graphic Items: Button1(GButton) 🗸

Every time the button is pressed during runtime it will log the message in the "Engineering Log" as shown in the image below:

ngineering Log						▼ ₽ ×
Description		Location	Line	Object	TimeStamp	
Configured Message		AppServer.dll			09/21/2020 01:04:50.122 PM	
Engineering Log	Runtime Log	Search Output	Build Outpu	t		
	ngineering Log Description Configured Messa Engineering Log	ngineering Log Description Configured Message Engineering Log Runtime Log	ngineering Log Description Location Configured Message AppServer.dll Engineering Log Runtime Log Search Output	ngineering Log Description Location Line Configured Message AppServer.dll Engineering Log Runtime Log Search Output Build Outpu	ngineering Log Description Location Line Object Configured Message AppServer.dll Engineering Log Runtime Log Search Output Build Output	ngineering Log Description Location Line Object TimeStamp Configured Message AppServer.dll 09/21/2020 01:04:50.122 PM Engineering Log Runtime Log Search Output Build Output

It is also possible to pass a tag as parameter to the SVApplication.Output() function and the result will be a new output log showing the value of the tag.

Graphics1	×				
1 SVApplicatio	ns.Output(@str	Message);			
<					>
Line: 1					
Mouse Up 🗾	Mouse Down	Mouse While	Mouse Right Up	Mouse Right Down	Mouse Double Click
Mode View 📘	Craphic 🗖	Items: Buttor	n1(GButton) 👻		

5.2. ShutdownWindows

How to use a function to shut down or restart Windows will be explained in this example.

SVApplications.ShutdownWindows()

Configure the "mouse up" script in a button object as shown in the image below:

2	Graphics1 X	Ŧ
	1 SVApplications.ShutdownWindows(true, 10, true);	1
	Line: 1	
	Mouse Up 📄 Mouse Down Mouse While Mouse Right Up Mouse Right Down Mouse Double Click	Γ
	Mode View 💽 🔁 Graphic Items: Button1(GButton) 🗸	

In this case, the first parameter "true" indicates that the machine will shut down, the second parameter "10" indicates that the shutdown will occur 10 seconds after the button is pressed, and the last parameter "true" indicates that the shutdown will be forced.

See another example below:

Graphics1 ×
1 SVApplications.ShutdownWindows(false, 0, false);
Line: 1
Mouse Up 🗐 Mouse Down Mouse While Mouse Right Up Mouse Right Down Mouse Double Click
Mode View 💽 🚬 — Graphic Items: Button1(GButton) 👻

In this case, the first parameter "false" indicates that the machine will restart, the second parameter 0 indicates that the restart will occur instantly after the button is pressed, and the last parameter "false" indicates that the restart will not be forced.

5.3. Send

How to use a function to trigger an event will be described in this example.

SVEvent.Send()

Configure the "mouse up" script in a button object as shown in the image below:

Graphics1 X	₹
1 SVEvent.Send("Message1"); 2 SVEvent.Send("Group2", "Message2"); 3 SVEvent.Send("Group3", 3, "Message3");	~ ~
	>
Line: 3	
Mouse Up 📄 Mouse Down Mouse While Mouse Right Up Mouse Right Down	Mouse Double Click
Mode View 💽 🚬 — Graphic Items: Button1(GButton) 👻	

The example below shows all the ways to trigger an event. It is important to understand that the first line of the script triggers the event, as seen in line 1 below with the message "Message1", the default priority, and no group. The second line of the script triggers event 2, with the message "Message2", the default priority, and "Group2". The third line of the script triggers event 3, with the message "Message3", priority "3" and "Group3".

	Group	Priority	Message	Event Time
3	Group3	3	Message3	09/21/2020 02:05:10 PM
2	Group2	0	Message2	09/21/2020 02:05:10 PM
1		0	Message1	09/21/2020 02:05:10 PM

5.4. Close

How to use a function to close a graphic will be explained in this example.

SVGraphics.Close()

Configure the "mouse up" script in a button object as shown in the image below:

Graphics1 ×
1 SVGraphics.Close();
Line: 1
Mouse Up 📄 Mouse Down Mouse While Mouse Right Up Mouse Right Down Mouse Double Click
Mode View 💽 🤁 🗖 Graphic Items: Button1(GButton) 👻

In this example, the button will close current graphic document. If the document is a parent document with children documents opened, all graphics will be closed.

There is an alternative to close graphics. A specific graphic can also be closed if the script passes the graphic's name as parameter to the SVGraphics.Close(). The image below shows a script closing "Graphics2" screen. If the Graphics2 document is a parent document with children documents opened, all graphics will be closed.

Graphics1 X				Ŧ
1 SVGraphics.Close("Graphics2"	');			
- C				>
Line: 1				
Mouse Up 📄 🛛 Mouse Down 🛛	Mouse While	Mouse Right Up	Mouse Right Down	Mouse Double Click
Mode View 🚺 🚬 🗖 Graphic It	ems: Button	n1(GButton) 👻		

5.5. Open

How to use a function to open a graphic will be explained in this example.

SVGraphics.Open()

Configure the "mouse up" script in a button object as shown in the image below:

Graphics1	x;
1 SVGraphics.0)pen("Graphics2");
<	>
Line: 1	
Mouse Up 📄	Mouse Down Mouse While Mouse Right Up Mouse Right Down Mouse Double Click
Mode View 🖪	Graphic Items: Button1(GButton) -

In this example, the button will open the graphic document "Graphics2".

5.6. ProjectPath

How to use a function that returns a string value containing the path of the project will be explained in this example.

SVApplications.ProjectPath()

Configure the script for the "Startup" Service document as show in the image below:

Navigation Tree 🛛 🔻 🕂 🗙	Startup X
 FunctionsProject Tags SystemTags 	Service Settings Description:
 I ags I Data Types I Graphics I Templates 	Service Script
 Services Startup Triggers Alarm History Tag History Recipes Reports Tunneling Drivers Drivers Images User Functions Library System Functions Library 	1 @tagA = SVApplications.ProjectPath();
	Line: 1

Every time the application starts, the script in the "Startup" service will execute once. In the example above, the tagA will be initialized with the SVApplication.ProjectPath().

5.7. ToLower

This function modifies the string parameter to lower case.

SVString.ToLower()

Configure a trigger document as show in the image below:

ggers Items				
Event Type: Trigger	Action Type: Expressions	Trigger: TagStr	Action: @TagStr = SVStrin	
Type: Trigger	Action Type: Express	ions ~		
Trigger: TagStr				
Expression: @TagStr = S	/String.ToLo			
Script Name: Trigger		×		
Script Name: Trigger				
1 @TagStr = SVString.To	Lower(@TagStr);			

Every time the tag "TagStr" changes its value it will execute the expression configured. In this example, it will transform the value of "TagStr" to lower case.