



SVFunctions Overview

Summary

ADISRA SmartView HMI/SCADA Software includes a built-in System Function Library that includes a wide-ranging, powerful set of functions to allow a user to easily build an application. These functions can be used in any script (i.e., service document, trigger, screen script, etc.) in the application to perform the desired functionality

The ADISRA SmartView System Functions are organized into different groupings based on the type of functionality these library modules perform. These functionality groupings include:

a. SVAlarms

The SVAlarms group contains functions that enables the user to edit tags alarms, interact with the alarms notified and receive their current alarm status. This makes the ADISRA SmartView alarms dynamic with the application that will be created.

b. SVApplications

The SVApplications group contains functions that enables the user to edit the application execution by changing its language, opening an external application, sending an email etc.

c. SVDatabase

The SVDatabase group contains functions that are used to get or edit the data from a database that ADISRA SmartView is connected with. You may also insert connection string to make a database connection only when these functions are executed.

d. SVEvent

The SVEvent group contains functions to send a message wherever the user sets it. Those messages can displayed in the Alarm/Event object if its type is "Event" or "Event and History".

e. SVFile

The SVFile group functions allows the user to manipulate directories or files, including, for example, editing their content or moving a file or a directory.

f. SVGraphics

The SVGraphics group functions enable you to create or delete graphics and get the mouse position in the open graphic.

g. SVHistory

The SVHistory group functions enable you to create, delete or edit history documents and get their configurations.

h. SVIntegration

The SVIntegration group functions enable the user to collect or edit the information of XML or JSON files.

i. SVMath

The SVMath group functions provide mathematical functions to facilitate the calculation of trigonometric values, logarithms, geometry functions etc. It also has functions that manipulate values such as rounding a fractional value.

j. SVRecipe

The SVRecipe group functions enable you to create, delete or edit recipe documents and perform their functionality.

k. SVReport

The SVReport group functions enable you to change the report document format, print the report and change its save path.

l. SVSecurity

The SVSecurity group functions allow you to configure security settings, enabling you to set the login, logout and even create or delete users and profiles

m. SVString

The SVString group functions enable you to format and filter string values or string tags.

n. SVTags

The SVTags group functions are used to manipulate tags or tag properties.

Using a System Library Function

ADISRA SmartView supports the modern, powerful C# programming language that can be used with an HMI/SCADA application. C# script can be used with graphics screens (e.g. On Open, On While (while open), and On Close) and graphics objects (e.g. when a button is pushed), as well as background Services (e.g. scripts that execute based on an interval) and Triggers (scripts run based on a trigger).

Depending on ADISRA SmartView System Function that you use, there may be various parameters (arguments) used when executing the System Library function to specify items such as a C# variable, a Tag name, a date/time value, alarm types, etc. that the System Function is to use. Note that the parameters may need to be of a specific data type (e.g. boolean, integer, float, string, datetime, a List Object or a Dictionary Object). The System Library function may return a status or one or more return parameter(s), depending on the System Function.

```
List<string> types = new List<string>(){ "Lo", "LoLo", "Hi", "HiHi" };  
SVAlarms.Ack("abc", types);  
//tag abc is an integer tag, acknowledges Lo, LoLo, Hi and HiHi alarm states
```

System Functions

ADISRA SmartView has a large variety of System Functions that can be used in a project. These System Functions use various parameters (arguments) to specify items such as the Tag name, date/time value, alarm types, etc. that the System Function is to use. Note that the parameters may need to be of a specific data type (e.g. boolean, integer, float, string, datetime or a list object). The following is a list of parameters used with the various System Functions.

The following is detailed information on the various ADISRA SmartView System Functions.

A. SVALarms

Tags can have various alarm types and alarms limits associated with the specified tag. Typically, alarms are configured in the Tags document/alarm mode view page. The SVALarms system functions can be used to enable/disable alarms, acknowledge alarms that were activated, and to get the alarm state of one or multiple tags. **Note that there are additional Alarm functions in the SVTags System Functions group (e.g. retrieving alarm configuration, retrieving alarm history settings, etc.).**

1. **Ack** SVALarms Function.

Used to acknowledge one or more alarm types for a specified tag.

2. **Disable** SVALarms Function

There are multiple **Disable** Alarm functions to disable different alarm types for a specified tag. Once the disable function is executed, in the Tags Document/Alarms page the checkbox for the alarm type will be unchecked and other fields such as limits, message, priority, etc. will be cleared (requires the ADISRA SmartView Engineering Environment to be closed and reopened for the updates to appear).

Notes:

- If the specified tag did not have the alarm function enabled, the function will do nothing. However, a message will be shown in the Engineering Log indicating an invalid tag error.
- Changes made by the execution of the SVALarms Disable system function will be only updated in the Tags/Alarms Document when ADISRA SmartView is closed and reopened.

3. **Enable** SVALarms Function

There are multiple **Enable** Alarm functions to enable different alarm types for a specified tag. Once the enable function is executed, in the Tags Document/Alarms page, the checkbox for the alarm type will be checked and other fields such as limits, message, priority, etc. will be specified (requires the ADISRA SmartView Engineering Environment to be closed and reopened for the updates to appear).

4. **GetAcknowledgedTagsNumber** SVALarms Function

The GetAcknowledgedTagsNumber is used to return the number of tags that were acknowledged after being activated or normalized. Additional parameters can be added to filter the results based on the alarms acknowledged after a specific date/time, based on a specific alarm type, or alarm state (an alarm that is still active but was acknowledged, or was normalized and was already acknowledged).

5. **GetAlarmedTagsNumber** SVALarms Function

These functions are used to count the number of tags with one or more alarms that are in an active state. It is possible to filter these alarms based on set filters to return, for example, the number of tags with the Deviation and Freeze alarms active.

6. **GetAllAcknowledgedTags** SVALarms Function

These functions are used to make a list with the name of tags with one or more alarms that were acknowledged after being activated or normalized. It is possible to filter the returned tag name list based on a start date/time or whether the alarm was acknowledged after being normalized or activated.

7. **GetAllAlarmedTags** SValarms Function

These functions are used to make a list with the name of tags with one or more alarms that are activated. It is possible to filter the returned tag name list based on a start date/time or type of alarm.

8. **GetAllNormalizedTags** SValarms Function

These functions are used to make a list with the name of tags with one or more alarms that were normalized. It is possible to filter the returned tag name list based on a start date/time or type of alarm.

9. **GetNormalizedTagsNumber** SValarms Function

These functions are used to count the number of tags with one or more alarms that were normalized. It is possible to filter these alarms based on set filters to return, for example, the number of tags with the Deviation and Freeze alarms that were normalized.

10. **GetTagAlarmsByState** SValarms Function

This function is used to make a list with only specified type of a tag's alarms in one of the three alarms states (Active, Acknowledged or Normalized). The state used to find the alarms type is decided by the user when setting the function parameter.

11. **GetTagAlarmStatus** SValarms Function

This function is used to get a string value that indicates the state of one alarm from a tag. The return can indicate if the alarm never activated, is activated, is acknowledged or normalized. The function parameters allow the user to choose the tag and the alarm type the function will interact with.

B. SVApplications

The SVApplications group contains functions that enables the user to edit the application execution by changing its language, opening an external application, sending an email, etc.

1. **AppPath** SVApplication Function

This function is used to return a string value indicating where the application file is located.

2. **CreateEmail** SVApplication Function

This function is used to set up an e-mail connection to an e-mail server. The e-mail connection function, once executed, defines the e-mail connection that will be kept for the duration of the project unless it is removed.

3. **CreateWhatsApp** SVApplication Function

This function creates a WhatsApp alias to communicate with a WhatsApp client device

4. **DeleteEmail** SVApplication Function

This function is used to delete an email configuration from the project

5. **DeleteWhatsApp** SVApplication Function

This function deletes a specified WhatsApp alias.

6. **DisconnectViewer** SVApplication Function

This function disconnects the current viewer. Once disconnected, the port that was being used by that viewer becomes available for another machine to use. This function does not stop the runtime when executed, and it does nothing when executed on the web.

7. **GetEmailList** SVApplications Function

This functions returns a list object containing all email alias's (unique email configurations)

8. **GetLanguage/SetLanguage** SVApplications Function(s)

This function is used to change the application language at runtime or convert a string to a different language based on settings in the Language Document. If the Language document is not configured, there will be no translation

9. **GetWhatsAppList** SVApplications function

This functions returns a list object containing all WhatsApp alias's (unique WhatsApp configurations)

10. **GetWhatsAppSettings** SVApplications function

This function returns a dictionary object containing the settings of a specified WhatsApp alias

11. **OpenDataWatcher** SVApplications Function

This is a function that opens the Data Watcher (allows the user to monitor/set tags at runtime).

12. **OpenRegisterLicense** SVApplications Function

This function allows the user to open the Register License executable at runtime.

13. **Output** SVApplications Function

This function allows the user to log a message to the Runtime Log.

14. **PauseEngine** SVApplications Function
This function can be used to Pause various functions (engines) in ADISRA SmartView such as Tag History Documents, Driver Documents, Trigger Documents, Alarm History, Service Documents, Recipe Documents, Graphics Documents, etc.
15. **ProjectPath** SVApplications Function
This function returns a string containing the project folder path.
16. **RestartEngine** SVApplications Function
This function can be used to Restart various functions (engines) in ADISRA SmartView such as Tag History Documents, Driver Documents, Trigger Documents, Alarm History, Service Documents, Recipe Documents, Graphics Documents, etc.
17. **Run** SVApplications Function
This function allows the user to execute an external application or open a file.
18. **RunAndWait** SVApplications Function
This function allows the user to execute an external application, and once executed, the user will not be able to interact with the Viewer until the application is closed. It is also possible to set a timer to decide for how long the application will remain open.
19. **SendEmail** SVApplications Function
This function will send an email to a defined email address.
20. **SendWhatsApp** SVApplications Function
This function will send a WhatsApp message to a specified number.
21. **SetDecimalNumbers** SVApplications Function
This function defines the number of decimal places that will be used by decimal numbers during runtime. It is necessary to reopen the screens, during runtime, to update the project with any changes this function implements. The ADISRA SmartView is configured to be able to show the maximum of 8 decimal places.
22. **ShutdownWindows** SVApplications Function
This function is used to force Windows to shut down or restart. It is possible to set when Windows will shut down/restart and whether a message informing the user that Windows will shut down/restart will be displayed or not
23. **Sleep** SVApplications Function
This function is used to suspend a script for a specified number of milliseconds.
24. **StartEngine** SVApplications Function
This function can be used to Start various functions (engines) in ADISRA SmartView such as Tag History Documents, Driver Documents, Trigger Documents, Alarm History, Service Documents, Recipe Documents, Graphics Documents, etc.
25. **StopApp** SVApplications Function
This function is used to stop the application runtime without the need of using the ADISRA SmartView engineering environment.

26. StopEngine SVAApplications Function

This function can be used to Stop various functions (engines) in ADISRA SmartView such as Tag History Documents, Driver Documents, Trigger Documents, Alarm History, Service Documents, Recipe Documents, Graphics Documents, etc.

27. StopViewer SVAApplications Function

This function allows the user to close the Viewer that executes this function. This function is typically used when multiple Viewers are executing on the same project. A Boolean return value indicates if it was successfully closed.

28. UnpauseEngine SVAApplications Function

This function can be used to Unpause various functions (engines) in ADISRA SmartView such as Tag History Documents, Driver Documents, Trigger Documents, Alarm History, Service Documents, Recipe Documents, Graphics Documents, etc.

C. SVDatabase

The SVDatabase group contains functions that are used to get or edit the data from a database that ADISRA SmartView is connected with. You may also insert connection string to make a database connection only when these functions are executed.

- 1. AppendTagsToDatabase** SVDatabase Function
This function is used to append one or more tags to a Database Driver.
- 2. CreateDatabase** SVDatabase Function
This function allows a user to create a new Database Driver Document or edit an existing one during runtime.
- 3. CreateTableQuery** SVDatabase Function
This function allows the user to add new queries to the Database document or edit an existing query.
- 4. DBDelete** SVDatabase Function
This function allow the user to delete one or more records from an existing database.
- 5. DBInsert** SVDatabase Function
This function allowing the user to insert a new record in a database table.
- 6. DBSelect** SVDatabase Function
This function allows the user to receive data from a database table in response to a SQL query.
- 7. DBUpdate** SVDatabase Function
This function allows the user to update one or more records in a database table.
- 8. GetDBConnections** SVDatabase Function
This function allows the user to get a List object with the name of each Database connection.
- 9. OpenDBConnections** SVDatabase Function
This function allows the user to open the Database connections dialog box, allowing the user to add additional database connection(s) during runtime.
- 10. RemoveTagsFromDatabase** SVDatabase Function
This function is used to remove one or more tags from a Database Driver Document.

D. SVEvent

SVEvent functions allows the editing of Documents or functionalities that store or execute Events. There are different Event types. These event types include;

- a Trigger (specify a tag and related expression to trigger an event,
- a Condition (specifies a C# expression, a tag/expression/function that returns a binary value),
- Calendar (a specific date/time, or a specific day/time interval), or
- Interval (executes an expression based on a time interval, e.g. every 15 minutes).

1. **CreateTrigger** SVEvent Function

This function allows the user to create a trigger document during runtime. You can use the SVEvent.SaveEvent function to configure the Event.

2. **GetEventProperties** SVEvent Function

This function allows the user to get a List Object with the configurations settings for the Event fields on the Settings ribbon.

3. **GetTriggerHeader** SVEvent Function

This function allows the user to get a List Object with the configuration items in a Trigger document.

4. **GetTriggerInfo** SVEvent Function

This function allows the user to get a List Object with the configuration settings in a Trigger Document based on the Alias settings.

5. **GetTriggerStatus** SVEvent Function

This function allows the user to confirm if the trigger item was successfully compiled during runtime. Example: If a trigger item expression contains an error, it will not be compiled, and this function will return a value of False.

6. **SaveEvent** SVEvent Function

This function allows the user to add a trigger item to a trigger document during runtime or edit the configuration of an existing trigger item.

7. **RemoveEvent** SVEvent Function

This function allows the user to remove a trigger item from a trigger document during runtime. Once removed, the action configured to the Trigger item will no longer be executed.

8. **Send** SVEvent Function

These functions are used to log an Event message to an ADISRA SmartView file or to a database (based on the Events Settings). Events can be used record different types of activity (e.g. when scripts or triggers were executed, when a button was pressed etc.). The events are displayed in the Alarm/Event object.

9. **SetEventProperties** SVEvent Function

This function allows the user to change the Event field settings during runtime. The changes made to the Event settings using this function are updated in real time. Note that if you use this function to change the Event save type, any Events already saved will be lost. If you want to change the save type and keep the previous Events saved, make the change using the ADISRA SmartView Engineering environment.

E. SVFile

The SVFile group functions allows the user to manipulate directories or files. For example, you can copy, delete, move or edit a file or directory.

1. **Copy** SVFile Function

This function allows the project during runtime to create a copy of a file or to overwrite the contents of one file with another one. This function can be used with any type of file.

2. **CreateDirectory** SVFile Function

This function will create one or various new directories during the application runtime.

3. **CreateFile** SVFile Function

This function creates a new file in a specific location during the application runtime. The file type can be chosen by adding its extension to the end of the function parameter. If no extension is added, the file created will use the machine generic extension.

4. **DeleteDirectory** SVFile Function

This function deletes a directory during the application runtime.

5. **DeleteFile** SVFile Function

This function deletes a file in a specific location during the application runtime. This function will only recognize the file if its extension is added to the function parameter.

6. **DirectoryExists** SVFile Function

This function checks whether a directory exist in a specific path. The value returned is used to indicate if the directory exists.

7. **EnumerateDirectories** SVFile Function

This function gets the path of all directories that are inside the directory they set as the functions' parameter value. Use these functions if you want the directories path saved in a list, which enables you to edit, move and delete them. If you want to save memory by adding the paths to an array, use the functions of the page "Functions to get an array with the directories/files path" instead.

8. **EnumerateFiles** SVFile Function

This functions gets the path of all files that are inside the directory they set as the functions' parameter value. Use these functions if you want the files path saved in a list, which enables you to edit, move and delete them. If you want to save memory by adding the paths to an array, use the functions of the page "Functions to get an array with the directories/files path" instead.

9. **FileExists** SVFile Function

This function checks whether a file exist in a specific path. The value returned is used to indicate if the file exists.

10. **FindFileName** SVFile Function

This function returns the name(s) of the files included in the specified path. It is also possible to filter which files this function will be included to the list by using the second parameter.

11. **GetDirectories** SVFile Function

This function returns the path of all directories that are inside the directory set as the function parameter value. Use this function if you want the directories path saved in an array.

12. GetFiles SVFile Function

This function returns the path of all files that are inside the directory set as the function parameter value. Use this functions if you want the file paths saved in an array.

13. GetDirectoryRoot SVFile Function

This function returns the root directory of the directory/file path set to the function parameter.

14. GetFileName SVFile Function

This function returns a file name with an extension from a specified path. It also possible to use to return a directory name.

15. GetFileNamewithoutExtension SVFile Function

This function returns a file name without an extension from a specified path. It also possible to use to return a directory name.

16. LineCount SVFile Function

This function provides a count of the lines in any text file or source code file. If the file being used contains anything besides text (such as an image, video etc.), the function will decode the numbers in it to read as a text, then, will count the number of lines in it.

17. Move SVFile Function

This function moves a specified file to another location. It is also possible to specify a new file name.

18. MoveDirectory SVFile Function

This function moves a specified directory with the content inside it to another location. It is also possible to specify a new directory name.

19. ReadBuffer SVFile Function

This function returns the text of a specified text file. This function will not return the correct value if the file set is not encoded, which means this function does not work with docx, pdf or rtf files.

20. ReadLine SVFile Function

This function returns a specified line from a specified text file. This function will not return the correct value if the file set is not encoded, which means this function does not work with docx, pdf or rtf files.

21. Write SVFile Function

This function writes a specified text string to a file's last line. This function will not return the correct value if the file set is not encoded, which means this function does not work with docx, pdf or rtf files.

22. WriteLine SVFile Function

This function writes a specified text string to a specified file text by writing to its last line and when it is done, another line is added to the file. This function will not return the correct value if the file set is not encoded, which means this function does not work with docx, pdf or rtf files.

F. SVGGraphics

The SVGGraphics group functions enable you to create or delete graphics and get the mouse position in the open graphic.

1. **Close** SVGGraphics Function

This function is used to close the screen where the script was executed or a specified screen during runtime. If the user closes a screen that was used to open other screens, the additional screens will be closed.

2. **GetMousePosition** SVGGraphics Function

This function returns the current mouse position (X,Y).

3. **Open** SVGGraphics Function

This function is used to open a specified screen during runtime. It is also possible to set the function to open a screen while connecting tags to its screen tags.

4. **SaveGraphicAsHTML** SVGGraphics Function

This function allows you to save a specified graphic as HTML at runtime.

5. **SaveAllGraphicsAsHTML** SVGGraphics Function

This function allows you to save all graphic documents as HTML at runtime.

G. SVHistory

The SVHistory group functions enable you to create, delete or edit Tag History documents.

- 1. AddSaveMode SVHistory Function**
This function allows a new save mode for the Tag History document to be added.
- 2. AppendTags SVHistory Function**
This function adds one or more tags into a Tag History document.
- 3. ChangeSettings SVHistory Function**
This function changes the Tag History document settings. E.g. Never delete, days to keep.
- 4. ChangeType SVHistory Function**
This function changes the type of the Tag History document. E.g. (proprietary, database)
- 5. Create SVHistory Function**
This function creates a new Tag History document.
- 6. Delete Document SVHistory Function**
This function deletes a Tag History document.
- 7. ExportTagHistoric SVHistory Function**
This functions creates a txt, csv or doc type file with the saved tag history of a selected tag.
- 8. ExportTagHistoricByTimestamp SVHistory Function**
This function creates a txt, csv or doc type file with the saved history of specified tags.
- 9. GetHistoryHeader SVHistory Function**
This function returns a List object with all the history documents settings.
- 10. GetHistoryTagList SVHistory Function**
This function returns a List Object with all the Tag History document tags.
- 11. LoadHistoryToTag SVHistory Function**
This function allows the user to load the latest value(s) of a tag's history that is/was being saved to another tag or an array of tags. You can also specify the date from which the history should be loaded, the number of days it should count from the start date onwards, and the number of history values to load.
- 12. RemoveSaveMode SVHistory Function**
This function that removes a Tag History document save mode.
- 13. RemoveTags SVHistory Function**
This functions removes one or more tags from a Tag History document.
- 14. UpdateTagsDeadBand SVHistory Function**
This function updates one or more tag's DeadBand value in a Tag History document

H. SVIntegration

The SVIntegration group functions enable the user to collect or edit the information of XML or JSON files.

JSON Functions

1. **JsonReadDictionary** SVIntegration Function
This function enables reads all values in a JSON file and returns these values in a C# Dictionary object.
2. **JsonReadNodes** SVIntegration Function
This function reads all values from an JSON file and puts the result into a C# List object.
3. **JsonReadString** SVIntegration Function
This function reads all values from an JSON file and puts the result into a string.
4. **JsonWriteDictionary** SVIntegration Function
This function enables writes values from a C# Dictionary object into an JSON file. Values are separated by commas.
5. **JsonWriteString** SVIntegration Function
This function writes a string to an JSON file.

XML Functions

6. **XmlReadDictionary** SVIntegration Function
This function enables reads all values in an XML file and returns these values in a C# Dictionary object.
7. **XmlReadNodes** SVIntegration Function
This function reads all values from an XML file and puts the result into a C# List object.
8. **XmlReadString** SVIntegration Function
This function reads all values from an XML file and puts the result into a string.
9. **XmlWriteDictionary** SVIntegration Function
This function enables writes values from a C# Dictionary object into an XML file. Values are separated by commas.
10. **XmlWriteString** SVIntegration Function
This function writes a string to an XML file.

I. SVMath

The SVMath group functions provide mathematical functions to facilitate the calculation of trigonometric values, logarithms, geometry functions etc. It also has functions that manipulate values such as rounding a fractional value.

1. **Abs** SVMath function

This function returns the absolute value of the number passed to it while retaining the data type of parameter.

2. **BigMul** SVMath function

This function computes the full product of two 32-bit numbers multiplied.

3. **Ceiling** SVMath function

This function calculates the largest integer of the passed argument.

4. **DivRem** SVMath function

This function calculates the quotient between two whole numbers. The result is returned by the function and the remainder is set to the function's third parameter.

5. **Exp** SVMath function

This function returns the "e" raised to the specified power. "e" is a mathematical constant whose value is approximately 2.71828.

6. **Floor** SVMath function

This function calculates the smallest integer of the passed argument.

7. **Geometry Functions**

a. **CircleArea** SVMath function

This function uses the radius parameter value to calculate and return the area of a circle.

b. **CircleSectorArea** SVMath function

This function considers two concentric circles, one with innerRadius value and the other with the outerRadius value, and subtracts them to find the area between them.

c. **PyramidVolume** SVMath function

This function uses the two parameters values to calculate and return the volume of a pyramid.

d. **RectangleArea** SVMath function

This function uses the two parameters values to calculate and return the area of a rectangle.

e. **SphereArea** SVMath function

This function uses the radius parameter value to calculate and return the area of a sphere.

f. **SphereVolume** SVMath function

This function uses the radius parameter value to calculate and return the volume of a sphere.

g. **SquareArea** SVMath function

This function uses the side parameter value to calculate and return the area of a square.

h. **SquareAreaDiagonal** SVMath function

This function uses the diagonal parameter value to calculate and return the area of a square.

i. **TrapezoidArea** SVMath function

This function uses the three parameters values to calculate and return the area of a trapezoid.

j. TriangleArea SVMath function

This function uses the two parameters values to calculate and return the area of a triangle.

k. TubeCylindreVolume SVMath function

This function finds the volume of two cylinders and subtract them to find and return the volume of a hollow cylinder.

8. GetE SVMath function

This function returns the approximate value of the constant e.

9. GetPI SVMath function

This function returns the approximate value of the number pi or π .

10. IEEERemainder SVMath function

This function returns the remainder resulting from the division of a specified number by another specified number.

11. Log SVMath function

This function returns the logarithm of a number to a specified base or base e.

12. Log10 SVMath function

This function returns the logarithm of a specified number with a base 10.

13. Max SVMath function

This function allow the user to get the larger of the two specified numbers. This function works with multiple variable data types.

14. Min SVMath function

This function allow the user to get the smaller of the two specified numbers. This function works with multiple variable data types

15. Pow SVMath function

This function is used to calculate a number raised to the power of some other number.

16. Round SVMath function

This function is used to round a value to the nearest integer or to the particular number of fractional digits.

17. Sign SVMath function

This function is used to specify the sign of a number.

18. Sqrt SVMath function

This function is used to return the square root of a specified number.

19. Trigonometric Functions

a. Cos SVMath function

This function calculates the cosine of the specified radian value.

b. Sin SVMath function

This function calculates the sine of the specified radian value.

c. Tan SVMath function

This function calculates the tangent of the specified radian value.

d. Acos SVMath function

This function calculates the inverse of the cosine operation, returning a radian value.

e. Asin SVMath function

This function calculates the inverse of the sine operation, returning a radian value.

f. Atan SVMath function

This function calculates the inverse of the tangent operation, returning a radian value.

g. Atan2 SVMath function

This function calculates the dividend (sine) and divides it by the divisor (cosine), returning the result, a tangent value, that is used to return the radian value.

h. Cosh SVMath function

This function calculates the hyperbolic cosine of the specified radian value.

i. Sinh SVMath function

This function calculates the hyperbolic sine of the specified radian value.

j. Tanh SVMath function

This function calculates the hyperbolic tangent of the specified radian value

20. Truncate SVMath function

This function computes the integral part of a specified decimal number or double-precision floating-point number.

J. SVRecipe

The SVRecipe group functions enable you to create, delete or edit Recipe Documents and perform their functionality.

- 1. AppendTagsToRecipe** SVRecipe Function
This function can add one or more tags into a Recipe document.
- 2. CreateRecipe** SVRecipe Function
This function creates a new Recipe Document with default values, or edits an existing Recipe document.
- 3. GetRecipeHeader** SVRecipe Function
This function returns a List Object with all the Recipe Document settings.
- 4. GetRecipeNames** SVRecipe Function
This function returns a List Object with all Recipe Documents names.
- 5. GetRecipeTags** SVRecipe Function
This function returns a List Object with all tags names from a Recipe Document.
- 6. LoadAndWait** SVRecipe Function
This function that loads a Recipe Document and waits until it is finished loading.
- 7. Load** SVRecipe Function
This function loads a Recipe Document from either the default path or specified path.
- 8. RemoveRecipe** SVRecipe Function
This function removes a Recipe Document at runtime.
- 9. RemoveTagsFromRecipe** SVRecipe Function
This function can be used to remove one or more tags from a Recipe Document.
- 10. ReplaceTagsFromRecipe** SVRecipe Function
This function replaces all tags in a Recipe Document with tags specified in a List object.
- 11. SaveAndWait** SVRecipe Function
This function saves a Recipe Document in either the default path or a specified path and waits until the Recipe Document is saved.
- 12. SaveRecipe** SVRecipe Function
This function saves a Recipe Document in the default path or a specified path.

K. SVReport

The SVReport group functions enable you to change the report document format, print the report and change its save path.

1. ChangeFormat SVReport Function

This function changes the format (.RTF, .TXT, .CSV, .PDF, .HTML) of chosen Report Document.

2. Print

This function prints the Report Document.

3. SaveFile

This function saves the Report Document in a specified path.

L. SVSecurity

The SVSecurity group functions allow you to configure security settings, enabling you to set the login, logout and even create or delete users and profiles

1. **AddUser** SVSecurity Function
This function adds a new User. Specified parameters include User Name, Login Name, Password and Profile.
2. **AllProfiles** SVSecurity Function
This function that returns a List of string values containing all existing Profiles.
3. **AllUsers** SVSecurity Function
This function that returns a List of string values containing all existing Users.
4. **AllUsersNames** SVSecurity Function
This function that returns a List of string values containing all existing Users Names.
5. **AuditLog** SVSecurity Function
Function that adds a new entry in the audit DataGrid.
6. **ChangeUserPassword** SVSecurity Function
This function changes a specified user's password during runtime.
7. **ChangeUserProfile** SVSecurity Function
This function changes a specified user's profile during runtime.
8. **CurrentUser** SVSecurity Function
This function returns a string value of the user currently logged in.
9. **Current User Name** SVSecurity Function
This function returns the name of the current user that is logged in.
10. **GetCurrentProfile** SVSecurity Function
This function returns the profile name of the user currently logged in.
11. **CanDoAction** SVSecurity Function
This function returns a boolean value representing the CanDoAction permission for a specified profile for a specified object, or returns a Dictionary Object containing the CanDoAction permissions for all profiles for a specified object.
12. **GetObjectCanESign** SVSecurity Function
This function returns a boolean value representing if a specified profile can CanESign a specified object, or returns a Dictionary Object containing all profiles that CanESign a specified object.
13. **Get User Profile** SVSecurity Function
This function returns the profile name of the specified user.
14. **Login** SVSecurity Function
This function opens the login windows to allow a new user to login.
15. **Logoff** SVSecurity Function
Function that logs off the present user logged on.

- 16. ProfileCount** SVSecurity Function
This function returns the number of Profiles in the project.
- 17. Remove User** SVSecurity Function
This function removes a specified user from the project.
- 18. SetObjectCanDoAction**
This function sets a CanDoAction permission for a specified profile for a specified object, or uses a Dictionary Object to set CanDoAction permissions for a defined set of profiles for a specified object.
- 19. SetObjectCanESign** SVSecurity Function
This function sets a CanESign permission for a specified profile for a specified object, or uses a Dictionary object to set CanESign permissions for a defines set of profiles for a specified object.
- 20. ShowEsignWindow** SVSecurity Function
This function opens the E-sign dialog to allow for an e-signature.
- 21. ValidateESign** SVSecurity Function
This function validates an object E-sign.
- 22. SetInitialUser** SVSecurity Function
This function sets initial user at runtime. Once the script is successfully executed, the next viewers (remote and web) that connect with the application will start logged in with the new initial user.

M. SVString

The SVString group functions enable you to format and filter string values or string tags.

1. **Analyze String** Functions Group

This group of System Functions are used to analyze string Tags for certain conditions.

a. **Contains** SVString Function

This function indicates whether the specified substring occurs within the specified string.

b. **EndsWith** SVString Function

This function indicates whether the end of the specified string "source" matches the "value" parameter.

c. **Equals** SVString Function

This function indicates whether two string objects have the same value or not.

d. **IsNormalized** SVString Function

This function indicates whether the specified source string is in a particular Unicode normalization form.

e. **IsNullOrEmpty** SVString Function

This function indicates whether the specified string is null or an empty string ("").

f. **IsNullOrWhiteSpace** SVString Function

g. This function indicates whether the specified string is null, empty ("") or only contains white spaces (" ").

h. **StartsWith** SVString Function

This function indicates whether the start of the specified string "source" matches the "value" parameter.

2. **CopyTo** SVString Function

This function copies a specified number of characters from a specified position of a specified string to a specified position in an array of Unicode characters.

3. **Clone** SVString Function

This function is used to clone the string object, which returns another copy of that data.

4. **Compare** SVString Function

This function compares two specified string objects and returns an integer that indicates their relative position in the alphabetical sort order.

5. **CompareOrdinal** SVString Function

This function compares two specified string objects by evaluating the numeric values of the corresponding char in each string. The characters values are seen as Unicode.

6. **Concat** SVString Function

This function concatenates one or more instances of String, or the String representations of the values of one or more instances of Object..

7. **Copy** SVString Function

This function creates a new string instance with the same value as a specified string.

- 8. Format** SVString Function
This function is used to replace one or more format items in the specified string with the string representation of a specified object.
- 9. GetHashCode** SVString Function
This function is used to get the HashCode of a specified string. Note that this function will return a different value every time the runtime starts.
- 10. IndexOf** SVString Function
This function returns the zero-based index of the first occurrence of a string in the string specified.
- 11. IndexOfAny** SVString Function
This function returns the zero-based index of the first occurrence a string in a specified array of Unicode characters. The search starts at a specified character position.
- 12. Insert** SVString Function
This function returns a new string in which a specified string is inserted at a specified index position in the specified string.
- 13. Intern** SVString Function
This function is used to intern string value is interned. If not, it will intern it. See C# System.String.Intern for more details.
- 14. IsInterned** SVString Function
This function is used to intern string value is interned. See C# System.String.Intern for more details.
- 15. Join** SVString Function
This function joins the elements of a specified array or the members of a collection, using a specified separator between each element or member.
- 16. LastIndexOf** SVString Function
This function is used to return the zero-based index of the last occurrence of a specified character or any of the character in an array.
- 17. Normalize** SVString Function
This function returns a new string whose textual value is the same as the specified string, but whose binary representation is in Unicode normalization form C.
- 18. PadLeft** SVString Function
This function returns a new string that right-aligns the characters in this string by padding them with spaces on the left with spaces, or with a specified Unicode character, for a specified total length.
- 19. PadRight** SVString Function
This function returns a new string that right-aligns the characters in this string by padding them with spaces on the right with spaces, or with a specified Unicode character, for a specified total length.
- 20. Remove** SVString Function
This function function allows the user to remove a specified number of characters from the specified string.

- 21. Replace** SVString Function
This function allows the user to replace all the specified Unicode characters or specified string from the current string object and returns a new modified string.
- 22. Split** SVString Function
This function splits a specified string into a maximum number of substrings based on the array of characters passed as parameter. You can specify the maximum number of substrings to return.
- 23. Substring** SVString Function
This function is used to separate a specified string and return the resulting substring.
- 24. ToLower** SVString Function
This function is used to return a version of a specified string with all characters lowercase.
- 25. ToUpper** SVString Function
This function is used to return a version of a specified string with all characters uppercase.
- 26. ToCharArray** SVString Function
This function copies the characters in a specified string to a Unicode character array.
- 27. ToString** SVString Function
This function convert the specified value to its equivalent string representation.
- 28. Trim** SVString Function
This function is used to remove all leading and trailing white-space or specified characters from the specified String object.
- 29. Trim End** SVString Function
This function returns the specified string with trailing characters added to the array removed.
- 30. Trim Start** SVString Function
This function returns the specified string with leading characters added to the array removed

N. SVTags

The SVTags group System Functions are used to manipulate tags or Tag Properties.

1. **AddDynamicTag** SVTags Function
This function adds an item (property) in a Dynamic Tag.
2. **AddTagsToGroup** SVTags Function
This function adds one or multiple tags to a Tags Group Document during runtime.
3. **ChangeTagProperty** SVTags Function
This function changes the property of the specified Tag during runtime.
4. **ChangeTagPropertyUserFunction** SVTags Function
This is a function that changes the property(s) of the tag specified. It uses either a string or a Dictionary Object to define the new Tag properties during runtime.
5. **ClearDynamicTag** SVTags Function
This function removes all tag indexes from a dynamic tag.
6. **CreateArray** SVTags Function
This function creates a new array tag (up to 3-dimensions).
7. **CreateDynamicTag** SVTags Function
This function adds a new dynamic tag to a specified Tags Document during runtime.
8. **CreateTag** SVTags Function
This function creates a new tag (defines tag name and tag type).
9. **Create** SVTags Function
This function creates a new Tags Document.
10. **CreateTagsGroup** SVTags Function
This function creates a Tags Group Document during runtime.
11. **DeleteTag** SVTags Function
This function deletes a Tag.
12. **DeleteTagsGroup**
This function allows the user to delete a Tags group document during runtime.
13. **DynamicLength** SVTags Function
This function returns the current length of a dynamic Tag.
14. **Filter** SVTags Function
This function that performs a search on all project Tags and returns a List Object with the names of the Tags that are in accordance with the filters used.

- 15. GetAlarmConfig** SVTags Function
This function returns a List Object (string values) of alarm history properties for a specified Tags Document.
- 16. GetAlarmHistoryInfo** SVTags Function
This function returns a List Object (string values) of alarm history settings for a specified tag.
- 17. GetAlarmInfo** SVTags Function
This function returns a List Object (string values) of alarm settings for a specified tag.
- 18. GetCommunicationInfo** SVTags Function
This function returns a List Object (string values) of communication settings for a specified tag.
- 19. GetHistoryInfo** SVTags Function
This function returns a List Object (string values) of tag history stings for a specified tag.
- 20. GetListOfTags** SVTags Function
This function returns a List Object (string values) containing the name of all tags inside a specified Document.
- 21. GetNotificationInfo** SVTags Function
This function returns a Dictionary Object (string values) with the Alarms Notification Settings of a specified Tags Group Document.
- 22. GetSaveHist** SVTags Function
This function returns a List Object (string values) containing the Save History information for a specified Tag.
- 23. GetSimulationInfo** SVTags Function
This function returns a List Object (string values) containing the specified Tag simulation configuration.
- 24. GetTriggersInfo** SVTags Function
This function returns a List Object (string values) containing the specified Tag trigger information.
- 25. GetValue** SVTags Function
This function returns the value of a specified Tag at runtime.
- 26. RemoveDynamicTag** SVTags Function
This function removes an index from a specified Dynamic Tag at runtime.
- 27. RemoveTagCommunication** SVTags Function
This function removes any existing communications configured for a specified Tag at runtime.
- 28. RemoveTagsFromGroup** SVTags Function
This function allows the user to remove a Tag or multiple Tags using a List Object containing Tag names from a Tags Group Document during runtime.
- 29. StartTagSimulation** SVTags Function
This function starts Tag Simulation.

30. StopTagSimulation SVTags Function

This functions stops Tag Simulation.

31. SetAlarmConfig SVTags Function

This function changes a Tag Document alarm properties during runtime to define the Save Type (Proprietary vs. Database) and a database connection.

32. SetSaveHist SVTags Function

This function changes the Tag Document (Alarms section) Alarm Properties during runtime to define how long to keep alarm history for the Tag.

4. Using SVFunctions

SVFunctions are a combination of C# functions and custom engineered functions designed to be used with ADISRA SmartView to provide important functionality to efficiently execute an HMI/SCADA application. To use a SVFunction in ADISRA SmartView, you need to execute the SVFunction in a scripting code segment.

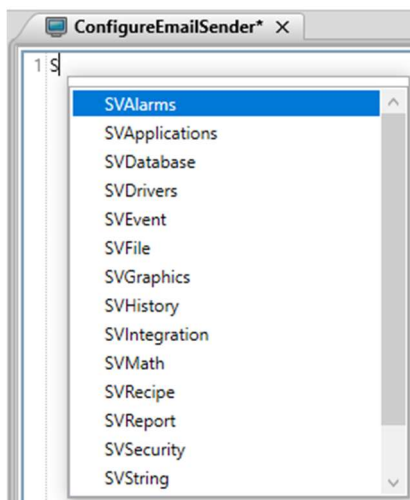
There are multiple locations that you can execute a C# script (and/or a SVFunction). These locations include:

- Graphics Document. Code can be executed when the screen is opened, while it remains open, or when the screen is closed)
- Graphics Object (e.g. a button). Code can be executed in a script segment on mouse down, mouse while down, mouse up, mouse right up, mouse right down, and mouse double-click.
- Services Document. Script executed on a Trigger or a Condition
- Trigger Document

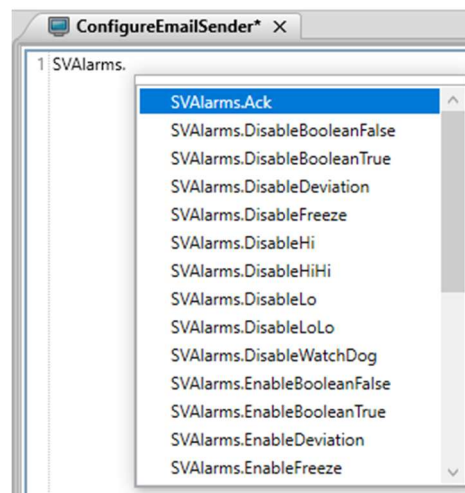
Additionally, ADISRA SmartView has a User Functions Library where you can create your own Functions and add these functions to the User Functions Library to be used throughout the application.

5. Intellisense

IntelliSense is a common code-completion tool that helps a user complete the code they are using in a Script segment. ADISRA SmartView uses IntelliSense with SVFunctions. For example, if you open a code segment (e.g. a Screen Script) and type in the letter “S”, this will assume that you are looking to add a System Function to the code. Then, you scroll down the list of System Functions groups to select (click on) the System Function you want to use. In the example below, the SVAlarms System Function was selected (clicked on). Next, when you type a period “.” following the System Function Group, this will present a list of System Functions within the Group.



System Function Groups



System Functions

Conclusion

ADISRA SmartView has a very robust System Function Library. You can visit the On-line Help for more details on each specific System Function.