

	<p>GECoe</p> <p>Software Manual</p>	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
<p>Schneider Electric Serial ASCII SmartDriver V1.6.0</p>			

**Schneider Electric Serial ASCII SmartDriver V1.6.0
Installation & User Guide**

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

Contents

1	THE DOCUMENT	3
1.1	PURPOSE	3
1.2	TERMINOLOGY	3
1.3	REFERENCE DOCUMENT	3
1.4	REVISION HISTORY	3
2	INTRODUCTION	4
2.1	DEPLOY	4
2.1.1	<i>Driver Information</i>	4
3	SMARTDRIVER INTERFACE	5
3.1	DRIVER INSTALLATION	7
3.2	SERIAL NETWORK CONFIGURATION	7
3.2.1	<i>Basic Properties</i>	7
3.2.2	<i>RS485 Properties</i>	8
3.2.3	<i>Custom Properties</i>	9
3.3	SERIAL COMMUNICATION	10
3.4	CONFIGURATION STATUS	11
3.5	STATUS	11
3.5.1	<i>Invalid</i>	11
3.5.2	<i>Online</i>	11
3.5.3	<i>Offline</i>	12
3.5.4	<i>Summary</i>	13
3.6	DRIVER LIMITATIONS	14
3.6.1	<i>Custom Properties</i>	14
3.6.2	<i>Maximum String Length</i>	14
3.6.3	<i>IsASCIIMode validation</i>	14
3.7	DRIVER OPERATION	14
3.7.1	<i>Initialization</i>	15
3.7.2	<i>Serial Port Handling</i>	16
3.7.3	<i>Transmit String Data</i>	17
3.7.4	<i>Receive</i>	18
3.7.5	<i>Custom Parameter Settings</i>	19
3.7.6	<i>Expected Custom Parameter Setup</i>	20
3.7.7	<i>Is ASCII Mode Feature Usage</i>	21
4	APPENDIX A: SERIAL TEXT MESSAGE HANDLER	22

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

1 The document

1.1 Purpose

The purpose of this document is to describe configuration and operational details of the Serial Driver. The driver has been developed to run on EBO servers as part of the EcoStruxure framework.

1.2 Terminology

SmartDriver plugin – A standard EBO server plugin loaded at server start-up. It is responsible for managing the SmartDrivers.

EBO – EcoStruxure Building Operation

ES – Enterprise Server

AS-P – Automation Server Premium

Serial device driver – A device driver that handles the data communications via the serial network.


1.3 Reference Document

EBO technical reference guide

<https://ecobuilding.schneider-electric.com/search?documentUUID=3d5cc9b3-bb99-403d-a0d0-15a12fc9c7d9&title=Technical%20Reference%20Guide%20-%20SmartStruxure%20Solution>

1.4 Revision history

Revision	Date	Comment
Rev A	2016-06-24	Initial version
Rev B	2016-07-15	Online/Offline added. Approved by Dave Chapman
Rev C	2016-08-03	Limitations added.
Rev D	2016-08-15	Driver operation described.
Rev E	2016-10-18	Schematic diagrams used
Rev F	2017-03-07	Approved by Dave Chapman
Rev G	2017-04-25	Added a sample application
Rev H	2017-04-27	Approved by Dave Chapman
Rev I	2019-06-13	Modifications to driver by J.Lutzover. New custom parameter isASCII Mode. Added sections 3.6.3, 3.7.6 and 3.7.7
Rev J	2019-08-19	Updated documentation to GECoe Template. P.Rawlingson
Rev K	2019-10-30	Fix applied to driver to correctly send 'space' 0x20 character in Hex mode.
1.4 Rev L	2020-09-14	Fix applied to handle message length longer than 128 bytes for certain configurations.
1.5.2	2020-10-29	Recompiled source.
1.6.0	2020-21-12	New Library source added. Function calls streamlined.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

2 Introduction

SmartDriver Serial ASCII Driver is a smart driver to support serial ASCII communications on EBO EcoStruxure Building Operation Servers such as the Enterprise server (ES) and Automation Server Premium (AS-P). On ES the communication will be on RS232 network and on AS-P it will be on RS485. SmartDriver Serial ASCII Driver will allow ASCII characters sent over the EBO serial port.

The driver supports only two types of points namely SmartDrivers String Output Point and SmartDrivers String Input Point.

2.1 Deploy

The serial driver is available in *sdv* packages that should be deployed on EBO servers using the EBO workstation. Two ASCII driver packages will be provided:

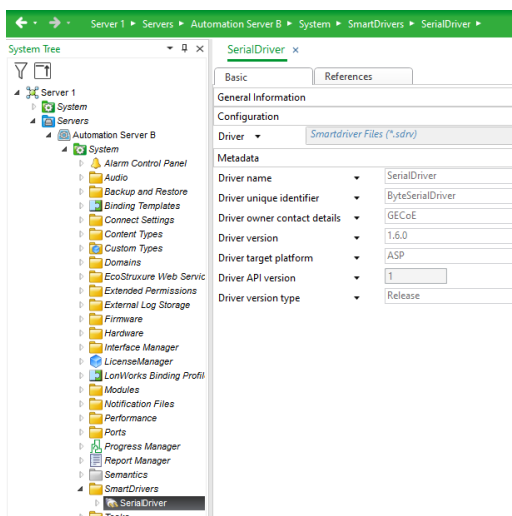
1. *ASCII ES V1_6xx.sdrv* to deploy on an ES
2. *ASCII ASP V1_6xx.sdrv* to deploy on an AS-P


Deployment allows the driver to become readily available in the server to install. It is possible to get more than one driver deployed in a server and not all have to be used.

The procedure to deploy and create SmartDrivers driver objects is described in the document EBO Technical Reference Guide.

2.1.1 Driver Information

Driver information is displayed in the property window of the SmartDrivers driver object. This information called the metadata of the driver is created when the driver is packaged by the software producer. Given below is the typical details seen for the Serial ASCII driver.

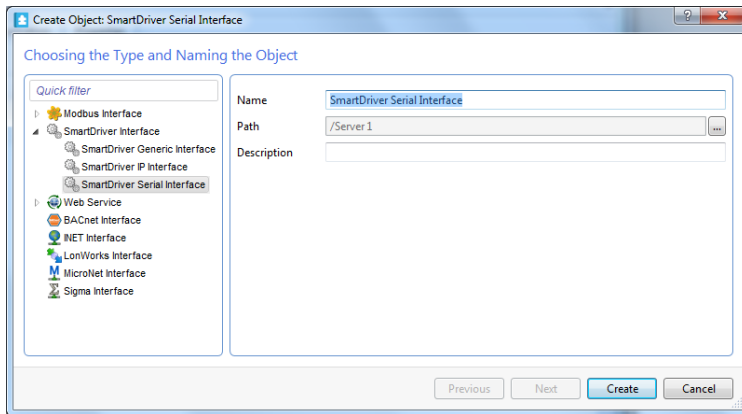


	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

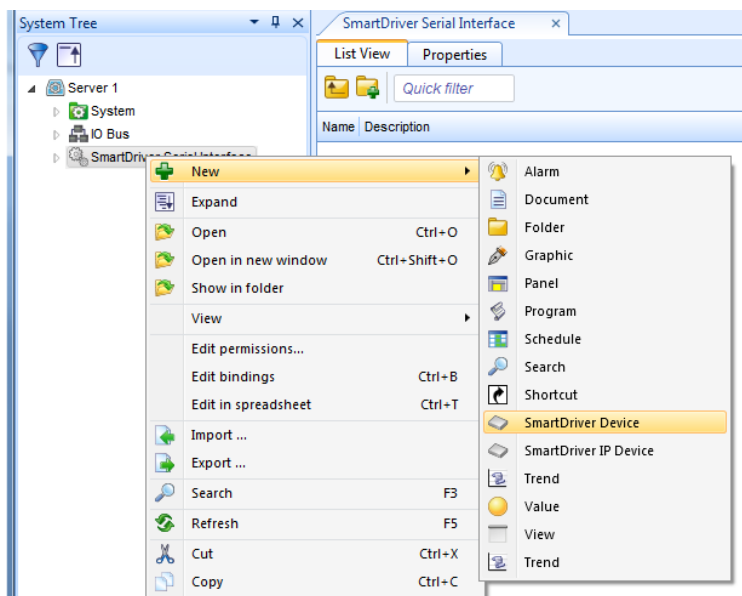
3 SmartDriver Interface


SmartDrivers design supports the hierarchical architecture. Create a Serial Interface, Device, and then String Input and Output Points as illustrated below.

Create SmartDriver Serial Interface as shown below.



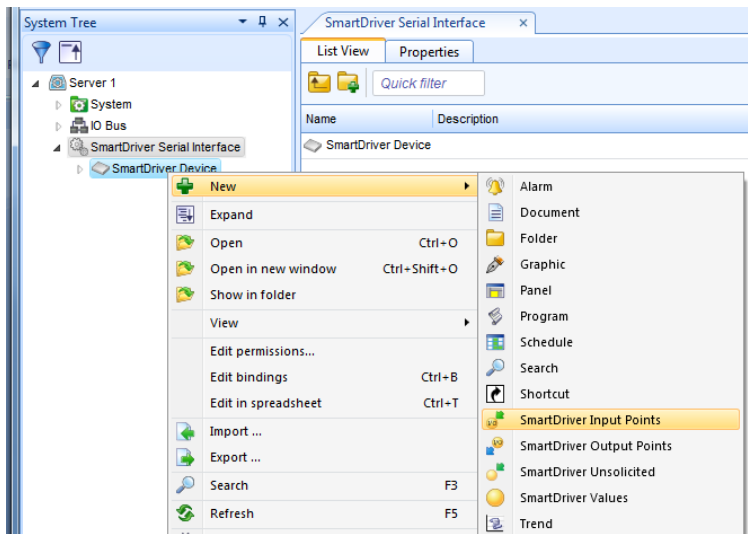
Create SmartDriver Device underneath the interface as shown below.



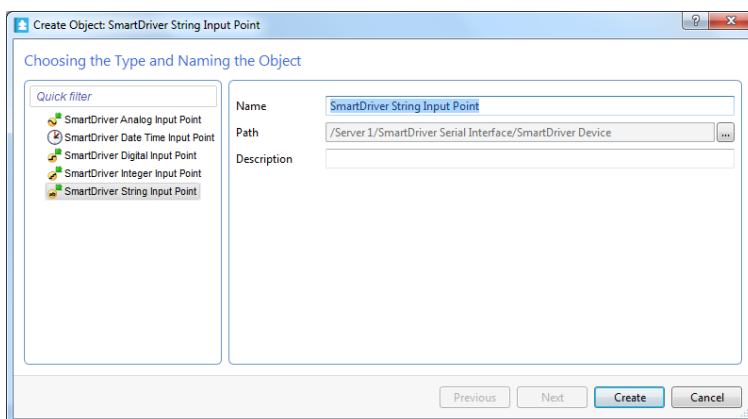
	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			


SmartDriver Serial ASCII Driver supports only two Point types namely the String Input Point and String Output Point. String Input Point enables receiving character data from the serial port depending on the communication settings configured in the system. String Output Point enables sending character data to the serial port.

Create SmartDriver Point as illustrated below.



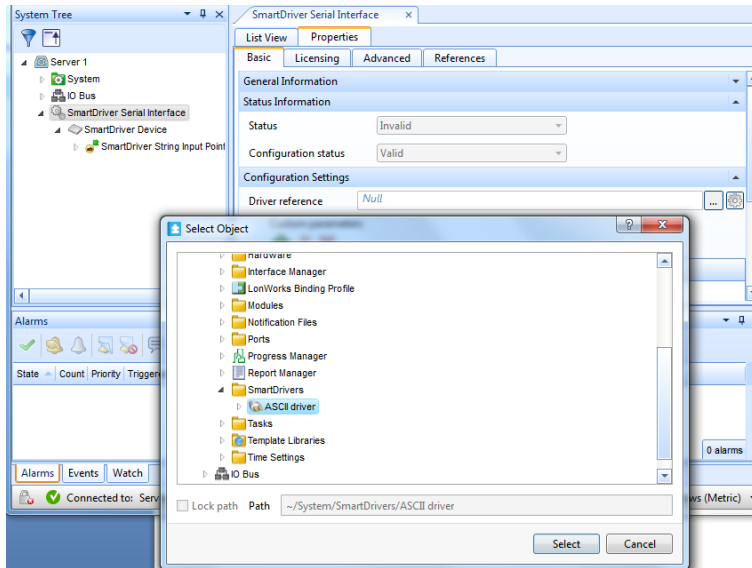
This will provide the option to choose the point data type as shown below.



	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.1 Driver Installation

SmartDriver Interface can choose to install a previously deployed driver. In the caption given below, the ASCII driver is shown to be selected for the field Driver reference.

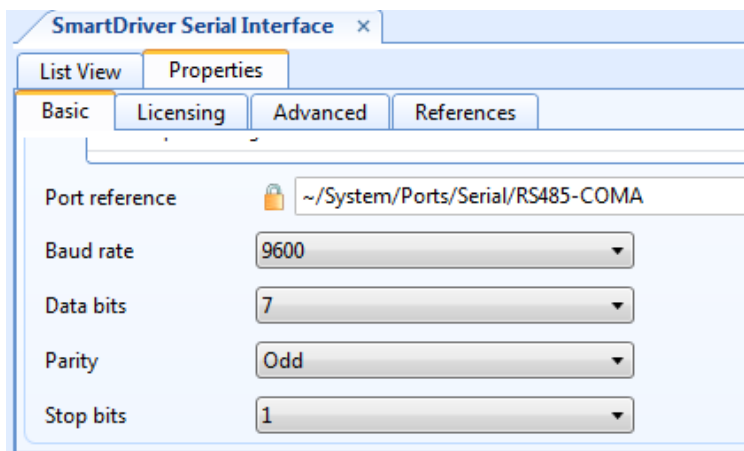


3.2 Serial Network Configuration


SmartDriver Serial Interface defines the communication properties of the network.

3.2.1 Basic Properties

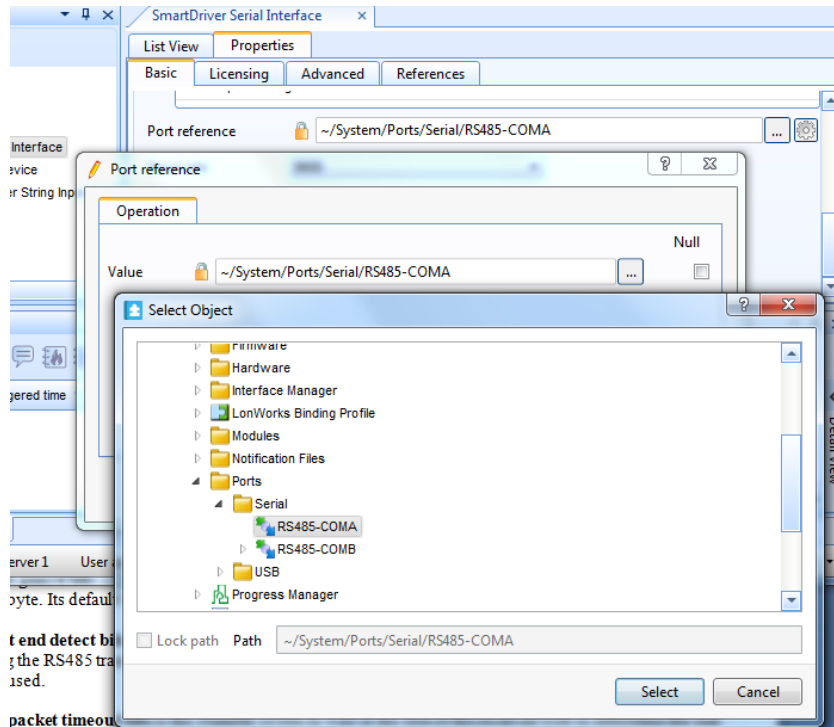
Properties common to both RS232 and RS485 networks are displayed on the property grid. The common properties and default values are show below.



Here the property Port reference is chosen from the list of ports available in the server.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

AS-P provides two ports to choose from namely RS485-COMA and RS485-COMB, as shown by the caption below.



ES provides one port by default namely COM1, but allows to create more COM ports if necessary.

3.2.2 RS485 Properties


Three RS485 driver specific properties are provided as shown in Advanced tab. These properties will help to adjust the communication characteristics on AS-P depending on the field device.

Transmit guard bits is the number of bit delays introduced in addition to the stop bits at the end of the transmit byte. Its default value is 2 bits.

Transmit end detect bits is the number of bit delays introduced at the end of the transmit packet before switching the RS485 transceiver to Receive Mode. For its default a value hardcoded in the RS485 device driver is used.

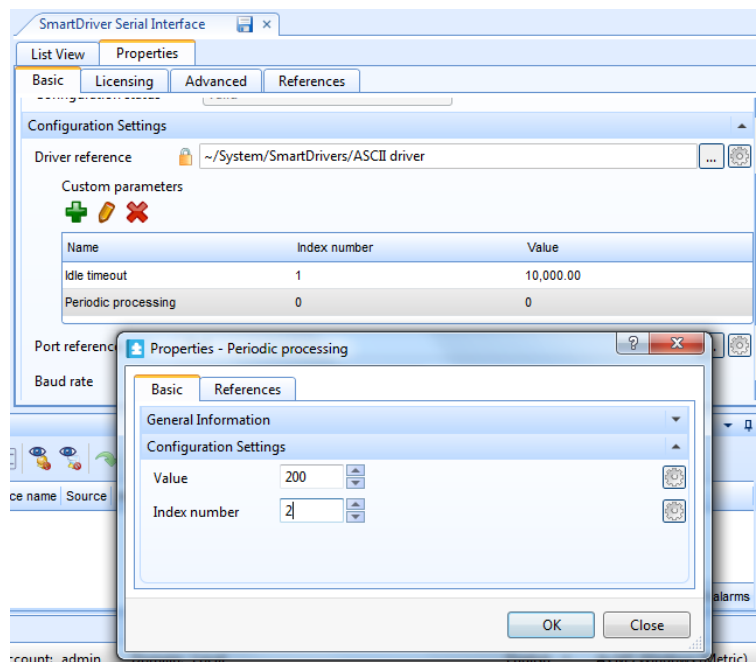
Receive packet timeout bits is the Number of bits to wait at the end of the received byte to determine the end of the packet. Its default value is 88 bits which is the bit equivalent to 11 characters.

It is recommended not to change any of these values unless absolutely necessary.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.2.3 Custom Properties

Network setting enables creating custom configuration properties that may be necessary to support the specific device or protocol. Custom properties are created and named by clicking the “+” sign. Right clicking the property will allow editing its fields. Each custom property in the interface will carry a unique index number, predetermined by the driver, and the value.




Serial Driver supports two such properties namely Idle timeout and Periodic processing.

Property	Index	Valid Range	Default	Data Type
Idle timeout	1	0 – 60,000 ms	100 ms	Integer
Periodic processing time	2	100 – 1000 ms	100 ms	Integer

Here Idle timeout determines the end of the text stream. If the receive network is inactive for this period of milliseconds, then the driver will assume the transmission coming in on the wire is complete. The character data received so far point will be copied to the String Input Point.

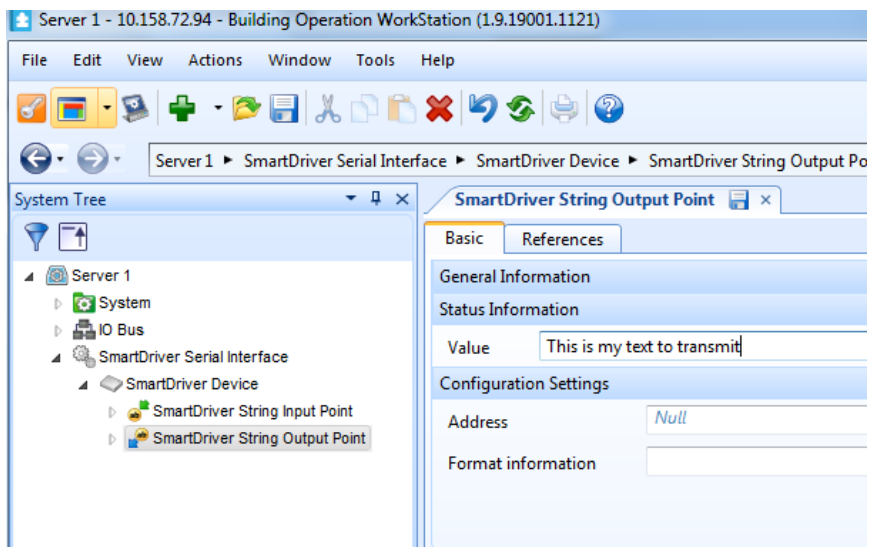
Periodic processing time is the time interval at which the driver run time operation engine is executed. This engine is responsible for transmitting, receiving and processing the characters. Lower time interval will produce faster response time to the driver.

If neither of the properties is defined, the driver will use its default value.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.3 Serial Communication

Create SmartDriver String Output Point to send ASCII string through the serial port defined in the property Port reference in SmartDrivers Serial Interface. Create SmartDriver String Input Point to receive ASCII data from the same serial port.




ASCII strings can be sent in two ways:

1. Enter the output sting in the field Value and save to initiate the transmission as shown in the caption above.
2. Drive the output point by another EBO object such as an EBO String Value or a program script. To enable this, bind the EBO object to the SmartDriver String Output Point.

SmartDriver String Input Point will hold the incoming ASCII string at the serial port. Similar to the output point described above, a binding may be used to transfer the input point ASCII string to an EBO object. The input point will collect data arriving at different intervals until it sees no activity for the continuous period defined by the custom parameter *Idle timeout*. If no parameter is defined, the point waits for 100 ms before updating Value with the ASCII data received so far in its receive buffer.

The driver will require one SmartDriver String Output Point to send and one SmartDriver String Input Point to receive the ASCII string. If configured with more than one input point, the string data may be shown to different input points on different occasions, which is not clearly acceptable. Hence only one input point is supported. If tried to create multiple input points, the property Configuration status in SmartDriver Serial Interface will display *Invalid configuration*. If configured with more than one output point, the port will transmit the ASCII string data as updated by each point. The serial ASCII driver supports multiple output points.

The maximum size of the text stream used by Serial ASCII Driver is 1024 characters.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.4 Configuration Status

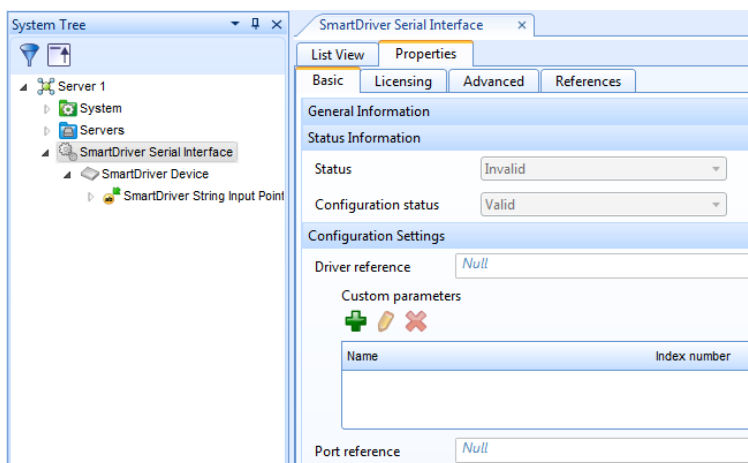
This property reports the validity of the configuration properties. As described in Section 3.3, the serial ASCII driver checks the number of **SmartDriver String Input Points configured**. No other properties are validated. The configuration status value is set to *Invalid configuration*, if two or more input points are configured and for all other settings it is set to *Valid*.

3.5 Status

The condition of the SmartDriver serial communication is reported by the property Status. This shows three states: Invalid, Offline and Online

3.5.1 Invalid


This is the default state of the interface. When created, the interface status will show Invalid. The state remains the same until the port reference is defined. The figure below shows such condition.

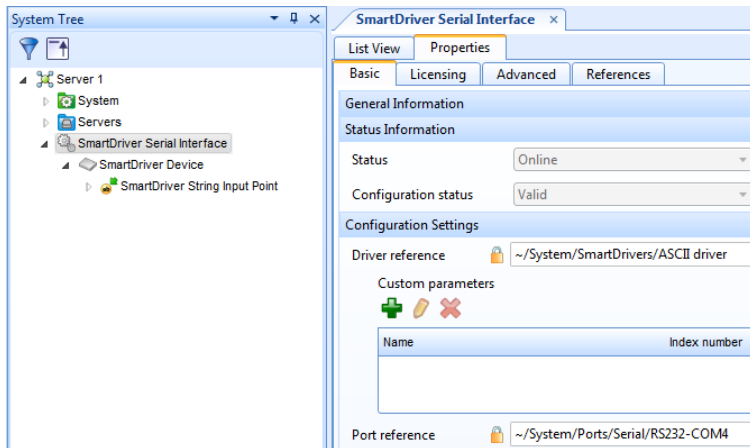


The system tree on the left pane shows no overlay icons on the SmartDriver objects.

3.5.2 Online

The Online state is displayed when the driver is installed and the port defined in the Port reference property is opened successfully. When online, no overlay icon is shown on the SmartDriver objects.

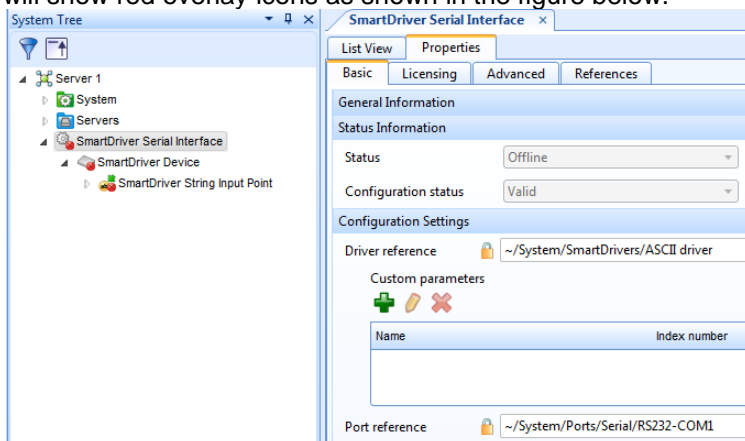
	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			




Clearing the property Port reference will change the state to Offline.

3.5.3 Offline

The offline state is displayed when the driver fails to open the port. SmartDriver objects in the system will show red overlay icons as shown in the figure below.

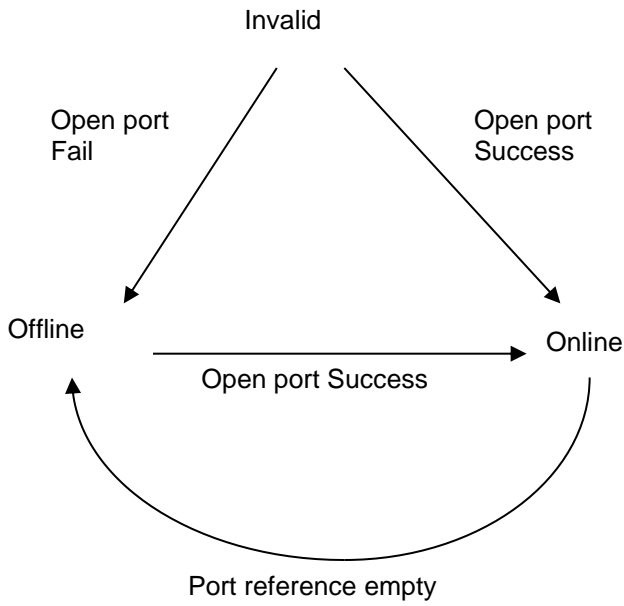



The status will become Online when the port is opened successfully to enable communications.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.5.4 Summary

Given below is the schematic of how the property Status is evaluated in SmartDriver Serial Interface.



	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.6 Driver Limitations

Given below is the table that shows the list of SmartDriver objects supported in the serial ASCII driver.

Object Type	Maximum number of objects supported	Attributes
SmartDriver Serial Interface	1	n/a
SmartDriver Device	1	n/a
SmartDriver String Input Point	1	Read only
SmartDriver String Output Point	Ideally 1 Multiple points are allowed	Read/Write

All other objects are not supported.

3.6.1 Custom Properties

Idle Timeout: 0 to 60,000 ms

Periodic Processing Time: 100 – 1000 ms

3.6.2 Maximum String Length

The ES or AS sends the ASCII string from the script program to the SmartDriver String Output Point. The output point transfers this string to the serial ASCII driver, which writes to the serial port.

The maximum number of ASCII characters handled by the script string object and also by the SmartDriver String Output point is 255 excluding the null termination character.

The maximum number of ASCII characters handled by the serial port buffer is 1024.

3.6.3 IsASCII Mode validation

When in HEX mode it is required and expected to provide a HEX sequence as a 2 digit output, no spaces or other formatting, under EBO. Invalid output might cause undesired operation of the driver.


3.7 Driver Operation

SmartDriver Plugin handles the communications between the server and the serial ASCII driver using appropriate API calls provided in the SmartDriver SDK. The heart of the serial ASCII driver is the PeriodicProcessing engine, which performs the following tasks:

- Opens and Closes the serial port.
- Sends the serial ASCII data from the transmit buffer to the serial port for transmission.
- Reads the serial ASCII data received by the serial port into the receive buffer
- Transfers the receive buffer data to the server database that updates the SmartDriver String Input Point value

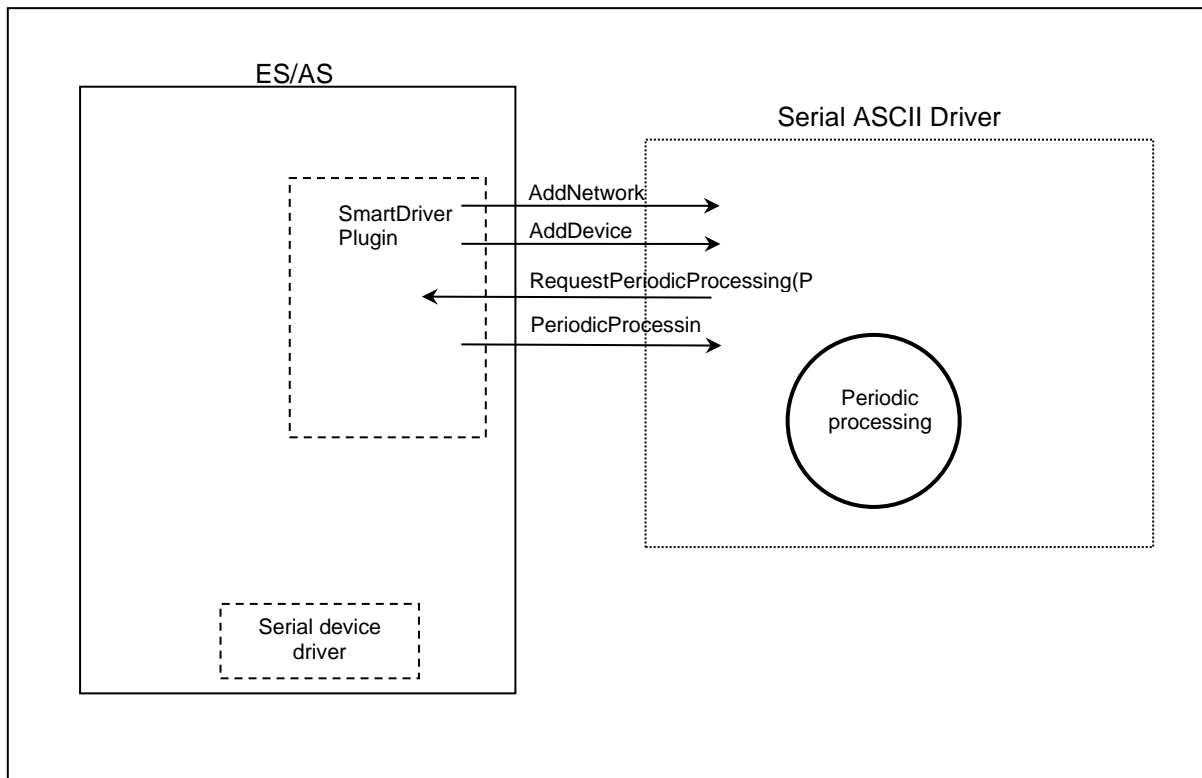
SmartDriver Serial Interface defines the rate at which the periodic processing is done. The default rate is 100 ms. However, the user can add a custom parameter to the interface to set a different processing rate as described in Section 3.2.3.


Note that the size of both the transmit and receive buffers is 1024 bytes.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.7.1 Initialization

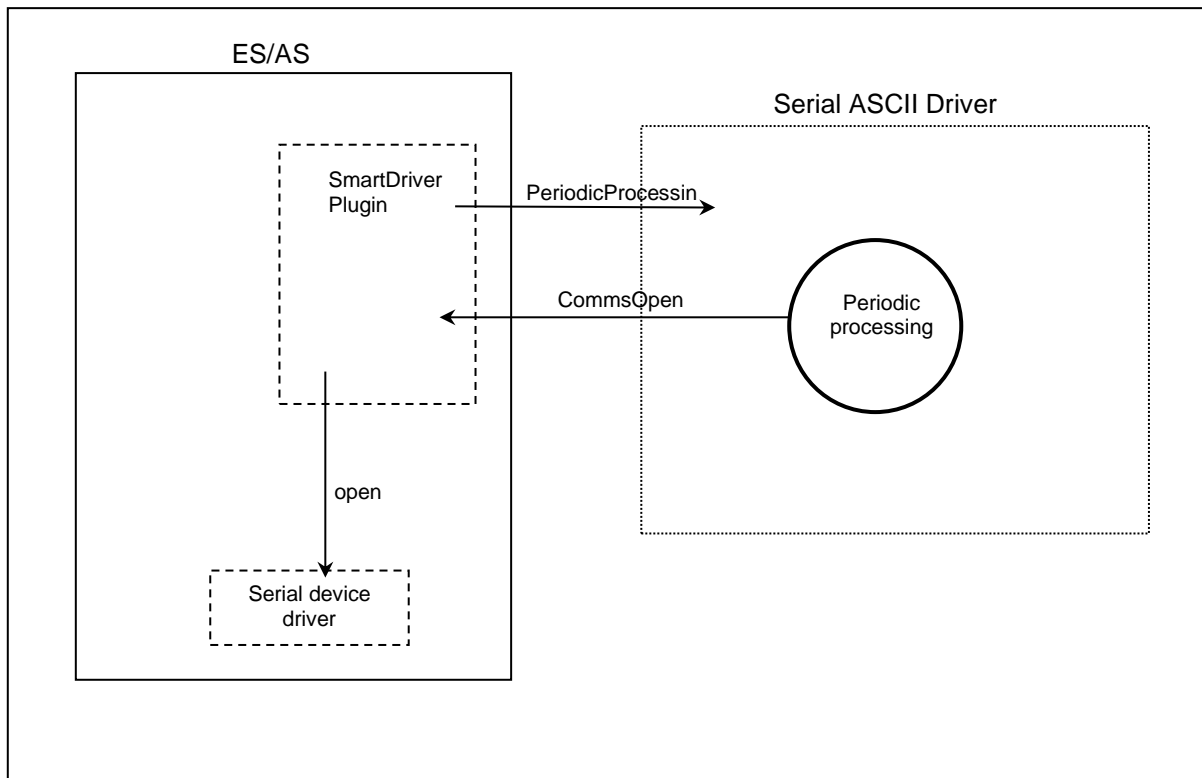
SmartDriver Plugin loads the serial ASCII driver, and then adds the serial network and the device as configured by the user. As part of the initialization, the serial ASCII driver will request the SmartDriver Plugin to make a call back to PeriodicProcessing at regular intervals. The rate at which the call made is defined by Periodic Processing Time (PPT). This sequence of events is illustrated in the schematic diagram below.




	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.7.2 Serial Port Handling

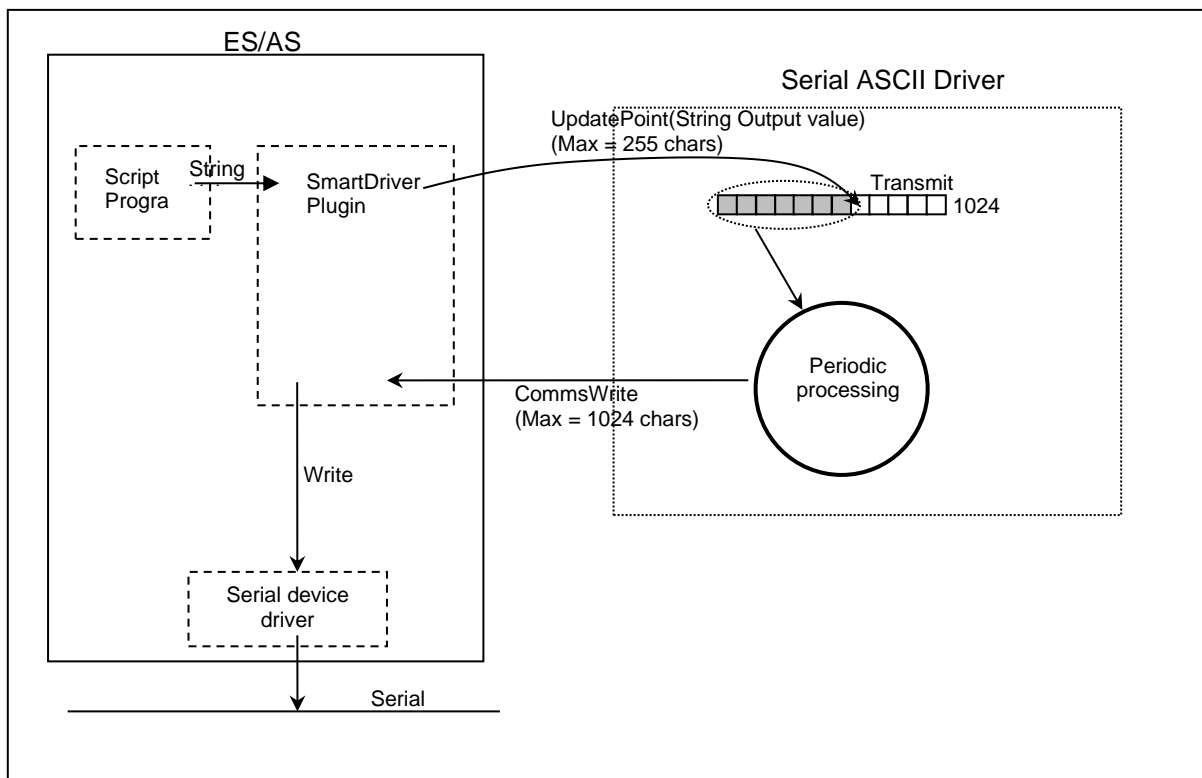
Serial ASCII driver will request the server using relevant API calls to perform desired operations on the serial port. These operations will include open, close, read, write and flush. SmartDriver Serial Interface will provide the port configuration details to set up the serial communication device. Given below is the schematic diagram showing the serial port open operation.




	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.7.3 Transmit String Data

Given below is the schematic diagram of the driver operation when sending the ASCII string data from the ES or AS to the serial network. The user or the script program updates the ASCII string data in SmartDriver String Output Point. This will force SmartDriver Plugin to send a maximum of 255 ASCII characters to the transmit buffer in the serial ASCII driver using the API call UpdatePoint. Entire string data will be discarded if the space available in the buffer is inadequate.

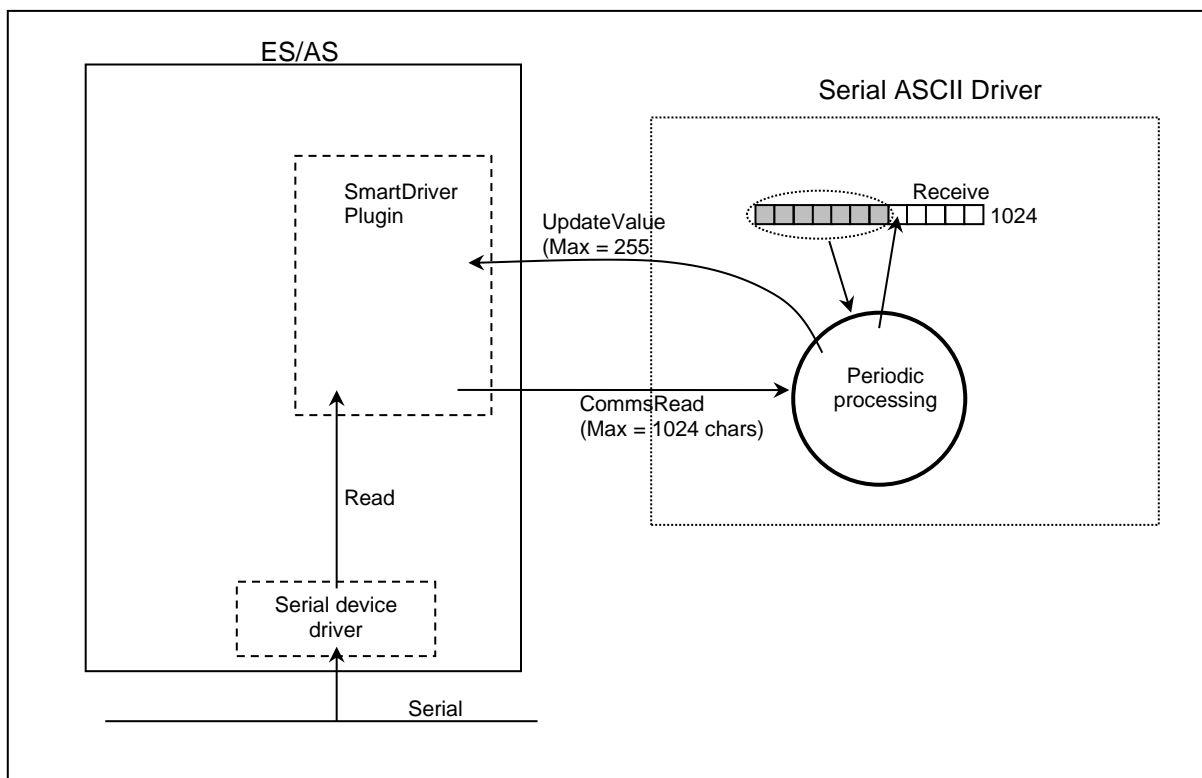


PeriodicProcessing engine will extract the ASCII string data found in the transmit buffer and send those data to the serial port using the API call CommsWrite. This process can transmit up to 1024 characters on each execution cycle of the engine. SmartDriver Plugin will invoke the serial device driver to transmit these ASCII characters on the serial network.


	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.7.4 Receive

Given below is the schematic diagram of the receive operation. Serial ASCII driver configures the AS-P RS485 communication to the packet mode. In this mode the RS485 device driver collects the incoming ASCII characters in its packet buffer until either the buffer is full or the RS network has no activity for the duration of 10 ms. Here packet buffer size has been set to 1024 bytes which is the same size as the receive buffer of the serial ASCII driver. That means the entire packet buffer data can be copied to the receive buffer.



Serial ASCII driver configures the ES RS232 communication to the character mode. In this mode the serial device driver simply reads the ASCII character data from the serial COM port as received. On each execution cycle, the PeriodicProcessing engine will read the ASCII data, if any, from the device driver using the API call CommsRead. The engine then copies these data to the receive buffer in the space available. Excess data will be discarded, which means the received string could be incomplete if the receive buffer data is not copied and flushed quickly enough. PeriodicProcessing engine can extract up to 255 ASCII characters from the receive buffer and update the SmartDriver String Input Point value with the extracted data. If the receive buffer contains more than 255 ASCII characters, then the engine will take multiple cycles to update the Input Point value. Each update will have a maximum of 255 characters extracted from the buffer. The data update to String Input Point value is only done when the time elapsed since the last update exceeds the Idle Timeout value.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			


3.7.5 Custom Parameter Settings

In general, there is no ideal setting for the Periodic Processing Time and the Idle Timeout values. For an efficient and smooth operation of the serial ASCII driver, their values should be selected appropriate to the application and network configuration settings.

Set both Periodic Processing Time and Idle Timeout to a lower value if faster response or update is required. This setting may produce incomplete messages to SmartDriver String Input Point. If the updated message is incomplete, then increase the Idle Timeout value until the whole message is seen. Set Idle Timeout to 0 and Periodic Processing to its default value 100 ms for instant data updates. This enables the incoming serial network data quickly read and updated to String Input Point value at the fastest possible rate.

Set Idle Timeout to a large value if the serial data transmission is infrequent.

For most applications, setting the Idle timeout value no less than the Periodic Processing Time will ensure the reception of complete messages.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.7.6 Expected Custom Parameter Setup

For the driver to operate it is required that the Customer Parameters and their <Index number>'s are setup as per fig a.

In the event of this being an upgrade from an older driver version, the parameter <isASCIIMode> will not exist. In that case the driver will operate as per the existing version without any Hex manipulation, this is to maintain backwards compatibility, and is by design. This being said, to tap the new features of the driver, you should add the <isASCIIMode> as per fig a below.

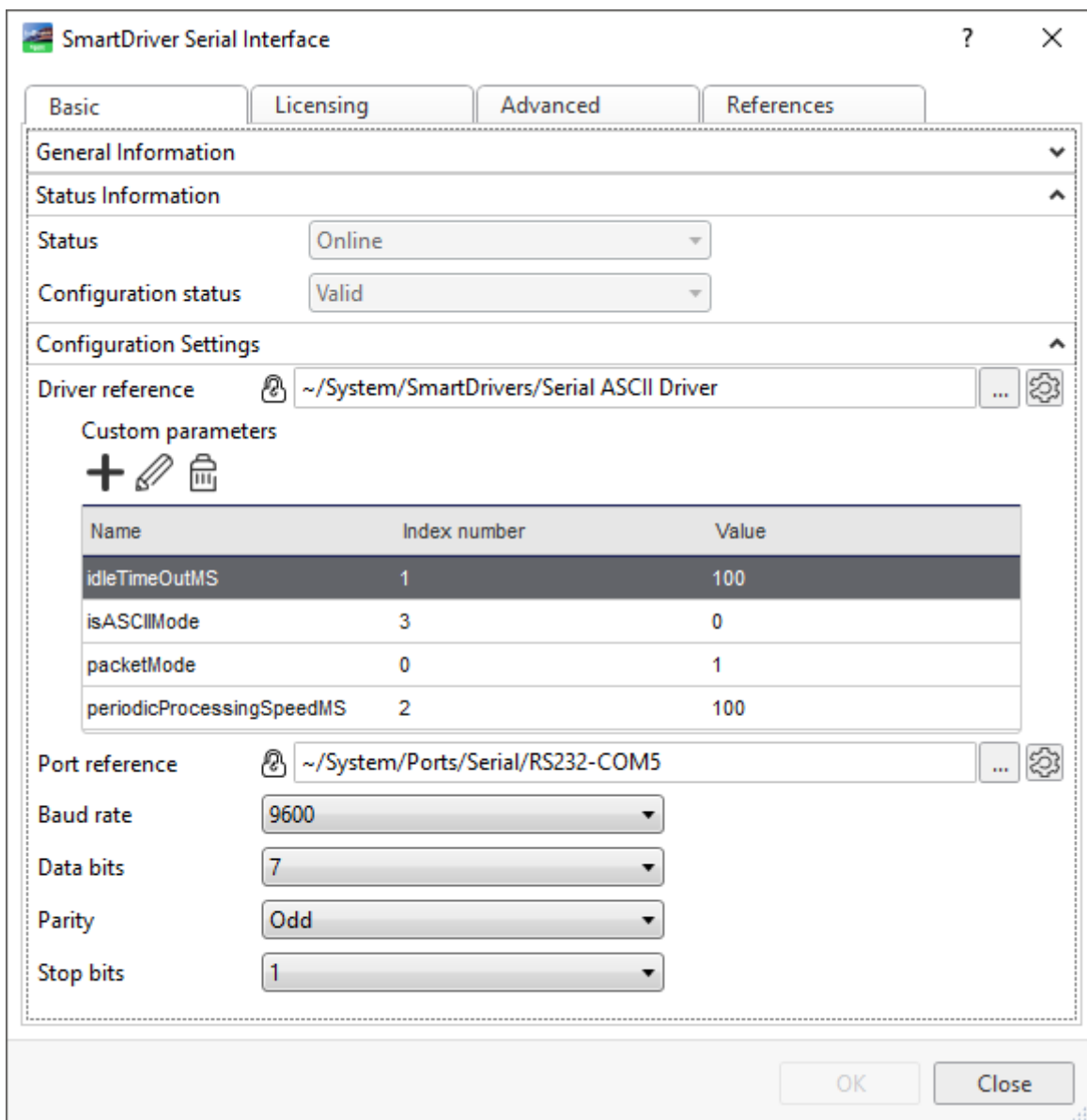



Fig a.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			

3.7.7 Is ASCII Mode Feature Usage

The driver has the ability to work in ASCII mode and HEX mode when reading and writing data. This functionality can be controlled by way of the customer parameter <**isASCII**Mode>.

The following Truth Table shows how the **isASCII**Mode impacts the bytes sent over the wire.


Tests performed below using AccessPort under Windows, the driver was installed using the 32bit build on EBO 2.

isASCII	Read via Input	Result	Rationale
1	0x61, 0x62, 0x63, 0x00, 0x64	abc d	All control chars are converted to blank space 20
1	0x61, 0x62, 0x63, 0x1F, 0x64	abc d	All control chars are converted to blank space 20
0 (hex mode)	0x61, 0x62, 0x63, 0x00, 0x64	6162630064	Presented as a 2 digit string in EBO Input Field. Control char preserved.
0 (hex mode)	0x61, 0x62, 0x63, 0x1F, 0x64	6162631F64	Presented as a 2 digit string in EBO Input Field. Control char preserved.

Fig a; EBO Input String Value

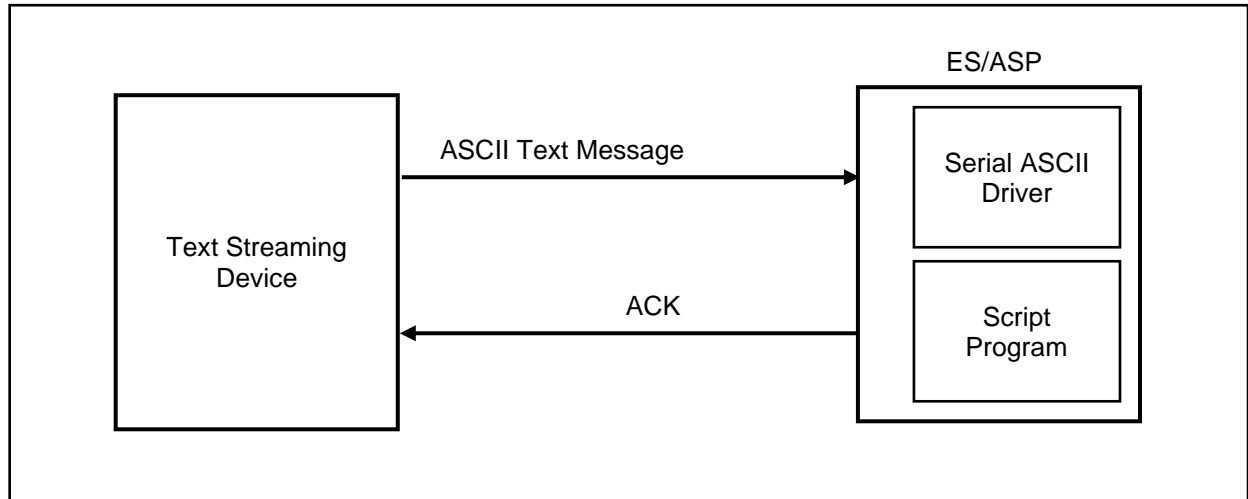
isASCII	Write via Output	Result	Rationale
1	abc	0x61, 0x62, 0x63	HEX shown on the wire under AccessPort
0 (hex mode)	6162630064	0x61, 0x62, 0x63, 0x00, 0x64	Control char preserved and passed to the wire
0 (hex mode)	6162631F64	0x61, 0x62, 0x63, 0x1F, 0x64	Control char preserved and passed to the wire. 1F in this instance

Fig b; EBO Output String Value

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			


4 Appendix A: Serial Text Message Handler

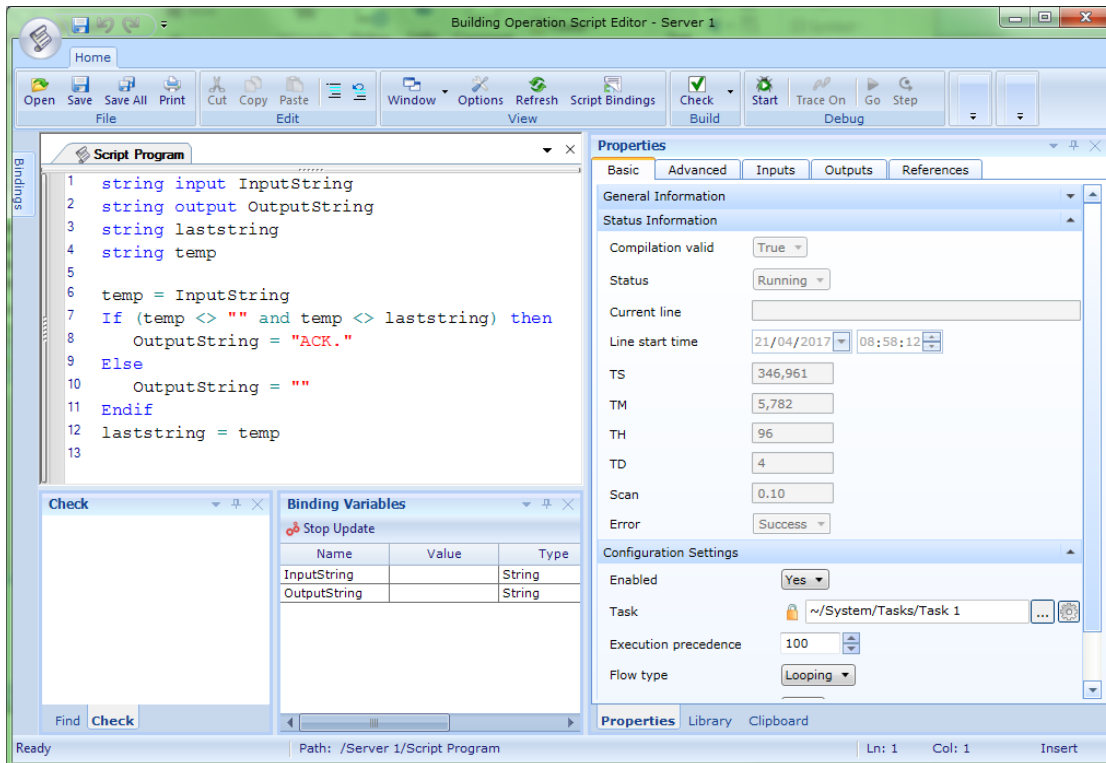
A sample application employing the serial ASCII driver is described here.



Here the ES/AS is required to collect ASCII text messages from its serial port. The message is sent in chunks and infrequently by, for example, the Text Streaming Device. Safe message reception has to be ensured before subsequent transmissions, for which the device expects a handshake mechanism from the server. The ES/AS is therefore required to respond with the acknowledgement string 'ACK' upon receiving each chunk of the text message.

Given below is the ES/AS Script Program that provides the specified functionality.

	GECoe Software Manual	Document Ref:	T100014RevM
		Author:	DC, JL & PR
		Date:	21-12-2020
			Rev 1.6.0
Schneider Electric Serial ASCII SmartDriver V1.6.0			



Here the variable InputString is bound to SmartDriver String Input Point and OutputString to SmartDriver String Output Point. The configuration property Task defines the frequency of execution of the Script Program.

For a continuous smooth operation of the Text Streaming Device, the ES/AS is required to promptly send the acknowledgement text in a timely manner. This is achieved by executing the serial ASCII driver function PeriodicProcessing at a slower rate than the Script Program. Configured to run at a faster rate, the Script Program will update the variable InputString more frequently than the rate at which the serial ASCII driver can export the received text message to the server. Also, the Script Program will promptly send the 'ACK' message to the serial ASCII driver once validated the variable InputString. The function PeriodicProcessing, will send the latest response to the serial port. In this example, the Script Program is set to run at 100 ms interval as defined by the property value ~/System/Tasks/Task 1. This is the fastest rate the ES/AS can run a task. The Periodic Processing Time value should therefore be set to 200 ms, or higher.