CODESYS API Driver Guide

Table of Contents

February 29, 2016

1. Seamless Integration with WebAccess

1.1 Introduction

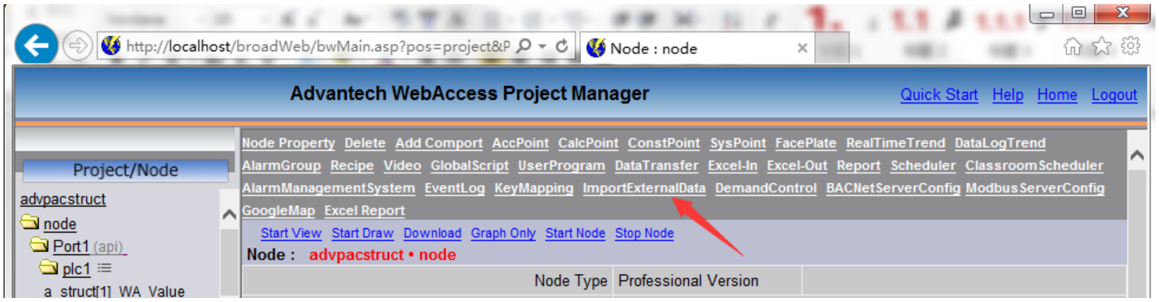
In order to be able to make CODESYS variables can be read and write in WebAccess, CODESYS use a PLC Handler API for seamless integration with WebAccess.

For CODESYS variables can be imported into WebAccess, WebAccess provides a Codesysxml.exe that will create a XML file which including all CODESYS variables, WebAccess will import this XML file to create tags.

The CODESYS RTE and CODESYS Gateway software is must be installed on the SCADA node (an adam5580). The WebAccess Project node can be on the same platform with SCADA node or not.

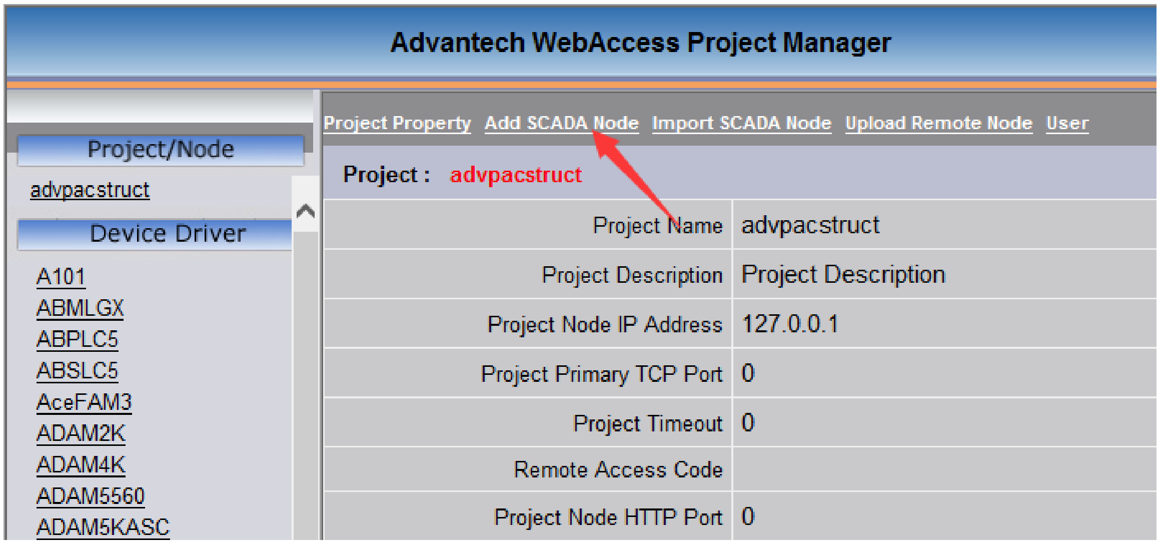
Driver use GatewayV3 mode and IP Address = 127.0.0.1 to communicate with CODESYS PLC, so a SCADA node can only add one device.

User only needs to create a project, add a SCADA node, import CODESYS variables and then download to run project.



The steps, in summary, are:

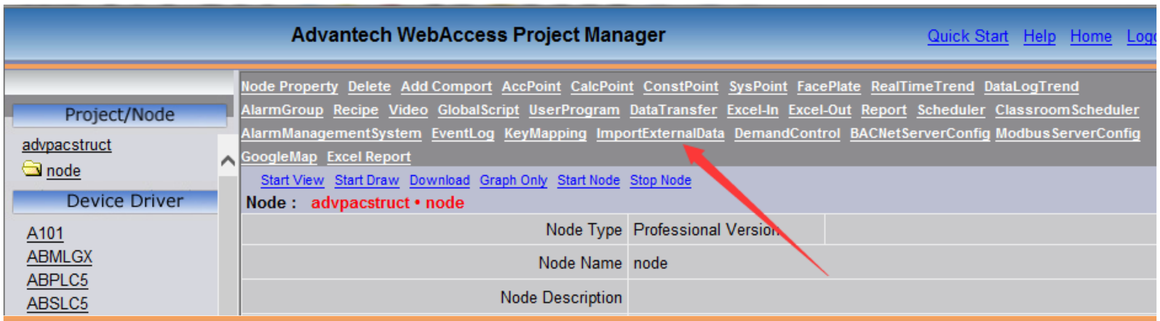
1. Start Internet Explorer **Web Browser**.
2. Enter IP address of the **Project Node**.
3. Use **WebAccess Configuration**.
4. Open or Create a **Projec**t.
5. Configure a **SCADA node**.


Advantech WebAccess Project Manager

Project : advpacstruct

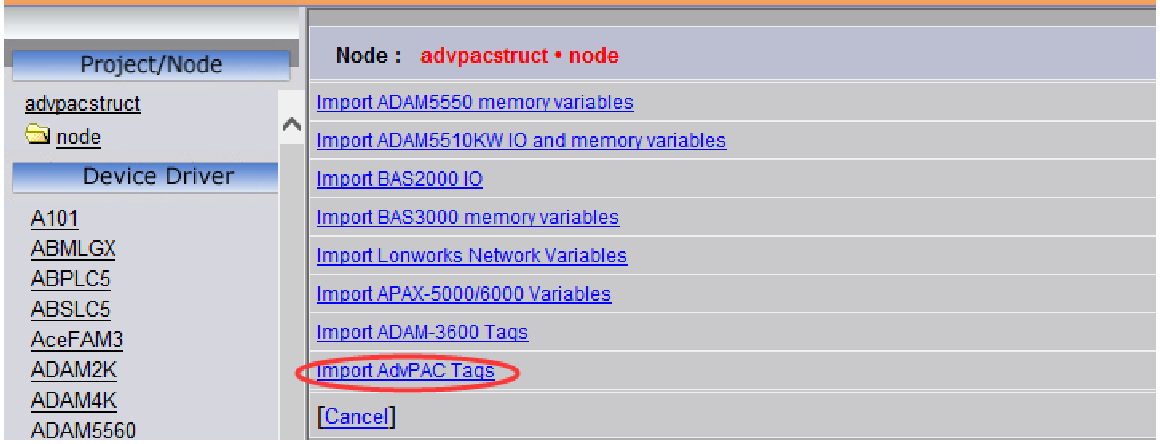
Project Name	advpacstruct
Project Description	Project Description
Project Node IP Address	127.0.0.1
Project Primary TCP Port	0
Project Timeout	0
Remote Access Code	
Project Node HTTP Port	0

6. Import Tags using Codesysxml.exe.


Advantech WebAccess Project Manager

Node : advpacstruct • node

Node Type	Professional Version
Node Name	node
Node Description	



Node : advpacstruct • node

Import ADAM5550 memory variables

Import ADAM5510KW IO and memory variables

Import BAS2000 IO

Import BAS3000 memory variables

Import Lonworks Network Variables

Import APAX-5000/6000 Variables

Import ADAM-3600 Tags

Import AdvPAC Tags

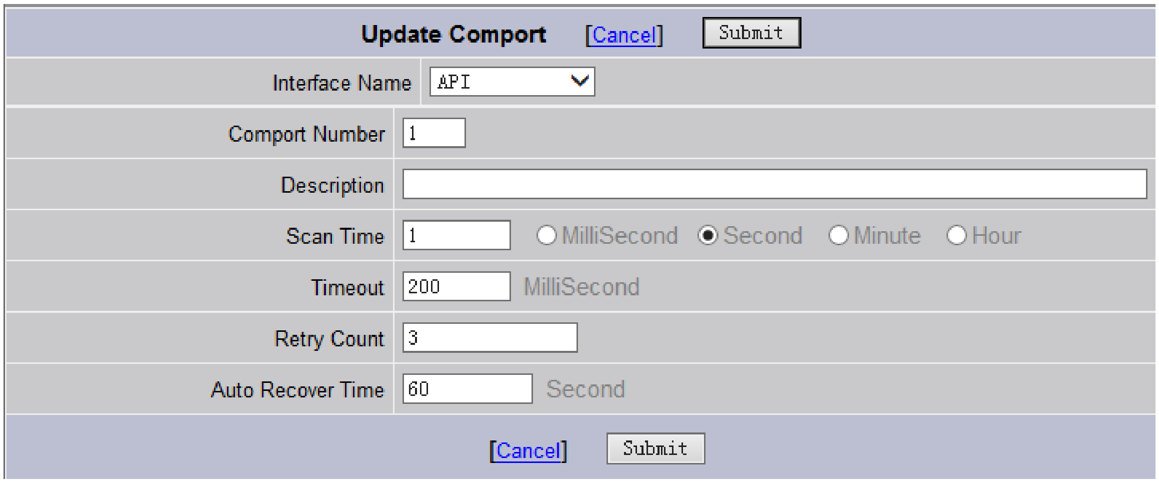
[Cancel]

7. Download project, then run.

1.2 Comport Properties

The Comport is associated with one CODESYS PLC.   If you have multiple CODESYS PLC, you must configure a SCADA node to each PLC. A comport can only add one device.

The port is automatically to be established when user importing tags. The default comport properties are as follow:



Update Comport [Cancel] Submit

Interface Name	API
Comport Number	1
Description	
Scan Time	1 ○ MilliSecond ● Second ○ Minute ○ Hour
Timeout	200 MilliSecond
Retry Count	3
Auto Recover Time	60 Second

[Cancel] Submit

1.2.1 Comport Number

In the comport properties, this is a Virtual number used for reference.   It can be any number. The only consideration might be to avoid conflicts with another device driver (i.e. a Serial Comport) which requires the comport number to match that of the physical interface.

1.2.2 Description

User defined field for reference. This appears only in Project Manager.

1.2.3 Scan Time

This is the time in seconds to scan the CODESYS PLC. This must match the ability of the PLC to respond.

1.2.4 Timeout

Timeout is the time waited before re-sending a communications packet that did not have a reply.

1.2.5 Retry Count

The number of times to retry communications if no reply is received from a device.

1.2.6 Auto Recover Time

Auto Recover Time is the Time to wait before attempting to re-establish communications with the CODESYS PLC.   It is recommended to accept the default of 60 seconds.   If communications to the CODESYS PLC is unusually slow due to software, communications or network issues, you might consider increasing this value.   If communications to the CODESYS PLC fails frequently, you may want to decrease this number in order to have WebAccess try to re-establish communications sooner.

If communications to the CODESYS PLC Fails, WebAccess will wait the Auto Recover Time before trying to re-establish communications. WebAccess will drop any "open handles" and re-establish new "software links" to the CODESYS PLC. WebAccess will try to restart the OPC Server if the OPC Server is not running.

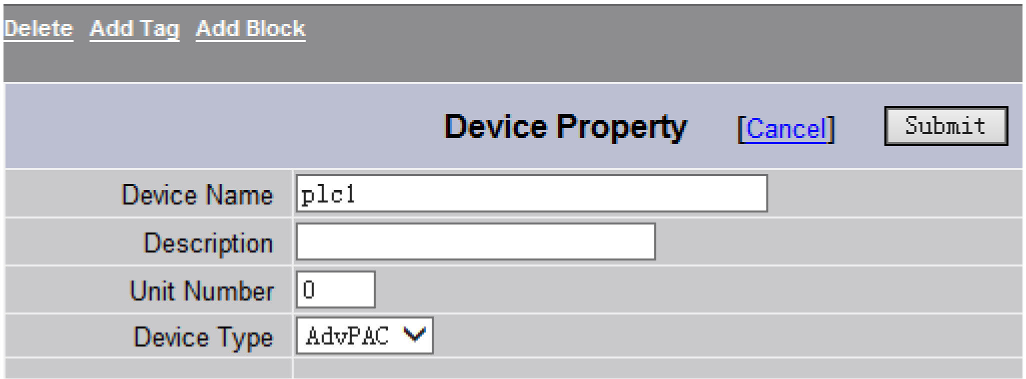
1.3 Device Properties

The Device on a CODESYS Comport is "virtual" device. You can use Device as an organizational tool (i.e. to represent process units, equipment, areas of a plant) or lump all the Tags into one Device.

The CODESYS Comport is associated with one CODESYS PLC. If you have multiple CODESYS PLC, you must configure a SCADA node to each PLC. The CODESYS Device represents a "virtual'" device. One Device is associated with one CODESYS PLC.

The CODESYS is software which installed on the SCADA node.

The port is automatically to be established when user importing tags. The default comport properties are as follow:



Device Property

Device Name	plc1
Description	
Unit Number	0
Device Type	AdvPAC

Device Name is any user-defined name.

Description is a user defined.

Unit Number, for most OPC interfaces, this is a "virtual number" and does not correspond to the actual Unit Number used in the protocol addressing.

The **Device Type** lists available Device Drivers for an API type Comport. If your device Type does not appear, you may have configured the wrong Com Port type.

1.4 Codesysxml.exe - Import Tags

The Codesysxml.exe is written to create a XML file which needed to import tags from CODESYS PLC into WebAccess. Codesysxml.exe must be put together with CODESYS Control RTE, the path is C:\WebAccess\Node.

1.4.1 Naming Rule

The CODESYS variable name is composed of three parts, those are application name, program name and tag name. WebAccess tag name is variable name's third part when imported. In order to avoid duplication of variable names between different programs, we make the following rules:

1. When there is one program in an application, WebAccess tag name is variable name's third part when imported.
2. When there are two or more programs in an application, starting from the second program the tag name prefixed with 1, and the third program of the tag name prefixed with 2, so on.

1.5 WebAccess Tag Error Code

1. *[ffff] indicates that driver failed to communicate with PLC/Gateway, the PLC/Gateway is in a STOP state.

2. *[error code1] indicates that in the process of communication PLC/Gateway is not running status, PLCHANDLER_STATE represents the status PLC/Gateway being now.

Error code1 = 0xb000 + PLCHANDLER_STATE

```
enum PLCHANDLER_STATE

{     STATE_TERMINATE              = -1,    // this state is shown as *[b0ff]

  STATE_PLC_NOT_CONNECTED    = 0,

  STATE_PLC_CONNECTED,

  STATE_NO_SYMBOLS,

  STATE_SYMBOLS_LOADED,

 STATE_RUNNING,

  STATE_DISCONNECT,

  STATE_NO_CONFIGURATION,

  STATE_PLC_NOT_CONNECTED_SYMBOLS_LOADED};
```

3. *[error code2] indicates that in the process of communication CODESYS project is not running status, PLC_STATUS represents the status the project being now.

Error code2= 0xc000 + PLC_STATUS

```
enum PLC_STATUS

{     PLC_STATE_RUNNNING  = 0,

  PLC_STATE_STOP              = 1,

  PLC_STATE_STOP_ON_BP     = 2,

  PLC_STATE_UNKNOWN   = 255};
```

4. *[d000] indicates that there is not this variable in CODESYS.

1.6 Datatype supported by driver

Tags which can only be read Simple data types Array data types User defined data types	SINT	8bit
	USINT	8bit
	INT	16bit
	UINT	16bit
	DINT	32bit
	UDINT	32bit
	LINT	64bit
	ULINT	64bit
	BOOL	8bit
	BYTE	8bit
	WORD	16bit
	DWOED	32bit
	LWORD	64bit
	REAL	32bit
	LREAL	64bit
	STRING	162bit
	WSTRING	81bit
	TIME	32bit
	LTIME	64bit
	TIME_OF_DAY	32bit
	DATE	32bit
	DATE_AND_TIME	32bit
Tags which can be read and write Simple data types Array data types User defined data types	SINT	8bit
	USINT	8bit
	INT	16bit
	UINT	16bit
	DINT	32bit
	UDINT	32bit
	LINT	64bit
	ULINT	64bit
	BOOL	8bit
	BYTE	8bit
	WORD	16bit
	DWOED	32bit
	LWORD	64bit
	REAL	32bit
	LREAL	64bit
	STRING	81bit
	WSTRING	162bit