

BECKHOFF AUTOMATION

TwinCAT3 ADS Provider

Version 1.0.2

User's guide

November 29, 2018

[Remarks]

Please use the provider in a secure network environment.

[Revision History]

Version	Date	Content
1.0.0	2013-10-04	First edition.
1.0.1	2016-06-13	Supported WSTRING type. Described other non-supported types.
1.0.1	2017-01-12	Attention described when reading the STRING/ STRING ARRAY type including null characters.
1.0.2	2018-11-02	Added 2.2.3. CaoController:: get_VariableNames.

[Hardware]

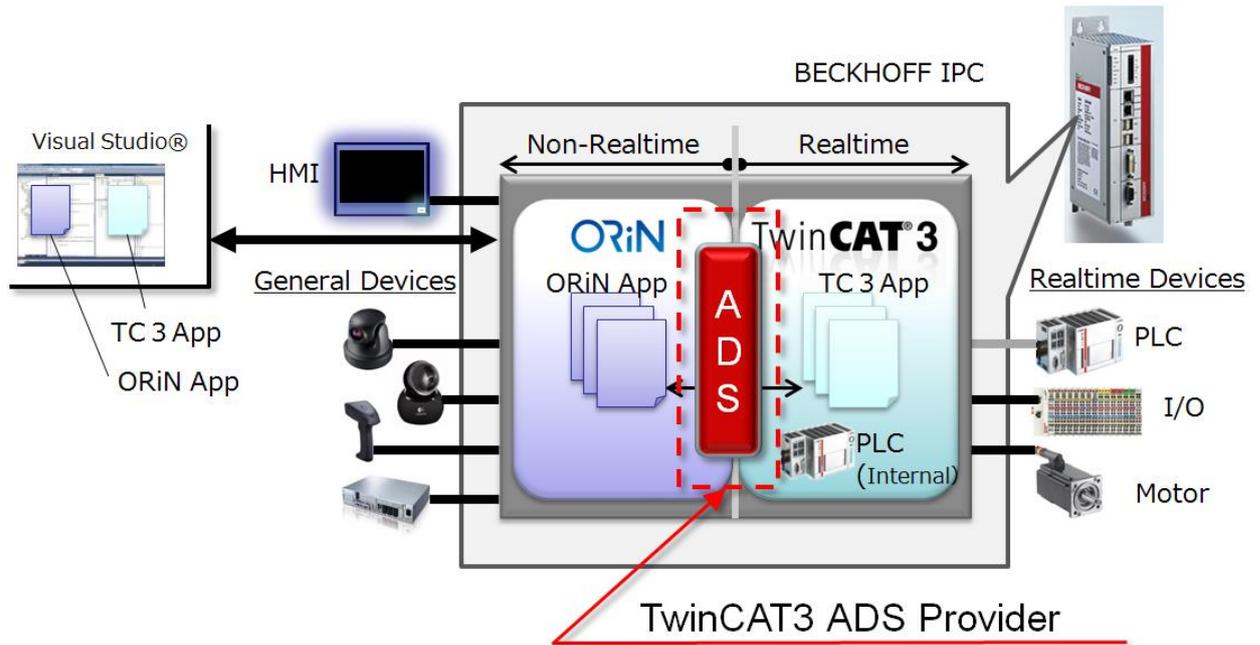
Model	Version	Notes

Contents

1. Introduction	4
2. Outline of this provider	5
2.1. Outline	5
2.2. Method and Property	6
2.2.1. CaoWorkspace::AddController method	6
2.2.2. CaoController::AddVariable method	7
2.2.3. CaoController:: get_VariableNames property	10
2.3. Variable list	11
2.3.1. Controller class	11
2.4. Error code	13
3. Sample program	15
Appendix A. Access to array type variables	16

1. Introduction

This is a CAO provider user's guide for utilizing ADS interface of TwinCAT3, which is manufactured by BECKHOFF AUTOMATION. CAO provider (CaoProvTC3ADS.dll) used in this guide is called TwinCAT3SDS provider. This provider works in the environment that installed ORiN2 and TwinCAT3 in IPC of BECKHOFF AUTOMATION.



The next section describes the outline of the TwinCAT3ADS provider and the command reference.

2. Outline of this provider

2.1. Outline

TwinCAT3ADS provider is a CAO provider which absorbs the dependence on ADS interface of BECKHOFF AUTOMATION TwinCAT3 and then provides functions specified by the CAO provider interface specification. The file format is DLL (Dynamic Link Library) and it is dynamically uploaded from the CAO engine. To use TwinCAT3ADS provider, you need to install ORiN2SDK, or, complete registration manually with the procedure shown below.

Table 2-1 TwinCAT3ADS provider

File name	CaoProvTC3ADS.dll
ProgID	CaoProv.BECKHOFF.ADS
Registry registration	regsvr32 CaoProvTC3ADS.dll
Remove registry registration	regsvr32 /u CaoProvTC3ADS.dll

TwinCAT3ADS provider requires ADS interface DLL module (TcAdsDll.dll) of TwinCAT3 system.

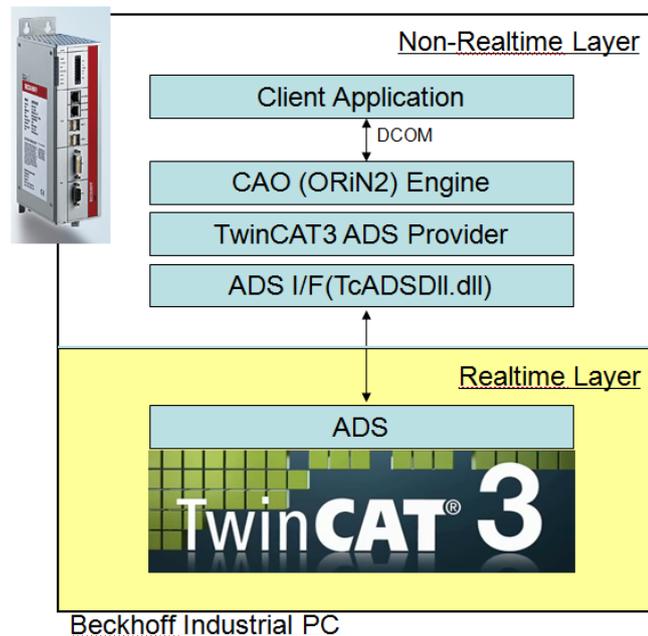


Figure 2-1 TwinCAT3 ADS provider's composition

2.2. Method and Property

2.2.1. CaoWorkspace::AddController method

At the timing of AddController execution, TwinCAT3ADS provider specifies parameters necessary for device connection based on the following format.

Syntax AddController(<bstrCtrlName:BSTR>, <bstrProvName>,
<bstrPCName:BSTR>, <bstrOption:BSTR>))

- <bstrCtrlName> : [in] Controller name (VT_BSTR)
Specify free format character strings.
- <bstrProvName> : [in] Provider name =ProgID (VT_BSTR)
Specify the fixed character string of "CaoProv.Beckhoff.ADS".
- <bstrPCName> : [in] Execution machine name (VT_BSTR)
Specify a computer name for the remote connection. Enter null ("") for the local connection.
- <bstrOption> : [in] Option string (VT_BSTR)
Specify the option character string to connect to the target ADS device. Each option item need to specify as the following format with comma separation.
<OptionName>=<Value>, <OptionName>=<Value>, ..
Ex. : "NetID=192.168.0.1.1.1, Port = 851"

Following is a list of <bstrOption> option string items.

Table 2-2 Option string of CaoWorkspace::AddController

Option	Description
NetID=X.X.X.X.X	Specify the value of AMS NetID. NetID is automatically set when it is omitted. For X part, enter numbers from 0 through 255. Use NetID when establishing remote connection against IPC. Ex. : "NetID=192.168.0.1.1.1"

<p>Port=<Number></p>	<p>Specify value of AMS port</p> <p>“851” is assigned when it is omitted. Port number corresponds with the ID number when the device in TwinCAT3 has been selected.</p> <p>Ex. : ” Port = 851”</p>

Example

Dim oCtrl As Object

Set oCtrl = caoWorkspace.AddController(“Ads”, ”CaoProv.Beckhoff.ADS”, ””, ”Port=851”)

2.2.2. CaoController::AddVariable method

AddVariable method of CaoController class is a method to access variables. In TwinCAT3ADS provider, assign system variable to variable name. For system variables mounted in the TwinCAT3ADS provider, refer to Table2-4.

Syntax

AddVariable(<bstrVariableName:VT_BSTR>[, <vntOption:VT_BSTR>])

<bstrVariableName> : [in] Variable name (VT_BSTR)

1. System variable = “@ any string”

Specify a system variable. See Table2-4.

2. User variable = any string whose first letter is not “@”

Specify the mapping for the variables defined by the device of TwinCAT3 (when Name option is specified) and for the memory area (when Add option is specified).

<bstrOption> : [in] Variable option (VT_BSTR)

Following options are available for user variables.

1. VT option

Specify the type of variable. Specify a value, numerical value or character string.

When VT option is omitted and Name option is specified, data type is automatically obtained by accessing ADS with the name. To specify Addr option, you need to specify VT.

VT=<Variable type> Ex.:VT=BOOL

or

VT=<VARTYPE value> Ex.:VT=11

Variable type	<i>VARTYPE(value)</i>	
"BOOL"	VT_BOOL(11)	<i>Boolean (True/False)</i>
"STRING"	VT_BSTR(8)	<i>string(ASCII)</i>
"BSTR"	VT_BSTR(8)	<i>string(ASCII)</i>
"WSTRING"	VT_BSTR(8)	<i>string(UNICODE)</i>
"I1"	VT_I1(16)	<i>1byte integer (signed)</i>
"SINT"	VT_I1(16)	<i>1byte integer (signed)</i>
"I2"	VT_I2(2)	<i>2-byte integer (signed)</i>
"INT"	VT_I2(2)	<i>2-byte integer (signed)</i>
"I4"	VT_I2(3)	<i>4-byte integer (signed)</i>
"DINT"	VT_I2(3)	<i>4-byte integer (signed)</i>
"I8"	VT_I2(20)	<i>8-byte integer (signed)</i>
"LINT"	VT_I2(20)	<i>8-byte integer (signed)</i>
"UI1"	VT_UI1(17)	<i>1byte integer (unsigned)</i>
"BYTE"	VT_UI1(17)	<i>1byte integer (unsigned)</i>
"USINT"	VT_UI1(17)	<i>1byte integer (unsigned)</i>
"UI2"	VT_UI2(18)	<i>2-byte integer (unsigned)</i>
"WORD"	VT_UI2(18)	<i>2-byte integer (unsigned)</i>
"UINT"	VT_UI2(18)	<i>2-byte integer (unsigned)</i>
"UI4"	VT_UI4(19)	<i>4-byte integer (unsigned)</i>
"DWORD"	VT_UI4(19)	<i>4-byte integer (unsigned)</i>
"UDINT"	VT_UI4(19)	<i>4-byte integer (unsigned)</i>
"UI8"	VT_UI8(21)	<i>8-byte integer (unsigned)</i>
"ULINT"	VT_UI8(21)	<i>8-byte integer (unsigned)</i>

"R4"	VT_R4(4)	4-byte real number
"REAL"	VT_R4(4)	4-byte real number
"R8"	VT_R8(5)	8-byte real number
"LREAL"	VT_R8(5)	8-byte real number

2. Name option

Option item to specify the ADS variable name as following format.

Name=<Program name>.<Variable name>

Ex.: Name=Main.bVar

```
'TwinCAT3declaration Main program bVar : BOOL
```

3. Addr option

Option item to specify the ADS variable pointed by index Group and index Offset as following format.

Addr=<Group>:<Offset>

Ex.: Addr=0x4021:80, VT=BOOL

```
'TwinCAT3 declaration
```

```
PROGRAM MAIN
```

```
VAR
```

```
bVar AT%MX10.0: BOOL;
```

```
END_VAR
```

※If neither Name nor Addr option are specified, "Name=<bstrVariableName>" will be specified. Variable names of ORiN can correspond with the variable of the device in TwinCAT3.

※ The following variable types are not supported.

LWORD, DATA, DATA_AND_TIME, DT,

TIME_OF_DAY, TOD, TIME, LTIME

※ There are some regulations to access array type variables.

For details, refer to 3.Appendix A.

Example

```
Dim oVar As Object
Set oVar = caoCtrl.AddVariable("Main.bVar", "")
Debug.Print oVar.Value
```

2.2.3. CaoController:: get_VariableNames property

Obtain variables described in 2.3.1.

2.3. Variable list

2.3.1. Controller class

Table 2-3 User variable list of Controller class

Variable identifier	Data type	Description	Attribute	
			get	put
<Program name>.<Variable name>	Depends on a target variable	Correspond a variable name defined by the device in TwinCAT3 with a ORiN variable name. Not specify Name.Addr option. Ex.: caoCtrl.AddVariable("Main.bVar", "")	√	√
Character strings other than the above	Depends on a target variable	Separate the variable name defined by the device in TwinCAT3 from the ORiN variable name. Be sure to specify either Name option or Addr option. Ex.: caoCtrl.AddVariable("aaa", "Name=Main.bVar") caoCtrl.AddVariable("bbb", "Addr=0x4021:80, VT=BOOL")	√	√

Table 2-4 System variable list of Controller class

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	Vender name = "Beckhoff Automation"	√	-
@VERSION	VT_BSTR	Version information of ADS module. <Version>.<Revision>.<Build> is returned.	√	-
@STATUS	VT_I4	Status of target ADS ADSSTATE_INVALID = 0 ADSSTATE_IDLE = 1 ADSSTATE_RESET = 2 ADSSTATE_INIT = 3 ADSSTATE_START = 4 ADSSTATE_RUN = 5 ADSSTATE_STOP = 6 ADSSTATE_SAVECFG = 7 ADSSTATE_LOADCFG = 8 ADSSTATE_POWERFAILURE = 9 ADSSTATE_POWERGOOD = 10 ADSSTATE_ERROR = 11 ADSSTATE_SHUTDOWN = 12 ADSSTATE_SUSPEND = 13 ADSSTATE_RESUME = 14 ADSSTATE_CONFIG = 15 ADSSTATE_RECONFIG = 16 ※Writing is possible by the value of a state.	√	√
@TIMEOUT	VT_I4	Communication timeout (milliseconds) of ADS connection Default:5000 ms	√	√

2.4. Error code

The table below lists the original error codes of TwinCAT3ADS provider.

For details about each error code, check the last four figures (16-bit) on the error code against the code listed in ADS DLL documents of TwinCAT3. (“XXXX” on the “0x8010XXXX” corresponds with the error code of ADS.)

For other ORiN2 common errors, refer to Error Code on the ORiN2 programming guide.

Error Macro	Error Number	Description
ADSERR_DEVICE_ERROR	0x80100700	Error class < device error >
ADSERR_DEVICE_SRVNOTSUPP	0x80100701	Service is not supported by server
ADSERR_DEVICE_INVALIDGRP	0x80100702	Invalid index Group
ADSERR_DEVICE_INVALIDOFFSET	0x80100703	Invalid index Offset
ADSERR_DEVICE_INVALIDACCESS	0x80100704	Reading/writing not permitted
ADSERR_DEVICE_INVALIDSIZE	0x80100705	Parameter size not correct
ADSERR_DEVICE_INVALIDDATA	0x80100706	Invalid parameter value(s)
ADSERR_DEVICE_NOTREADY	0x80100707	Device is not in a ready state
ADSERR_DEVICE_BUSY	0x80100708	Device is busy
ADSERR_DEVICE_INVALIDCONTEXT	0x80100709	Invalid context (must be InWindows)
ADSERR_DEVICE_NOMEMORY	0x8010070A	Out of memory
ADSERR_DEVICE_INVALIDPARM	0x8010070B	Invalid parameter value(s)
ADSERR_DEVICE_NOTFOUND	0x8010070C	Not found (files, ...)
ADSERR_DEVICE_SYNTAX	0x8010070D	Syntax error in command or file
ADSERR_DEVICE_INCOMPATIBLE	0x8010070E	Objects do not match
ADSERR_DEVICE_EXISTS	0x8010070F	Object already exists
ADSERR_DEVICE_SYMBOLNOTFOUND	0x80100710	Symbol not found
ADSERR_DEVICE_SYMBOLVERSIONINVALID	0x80100711	Symbol version invalid
ADSERR_DEVICE_INVALIDSTATE	0x80100712	Server is in invalid state
ADSERR_DEVICE_TRANSMODENOTSUPP	0x80100713	AdsTransMode not supported
ADSERR_DEVICE_NOTIFYHNDINVALID	0x80100714	Notification handle is invalid
ADSERR_DEVICE_CLIENTUNKNOWN	0x80100715	Notification client not registered
ADSERR_DEVICE_NOMOREHDLS	0x80100716	No more notification handles
ADSERR_DEVICE_INVALIDWATCHSIZE	0x80100717	Size for watch to big
ADSERR_DEVICE_NOTINIT	0x80100718	Device not initialized
ADSERR_DEVICE_TIMEOUT	0x80100719	Device has a timeout
ADSERR_DEVICE_NOINTERFACE	0x8010071A	Query interface failed
ADSERR_DEVICE_INVALIDINTERFACE	0x8010071B	Wrong interface required

ADSERR_DEVICE_INVALIDCLSID	0x8010071C	Class ID is invalid
ADSERR_DEVICE_INVALIDOBJID	0x8010071D	Object ID is invalid
ADSERR_DEVICE_PENDING	0x8010071E	Request is pending
ADSERR_DEVICE_ABORTED	0x8010071F	Request is aborted
ADSERR_DEVICE_WARNING	0x80100720	Signal warning
ADSERR_DEVICE_INVALIDARRAYIDX	0x80100721	Invalid array index
ADSERR_CLIENT_ERROR	0x80100740	Error class < client error >
ADSERR_CLIENT_INVALIDPARG	0x80100741	Invalid parameter at service call
ADSERR_CLIENT_LISTEMPTY	0x80100742	Polling list is empty
ADSERR_CLIENT_VARUSED	0x80100743	Var connection already in use
ADSERR_CLIENT_DUPLINVOKEID	0x80100744	Invoke id in use
ADSERR_CLIENT_SYNCSTIMEOUT	0x80100745	Timeout elapsed
ADSERR_CLIENT_W32ERROR	0x80100746	Error in win32 subsystem
ADSERR_CLIENT_TIMEOUTINVALID	0x80100747	Timeout is invalid
ADSERR_CLIENT_PORTNOTOPEN	0x80100748	Port is not opened
ADSERR_CLIENT_NOAMSADDR	0x80100749	No AMS address
ADSERR_CLIENT_SYNCINTERNAL	0x80100750	Internal error in ads sync
ADSERR_CLIENT_ADDHASH	0x80100751	Hash table overflow
ADSERR_CLIENT_REMOVEHASH	0x80100752	Key not found in hash table
ADSERR_CLIENT_NOMORESVM	0x80100753	No more symbols in cache
ADSERR_CLIENT_SYNCRESINVALID	0x80100754	Invalid response received
ADSERR_CLIENT_SYNCPORTLOCKED	0x80100755	Sync port is locked

3. Sample program

```
// How to access the Main.bVar variable of TwinCAT3
'TwinCAT3 declaration -----
PROGRAM MAIN
VAR
    bVar AT%MX10.0: BOOL;
END_VAR
```

List 3-1 Sample.frm

```
Private caoEng As CaoEngine
Private caoWS As CaoWorkspace
Private TwinCAT3ADSctrl As CaoController

Private Sub Form_Load()
    'Create object of CAO engine and CAO workspace
    Set caoEng = New CaoEngine
    Set caoWS = caoEng.Workspaces.Item(0)

    ' Connect to TwinCAT3 ADS device
    Set TwinCAT3ADSctrl = caoWS.AddController("Ads", "CaoProv.Beckhoff.ADS", "", "Port=851")

    'Create CAO variable object for "Main.bVar" of TwinCAT3 ADS
    Set TwinCAT3ADSbVar = TwinCAT3ADSctrl.AddVariable("Main.bVar", "")

    TwinCAT3ADSbVar.Value = True

    Debug.Print TwinCAT3ADSbVar.Value

    TwinCAT3ADSbVar.Value = False

    Debug.Print TwinCAT3ADSbVar.Value

End Sub
```

Appendix A. Access to array type variables

TwinCAT3ADS provider can access array type variables that are defined in TwinCAT3. (Some variable types have restrictions.) The following table shows which variable types can be accessed and remarks to access.

Table 3-1 Accessibility per variable types when each type is used as array elements

Variable type	Accessibility*		Remarks
	PUT	GET	
BOOL	Yes	Yes	
BYTE	Yes	Yes	
WORD	Yes	Yes	
DWORD	Yes	Yes	
LWORD	No	No	LWORD-type itself is not supported.
SINT	Yes	Yes	
USINT	Yes	Yes	
INT	Yes	Yes	
UINT	Yes	Yes	
DINT	Yes	Yes	
UDINT	Yes	Yes	
LINT	Yes	Yes	
ULINT	Yes	Yes	
REAL	Yes	Yes	
LREAL	Yes	Yes	
STRING	No	Yes	Support GET only.
WSTRING	No	Yes	Support GET only.
DATE	No	No	DATE-type itself is not supported.
DATE_AND_TIME	No	No	DATE_AND_TIME-type itself is not supported.
DT	No	No	DT-type itself is not supported.
TIME_OF_DAY	No	No	TIME_OF_DAY-type itself is not supported
TOD	No	No	TOD-type itself is not supported.
TIME	No	No	TIME-type itself is not supported.
LTIME	No	No	LTIME-type itself is not supported.

* Yes : Access available No: Access not available

```
PROGRAM MAIN
VAR (* Example of TwinCAT3 array variables declaration *)

(*BOOL*)
bArray: ARRAY[0..1] OF BOOL := [TRUE, FALSE];

(*BYTE*)
byArray: ARRAY[0..2] OF BYTE := [1, 2, 3];

(*WORD*)
wArray: ARRAY[0..2] OF WORD := [1, 2, 3];

(*DWORD*)
dwArray: ARRAY[0..2] OF DWORD := [1, 2, 3];

(*SINT*)
siArray: ARRAY[0..2] OF SINT := [-127, 0, 127];

(*USINT*)
usiArray: ARRAY[0..1] OF USINT := [0, 255];

(*INT*)
iArray: ARRAY[0..2] OF INT := [-32768, 0, 32767];

(*UINT*)
uiArray: ARRAY[0..1] OF UINT := [0, 65535];

(*DINT*)
diArray: ARRAY[0..2] OF DINT := [-134217728, 0, 134217727];

(*UDINT*)
udiArray: ARRAY[0..1] OF UDINT := [0, 4294967295];

(*LINT*)
liArray: ARRAY[0..2] OF LINT := [-9223372036854775807, 0, 9223372036854775807];

(*ULINT*)
uliArray: ARRAY[0..1] OF ULINT := [0, 18446744073709551615];

(*REAL*)
rArray: ARRAY[0..1] OF REAL := [1.234, 5.678];

(*LREAL*)
lrArray: ARRAY[0..1] OF LREAL := [1.234, 5.678];

(*STRING*)
sArray1: ARRAY[0..3] OF STRING := ['aaa', 'bbb', 'ccc'];
sArray2: ARRAY[0..3] OF STRING(2) := ['aa', 'bb', 'cc'];

(*WSTRING*)
wsArray1: ARRAY[0..3] OF WSTRING := ["あいうえお", "かきくけこ", "さしすせそ"];
wsArray2: ARRAY[0..3] OF WSTRING(2) := ["ああ", "いい", "うう"];

END_VAR
```

■ Size and the range of value per one element in array

If a variable is declared as an array element, it is prohibited to PUT more than the defined elements count. Also, before the access for PUT, TwinCAT3DS provider converts data from VARIANT type to each variable type. (When GET operation, data is obtained as array of supported-data type.)

For values and the available range for each data type, refer to “2.2.2” or BECKHOFF AUTOMATION TwinCAT3 manual.

■ Variables that treats character strings

If the data type is STRING type that treats ASCII characters or WSTRING type that treats UNICODE, the number of characters (buffer size secured in TwinCAT3) can be variable, depending on the way of variable declaration,

If STRING or WSTRING is simply declared as a data type, the area of 80 characters (80 + 1 bytes for STRING, 160 + 1 bytes for WSTRING) are secured. If STRING or WSTRING is declared with any type-definition, such as STRING(2) or WSTRING(2), it is regarded that the number of characters is specified. In this case, the area of two characters (2 + 1 bytes for STRING, 4 + 2 bytes for WSTRING) are secured.

If the character string whose size is less than the secured area size is PUT, the remaining area will be filled with NULL(0x00).

For about array of character string, PUT is not supported. This is because TwinCAT3ADS provider cannot identify the writing address of the second element since it is impossible to obtain the information of the number of character strings per one element from TwinCAT3.

For GET operation, only when the obtained data contains multiple data which consists of a NULL terminator and character string data, it is returned as array. The return data is divided by NULL terminators and is returned as an array data of (VT_ARRAY | VT_BSTR).

However, if the value of string variables is a null character, or if the element in its array strings, includes a null character, an attention is required because the read-out value cannot be read correctly.

Example 1) The execution result of “GET” of “sArray1” (see the sample program of the previous page).

(VT_ARRAY | VT_BSTR)

[0] VT_BSTR : aaa

[1] VT_BSTR : bbb

[2] VT_BSTR : ccc

Example 2) The execution result of “GET” of “wsArray1”(see the sample program of the previous page).

(VT_ARRAY | VT_BSTR)

[0] VT_BSTR : あいうえお

[1] VT_BSTR : かきくけこ

[2] VT_BSTR : さしすせそ