

APPLICATION EXAMPLE

AC500 V3 DIAGNOSIS IN THE IEC APPLICATION EXAMPLE

GET DIAGNOSIS FROM CPU AND FIELDBUS
DEVICES IN THE IEC APPLICATION PROGRAM



Contents

1 Disclaimer	3
2 Introduction	4
2.1 Scope of the document	4
2.2 Compatibility	4
2.3 Overview	4
3 Diagnosis in the IEC Code	5
3.1 Diagnosis library	5
3.2 Functionality of the program	6
3.3 Including the diagnosis functionality into an existing project	6
3.4 Missing TextListUtils library	8

1 Disclaimer

A. For customers domiciled outside Germany /

Für Kunden mit Sitz außerhalb Deutschlands

„Warranty, Liability:

The user shall be solely responsible for the use of this products described within this file. ABB shall be under no warranty whatsoever. ABB's liability in connection with application of the products or examples provided or the files included within this products, irrespective of the legal ground, shall be excluded. The exclusion of liability shall not apply in the case of intention or gross negligence. The present declaration shall be governed by and construed in accordance with the laws of Switzerland under exclusion of its conflict of laws rules and of the Vienna Convention on the International Sale of Goods (CISG)."

„Gewährleistung und Haftung:

Der Nutzer ist allein für die Verwendung des in diesem Dokument beschriebenen Produkte und beschriebenen Anwendungsbeispiele verantwortlich.

ABB unterliegt keiner Gewährleistung. Die Haftung von ABB im Zusammenhang mit diesem Anwendungsbeispiel oder den in dieser Datei enthaltenen Dateien - gleich aus welchem Rechtsgrund - ist ausgeschlossen. Dieser Ausschluss gilt nicht im Falle von Vorsatz oder grober Fahrlässigkeit. Diese Erklärung unterliegt Schweizer Recht unter Ausschluss der Verweisungsnormen und des UN-Kaufrechts (CISG)."

B. Nur für Kunden mit Sitz in Deutschland

„Gewährleistung und Haftung:

Die in diesem Dokument beschriebenen Anwendungsbeispiele oder enthaltenen Dateien beschreiben eine mögliche Anwendung der AC500 bzw. zeigen eine mögliche Einsatzart. Sie stellen nur Beispiele für Programmierungen dar, sind aber keine fertigen Lösungen. Eine Gewähr kann nicht übernommen werden.

Der Nutzer ist für die ordnungsgemäße, insbesondere vollständige und fehlerfreie Programmierung der Steuerungen selbst verantwortlich. Im Falle der teilweisen oder ganzen Übernahme der Programmierbeispiele können gegen ABB keine Ansprüche geltend gemacht werden.

Die Haftung von ABB, gleich aus welchem Rechtsgrund, im Zusammenhang mit den Anwendungsbeispielen oder den in dieser Datei enthaltenen Beschreibung wird ausgeschlossen. Der Haftungsausschluss gilt jedoch nicht in Fällen des Vorsatzes, der groben Fahrlässigkeit, bei Ansprüchen nach dem Produkthaftungsgesetz, im Falle der Verletzung des Lebens, des Körpers oder der Gesundheit oder bei schuldhafter Verletzung einer wesentlichen Vertragspflicht. Im Falle der Verletzung einer wesentlichen Vertragspflicht ist die Haftung jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht zugleich ein anderer der in Satz 2 dieses Unterabsatzes erwähnten Fälle gegeben ist. Eine Änderung der Beweislast zum Nachteil des Nutzers ist hiermit nicht verbunden.

Es gilt materielles deutsches Recht unter Ausschluss des UN-Kaufrechts."

2 Introduction

2.1 Scope of the document

This application example shows how to integrate and use the function blocks to get the diagnosis messages in the IEC601131 program of an AC500-V3 PLC

2.2 Compatibility

The application example explained in this document have been used with the below engineering system versions. They should also work with other versions, nevertheless some small adaptations may be necessary, for future versions.

- AC500 V3 PLC
- Automation Builder 2.4.1 or newer

2.3 Overview

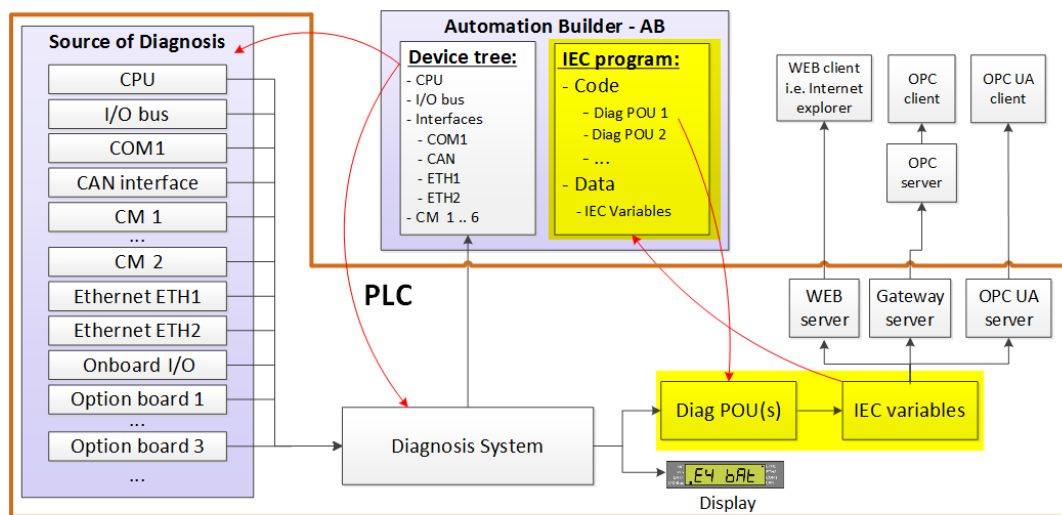


Fig. 1: Overview of the diagnosis system

3 Diagnosis in the IEC Code

3.1 Diagnosis library

For the diagnosis the 'Diag' library is used. With the help of this library and the methods of the Diag function block the diagnostic messages can be read into the IEC code.

The screenshot shows the 'Diag, 1.3.4.2 (ABB)' library in the project tree. The 'Diag' function block is expanded, showing sub-functions like 'Ack', 'Get', 'Val', 'ValAndTxt', 'Ext', 'GetFirstValAndTxt', 'GetLastValAndTxt', 'GetNextValAndTxt', 'GetPrevValAndTxt', 'Num', 'NumClass', and 'NumTotal'. The 'Documentation' tab for 'Diag.GetFirstValAndTxt (METH)' is active, showing the method name, its purpose, and its input/output parameters.

Diag.GetFirstValAndTxt (METH)

METHOD GetFirstValAndTxt : AC500_DiagTypes.ERROR_ID

Get first (oldest) diagnosis of all devices, numeric and textual

InOut:

Scope	Name	Type	Comment
Return	GetFirstValAndTxt	ACS00_DiagTypes.ERROR_ID	
Inout	DataVal	ACS00_DiagTypes.DIAG_VAL_TYPE	variable to write data to
Inout	DataTxt	ACS00_DiagTypes.DIAG_TXT_TYPE	variable to write text to

It is possible to get the values in a raw value format or a text format.

The screenshots show the 'DiagTypes, 1.2.4.1 (ABB)' library. The 'Types' folder is expanded, showing 'DIAG_VAL_TYPE' and 'DIAG_TXT_TYPE'. The 'Documentation' tab for 'DIAG_VAL_TYPE (STRUCT)' is active, showing its definition and input/output parameters.

DIAG_VAL_TYPE (STRUCT)

TYPE DIAG_VAL_TYPE : STRUCT

Diagnosis of a single node/device as plain numbers and enumerations

InOut:

Name	Type	Initial	Comment
dtTimestamp	DT	DATE_AND_TIME#1970-1-1-0:0	RTC time of event
uiMs	UINT	0	Milliseconds of event
eClass	teClass	teClass.eDiagClass_4_Warning	Severity of error event
szDevice	STRING(80)	''	Name of device
eEvent	teEvent	teEvent.eDiagEvent_Disappeared	Type of event
eHwInterfaceId	teHwId	teHwId.eDiagHwId_CPU	Identifier of hardware interface
dwSubSystemInfo	DWORD	0	Any number describing detail/location within device, device specific
dwAdditional	DWORD	0	Additional number describing detail/location within device, optional, device specific
udErrorCode	UDINT	0	Error code
uiSizeExtDiag	UINT	0	Number of bytes of extended diagnosis data
hSource	RTS_IEC_HANDLE	RTS_INVALID_HANDLE	Internal reference needed for text conversion
pConn	POINTER TO	0	Internal reference needed for text conversion

The screenshot also shows the 'DIAG_TXT_TYPE (STRUCT)' definition, which is a structure for diagnosis as interpreted text.

DIAG_TXT_TYPE (STRUCT)

TYPE DIAG_TXT_TYPE : STRUCT

Diagnosis as interpreted text

InOut:

Name	Type	Initial	Comment
szDiag	STRING(512)	''	Diagnosis entry as text
szColumn	STRING(1)	','	Character to be inserted as separator between columns

The IEC variables in the code can be visualized or shared to any other system.

Further information can be found in the library description or the online help.

The screenshot shows the 'Diagnosis in IEC application' section of the online help. It provides information on how to access diagnosis messages in the IEC application, including system diagnosis and device diagnosis. It also lists common data types (structures and enumerations) defined in the library ACS00_DiagTypes.

Diagnosis in IEC application

There are two possibilities for accessing the diagnosis messages in the IEC application:

- System diagnosis: Access to diagnosis messages of the whole PLC
- Device diagnosis: Access to the diagnosis messages of a device

For both possibilities common data types (structures and enumerations) are defined in the library ACS00_DiagTypes: "Data types in library ACS00_DiagTypes". The library is automatically included in PLC project.

Displayed text

For output of diagnosis messages in textual format Automation Builder and IEC application use text lists. Both application use the same text lists. The text lists are part of the device description. When inserting a new device in device tree of project, the corresponding text list is loaded. This text lists are part of PLC program and will be downloaded into the PLC.

The screenshot also shows a table of error messages (Err_10 to Err_23) with their descriptions and the 'Diag_V3_PLC' library.

3.2 Functionality of the program

The Program *Diagnosis* should be called by any slow task with a low priority. The *Diagnosis* program is reading all active diagnostic messages as values and text type. With the method *CreateVisuDiagStruct* the message in value and text format is stored into a structure.

The *Diagnosis* program is storing these messages into an array. This array *asErrVal* has a maximum size of 100 entries. This size can be changed. It is used for the visualization or can be shared to other systems.

Active diagnostic messages					
	Timestamp	Class	Device	Interface	Message
1	DT#2021-08-11-12:05:56	E4	CI502_IO	Coupler 1	Subslot 0x1,5500 Diagnosis, No process voltage UP or UP3 -> Check process voltage
2	DT#2021-08-11-12:05:56	E4	CI502_IO	Coupler 1	Subslot 0x1,5500 Diagnosis, Process voltage UP or UP3 too low -> Check process voltage
3	DT#2021-08-11-12:05:54	E4	CD522	I/O Bus	Process voltage switched off
4	DT#2021-08-11-12:05:54	E3	CD522	I/O Bus	Process voltage too low -> Check process voltage
5	DT#2021-08-11-12:05:44	E4	AX522	I/O Bus	Channel 4 Measurement overflow or not configured module

Further information can be found as comments inside the code.

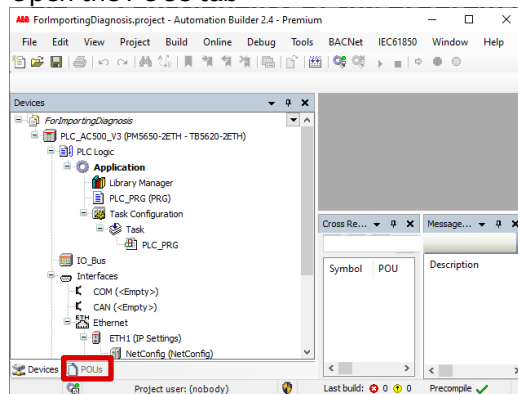
3.3 Including the diagnosis functionality into an existing project

The application example 'AC500_V3_Diagnosis' can also be used with other hardware. The export file which is also included in the package can be imported to existing applications.

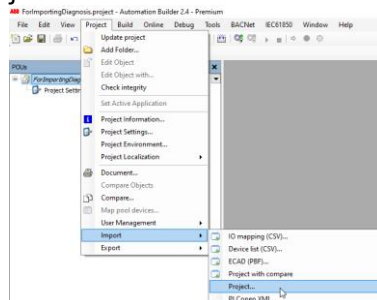
1. Open the project where the diagnosis functionality shall be included

It is possible to include the export file to the POU's or to any PLC Application. Here the import to the POU's is shown. In case of importing it to the application skip step 2 and select "Application" in step 3.

2. Open the POU's tab

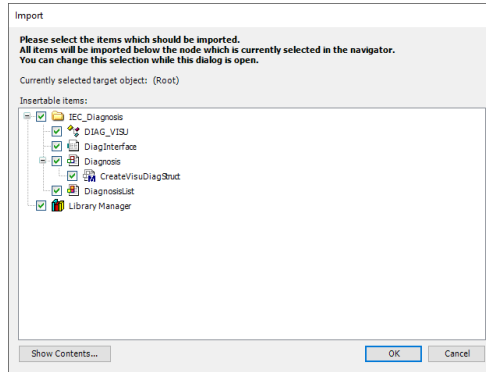


3. Select the project name here 'ForImportingDiagnosis' and click Project>Import>Project...



4. Open the AC500_V3_Diagnosis.export file

5. Click “OK”



6. Find the new objects in the POU's tree or below the Application

7. Change back to Devices

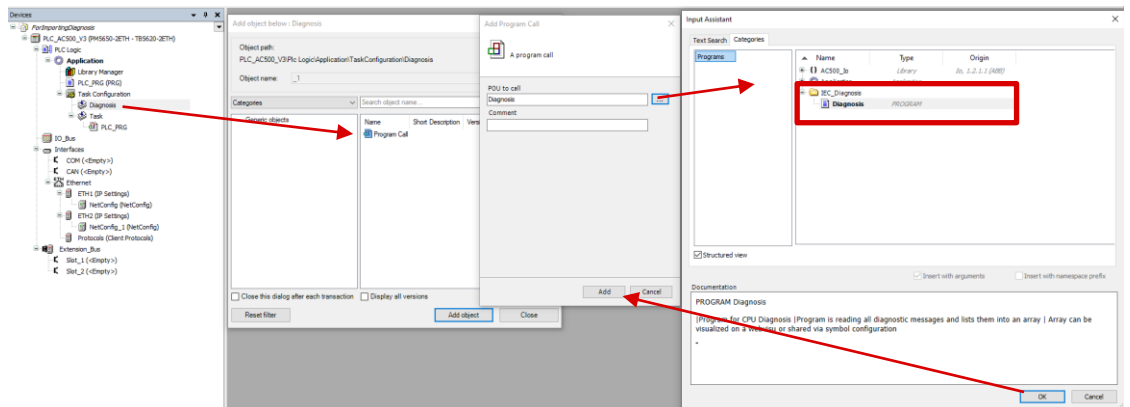


8. Right-click the Task Configuration and select 'Add object'

9. Add another task to the project

10. Add another object to the new task and select Program Call

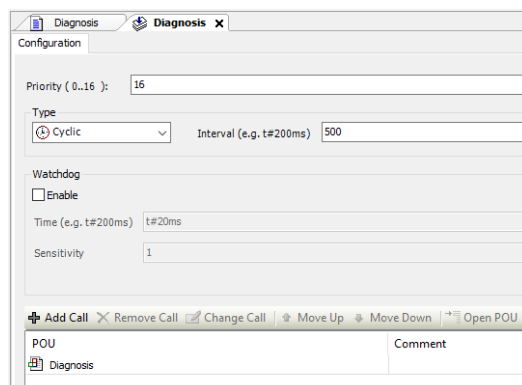
11. With the input assistant browse to 'Diagnosis'



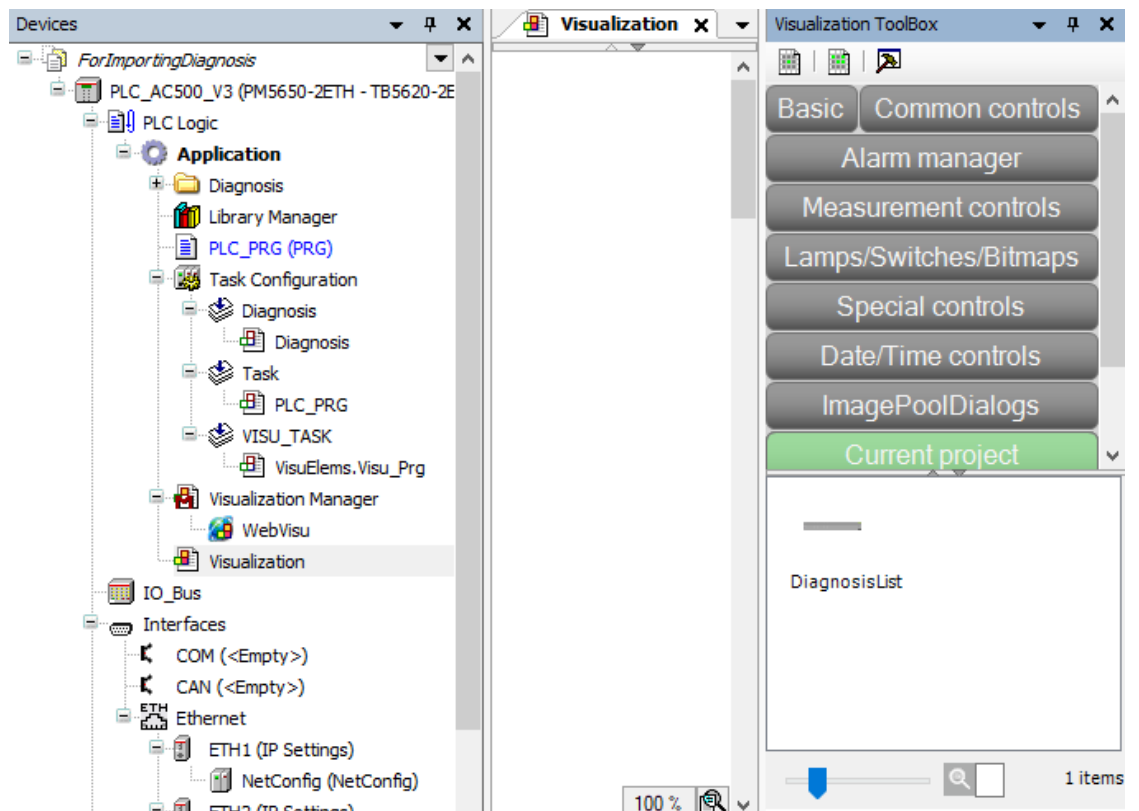
12. Confirm with 'OK' and 'Add'

13. Double click the new task here called 'Diagnosis'

14. Use a low priority for this task and a high cycle time. Disable the watchdog. Here Prio 16 and task cycle time 500ms is used



15. To visualize the Diagnosis messages, drag and drop the 'DiagnosisList' from the 'Current project' to any existing visualization



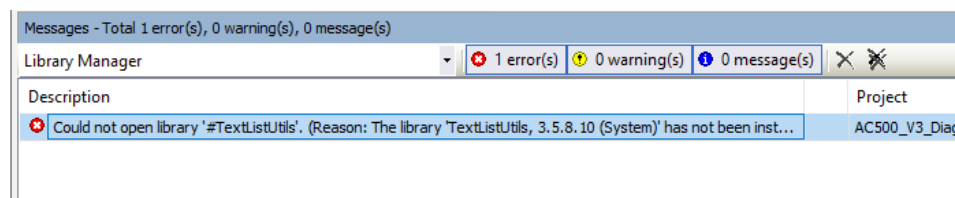
16. The project can now be compiled and downloaded including the diagnosis features. All active diagnostic messages will be displayed in the visu and are available in the program in the array 'asErrVal'

Active diagnostic messages

	Timestamp	Class	Device	Interface	Message
1	DT#2021-08-11-12:05:56	E4	CI502_IO	Coupler 1	Subslot 0x1,S500 Diagnosis, No process voltage UP or UP3 -> Check process voltage
2	DT#2021-08-11-12:05:56	E4	CI502_IO	Coupler 1	Subslot 0x1,S500 Diagnosis, Process voltage UP or UP3 too low -> Check process voltage
3	DT#2021-08-11-12:05:54	E4	CD522	I/O Bus	Process voltage switched off
4	DT#2021-08-11-12:05:54	E3	CD522	I/O Bus	Process voltage too low -> Check process voltage
5	DT#2021-08-11-12:05:44	E4	AX522	I/O Bus	Channel 4 Measurement overflow or not configured module

3.4 Missing TextListUtils library

Depending on the used Automation Builder installation it could be that the TextListUtils library is not installed to your library repository. If this is the case an error like shown below will be visible.



To solve it open the library manager. Depending on where you inserted the code the library manager in the Devices or POU's tree. The missing TextListUtils library can be installed by clicking "Download Missing Libraries". Then the library is downloaded and installed from the CODESYS store.

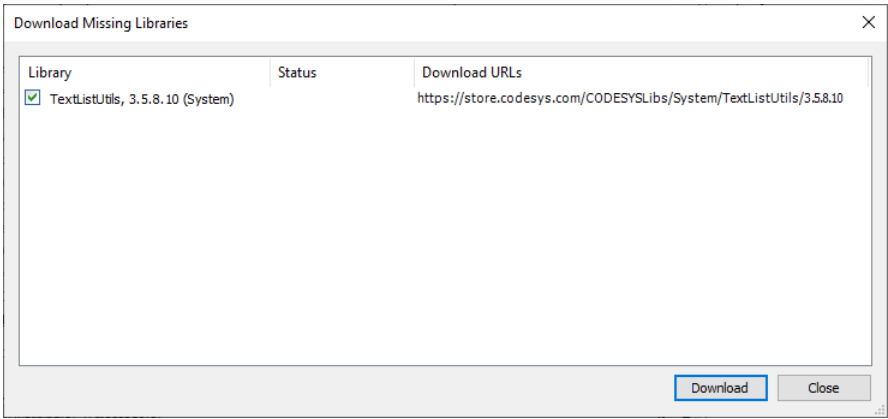
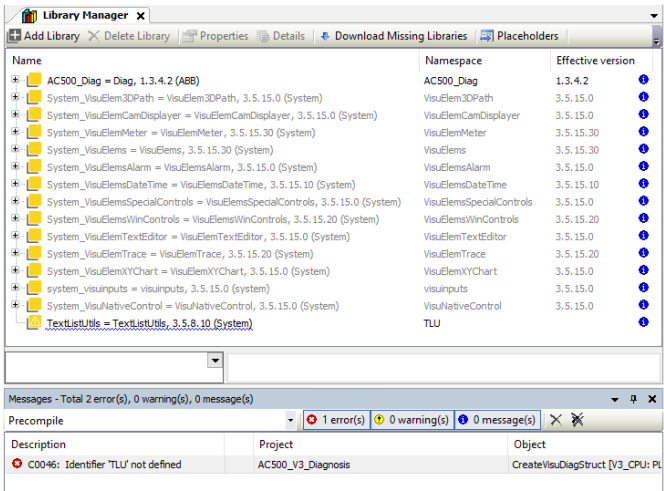


ABB Automation Products GmbH
Eppelheimer Straße 82
69123 Heidelberg, Germany
Phone: +49 62 21 701 1444
Fax: +49 62 21 701 1382
E-Mail: plc.support@de.abb.com
www.abb.com/plc

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB AG.
Copyright© 2021 ABB. All rights reserved