ENGLISH



Using OPC UA Application Defined Information Models -Anybus® CompactCom™ IIoT Secure

APPLICATION NOTE

SCM-1202-182 Version 3.1 Publication date 2022-04-08



Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Networks of any inaccuracies or omissions found in this document. HMS Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Networks and is subject to change without notice. HMS Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific applications, codes and standards. Further, HMS Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

Copyright © 2021

Box 4126 300 04 Halmstad, Sweden

info@hms.se

Table of Contents

1. Using OPC UA Application Defined Information Models	1
1.1. Anybus OPC UA NodeSet Encoder	2
1.2. From Idea to Running Application	3
2. Siemens OPC UA Modeling Editor Example	5
3. Related Documents and Tools	12

This page is intentionally left blank.

1. Using OPC UA Application Defined Information Models

By default, the information model of the CompactCom IIoT module defines a CompactCom40DeviceType based on the OPC UA for devices DeviceType and presents asset information (product name, serial number, software version) and parameters (Application Data Instances, ADIs).

The default information model can be replaced by an application-defined information model created in an OPC UA modeler tool. The application-defined information model can be completely custom or based on existing companion specifications.

The information model is defined by a Nodeset2 XML file that can be generated by an OPC UA Modeler Tool. The finished Nodeset2 file is converted by a tool from HMS, 'Anybus OPC UA NodeSet Encoder' to a binary file that is downloaded to the file system of the Anybus CompactCom 40 IIoT Secure.

Variable nodes must be tied to ADIs in the host application. A separate namespace must be created for those nodes.

- OPC Unified Architecture Specifications can be downloaded here: https://opcfoundation.org/developer-tools/specifications-unified-architecture
- Existing Companion Specifications can be downloaded here: https://opcfoundation.org/developer-tools/specifications-opc-ua-information-models
- Information Model Nodesets based on different Companion Specifications can be downloaded here: https://github.com/OPCFoundation/UA-Nodeset
- UA Modeling Best Practices document: https://opcfoundation.org/wp-content/uploads/2020/09/OPC-11030-Whitepaper-UA-Modeling-Best-Practices-1.00.00.pdf

1.1. Anybus OPC UA NodeSet Encoder

The Anybus OPC UA NodeSet Encoder for Windows is a command-line tool used to convert the NodeSet2 XML file generated by the OPC UA Modeler Tool to a binary file used by the Anybus CompactCom 40 IIoT Secure.

The Anybus OPC UA NodeSet Encoder can also be used to simulate a server with the developed information model before downloading it to the Anybus CompactCom 40 IIoT Secure module.

The Anybus OPC UA NodeSet Encoder can be downloaded from the support pages for the Anybus CompactCom 40 IIoT Secure modules. https://anybus.com/support.



Figure 1. Command line options:

Example (generate files):

OpcUaNodesetEncoder.exe -b Opc.Ua.Robotics.NodeSet2.xml MyCustomRobotNodeSet.xml

Example (simulate OPC UA server):

```
OpcUaNodesetEncoder.exe -s 1 -p 4840 Opc.Ua.Robotics.NodeSet2.xml
MyCustomRobotNodeSet.xml
```



NOTE

The mandatory namespace 0 (Opc.Ua.Di.NodeSet2.xml) will always be added automatically by the Anybus OPC UA NodeSet Encoder, and does not need to be explicitly added as an input.

1.2. From Idea to Running Application



Figure 2. Overview

 Model the device in an OPC UA Modeler Tool of your choice. The model can be based on Existing UA Nodesets, imported into the modeler tool. Variable nodes must be moved to a separate NameSpace named 'urn:compactcom40:parameters', and the node ID must be built up according to the following:

Table 1. Node ID Encoding

Node ID	Descript	ion						
0x01000000 -	The node	e IDs have the following encoding: 0x01MMMMNN						
0x01FFFFFF	• MMM	MMMM = ADI number						
	• NN = I	NN = ID of a node that models certain information of an ADI						
	Table 2. I	e 2. NN Encoding						
	NN:	Description						
	0x00	Value variable nodes of the ADIs						
	0x01	Max value variable nodes of the ADIs						
	0x02							
	0x03	0x03 Default value variable nodes of the ADIs						
	0x04	OptionSetValues property of OptionSetType (only exists for ADIs with BITS or BITx data types)						
	0x05	EnumStrings property of MultiStateDiscreteType (only exists for ADIs with ENUM data type)						

- 2. Export the information model as a NodeSet2 XML file.
- 3. Run the 'Anybus OPC UA NodeSet Encoder' in a command window.



NOTE

If all nodesets used when modelling the information model are not included in the exported file, the included nodesets must be added as inputs when running the encoder. The files must be added in order, with the nodeset highest in the hierarchy first. See example below.

Example (generate files):

```
OpcUaNodesetEncoder.exe -b 1 Opc.Ua.Robotics.NodeSet2.xml
MyCustomRobotNodeSet.xml
```

- 4. A binary file is generated by the Command Line Tool; binarynodeset.hiff.
- In this step, the command line tool can be used to simulate the information model without downloading it to the Anybus CompactCom. Use the -s option to start the simulation.
 Example (simulate OPC UA server):

```
OpcUaNodesetEncoder.exe -s 1 -p 4840
Opc.Ua.Robotics.NodeSet2.xml MyCustomRobotNodeSet.xml
```

- 6. Download the binary file (binarynodeset.hiff) to the Anybus file system root by using WebDAV or directly from the host application by using the Anybus File System Interface Object (0Ah). The user account used when downloading the file via WebDAV must have access to the Anybus file system root. During development, admin mode can be convenient to grant access for all users to the Anybus file system root. Set attribute #7 in the Ethernet host object (F9h) to TRUE to enable admin mode.
- 7. Include the corresponding ADI list in the host application source code.
- 8. Reset and re-initialize the Anybus CompactCom 40 IIoT Secure. For the OPC UA server to be enabled with the downloaded nodeset, the OPC UA model, attribute #1 in the OPC UA Host Object (E3h), must be set to 2.
- 9. The application-defined information model is now running, and a Client can connect and exchange data with the host application.

2. Siemens OPC UA Modeling Editor Example

In this example, which is a simple example of the workflow and nowhere near a complete information model, the OPC UA for Robotics NodeSet is used, and a Motion Device System is created. The example is concentrating on the motor in a power train. One node is kept static (Manufacturer) and the value is set directly in the information model. Two nodes are variable (MotorTemperature and SerialNumber) and are connected to ADIs in the Anybus CompactCom host application.

2:ComponentType PropertyType: 2:AssetId
MotorType
PropertyType: 2:SerialNumber
PropertyType: 2:Manufacturer
PropertyType: 2:Model
PropertyType: 2:ProductCode
+ 2:ParameterSet
BaseDataVariableType: BrakeReleased
AnalogUnitType: MotorTemperature
BaseDataVanableType: EffectiveLoadRate

Companion specification: OPC 40010-1 - UA Companion Specification Part 1 for Robotics 1.00

Start the Siemens OPC UA Modeling Editor and follow the workflow below.

1. Import NodeSets.

First, all related nodesets must be imported. Click the 'Import XML' button 🕒 and browse to the required nodesets. For the Robotics nodeset the following XML-files must be imported:

- Opc.Ua.Di.NodeSet2.xml
- Opc.Ua.Robotics.NodeSet2.xml

2. Add a new NameSpace for your InformationModel. Click the drop-down with the text 'Please add or choose a namespace'.

Vamespaces:	Editing (curren	nt ざ Ad	d New Name	space			
ncfoundation org		🔒 1: l	http://opcfou	ndation.org/	UA/DI/		
,percanation.org,	Chy .	2 :1	http://opcfou	ndation.org/	UA/Rob	otics/	
		nume	opueco	-			
∆dd Nan	IECHARE						
Add Nan	lespace						
Add Nan	lespace						
Add Nan		1					
Add Nan	NameSpaceURI	1					
Add Nan	NameSpaceURI	1					
Add Nan Namespace URI * http://MyNew Version 1.00	NameSpaceURI	(
Add Nan Namespace URI * http://MyNew Version 1.00	NameSpaceUR	1					
Add Nan Namespace URI * http://MyNew Version 1.00 PublicationDate		1					
Add Nan Namespace URI * http://MyNew Version 1.00 PublicationDate 2021-05-26T0	NameSpaceURI	1					
Add Nan Namespace URI * http://MyNew/ Version 1.00 PublicationDate 2021-05-26T0	NameSpaceURI	1		Ē			
Add Nan Namespace URI * http://MyNew/ Version 1.00 PublicationDate 2021-05-26T0	NameSpaceURI	1		ē			

Add a new NameSpace for variable nodes to be connected to ADIs in the host application. Click the 3. drop-down with the text 'Please add or choose a namespace'.

Namespaces: Editing (curren	🛉 📑 Add New Namespace	
	1: http://opcfoundation.org/UA/DI/	
	2: http://opcfoundation.org/UA/Robotics/	
Add Namespace	3: http://MyNewNameSpaceURI	
Add Namespace	3: http://MyNewNameSpaceURI	
Add Namespace	3: http://MyNewNameSpaceURI	
Add Namespace	a: http://MyNewNameSpaceURI	
Add Namespace Namespace URI * urn:compactcom40:paramet Version 1.00	ers	

4. Create a Motion Device System in the http://MyNewNameSpaceURI namespace. A new instance is added by right-clicking 'DeviceSet' and selecting 'Add Instance'.

No filter OPE Root OPE Objects Move Node OnlineAcce OnlineAcce OnlineAcce Ope Types Add Interface AddIn OPE DataTypes Ope ObjectTypes Nore ColjectTypes Nore ColjectTypes Nore ColjectTypes Nore ColjectTypes Nore ReferenceType	monnation model	http://operodildation.org/op/	
Image: Provide and the set of the s	No details	▼ No filter	
▼ OPC Objects ▲ Aliases ▼ OPViceSet ● DeviceFeat ● DeviceTopoloc ▲ dd Instance □ OnlineAcce □ OnlineAcce ● NetworkSet ▶ OPC DataTypes ▶ OPC DisceTypes ▶ OPC ObjectTypes ▶ OPC ObjectTypes ▶ OPC ReferenceTyp	OPC Root		
 Aliases DeviceSet DeviceFeat Move Node Add Instance OnlineAcce NetworkSet Server Add Interface Add Interface Add Interface AddIn Change Access Level Core InterfaceType Ore ObjectTypes Ore CobjectTypes Ore ReferenceTyp 	Objects		
 DeviceSet DeviceFeat Move Node DeviceTopoloc Add Instance OnlineAcce NetworkSet Server Add Interface Add Interface Add Interface Change Access Level Der EventTypes Der Dipec DipectTypes Der ObjectTypes Bookmarks 	🕨 🥡 Aliases		
DeviceFeat Move Node Add Instance OnlineAcce OnlineAcce NetworkSet Server Server Occ Types Occ DataTypes Occ DistecTypes Occ Change Access Level Move to namespace (recursive) Occ ObjectTypes Occ ReferenceTyp	🔻 🨝 DeviceSet		
 DeviceTopolos OnlineAcce Delete Node Delete Node Delete Node Add Interface Add Interface Add Interface Add Interface Change Access Level Dec InterfaceType Move to namespace (recursive) Bookmarks 	📦 DeviceFeat	Move Node	
 OnlineAcce NetworkSet Server Add Interface AddIn Opc DataTypes Opc LinterfaceType Opc ObjectTypes Bookmarks 	🔻 🧊 DeviceTopolo <u>c</u>	Add Instance	
 NetworkSet Server Add Interface AddIn Opc DataTypes Opc EventTypes Opc InterfaceType Opc ObjectTypes Bookmarks 	📑 OnlineAcce	Delate Made	
Server Add Interface Add Interface Add Interface Add Interface Add In Change Access Level Change Access Level Ore CobjectTypes Ore ObjectTypes Ore ReferenceTyp	🔋 NetworkSet	Delete Node	
 Opec Types Opec DataTypes Opec DataTypes Change Access Level Change Access Level Move to namespace (recursive) Opec ObjectTypes Bookmarks 	🕨 🧊 Server	Add Interface	
Ore DataTypes Ore EventTypes Ore InterfaceType Ore ObjectTypes Ope ReferenceTyp	▼ OP¢ Types	Addin	⊧
EventTypes Change Access Level Move to namespace (recursive) Opc ObjectTypes Dopc ReferenceTyp	DataTypes		
InterfaceType Move to namespace (recursive) Move to namespace (recursive) Bookmarks Move to namespace (recursive)	DPC EventTypes	Change Access Level	
Ope ObjectTypes Bookmarks Ope ReferenceTyp	opc InterfaceType	Move to namespace (recursive)	Þ
OPC ReferenceTyp	ObjectTypes	Bookmarks	▶
	opc ReferenceTyp		

Add a motion system instance with a MotionDeviceSystemType. Under MotionDevices->PowerTrains, add a power train with a PowerTrainType. Now an instance with the MotorType is available, and the variables can be configured.

Now, the model will look something like this:

Information model http://MyNewNameSpaceURI -
No details 🔹 No filter 💌
🔻 📴 Root 🗠
▼ OP¢ Objects
Aliases
🔻 🍘 DeviceSet
DeviceFeatures
Motion_System_1
Controllers
▼ OP¢ MotionDevices
Motion_Device_1
OPC Axes
Manufacturer
OPC PowerTrains
▼ 📦 Power_Train_1
▼ 😝 Motor_1
🛅 Manufacturer
🛅 Model
🔻 🥡 ParameterSet
MotorTemperature
ProductCode
🛅 SerialNumber
ProductCode
🔲 SerialNumber
OPC SafetyStates

5. Configure the ProductCode with a static value ('SuperMotor' in this example).

MotionDevices
Motion_Device_1
OPC Axes
Manufacturer
🗐 Model
MotionDeviceCategory
🕨 🥡 ParameterSet
PowerTrains
🔻 🥡 Power_Train_1
🔻 🧯 Motor_1
🗐 Manufacturer
🗐 Model
🔻 🥡 ParameterSet
🕨 📶 MotorTemperature
ProductCode
🗐 SerialNumber

Value	SuperMotor
Value	Supermotor
DataType	String
ValueRank	Scalar
ArrayDimensions	0
AccessLevel	1
UserAccessLevel	1
MinimumSamplingInterval	0
Historizina	false

 Connect the MotorTemperature node to an ADI in the host application. Change namespace by right-clicking and select 'Move to namespace' and select the 'urn:compactcom40:parameters' namespace.

PowerTrains			÷
🔻 尊 Power_Train_1			÷
🔻 🧊 Motor_1			÷
🔟 Manufactu	irer (LocalizedText)		÷
🔟 Model (Lo	calizedText)		÷
🔻 🈝 Parameter	Set		÷
Motor	emperature (Double)		÷
ProductCo	Move Node		÷
🛄 SerialNum	Delete Node		÷
ProductCode (String	Delete Hode		÷
💼 SerialNumber (Strin	Add New Variable		>
OP¢ SafetyStates	Move to namespace (recursive) 🕨	0: http://opcfou	undation.org/UA/
eviceTopology	Bookmarks	1: http://opcfou	undation.org/UA/DI/
model online			
		2: http://opcfoi	undation.org/UA/Robotics/
		4: urn:compact	tcom40:parameters

Enter the Node ID.

OPC UA Attributes		
▼ Nodeld	ns=4;i=16802816	
identifierType	NUMERIC	•
value	16802816	
namespace	4: urn:compactcom40:parameters	•

NodeID for the value of ADI number 100, 0x01006400 = 16802816.

 Connect the SerialNumber node to an ADI in the host application. Change Namespace by right clicking and select 'Move to namespace' and select the 'urn:compactcom40:parameters' namespace.

PowerTrains				→
🔻 🧊 Power_Train	_1			÷
🔻 🏮 Motor_1				÷
💷 Manu	facturer (LocalizedText)			÷
💷 Mode	l (LocalizedText)			÷
🔻 🃦 Paran	neterSet			÷
🔻 📶 Mo	otorTemperature (Double)			÷
	EngineeringUnits (EUInformation	n)		÷
🛄 Produ	ctCode (String)			→
🔳 Seria				÷
ProductCode ()	Move Node			÷
🛅 SerialNumber	Delete Node			÷
SafetyStates	Move to namespace (recursive) 🕨	0: http://opcfoun	dation.org/UA/
eTopology orkSet	Bookmarks	÷	1: http://opcfoun	dation.org/UA/DI/
		-	2: http://opcfoun	dation.org/UA/Robotics/
			4: urn:compactco	om40:parameters
OPC UA Attri	butes			
▼ Nodeld		ns=4	;i=16803072	
identifierType		NUN	IERIC	•
value 16803072				
namesnace		4: ur	n:compactcom40	:parameters 🔹

NodeID for the value of ADI number 101, 0x01006500 = 16803072.

The corresponding ADI-list for the Anybus CompactCom Host Application Example Code will look like this. The built-in data types on the OPC UA side is translated to Anybus CompactCom data types according to the translation table that can be found in the network guide for the Anybus CompactCom 40 IIoT Secure.

```
const AD_AdiEntryType APPL_asAdiEntryList[] =
{
    {
        { 100,
        "MotorTemperature", ABP_DOUBLE, 1, APPL_NOT_MAP_WRITE_ACCESS_DESC, {
        & & appl_MotorTemperature, NULL } } ,
        { 101, "SerialNumber", ABP_CHAR,
        &, APPL_NOT_MAP_WRITE_ACCESS_DESC, { { appl_SerialNumber,
        NULL } } ;
};
```

8. Export XML.

Click the 'Export XML' button → and fill out the information. Select all included namespaces and check the 'Include Type Dictionary' and 'Include Values' checkboxes. Click 'Ok'.

Export C:\In	file path * formationModelExample\InformationModelExample.xi	ml C
Sele	ect namespace	
	http://opcfoundation.org/UA/	
\checkmark	http://opcfoundation.org/UA/DI/	
\checkmark	http://opcfoundation.org/UA/Robotics/	
\checkmark	http://MyNewNameSpaceURI	
\checkmark	urn:compactcom40:parameters	
	nclude mappings	
~	nclude Type Dictionary	
\checkmark	nclude Values	
	Cancel	Ok

9. The exported information model is now ready to be encoded by the Anybus OPC UA NodeSet Encoder.

3. Related Documents and Tools

To help with the certificate handling when connecting to the new information model, there is an application note and a certificate generator available on the support pages.

- Application Note How to connect the Anybus CompactCom 40 IIoT Secure to UaExpert (Unified Automation)
- Certificate Generator