



Integration of S7-1500 Package Units with SIMATIC PCS 7/OPEN OS



SIMATIC PCS 7 V9.0

https://support.industry.siemens.com/cs/ww/en/view/49740087

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place. For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: <u>https://www.siemens.com/industrialsecurity</u>.

Table of contents

Lega	al informa	ation	2
1	Task		4
2	Solutio	n	5
	2.1 2.2	Overview Description of the core functionality	
3	Basic p	principles	7
	3.1 3.2 3.2.1 3.2.2 3.2.3	Configuration guidelines Editors PCS 7 DBA AS Node Type Wizard PCS 7 DBA PCS 7 DBA Type Editor	10 10 11 11
4	Config	uring an S7-1500 package unit	13
	Structu	Acquiring data Generating OS block icons and faceplates Generating the AS node with the AS Node Type Wizard General tab Instance Source tab "Connection" tab Addressing tab features when addressing an S7-1500 PU red program design in the S7-1500 PU tured program design in the S7-1500 PU Creating the AS node Generating AS object types Creating a new type Including default types Creating tags Configuring alarms Attributes Creating ASO instances	16 18 19 20 23 24 27 31 33 34 37 39 42 46
5	Config	uring the PC station with the DBA	50
	5.1 5.2 5.2.1 5.2.2 5.2.3 5.2.4 5.2.5 5.2.6 5.2.7 5.3	Adding a PC Station Creating a plant hierarchy Setting project properties Synchronizing the plant hierarchy Extending the plant hierarchy Assigning hierarchy folders to an OS Assigning AS object instances Editing the properties of the objects Compile OS PCS 7 operator station	52 52 54 54 55 56 57 57
6	Append	dix A	61
	6.1 6.2	Skript "CalcS71500Addr" Skript "CalcS71500ASOAddr"	
7	Append	dix B	63
	7.1 7.2 7.3 7.4	Service and support Industry Mall Literature History	64 64

1 Task

Introduction

System landscapes that have grown over time often consist of heterogeneous automation technologies. With the introduction of the TIA portal, there exists the requirement that new Package Units containing S7-1500 CPUs can also be operated and monitored using a PCS 7 operator system.

SIMATIC PCS 7/OPEN OS is a PCS 7 option with which controllers that does not belong to the range of SIMATIC PCS 7 system components can be integrated into the PCS 7 process control system.

PCS 7/OPEN OS V9.0 enables the data exchange between the PCS 7 operator station and various automation systems via the existing WinCC channels or via the OPC channel. For third-party systems that can only be integrated via the OPC channel, only the appropriate OPC server for the particular controller type is necessary. PCS 7/OPEN OS supports the data exchange with the controllers via Classic OPC DA, Classic OPC A&E and OPC UA DA.

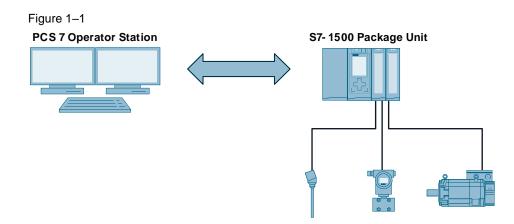
The core of PCS 7/OPEN OS is based on the database automation software (DBA) familiar from other OS options. This software is mainly composed of the following components:

- SIMATIC PCS 7 OS engineering and runtime software
- PCS 7/OPEN OS DBA data base automation software
- PCS 7/OPEN OS runtime software option

The supplied base functionality of PCS 7/OPEN OS allows configuration engineers to effectively integrate the S7-1500 controllers into the process control using the PCS 7 operator system. This gives operators the option of operating and monitoring the entire system from a single operator station, even if it contains package units with S7-1500 controllers.

Overview of the automation task

An S7-1500 package unit is to be operated and monitored using the SIMATIC PCS 7 operator station. The integration of the system into the PCS 7 landscape will take place using SIMATIC PCS 7/OPEN OS via S7-1500 OMS+ Channel.

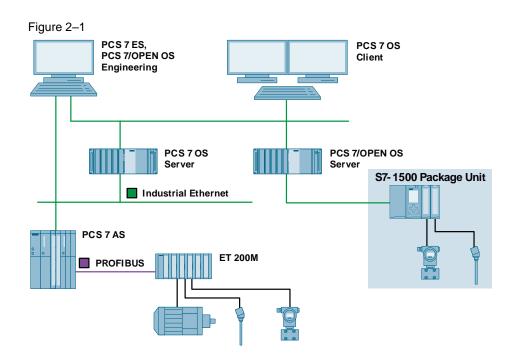


2 Solution

Using the PCS 7 SIMATIC PCS 7/OPEN OS option, this application example will show you how you can effectively integrate S7-1500 controllers that do not belong to the PCS 7 range into the PCS 7 system landscape.

2.1 Overview

The example contained in this document shows you how to integrate an S7-1500 package unit into an existing PCS 7 landscape. The S7-1500 controller is integrated with the aid of the SIMATIC S7-1200 or S7-1500 channel of the PCS 7 operator station.



Advantages

The solution presented in this document offers you the following advantages:

- Complete integration of controllers that do not conform to PCS 7 into the PCS 7 operator system
- Step by step instruction for configuration with PCS 7/OPEN OS
- Shared alarm and tag logging management of PCS 7 and third-party systems on one operator system

Required knowledge

The following basic knowledge is required:

- Systems configuration with PCS 7 AS engineering
- Generation of a visualization with PCS 7 OS engineering

2.2 Description of the core functionality

A core component for the PCS 7/OPEN OS engineering is the PCS 7/OPEN OS DBA database automation software.

The DBA generates OS data such as:

- Plant hierarchy
- Tags and archive tags
- Connections
- Alarms and messages

The DBA can use the channels available in the PCS 7 OS to connect the systems that are not PCS 7-conform. For example, drivers are integrated in WinCC for the following third-party systems:

- SIMATIC S7-1200, S7-1500
- SIMATIC 505
- Allen Bradley
- Mitsubishi
- ...

Third-party systems for which no special connections exist can use the OPC or Modbus TCP open standards.

3 Basic principles

3.1 Configuration guidelines

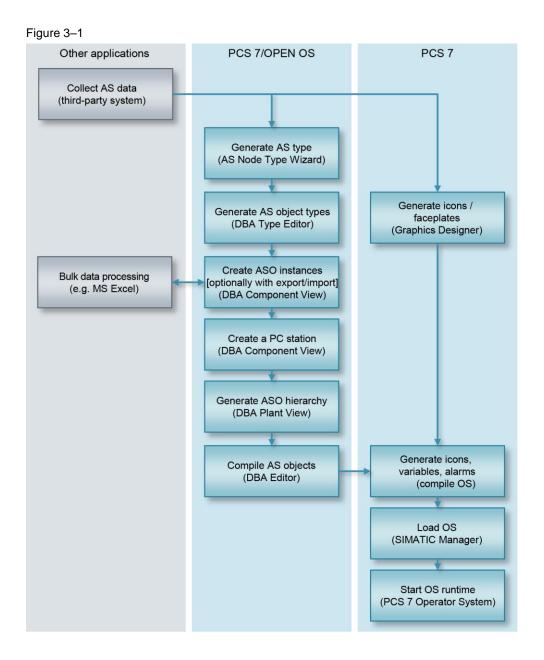
You will find here an overview of the steps necessary for integrating a third-party system into the PCS 7 operator station using PCS 7/OPEN OS.

-1

No.	Description
1.	Collecting data
	Before you start configuring, you will need the following information in order to
	operate and monitor the package unit:
	 Access type (S7 connection, OPC DA, OPC A&E,) Network addresses (OPC server, automation systems)
	 Network addresses (OFC server, additiation systems) Syntax of tag addresses
	 Tags, interfaces, alarms
	 Project file path of the target operator system
	Plant hierarchy
2.	Generating OS block icons and faceplates
	Using the available knowledge of the structure of the objects (type of objects and related tags) in the AS, you can generate the block icons and faceplates. In conformance with PCS 7, the icons are stored in a Typicals file and the individual views of the faceplates are generated.
	Further information on the generation of icons and faceplates can be found in the "SIMATIC PCS 7 Process Control System APL Style Guide" configuration instructions in the following article:
	https://support.industry.siemens.com/cs/ww/en/view/65601446
	Note
	It is imperative that this work be completed before the initial compilation of the plant hierarchy.
3.	Generating the AS node type
	You will generate the AS node type using the AS Node Type Wizard. The Wizard already contains information about the channel used and the corresponding connections as well as the syntax of the tag addresses.
4.	Creating the AS node
	In the AS View of the DBA, you will generate an instance of the previously generated AS node type. Depending on the setting, you may still be able to adjust the connection parameters and assign an instance name in this view.
5.	Generating AS object types
	The AS objects (ASO) represent the objects present in the automation system, e.g. motors, valves, closed-loop controllers, analog and digital values, and are connected to the AS node type.
	The ASO types can contain tag addresses, the tag format, tag attributes, and tag alarms and messages.

No.	Description
6.	Creating ASO instances
	Create the ASO instances. These instances are always assigned to an AS node.
7.	Creating a PC station
	In the PC Station View of the DBA, you can define the OS project to which the data will be compiled. Here, you will set the project path of the target system and also specify a data log if one exists.
8.	Specifying the plant hierarchy
	An existing plant hierarchy can be read out from the SIMATIC project and synchronized with the DBA project. In the Plant View of the DBA, you can then create additional hierarchy folders and subsequently synchronize them with the SIMATIC project again.
9.	Assigning ASO instances to the plant hierarchy
	Drag the ASO instances on to the corresponding hierarchy folder in the plant view to establish the relationship of the AS objects to the operator system. Depending on the mode of configuration, you may still be able to assign or adjust tag addresses and attributes.
10.	Compiling the DBA project
	This process roughly corresponds to that of the OS compilation from the SIMATIC Manager. Here, the necessary connections, tags, messages and icons are created in the OS project.
11.	Loading the OS and starting the runtime
	If all sequences have been executed properly, the OS project can be loaded and the OS runtime can be started.

The following flow diagram shows the procedure for configuring third-party systems using SIMATIC PCS 7/OPEN OS in abbreviated form.



3.2 Editors

PCS 7/OPEN OS includes the following editors, which are briefly described below:

- PCS 7 DBA AS Node Type Wizard
- PCS 7 DBA
- PCS 7 DBA Type Editor

3.2.1 PCS 7 DBA AS Node Type Wizard

You will generate the AS node types using the AS Node Type Wizard. Here, the following information will be stored:

- AS name
- Typical display
- Instance source (DBA or XML)
- Connection parameters
- Address syntax

You can start the AS Node Type Wizard at "Start > Siemens Automation > AS Node Wizard".

You will find detailed information in the "PCS 7 Open OS Engineering Workflow Guide" manual. The manuals are copied to your system during the OPEN OS installation.

General	Instance Source	Connection	Addressing	
Channel & Channel Unit -	madance Source	connection	Addressing	
OPC Channel OPC Data Acc	ess Server (DA)			
Default OPC I				
Default OPC I	1			
	chitecture Server (UA)			
Enable OPC A&E S				
Default OPC A&E Serv Default OPC A&E Proc	,			
Allow Redundant (1			
Other Channel	connections			
Channel	0144710 071000 071	1500 Ci		
Channel Unit:	SIMATIC \$7-1200, \$7-1	1500 Channel		
Channel Unic	OMS+			
Connection				
Connection Name:	*ASName*			
	ame of the Sustem (as config	arred on the 'General' tab)		_
StratenManas N		parea errerer a errerar (ab)		
<pre>%SystemName% N %ASName% N</pre>	ame of the AS Node			
ASName N				
* ASName * N	to insert at cursor location	Node		
* ASName * N * Double click blue text I Allow the Connection		Node		
* ASName * N	to insert at cursor location Name to be changed per AS		MATIC-ROOT-HMI: hFaiaZYS:	IhbJIcVx/
* ASName * N * Double click blue text I Allow the Connection Initial Parameter:	to insert at cursor location Name to be changed per AS	onn!::192.168.0.1:SIN	MATIC-ROOT-HMI: !hFmimZYS;	IhbJIcVx/
* ASName * N * Double click blue text Allow the Connection Initial Parameter: Allow the Initial Param	to insert at cursor location Name to be changed per AS	onn ! : : 192.168.0.1: SIN	BATIC-ROOT-HMI: hFmim2YS	IhbJIcVx/
* ASName * N * Double click blue text Allow the Connection Initial Parameter: Allow the Initial Param	to insert at cursor location Name to be changed per AS 6 ! : : : : : : : : : : : : : : : : : :	onn ! : : 192.168.0.1: SIN	MATIC-ROOT-HHI: !hFmimZYS;	IhbJIcVx/

3.2.2 PCS 7 DBA

The DBA Editor is the heart of PCS 7/OPEN OS. This is where you configure the AS structures and connect them to the PCS 7 operator system. Here, you must carry out the following tasks:

- Create the AS node and AS object instances (ASO's)
- Generate the plant hierarchy and assign ASO instances
- Compile the DBA project

You can start the DBA at "Start > Siemens Automation > PCS 7 DBA".

You will find detailed information in the "PCS 7 Open OS DBA" manual. The manuals are copied to your system during the OPEN OS installation. Figure 3–3

<u> E</u> dit <u>R</u> un <u>T</u> ools <u>H</u> elp					
Plant View	4 🗙 🚰 🛄 Objects: 0 A	uttributes: 1 Grap	hics: 19 Alarm Messages		
⊡∽ <mark>¶N</mark> S7_1500_Demo ⊡∽ <mark>©a</mark> Plant1	Name		Value	Category	Description
	AH - Con	nments	Limit Alarm High	Graphics	AH - Comments
E BMT1	😽 AH - Star			Graphics	AH - Start Value
	AL - Com		Limit Alarm Low	Graphics	AL - Comments
ATER	😽 AL - Star			Graphics	AL - Start Value
- F1001	Sector Commer	it	L1001	Graphics	Comment
F2001	😽 MLH - Ca	omments	Measuring Limit Ligh	Graphics	MLH - Comments
🕞 L1001	MLL - Co	mments	Measuring Limit Low	Graphics	MLL - Comments
MOT1	VOUT_AH	I - Comments	Alarm High	Graphics	OUT_AH - Comments
- MOT2	VOUT_AL	- Comments	Alarm Low	Graphics	OUT_AL - Comments
P1001	VT_W	H - Comments	Warning High	Graphics	OUT_WH - Comments
		- Comments	Warning Low	Graphics	OUT_WL - Comments
VAL1	™ SF		@S71500_Typicals.pdl	Graphics	Symbol File
VAL2	™ SN		@S7Meas/1	Graphics	Symbol Name
		ommonto	1 Init	Crophico	LINIT Commonto
S View PC Station View Validation					
omponents	AS Objects: 1				,,
] S7_1500	Name	Туре	AS Address	Status	Assignment
	35 F1001	S7Meas	0001:CL:%DB101.%DBB80		\Plant1\WATER
	M F2001	S7Meas	0001:CL:%DB101.%DBB16	0	\Plant1\WATER
	35 L1001	S7Meas	0001:CL:%DB101.%DBB0		\Plant1\WATER
	™ MOT1	S7Motor	0001:CL:%DB100.%DBB0		\Plant1\WATER
	MOT2	S7Motor	0001:CL:%DB100.%DBB2		\Plant1\WATER
	27 P1001	S7Meas	0001:CL:%DB101.%DBB40		\Plant1\WATER
	N P2001	S7Meas	0001:CL:%DB101.%DBB12	0	\Plant1\WATER
	SFC1	S7SFC	0001:CL:%DB400.%DBB0		\Plant1\WATER
Include All Resources	VAL1	S7Valve	0001:CL:%DB100.%DBB12		\Plant1\WATER
Include Assigned Include Only Changes	VAL2	S7Valve	0001:CL:%DB100.%DBB14		\Plant1\WATER

3.2.3 PCS 7 DBA Type Editor

Using the DBA Type Editor, you can generate the AS objects, which map the structure of the blocks in the third-party system. They contain:

- Tags, possibly with addresses
- Enumerations
- Alarms and messages
- Attributes
- Runtime scripts

Figure 3-4

You can start the Editor via the shortcut menu of an AS node in the DBA by selecting the "Edit AS Object Types..." item.

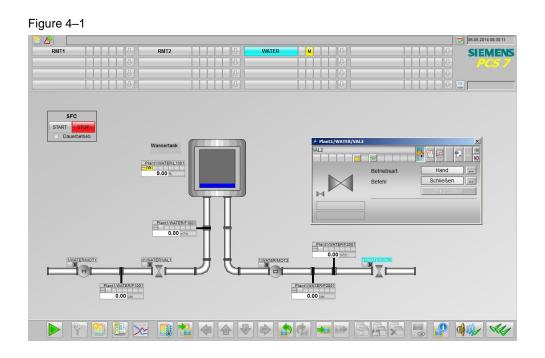
You will find detailed information in the "PCS 7 Open OS DBA Type Editor" manual. The manuals are copied to your system during the OPEN OS installation.

iystem	Ĺ	f ∕ Type <u>S</u> cripts		Test					v— <u> </u>	
71500		≣∐ags	<u>⊡M</u> es:	sages 🧉	/ Att	ibutes 🏾	Ŭ <u>I</u> ncludes ∐	QDocs	🎽 🔲 OS <u>R</u> u	
Types\Enumerations		Include >>	N	ame >>		Type >>	Int/Ext>>	Requir	ed >> 37	of 37 t
AlarmGroupDisplaySupport		Parent Type		Name		Туре	Int/Ext	Address	Text_	Ŧ
CommonMembers	10	AlarmGroupDispla	ySupport	EventState		SDWORD	Internal			<u> </u>
CommonMemoMembers	10	CommonMembers		#blocktype		TEXTREF	Internal		@T\	
S7Meas	10	CommonMembers		#comment		TEXTREF	Internal		@IN	11
S7Motor	0	CommonMembers		#areaname		TEXTREF	Internal			
S7SFC	0	CommonMemoMe	embers	#StatusPerma	m	DWORD	Internal			+
S7Valve	0	CommonMemoMe	embers	#TextPermane	ent	TEXT8	Internal			· ·
💁 SymbolAndFaceplateExtendedAttributes		S7Valve		RESET#short	cut	TEXTREF	Internal		Error	
		S7Valve		RESET		BIT	External	0.7		⊻
	/ 🖆	S7Valve		QOPCOND#s	h	TEXTREF	Internal		Орег	_
	< 🛛	S7Valve		QOPCOND		BIT	External	1.4		
		S7Valve		MAN#shortcu	ıt	TEXTREF	Internal		Man	
		S7Valve		MAN		BIT	External	0.3		
		S7Valve		AUT0#shortc	ut	TEXTREF	Internal		Auto	
		S7Valve		AUTO		BIT	External	0.4		
4 Þ		S7Valve		QAUT0#shor	tcut	TEXTREF	Internal		State	
		S7Valve		QAUTO		BIT	External	1.1		
		S7Valve		QMAN#shorte	sut	TEXTREF	Internal		State	
rowse Reference		S7Valve		QMAN		BIT	External	1.2		
				QERR#shortc	ut	TEXTREF	Internal		Error	
		S7Valve		QERR		BIT	External	1.5	-	1
		[F	
Expert Mode		Show All Included	Tags				A	dd <u>E</u> c	lit <u>D</u> ek	ete
Tools							<u>D</u> K	<u>C</u> ancel	н	elp

© Siemens AG 2014 All rights reserved

4 Configuring an S7-1500 package unit

The following chapters describe how to integrate an S7-1500 PU into PCS 7 using OPEN OS. The SIMATIC project used in the example is provided solely for purposes of demonstration. However, the basic procedure for real package units with S7-1500 controllers is largely identical.



In order to purge a pipe system, a tank is filled with water and the water is then fed into the pipe system. A maximum level and a minimum level of the tank can be predetermined for the purging sequence.

The sequence places the valves and motors in automatic mode and controls them until the purging process is complete. The actuators are then switched into manual mode again.

4.1 Acquiring data

This example project contains the following components, which are relevant for operation and monitoring at the OS:

Туре	Name	AS address	Description
Motor	MOT1	DB100, DBB0	Pump drive for filling the tank
Valve	VAL1	DB100, DBB12	Tank intake valve
Analog	P1001	DB101, DBB40	Tank intake pressure transducer
Analog	F1001	DB101, DBB80	Tank intake flow sensor
Motor	MOT2	DB100, DBB2	Pump drive for emptying the tank
Valve	VAL2	DB100, DBB14	Tank drain valve
Analog	P2001	DB101, DBB120	Tank drain pressure transducer
Analog	F2001	DB101, DBB160	Tank drain flow sensor
Analog	L1001	DB101, DBB0	Tank level
Sequenc e	SFC1	DB400, DBB0	Sequence for purging the pipelines

Tab	ما	4-	1
rap	ie	4-	-1

The object types motor, valve and analog value are always based on the same function block. This makes it possible to derive a structure from the instance DB of a block.

Motor block

The following motor block variables are required for configuring the OS:

Tag	Туре	DB Offset	Туре	Description			
MAN_ON	BOOL	0.0	IN	Start motor manually			
MAN_OFF	BOOL	0.1	IN	Stop motor manually			
AUTO_ON	BOOL	0.2	IN	Start motor in automatic mode			
MAN	BOOL	0.3	IN	Manual mode			
AUTO	BOOL	0.4	IN	Automatic mode			
FB	BOOL	0.5	IN	Motor feedback signal			
RESET	BOOL	0.6	IN	Reset of a feedback error			
QSTART	BOOL	0.7	OUT	Signal for starting the motor			
QAUTO	BOOL	1.0	OUT	Automatic mode active			
QMAN	BOOL	1.1	OUT	Manual mode active			
QMOTRUN	BOOL	1.2	OUT	Motor running			
QOPCOND	BOOL	1.3	OUT	Operating condition			
QERR	BOOL	1.4	OUT	Motor feedback error			

Та	b	e	4-	-2	

The time limit for the block to report a feedback error is permanently set to 5 seconds.

Valve block

The following valve block variables are required for configuring the OS:

Тад	Туре	DB Offset	Туре	Description			
MAN_OPEN	BOOL	0.0	IN	Open valve manually			
MAN_CLOSE	BOOL	0.1	IN	Close valve manually			
AUTO_OPEN	BOOL	0.2	IN	Open valve in automatic mode			
MAN	BOOL	0.3	IN	Manual mode			
AUTO	BOOL	0.4	IN	Automatic mode			
FB_OPEN	BOOL	0.5	IN	Open valve feedback signal			
FB_CLOSE	BOOL	0.6	IN	Close valve feedback signal			
RESET	BOOL	0.7	IN	Reset of a feedback error			
QOPEN	BOOL	1.0	OUT	Signal for opening the valve			
QAUTO	BOOL	1.1	OUT	Automatic mode active			
QMAN	BOOL	1.2	OUT	Manual mode active			
QVALOPEN	BOOL	1.3	OUT	Valve open			
QOPCOND	BOOL	1.4	OUT	Operating condition			
QERR	BOOL	1.5	OUT	Valve feedback error			

Table 4–3

The time limit for the block to report a feedback error is permanently set to 5 seconds.

Analog value monitoring

The following analog block variables are required for configuring the OS:

Table 4–4				
Tag Type		DB Offset	Туре	Description
VALUE	REAL	0.0	IN	Process value
AL	REAL	4.0	IN	Lower alarm limit
WL	REAL	8.0	IN	Lower warning limit
WH	REAL	12.0	IN	Upper warning limit
AH	REAL	16.0	IN	Upper alarm limit
OUT_AL	BOOL	20.0	OUT	Lower alarm limit reached
OUT_WL	BOOL	20.1	OUT	Lower warning limit reached
OUT_WH	BOOL	20.2	OUT	Upper warning limit reached
OUT_AH	BOOL	20.3	OUT	Upper alarm limit reached
MLH	REAL	22.0	IN	Maximum value
MLL	REAL	26.0	IN	Minimum value
UNIT	STRING [8]	30.0	IN	Process value unit

Block sequence

The following block sequence variables are required for configuring the OS:

able 4–5				
Tag	Type DB address		Туре	Description
START	BOOL	0.0	IN	The sequence is started.
STOP	BOOL	0.1	IN	The sequence is stopped.
CYCLE	BOOL	0.3	IN	The sequence runs in a continuous loop

Table 4–5

4.2 Generating OS block icons and faceplates

You can generate OS block icons and faceplates with the aid of the WinCC Graphics Designer. The process conforms to the PCS 7 standard. For this reason, the topic is not discussed in greater detail in this documentation. Detailed information on the generation of user-defined icons and faceplates can be found in the "SIMATIC Process Control System PCS 7 APL Style Guide" manual in the following article:

https://support.industry.siemens.com/cs/ww/en/view/65601446

Use the collected data to generate the block icons and faceplates. Taking into consideration the naming conventions for Typicals pictures, use a meaningful name for the icon picture, e.g. "@S71500_Typicals.pdl".

Note

When creating the block icons, please bear in mind the correct labeling for the object name. E.g., for the "S7MEAS" block, the object name for the associated block icon is "S7MEAS/1".

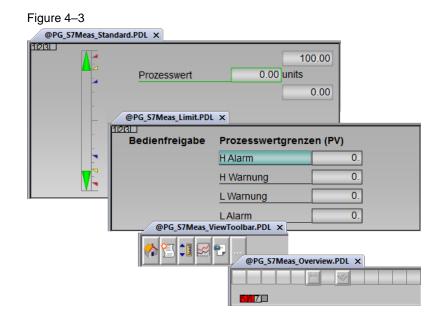
In the present example, these are the blocks for:

- Motor
- Valve
- Analog value
- SFC (no faceplates)

Figure 4–2

@S71500_Typicals.PDL ×	
S7-1500 PC	S 7 Typicals
Bausteinfamilie: Antriebe	
S7Motor	S7Valve
Bausteinfamilie: Beobachten	Bausteinfamilie: Bedienen
S7Meas	S7SFC SFC
-99999.99	START STOP Dauerbetrieb

One faceplate set each was created for the motor, valve and analog value icons. The block icon for the sequence does not have a faceplate.



4.3 Generating the AS node with the AS Node Type Wizard

The AS node type is generated with the aid of the PCS 7 DBA AS Node Type Wizard. You can find the editor in the Start menu at: "Start > Siemens Automation > AS Node Wizard".

The AS node type is not part of a specific DBA project and is therefore stored in the installation directory of OPEN OS. You can use all the created AS node types in each of your DBA projects. The following sections explain how to create an AS node type.

4.3.1 General tab

In the "General" tab, you assign the name of the AS node type and define the name of the typical picture from which the OS block icons are copied during compilation.

- 1. Enter the name for the AS node type.
- 2. Enter the name of the WinCC screen that contains the generated picture symbols.

Figure 4-4

General	Instan	ice Source	Connection	Addressing	
System Name:	\$71500				Ŧ
Graphics Options					
Typical Name:	@S71500_Typic	cals.PDL			
					Version 9.0.

4.3.2 Instance Source tab

This document only describes the DBA Integrated Instances option.

1. Select the DBA Integrated Instances option.

	izard			
General	Instance Source	Connection	Addressing	
• DBA Integrated	Instances			
Choose this option to a techniques.	illow the DBA project to create n	ew instances using the integr	rated tools and/or through	CSV Import/Export
Use this option if you p	refer Excel or CSV formats for de	efining instances.		
* the ability to create n	ill result in: pes using the Type Editor ew AS Objects from directly with xport AS Objects from directly w	in the DBA thin the DBA using IEA-style	CSV files	
Note: This option mak	es extensive use of Extended A	ttributes.		
← XML Instances				
Choose this option to a automatically from som	Ilow the DBA to automatically cr external program or script that the XML Input file therefore the	converts some exported cont	roller data. This technique	requires that all instances be
Choose this option to a automatically from som created and defined by	e external program or script that	converts some exported cont user cannot manually create	roller data. This technique new instances from within	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the c	e external program or script that the XML Input file therefore the ontroller's configuration is easily (converts some exported cont user cannot manually create	roller data. This technique new instances from within	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the cr Choosing this option w " the ability to define al " the ability to define al	e external program or script that) the XML Input file therefore the pontroller's configuration is easily p ill result in: pes using the Type Editor IAS Object instances and assoc	converts some exported cont user cannot manually create parsed and you have the tech isted configuration data in a	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the cr Choosing this option w * the ability to define by * the ability to define al * the inability to create	e external program or script that v the XML Input file therefore the ontroller's configuration is easily p ill result in: per using the Type Editor IAS Diject instances and assoc new AS Dbjects from directly wil	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the cr Choosing this option w * the ability to define by * the ability to define al * the inability to create	e external program or script that) the XML Input file therefore the pontroller's configuration is easily p ill result in: pes using the Type Editor IAS Object instances and assoc	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the cr Choosing this option w * the ability to define by * the ability to define al * the inability to create	e external program or script that v the XML Input file therefore the ontroller's configuration is easily p ill result in: per using the Type Editor IAS Diject instances and assoc new AS Dbjects from directly wil	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the cr Choosing this option w * the ability to define by * the ability to define al * the inability to create	e external program or script that v the XML Input file therefore the ontroller's configuration is easily p ill result in: per using the Type Editor IAS Diject instances and assoc new AS Dbjects from directly wil	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the c Choosing this option w * the ability to define y * the ability to define al * the inability to create	e external program or script that v the XML Input file therefore the ontroller's configuration is easily p ill result in: per using the Type Editor IAS Diject instances and assoc new AS Dbjects from directly wil	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the cr Choosing this option w * the ability to define by * the ability to define al * the inability to create	e external program or script that v the XML Input file therefore the ontroller's configuration is easily p ill result in: per using the Type Editor IAS Diject instances and assoc new AS Dbjects from directly wil	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.
Choose this option to a automatically from som created and defined by Use this option if the c Choosing this option w * the ability to define y * the ability to define al * the inability to create	e external program or script that v the XML Input file therefore the ontroller's configuration is easily p ill result in: per using the Type Editor IAS Diject instances and assoc new AS Dbjects from directly wil	converts some exported cont user cannot manually create carsed and you have the tech isited configuration data in a him the DBA or Import/Export	roller data. This technique new instances from within nnical ability to create XML single XML file.	requires that all instances be the DBA.

You can find a description of the use of XML instances in the following article: "<u>How do you create an XML input file for automatic creation of instances in DBA?</u>" (Entry ID: 78032500).

4.3.3 "Connection" tab

Because the S7-1500 is a SIMATIC product, using the compatible SIMATIC S7-1200 or S7-1500 channel is a logical choice.

- 1. Select the Other Channel option.
- 2. Enter the channel and the channel unit that are used.
- 3. Enter a permanent connection name or use the following possible tags:
 - %SystemName%: The name of the AS node type is used (General tab).
 - %ASName%: The name of the AS node is used that is entered in the DBA.
- 4. Enter the connection string used by WinCC as the initial parameter.

Figure 4-6

	Instance Source	Connection	Addressing		
Channel & Channel Unit:					
OPC Channel OPC Data Acces	ss Server (DA)				
Default OPC D/	A Server: <local></local>				
Default OPC DA	A ProgID:				
C OPC Unified Arc	hitecture Server (UA)				
Enable OPC A&E Su	une ad				
Default OPC A&E Serve					
Default OPC A&E Progl)				
Allow Redundant Co	1				
Other Channel	111000013				
Channel:	SIMATIC \$7-1200, \$7-	1500 Channel			
Charlino.	JSIMATIC 57-1200, 57-	1 JUU Cridninei			
Channel Unit:	OMS+				
Channel Unit:	OMS+				
Channel Unit:	OMS+				
	OMS+				
Connection Connection Name: \$SystemName\$ Na		gured on the 'General' tab)			
Connection Connection Name: *SystemName* Na	\$ASName* me of the System (as config me of the AS Node	gured on the 'General' tab)			
Connection Connection Name: \$SystemName\$Na \$ASName\$Na *Double click blue text to	\$ASName\$ me of the System (as confi me of the AS Node insert at cursor location				
Connection Connection Name: *SystemName* Na *ASName* Na *Double click blue text to V Allow the Connection N	\$ASName\$ me of the System (as config me of the AS Node insert at cursor location ame to be changed per AS	Node			
Connection Connection Name: \$SystemName\$Na \$ASName\$Na *Double click blue text to	\$ASName\$ me of the System (as config me of the AS Node insert at cursor location ame to be changed per AS	Node	LATIC-ROOT-HMI:!hFai	aZYSIhbJIcV	
Connection Connection Name: *SystemName* Na *ASName* Na *Double click blue text to V Allow the Connection N	\$ASName\$ me of the System (as confi me of the AS Node insert at cursor location ame to be changed per AS 61:::::CP_H1_1:	Node	LATIC-ROOT-HMI: !hFai	aZYSIhbJIcVy	
Connection Connection Name: \$SystemName\$ Na \$ASName\$ Na "Double click blue text to I Allow the Connection N Initial Parameter:	\$ASName\$ me of the System (as configme of the AS Node insert at cursor location ame to be changed per AS 61:::::CP_H1_1:1	Node !::10.250.74.152:SIM	LATIC-ROOT-HMI: !hFai	aZYSIhbJIc∀;	
Connection Connection Name: SystemName* Na *ASName* Na * Double click blue text to I Allow the Connection N Initial Parameter: I Allow the Initial Parameter	\$ASName\$ me of the System (as configme of the AS Node insert at cursor location ame to be changed per AS 61:::::CP_H1_1:1	Node !::10.250.74.152:SIM	LATIC-ROOT-HMI: !hFai	aZYSIhbJIcV)	

The text in the "Channel" and "Channel Unit" fields must be identical to the corresponding WinCC channel. You can find the necessary inputs in the WinCC Tag Management. Follow the steps below:

- 5. Open an existing or new OS project.
- 6. Open the WinCC Tag Management.
- 7. Click on "Tag management" with the right hand mouse button and add the new "SIMATIC S7-1200, S7-1500 Channel" driver.

Figure 4–7

Tag Management	~<
Tag Management	
🕀 🍄 Internal tags	
🖅 🖳 SIMATIC S7 Protocol Suite	
🖃 🕂 SIMATIC S7-1200, S7-1500 Ch	annel
OMS+	

The "initial parameter" corresponds to the string that WinCC creates for a connection. If you do not know the exact string syntax, you can read the parameter out of WinCC. Proceed as follows:

- 8. Open an existing or new OS project.
- 9. Open the WinCC Tag Management.
- 10. In the desired channel unit, temporarily create a new connection with the desired connection parameters (it can be deleted later).

Note Since the existing access points might be used by other channels, it is advisable to create a new access point for the TCP/IP protocol for the S7-1500 channel in the SIMATIC NET "Communications settings" editor.



11. Select the created access point in the "New Connection" window.

Tag Management « Tag Management • Internal tags • IsinATIC S7 Protocol Suite • • • Internal tags • IsinATIC S7 Protocol Suite • • • IsinATIC S7-1200, S7-1500 Channel •	Figure 4–8			
Image: Tag Management Internal tags Image: Simatic S7 Protocol Suite Image: Simatic S7 Protocol Suite Image: Simatic S7-1200, S7-1500 Channel Image: Simatic S7-1200, S7-1200, S7-1500 Channel Image: Simatic S7-1200, S7-1200, S7-1500 Channel Image: Simatic Sima	-	**		
Enter the IP address of the automation system. Example: 142.11.0.123	Tag Management Tag Management Tags Internal tags The second	Suite 7-1500 Channel New Connection - Connection Options S7Plus network add IP address: Access point: Product family: Enter the IP address	ress [172.10.1.67 [S7_1500_conn [S7 1500] [S7	
OK Cancel Help				

 Use the option button next to the text field for the initial parameter to start the WinCC Connection String Inspector. It is vital that the WinCC project with the corresponding connections is open. Otherwise, no connections will be displayed.

Figure 4–9

1		Allow the Connection	n Name to b	e changed p	er AS Node							
l		Initial Parameter:										
		🔽 Allow the Initial Para	meter to be (changed per	AS Node							
á	ţ١	WinCC Connection Stri	ing Inspect	tor								
	Ch	annel			Channel	Conn	Parameter					
Γ	SI	MATIC S7-1200, 3	S7-1500	Cha	OMS+	Temp	4!::::S7	1500	conn!:::172.10.1.67:			
L	•								Þ			
				Clie	ck OK to choose the	selected Paran	neter string					
					ОК	Cancel						

- 13. In the event that multiple connections are available, select the appropriate connection and click OK in the dialog window to confirm the selection.
- 14. Do not delete the temporary connection yet. You will need it again at a later point to determine the address syntax.

4.3.4 Addressing tab

In the Addressing tab, you can specify the structure of the AS and variable addressing that will be used when the project is compiled at a later point. The address structure depends on the communications driver that is set. Here, the DBA offers the option of assigning parameters for all conceivable constellations. These range from a structured approach and the free assignment of addresses for all tags, to the use of scripts that can calculate the addresses individually.

The following tags are available for the assignment of ASO address parameters:

- %ASName% name of the AS type
- %ASOName% name of the AS instance
- %Instance% assigned during instantiation
- %Attribute:Name% value of the specified attribute
- %UID% unique identifier generated by the DBA

The following tags are available for the assignment of variable address parameters:

- %ASName% name of the AS type
- %ASOName% name of the AS instance
- %Instance% assigned during instantiation
- %Tagname% name of the tags for which the address is calculated
- %Address% value of the address field of the tags
- %Field#TagAddress% value of the extended attribute. Input during parameter assignment
- %Attribute:Name% value of the extended attribute with the name "Name"
- %UID% unique identifier generated by the DBA
- %OPCDataType% OPC data type number for the tag type
- %UANamespace% OPC UA adress namen space
- %UAType% OPC UA identifier type
- %UAValue% OPC UA identifier

Special features when addressing an S7-1500 PU

Access by the PCS 7 OS to S7-1500 DBs basically only functions if the "Optimized block access" attribute is deactivated in the TIA project (Figure 4–10). Otherwise it is not possible to recognize which data are in which area of the corresponding DB.

Figure 4–10		
MEAS_DB [DB101]		×
General		_
General Information	Attributes	
Time stamps Compilation Protection Attributes Download with	 Only store in load memory Data block write-protected in the device Optimized block access 	
< III >	< m 3	•
	OK Cancel	

Note also that when an S7-1500 CPU is used, the tag addresses in WinCC and in the SQL database are different. DBA must create the addresses as they appear in the SQL database.

Proceed as follows to read the correct address from the SQL database:

- 1. Create a temporary connection in the WinCC tag manager or use the temporary connection from chapter "4.3.3 "Connection" tab".
- 2. Create a new structure with at least one element.
- 3. Generate an instance of the structure and select the temporary connection.

4. In the "Address" field you can see the syntax of the tag connection in WinCC (Figure 4–11).

Figure 4–11

<u>File Edit View H</u> elp			_										
Tag Management		~	4\$	S7_150	0_Struc			Find			Q	-	
E. A CPU_RT				Name	Data Type	Length	Format	Connection	Group	Address	Linea		
			1	Testvar1	S7_1500_Struc	0		Temporary		DB100,DBB0			
🗄 🖓 Viv2WayL			2										
🖶 🕂 S7_1500_Struc			3										
NewElement_1			4										
🔗 NewElement_2			5										
VinCC Configuration Studio													a'
le <u>E</u> dit <u>V</u> iew <u>H</u> elp													1
ng Management «	10	Tempo	rary						Find			م	5
Tag Management		Name			Data Type	Leng	th Forma	at adaptation		Connection	Group	Address	1
													7
	1	Testvar1	L.Nev	vElement	_1 Unsigned 32-bit		Dword			Temporary		DB100,DD0	
	1					value 4		ToUnsigned	Dword			DB100,DD0 DB100,DD4	
Internal tags Internal tags Internal tags Internal tags	1 2 3	Testvar1	L.Nev	vElement	1 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary			ŀ
Internal tags Internal tage Internal tage Internal tage Internal tage Interna	3 4	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ł
Internal tags Internal tage Internal tage Internal tage Internal tage Interna	3 4 5	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ŀ
	3 4 5 6	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ŀ
Gradient Construction	3 4 5 6 7	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ŀ
STMATIC S7 Protocol Suite STMATIC S7 Protocol Suite STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1200, S7-1500 Channel STMATIC S7-1500 Channel S	3 4 5 6 7 8	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ŀ
Similar Source State Similar State	3 4 5 6 7 8 9	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ł
	3 4 5 6 7 8	Testvar1	L.Nev	vElement	1 Unsigned 32-bit 2 Unsigned 32-bit	value 4 value 4	Dword	ToUnsigned ToUnsigned	IDword IDword	Temporary Temporary		DB100,DD4	ŀ

- 5. Open the "Microsoft SQL Server Management Studio".
- In your corresponding WinCC project, click on "Select Top 1000 Rows" in the context menu for the "dbo.MCPTCONNECTION" table. You can read the "CONNECTIONID" for the S7-1500 connection from the results table (Figure 4–12).

Figure 4–12

K Microsoft SQL Server Management St	udio											
File Edit View Query Project Deb	ug To	ols Wind	low Co	mmunity	Help							
🔛 New Query 📄 🜇 🜇 📴	🛄 New Query 📘 📸 📸 🔚 🕞 🚔 🖼 🚭 🜉 🖕											
	📑 🔢 master 🔹 🔹 🚦 🗸 📅 🗃 🔚 🎌 🝓 🎆 🌑 🗏 😤 👙 🖓 👼											
Object Explorer 🚽 🗸 🗸	S	LQuery3.	sql - ESMI	(G470)) [*]	SQLQuery2.s	ql - ESMIG4B(67))	~ ×					
Connect - 📑 📑 🔳 🖉 📓		/*****	Scri	pt for	SelectTopNR	ows command from SSMS *****/						
dbo.HornAdditionalResetTags dbo.HornGeneral dbo.HornSignals dbo.HornTags dbo.LBM_DEVICELIST dbo.LBM_INFO dbo.MCPTCHANNEL dbo.MCPTCHANNELUNIT dbo.MCPTCONNECTION	-	,	[MACH [CHAN [CONN [PARA] [TIME	INEID] NELUNI ECTION METER] SYNCCY	NAME]							
dbo.MCPTCYCLETIME		CON	MAC	CHAN	CONNECTION	PARAMETER	TIMESYNCCYC					
dbo.MCPTHOTKEY	4	8	2	569	CLMIG2A	OPCServer.WinCC; CLMIG2A; 0.00; 0; 1; 1	0					
dbo.MCPTMACHINE	5	10	2	569	SVRMIG3C	OPCServer.WinCC; SVRMIG3C; 0.00; 0; 1; 1	0					
dbo.MCPTPROJECT dbo.MCPTSTARTUNIT	6	11	2	569	SVRMIG4B	OPCServer.WinCC; SVRMIG4B; 0.00; 0; 1; 1	0					
+ do.MCPTSYSTEM	7	12	2	567	S7\$Pro_REF	NC,"S7 connection_1","WinCC Appl."	0					
dbo.MCPTSYSTEMTABLES	8	14	2	571	Temporary	4!:::::S7_1500_conn!::172.10.1.67:SIMATIC-ROOT-H	0 🖵					

7. Select the command "Select Top 1000 Rows" from the context menu for the "dbo.MCPTVARIABLEDESC" table.

At the end of the "SELECT TOP 1000" script, insert the following syntax with the CONNECTIONID just determined in Step 6 (in our example: 14), (Figure 4–13): where CONNECTIONID like '14'

 When the "Execute" button is clicked, the entries in the results table will be filtered according to the CONNECTIONID and you will be able to see the address of the S7-1500 structure in the "ADDRESSPARAMETER" column (Figure 4–13):

0001:CL:%DBxxx.%Dxxx

Note You will need this addressing syntax for any unstructured program design in the S7-1500 PU (see further on in this chapter).

For the structured program design in the S7-1500 PU in this example we will use a script to calculate the correct tag addresses.

Figure 4–13

🍢 Microsoft SQL Server Management St	udio								
File Edit View Query Project Deb	ug To	ols Window	Commu	nity Help					
🗄 🔔 New Query 🛅 📸 📸 🔂 📑	📬 🛃	🕘 🖾 🖕							
			1 😳 🛛	s 🔒 🚏 🖷 🖓 🌠	0 = 9 #		Ŧ		
Object Explorer 🚽 🕂 🗙	SQL	Query3.sql - E	5MIG4.	70))* SQLQuery2.sql - E	SMIG48(67))				→ ×
Connect 🕶 📑 📑 🛒 📓				for SelectTopNRows					A
🖅 🧾 dbo.MCPTCHANNELUNIT	1 2	SELECT * FR	ROM [(CC_OS_113_10_02_	11_04_23].[d]	bo].[MCP	TVARIABLEDESC] wh	ere CONNECTIONID	like '14'
dbo.MCPTCONNECTION									_
dbo.MCPTCYCLETIME	-								
I dbo.MCPTHOTKEY	E F	Results 🔚 🚮 Mes	sages						
dbo.MCPTMACHINE		VARIABLEID	VAR	LASTCHANGE	FORMATFITTING	SCALEID	VARIABLENAME	ADDRESSPARAMETER	PROTOKOLL V
dbo.MCPTSTARTUNIT	1	1306952	2411	2014-05-06 12:08:01.000	0	0	Testvar1	0001:CL:%DB100.%DBB0	0 -
🛨 🔟 dbo.MCPTSYSTEM	2	1306953	7	2014-05-06 12:08:01.000	119537664	0	Testvar1.NewElement_1	0001:CL:%DB100.%DD0	0 -
dbo.MCPTSYSTEMTABLES	3	1306954	7	2014-05-06 12:08:01.000	119537664	0	Testvar1.NewElement_2	0001:CL:%DB100.%DD4	0 -
dbo.MCPTUTYPESTRUCTMEMBER	4	1306955	7	2014-05-06 12:08:01.000	119537664	0	Testvar1.NewElement 3	0001:CL:%DB100.%DD8	0 -
dbo.MCPTVARGROUP									
dbo.MCPTVARIABLEDESC									
dbo.MCPTVARIABLETYPE									
dbo.MSActions									

CAUTION	The PCS 7 OS project can be damaged
	Only use the SQL database and filtering options to check the tag addresses. Do not make any changes to the database and use only reading operations.

Structured program design in the S7-1500 PU

In this example, the S7-1500 program of the PU is designed in a structured manner. There exist structure objects from which instances which interact with the field have been created. These structure objects are defined as data types (UDT) (Figure 4–14), which communicate with the PCS 7 OS by means of data blocks. These data blocks reproduce the structure of the data types and the addresses of the tags can be precisely calculated from the offset of the structure address (Figure 4–15).

Figure 4–14

-									
S7	_15	00_OPEN_OS_V12	2_SP1 → I	PLC_1 [CPU 1	516-3 PN/DP] 🕨 F	LC data type	s ▶ MEA	\S_TYP	
1	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2								
	ME/	AS_TYP							
_		Name	Data type	Default value	Accessible from HMI	Visible in HMI	Setpoint	Comment	
1		VALUE	Real 🔳	0.0		\sim		ProcessValue	
2	-	A_MIN	Real	0.0		\checkmark		Limit Alarm Low	
з	-00	W_MIN	Real	0.0	\checkmark	\sim		Limit Warning Low	
4	-00	W_MAX	Real	0.0	\checkmark	\sim		Limit Warning High	
5	-00	A_MAX	Real	0.0	\checkmark	\checkmark		Limit Alarm High	
6	-00	OUT_A_MIN	Bool	false	\checkmark	\sim		Alarm Low	
7	-00	OUT_W_MIN	Bool	false		~		Warning Low	
8	-	OUT_W_MAX	Bool	false	\checkmark	~		Warning High	
9	-00	OUT_A_MAX	Bool	false				Alarm High	
10	-00	MLH	Real	0.0	\checkmark	~		Measuring Limit High	
11	-	MLL	Real	0.0	\checkmark	~		Measuring Limit Low	
12	-	UNIT	String[8]		\checkmark	~		Unit of the Measured Values	

Figure 4–15

		_									
\$7 _	_15	00	_OPEN_OS_V	12_SP1 → PLC_	_1 [CPU	1516-3 P	N/DP]	Program blocks	MEAS_DB	[DB101]	
ý	1	*	🎭 🍢 🛤 🛛	- e- e 🗄 I							
	ME	AS_	_DB								
		Nar	me	Data type	Offset	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
1 -	-	Ŧ	Static								
2 -	-00	•	▶ L1001	"MEAS_TYP"	0.0		~		\checkmark		
з.		•	P1001	"MEAS_TYP"	40.0				\checkmark		
4 -		•	▶ F1001	"MEAS_TYP"	80.0				\checkmark		
5 -		•	P2001	"MEAS_TYP"	120.0				\checkmark		
6 -	-	•	▶ F2001	"MEAS_TYP"	160.0				\checkmark		
7	-	•	Dummy	"MEAS_TYP"	200.0						

In this case, based on the structure of the object, the offset in the DB and the various data types of the tags, the addresses of the individual tags can be automatically calculated using scripts.

Then only the DB number and the offset of the structure within the DB are entered for the definition of instances (Figure 4-16).

Figure 4–	16	
🍄 F1001	S7Meas	0001:CL:%DB101.%DBB80
🋂 F2001	S7Meas	0001:CL:%DB101.%DBB160
🋂 L1001	S7Meas	0001:CL:%DB101.%DBB0
🌄 мот1	S7Motor	Single Instance Editor
🛂 мот2	S7Motor	General
🋂 P1001	S7Meas	
🋂 P2001	S7Meas	Name: L1001
💁 SFC1	S7SFC	Type: S7Meas
💁 VAL1	S7Valve	
VAL2	S7Valve	Address: 101,0
		<u>O</u> K <u>C</u> ancel

The scripts for calculating the tag addresses must be contained in the AS node types in DBA. They are only run for the definition of the ASO instance or during an updating run of the AS objects, and do not appear in the OS runtime.

- 1. Enter the following syntax in the AS object instance address text field: %@@CalcS71500ASOAddr("%Instance%")@@%
- Enter the following syntax in the tag address text field: %@@CalcS71500Addr(oTag,"%Instance%","%Address%")@@%

Note Scripts in this context are a later topic. They will not be dealt with here, rather only their effects will be considered.

Figure 4–17

*88@CalcS71SODASOAddr ("%Instance%")88% *ASSName* Name of the AS Node *ASSName* *Instance* Value that is entered by the user at the time the instance is created *Attribute:Name* Value the Attribute with the name "Name" *UID* DBA generated Unique Identifier per AS Object *XHLAttrix AnyXML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node seqnum="34"> %B@CalcS71SOOAddr (oTag, "%Instance%", "%Address%") 80% *ASSName* Name of the AS Node *ASSName* *ASSName* Name of the AS Node *ASSName* Name of the Base and the bit is entered by the user at the time the instance is created *Assname* Name of the AS Node *ASSName* Value of the Attribute with the name Name' *UID* DBA generated Unique Identifier per AS Object *OPC DataTyppe*</node>	General	Instance Source	Connection	Addressing							
* ASName* Name of the AS Node * ASOName* Name of the instance * Instance* Value that is entered by the user at the time the instance is created * ALTE: DUTE: Name* Value of the Attribute with the name 'Name' * UID* DBA generated Unique Identifier per AS Object * XXII.Attrix Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node 34''="" seqnum=""> AS Object Tag Address Syntax *@@CalcS71500Addr (oTag, "%Instance*", "%Address*") @@% * ASName* Name of the AS Node * ASName* Name of the Tag whose address is being calculated * ASName* Name of the Tag * TagName* Name of the Address field of the Tag * Address* Value of the Address field of the Tag * Address* Value of the Address field of the Tag * Attribute: Name* Value of the Address field of the Tag * UID* DBA generated Unique Identifier per AS Object * OPC Data Type * OPC UA Address Value for the tag type (e.g. BIT=11, FLOAT=4) * UID* DPC UA Address Value * UID* OPC UA Address Value * UIN* OPC UA Address Value * UANamespace* OPC UA Address Value * UPC OAd</node>	Allow instance addres	es to be defined when the Ins	tance is created (value used	by %Instance%)							
* ASName* Name of the AS Node * ASOName* Name of the instance * Instance* Value that is entered by the user at the time the instance is created * Attribute:Name* Value of the Attribute with the name Name' * UTD* DBA generated Unique Identifier per AS Object *XMLAttri Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node 34''="" seqnum=""> AS Object Tag Address Syntax *8@CalcS71500Addr (oTag, "*Instance*", "*Address*") @@* *ASOName* Name of the AS Node *ASOName* Name of the instance *Instance* Value that is entered by the user at the time the instance is created *Instance* Value that is entered by the user at the time the instance is created *Instance* Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) *Attribute:Name* Value of the Attribute with the name Name' *UID* DBA generated Unique Identifier per AS Object *OPCDataType* OPC UA Address Value *UAType* OPC UA Address Value *UAType* OPC UA Address Value *UAValue* OPC UA Address Value *UAValue* OPC UA Address Value *UAValue* OPC UA Addr</node>	AS Object Instance Add	ress Syntax (only for non-OPC	Channels)								
* ASOName* Name of the instance * Instance* Value that is entered by the user at the time the instance is created * Attribute:Name* Value of the Attribute with the name "Name" * WilhAttri> DBA generated Unique Identifier per AS Object * XMLAttri> Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node seqnum="34"> AS Object Tag Address Syntax *@@CalcS71500Addr(oTag, "%Instance*", "%Address*") @@% * ASOName* Name of the AS Node * ASOName* Name of the Tag Moose address is being calculated * Aldress? Value that is entered by the user at the time the instance is created * Address? Value of the Address field of the Tag * Field#TagAddress Value of the Attribute with the name Name' * UD1* DBA generated Unique Identifier per AS Object * OPCDataType* OPC DAta Type number for the tag type (e.g. BIT=11, FLOAT=4) * UDANamespace* OPC UA Address Namespace * UD2 OPC UA Address Value * UAXvalue* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> * UD2 OPC UA Address Namespace * UD2 OPC UA Address Value * UAXvalue* Any XML Attribute (e.g. %SeqNum% would us</t></node>		\$00CalcS715	DOASOAddr ("% Instan	nce%")00%		ŝ					
Instance Value that is entered by the user at the time the instance is created *Attribute:Neame* Value of the Attribute with the name 'Name' *UID* DBA generated Unique Identifier per AS Object *XIILAttr* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node seqnum="34"> AS Object Tag Address Syntax *88CalcS71S00Addr (oTag, "%Instance*", "%Address*") 88* *ASName* Name of the AS Node *ASName* Name of the instance *Instance* Value that is entered by the user at the time the instance is created *Instance* Name of the instance *ASName* Name of the instance *Instance* Value that is entered by the user at the time the instance is created *Instance* Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) *Attribute:Name* Value of the Attribute with the name Name' *UID* DBA generated Unique Identifier per AS Object *OPCDataType* OPC UA Address Value *UID* DBA generated Unique Identifier per AS Object *UID* DPC DAS Type number for the tag type (e.g. BIT=11, FLOAT=4) *UNAmespace @UC UA Address Value *UAVa Lue* OPC UA Address Value <td>*ASName*</td><td>Name of the AS No</td><td>de</td><td></td><td></td><td>_</td></node>	*ASName*	Name of the AS No	de			_					
Attribute:Name Value of the Attribute with the name "Name" *UID* DBA generated Unique Identifier per AS Object *XMLAttrix Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node seqnum="34"> AS Object Tag Address Syntax *889CalcS71500Addr (oTag, "*Instance*", "*Address*") 88* *ASSName* Name of the AS Node *ASSName* Name of the instance *Instance* Value that is entered by the user at the time the instance is created *TagName* Name of the Tag whose address is being calculated *Address* Value of the Attribute with the name "Name" *UID* DBA generated Unique Identifier per AS Object *OPCDataType* OPC UA Address Namespace *UID* DBA generated Unique Identifier per AS Object *OPCDataType* OPC UA Address Value *UAName space* OPC UA Address Value *UAName space* OPC UA Address Value *UAName space* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> *UAName space* OPC UA Address Value *UAName space* OPC UA Address Value *UAName space* OPC UA Address Value *UAName Syntex (only for OPC A&E Channels) *<td>*ASOName*</td><td>Name of the instanc</td><td>e</td><td></td><td></td><td></td></t></node>	*ASOName*	Name of the instanc	e								
UID DBA generated Unique Identifier per AS Object *XNLAtts* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node seqnum="34"> AS Object Tag Address Syntax *@@CalcS71500Addr (oTag, "%Instance*", "%Address*")@@% *ASOName* Name of the AS Node *ASOName* Name of the instance *Instance* Value that is entered by the user at the time the instance is created *TagName* Name of the Tag whose address is being calculated *Address* Value of the Address field of the Tag *Field#TagAddress* Value of the Address field of the Tag *Ltribute:Name* Value of the Address field of the Tag *Ltribute:Name* Value of the Address field of the Tag *Ltribute:Name* Value of the Address field of the Tag *Ltribute:Name* Value of the Address field of the Tag *UANamespace* OPC UA Address Namespace *UANamespace* OPC UA Address ValueRank *UAValue* OPC UA Address Value *UAValue* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"></t></node>	<pre>%Instance%</pre>	Value that is entered	d by the user at the time the in	istance is created							
*XMLAttribute Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <node seqnum="34"> AS Object Tag Address Syntax *88@CalcS71500Addr (oTag, "*Instance*", "*Address*") 80* *ASName* Name of the AS Node *ASOName* Name of the instance *Instance* Value that is entered by the user at the time the instance is created *TagName* Name of the Tag whose address is being calculated *Address* Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) *Attribute:Name* Value of the Attribute with the name "Name" *UID* DBA generated Unique Identifier per AS Object *OPCDataType* OPC UAA Address Value *UAType* OPC UA Address Value *UAType* OPC UAA Address Value *XMLAttribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> *ASIName* Name of the AS Node *ASIName* Name of the AS Node *ASIName* Name of the AS Node *ASIName* Value of the Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> *ASIName* Name of the AS Node *ASIName* Name of the AS Node *ASIName* Value of the Attribute with the name Name' Value</t></t></node>	<pre>%Attribute:Na</pre>	mes Value of the Attribut	e with the name 'Name'								
AS Object Tag Address Syntax *@@CalcS71500Addr (oTag, "%Instance%", "%Address%") @@% *ASName% Name of the AS Node *ASOName% Name of the AS Node *ASOName% Name of the Tag whose address is being calculated *TagName% Name of the Tag whose address is being calculated *Address% Value that is entered by the user at the time the instance is created *Address% Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) *Attribute:Name% Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) *Attribute:Name% Value of the Attribute with the name Name' *UID% DBA generated Unique Identifier per AS Object *OPCDataType% OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) *UANamespace% OPC UA Address Value *UAType% OPC UA Address Value *XNLAttribute in OPC UA Address Value *XNLAttribute in OPC UA Address Value *XNLAttribute in OPC A&E Channels) *ASOName% Name of the AS Node *ASOName% Name of the AS Node *ASOName% Name of the AS Node *ASOName% Name of the AStribute with the name Name'	%UID%	DBA generated Unio	que Identifier per AS Object								
%88Ca1cS71500Addr (oTag, "%Instance%", "%Address%")88% %ASName% Name of the AS Node %ASName% Name of the instance %Instance% Value that is entered by the user at the time the instance is created %TagName% Name of the Tag whose address is being calculated %Address% Value of the Address field of the Tag %Field#TagAddress% Value of the Address field of the Tag %Field#TagAddress% Value of the Address Value with the name Name' %UAVpe% DBA generated Unique Identifier per AS Object %OPCDataType% OPC Data Type number for the tag type (e.g. BIT=11, FLDAT=4) %UAType% OPC UA Address Value %UAType% OPC UA Address Value %UAType% OPC UA Address Value %UAValue% OPC UA Address Value %UAValue% OPC CA&E Channels) %ASOName% %ASOName% Name of the AS Node %ASOName% Name of the Atribute with the name Name'	*XMLAttr*	Any XML Attribute (e	e.g. %SeqNum% would use th	e value of 34 from this XML:	<node 34"="" seqnum="34</td><td>''></td></tr><tr><td>* ASName* Name of the AS Node * ASOName* Name of the instance * Instance* Value that is entered by the user at the time the instance is created * TagName* Name of the Tag whose address is being calculated * ASOName* Value of the Address field of the Tag * Field#TagAddress* Value of the Address field of the Tag * Field#TagAddress* Value of the Address Value of the Attribute with the name *Name' * UD* DBA generated Unique Identifier per AS Object * OPC Data Type OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) * UAType* OPC UA Address Value * UAName* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <T SeqNum="> * ASOName* Name of the AS Node * ASOName* Name of the Attribute with the name *Name' * ASOName* Name of the Attribute with the name *Name'</node>	AS Object Tag Address	Syntax				
* ASOName* Name of the instance * Instance* Value that is entered by the user at the time the instance is created * TagName* Name of the Tag whose address is being calculated * Address* Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) * Attribute:Name* Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) * Attribute:Name* Value of the Attribute with the name 'Name' * UID* DBA generated Unique Identifier per AS Object * OPCDataType* OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) * UAType* OPC UA Address ValueRank * UAType* OPC UA Address ValueRank * UAValue* DPC UA Address ValueRank * UAValue* OPC UA Address Value * XMLAttribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> * ASOName* Name of the AS Node * ASOName* Name of the AS Node * ASOName* Name of the Attribute with the name 'Name'</t>	4	@@CalcS71500Addr()	oTag,"%Instance%",	."%Address%")@@%		N					
% Instance% Value that is entered by the user at the time the instance is created % TagName% Name of the Tag whose address is being calculated % Address% Value of the Address field of the Tag % Field#TagAddress% Value of the Address field of the Tag % Field#TagAddress% Value of the Address field of the Tag % Field#TagAddress% Value of the Address field of the Tag % Address% Value of the Attribute with the name 'Name' % UD% DBA generated Unique Identifier per AS Object % OPCDataType% OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) % UANamespace% OPC UA Address ValueRank % UAValue* OPC UA Address Value % UAValue* OPC UA Address Value % XMLAttres Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t 34''="" seqnum=""> OPC A&E Source Syntax (only for OPC A&E Channels) * % ASOName* Name of the AS Node % ASOName* Name of the Attribute with the name 'Name'</t>	<pre>%ASName%</pre>										
% TagName* Name of the Tag whose address is being calculated % Address* Value of the Address field of the Tag % Field#TagAddress* Value of the Address field of the Tag % Attribute:Name* Value of the Attribute with the name 'Name' % UDD* DBA generated Unique Identifier per AS Object % OPCDataType* OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) % UAValue* OPC UA Address ValueBank % UAValue* OPC UA Address ValueBank % UAValue* OPC UA Address Value % XNLAttribute (e.g. %SeqNum% would use the value of 34 from this XML: <t 34''="" seqnum=""> OPC A&E Source Syntax (only for OPC A&E Channels) * % ASSName* Name of the Attribute with the name 'Name' % ASSName* Name of the Attribute with the name 'Name'</t>	*ASOName*										
* Address* Value of the Address field of the Tag * Field#TagAddress* Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) * Attribute::Name* Value of the Attribute with the name 'Name' * UD* DBA generated Unique Identifier per AS Object * OPCDataType* OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) * UANamespace* OPC UA Address ValueRank * UAX alue* OPC UA Address ValueRank * UXValue* OPC UA Address Value * XNLAttra* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> OPC A&E Source Syntax (only for OPC A&E Channels) * ASName* Name of the AS Node * ASSName* Name of the Attribute with the name 'Name'</t>	<pre>%Instance%</pre>										
* Field#TagAddress Value that is entered by the user in the DBA for each tag of the instance (if more than 1 field) * kttribute:Name* Value of the Attribute with the name 'Name' * UID* DBA generated Unique Identifier per AS Object * OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) * UAType* OPC DAt Address ValueRank * UAType* OPC UA Address ValueRank * UAType* OPC UA Address Value * ASINAME* Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> OPC CA&E Source Syntax (only for OPC A&E Channels) * * ASIName* Name of the ASIN ode * * ASIName* Name of the Attribute with the name 'Name' Value of the Attribute with the name 'Name'</t>	%TagName%	-	-	ted							
* Attribute : Name* Value of the Attribute with the name 'Name' * UID* DBA generated Unique Identifier per AS Object * OPCDataType* OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) * ULANamespace* OPC UA Address Namespace * UAValue* OPC UA Address ValueRank * UAValue* OPC UA Address Value * XMLAttribute (e.g. %SeqNum% would use the value of 34 from this XML: <t 34''="" seqnum=""> OPC A&E Source Syntax (only for OPC A&E Channels) * ASOName* * ASOName* Name of the AS Node * ASOName* Value of the Attribute with the name 'Name'</t>			-								
% UID% DBA generated Unique Identifier per AS Object % OPCDataType% OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) % UANamespace% OPC UA Address Namespace % UAValue% OPC UA Address ValueRank % UAValue% OPC UA Address ValueRank % UAValue% OPC UA Address ValueRank % UAValue% OPC UA Address Value % XMLAttrs Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> OPC A&E Source Syntax (only for OPC A&E Channels) % &SOName % Name of the AS Node % &SOName % Name of the instance % &Lttribute:Name% Value of the Attribute with the name 'Name'</t>				ach tag of the instance (if mo	re than 1 field)						
\$ OPCDataType\$ OPC Data Type number for the tag type (e.g. BIT=11, FLOAT=4) \$ UANamespace\$ OPC UA Address Namespace \$ UAType\$ OPC UA Address ValuePank \$ UAValue\$ OPC UA Address Value \$ XXNLAttra Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> OPC A&E Source Syntax (only for OPC A&E Channels) \$ ASName\$ Name of the AS Node \$ ASNAme\$ Name of the instance \$ Attribute : Name\$ Value of the Attribute with the name Name'</t>											
* UANamespace OPC UA Address Namespace * UAType* OPC UA Address ValueRank * UAValue* OPC UA Address ValueRank * UAValue* OPC UA Address Value * XMLAttribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> OPC A&E Source Syntax (only for OPC A&E Channels) * ASName* Name of the AS Node * ASOName* Name of the instance * Attribute : Name* Value of the Attribute with the name 'Name'</t>											
UAType DPC UA Address ValueRank *UAType* DPC UA Address Value *UAType* DPC UA Address Value *XNLAttriant OPC UA Address Value *XNLAttriant Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> DPC A&E Source Syntax (only for OPC A&E Channels) *ASOName* *ASOName* Name of the AS Node *ASOName* Name of the instance *Attribute : Name* Value of the Attribute with the name 'Name'</t>				=11, FLUAI =4J							
UAValue DPC UA Address Value *XMLAttr> Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> DPC A&E Source Syntax (only for OPC A&E Channels) *ASOName* *ASOName* Name of the AS Node *ASOName* Name of the instance *Attribute : Name* Value of the Attribute with the name 'Name'</t>											
*XNLAttri Any XML Attribute (e.g. %SeqNum% would use the value of 34 from this XML: <t seqnum="34"> OPC A&E Source Syntax (only for OPC A&E Channels) ************************************</t>											
DPC A&E Source Syntax (only for DPC A&E Channels) *ASOName* Name of the AS Node *ASOName* Name of the instance *Attribute : Name* Value of the Attribute with the name 'Name'				e value of 34 from this XML: ∙	<t seanum="34"></t>						
%ASName% Name of the AS Node %ASName% Name of the AS Node %ASOName% Name of the instance %Attribute:Name% Value of the Attribute with the name 'Name'											
\$ASName\$ Name of the AS Node \$ASOName\$ Name of the instance \$Attribute:Name\$ Value of the Attribute with the name 'Name'	DPC A&E Source Synta	(only for OPC A&E Channels)				_					
\$ASOName\$ Name of the instance \$Attribute:Name\$ Value of the Attribute with the name 'Name'						5					
<pre>%Attribute:Name% Value of the Attribute with the name 'Name'</pre>											
Double click blue text to insert at cursor location	<pre>%Attribute:Na</pre>	me% Value of the Attribut	e with the name 'Name'								
Double click blue text to insert at cursor location											
Double click blue text to insert at cursor location											
	Double click blue text to	insert at cursor location									

Here, the labels "CalcS71500ASOAddr" and "CalcS71500Addr" are the names of the two scripts which calculate exactly the addresses of the tags. These scripts can be found in Chapter "6 A". Copy these into the corresponding script file for your AS node type. The scripts can be inserted at the end of the script file, in the "USER DEFINED FUNCTIONS" area.

The script file can be found in the following directory:

"C:\Program Files (x86)\SIEMENS\DBA\<Name of AS node type>\<Name of AS node type>Scripts.txt"

In our example:

"C:\Program Files (x86)\SIEMENS\DBA\S71500\S71500Scripts.txt"

Without going into the details of these scripts, it is possible to see the result in the WinCC tag manager after successful configuration (Figure 4–18). It can be seen that, based on the DB offset value of 0 entered during the instance definition (Figure 4–16), the byte offset for each tag has been calculated and entered as an address.

4 Configuring an S7-1500 package unit

Figure	4–18
--------	------

MinCC Configuration Studio	a WinCC Configuration Studio									
Datei Bearbeiten Ansicht ?										
Variablenhaushalt «	165	S7_1500 (Su	cherg	gebnisse	e)					
P		Name			Datentyp	Länge	Formatanpassung	Verbindung Gruppe	Adresse	Lineare Skalierung
🗄 🎯 Interne Variablen	1	Plant1/WATER	L1001	.AH	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD16	
Image: SIMATIC S7 Protocol Suite	2	Plant1/WATER	L1001	.AL	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD4	
SIMATIC S7-1200, S7-1500 Channel	3	Plant1/WATER	L1001	.MLH	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD22	
- M OMS+	4	Plant1/WATER	L1001	.MLL	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD26	
S7_1500	5	Plant1/WATER	L1001	.OUT_AH	Binäre Variable	1		S7_1500	DB101,D20.3	
Strukturvariablen	6	Plant1/WATER	L1001	.OUT_AL	Binäre Variable	1		S7_1500	DB101,D20.0	
	7	Plant1/WATER	L1001	.OUT_WH	Binäre Variable	1		S7_1500	DB101,D20.2	
	8	Plant1/WATER	L1001	.OUT_WL	Binäre Variable	1		S7_1500	DB101,D20.1	
	9	Plant1/WATER	L1001	.UNIT	Textvariable 8-Bit Zeichensatz	32		S7_1500	DB101,DBB30	
	10	Plant1/WATER	L1001	.VALUE	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD0	
	11	Plant1/WATER	L1001	.WH	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD12	
	12	Plant1/WATER	L1001	.WL	Gleitkommazahl 32-Bit IEEE 754	4	FloatToFloat	S7_1500	DB101,DD8	

Note

Note that the addresses in WinCC and in the SQL database are different. With the aid of the scripts, DBA creates the addresses just as they appear in the SQL database (e.g., "0001:CL:%DB101.%DD16"). They are then automatically displayed correctly in WinCC (e.g., "DB101,DD16").

In cases where the address syntax in your SQL database differs from our example, you must adapt the scripts at the appropriate points ("sAddr" tag).

Unstructured program design in the S7-1500 PU

If the S7-1500 program to be integrated has been created in an unstructured manner and no rules for addressing can be derived, you can set the parameters for the addresses of all the tags when an instance is being created.

- Enter the following syntax in the AS object instance address text field (Figure 4–19): 0001:CL:%Instance%
- Note With this setting, the address must be entered as follows when creating an AS object in DBA at a later time (Chapter "4.6 Creating ASO instances"): %DBxxx.%Dxxx.

🚍 Single Inst	tance Editor	×
General		
Name:	L1001	
Туре:	S7Meas	•
Address:	%DB101.%DBB0	
	<u>o</u> k	<u>C</u> ancel

- Enter the following syntax in the tag address text field (Figure 4–19): 0001:CL:%DBNum#TagAddress%.%DBOffset#TagAddress%
- Note With this setting, a new "Address" tab with the columns TagName, DBNum and DBOffset is available when you assign the ASO instance parameters at a later time. Here, you can individually assign the address parameters for each tag

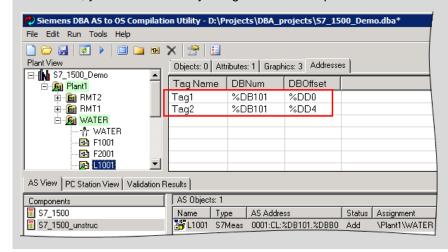


Figure 4–19

7 Allow instance addresses to be AS Object Instance Address Syn	defined when the Instar				_
AS Object Instance Address Syn		ice is created (value used l	by %Instance%)		
	tax (only for non-OPC Ch	annels)	· · ·		
		:CL:%Instance%			5
ASName	Name of the AS Node				_
ASOName	Name of the instance				
Instance		, the user at the time the in	stance is created		
<pre>%Attribute:Name%</pre>	Value of the Attribute w				
UID		Identifier per AS Object			
XMLAttr			e value of 34 from this XML:	<node <="" seqnum="34" td=""><td>5</td></node>	5
40.01: 17 AU					_
AS Object Tag Address Syntax-	· CL · DRNm#Tear	ddress%.%DBOffs	at #Tag bildy aget		5
1	Name of the AS Node	uuresst.tpb0IIS	ec#idyAuuress4		
ASName	Name of the instance				
ASOName		. No a constant share the state	stances in an advant		
<pre>%Instance%</pre>		the user at the time the in e address is being calculat			
% TagName%			eu		
<pre>%Address%</pre>	Value of the Address fie		ala bara af Nara Santana ar fitan	and the second second	
%Field#TagAddress%	Value of the Attribute w		ach tag of the instance (if mo	ore (nan 1 rield)	
<pre>%Attribute:Name%</pre>					
\$UID\$		ldentifier per AS Object for the tag type (e.g. BIT:	-11 ELOAT-4)		
<pre>%OPCDataType%</pre>	OPC UA Address Name		=11, FLUA1=4j		
UANamespace	OPC UA Address Name OPC UA Address Value				
UAType	OPC UA Address Value OPC UA Address Value				
<pre>%UAValue% %XMLAttr*</pre>			e value of 34 from this XML:	<t sednum="34"></t>	
OPC A&E Source Syntax (only fo	r OPC A&E Channels)—				
		%ASOName%			K)
ASName	Name of the AS Node				
<pre>%ASOName%</pre>	Name of the instance				
<pre>%Attribute:Name%</pre>	Value of the Attribute w	ith the name 'Name'			
Double click blue text to insert at	cursor location				
					_

4.4 Creating the AS node

All the created AS node types are stored in the OPEN OS system folder. From these AS node types, you can now create instances in DBA that map the controllers of the third-party system.

Once you have generated the AS node type, you can then create the AS node in the DBA.

- 1. Open a new or existing project in DBA.
- 2. In the "AS View" tab of the component view, add a new AS node. Select the "Add AS Source Node..." command in the shortcut menu.
- 3. Select the AS node type that was generated.

Figure 4-20

AS View PC Station View Valid	lation Results	1			
Components		AS Objects:	0		
		Name		Туре	
Add AS Source Node	Select a	an AS Type			X
		type of AS Noo	le to create:		
	System Ty	ре		AS Type	▲
	HIMA2 MODBUST	TCP		HIMA2 MODBUSTCP	
	MYOPC OPC Serve	er		MYOPC OPC Server	
	S71500 S7RAND			S71500 S7RAND	
Include All Resources	S7STRUC TEST1			S7STRUC TEST1	
Include Assigned	TEST2 TEST3			TEST2 TEST3	
Include Only Changes	•				
			<u>0</u> K	<u>C</u> ancel	

- 4. Assign a name for the station.
- 5. Depending on the setting that was chosen during the generation of the AS node type, you may still be able to adjust the connection name and the addressing.

Figure 4–21				
🖭 PCS 7 - OPEN O	S S71500 AS Properti	25	×	
General	Connection			
AS Node Name:	S71500			
	* All types and instanc	es must be manually configured.		
	🖭 PCS 7 - OPEN OS S	71500 AS Properties		×
	General	Connection		
	Channel Name:	SIMATIC S7-1200, S7-1500 Channel		
	Channel Unit:	OMS+		
	Connection Name:	%ASName%		
	Connection Parameter:	4!::::S7_1500_connl::172.10.1.67:SIMATIC-	ROOT-HMI:!hFa	aiaZYSIhbJlcV
			<u> </u>	<u>C</u> ancel

The new AS node has now been crated in the DBA AS View.

4.5 Generating AS object types

The AS object types in the DBA map the function blocks from the AS. An ASO type can be instantiated in the DBA as often as needed. Changes made to a type effect all instances of the type. The following steps use a block for an analog value measuring point to demonstrate how to generate this type. These tasks should then be carried out for all required blocks.

The data established in the chapter "4.1 Acquiring data" are the basis for generating the ASO types.

An ASO type can contain the following objects:

- Tags (type and address)
- Messages (message text, triggering event, accompanying values)
- Enumerations
- Attributes
- Runtime scripts

Start the DBA Type Editor by selecting the "Edit AS Object Types..." command from the AS node shortcut menu.

Figure 4–22					
AS View PC Station Vi	ew Validation Results	:			
Components		AS Objects: 0			
S71500 (changed)	Add AS Source Node	Nomo 2		Туре	
	Edit Object Propertie Update Controller O Update All Controlle	bjects			
	Delete		<u> </u>		
	Reset Attributes to	AS Default			
	Create New AS Obje				
	Export Instances to Import Instances fro				
 Include All Resource Include Assigned 	Edit AS Object Type	·S…			
Include Only Change	Auto-Assign OPC Re	esource to Plant View			
	Inspect Objects				
	Edit Attributes				
-	Export Attributes Import Attributes				

Note The AS object parameters are assigned for an AS node but are project specific. If you wish to reuse the objects in another DBA project, which uses the same AS node type, you can use the Export/Import function for ASO types.

Some attributes and tags in DBA must be added to the AS objects so that the thirdparty system objects behave like PCS 7 objects at a later stage. For this, the following types have already been created by default in DBA:

- AlarmGroupDisplaySupport
- CommonMembers
- SymbolAndFaceplateExtendedAttributes

These types introduce tags or attributes which are required for managing alarms or for managing block icons or faceplates. They can be included in any type that these functionalities are intended to work with.

Figure 4–23												
🛢 DBA Type Editor												
System		<u> </u>	<u>⊡M</u> ess	ages	🧷 Attributes) U	Includes		Docs) 🔲 OS <u>B</u>	untime	
\$71500		Include >:	>	Name >>		Type >>			Int/Ext>>			
Types\Enumerations	Parent Typ)e		Name		Туре		Int/Ext		Ŧ	
Support GroupDisplaySupport CommonMembers		AlarmGrou	pDisplaySu	ipport	EventState	\$	SDWORD		Internal			
SymbolAndFaceplateExtendedAttributes												
											*	
											1	
Browse Reference												
										Þ		
	💌 S	how All Inc	luded Tag	s			<u>A</u> dd		<u>E</u> dit	Dele	te	
Tools						<u>0</u> K		<u>C</u> anc	el	Help		
Last Change: 01.01.2000 01:00:00												

The following sections contain more detailed descriptions of the "Variables, Messages, Attributes, Includes, Docs and OS Runtime" tabs. In the "Docs" tab you can link descriptions to your types in HTML format, and in the "OS runtime" tab you can add scripts for the OS runtime. You can find more detailed information about this topic in the "PCS 7 Open OS DBA Type Editor" manual.

4.5.1 Creating a new type

To create a new type in the DBA Type Editor, click the "Add New Type" button.

The dialog window that is then displayed prompts you to enter the following data:

- AS Type Name Choose a name that creates a relation to the object in the AS.
- HMI Type Name Choose the same name that you assigned for the HMI symbol. It is recommended additionally to work in a reference to the name of the third-party system.

Figure 4	1–24
----------	------

🔳 DBA Type Editor						_	
System	≣ <u>I</u> ags ⊡Mess	ages 🖉 🖉 At	tributes 🕻	<u>5 I</u> ncludes) <u>Q</u> Do	cs 🛾 🔲 OS <u>B</u> i	untime
\$71500	Include >>	Name >>		Туре >>	l Ir	it/Ext>>	
Types\Enumerations	Parent Type	Name	Туре	Int/Ext	Address	Text	Ŧ
🚰 AlarmGroupDisplaySupport	O AlarmGroupDisplay	EventState	SDWORD	Internal			
🚰 CommonMembers	CommonMembers	#blocktype	TEXTREF	Internal		@TYPE_NAME	
🚰 MEAS*	CommonMembers	#comment	TEXTREF	Internal		@INSTANCEN	T
🐕 SymbolAndFaceplateExtendedAttributes	CommonMembers	#areaname	TEXTREF	Internal			
Ge ₽₩X BB B	Type Properties neral Type Name S7MEAS S7MEAS Can be u Cocked	sed by Injecter	d Instances			.dit Dele	♥ ♥
Last Change: 06.05.2014 15:25:44			<u>0</u>	ĸ	<u>C</u> ancel	Help	

4.5.2 Including default types

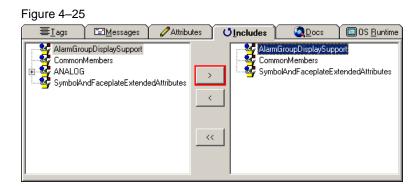
With type inclusion, it is possible to add variables and attributes to the new types to achieve specific functionalities.

The following types are included by default:

- CommonMembers
- SymbolAndFaceplateExtendedAttributes

All available types are displayed in the left-hand window. The right-hand window contains the types that are already included. If the block also features an alarm function, include the "AlarmGroupDisplaySupport" type.

- 1. Select the "AlarmGroupDisplaySupport" type in the left-hand window.
- 2. Include the type by clicking the button (right arrow).



The new type is now equipped with the basic functionality for PCS 7 blocks. You can remove the functionality of the included types by clicking the "Left arrow" button.

4.5.3 Creating tags

The tags form the interface between the OS and the automation system. You can configure the tags in the "Tags" tab in the DBA Type Editor.

lude >> Int Type mcGroupDispl monMembers monMembers monMemoM leas leas leas leas leas leas leas leas	Name >> Name EventState #blocktype #comment #areaname #StatusPerman #TextPermanent VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH	Type >> Type SDWORD TEXTREF TEXTREF TEXTREF DWORD TEXT8 TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Int/Ext Internal Internal Internal Internal Internal External External External External	Int/Ext >> Address 0 4	Require Text @TYPE_NAME @INSTANCEN Measured Value Limit Alarm Low Limit Warning L	ed >> 36 Optional R R R R R R R R R R R R R	
nGroupDispl monMembers monMembers monMemoM eas eas eas eas eas eas eas eas eas eas	EventState #blocktype #comment #areaname #StatusPerman #TextPermanent VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH	SDWORD TEXTREF TEXTREF TEXTREF DWORD TEXT8 FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal Internal Internal Internal Internal External Internal External Internal	0	@TYPE_NAME @INSTANCEN Measured Value Limit Alarm Low	R R R R R R R R R R	
monMembers monMembers monMemoM eas eas eas eas eas eas eas eas eas eas	#blocktype #comment #areaname #StatusPerman #TextPermanent VALUE#shortcut VALUE AL #shortcut AL WL#shortcut WL WH#shortcut WH	TEXTREF TEXTREF TEXTREF DWORD TEXT8 TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal Internal Internal Internal External Internal External Internal Internal	4	@INSTANCEN Measured Value Limit Alarm Low	R R R R R R R R R	
monMembers monMemoM monMemoM eas eas eas eas eas eas eas eas eas eas	#comment #areaname #StatusPerman #TextPermanent VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WHshortcut WH	TEXTREF TEXTREF DWORD TEXT8 TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal Internal Internal Internal External External External Internal	4	@INSTANCEN Measured Value Limit Alarm Low	R R R R R R R R R	
monMembers monMemoM leas leas leas leas leas leas leas leas	#areaname #StatusPerman #TextPermanent VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH	TEXTREF DWORD TEXT8 TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal Internal Internal External Internal External Internal	4	Measured Value	R R R R R R R	
monMemoM monMemoM leas leas leas leas leas leas leas leas	#StatusPerman #TextPermanent VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH	DWORD TEXT8 TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal Internal External Internal External Internal	4	Limit Alarm Low	R R R R R R	
monMemoM eas eas eas eas eas eas eas eas eas eas	#TextPermanent VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH#shortcut WH	TEXT8 TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal Internal External Internal External Internal	4	Limit Alarm Low	R R R R R	
eas eas eas eas eas eas eas eas eas eas	VALUE#shortcut VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH	TEXTREF FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal External Internal External Internal	4	Limit Alarm Low	R R R R	
eas eas eas eas eas eas eas eas eas	VALUE AL#shortcut AL WL#shortcut WL WH#shortcut WH	FLOAT TEXTREF FLOAT TEXTREF FLOAT TEXTREF	External Internal External Internal	4	Limit Alarm Low	R R R	
eas eas eas eas eas eas eas eas	AL#shortcut AL WL#shortcut WL WH#shortcut WH	TEXTREF FLOAT TEXTREF FLOAT TEXTREF	Internal External Internal	4		R R	
eas eas eas eas eas eas	AL WL#shortcut WH#shortcut WH	FLOAT TEXTREF FLOAT TEXTREF	External Internal			R	- -
eas eas eas eas eas	WL#shortcut WL WH#shortcut WH	TEXTREF FLOAT TEXTREF	Internal		Limit Warning L		
eas eas eas eas	WL WH#shortcut WH	FLOAT TEXTREF			Limit Warning L	D	
eas eas eas	WH#shortcut WH	TEXTREF	External		Linic waning L	п	
eas eas	WH			8		R	
eas			Internal		Limit Warning H	R	
		FLOAT	External	12		R	
eas	AH#shortcut	TEXTREF	Internal		Limit Alarm High	R	
	AH	FLOAT	External	16		R	
eas	OUT_AL#short	TEXTREF	Internal		Alarm Low	R	
eas	OUT_AL	BIT	External	20.0		R	
eas	OUT_WL#shor	TEXTREF	Internal		Warning Low	R	
eas	OUT_WL	BIT	External	20.1		R	
eas	OUT_WH#shor	TEXTREF	Internal		Warning High	R	
eas	OUT_WH	BIT	External	20.2		R	
eas	OUT_AH#short	TEXTREF	Internal		Alarm High	R	
eas	OUT_AH	BIT	External	20.3		R	
eas	MLH#shortcut	TEXTREF	Internal		Measuring Limit	R	
eas	MLH	FLOAT	External	22		R	
eas	MLL#shortcut	TEXTREF	Internal		Measuring Limit	R	
eas	MLL	FLOAT	External	26		R	
eas	UNIT#shortcut	TEXTREF	Internal		Unit	R	
eas	UNIT	TEXT8	External	30		R	
ie ie ie ie	2865 2865 2865 2865 2865 2865 2865 2865	eas OUT_WH#shor eas OUT_WH eas OUT_AH#short eas OUT_AH eas MLH#shortcut eas MLH eas MLL eas MLL eas MLL eas UNIT#shortcut	bas OUT_WH#shor TEXTREF bas OUT_WH BIT bas OUT_AH#short TEXTREF bas OUT_AH BIT bas OUT_AH BIT bas OUT_AH BIT bas MLH#short.ut TEXTREF bas MLH#shortcut TEXTREF bas MLL#shortcut TEXTREF bas MLL#shortcut TEXTREF bas MLL#shortcut TEXTREF bas MLL FLOAT bas UNIT#shortcut TEXTREF	Bass OUT_WH#shor TEXTREF Internal Bass OUT_WH BIT External Bass OUT_AH#short TEXTREF Internal Bass OUT_AH BIT External Bass OUT_AH BIT External Bass MLH#shortcut TEXTREF Internal Bass MLH#shortcut TEXTREF Internal Bass MLL#shortcut TEXTREF Internal Bass MLL FLOAT External Bass UNIT#shortcut TEXTREF Internal	eas OUT_WH#shor TEXTREF Internal eas OUT_WH BIT External 20.2 eas OUT_AH#short TEXTREF Internal eas OUT_AH BIT External 20.3 eas OUT_AH BIT External 20.3 eas MLH#shortcut TEXTREF Internal eas MLH FLOAT External 22 eas MLL#shortcut TEXTREF Internal eas MLL FLOAT External 26 eas MLL FLOAT External 26 eas UNIT#shortcut TEXTREF Internal	eas OUT_WH#shor TEXTREF Internal Warning High eas OUT_WH BIT External 20.2 eas OUT_AH#short TEXTREF Internal Alarm High eas OUT_AH BIT External 20.3 eas OUT_AH BIT External 20.3 eas MLH#shortcut TEXTREF Internal Measuring Limit eas MLH FLOAT External 22 eas MLL FLOAT External 26 eas MLL FLOAT External 26 eas UNIT#shortcut TEXTREF Internal Measuring Limit	bass DUT_WH#shor TEXTREF Internal Warning High R bass DUT_WH BIT External 20.2 R bass DUT_AH#short TEXTREF Internal Alarm High R bass DUT_AH BIT External 20.3 R bass DUT_AH BIT External 20.3 R bass MLH#shortcut TEXTREF Internal Measuring Limit R bass MLH FLOAT External 22 R bass MLL FLOAT External 26 R bass UNIT#shortcut TEXTREF Internal Unit R

To add new tags, click the "Add" button. An additional dialog window is opened. In the "General" tab, you can enter the following data:

- Name of the tag
- Data type
- WinCC type (default is structure tag)
- Source (external, internal, indirect)
- Address
- Runtime options
- Inclusion rules

Use the data type as illustrated in Table 4–6 in the "Address Syntax" column below: Table 4–6

WinCC data type	Address Syntax	AS Data Type
Binary	D	Bit
Unsigned 8-bit	DBB	Byte
Signed 8-bit	DBB	Byte
Unsigned 16-bit	DBW	Word
Signed 16-bit	DBW	Word
Unsigned 32-bit	DD	Double word
Signed 32-bit	DD	Double word
Float 32-bit	DD	Double word
Float 64-bit	DD	Double word
Text 8-bit	DBB	Byte
Text 16-bit	DBW	Word

The entry in the "Address" field (Figure 4-27) is used in the script referred to in Chapter "4.3.4 Addressing tab" to calculate the tag addresses automatically.

In the "Advanced" tab, you can configure additional settings for the tag:

- Upper and lower limit
- Start value
- Auxiliary tags (description, unit, ...)
- Archiving

The "DBA Edit" option allows you to customize the specified start values at instantiation of the type.

Figure 4–27	
🚍 Tag Editor	×
General Advanced	
Name: VALUE	Tag Editor
DataType: FLOAT	General Advanced
WinCC Tag 🔿 None (no tag created) Type: 💿 Structure Attribute	- Attributes DBA Edit
Single Tag (not part of the second	
Source: 💿 External O Interna	
Address:	Start Value:
Runtime Options	Enable Parameter Readback:
Synchronize value with backup serv	Length:
Persist & Restore value between ser	Tracked Tags DBA Edit
,	🔽 %tagname%#shortcut: Measured Value 💷 🕅
	🗖 %tagname%#unit:
Accessible to OS Runtime Scripts	🗖 %tagname%#string_0:
Inclusion Rules G Required	🗖 %tagname%#string_1:
O Optional, default included	🗖 %tagname%#string_x:
C Optional, default NOT included	Tag Logging DBA Edit
	Frequency: OnChange
< >	Enabled by default
	Long Term (Store in Central Archive Server)
	< > <u>QK</u> <u>C</u> ancel

Click the "OK" button to close the dialog window and create the type tag.

4.5.4 Configuring alarms

In the "Messages" tab, you can configure the process messages of the AS block. The messages can be triggered by different tags or by means of an OPC A&E server. You can configure the following parameters:

- Alarm name
- Message text
- Message class
- Priority
- Trigger tag
- Associated message values

Figure 4–28

	≣∐ags	⊡ <u>M</u> essag	jes 🚺 🧷 A	ttributes	U Inclu	des	QDoc:	s 🎽 🖪	0S <u>R</u> untime
	Parent Type	Name	Display Name	Class	Туре	Enabled	Priority	T	rigger Tag
	S7Meas	Alarm High	AH	1 - Alarm	1	Yes	1	0	UT_AH
r S	S7Meas	Alarm Low	AL	1 - Alarm	2	Yes	1	C	IUT_AL
r T	S7Meas	Warning High	WH	2 - Warning	20	Yes	2	0	UT_WH
r	S7Meas	Warning Low	WL	2 - Warning	19	Yes	2	0	UT_WL
<									>
							Add	<u>E</u> dit	<u>D</u> elete

Click "Add" to configure new messages of the ASO type. A dialog with the following tabs will open:

- General
- Tags
- Process Vars
- Free Vars

General tab

Assign the following parameters in the "General" tab:

- Name
- Display name
- Message class
- Priority
- Message text

Figure 4–29

🖃 Message Edito	or - Alarm High		×
General	Tags	Process Vars	Free Vars
			DBA Edit
Name:	Alarm High		
Display Name:	AH		
Class:	1 - Alarm		
0.000.			▼
Туре:	1		• V
Enabled:	\mathbf{v}		$\overline{\mathbf{v}}$
Priority:	1		V
Text	Alarm High @1%s@ @	2%3.2f@ @3%s@	V
Batch Name:			
Operation:			
Lockable:	$\overline{\mathbf{v}}$		
- Inclusion Rules -			
C Required			
 Optional, defa Optional, defa 	ault included ault NOT included		
< >		<u>0</u> K	Cancel

Associated message values are integrated using the control commands known from PCS 7. For example, associated value 2 of the floating-point number type: @2%3.2f@.

The associated values must be declared in the "Process Vars" tab.

Tags tab

In the "Tags" tab, the triggering tags for the message in WinCC and the initiating trigger in the AS are configured.

Message Editor - A			
General	Tags	Process Vars	Free Vars
ontroller / Calculated			
	<u> </u>		
0	f_x		
ontroller Acknowled	ge Tag (Optional)		
ig:	-		
	Sense on 0		

Here you can configure the trigger and optional acknowledge in the controller for the OS – if the connected third-party system supports this. The message behavior then corresponds to that of the PCS 7 blocks. Messages are shown in the alarm group display and in the message line, and the "Loop in Alarm" function can be used.

With button "fx" any desired terms can be build. Basing of variables and limit values these terms calculate and – if necessary - activate the trigger of a message.

Process Variables tab

In this tab, you can configure the tags that you wish to use as associated values in your message texts. The system provides 10 fields.

You may select any of the tags you created when generating the type.

Ge	neral	Tags	Process Vars	Free Vars
P1:	OUT_AH#	shortcut		
P2:	VALUE			
P3:	UNIT			
P4:				
P5:				
P6:				
P7:				
P8:				
P9:				
P10:				

Free Variables tab

In the "Free Tags" tab, you can parametrize constantly recurring text blocks, for example. You can find further information in the WinCC Alarm Logging description.

4.5.5 Attributes

The attributes determine the behavior of a tag, a message or the instance of a type. Most attributes are automatically created during the generation of tags and messages or copied during the inclusion of other types. It is not necessary to make changes to these attributes. By default, attributes generated by the system are not displayed. You can display these attributes by activating the "Show All Attributes" selection box.

You will find detailed information about the attributes in the "PCS 7 Open OS DBA Type Editor" manual under chapter 6.5 "Working with the Attribute Editor".

≣∐ags	<u>⊡M</u> essages	🛛 🧷 Attributes 📗	<u>O</u> ln	cludes	0	<u>)</u> ocs		OS <u>R</u> untin
Parent Type		Name		Display Na	ame		Catego	<u> </u>
-	plateExtendedAttrib			SF			Graphi	
	plateExtendedAttrib			SN			Graphi	
CommonMembe		#comment		Comment			Graphi	cs
5 CommonMembe		ResetEAs		Reset Attr	butes to De	fault	DBA	
CommonMembe		UID		UID			Addres	
2 CommonMembe	18	InstallHook		InstallHoo			DBAT	ypeManag
S7Meas		VALUE#shortcut		VALUE - 0	Comments		Graphi	cs
S7Meas		AL#shortcut		AL - Comn	nents		Graphi	cs
S7Meas		WL#shortcut		WL - Com	ments		Graphi	cs
🖥 S7Meas		WH#shortcut		WH - Com	ments		Graphi	ics
🖺 S7Meas		AH#shortcut		AH - Comr	nents		Graphi	cs
🗑 S7Meas		OUT_AL#shortcu	ut	OUT_AL -	Comments		Graphi	cs
🖺 S7Meas		OUT_WL#shortc	ut	OUT_WL	 Comments 		Graphi	cs
S7Meas		OUT_WH#shorte	out	OUT_WH	- Comment:	s	Graphi	cs
🔓 S7Meas		OUT_AH#shorter	ut	OUT_AH	Comments		Graphi	cs
🔓 S7Meas		MLH#shortcut		MLH - Cor	nments		Graphi	ics
🖞 S7Meas		MLL#shortcut		MLL - Con	nments		Graphi	cs
🖞 S7Meas		UNIT#shortcut		UNIT - Co	mments		Graphi	cs
🖞 S7Meas		AL#Start		AL - Start 1	Value		Graphi	cs
🔓 S7Meas		WL#Start		WL - Start	Value		Graphi	cs
🔓 S7Meas		WH#Start		WH - Star	t Value		Graphi	cs
🖀 S7Meas		AH#Start		AH - Start	Value		Graphi	cs
🖞 S7Meas		Alarm Low - Class	3	Alarm Low	- Alarm Cla	ss	Alarm (Class
🖞 S7Meas		Alarm Low - Type		Alarm Low	- Alarm Typ)e	Alarm `	Туре
🔓 S7Meas		Alarm Low - Enab	oled	Alarm Low	- Alarm Ena	abled	Alarma	bility
🔓 S7Meas		Alarm Low - Priori	ty –	Alarm Low	- Alarm Pric	ority	Alarm F	Priority
S7Meas		Alarm Low - Text		Alarm Low	- Alarm Tex	(t	Alarm 1	Text
S7Meas		Alarm High - Class	s	Alarm High	n - Alarm Cla	ISS	Alarm (Class
S7Meas		Alarm High - Type		Alarm High	h - Alarm Typ	ре	Alarm [*]	Туре
S7Meas		Alarm High - Enat		-	n - Alarm En		Alarma	
S7Meas		Alarm High - Prior		-	n - Alarm Prie		Alarm F	-
S7Meas		Alarm High - Text		-	- Alarm Te:		Alarm `	
S7Meas		Warning Low - Cl		-	ow - Alarm (Alarm (
S7Meas		Warning Low - Ty		-	ow - Alarm [*]		Alarm `	
S7Meas		Warning Low - Er		-	ow - Alarm B	•••	Alarma	•••
S7Meas		Warning Low - Pr		-	ow - Alarm F		Alarm F	
S7Meas		Warning Low - Te	-	-	ow - Alarm ⁻	-	Alarm	
S7Meas		Warning High - C		-	ligh - Alarm		Alarm (
S7Meas		Warning High - T		-	ligh - Alarm		Alarm	
S7Meas		Warning High - E		-	ligh - Alarm		Alarma	
S7Meas		Warning High - P		-	ligh - Alarm		Alarm F	-
S7Meas		Warning High - T		-	ligh - Alarm	•	Alarm 1	
		a aning right f					- SQUIT	, ynv
Show All Attribute	s				Add	<u>E</u> d	it	D

Figure 4–32

4.6 Creating ASO instances

The ASO types created in the last chapter emulate the structures (UDTs) of the S7-1500. Using the ASO types, instances will now be generated that emulate the instances in the data blocks of the AS.

As established in the chapter "4.1 Acquiring data", two motors, two valves, five analog value measuring points and one sequence (SFC) are used for our example PU. You can generate these instances by following the steps below:

Open the AS node shortcut menu and select the "Create New AS Object..." command.

Figure 4	-33
----------	-----

AS View PC	Station View Validation Results					
Components	AS Object	s: O				
S71500 (ch			Туре			
	Add AS Source Node					
	Edit Object Properties					
	Update Controller Objects					
	Update All Controller Objects					
	Delete					
	Reset Attributes to AS Default	Single Instance Editor				
	Create New AS Object	General				
	Export Instances to CSV					
I Include All F	Import Instances from CSV	Name:	L1001			
Include Air	Edit AS Object Types	Туре:	S7MEAS			
Include Only	Auto-Assign OPC Resource to Plant View	Address:	100,0			
	Inspect Objects					
	Edit Attributes		 _0K Cancel			
	Export Attributes					
	Import Attributes					

Enter the following data in the "Single Instance Editor" dialog window:

- Name of the instance (choose a name that you can associate with the AS program)
- ASO type
- AS address

Note The "Address" text box is only available if you defined it when generating the AS type. In our case, the address string was as follows:

%@@CalcS71500ASOAddr("%Instance%")@@%

With %Instance%, we specified that the input was to occur in the "Address" field. By entering the DB number and the DB offset, the correct address can then be put together.

e.g.: DB100,DBB0 (WinCC) → 0001:CL:%DB100.%DBB0 (SQL database)

The addresses of the individual tags are calculated from the following address string:

%@@CalcS71500Addr(oTag,"%Instance%","%Address%")@@%

As above, each individual tag address is calculated from the DB number, the DB offset and additionally the address that was entered in the ASO type (Figure 4–27).

In the event that you have an unstructured S7-1500 program, take note of the corresponding paragraph in Chapter "4.3.4 Addressing tab".

Carry out these steps for all required objects.

Figure 4-34

AS View | PC Station View | Validation Results |

Components	AS Objects:	: 10			
🖬 S71500 (changed)	Name	Туре	AS Address	Status	Assignment
	💏 F1001	S7Meas	0001:CL:%DB101.%DBB80	Add	
	💏 F2001	S7Meas	0001:CL:%DB101.%DBB160	Add	
	L1001	S7Meas	0001:CL:%DB101.%DBB0	Add	
	📑 MOT1	S7Motor	0001:CL:%DB100.%DBB0	Add	
	MOT2	S7Motor	0001:CL:%DB100.%DBB2	Add	
	📑 P1001	S7Meas	0001:CL:%DB101.%DBB40	Add	
	🐉 P2001	S7Meas	0001:CL:%DB101.%DBB120	Add	
The shide All Deserves	🚰 SFC1	S7SFC	0001:CL:%DB400.%DBB0	Add	
Include All Resources ✓ Include Assigned ✓ Include Only Changes	🎥 VAL1	S7Valve	0001:CL:%DB100.%DBB12	Add	
	🚰 VAL2	S7Valve	0001:CL:%DB100.%DBB14	Add	

Whenever changes are made to the settings of the AS node, to the related types or to the created instances, the text "(changed)" is added to the AS node as a note.

Note With the "DBA Object Inspector", you can read out information about which data was created on the OS. For example, you can already verify whether the structure of the address syntax is correct. The command to open the "DBA Object Inspector" can be found in the shortcut menu of an AS node or an ASO instance. By calling the "Update Controller Objects" function from the AS node shortcut menu, you can update all instances of the AS object.

Components	Station View Validation Results			
S71500		ne	AS Address	Sta
	Add AS Source Node	hs	0001:CL:%DB101.%DBB80	10.0
	Edit Object Deservice	- as	0001:CL:%DB101.%DBB160	
	Edit Object Properties Update Controller Objects	as	0001:CL:%DB101.%DBB0	
	Update All Controller Objects	or	0001:CL:%DB100.%DBB0	
	opdate Air Controller Objects	or	0001:CL:%DB100.%DBB2	
	Delete	as	0001:CL:%DB101.%DBB40	
	Reset Attributes to AS Default	as	0001:CL:%DB101.%DBB120	
Include All F	Reset Attributes to AS Default	_	0001:CL:%DB400.%DBB0	
Include All F	Create New AS Object	/e	0001:CL:%DB100.%DBB12	
Include Assi	Export Instances to CSV	/e	0001:CL:%DB100.%DBB14	
Include only	Import Instances from CSV			
	Edit AS Object Types			_
	Auto-Assign OPC Resource to Plant View			
	Inspect Objects			
	Edit Attributes			
	Export Attributes			
	Import Attributes			

The configuration of the AS part for the S7-1500 package unit is now complete. The next steps can be found in the chapter "5 Configuring the PC station with the DBA".

5 Configuring the PC station with the DBA

The PC station in the DBA Editor serves as an interface to the operator system on which the third-party system is operated and monitored.

To create a PC station in the DBA, you first need a PCS 7 project complete with an operator station (OS) and a plant hierarchy. This can be a finished project that is to be expanded to include the functionality of the package unit, or you can create a new project.

5.1 Adding a PC Station

To create a new PC station in the DBA, follow the steps below:

- 1. Switch to the "PC Station View" tab.
- 2. Select the "Add PC Station" command in the DBA project shortcut menu.

Figure 5–1

AS View PC Station View Validation Re	esults
Nev	Name
Add PC Station	

- 3. Assign the name of your ES to the PC station.
- 4. Also add the name of the ES (target OS) at the "Computer Name" line in the "Value" column.

Figure 5–2

AS View PC Station View Validation Results

⊡- ∭ Neu	Name	Value	Category	Description
	Author		Standard	Name and/or department of the folder creator.
	🚰 Comment		Standard	Additional documentation.
	🚰 Computer Name	ES1	Standard	Name of the Computer.
1				

- 5. Select "Add Application" from the context menu of the PC station. A dialog window will then open.
- 6. Select the appropriate project type and click OK in the dialog window to confirm your selection. Name the application using the name of the corresponding OS project in the SIMATIC Manager.

Figure 5–3



- 7. You can link the DBA project to the PCS 7 OS project by entering the path for the OS project at the "Offline MCP File" line in the "Value" column.
- 8. By clicking the "..." button you can navigate to the OS project with the aid of a file selection dialog window and then insert the path string.

Figure 5–4

AS View	PC Station View	Validation Results	Ĺ

⊡ 🚺 Neu	Name	Value	Category	Description
🖻 黑 ES1	🚰 Author		Standard	Name and/or department of the
🔤 💭 OS(1)	🚰 Comment		Standard	Additional documentation.
	👋 Context Number	0	OS	Context Number
	📲 Language Dictionary		OS	Language dictionary used to ins
	🚰 Offline MCP File		OS	Engineering OS Project File Pat
	🛛 🍟 Online MCP File		OS	Target OS Project File Path
	🚰 Standby OS	<none></none>	OS	Name of the Standby OS
	🍟 Tag Logging Archive		OS	Archive for Tag Logging
	 			•

9. Update the project data in the DBA by selecting the "Refresh OS Cache" function in the shortcut menu. This function opens the PCS 7 OS project, reads the alarm classes, alarm types and tag logging archive information, and saves all of this information in the DBA project.

Figure 5–5

⊡- [N Neu	Name		Value	Category	Description
🖻 💂 ES1	🚰 Author			Standard	Name and/or department of the
	Edit Object Descention	1		Standard	Additional documentation.
-	Edit Object Properties	-	0	OS	Context Number
	Delete	inary		OS	Language dictionary used to ins
	Rename		D:\Projects\PCS7\	OS	Engineering OS Project File Pat
				OS	Target OS Project File Path
	Refresh OS Cache		<none></none>	OS	Name of the Standby OS
	Compile	chive		OS	Archive for Tag Logging
	•				
,					

10. Select a Tag Logging archive if one is present.

AS View PC Station View	Validation Results			
⊡ N eu	Name	Value	Category	Description
🖻 🖳 🧱 ES1	🚰 Author		Standard	Name and/or department of the
	🚰 Comment		Standard	Additional documentation.
	👋 Context Number	0	OS	Context Number
	📲 Language Dictionary		OS	Language dictionary used to ir
	🍟 Offline MCP File	D:\Projects\PCS7\	OS	Engineering OS Project File Pr
	🚰 Online MCP File		OS	Target OS Project File Path
	🚰 Standby OS	<pre><none></none></pre>	OS	Name of the Standby OS
	🍯 Tag Logging Archive	System_Archive 🛛 💌	OS	Archive for Tag Logging

5.2 Creating a plant hierarchy

In the DBA, you can use the plant hierarchy to define the picture hierarchy as it appears on the OS. It is possible to use and extend an existing hierarchy from a PCS 7 project or to generate a new hierarchy.

You will find detailed information in the PCS 7 Open OS DBA user manual. In this example, an existing PH will be extended.

5.2.1 Setting project properties

1. Click the "Change Project Properties" button.

Figure 5–7								
🗭 Siemens DBA AS to OS Compilation Utility - D:\Projects\DBA_projects\Neu.dba*								
<u>File E</u> dit <u>R</u> un <u>T</u> ools <u>H</u> elp								
Plant View	Objects: 0 Attributes: 0							
	Name	OS Tag Name	Status					

2. In the "General" tab, you can define the project and the project path with which you intend to synchronize the PH. With the "Automatic Synchronize with Simatic Plant View" option, you can trigger an automatic synchronization with each compiling operation.

Figure 5–8								
Project Options								
General Hierarchy								
Name D:\Projects\DBA_projects\Neu.dba								
Synchronize Simatic Project Hierarchy								
Simatic Project: color_gs_prj Browse								
Simatic Path: D:\Projects\PCS7\S71500_DEM0\color_gs\color_								
✓ Automatic Synchronize with Simatic Plant View Preview								
Enable MultiProject Features								
Apply <u>D</u> K <u>C</u> ancel								

3. In the "Hierarchy" tab, you can configure the PH settings. Here, you must choose the same settings as the ones in the PCS 7 project.

Figure	5–9											
Custom	ize Plar	nt Hie	rarchy						×			
Numb	ber of hie	erarchy	levels:	[3	<u>+</u>						
	l Setting:											
Le	evel		number aracters	Inclue H			With eparator	OS area				
	1:	24	÷	R	7		2	C				
	2:	24	+	Γ	-		2	0				
	3:	24	*	Γ	-		2	0				
	4:	24	🧮 Pro	oject Op	tions							X
	5:	24	Gen	eral Hie	erarchy							
	6:	24										
	7:	24	Ma: Lev		Max Charact	ers	Include In Name	With Sep	arator	OS Area		
	8:	24		Level 1	24	•		V		۲		
Prev	view:	111	_	Level 2	1.1	÷				0		
			•	Level 3 Level 4		÷				0		
	erive pic			Level 4	1	* *						
	Derive		0	Level 6	_	×						
	🖸 Mair 🔿 Mair		0	Level 7	- <u></u>	* *						
	O Mair		0	Level 8	24	* *						
				review	1111	1111	1111111111	111111/			 	
	🖸 Deriv			Short S	ymbol N	ames						
	C Deriv	ve PH										
	Mig	jrate di					Apply	<u>о</u> к		Cancel		
)K				[С	ancel	Help				

4. Click the "OK" button to confirm the settings.

5.2.2 Synchronizing the plant hierarchy

Select the "Synchronize Plant View" command in the DBA project shortcut menu.

Figure 5–10	
Siemens DBA AS to OS Compilation Utility - D:\	Projects\DBA_projects\
File Edit Run Tools Help	
□ □ □ □ ∞ × ! Ξ Ξ Plant View Objects: 1 A	
⊡[<mark>M</mark> S7_1500_Demo	1 OS Tag Na
Add Folder	
Expand Node	
Export Instances to CSV	
Import Instances from CSV	
Upgrade Instances to New Versions of Types	
Edit Attributes	
Export Attributes	
Import Attributes	
Export Plant View	
Import Plant View	
Readback Parameters from OS	
Synchronize Plant View	

In this case, the PH is read from the PCS 7 project and created in the DBA, since no hierarchy was present.



•							
🔗 Siemens DBA AS to OS Compilation Utility - D:\Projects\DBA_projects\S7_1500_Demo.dba							
<u>File E</u> dit <u>R</u> un <u>T</u> ools <u>H</u> elp							
🗋 🗁 🛃 💽 🕨 🔲 🔛 🖻	X 🚰 🗉						
Plant View	Objects: 1 Attrib	outes: 0					
□- 1 S7_1500_Demo				$ \rightarrow$			
🖃 👰 Plant1	Name	OS Tag Name	Status	AS			
	🕅 Plant1		Changed				
🗄 🖻 🛅 BMT1							
🗄 🖷 🙆 WATER							

5.2.3 Extending the plant hierarchy

You will now extend the loaded hierarchy to include the hierarchy of the third-party system. By selecting the "Add Folder" command in the shortcut menu, you can create additional hierarchy folders. Every new folder is automatically assigned an OS picture with the same name as the folder.

Add additional hierarchy folders according to your plant structure.

Figure 5–12 🗭 Siemens DBA AS to 05 Compilation Utility - D:\Projects\DBA_projects\S7_1500_D File Edit Run Tools Help) 🗁 🛃 🛃 🕨 🔲 💷 🛥 🗙 😁 🗄 Plant View Objects: 3 Attributes: 3 ⊡ 🚺 S7_1500_Demo OS Tag Name Name Status AS 🖻 👧 Plant1 Add Folde 🕂 💮 📴 F Add Picture 🗄 🛯 📴 F 🗄 -- 🙆 V Edit Folder Properties... Enable All Picture Changes Disable All Picture Changes Delete Rename

If you have not enabled automatic synchronizing, you should synchronize the PH with the OS project again after expanding it in the DBA.

Note PH synchronization can only be used to add folders. If you wish to remove existing folders, you must delete them in the SIMATIC project and in the DBA.

5.2.4 Assigning hierarchy folders to an OS

If multiple OS projects have been created, it is possible to select a specific OS for each hierarchy folder. Lower-level hierarchy folders inherit the setting of higher-level folders.

To assign a hierarchy folder to an OS project, select the "Edit Folder Properties..." command in the shortcut menu of the relevant folder. In the dialog window that then opens, you can select an OS project for the chosen PH folder and its subfolder.

Figure 5–13

🚔 Folder Prop	erties		<u>- 0 ×</u>
Name PUS7			
Assigned Compo	onents		
Name	Туре	Machine Name	Assignable
☑OS(1)	PCS 7 OS (Server/Single Station)		Yes
□ <u>■</u> 0S(2)	PCS 7 OS (Server/Single Station)		Yes
-	type can be selected. wailable Components		
	<u> </u>	el	

5.2.5 Assigning AS object instances

There are various ways to assign the process objects of the automation system to the PH. The simplest version is to drag-and-drop the objects from the AS View to the corresponding hierarchy folder.

Figure 5–14							
🗭 Siemens DBA AS to 05 Compilatio	on Utility - D:\	Projects\D	BA_projects	\$\ 57_1500_Demo	.dba		_ 🗆 ×
<u>File Edit R</u> un <u>T</u> ools <u>H</u> elp							
🗋 🗁 🛃 🛃 🕨 🗐 🔛 😕	X 🔗 🗄						
Plant View	Objects: 11	Attributes: 4					
⊡- [N S7_1500_Demo							
⊟ <u>∭</u> Plant1	Name		Value	Category	Descripti		
E BMT2	Author			Standard		id/or department of	the folder cre
in finite for the second seco	Commer			Standard		al documentation.	
	Picture T		0.000	Standard	Use Fold	ler name if blank	
	🐮 TargetA:	ssignment	OS(1)	Target			
F2001							
- File L1001							
MOT1							
- 🔂 MOT2							
📴 P2001							
🕞 🕞 SFC1 💦							
- 🔂 VAL1							
WAL2	N						
Line 1							
AS View PC Station View Validation R	esults						
Components	AS Objects:	10					
🔚 S7_1500	Name	Туре	AS Addre	226	Status	Assignment	
	🚰 F1001	S7Meas	0001:CL:	%DB101.%DBB80		\Plant1\WATER	
	🍟 F2001	S7Meas	0001:CL:	%DB101.%DBB160		\Plant1\WATER	
	😽 L1001	S7Meas	0001:CL:	%DB101.%DBB0		\Plant1\WATER	
	醤 мот1	S7Motor	0001:CL:	%DB100.%DBB0		\Plant1\WATER	
	醤 мот2	S7Motor	0001:CL:	%DB100.%DBB2		\Plant1\WATER	
	🚰 P1001	S7Meas	0001:CL:	%DB101.%DBB40		\Plant1\WATER	
	🚰 P2001	S7Meas	0001:CL:	%DB101.%DBB120		\Plant1\WATER	
Include All Resources	SFC1	S7SFC	0001:CL:	%DB400.%DBB0		\Plant1\WATER	
Include Assigned	🚰 VAL1	S7Valve	0001:CL:	%DB100.%DBB12		\Plant1\WATER	
Include Only Changes	VAL2	S7Valve		%DB100.%DBB14		\Plant1\WATER	

Changes to the plant hierarchy are indicated in the Plant View by a green background. This means that the OS project still needs to be compiled at a later point in time.



•							
🇭 Siemens DBA AS to OS Compilation Utility - D:\Projects\DBA_projects\S7_1500_Demo.dba							
File Edit Run Tools Help							
📄 🗁 🛃 🛃 🕨 💷 🔤	X 😤 🗄						
Plant View	Objects: 0 Attributes: 1 Graphics	s: 16 Alarm Messages					
⊡- ∭ S7_1500_Demo			<u>.</u>				
📄 🖻 👰 Plant1	Name	Value	Catego	Description 🔼			
😟 💼 🖻 BMT2	🎦 MAN - Comments	Manual Mode	Graphics	MAN - Comments			
📄 💼 🛅 BMT1	MAN_OFF - Comments	Manual Off	Graphics	MAN_OFF - Comment			
🖻 🖻 WATER	MAN_ON - Comments	Manual on	Graphics	MAN_ON - Comments			
- 🕆 WATER	🚰 QAUTO - Comments	State Automatic	Graphics	QAUTO - Comments			
🖬 F1001	🚰 QERR - Comments	Error	Graphics	QERR - Comments			
🖬 F2001	🚰 QMAN - Comments	State Manual	Graphics	QMAN - Comments			
	🚰 QMOTRUN - Comments	Motor Running	Graphics	QMOTRUN - Commei			
MOT1	🚰 QOPCOND - Comments	Operating Condition	Graphics	QOPCOND - Commer			
MOT2	🚰 SF	@S71500_Typicals.pdl	Graphics	Symbol File			
P1001	🚰 SN	@S7Motor/1	Graphics	Symbol Name			
	🚰 QSTART - Comments	Motor starting	Grenhico	OSTART - Comments			
	BESET Com						

5.2.6 Editing the properties of the objects

After assigning the AS objects to the PH, you can still make changes to a few OS-relevant settings. For example, a different symbol version can be selected, or the alarm class can be changed for messages. If, as mentioned in the chapter "4.3.4 Addressing tab", the addressing parameters of the process variables were assigned individually, an additional tab labeled "Addresses" will still be present. In this tab, it is then possible to set the AS address individually for each variable (Figure 5–16).

Figure 5–16

Ciamana DRA AC to OC Compilati			main ata\C7_1E	00 Dam	a dha#		
🗭 Siemens DBA AS to OS Compilati	on Utility - D:\P	rojects\DBA_p	rojects \S7_15	uu_ben	lo.dba*		
File Edit Run Tools Help							
🗋 🗁 🛃 💽 🕨 🔲 🛄 🖼	X 😤 🗄						
Plant View	Objects: 0 Att	ributes: 1 Graph	nics: 3 Addresse	s			
⊡ 🚺 S7_1500_Demo 🔺	TenNeuro	DBNum	DBOffset				
⊡- <u>ka</u> Plant1	Tag Name	DEINUM					
😟 💼 BMT2	Tag1	%DB101	%DD0				
	Taq2	%DB101	%DD4				
📄 👰 WATER 🔤							
- 🌴 WATER							
🖷 F1001							
- 📭 🚺 🔽							
AS View PC Station View Validation Results							
Components	Components AS Objects: 1						
S7_1500	Name Ty	pe 🛛 AS Addre	ISS	Status	Assignment		
S7_1500_unstruc	蹐 L1001 S7	Meas 0001:CL:	%DB101.%DBB0	Add	\Plant1\WATER		

5.2.7 Compile OS

With the Compile OS function, all objects configured in the DBA (tags, symbols, messages) are created in the OS project. Click the button with the triangle pointing to the right:

Figure 5	5–17
----------	------

•							
🇬 Siemens DBA AS to OS Compilation Utility - D:\Projects\DBA_projects\57_1500_Demo.dba							
File Edit Run Tools Help							
🗋 🗁 🛃 🛃 🕨 🖬 🖕	IX 😤 🗄						
Plant View	_ Objects: 0 Attributes: 1 Graphic:	s: 16 Alarm Messages					
⊡- 1 S7_1500_Demo		[
📄 🖳 👰 Plant1	Name	Value	Catego	Description _			
😟 💼 🖻 BMT2	🎽 MAN - Comments	Manual Mode	Graphics	MAN - Comments			
📄 🕀 💼 BMT1	MAN_OFF - Comments	Manual Off	Graphics	MAN_OFF - Comment			
📄 🖻 🙆 WATER	MAN_ON - Comments	Manual on	Graphics	MAN_ON - Comments			
₩ATER	🚰 QAUTO - Comments	State Automatic	Graphics	QAUTO - Comments			
F1001	🚰 QERR - Comments	Error	Graphics	QERR - Comments			
F2001	🚰 QMAN - Comments	State Manual	Graphics	QMAN - Comments			
L1001	🚰 QMOTRUN - Comments	Motor Running	Graphics	QMOTRUN - Commei			
MOT1	🚰 QOPCOND - Comments	Operating Condition	Graphics	QOPCOND - Commer			
MOT2	🚰 QSTART - Comments	Motor starting	Graphics	QSTART - Comments			
P1001	🚰 RESET - Comments	Error Reset	Graphics	RESET - Comments			
	🚰 SF	@S71500_Typicals.pdl	Graphics	Symbol File			
		@S7Motor/1	Graphics	Symbol Name			

In the dialog window that then opens, you can still configure the usual SIMATIC Manager settings for an entire compilation or the compilation of changes.

Figure 5–18	
🗘 Compile	×
Compile Targets Type Type Cos(1) PCS 7 OS (Server/Single :	OS(1) Compile Options Changes Changes Tags and Messages Create/Update Block Icons
✓ ► <u>C</u> ompile	Create Archive Tags Picture Tree Cancel

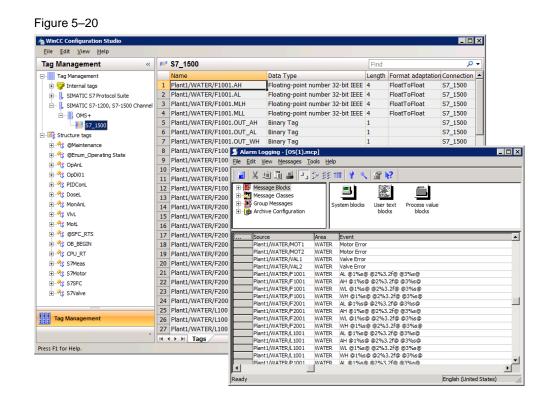
You can track the progress of the OS compilation in another dialog window. If the compiling process cannot be executed without errors, you can display the log by activating the "Show Details" option.



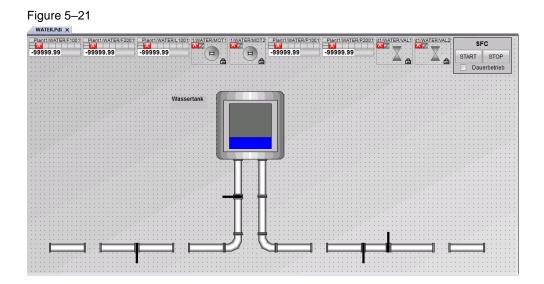
Command Status			
- Command			
CompilingCompleted			
	100%		
Step			
Jiep			
	100%		
	100%		
Timestamp	Message	Level	
07.05.2014 11:39:57	Compiling Picture Object: '@Plant1/WATER/P1001'	3	
07.05.2014 11:39:57	Changing Existing Symbol '@Plant1/WATER/P1001'.	4	
07.05.2014 11:39:59	Compiling Picture Object: '@Plant1/WATER/P2001'	3	
07.05.2014 11:39:59	Changing Existing Symbol '@Plant1/WATER/P2001'.	4	
07.05.2014 11:40:01	Compiling Picture Object: '@Plant1/WATER/VAL1'	3	
07.05.2014 11:40:01	Changing Existing Symbol '@Plant1/WATER/VAL1'.	4	
07.05.2014 11:40:02	Compiling Picture Object: '@Plant1/WATER/VAL2'	3	
07.05.2014 11:40:02	Changing Existing Symbol '@Plant1/WATER/VAL2'.	4	
07.05.2014 11:40:06	Compiling Picture Tree	2	
07.05.2014 11:40:06	Inserting Container: 'RMT2'	3	
07.05.2014 11:40:06	Inserting Container: 'RMT1'	3	
07.05.2014 11:40:06	Inserting Container: 'WATER'	3	
07.05.2014 11:40:06	Saving Picture Tree	3	
07.05.2014 11:40:21	Closing Target	2	
07.05.2014 11:40:22	Number of Compiled AS Objects12	1	
07.05.2014 11:40:22	Number of Compiled Tags336 (Internal = 255; External = 117; Single = 36)	1	
07.05.2014 11:40:22	Number of Compiled Messages44	1	
07.05.2014 11:40:22	Compiling 'OS(1)' PCS 7 OS (Server/Single Station)Completed	1	
07.05.2014 11:40:22	CompilingCompleted	1	•
Show Details			
Close When Done		<u>S</u> a	ve

5.3 PCS 7 operator station

When you now open the OS project on the Engineering Station, all connections, tags, messages, process pictures and block icons should be created.



- With the Graphics Designer, open the process pictures that were created using the DBA. The block icons for the AS objects will have been inserted next to one another.
- 6. Draw the process picture according to your requirements and move the block icons into the desired position.



7. In the SIMATIC Manager, execute the "Load OS" function and start the runtime at the OS server.

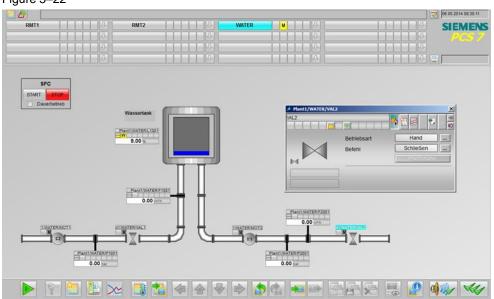


Figure 5–22

6 Appendix A

6.1 Skript "CalcS71500Addr"

```
Function CalcS71500Addr(ByVal oTag, ByVal sInstAddr, ByVal sTagAddr) 'V2.0
   Dim sInstAddrTokens, sTagAddrTokens, sTag, sType, sTagAddrOffsetPart
   Dim sDBNum, nDBStartOffset, fDBOffset, sQualifier, nNumDecPlaces
  Dim sAddr
   sTag = GetAttr(oTag,"Na","")
   sType = GetAttr(oTag, "Ty", "")
   sInstAddrTokens = SPlit(sInstAddr, ",")
   sTagAddrTokens = SPlit(sTagAddr, ",")
   sDBNum = sInstAddrTokens(0)
  nDBStartOffset = CLng(sInstAddrTokens(1))
   sTagAddrOffsetPart = SPlit(sTagAddrTokens(0),".")
   fDBOffset = CDbl(sTagAddrOffsetPart(0))
   If UBound(sTagAddrTokens) > 0 Then
          sQualifier = sTagAddrTokens(1)
          If Len(sQualifier) = 0 Then
                  sQualifier = VarTypeToS7Type(sType)
          End If
  Else
          sQualifier = VarTypeToS7Type(sType)
  End If
   If UBound (sTagAddrTokens) > 1 Then
          nNumDecPlaces = sTagAddrTokens(2)
          If Len(nNumDecPlaces) = 0 Then
                  nNumDecPlaces = 0
          End If
  ElseIf sType = "BIT" Then
          nNumDecPlaces = 1
  Else
          nNumDecPlaces = 0
  End If
   sAddr = "0001:CL:%DB"
  sAddr = sAddr & sDBNum
   sAddr = sAddr & ".%" & sQualifier
   sAddr = sAddr & FormatNumber(nDBStartOffset + fDBOffset,0, , ,False)
If nNumDecPlaces > 0 Then
   sAddr = sAddr & "." & sTagAddrOffsetPart(1)
End If
  CalcS71500Addr = sAddr
End Function
```

6.2 Skript "CalcS71500ASOAddr"

```
Function CalcS71500ASOAddr(ByVal sInstAddr)
Dim sInstAddrToken, sAddr
sInstAddrToken = SPlit(sInstAddr, ",")
sAddr = "0001:CL:%DB"
sAddr = sAddr & sInstAddrToken(0)
sAddr = sAddr & ".%DBB"
sAddr = sAddr & sInstAddrToken(1)
CalcS71500ASOAddr = sAddr
End Function
```

7 Appendix B

7.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.

Please send queries to Technical Support via Web form:

support.industry.siemens.com/cs/my/src

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android: support.industry.siemens.com/cs/ww/en/sc/2067

7.2 Industry Mall



The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location: mall.industry.siemens.com

7.3 Literature

	Subject area	Title
\1\	Siemens Industry Online Support	https://support.industry.siemens.com
\2\	Article download page	https://support.industry.siemens.com/cs/ww/en/view/49740087
\3\	SIMATIC PCS 7 in Industry Online Support	https://support.industry.siemens.com/cs/ww/en/view/63481413
\4\	Product information PCS 7/OPEN OS	https://support.industry.siemens.com/cs/ww/en/view/109750265
\5\	Doenload PCS 7/OPEN OS V9.0 Update 1	https://support.industry.siemens.com/cs/ww/en/view/109781622
\6\	SIMATIC PCS 7 APL Style guide	https://support.industry.siemens.com/cs/ww/en/view/65601446

Table 7–1

7.4 History

Toh	6	7	2
Tab	e	1-	-2

Version	Date	Change
V1.0	09/2014	First edition
V1.1	07/2018	Correction in script "CalcS71500Addr"
V1.2	06/2021	Update to PCS 7/OPEN OS V9.0 and further correction in Script "CalcS71500Addr"
V1.3	09/2021	Fixing an error in chapter 6