# SIMOTION

## Guidelines for implementing Automation Projects in the Pharmaceutical Industry

### Engineering Manual

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Table of contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites for configuring automated systems in the GMP environment</td>
<td>1</td>
</tr>
<tr>
<td>Requirements of automated systems in the GMP environment</td>
<td>2</td>
</tr>
<tr>
<td>System specification</td>
<td>3</td>
</tr>
<tr>
<td>Guidelines for implementing SIMOTION in a GMP environment</td>
<td>4</td>
</tr>
<tr>
<td>Supplementary software components</td>
<td>5</td>
</tr>
<tr>
<td>Supporting functions during qualification</td>
<td>6</td>
</tr>
<tr>
<td>Index</td>
<td></td>
</tr>
</tbody>
</table>

09/2007
ASE01257283-01
Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol. The notices shown below are graded according to the degree of danger.

**Danger**
indicates that death or severe personal injury will result if proper precautions are not taken.

**Warning**
indicates that death or severe personal injury may result if proper precautions are not taken.

**Caution**
with a safety alert symbol indicates that minor personal injury can result if proper precautions are not taken.

**Caution**
without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

**Notice**
indicates that an unintended result or situation can occur if the corresponding notice is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

**Qualified Personnel**

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by qualified personnel. Within the context of the safety notices in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

**Prescribed Usage**

Note the following:

**Warning**
This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

**Trademarks**

All names identified by ® are registered trademarks of the Siemens AG.
The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

**Disclaimer of Liability**

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.
Introduction

Purpose of this manual

This manual describes what is required in Good Manufacturing Practice (GMP environment) of an automated system, the software, and the procedure for configuring SIMOTION in terms of pharmaceutical regulations. Practical examples are used to explain the relationship between requirements and implementation.

Intended audience

This manual describes approaches for the implementation of automation solutions where GMP is mandatory. It is intended for all those who use the SIMOTION motion control system in the GMP environment, especially:

• Planners, project managers, and project engineers
• Commissioning and maintenance personnel
• Application personnel in machine construction
• Plant operators

Basic knowledge required

Basic knowledge of SIMOTION is required to understand this manual. Knowledge of GMP as practiced in the pharmaceutical industry is also an advantage.

Disclaimer of liability

This manual contains instructions for system users and project engineers for integrating SIMOTION into the GMP environment with regard to validation, taking into account special aspects such as 21 CFR Part 11.

We have checked that the contents of this document correspond to the hardware and software described. Nevertheless, we cannot assume responsibility for any deviations that may arise. The information in this document is checked regularly for system changes or changes to the regulations of the various organizations and necessary corrections will be included in subsequent editions. We welcome any suggestions for improvement and ask that they be sent to the Competence Center Pharma in Karlsruhe (Germany).
Introduction

Scope of the manual
The information described in this manual is valid for SIMOTION hardware and SCOUT engineering software. Up-to-date information about SIMOTION can be found online at www.siemens.com/SIMOTION.

Place in the information environment
This manual describes the requirements, prerequisites, and application of SIMOTION in Good Manufacturing Practice in terms of pharmaceutical regulations, and supplements the system documentation relating to the SIMOTION motion control system.

Structure of the guidelines
The guidelines are intended to provide an overview of the prerequisites for configuring the requirements of automation systems in the GMP environment and to help in their implementation.

All the necessary functions and requirements for hardware and software components are described, which should make the selection of components easier.

Examples are used to briefly explain how the hardware and software is used and configured or programmed in order to meet the relevant requirements. More detailed explanations can be found in the standard documentation.

In the appendix you will find an index.

Conventions
The following conventions are used in this manual:

Operating instructions involving several steps are shown in the form of a table and numbered in the order in which they should be performed.

Operating instructions involving only a few steps are indicated by a bullet point (•).

References to other manuals are shown in bold italic type.

Menu commands are shown in bold type.
Further assistance

Please talk to your local Siemens contact if you have further questions about how to use the products described in this manual.

Find your contact at:

http://www.siemens.com/automation/partner

You will find a guide to the technical documentation we offer for individual SIMOTION products and systems at:

http://support.automation.siemens.com

The online catalog and ordering system are available at:

http://mall.automation.siemens.com/

If you have questions about the manual, please contact the Competence Center Pharma:

E-mail: pharma.aud@siemens.com
Fax: +49 721 595 6930

Further information about the products, systems, and services available from Siemens for the pharmaceutical industry can be found at:

http://www.siemens.com/pharma

Training centers

Siemens offers a number of training courses to familiarize you with the SIMOTION motion control system. Please contact your regional training center, or the central training center in Nuremberg, Germany.

Telephone: +49 (911) 895-3200.
Internet: http://www.sitrain.com

Technical support

You can access technical support for all A&D products

• By submitting the Support Request web form
  http://www.siemens.com/automation/support-request
• Telephone: + 49 180 5050 222
• Fax: + 49 180 5050 223

Further information about Siemens technical support is available on the Internet at

http://www.siemens.com/automation/service
Service & Support on the Internet

In addition to our pool of documentation, you can also access our expertise online. 
http://www.siemens.com/automation/service&support

There, you will find the following information:

• The Newsletter containing the latest information on your products.
• The documents you need via our Search function in Service & Support.
• A forum in which users and specialists around the world swap experiences.
• Your local Automation & Drives representative.
• Information on local service, repairs, spare parts. And lots more on the "Services" pages.
# Table of contents

**Introduction** iii  
**Table of contents** vii  
1 **Prerequisites for configuring automated systems in the GMP environment** 1-1  
   1.1 Life-cycle model for package units 1-2  
   1.2 Laws and guidelines 1-8  
   1.3 Responsibilities 1-10  
   1.4 Approval and change procedure 1-10  
   1.5 Software categorization of automated systems 1-11  
2 **Requirements of automated systems in the GMP environment** 2-1  
   2.1 Hardware categorization 2-2  
   2.2 Software categorization 2-3  
   2.3 Configuration management 2-5  
   2.3.1 Configuration identification 2-6  
   2.3.2 Configuration control 2-6  
   2.3.3 Version control 2-6  
   2.3.4 Change control and as-built documentation 2-6  
   2.4 Software creation 2-7  
   2.4.1 Using typicals for programming 2-7  
   2.4.2 Identifying software modules/typicals 2-7  
   2.4.3 Changing software modules/typicals 2-7  
   2.5 Access protection and user administration 2-8  
   2.5.1 Applying access protection to a system 2-8  
   2.5.2 Requirements of the user ID and password 2-9  
   2.5.3 Chip cards 2-9  
   2.6 Electronic signatures 2-10  
   2.6.1 Security measures for user IDs/passwords 2-10  
   2.7 Audit trail 2-11  
   2.8 Time synchronization 2-12  
   2.9 Archiving data 2-13  
   2.10 Logging batch data 2-14  
   2.11 Data backup 2-15  
   2.11.1 Application software 2-16  
   2.11.2 Process data 2-17  
   2.12 Using third-party components 2-18  
3 **System specification** 3-1  
   3.1 Selection criteria for SIMOTION hardware 3-2  
   3.2 Required and optional software packages 3-3  
   3.2.1 SIMOTION basic engineering software 3-3  
   3.2.2 HMI and communication basic engineering software 3-3  
   3.2.3 Additional software packages for user administration, access protection, versioning, audit trail 3-4
# Guidelines for implementing SIMOTION in a GMP environment

4.1 Software categorization of SIMOTION SCOUT ......................................................... 4-2
4.2 Software installation ................................................................................................. 4-3
4.3 Principles of configuration ..................................................................................... 4-4
4.3.1 Software creation using SIMOTION SCOUT ..................................................... 4-4
4.3.2 Programming languages ..................................................................................... 4-8
4.4 Configuration management .................................................................................... 4-13
4.4.1 Changing the system and application software ............................................... 4-13
4.4.2 Versioning of the application software ............................................................ 4-16
4.4.3 Versioning individual blocks ............................................................................. 4-17
4.4.4 Change control ................................................................................................. 4-22
4.5 Access protection ................................................................................................. 4-25
4.5.1 Access protection for a SIMOTION project ....................................................... 4-25
4.5.2 Protective measures in the application software ............................................... 4-28
4.6 Audit trail ............................................................................................................. 4-29
4.7 Time synchronization ............................................................................................ 4-30
4.7.1 Time synchronization: WinCC - SIMOTION .................................................... 4-32
4.7.2 Time synchronization: SIMOTION - WinCC flexible ....................................... 4-33
4.8 Interrupt and alarm handling ................................................................................. 4-36
4.8.1 Time stamping of GMP-relevant data .............................................................. 4-36
4.8.2 Alarm procedure ............................................................................................... 4-36
4.8.3 Alarm configuration in SIMOTION and WinCC flexible .................................... 4-38
4.9 Saving and restoring SIMOTION application software ........................................... 4-48
4.9.1 Saving and restoring projects ........................................................................... 4-48
4.9.2 Saving and restoring tags ................................................................................. 4-49

# Supplementary software components

5.1 SIMOTION SCOUT diagnostic tools ....................................................................... 5-1
5.1.1 SIMOTION SCOUT diagnostics overview ......................................................... 5-2
5.1.2 SIMOTION trace tool ........................................................................................ 5-6

# Supporting functions during qualification

6.1 Introduction ............................................................................................................ 6-1
6.2 Qualification of automation hardware ................................................................... 6-2
6.3 Qualification of automation software ..................................................................... 6-4
6.3.1 Qualification of standard software .................................................................... 6-4
6.3.2 System software common to SIMOTION and SIMATIC ................................... 6-5
6.3.3 Qualification of the application software .......................................................... 6-7

Index
1 Prerequisites for configuring automated systems in the GMP environment

The availability of approved specifications, such as requirement and function specifications, is a prerequisite for the configuration of automated systems in the GMP environment. Requirements contained in standards, recommendations, and guidelines must be observed when creating these specifications. This chapter deals with the most important of these regulations and various specifications (URS, FS, DS).
1.1 **Life-cycle model for package units**

A central component of Good Engineering Practice (GEP) is the application of a recognized project methodology, including a defined life cycle. The aim is to deliver a solution that meets the relevant requirements and is also cost-effective.

The figure below shows the life-cycle model used in this manual. It is based on the recommendations of the latest GAMP ® Guide for Validation of Automated Systems. Its stages have been simplified, due to the fact that package units are not as complex as whole plant systems. The life-cycle model starts with the machine planning stage and ends with the initiation of pharmaceutical production following qualification and validation.

![Figure 1-1: Life-cycle model](image-url)
**Legend for the life-cycle model**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP</td>
<td>Validation plan(^1)</td>
</tr>
<tr>
<td>QP</td>
<td>Qualification plan</td>
</tr>
<tr>
<td>QPP</td>
<td>Quality and project plan</td>
</tr>
<tr>
<td>URS</td>
<td>User requirement specification(^2)</td>
</tr>
<tr>
<td>FDS</td>
<td>Function and design specification</td>
</tr>
<tr>
<td>DS</td>
<td>Design specification</td>
</tr>
<tr>
<td>FAT</td>
<td>Factory acceptance test</td>
</tr>
<tr>
<td>SAT</td>
<td>Site acceptance test</td>
</tr>
<tr>
<td>IQ</td>
<td>Installation qualification</td>
</tr>
<tr>
<td>OQ</td>
<td>Operational qualification</td>
</tr>
<tr>
<td>PQ</td>
<td>Performance qualification</td>
</tr>
<tr>
<td>VR</td>
<td>Validation report</td>
</tr>
<tr>
<td>QR</td>
<td>Qualification report</td>
</tr>
<tr>
<td>SOP</td>
<td>Standard operation procedure</td>
</tr>
</tbody>
</table>

\(^1\) In the interests of making certain terms easier to read and recognize, not all terms have been translated into German in the German-language version of this document, nor have German abbreviations been provided for all terms.

\(^2\) The terms "user requirement specification/Anwender-Lastenheft" and "function specification/Pflichtenheft" used in GAMP ® 4 do not have the same meanings as the German terms "Lastenheft" and "Pflichtenheft" as used in VDI 3694 and VDI 2519, for example.
Validation plan
In the validation plan (VP), the customer defines the overriding strategy and the parties responsible for validating a system in its operational environment [PDA, GAMP ® 4].

- In the case of a complex plant (a production line with multiple process plants and automation systems, for example), a further distinction can be made between an overriding master document (validation master plan, also referred to as VMP or MVP) and a VP which is valid for individual plants and systems.
- See also GAMP ® 4, Appendix M1 "Guideline for Validation Planning".

Quality and project plan
The quality and project plan (QPP) defines the scope of and procedures relating to project and quality management, with document and change control procedures, for example, being specified. The life cycle is defined in such a way in the QPP that it not only includes project stages which are relevant for validation, but also other organizational relationships (different time schedules from the various sections, for example).

With package units, it is recommended that all required qualification steps are defined in the QPP.

- See also GAMP ® 4, Appendix M6 "Guideline for Quality and Project Planning".
Specification

The specification stage starts with the creation of the user requirement specification. As a rule, this is created by the operator and describes the requirements which the system has to meet. Once the user requirement specification has been created, a function specification is then drawn up, usually by the supplier. The function specification (FS) renders the requirements defined in the URS more precisely on a functional level. The subsequent design specification (DS) contains detailed requirements as regards implementation.

The function and design specifications both form the basis for later qualification and validation tests. The following issues also have to be addressed during the function and design specification stages:

- Software structure
- Programming standards
- Naming convention
- File naming convention

User requirement specification (URS)

The user requirement specification describes the requirements the system has to meet from the operator's point of view. As a rule, it is created by the system operator, who may be supported in this task by system suppliers. It forms the basis for all subsequent specifications.

- See also GAMP ® 4, Appendix D1 "Production of a User Requirement Specification".
Function and design specification (FDS)

As a rule, the FDS is created by the system supplier, occasionally in collaboration with the end user. Based on the user requirement specification, it describes the system functions in detail and also includes in-depth descriptions of the hardware and software to be used, measuring-point lists, etc. The approved function specification provides the foundation on which detailed specifications are drawn up.

- See also GAMP ® 4, Appendix D2 "Production of a Function Specification", GAMP ® 4, Appendix D3 "Production of a Hardware Design Specification", and GAMP ® 4, Appendix D4 "Production of Software Design Specifications and Software Module Design Specifications".

Implementation

The system is implemented in accordance with the design specification during the implementation stage. Along with the procedures defined in the QPP and additional guidelines (coding standards, naming conventions, and data backups, for example), change management, which aims to enable changes to and deviations from the original specifications to be traced, plays an important role.

See also GAMP ® 4, Appendix M8 "Guideline for Project Change Control" and GAMP ® 4, Appendix M10 "Guideline for Document Management".

FAT

Once the implementation steps have been completed, a factory acceptance test (FAT) is often carried out on the supplier's premises and documented, enabling any programming errors to be identified and remedied prior to delivery.

The aim of the FAT is for the customer to accept the system for delivery in its tested state.
Test stage/qualification

The FAT is followed by technical commissioning (commissioning stage). This involves installing the system on the system operator's premises along with the created user program, followed by technical commissioning, testing, and qualification.

The commissioning and qualification stages can follow on from one another or can be combined. To save time and money, it is recommended that commissioning and qualification activities are coordinated.

The test plan must be drawn up at an early stage so that a check can be made to see whether tests already carried out as part of the FAT can be omitted during qualification. In this case, the documented FAT tests become part of the qualification documentation.

When test documents are created, tests and acceptance criteria must be clearly described.

Qualification report

The qualification report (QR) summarizes the results of the tests performed, based on the qualification plan, and confirms that the qualification stages have been completed successfully.
1.2 Laws and guidelines

The recommendations and guidelines of various organizations have to be taken into account when configuring automation systems requiring validation in the GMP environment. These are usually based on general guidelines, such as Code of Federal Regulations Title 21 (21 CFR) of the US Food and Drug Administration (FDA) or the EU GMP Guide Annex 11.

<table>
<thead>
<tr>
<th>Law/Guideline</th>
<th>Author/Organization</th>
<th>Title</th>
<th>Law/Recommendation</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 CFR Part 11</td>
<td>US FDA</td>
<td>Electronic records; electronic signatures</td>
<td></td>
<td>Law</td>
</tr>
<tr>
<td>21 CFR Part 210</td>
<td>US FDA</td>
<td>Current good manufacturing practice in manufacturing, processing, packing, or holding of drugs; general</td>
<td></td>
<td>Manufacturers and importers of pharmaceutical products for the US market</td>
</tr>
<tr>
<td>21 CFR Part 211</td>
<td>US FDA</td>
<td>Current good manufacturing practice for finished pharmaceuticals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annex 11 of the EU GMP Guide</td>
<td>European Commission Directorate General III</td>
<td>Computerized Systems</td>
<td>Guideline</td>
<td>Europe</td>
</tr>
<tr>
<td>Annex 18 of the EU GMP Guide</td>
<td>European Commission Directorate General III</td>
<td>Good Manufacturing Practice for Active Pharmaceutical Ingredients</td>
<td>Guideline</td>
<td>Europe</td>
</tr>
<tr>
<td>GAMP ® 4</td>
<td>ISPE</td>
<td>GAMP ® 4 Guide for Validation of Automated Systems</td>
<td>Guideline</td>
<td>Worldwide</td>
</tr>
<tr>
<td>NAMUR Recommendation NE 71</td>
<td>NAMUR</td>
<td>Operation and Maintenance of Validated Systems</td>
<td>Recommendation</td>
<td>Europe</td>
</tr>
</tbody>
</table>

Note
This manual is based on the requirements of GAMP ® 4 and US 21 CFR Part 11.
Code of Federal Regulations Title 21 (21 CFR), Food and Drugs

Code of Federal Regulations Title 21 is comprised of different Parts, such as Parts 11, 210, and 211. Part 11 is of particular significance for computerized systems (21 CFR Part 11). This Part deals with electronic records and electronic signatures.

Annex 11 of the EU GMP Guide

Annex 11 of the EU GMP Guide contains 19 points which describe the configuration requirements, operation, and change control of computer systems in the GMP environment. An interpretation of Annex 11 can be found in the GAMP ® 4 Guide for Validation of Automated Systems in the form of an APV (International Association for Pharmaceutical Technology) guideline.

Annex 18 of the EU GMP Guide

Annex 18 of the EU GMP Guide deals with good manufacturing practice (GMP) for active pharmaceutical ingredients. It is designed to be used as a GMP guide when manufacturing active pharmaceutical ingredients in the context of a suitable quality management system. Section 5 of Annex 18 deals with process equipment and its use.

GAMP ® Guide for Validation of Automated Systems "GAMP ® 4"

The GAMP ® (Good Automated Manufacturing Practice) Guide for Validation of Automated Systems was compiled to be used as a recommendation for suppliers and a guide for the operators of automated systems in the pharmaceutical manufacturing industry. The current version, "GAMP ® 4", was published in December 2001.

NAMUR recommendations

NAMUR recommendations are field reports compiled by the "User Association of Process Control Technology in Chemical and Pharmaceutical Industries" for their members to use on an optional basis. They should not be viewed as standards or guidelines. The NAMUR recommendations below are of particular interest for the configuration and use of automated systems in the GMP environment:

NE 71 "Operation and Maintenance of Validated Systems"
1.3 Responsibilities

Responsibilities for the activities included in the individual live-cycle stages must be defined when configuring automated systems in the GMP environment and creating corresponding specifications. As this definition is usually laid down on a customer- and project-specific basis and requires a contractual agreement, it is recommended that the definition is integrated into the quality and project plan. See also GAMP ® 4, Appendix M2.

1.4 Approval and change procedure

When new systems requiring validation are set up or when existing systems are changed, the top priority is to achieve or retain validated status.

Setting up new systems

If a new system is set up, document approval and the transitions between life-cycle stages are defined prior to commencement of the project. This is usually carried out in conjunction with the definition of responsibilities contained in the quality and project plan. A life cycle like the one described in Chapter 1.1 "Life-cycle model" is used.

Changing validated systems

Changes to an existing, validated system are regulated as per the company's change control procedures. Before any changes are carried out they must be described, potential consequences must be identified, and associated steps (performing tests, updating as-built documentation, for example) must be defined. Once final approval has been received, the planned change is carried out, as are the defined steps.

If comprehensive changes are needed, a life cycle similar to the one shown in this manual may be used if required.
1.5 **Software categorization of automated systems**

As described in Chapter 2.2 "Software categorization", a system's software can be classified into five software categories in accordance with the GAMP ® 4 Guide for Validation of Automated Systems. The software categories have a considerable effect on the amount of work required during the test and qualification stage and must be determined during the specification stage for the software used.
Prerequisites for configuring automated systems in the GMP environment
2 Requirements of automated systems in the GMP environment

This chapter describes the essential requirements which an automated system must meet in the GMP environment. These requirements must be defined in the specification and implemented during configuration. In general, it must always be specified that proof of who has changed or performed what and when they have done it must always be available (the "why" is optional). The requirements of this task are implemented in various functions and described in the following chapters.

The following graphic shows the life-cycle model. The requirements focused on in this chapter belong to the Specification area in the graphic.
2.1 Hardware categorization

A system’s hardware components are assigned to one of two hardware categories in accordance with the GAMP ® 4 Guide, Appendix M4. The hardware categories are listed below:

Category 1, Standard hardware

Category 1, Standard hardware includes established, commercially-available hardware components. This type of hardware is also subject to the relevant quality and testing mechanisms.

The hardware is accepted and documented by means of an IQ test.

Category 2, Customized hardware

The functionality of such hardware must be specified, then checked and documented in detail by means of appropriate, documented tests.
2.2 Software categorization

According to the GAMP ® Guide for Validation of Automated Systems, the software components of a system are assigned to one of five software categories.

Category 1, Operating systems

Category 1, Operating systems includes established, commercially-available operating systems. The operating system itself is not subject to individual validation, although its name and version must be documented and verified in the context of an installation qualification (IQ).

Category 2, Firmware

Category 2 covers the firmware found in field instruments, compact controls, and motion controls, for example, whose configuration has been customized in accordance with local conditions, for instance. Here too, the name and version of the firmware must be documented, along with its configuration, and checked in the context of an installation qualification (IQ). The device functionality must be checked by means of an operational qualification (OQ).

Category 3, Standard software packages

Category 3 includes commercially-available standard software packages, as well as standard solutions for particular processes. The configuration of these software packages must be restricted to customizing them in accordance with the runtime environment (e.g. network and printer connections) and configuring the process parameters. The name and version of the standard software packages must be documented and checked as part of an installation qualification (IQ). User requirements (such as access protection, interrupts, alarms, or calculations) must be documented and checked in the context of an operational qualification (OQ).

Category 4, Configurable software packages

Category 4 includes configurable software packages, which facilitate specific business and manufacturing processes. Standard software modules need to be configured for such packages. These software packages should only be considered to be Category 4 if they are well known and fully developed, otherwise Category 5 would be more suitable. If critical and/or complex applications are involved, a supplier audit is usually carried out.

The name, version, and configuration must be documented and checked as part of an installation qualification (IQ). The functions of the software packages must be checked against the user requirements as part of an operational qualification (OQ). The validation plan must take the life-cycle model and an assessment of suppliers and software packages into account.
Category 5, Customized software

Category 5 includes customized software, which has been developed in order to meet the specific needs of the user's company.

A supplier audit is usually required in order to verify that suitable quality systems were used to check the development and subsequent maintenance of the software. Otherwise, suppliers must use the GAMP® 4 Guide as a basis for their own development life cycles.

Once again, the name, version, and configuration must be documented and checked as part of an installation qualification (IQ). A detailed software specification must be drawn up and the software's function checked as part of an operational qualification (OQ). The validation plan must define a complete life-cycle approach for validation.

The amount of test work involved is considerably greater if software of higher categories is used, as opposed to software of lower categories.

The amount of validation and testing work can be reduced by making use of standardized software wherever possible.

Figure 2-2: Work involved in validation according to software category
2.3 Configuration management

The GAMP ® 4 Guide defines configuration management as the process which needs to be followed in order to precisely define an automated system at any point during its life cycle, from initial development right through to decommissioning of the system.

Configuration management involves using administrative and technical procedures in order to:

• Identify, define, and essentially specify system components (often in the FDS)
• Control changes to and approvals of elements
• Record and document element statuses and modifications
• Ensure elements are complete, consistent, and correct
• Control the storage, treatment, and delivery of elements

Configuration management comprises the following activities:

• Configuration identification (WHAT is to be kept under control)
• Configuration control (how the control is performed)
• Configuration status report (how the control is documented)
• Configuration evaluation (how the control is verified)

This chapter deals with configuration identification and configuration control.
2.3.1 Configuration identification

Version and change management is only practical in an appropriate configuration environment. Therefore, every Siemens software and hardware package is identified by a unique product designation (order number) and a version number. For application software, the parts of an automated system which are to be subject to configuration management must be unambiguously defined. The system must be divided into configuration elements to this end. These must be defined at an early stage of implementation to ensure that a complete list of configuration elements can be created and maintained. Application-specific elements should have a unique ID (name or identification number). The amount of detail required when defining elements is determined by the requirements of the system and the supplier who is developing the application.

2.3.2 Configuration control

The maintenance of configuration elements must be checked at regular intervals by means of reviews, for example. Particular attention must be paid to change control and the associated version control. Furthermore, attention must also be paid to the archiving and approval of individual configuration elements.

2.3.3 Version control

The configuration elements must be versioned in order to ensure that change management functions correctly. The version must be updated each time a change is made.

2.3.4 Change control and as-built documentation

Suitable control mechanisms must be in place during configuration in order to ensure that changes are documented and transparency achieved. The control mechanisms can be described by means of SOPs and must cover the following:

- Software versioning
- Specifications such as programming guidelines, naming conventions, etc.
- Safeguarding of the traceability of changes to program codes
- Unique identification of software and all components contained within
- "As-built documentation" can be used in addition or as an alternative to change control for package units. Individual changes are purposely not recorded in detail, with the actual status of the machine being documented once a specific project stage has been completed (after the machine has been set up, prior to the FAT, for example). However, once the machine has been commissioned at the latest, a formal change control procedure is established, which enables individual changes to be traced. New as-built documentation can also then be generated at regular intervals or after a specific number of changes.
2.4 Software creation

Certain guidelines must be followed during software creation, which must be documented in the quality and project plan (GEP idea). Software creation guidelines can be taken from the GAMP ® 4 Guide for Validation of Automated Systems and from relevant standards and recommendations.

2.4.1 Using typicals for programming

As shown in Chapter 2.2 "Software categorization", the amount of validation work required increases enormously as you go up through the GAMP ® software categories. While the validation of category 1 software only calls for the software name and version to be checked, category 5 software validation requires the entire range of functions to be checked and a supplier audit to be performed.

To keep the required level of validation work as low as possible, priority must be given to standardized function blocks (products, in-house standards, project standards) during configuration. Standard function blocks are used to create and test customized typicals in accordance with design specifications.

2.4.2 Identifying software modules/typicals

During software creation the individual software modules must be assigned a unique name, a version, and a short description of the module. If changes are made to software modules, this must be reflected in the module ID.

2.4.3 Changing software modules/typicals

If changes are made to software modules, this must be noted in the corresponding module ID. As well as incrementing the version identifier, the date of the change and the name of the change initiator must also be included in the software module's ID. If required, the software modules to be changed must be indicated by means of a comment and a reference to the corresponding change request/order.
2.5 Access protection and user administration

To ensure that automated systems in the GMP environment are secure, such systems must be equipped with an access-control system. Access-control systems can not only deny or permit users access to certain rooms, but can also protect systems against unauthorized access. Users are put into groups which are in turn used to manage user rights. Individual users can be granted access authorization in various ways:

- Combination of unique user ID and password
- Chip cards together with a password

The system owner or an employee (administrator) nominated by the operator controls the assignment and management of user authorizations to ensure that access is suitably protected.

2.5.1 Applying access protection to a system

In general, actions which can be executed on an automated system must be protected. Depending on his or her particular field of activities, a user can be assigned various rights. Only the system owner or an employee nominated by the system owner may have access to user administration. Furthermore, unauthorized persons must be prevented from accessing electronic data records.

An automatic logout function must be installed on the system. The logout time must be agreed with the operator and specified in the function specification.

Note

Please note that only authorized persons must be able to access PCs and the system. This can be ensured by using appropriate measures such as mechanical locks and software for remote access.
2.5.2 **Requirements of the user ID and password**

**User ID:**

The user ID for a system must be of a minimum length defined by the customer and be unique within the system.

**Password:**

A password should usually be a combination of numeric and alphanumeric characters. When passwords are created, a minimum number of characters and a password expiry date must be defined. Generally, the password structure is defined on a customer-specific basis.

Password structure criteria:

- Minimum password length
- Use of numeric and alphanumeric characters
- Use of uppercase and lowercase letters

2.5.3 **Chip cards**

As well as the traditional method of identification requiring a user ID and password, users can also be identified by means of chip cards.
2.6 Electronic signatures

Electronic signatures are computer-generated character strings, which act as legally binding equivalents to handwritten signatures.

Regulations concerning the use of electronic signatures are defined in US FDA 21 CFR Part 11, for example.

Electronic signatures are of practical relevance when it comes to manual data input and operator intervention during runtime, approving process actions and data reports, and changing recipes, for example.

As electronic signatures come under the domain of HMI, MES, and other data management systems, the requirements of 21 CFR Part 11 in relation to electronic signatures do not apply to motion control systems such as SIMOTION.

The requirements of electronic signatures are described in detail in the GMP engineering manuals for WinCC flexible, WinCC, and PCS 7.

However, other requirements of 21 CFR Part 11, such as those relating to password security, audit trails, and data archiving and integrity, do still apply and are listed below.

Note
The regulations found in 21 CFR Part 11, published by the FDA, must be observed in the manufacture of all pharmaceutical products and medical devices intended for the US market.

2.6.1 Security measures for user IDs/passwords

The following points must be observed in order to safeguard the security of electronic signatures where user IDs and passwords are used:

- Unique user ID and password
- User IDs issued in a controlled manner
- Permissions retracted if user IDs/passwords are lost or found to be insecure or compromised
- Security precautions used to prevent the unauthorized use of user IDs/passwords or to report any misuse
- Personnel trained and provided with proof of such training
2.7 Audit trail

The audit trail is a system-side control mechanism, which ensures that data input and changes to data can be traced. A secure audit trail is particularly important as regards the creation, modification, or deletion of GMP-relevant electronic records.

In this case, the audit trail must archive and document all changes or actions performed, together with the corresponding date and time. The typical content of an audit trail must be specified and must cover "who" has changed "what" and "when" (old value/new value).

The archiving period must match that laid down in the specification.

Sufficient hard-disk capacity must be provided to store the entire audit trail until it is swapped out to an external data carrier.

Systems which provide adequate data integrity must be used (redundant systems, standby systems, RAID 5, for example).
2.8 Time synchronization

Within a system, a uniform time reference must be guaranteed to allow alarms, interrupts, etc. to be archived with unique time stamps. A time synchronization to a standard time is desirable, but not mandatory. Time synchronization is especially recommended for logging data and analyzing faults.
2.9 Archiving data

"Archiving" refers to the permanent retention of a computer system's electronic data and records in a long-term memory.\(^3\)

The customer is responsible for defining procedures and controls relating to the retention of electronic data.

Based on predicate rules, the customer must decide how electronic data will be retained and, in particular, which data will be affected by this procedure. This decision must be founded on a sound and documented risk assessment, which also takes the relevance of the electronic data over the time period it is to be archived into account.

The customer must define the following requirements\(^4\):

- Whether any archiving is even required for the application in question (backup/restore function may be kept separate from the archive function)
- Required archiving duration for the relevant data types, based on legal and commercial requirements
- An archiving procedure, which takes readback capability, data format, feasibility, and the durability of media and platform into account.
  - If electronic records are selected, electronic data can be saved in a standard format, such as PDF, XML, SGML, etc.
- A procedure for recording metadata used to correctly interpret saved data
- Demonstration of ongoing controls on the contiguous recording of electronic data and of the authenticity of electronic signatures

Process values (often in the form of trends), alarms (interrupts, warnings, etc.), audit trails, and, under certain circumstances, other batch data can be archived for SIMATIC systems.

The memory space on a system's data carriers is restricted. Data can be swapped out to external data carriers at regular intervals in order to free up space on these system data carriers.

If archived data is migrated or converted, the integrity of that data must be safeguarded throughout the entire conversion process.\(^5\)

---


2.10 Logging batch data

Batch documentation is particularly important where the manufacture of pharmaceutical products and medical devices is concerned. Correctly created batch documentation is often the only documentary evidence that pharmaceutical manufacturers can provide in the context of product liability.

Alarm data, for example, which is passed from SIMOTION to an HMI, is also electronically logged along with batch data. Usually, this batch data is actually handled and subjected to further processing in the higher-level HMI system. The requirements of how to handle batch data are described in detail in the GMP engineering manuals for WinCC flexible, WinCC, and PCS 7.
2.11 Data backup

In contrast to the archiving of electronic data, data backups create backup copies which ensure that the system can be restored if the original data or the entire system is lost.\(^6\)

The backup procedure must cover the periodic backup of vulnerable information in order to prevent all data being lost due to faulty system components or inadvertent deletion of the data. Backup procedures must be tested to ensure that data is saved correctly. Backup records must be clearly and easily identifiable and they must be assigned a date.\(^7\)

Data backups are created on external data carriers, which must comply with the recommendations made by the device manufacturer.

In terms of electronic-data backups, a distinction is made between software backups (e.g. application software, partition image) and archivedata backups.

Particular attention must be paid to the storage of databackup media (copy and original must be kept separately, backups must be protected from magnetic fields and the elements).


\(^7\) "Electronic Records and Electronic Signatures Compliance Assessment". Chris Reid & Barbara Mullendore, PDA 2001.
2.11.1 Application software

Software backups have to be created following every software change on a system and must document the system's last valid software version. If parts of the software are modified, it is sufficient to only back up the modified part of the application software. Complete software backups still have to be created at regular intervals, however. If software backups are to be created as part of a software change on an existing system or a system reinstallation, they must be created once the installation has been performed. During the course of the project the software version must be backed up and documented at defined milestones, such as at the end of the FAT (i.e. prior to delivery of the system), once the installation qualification (IQ) has been completed, prior to the tests involved in the operational qualification (OQ), and, of course, when the system is handed over to the operator.

Software versions must also be retained in the form of software backups at regular intervals during the creation of new software versions.

Software backups of the application software and configuration parameters must be created.

Labeling software backups

According to the GAMP ® 4 Guide for Validation of Automated Systems, the following information about software backups must be provided, both on the label of the backup medium and in a separate log:

• Creation date
• System name
• Software name
• Software or version name
• Serial number of backup
• Reason for the software backup
• Date of first use
• Date of backup
• Date and signature of the person performing the backup
• Identity of the user
Retaining software backups

At least the two most recent software backups must be archived. For security reasons, these backups must be kept separately from the system (in a fireproof area away from the system, for example).

A suitable backup strategy must be defined, based on the frequency with which changes are made to the software.

The data carrier's shelf life must be defined (based on manufacturer documentation or on publications issued by the Bundesamt für Sicherheit in der Informationstechnologie, the German Federal Office for Information Security) and the software backup must be appropriately migrated by copying it to a new data carrier, for example, before this period expires.

2.11.2 Process data

The data stored in automated systems, such as trends, measured values, or interrupts, should be backed up to external data carriers at regular intervals. This will minimize the risk of data being lost should a fault occur.

Process data is managed in the higher-level HMI system. The requirements of how to handle batch data are described in detail in the GMP engineering manuals for WinCC flexible, WinCC, and PCS 7.
2.12 Using third-party components

If third-party components (hardware and software) specifically tailored to individual customers are used, a supplier audit must be performed in order to check the supplier and their quality management system. It must be confirmed that such hardware components are compatible.

Compatibility must also be confirmed when standard hardware and software components provided by other manufacturers are used.

Note

NAMUR Recommendation 72 contains a great deal of information as regards the auditing of product suppliers. Methods of auditing service and solution providers can be found in the GAMP ® 4 Guide, Appendix M2, among other publications.
3 System specification

This chapter focuses on the selection criteria for the hardware and software. The activities for the selection of products, product variants, and system constellations are performed in the specification stage of an automated system. This is demonstrated in the following life-cycle model by the highlighting in the left-hand area.

Figure 3-1: Life-cycle model: System specification

Hardware and software selection criteria for production plants with mandatory GMP are modularity, scalability, and support functions for the configuration management of hardware and software components (e.g. version control, change control), with the aim of minimizing validation work.
3.1 Selection criteria for SIMOTION hardware

SIMOTION is a motion control system with a scalable power spectrum for automation in production sectors. The CPU is selected based on the requirements of the drive and automation task and the hardware platform preferred by the machine manufacturer or plant operator.

Bus communication

SIMOTION CPUs feature standard interfaces for connecting to networks based on PROFIBUS DP and Industrial Ethernet, with PROFINET I/O as an option. The interfaces based on PROFIBUS DP and PROFINET I/O can also be operated isochronously.

HMI

Communications services are already integrated into the basic functions of SIMOTION CPUs, supporting user-friendly data exchange with SIMATIC HMI devices, such as touch panels (TP), operator panels (OP), or multi panels (MP). These devices can be connected via PROFIBUS, PROFINET I/O, or Ethernet interfaces, and are configured using WinCC flexible. With the licensed SIMATIC NET communications software, an OPC interface is available for accessing SIMOTION from other Windows- or Linux-based HMI systems.

Note

For technical details of the SIMOTION motion control systems and the range of I/Os, please refer to the current SIMOTION Catalog PM 10.
3.2 Required and optional software packages

This chapter introduces basic SIMOTION functions along with additional software packages which play their part in meeting the "Requirements of automated systems in the GMP environment" as described in Chapter 2.

3.2.1 SIMOTION basic engineering software

The SIMOTION SCOUT engineering software package covers engineering, including drive commissioning and runtime software, for all SIMOTION hardware platforms.

Through integration in the SIMATIC Manager, SIMOTION SCOUT has many features in common with the familiar STEP 7 programming environment, such as:

- The common system configuration with HW Config
- The common bus configuration with NetPro
- The common project data management
- The integration of WinCC flexible into the STEP 7 or SCOUT project

The SIMATIC engineering software STEP 7 SIMATIC Manager must be installed before the installation of SIMOTION SCOUT so that it can be integrated in the SIMATIC Manager.

If only SIMOTION projects are executed, the stand-alone version of SIMOTION SCOUT should be used. This version contains all required compatible sections of the SIMATIC Manager.

3.2.2 HMI and communication basic engineering software

SIMATIC WinCC flexible is the software package for engineering HMI panels and HMI PCs. As part of the Totally Integrated Automation (TIA) concept, the WinCC flexible project can be integrated into the SIMATIC Manager along with the SIMOTION project. The subsequent integration of an already existing project is also supported. After integration of the WinCC flexible project, the tags of the SIMOTION project are accessed for connecting the tags in WinCC flexible.

- SIMATIC NET is the software package for implementing OPC interfaces in Windows environments.
3.2.3 Additional software packages for user administration, access protection, versioning, audit trail

User administration, access protection

The licensed SIMATIC Logon software is required for user administration in the SIMATIC Manager. The user logs on via the SIMATIC Logon Service dialog box with user ID and password. Only users who have been set up with a user account can access the SIMOTION project. The permissions for operating the machine are configured in the User Administrator. Machine operation is divided into individual operator-control functions that can be enabled for selected user groups. To be able to use these functions, the user must be a member of the appropriate user group. At runtime, the User Administrator checks the operator permissions and SIMATIC Logon checks user authentication. This is explained in more detail in Chapter Access protection.

Electronic signature

*Electronic Signature* software is also provided in SIMOTION as part of the SIMATIC Logon software. The *Electronic Signature* package ships with SIMATIC Logon. This package provides the interface that can be addressed in WinCC or WinCC flexible using script functions for checking the user ID and password.

Version Trail

The licensed SIMATIC Version Trail software is used to version SIMOTION projects. SIMATIC Logon also needs to be installed. Main versions and secondary versions can be specified by the user for the versioning. The criteria which decide whether the SIMOTION project is to be versioned as a main version or a secondary version, must be specified in the configuration management as well as the configuration elements that are to be versioned. To better understand the versioning, the individual versions should be assigned a version name and an informative comment.

**Note**

Versioning using SIMATIC Version Trail relates to complete SIMOTION projects. If an HMI system such as SIMATIC WinCC or SIMATIC WinCC flexible is integrated in the SIMOTION project, the HMI system is also included in the versioning.
Audit trail

For production plants operated in a GMP environment, in 21 CFR Part 11, the FDA specifies the recording of changes to data relevant for GMP including the time stamp, user name, old value, and new value in the form of an audit trail.

The audit trail of operations performed on the machine with SIMOTION is generated in the operator control and monitoring system (HMI).

The audit trail functionality was developed as a software option for WinCC flexible Audit and is part of the HMI. The Audit option for WinCC flexible is described in more detail in the GMP engineering manuals for WinCC flexible.

WinCC flexible Audit works in conjunction with SIMATIC Logon software in terms of access control and user administration.
4 Guidelines for implementing SIMOTION in a GMP environment

This chapter uses examples to explain the configuration of automation systems with SIMOTION SCOUT in a GMP environment. The configuration of HMI systems in a GMP environment is not described in this chapter. For more information on that topic, see the GMP manuals for STEP 7, WinCC, and WinCC flexible.

The following graphic shows the life-cycle model. This chapter focuses on implementation.

Figure 4-1: Life-cycle model: Implementation
4.1 Software categorization of SIMOTION SCOUT

According to the GAMP ® 4 Guide for Validation of Automated Systems, the software components of a system can be assigned to five software categories. Below you will find examples illustrating how this categorization relates to SIMOTION SCOUT.

- Category 1: Operating systems
  - WINDOWS XP
  - WINDOWS VISTA

- Category 2: Firmware
  - Present in the CPU firmware
  - Present in communications processors firmware

- Category 3: Standard software
  - SIMOTION Technology packages
  - SIMOTION Scout ST/-MCC/LAD-/FBD-Editor etc.

- Category 4: Configurable software packages
  - Parameter settings of Scout standard functions

- Kategorie 5: Customized software
  - Units, Programs
4.2 Software installation

The SIMATIC Manager software is installed with SIMOTION SCOUT on the programming device, such as the SIEMENS Field PG, in order to configure the automation system. Hardware requirements and permissible operating systems are described in the appropriate Readme file on the product CD of the corresponding SIMOTION SCOUT software version.

The automation system is either connected via PROFIBUS DP or an Industrial Ethernet interface, which are already integrated in the SIEMENS Field PG.
4.3 Principles of configuration

The automation task is described in detail in the user requirement specification (URS), function specification (FS), and design specification (DS). The hardware is planned for the automation system and the customized software developed on the basis of this documentation.

4.3.1 Software creation using SIMOTION SCOUT

The Work Bench is the central user interface of the SIMOTION SCOUT basic software. The SIMOTION project is created and managed in the SIMATIC Manager or SIMOTION SCOUT Stand-Alone. All objects of the project are displayed in a clear tree topology. During configuration and creation of the program, various editors, for example HW Config, symbol editor, ST/LAD/FBD/MCC editor, etc. are opened.

A user program or project comprises the hardware configuration of one or more automation devices, the application software for the individual automation devices, the configuration of the network connections, and the documentation.
Organization of the SIMOTION user program

Unit concept

In organizing the user program, SIMOTION applies the unit concept, where you can access the global tags, data types, functions (FCs), function blocks (FBs), and programs of other source files. The unit is therefore also referred to as a program source (ST source file, LAD/FBD source file, MCC unit). Reusable subroutines can be compiled in a unit and made available to other units.

The unit properties are configured first when creating a new unit.

Figure 4-3: Insert unit

The dialog box for configuring the properties is opened automatically according to which unit type is selected (ST, MCC, LAD/FBD).
The unit name, author, etc. are defined in the properties dialog box. The Comment field displays the comments relating to the unit. The version number can be entered manually for the version tracking of different unit versions. The know-how protection property can only be viewed here, not modified.

After the properties have been entered, the unit is created as an object in the Programs folder. Double-clicking the unit opens the editor for program creation.

Program organization units (POUs)

Programs within a unit are referred to as program organization units (POUs). A unit is divided into an interface and an implementation section. All elements which can be exported by the unit are defined in the interface section. These include user-defined data types (structures), data (tags and constants), as well as the names of programs, functions, and function blocks. Not only other units, but also external components (e.g. HMI systems) can access these elements.

Data types and data defined in the implementation section are global within the unit and can be used by all POUs within the unit. The implementation section also contains the program code of the POUs. Any POUs not specified in the interface section can only be used within the unit.

It is recommended that a line comment be entered in the header of the unit containing versioning information, creator, date/time, and a change comment. (Details of line comments can be found in the chapter titled Versioning application software.)

Program organization units (POUs) are executable source file sections, i.e. programs, function blocks (FBs), and functions (FCs).
Programs

Programs are code in ST, LAD/FBD, MCC. It is possible to call a program from within another program.

Function blocks (FBs)

Function blocks are recurring functions that can be used several times by the main program. They can also be called from other units. They are parameterizable source file sections with static data (internal tags retain their value between calls). Since an FB has memory, its output parameters can be accessed at any time and from any point in the user program.

Functions (FCs)

Functions are recurring functions that can be used several times by the main program. They can also be called from other units. They are parameterizable source file sections with temporary data. All internal tags lose their value once the function has been exited. The next time the FC is called, they are reinitialized.

Libraries

Libraries provide you with user-defined types, functions, and function blocks, which can be used by all SIMOTION devices. Libraries can be written to in all programming languages. They can be used in all program sources (ST source files, LAD/FBD source files, MCC units).

Note

To take an FB/FC into a library, the unit in which the FB/FC is located must be copied to the library.
## 4.3.2 Programming languages

### Ladder Logic and Function Block Diagram (LAD/FBD) programming languages

The software is created in graphical form not only in LAD, but also in FBD. LAD is read as a circuit diagram, FBD uses the graphic symbols of Boolean algebra. Ladder logic and function block diagrams are ideally suited for the creation of cyclic programs with a large logic component. In principle, it is possible to call functions and function blocks from LAD and FBD. The direct editing of motion commands is not recommended. It is advisable to use PLCopen blocks here. These blocks are designed for integration in logic-oriented programs. LAD and FBD enable simple graphical tracking of the signal flow, equivalent to SIMATIC STEP 7.

LAD and FBD programs consist of elements and boxes, which are graphically connected to form networks. Their operations work mostly according to the rules of Boolean logic or simple arithmetic expressions and equations. They are therefore only suitable for control-relevant programs.

Functions, function blocks, and programs can be programmed in LAD and FBD. A unit can contain several LAD and FBD blocks. Only one POU can be implemented in an LAD or FBD block.

### Structured Text (ST) programming language

As not all operations can be implemented with LAD or FBD (mathematical calculations, for example), Structured Text (ST), a high-level language similar to PASCAL, is used in these cases.

ST is the basic language of SIMOTION and supports all system functions and all functions of the technology packages. Therefore all programming tasks can be solved with ST. Even complex problems, which result in programs with lots of nesting and arithmetic, can be solved. Discretely modular applications with a large number of functions and function blocks can also be clearly handled in ST. System functions and functions of the TOs can be copied from the command library to the ST unit using drag and drop. However, the large number of parameters (especially with TO blocks) results in very long call structures.

In principle, it is possible to create logic programs in ST, but because of the language syntax and the debugging representation, this is difficult to handle for typical control programmers.
Motion Control Chart (MCC) programming language

The programming with MCC (Motion Control Chart) is in the form of a flowchart. Related to this, there are a number of MCC commands available for functions provided by the system. Each command is represented by a rectangular symbol in MCC. Each MCC command can be assigned a user comment.

Complex axis functions can be parameterized using interactive screen forms. MCC supports the graphical nesting of MCC commands (module formation), so that a clear overview can be retained in the display in spite of the large number of functions. Functions, function blocks, and programs can be programmed in MCC. A unit (MCC unit) can have several MCC charts. Only one POU can be implemented in an MCC chart. Because of the graphical display of motion commands, MCC is the optimum solution for the implementation of sequential programs with a large share of motion functionality (e.g. axis enable, axis homing). As a rule, a motion functionality can be implemented quicker and clearer in MCC than in ST.

Generally MCC charts can be derived directly from program flowcharts. However, with increasing size MCC programs quickly become unclear. Module formation is helpful here. Unique comments must be assigned to the modules (details can be found in the section on line comments).

MCC is not very suitable for the implementation of cyclic programs or programs with a large logic or arithmetic component, as the transparency of graphical programming is not needed for this. With a high degree of modularization (lots of functions, function blocks, and programs) programming in MCC becomes unclear, as each MCC chart can only contain one POU.

Rules for achieving a clear structure for SIMOTION application software

It is the software engineer's responsibility to ensure that the user program is structured in a clear and understandable manner. Detailed commenting ensures that the SIMOTION user program is versioned correctly.

- In principle, the architecture should be built up modularly. The aim of a modular design is above all improved readability, transparency, expandability, reusability, and testability of the resulting code (project).
Software interlocks and safety

It is the responsibility of the software engineer to ensure that the user program functions in a safe way under all conditions. Before creating the software, events that could lead to a dangerous reaction must be considered and the appropriate interlocks must be created.

Such events are, for example:

- Restart after line voltage failure
- Opening/closing of valves or similar components without an operator action and acoustic signal
- Starting of motors without an operator action and acoustic signal
- Faulty input from external operator panels must not lead to any incorrect reactions
- Reaction to an unexpected absence of an analog input signal
- Output values of the analog output signals when the CPU unexpectedly goes into STOP
- For legal reasons, emergency stop functions must not be implemented with a standard PLC.

Safety functions for machine and operator protection in accordance with IEC 61800-5-2 and EN 60204-1:

- Safe standstill (safe torque off)
- Safe OFF, category 1 (safe stop 1)
- Safe brake control

Other functions (available as an option package) such as safe limited speed are already integrated in the Sinamics S120 drive module and can be directly controlled by a higher-level safety control.

Note

The selection of the right hardware is also important for ensuring the correct logic decisions are made in a production plant (DI = 0 means 'Sensor open' or 'Wire breakage'). Wire breakage monitoring can also be performed, for example, via signal modules with NAMUR technology.
Integrated HMI

As part of the Totally Integrated Automation (TIA) concept, the WinCC flexible project can be integrated in a SIMOTION project.

**Advantages of the integration:**

- Central overview in the tree topology of the SIMATIC Manager
- Central tag management in the symbol table of the SIMOTION SCOUT
- Central connection overview of all participating components via NetPro
- The WinCC flexible HMI system software can be directly integrated in the SIMATIC Manager during installation (observe the instructions on the installation CD).
- New projects can be edited in direct conjunction with the SIMOTION project.
- The subsequent integration of an already existing project is also supported.
- After integration of the WinCC flexible project, the SIMOTION project's tag management is accessed to connect the tags in WinCC flexible.
- The integration of WinCC flexible is described in greater detail in the SIMATIC WinCC flexible GMP manual.

Software documentation

Detailed commenting is required for good readability of the user program. The following comments are available in the SIMOTION basic software:

- Block title and block comment
- Network title and network comment
- Line comment
- Comments relating to tags or I/Os
- Comments relating to units or libraries
- Symbolic name
Code review

The following activities must be performed for a code review:

- Check whether the previously defined "Device identifier name convention" has been used consistently in the software
- Check whether the previously defined versioning (SOP configuration management) has been performed
- Check whether a unique symbol name has been assigned for all operands
- Check whether each operand has been used in the user program
- Check whether all inputs/outputs are clearly contained in the symbol table
- Check whether the software has been adequately commented
- Check for blocks that are not called, networks without program (dead code)
- Check of the program structure
- Check whether all operands used in the software are set/reset/assigned only once; auxiliary development code must be identifiable
4.4 Configuration management

SIMOTION SCOUT is the basic package for the configuration and programming of SIMOTION. The programming languages integrated in SIMOTION meet standard DIN EN 6.1131-3/IEC 1131-3.

The configuration and program creation is customized, dependent on the requirements of the automation task. The software creation is difficult to trace without version and change management. Therefore, professional configuration management should be used right from the start of software creation.

The configuration management should be described in an SOP. Everyone involved in the project must be trained in the use of this SOP; this produces a common base for the configuration.

Note
Changes made to a plant in operation should always be agreed with the plant operator.

4.4.1 Changing the system and application software

Whenever changes are made to the system software, the compatibility lists of the jointly approved software versions and product-specific general conditions must be taken into account.

Hotfixes, service packs, updates in SIMOTION SCOUT engineering software

Hotfixes, service packs, and updates are provided in the SIMOTION SCOUT engineering software for the purposes of functional expansion or fault correction.

The following forms are available:

- Hotfix for short-term fault correction, refers to a clearly defined fault description.
- Service pack with fault correction (i.e. contains previous hotfixes) and minor functional expansions. A risk evaluation must be performed.
- Update for upgrading to a new version with major functional expansions. A general risk evaluation must be performed.

However, it is generally the case that all updates within a version are upward compatible, i.e. existing functionalities remain unchanged. The validation work with respect to changes is specified within the framework of a risk evaluation.
Upgrades to SIMOTION SCOUT engineering software

The SIMOTION SCOUT engineering software is updated to the next version by means of an upgrade. In addition to the expansion of the range of functions, the upgrade can also mean a supplement to the integrated hardware catalog.

The validation work with respect to changes is specified within the framework of a risk evaluation.

Adaptations may be necessary in the HW Config editor or in the connection settings in the NetPro editor.

Adaptations also need to be made in the application software, such as translating the libraries with the new HW version. Details on this can be found in the upgrade instructions.

The exact software version with which the SIMOTION/STEP 7 project has been created as well as the option packages used are read out to the top node in the project structure in the SIMATIC Manager or SIMOTION SCOUT using the context menu command **Object Properties > Required Software Packages**.

Figure 4-5: Project properties

The *Execute* button checks the current project or the current library for software packages that are required to process certain objects. This function is used when a project or library with an older STEP 7 or SIMOTION version is processed for the first time.

**Note**

With updates, a validation check should be performed on the tools used. Standard tools can be used without any additional validation work.
Guidelines for implementing SIMOTION in a GMP environment

Replacing/changing the hardware/firmware

When upgrading the hardware/firmware, certain measures are required to retain the validated status of the plant:

- Basis for a change is always the change request of the operator
- Description of the causes for the upgrade of the hardware
- Description of the new, improved functions
- Data on the compatibility to the previous version
- Update of the technical documentation
- Installation according to manufacturer documentation (e.g. standard documentation)
- It is recommended that a risk evaluation be carried out beforehand and the main qualification testing points specified.
- Qualification

Identification of SIMOTION hardware

Before replacing modules, the module type must be checked, e.g. manufacturer, version, ordering data. This information is partly printed on the housing of the module. It can also be read as part of the device diagnostics with SIMOTION SCOUT ("General" tab of the device diagnostics).

Figure 4-6  Device diagnostics with SIMOTION SCOUT

Plant-dependent information such as the plant identifier, location identifier, and installation date should be entered in the object properties as a comment relating to the module in HW Config by the project engineer. If a new module type is available, the compatibility to the previous type must be checked. If it is compatible, the module can be replaced directly.
Note
When checking the compatibility of a CPU module, it must be taken into consideration whether faster runtimes have an effect on the user program. The validation work with respect to changes is specified within the framework of a risk evaluation.

If there is no compatibility, configuration measures are required. The SIMOTION project must be updated with the required hardware update. The appropriate adaptations are made in the HW Config editor.

Changing the SIMOTION kernel
The SIMOTION SCOUT engineering software, SIMOTION HW firmware (also known as the SIMOTION kernel), and the technology packages must always have the same software version. The user can install newer or older versions via an update. The SIMOTION kernel in the CPU can be updated to the new version using SIMOTION SCOUT standard functions.

4.4.2 Versioning of the application software

General information about versioning
The software version provides information on the current version of the application software.

The following data are specified for the versioning of the application software:

- Name
- Date
- Version number
- Comment on the change
- Change history

The procedure for versioning is part of configuration management and must be described in an SOP, which is binding for all persons participating in the project.

Examples of and options for versioning in SIMOTION are described below.
4.4.3 Versioning individual blocks

In the GMP environment, comments are used for manually maintaining the application software's change history. An SOP determines the various responsibilities involved. Depending on the programming language, the following comments are available for versioning in SIMOTION:

- Unit comment for units with ST, LAD, FBD source files, MCC units
- Line comment of the ST source file
- Comment block of the MCC unit
- Block comment of the LAD/FBD source file

Versioning with the unit comment

Generally, the function of the unit is described in the unit comment. The unit comment is displayed in the Properties dialog box, which is opened in SIMOTION SCOUT via the Properties context menu.

Figure 4-7: Unit comment in the Properties context menu
Line comment of an ST source file:

To better understand the versioning, the creator must assign the individual versions a version number, creator, and informative comment relating to the change in the line comment in every ST source file.

Figure 4-8: Example of an ST source file line comment

```st
1 //---DP_Diaa---------------------------------------------------------------
2 //STYLMNS AG
3 //j\c\Copyright 2003  All Rights Reserved
4 //---------------------------------------------------------------
5 // project name:          DP_Diaa.xml
6 // file name:             DP_Diaa.xml
7 // library:               SIMOTION / Scout V4.0
8 // version:               03.03.00
9 // restrictions:          
10 // requirements:          
11 // functionality:         information about the various bus nodes is provided from the DP
12 // error diagnostic function
13 //---------------------------------------------------------------
14 // change log table:
15 // version date  expert in charge changes applied
16 01.08.00  06.04.01  S. Soehner  created
17 02.00.00  09.11.04  C. Xisfeld  PRG,FU
18 03.08.00  31.08.06  C. Yecke & B18  changes of term, variables, ...
19 // with reference to the styleguide V3.3
20 //---------------------------------------------------------------
21 INTERFACE
22 //--------- Export------------------------------------------------------------
23 FUNCTION_BLOCK FBOperatingHoursCounter;
24 FUNCTION FBDBleveling;
25 PROGRAM FBGSFSlevelingStartUp;
26 PROGRAM FBGSFSlevelingPerFault;
27 ```
Comment block of the Motion Control Chart (MCC)

To better understand the versioning, the creator must assign the individual versions a version number, creator, and informative comment relating to the change in the comment block in every MCC unit.

Figure 4-9: MCC comment block
Guidelines for implementing SIMOTION in a GMP environment

Block comment of the LAD/FBD source file
To better understand the versioning, the creator must assign the individual versions a version number, creator, and informative comment relating to the change in the block comment in every LAD/FBD source file.

Figure 4-10: LAD/FBD comments

Versioning of the SIMOTION project
To manage the SIMOTION projects (application software) from a GMP perspective through the life cycle of an automation system, the licensed Version Trail software is used.

Note
To use Version Trail, the licensed software SIMATIC Logon is required. A user must be logged on over the SIMATIC Logon Service or over the SIMATIC Manager, with all relevant actions being entered in the change history under this user's name.

With Version Trail, it is possible to version several STEP 7/SIMOTION projects/libraries/multiprojects. Each STEP 7/SIMOTION project is archived under a major and a minor version number. The criteria for the assignment of a major or minor version number must be set down in an SOP along with information on the configuration elements to be versioned. Version Trail ensures a continuous incrementation of the version from the perspective of validation. A completed version can no longer be changed. Projects can also be read back (retrieved) from an archive of project data created with Version Trail (version project).

Version Trail is started with Start > Programs > SIMATIC > STEP 7 > Version Trail. Only closed projects can be archived.
Every archiving action in Version Trail is recorded with the action, project name, version number, time stamp, logged on user ID, and comment. The history can be viewed using the Options > Version History menu command.

The Version Trail software can be directly accessed in the SIMATIC Manager for the archiving/dearchiving of a STEP 7/SIMOTION project. The File > Version > Archive (Dearchive) menu command is used for this.

**Note**

If an HMI system such as WinCC flexible is integrated in the STEP 7 project, the HMI project is also included in the versioning.

The help system of SIMATIC Version Trail has detailed information on creating and managing project versions.
### 4.4.4 Change control

Change control provides information about changes made to application software (Who has changed What and When).

#### Activating the change log

In conjunction with the licensed SIMATIC Logon software, a change log can be activated for the SIMOTION project. Online actions with time stamp and logged-on user ID are recorded in the change log. The input of a comment is mandatory.

The following online actions are recorded:
- Activation/deactivation/configuration of access protection and change log
- Opening/closing projects and libraries
- Memory reset

The change log can be displayed in the SIMATIC Manager. To do this, the object Station, for example, is selected in the tree topology and the context menu **Display Change Log** is called.

**Figure 4-13: Change Log**
Online/offline comparison based on SIMOTION SCOUT

An online/offline comparison is integrated in SIMOTION SCOUT. This comparison can be used to compare the target system and engineering system.

With the online/offline comparison, the SIMOTION SCOUT provides a tool to display differences in the target system (CPU) and project version (offline).

The online/offline comparison is displayed in SIMOTION SCOUT, when connecting to the target system, by means of color coding of the directory tree icons.

Figure 4-14: Online/offline comparison in SIMOTION SCOUT
Icons in the project navigator:
Table 4-1

<table>
<thead>
<tr>
<th>Mode of the element/device</th>
<th>The current state of the element/device is displayed in front of each icon in ONLINE mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Element is in ONLINE mode. Element exists physically in the target system, or the project data in the PC/PG is identical to the project data of the element stored in the target system.</td>
</tr>
<tr>
<td></td>
<td>Element is not in ONLINE mode. Element does not physically exist in the target system, or the connection has been interrupted.</td>
</tr>
<tr>
<td></td>
<td>The device was not selected while changing into ONLINE mode. There is no connection to this device. Select the Target System &gt; Select Target Devices menu command to specify the device with which a connection is to be established in ONLINE mode.</td>
</tr>
<tr>
<td></td>
<td>Element/device is in ONLINE mode. However, there is an inconsistency between the project data in the PC/PG and the project data of the element stored in the target system. The project data in the PC/PG is not identical to the project data in the target system. Possible reasons for inconsistencies:</td>
</tr>
<tr>
<td></td>
<td>• Device/element data was changed in the project in ONLINE mode and the project was not saved or downloaded to the target system (e.g. name changes).</td>
</tr>
<tr>
<td></td>
<td>• Project data uploaded from target system to the PC/PG, data was changed, not saved or downloaded to the target system.</td>
</tr>
<tr>
<td></td>
<td>If there are any inconsistencies, the project should be saved again and the new project data reloaded to the target system.</td>
</tr>
<tr>
<td></td>
<td>Element is in ONLINE mode, but the drive below the control is inconsistent with this. Reasons for this can be, for example:</td>
</tr>
<tr>
<td></td>
<td>• Name of the drive in the PG/PC is different to the name in the target system</td>
</tr>
<tr>
<td></td>
<td>• Different drive objects, e.g. servo parameterized in the PG/PC and vector in the target system</td>
</tr>
<tr>
<td></td>
<td>• Topology is different, e.g. DRIVE-CLiQ cables interchanged</td>
</tr>
<tr>
<td></td>
<td>Element is in ONLINE mode, but parameter values in the PG/PC are different to the values in the target system. This inconsistency is usually not critical as the parameters in OFFLINE mode often have different values to those in ONLINE mode.</td>
</tr>
<tr>
<td></td>
<td>During the program execution monitoring for MCC charts, the element just addressed by the chart or the active element is selected in the project navigator. For the program execution monitoring, the single-step option must be activated under the properties of the MCC chart. The execution is started with the MCC Chart &gt; Single-Step menu command.</td>
</tr>
</tbody>
</table>
Code comparison of different versions

Different versions of SIMOTION user programs that have been created with the ST programming language can be saved as text format and compared using a text comparison program. The differences are displayed in this program in color-coded format.

4.5 Access protection

4.5.1 Access protection for a SIMOTION project

The SIMATIC Logon option package can be used to provide SIMOTION projects with access protection. If SIMOTION projects are opened in SIMOTION SCOUT or in the SIMATIC Manager with integrated SIMOTION SCOUT, the SIMATIC Logon access protection function requests a user ID and password.

The SIMATIC Manager with integrated SIMOTION SCOUT must already be installed in order to use the licensed SIMATIC Logon software. The changes made are recorded in the SIMATIC Logon change log. If an access-protected SIMOTION system or HMI system such as WinCC flexible is integrated in a STEP 7 project, that system’s application software will also be subjected to SIMATIC Logon’s access protection and included in its change log.

Note

A memory reset of the CPU via the hardware cannot be prevented. Therefore, it is the responsibility of the operator to physically secure the automation systems against access (e.g. by keeping them in a locked cabinet).
Application of SIMATIC Logon

As of Version V4.1 of the SIMOTION SCOUT basic software, user administration can be set up in the SIMOTION project in conjunction with the licensed SIMATIC Logon software. The **Options > Access Protection > Manage Users** menu command is selected for this.

Figure 4-15 Application of SIMATIC Logon

The users must be set up in Windows under **Control Panel > Administrative Tools > Computer Management > Local Users and User Groups**. The users and user groups set up in Windows are displayed in the bottom left-hand area.

Two roles with different rights are available in the system for the access protection of the STEP 7 projects:

- **Project administrator**
  The project administrator has rights for the administration of users, for the activation/deactivation of access protection and of the change log, and for the complete editing of the project.

- **Project editor**
  The project editor has the right for complete editing of the project, but cannot administrate users or change the settings for access protection or the change log.
When the access protection is activated for the first time, a dialog box appears for the entry of a project password. The SIMOTION project can only be opened with the project password.

Figure 4-16 Change Project Password

![Change Project Password Dialog Box]

After access protection has been activated for a SIMOTION project, the SIMATIC Logon Service for logging the user on is displayed when the project is opened. SIMATIC Logon checks the user ID and password.

Figure 4-17 One-time logon

![One-time logon Dialog Box]

**Note**

If a user who is not entered in the user administration of the SIMOTION project logs on with the SIMATIC Logon Service, the project password is requested. After entering the correct password, the logged-on, new user is automatically taken into the user administration as a project administrator.

**Note**

It is recommended that the access protection and the change log be activated after the first qualified version.
4.5.2 Protective measures in the application software

Block protection

Know-how protection prevents the code being output in the LAD/FBD/ST/MCC editor. As the blocks run in the automation system as runtime code in compiled format, the source code can be removed from the scope of delivery of the application software. This means that blocks cannot be changed.

Access protection for SIMOTION application software, along with the know-how protection function, helps machine-construction companies to protect their mechatronic expertise. Libraries, LAD/FBD source files, ST source files, MCC units, and all associated MCC charts can only be accessed if the know-how protection log-in ID and corresponding password are entered.

The safety-relevant procedural requirements of 21 CFR Part 11 (e.g. clarification of system users’ responsibilities and access authorizations) are met if SIMATIC Logon is used for the SIMOTION project.

User administration in WinCC flexible via SIMATIC Logon

As of the 2007 version, WinCC flexible offers the opportunity to perform user administration from a central location via SIMATIC Logon. In the WinCC flexible engineering system, user administration is performed via SIMATIC Logon in the runtime security settings.

Figure 4-18: WinCC flexible engineering system - Simatic Logon

Further information about configuring users in SIMATIC Logon can be found in the WinCC flexible documentation and the WinCC flexible GMP engineering manual.
4.6 Audit trail

The audit trail of operations performed on the machine with SIMOTION is generated by the operator control and monitoring system (HMI). The WinCC flexible software should be used, with the "Audit" software package (license), which is subject to an additional charge, for user management with access protection on the HMI and for the audit trail. The Audit software package is already available in WinCC flexible runtime and can be enabled by means of a license disk, which has to be ordered.

Only SIMATIC panels which support data archiving should be used in order to meet the requirements of automation projects in the GMP environment. SIMATIC panels of series 270 and above can be used. An Ethernet connection is required in order to store data on a network drive. The current panels beginning with the 277 series have an on-board Ethernet port. A panel PC or a standard PC can be also used as an alternative to a panel. Panel PCs are offered in a variety of expansion stages (SIMATIC Panel PC, SIEMENS Panel PC, SIMOTION Panel PC).

Further information on this can be found in the manuals titled "Guidelines for Carrying Out Automation Projects in a GMP Environment" for the WinCC flexible, WinCC, or PCS 7 systems.

A validated version of the SIMOTION user program (project) is used to operate the machine. Therefore, changes are no longer made to the automation system during runtime.
4.7 Time synchronization

A consistent time reference must be guaranteed within plant automation, in order to be able to assign clear time stamps when archiving alarms, interrupts, etc. A time synchronization to a standard time is desirable, but not mandatory. A time synchronization is recommended especially for the archiving of data and the analysis of faults (sequence of events, SOE).

The time synchronization is based on the standardized world time UTC (Universal Time Coordinated).

Time synchronization: WinCC - SIMOTION - WinCC flexible

The time synchronization between WinCC and SIMOTION as well as between SIMOTION and WinCC flexible is described below, using an example application.

The time synchronization between WinCC and SIMOTION cannot be implemented automatically via the WinCC "Time Synchronization Editor". This is why the "TimeSync" SIMOTION project is needed to set the system time in SIMOTION from WinCC. SIMOTION then sets this time in the HMI panel (WinCC flexible).

Note

You will find the "TimeSync" project and a more detailed description of time synchronization between SIMOTION and WinCC flexible as well as WinCC in SIMATIC Customer Support, as an FAQ with the entry ID 26234196, or directly via this link:  http://support.automation.siemens.com/WW/view/en/26234196
Structure of the project

Figure 4-19 provides an overview of the project structure. The relevant input and output parameters of the function blocks used and the connections between them are shown.

- WinCC sets the time in the SIMOTION CPU in a programmable refresh time via OPC.
- Time synchronization between SIMOTION and the HMI panel (WinCC flexible) is performed by means of an area pointer.

Figure 4-19

Overview

The SIMOTION SCOUT project consists of two ST units, which are independent of the SIMOTION SCOUT version.

The "TimeSync" unit contains the "timesync_bkg" program, which must be called in a cyclic task, e.g. the background task. WinCC assigns the DATE_AND_TIME tag and the "Set" trigger to this program via OPC, thus setting the time in SIMOTION.

The "FBTimeSyncHMItoSIMOTION" POU contains two function blocks, which are needed to generate the area pointer.
4.7.1 Time synchronization: WinCC - SIMOTION

SIMOTION receives the dtDateTimeWinCC (DATE_AND_TIME) and gboSetSimotionTime (BOOL) tags from WinCC via OPC.

If no SCADA system (WinCC) is available, the additional licensed SIMATIC NET engineering software can be used to simulate synchronization between WinCC and SIMOTION. OPC export in SCOUT can be used to export tags to OPC tag files, which are then used to organize the exchange of data between the OPC client and the SIMOTION device.

SIMOTION real-time clock

The SIMOTION "RTC" system function is then used to set the SIMOTION time. The real-time clock (RTC) on the SIMOTION CPU is defined in the time type "DATE AND TIME".

The time is reset in SIMOTION after the refresh time parameterized in WinCC (e.g. 60 min). This refresh time is adjustable.

Table 4-2 provides an overview of the call parameters for the RTC function.

<table>
<thead>
<tr>
<th>Name of identifier</th>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>Input</td>
<td>BOOL</td>
<td>Set time, default FALSE</td>
</tr>
<tr>
<td>READ</td>
<td>Input</td>
<td>BOOL</td>
<td>Read time, default FALSE</td>
</tr>
<tr>
<td>PDT</td>
<td>Input</td>
<td>DT</td>
<td>Value to which the real-time clock is to be set, default DT#0001-01-01-0:0:0. If the value is less than the default value of the SIMOTION device real-time clock, the real-time clock will be set to its default value (e.g. in SIMOTION C: DT#1994-01-01-00:00:00).</td>
</tr>
<tr>
<td>CDT</td>
<td>Output</td>
<td>DT</td>
<td>Current system time</td>
</tr>
</tbody>
</table>
4.7.2 Time synchronization: SIMOTION - WinCC flexible

The synchronization of the system time and date of an HMI panel with WinCC flexible to the SIMOTION CPU is described below.

An area pointer is required to synchronize WinCC flexible with SIMOTION. The area pointer transfers the date and time from the control to the operator panel.

The control uses the "HMI area pointer" function to load the area pointer's data area. The operator panel reads the data cyclically at the configured acquisition interval and synchronizes itself.

**Figure 4-20**

**Settings in SIMOTION**

The area pointer must be structured as follows:

<table>
<thead>
<tr>
<th>Data word</th>
<th>High-order byte</th>
<th>Low-order byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>n+0</td>
<td>7 . . . . . . . 0</td>
<td>7 . . . . . . . 0</td>
</tr>
<tr>
<td>n+1</td>
<td>Year (80 to 99/0 to 29)</td>
<td>Month (1 to 12)</td>
</tr>
<tr>
<td>n+2</td>
<td>Day (1 to 31)</td>
<td>Hour (0 to 23)</td>
</tr>
<tr>
<td>n+3</td>
<td>Minute (0 to 59)</td>
<td>Second (0 to 59)</td>
</tr>
<tr>
<td>n+4</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>n+5</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

1) The two data words must be present in the data area in order to ensure a correlation of the data format with WinCC flexible and to avoid the reading of wrong information.

The "HMI area pointer" function block is required in SIMOTION to describe the area pointer.
Guidelines for implementing SIMOTION in a GMP environment

**Note**
Note that when you enter the year, values 80 to 99 result in years 1980 through 1999 and the values 0 to 29 result in the years 2000 through 2029.

**Settings in WinCC flexible**
The settings required in WinCC flexible to perform time synchronization are described below.

**Communications settings:**
1. Select: Project → Device → Communication → Connections

![Figure 4-21](image)

2. Make the following settings:
   - **Name:** Connection_1
   - **Active:** On
   - **Communication driver:** SIMOTION
   - **Station:** timesync\SIMOTION D
   - **Partner:** D435
   - **Online:** On

![Figure 4-22: Connections](image)
3. Select the Area pointer tab and then the area pointer from the TimeSync SIMOTION project for "All connections". Finally, specify the required acquisition cycle time.

Figure 4-23: Area pointer

<table>
<thead>
<tr>
<th>Connection</th>
<th>Name</th>
<th>Symbol</th>
<th>Length</th>
<th>Acquisition cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection_1</td>
<td>Data/Time PLC</td>
<td>time_hmi_area_pointer</td>
<td>5 ms</td>
<td>10 s</td>
</tr>
<tr>
<td>&lt;undefined&gt;</td>
<td>Project ID</td>
<td>&lt;undefined&gt;</td>
<td>1</td>
<td>&lt;undefined&gt;</td>
</tr>
<tr>
<td>&lt;undefined&gt;</td>
<td>Screen number</td>
<td>&lt;undefined&gt;</td>
<td>5</td>
<td>&lt;undefined&gt;</td>
</tr>
</tbody>
</table>

**Note**

Do not configure an too short an acquisition cycle for the date/time area pointer, because this will have a negative influence on operator panel performance. Values of greater than 10 minutes are recommended.
4.8 Interrupt and alarm handling

4.8.1 Time stamping of GMP-relevant data

Not only GMP-relevant process data, but also alarm events are archived for production plants in the GMP environment. The process values are archived in the WinCC SCADA system or the WinCC flexible operator control and monitoring system (HMI).

The time stamp of an archived process value is specified by the SCADA or HMI system. The time stamp of an archived alarm depends on the alarm procedure used:

- In the discrete alarm procedure, the time stamp is specified by the SCADA or HMI system; therefore, the acquisition cycle, bus runtime, and processing time are contained in the time stamp.

- In the alarm number procedure, the time stamp is specified by the SIMOTION CPU; it refers to the time when the alarm was triggered in the control program.

Further details on time stamping can be found in the section on time synchronization.

4.8.2 Alarm procedure

The use of the alarm procedure is a decisive criterion when selecting hardware components.

Discrete alarm procedure

An alarm is detected by the control and signaled to the operator panel via a bit change in a tag. The alarms can be acknowledged either via the HMI or via the application.

The time stamp of the alarm is specified by the operator panel when evaluating the tag. In the discrete alarm procedure, the acquisition cycle, bus runtime, and processing time are contained in the time stamp. Alarms that are present for a shorter time than the acquisition cycle are not detected by the operator panel.

The bus load is high in this alarm procedure as the operating system has to constantly poll the alarm tag.
Alarm number procedure (Alarm_S)

The Alarm_S procedure can be used to configure user-defined error messages and notices, to trigger these, and to display them on the HMI. The control only sends an alarm number to the operator panels registered for Alarm_S.

In SIMOTION, these alarms are configured via the alarm blocks ALARM_S/SQ.

The time stamp is specified by the SIMOTION CPU; it refers to the time when the interrupt block was called.

It must be noted that, with regard to the alarm number procedure, a maximum of 40 alarms can be triggered simultaneously by SIMOTION.

Time stamping via the alarm number procedure is much more precise. The bus load is low (CPU signals active).

In systems with several operator stations, alarms should be sent in the correct chronological sequence (alarm number procedure) for the coordinated acknowledgment and sending of alarms.

Alarms of the SIMOTION technology objects (TO interrupts)

Technology interrupts are generated by the technology objects in the SIMOTION RT. The data volume of the available technological interrupts depends on the loaded technology packages and the technology objects contained therein. A default system response is defined for each technological interrupt. A system interrupt is triggered in the execution system when a technological interrupt occurs. Alternatively, another response can be specified in the interrupt configuration (e.g. SIMOTION device in STOP mode, stop axis). Technological interrupts can be visualized in the SIMOTION SCOUT and in the HMI, once they have been connected to one another.
4.8.3 Alarm configuration in SIMOTION and WinCC flexible

Alarm_S setting in SIMOTION

During alarm configuration in SIMOTION SCOUT, the saved alarm texts and attributes are stored in the SIMOTION project data. The configured alarms are available throughout the entire project. WinCC flexible automatically imports the required data and transfers it to the operator panel later.

The condition for displaying the alarm is defined in the user program (MCC chart or ST source file).

To configure an alarm:
Select Target System > Configure Alarms. The Alarm Configuration window appears.
1. Activate the Alarm_S checkbox.
2. Click the New button. The Edit Alarm window is displayed.
3. Enter the symbolic name under Symbol, followed by the alarm text and infotext.
4. Select the alarm type and display class.
5. Click OK to close the window. The alarm is displayed in the table.

Alarm_S alarms are programmed in the user program with the following functions:
- _alarmSId (generation of an alarm without acknowledgment)
- _alarmSqId (generation of an alarm with acknowledgment)
- _alarmScId (query of alarm status)
Alarm_S settings in WinCC flexible

During alarm configuration in SIMOTION, the saved texts and attributes are stored in the SIMOTION project data. WinCC flexible automatically imports the required data and transfers it to the operator panel later.

In WinCC flexible, the display of ALARM_S alarms can be filtered via display classes. In the project window select "Alarm Management → Settings" and then double-click "Alarm Settings".

Figure 4-24

The existing connections are displayed in the "Alarm procedures" area.

In the line of the required connection, select the field in the "ALARM_S" column and open the selection dialog with the selection button. Select the desired display class.

Figure 4-25

Alarm windows are used to display the alarms.
Configuring an alarm window:
Insert an alarm window object in a screen of the WinCC flexible project.
Click the "General" group in the Properties window.
In the "Alarm classes" area, specify the alarm classes for which alarms are to be displayed in the alarm window.

Figure 4-26

"S7 Alarm" must be selected in order to display the interrupt alarms in the interrupt field.
It is important that an Acknowledge button be created. This can also be created in the alarm window under "Properties".

Figure 4-27

Discrete alarm procedure settings in SIMOTION
The following tag must be declared in SIMOTION:
alm_trg : WORD;
This tag is the trigger for the alarm. A bit of this tag is assigned to each alarm in WinCC flexible.
Discrete alarm procedure settings in WinCC flexible

At least the following properties must be defined for a new discrete alarm:
- Alarm text
- Alarm class
- Trigger tag and bit number

The general procedure is described below, along with the exact settings required for pharmaceutical applications.

General procedure

1. Open the "Alarm Management" group in the project window and select the "Add Discrete alarm" command in the "Discrete Alarms" context menu. The "Discrete Alarms" editor opens and a new discrete alarm is displayed.

2. If the Properties window is not open, select the "Properties" command in the "View" menu. In the Properties window, select the "General" group and enter the alarm text. You can format the alarm text character by character and insert static text fields for tag values or text lists in the text.
3. Select the alarm class, e.g. Warning, Pharmaceutical, etc. and select the "Properties > Trigger" group in the Properties window. Select the tag and the bit which will trigger the alarm.

![Diagram of Discrete Alarms](Image)

**Notice**

Note the method used to count bits in the control used when specifying the bit. For more information, refer to the section on "Communication" in the online help for the relevant control.

**Note**

If an object (such as a tag or an alarm class) you want to select does not yet exist, you can create it directly in the "Object list" and change its properties at a later time.
Additional settings for discrete alarms

1. To assign the alarm to an alarm group, select the alarm group in the "General" group in the Properties window.

   Figure 4-31

   ![Discrete alarm (Discrete alarm) General](image1)

2. In order to ensure automatic alarm logging, select the "Properties > Process" group in the Properties window and select "Report". Check whether "Report" is also activated in the alarm settings.

   Figure 4-32

   ![Discrete alarm (Discrete alarm) Process](image2)

3. To enter an infotext for the alarm, select the "Properties > Infotext" group in the Properties window and enter the required text.

   Figure 4-33

   ![Discrete alarm (Discrete alarm) Infotext](image3)

4. To execute event-controlled tasks, select the "Events" group in the Properties window and configure a function list for the required event.
5. In this example, four interrupt types (alarm classes) have been created for an application:

Table 4-4

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation</th>
<th>Color</th>
<th>Acknowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pharmaceutical</td>
<td>P</td>
<td>Magenta</td>
<td>Yes</td>
</tr>
<tr>
<td>Technical</td>
<td>T</td>
<td>Red</td>
<td>Yes</td>
</tr>
<tr>
<td>Warning</td>
<td>W</td>
<td>Orange</td>
<td>No</td>
</tr>
<tr>
<td>Event</td>
<td>E</td>
<td>Yellow</td>
<td>No</td>
</tr>
</tbody>
</table>

The interrupt types can be defined as follows in WinCC flexible:

As can be seen in the above figure, the interrupt type, the color, and the acknowledgement can be selected.

The alarm classes Pharmaceutical, Technical, Warning, and Event must be selected in the "Alarm Window".
Settings for alarms of the SIMOTION technology objects (TO interrupts)

In order to display the interrupts in SIMOTION SCOUT, this must be connected to the target system. All pending alarms are displayed on the Interrupts tab in the detail view and can be acknowledged there.

Figure 4-36: Display of TO interrupts in SIMOTION SCOUT

To display the technological interrupts in the HMI, they must be activated in the WinCC flexible configuration.
Open the "Alarm Management" group in the project window and select "Settings" → "Alarm Settings" in the context menu.

Figure 4-37

The technological interrupts can then be activated in the window that appears.

Figure 4-38: Activation of TO interrupts in WinCC flexible

"S7 Alarm" and "S7 Warning" must be selected in order to display the interrupt alarms in the interrupt field.
It is important that an Acknowledge button be created. This can be created in the properties of the alarm window.

Figure 4-40: Acknowledge button for the alarm window
4.9 Saving and restoring SIMOTION application software

In order to be able to fall back on the created software when required, backup copies of the software versions must be made at regular intervals during the configuration stage. It is also recommended that a backup be made of the system partition of the engineering system on the programming device with the operating system, SIMOTION SCOUT, etc.

If the entire SIMOTION platform has to be replaced, e.g. during a service job or before an upgrade, you have to back up the data in order to be able to use it on the new SIMOTION platform.

You can save data that has been changed during operation and is only saved in the runtime system via the Save tags function.

4.9.1 Saving and restoring projects

The SIMOTION application software can be backed up in different ways:

• The File > Save As menu command makes a copy of the SIMOTION project, for example, a backup copy. Adequate memory space should be available. At regular intervals, the With Reorganization check box should be enabled to reduce the storage requirements for the project. A backup copy can be edited with the File > Open menu command.

• In the SIMATIC Manager via the File > Archive menu command...
  The SIMOTION project is archived and compressed with this function. The tool used for archiving can be configured via the Options > Settings > Archive menu command.

• In the licensed Version Trail software, the STEP 7 project is backed up with a major and minor version. Each version can be decompressed with Version Trail.

Figure 4-41: SIMATIC Version Trail
4.9.2 Saving and restoring tags

If you use the Save tags and restore tags function, XML files are created that can be stored in a folder of your choice. With this function, all segments contained in the data sets are saved and restored. In this way, you can restore the original tag values after replacing the hardware.

The following tags can be saved:

- Retentive data, global device tags for MCC, LAD/FBD, and ST defined as RETAIN; all tags of the type VAR_GLOBAL RETAIN
- Non-retentive global unit tags of the interface and implementation sections of the program sources: ST source file, MCC unit, and LAD/FBD source file

If the "Save tags" function is selected for a SIMOTION device, a directory structure is created in the selected path. This directory is assigned the name of the selected SIMOTION device or the selected unit, depending on whether you want to save the tags for the entire SIMOTION device or a unit.
### Saving tags

#### Prerequisites:
- Tag is declared as a retentive or non-retentive global unit tag in the interface section of the source or as a retentive global device tag.
- The project has been downloaded to the target system.
- SIMOTION SCOUT is in online mode.
- Only for unit tags saved via system functions:
  - The system function `_saveUnitDataSet` or `_exportUnitDataSet` must be executed in the ST program for every source so that the global unit data is created as a binary data set in the SIMOTION device.
  - When executing the system function, the memory medium (RAM or memory card) must match the selection made in the Save Tag dialog box.

#### Procedure:

<table>
<thead>
<tr>
<th>No.</th>
<th>Saving tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Switch the SIMOTION device to STOP mode. This ensures that all the tag values have the same time stamp.</td>
</tr>
<tr>
<td>2</td>
<td>Select the SIMOTION device or the unit, e.g. C230-2.</td>
</tr>
<tr>
<td>3</td>
<td>In the context menu, select Save Tags. The Save Tags window opens.</td>
</tr>
<tr>
<td>4</td>
<td>Activate the Retain tags checkbox if you want to save all retentive tags from the current data memory.</td>
</tr>
<tr>
<td>5</td>
<td>Activate the Unit tags (ST)/global tags (MCC, LAD/FBD) saved via system functions checkbox if you want to save the previously saved or exported data sets.</td>
</tr>
<tr>
<td>6</td>
<td>Select whether the data sets should be read from the memory card or from the RAM disk. Ensure that this selection corresponds to the memory location where the data sets were saved by the <code>_saveUnitDataSet</code>/_<code>exportUnitDataSet</code> function.</td>
</tr>
<tr>
<td>7</td>
<td>Select the desired settings and click OK. The Select Folder window appears.</td>
</tr>
<tr>
<td>8</td>
<td>Select the memory location for the backup files and confirm with OK. The tags are saved.</td>
</tr>
</tbody>
</table>

#### Note
For more detailed information, refer to the SIMOTION SCOUT online help.
Restoring tags

Prerequisites:

- The interface section of the source has not been changed.
- The tags have been saved.
- The project has been downloaded to the target system.
- The SIMOTION device is in the STOP mode.
- SIMOTION SCOUT is in online mode.
- Only for unit tags saved via system functions:
  - The system function _loadUnitDataSet or _importUnitDataSet must be executed in the ST program for every source in order to initialize the tags with the values from the backup files.
  - When executing the system function, the memory medium (RAM or memory card) must match the selection made in the Save Tag dialog box.

Note

General conditions for restoring tags if there are differences between the backup file and the project:

If the unit tag or the global device tag exists in the project, but not in the backup file, then the contents of the unit tag in the unit or the global device tag are retained unchanged.

If the unit tag or the global device tag exists in the backup file, but not in the unit or on the device, then the tag is ignored.

If the unit tag or the global device tag exists in the backup file and on the unit or device, but is of a different data type, then a type conversion is performed between ANY_INT <--> ANY_INT, ANY_INT <-> ANY_REAL, ANY_REAL <-> ANY_REAL. If a type conversion is possible, the value in the backup file must fit in the new value range of the unit tag or global device tag, otherwise type conversion is initialized. If a type conversion is not possible (e.g. date, time), then the unit tag or global device tag is initialized.

If the unit tag or the global device tag exists in the backup file and on the unit or device and the data type is identical, then the contents of the tags from the backup file is transferred to the unit tag of the unit or to the global device tag on the device.
Procedure:

<table>
<thead>
<tr>
<th>No.</th>
<th>Restoring tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the SIMOTION device or the unit, e.g. C230-2.</td>
</tr>
<tr>
<td>2</td>
<td>In the context menu, select Restore Tags. The Restore Tags window opens.</td>
</tr>
<tr>
<td>3</td>
<td>Activate the Retain tags checkbox if you want to restore all previously saved retentive tags from the current data memory.</td>
</tr>
<tr>
<td>4</td>
<td>Activate the Unit tags (ST)/global tags (MCC, LAD/FBD) saved via system functions checkbox if you want to restore the previously saved data sets.</td>
</tr>
<tr>
<td>5</td>
<td>Select whether the data sets should be written to the memory card or to the RAM disk.</td>
</tr>
<tr>
<td>6</td>
<td>Select the desired settings and click OK. The Select Folder window appears.</td>
</tr>
<tr>
<td>7</td>
<td>Select the folder in which the backup files are stored.</td>
</tr>
<tr>
<td>8</td>
<td>Select the backup file. Note that with the selection of the backup file, only tags that were previously saved on the device can be restored to the device.</td>
</tr>
<tr>
<td>9</td>
<td>Click OK to confirm the following window.</td>
</tr>
<tr>
<td>10</td>
<td>Switch the SIMOTION device to RUN mode.</td>
</tr>
<tr>
<td>11</td>
<td>Execute the system function <code>_loadUnitDataSet</code> or <code>_importUnitDataSet</code> to initialize the tags with the values from the backup files.</td>
</tr>
</tbody>
</table>

**Note**

For more detailed information, refer to the SIMOTION ST (Structured Text) Programming Manual.
5 Supplementary software components

5.1 SIMOTION SCOUT diagnostic tools

A number of easy-to-use functions for hardware and software diagnostics are already integrated in the SIMOTION SCOUT engineering software. These functions are mainly used for commissioning the user program and for diagnostics when an error occurs.

A wide range of diagnostic functions can be used for the operation of SIMOTION devices in the ONLINE mode. These diagnostic functions are summarized in the diagnostics overview.

The Diagnostics overview is a tab in the detail view and is available by default in ONLINE mode. You can call up detail displays from here.

An Alarms tab is also available in the detail view, which provides a tabular overview of the following:

- Technological interrupts (from technology objects)
- Alarm_S alarms (from user programs)
  The interrupt alarms can be acknowledged either individually or all together.

You can record signal charts with the trace tool. The values of system tags can be recorded during runtime for diagnostic purposes.
5.1.1 SIMOTION SCOUT diagnostics overview

In the detail view, the Diagnostics overview tab displays a tabular overview of all devices and drives available in the project. You can use the diagnostics overview to display more detailed device diagnostics in the working area. The diagnostics overview is automatically displayed when you switch to ONLINE mode.

Figure 5-1: Diagnostics overview in SIMOTION SCOUT
SIMOTION SCOUT device diagnostics

The device diagnostics function provides a detailed display of the diagnostic events of the individual devices.

**Note**
The device diagnostics and the diagnostics overview are only active in ONLINE mode.

1. To open the device diagnostics:
2. Select Project > Connect to Target System. The PC/PG is connected to the target system.
3. Select the device in the project navigator or on the Diagnostics overview tab in the detail view.
4. Select Target System > Device Diagnostics. The device diagnostics is displayed in the working area.

Figure 5-2: Device diagnostics in SIMOTION SCOUT

The **General** tab displays the basic information about the SIMOTION devices.

The module states are logged on the **Diagnostics buffer** tab. The diagnostic events which have occurred are listed in chronological order.

All devices configured in HW Config and devices (drive units, CPU) configured as slaves are displayed on the **Slaves** tab.

The status and runtime for the tasks created in the project are displayed on the **Task runtimes** tab. The resolution of the display depends on the set position control cycle clock.
On the **System utilization** tab, the current memory utilization in KB and percent, the available memory in KB, and the CPU utilization are displayed.

You can save your own text strings in the RT system on the **Userlog** device diagnostics tab. This is necessary or useful, for example, if changes that are to be documented are made in the SIMOTION system to a machine which has already been commissioned. For the documentation, you can describe the changes in the SCOUT and load them into the RT or in the ROM of the target CPU. These text strings can be read out and can thus record the RT state.

In addition to the user-defined UserLog file, the SIMOTION CPU (C2xx, P350, D4x5) has a SysLog file on the **Syslog** device diagnostics tab. It contains the ROM actions which enable a subsequent diagnosis to be performed.

**Software diagnostics using the watch table**

Tag tables (the "watch table" in SIMOTION SCOUT) and the status display of the individual blocks are used for software diagnostics or software tests. Various examples are listed below. The test options are described in detail in the SIMOTION online help under **DIAGNOSTICS**.

**Tag tables**

Tags are arbitrarily combined into tag tables in order to read out their current values. The values can be manipulated via the control function. The tag tables are saved under a name and can be called at any time.

**Figure 5-3 Watch table**
Note
The control function in the tag table is very dangerous and only permitted during the user program’s development stage.

Status monitoring of individual units
Individual units can be viewed online for the purpose of monitoring program execution. The unit is opened in the appropriate editor for this. The status of the unit is activated or deactivated in the “ST source file”, “MCC chart”, or “LAD/FBD program” menu. Active elements are now highlighted in color in the display and the status of the different tags is displayed.
5.1.2 **SIMOTION trace tool**

With the trace tool, you can record and save signal charts of inputs and outputs and the contents of tags. The tool is used, for implementing and documenting the optimization of, for example, axes. You can set the recording duration, select the tags to be recorded, define trigger conditions, parameterize delays, and switch between different curve displays and scales. The trace tool can be called via the menu command: Target System -> Trace.

Figure 5-4: SIMOTION trace tool
6 Supporting functions during qualification

6.1 Introduction

The following graphic shows the life-cycle model. The focus of this chapter, the selection criteria, are typified by the system commissioning.

Figure 6-1

The aim of the qualification is to provide documented proof that the system was set up according to the specifications and that all specified requirements have been met. The qualification describes, executes, and finally evaluates all the activities necessary for this. Various standard functionalities of SIMOTION can be used as support in qualification during IQ and OQ.
6.2 Qualification of automation hardware

The design specification of the installed hardware is used to set up the system according to detailed specifications and adherence to these specifications can and must be verified during the subsequent system tests. The design specification describes all hardware used with information such as order number, firmware/version, serial number, installation location, etc. The components such as the CPUs used, input and output cards, interfaces to third-party systems, etc. are listed below.

Qualification of field devices

In the qualification of field devices, checks are necessary to ensure that the requirements of the hardware design specification were implemented. This means verifying the following:

- Manufacturer
- Order number
- Serial number
- Function of the field device
- Installation location
- Measuring point name
- Electrical connection type/bus type
- Physical connection type
- Address number
- Unit
- Measuring range

**Note**
A visual inspection is made of the field device.
Qualification of the automation hardware

In the qualification of automation hardware, checks are necessary to ensure that the requirements of the hardware design specification were implemented. All the hardware components as defined in the design specification must be configured in the SIMOTION hardware configuration. This includes:

- Number of racks
- Verifying the hardware components used (CPU, CP, etc.)
- Number of distributed I/O stations
- Interfaces to third-party systems
- Verifying the order numbers of the hardware used
- Address description
- Symbolic naming of inputs/outputs
- Etc.

Note

The HW Config can be printed and used to verify the qualification (IQ/OQ) of the installed hardware components. A visual inspection of the installed hardware can be made at the same time. The hardware used must match that in the control cabinet documentation.

Qualification of the network structure

In the qualification of network structure, checks are necessary to ensure that the requirements of the hardware design specification were implemented. All the connections must be configured in the SIMATIC MANAGER NetPro configuration of SIMOTION. This includes:

- Name of: station, PC, CPU, clients, etc.
- Communication module, type of connection, and communication partner (Ethernet, PROFINET, PROFIBUS, MPI, serial, etc.)
- MAC address (when using the ISO protocol on the plant bus)
- TCP/IP address and subnet mask (when using clients)
- PROFIBUS addresses
- Etc.

Note

The SIMOTION NetPro configuration can be printed and used to verify the qualification (IQ/OQ) of the configured network structure. A visual inspection of the configured network structure can be made at the same time.
6.3 Qualification of automation software

6.3.1 Qualification of standard software

In the qualification of standard software used, checks are necessary to ensure that the requirements of the software design specification were implemented. This includes:

- Operating system of the engineering system
- SIMOTION engineering system (SCOUT, SIMATIC Manager)
- SIMATIC options (Logon, etc.)
- Standard libraries

**Note (operating system)**

The installed software can be verified by operating system functions. The information can be found under Control Panel > Add or Remove Programs. All installed software components are displayed here. A screenshot can be printed and used for the qualification (IQ/OQ).

**Note (SIMOTION software)**

The verification of installed SIMOTION software can be performed with the "Installed software" software tool. This tool provides information on the SIMOTION software currently installed on the computer. The installed components can be printed and used for the qualification (IQ/OQ).
6.3.2 System software common to SIMOTION and SIMATIC

Installed software

The installed software packages must be documented for the engineering system used to configure the automation systems. Detailed documentation of the installed SIMATIC software can be found under Programs > SIMATIC > Product Notes > Installed Software.

Figure 6-2

The list provides information on the software products, software components, and DLLs installed on the local computer. This information can be used, for example, to include the installed software in the installation qualification.
Installed licenses

The Automation License Manager program provides information on the licenses currently installed on the engineering system. To view the licenses, open the Automation License Manager. Select the PC partition on which the licenses are installed in the Explorer window on the left-hand side. All the available SIMATIC/SIMOTION licenses of the system are now displayed on the right-hand side of the window.

Figure 6-3

In conjunction with the SIMATIC Logon option, access protection can be activated for the Automation License Manager application. To do this, select File > Settings > Activate SIMATIC Logon Access Protection.

When the Automation License Manager is started, the SIMATIC Logon Service opens for logging on a user. SIMATIC Logon checks the user ID and password.

**Note**

The installed licenses must correspond to the requirements defined in the specification.

The installed licenses can be printed and used to verify the qualification (IQ/OQ).
6.3.3 Qualification of the application software

In the qualification of application software, checks are necessary to ensure that the requirements of the software design specification were implemented. Test descriptions must be agreed with the user (e.g. for FAT/SAT) and generated. These test descriptions must be created individually to meet the software design specifications.

As a minimum, the following must be checked and tested and can be used as a reference for the qualification:

- Check of the name of the application software
- Check of the plant hierarchy (plant, plant section, equipment module, individual control element, etc.)
- Software module test (typical test)
- Check of communication with other nodes (HMI, SCADA, third-party controllers, MES systems, etc.)
- Check of all inputs and outputs
- Check of all control modules (individual control level)
- Check of all equipment phases and equipment operations (technical functions)
- Check of the relationships between modes (MANUAL/AUTOMATIC switchovers, interlocks, start, running, stopped, aborting, completed, etc.)
- Check of the measuring point names
- Check of the visualization structure (P&I representation)
- Check of the operating philosophy (access control, group rights, user rights)
- Check of the archiving concepts (short-term archives, long-term archives)
- Check of the alarm concept
- Check of the trends, curves
- Check of the time synchronization
Index

2
21 CFR Part 11  1-8

A
Access protection  2-8
Access protection for a STEP 7 project  4-25
Application-software backup  2-16
Archiving  2-13
Audit trail  2-11

B
Backup  2-15
Batch data  2-14
Block protection  4-28
Block title and block comment  4-11

C
Change control  2-6, 4-22
Change log  4-22
Changing the system software  4-13
Chip card  2-9
Code review  4-12
Configuration control  2-6
Configuration identification  2-6
Configuration management  2-5, 4-13

D
Diagnostic tools  5-1

E
Electronic signature  2-10, 3-4
EU GMP Guide  1-8, 1-9

F
FAT  1-6
FDA  1-8
Function specification  1-6

G
GAMP  1-8, 1-9
GMP requirements  2-1

H
Hardware categorization  2-2

I
Implementation  1-6

J

K

L
Life-cycle model  1-2
Line comment  4-11
Logging  2-14

N
NAMUR  1-9
Network title and network comment  4-11

P
Password  2-9, 2-10
Process-data backup  2-17

Q
Qualification  1-7, 6-1
Qualification report  1-7
Quality and project plan  1-4, 1-10

R
Replacing/changing the hardware/firmware  4-15, 4-16
Required/optional software-packages  3-3

S
Saving/restoring the system/application software  4-48
Selection criteria for hardware  3-2
Software categorization  1-11, 2-3
Software categorization of SIMOTION  4-2
Software creation  4-4
Software diagnostics  5-4
Software documentation  4-11, 4-12
Software installation  4-3
Software interlocks/safety  4-10
Specification  1-5
Symbolic name  4-12
System specification  3-1

T
Third-party components  2-18
Time synchronization  2-18
Typicals  2-7
<table>
<thead>
<tr>
<th><strong>U</strong></th>
<th><strong>V</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>User administration</td>
<td>Validation plan</td>
</tr>
<tr>
<td>User ID</td>
<td>Version control</td>
</tr>
<tr>
<td>User requirement specification</td>
<td>Version Trail</td>
</tr>
<tr>
<td></td>
<td>Versioning</td>
</tr>
<tr>
<td></td>
<td>Versioning - Project</td>
</tr>
<tr>
<td></td>
<td>Versioning of the application software</td>
</tr>
</tbody>
</table>

- User administration: 2-8
- User ID: 2-9, 2-10
- User requirement specification: 1-5
- Validation plan: 1-4
- Version control: 2-6
- Version Trail: 3-4
- Versioning: 3-4
- Versioning - Project: 4-20
- Versioning of the application software: 4-16