IFA
Institut für Arbeitsschutz der
Deutschen Gesetzlichen Unfallversicherung

# The SISTEMA Cookbook 5

SISTEMA libraries

Version 2.0 (EN) from SISTEMA Version 2.0
onwards

# Contents

# 1    About this guide

The IFA's SISTEMA software application enables libraries of components to be created and imported. This guide describes the creation, publication and use of these libraries.

The terms "database" and "library" are used synonymously in this guide. Two different formats must be distinguished within SISTEMA:

- The "SISTEMA library" (abbreviation: SIS-lib) is in SISTEMA's own format. These are files of the type ".slb".

- From SISTEMA Version 2 onwards, a further format to VDMA Specification 66413 is supported (abbreviation: "VDMA-lib"; see Annex B). These files are of the type ".xml".

This guide is intended for several groups of readers:

- Manufacturers of components (such as safety components or contactors): these companies wish to communicate the reliability parameters of their products to users by means of the libraries. Although these companies may themselves be users of SISTEMA, their role is described in this cookbook as that of **"manufacturer"**.

- Control system manufacturers: these companies use the products referred to above to produce complete safety-related control systems in accordance with EN ISO 13849-1[1]. The manufacturers' libraries are required for analysis of these control systems. The control system manufacturers may however themselves be manufacturers of discrete parts of control systems (produced for example from standard components), and may also create SISTEMA libraries for these parts of control systems for their own use. The role of these control system manufacturers is however described throughout this cookbook as that of **"user"**. This group also includes persons and companies who analyse/test the control systems on behalf of the control system manufacturers.

- Control system operators: during the mission time of the machine, the operator of the machine must also be in possession of information on the components/parts of control systems used within it. The corresponding documentation can be supplemented with a machine-specific SIS-lib for this purpose. The operator's role will be described below as that of **"operator"**.

Introductory information for all target readerships can be found in Chapters 2 and 3. Section 3.1 outlines the typical activities associated with the libraries, with cross-references to more detailed information in the relevant chapters.

Chapters 4 to 8 are intended primarily for the users; Chapters 10 to 13 primarily for the manufacturers. If you use SISTEMA and wish to produce SISTEMA libraries of your own, you should also read Chapters 10 and 11.

Chapter 9 provides information for operators who need to extend a safety control system or replace components of it during the mission time of a machine.

The annexes in Chapter 14 contain further information and useful tables.

**Understanding application of the EN ISO 13849 series of standards is essential for work with SISTEMA and the libraries. The same applies to the use of other analysis tools. Appropriate training is recommended.**

---

[1] EN ISO 13849: Safety of machinery – Safety-related parts of control systems

**Publications available free of charge:**

Information from the IFA on the EN ISO 13849 series of standards is available at:

http://www.dguv.de/webcode/e89507

Further information, guides and numerous examples relating to EN ISO 13849-1 can be found in the IFA reports and SISTEMA examples. These can be downloaded here:

http://www.dguv.de/webcode/e91335

http://www.dguv.de/webcode/e635980

The definition of safety functions, the analysis of control systems and transfer to SISTEMA is described in SISTEMA Cookbooks 1, 4 and 6:

http://www.dguv.de/webcode/e109249

SISTEMA Cookbook 2, "Use of network libraries", and SISTEMA Cookbook 3, "Running several instances of SISTEMA in parallel (terminal server)", this cookbook (Cookbook 5) and future publications can also be found here.

The IFA publications referred to can all be downloaded free of charge in PDF format in English and German. A small number of the publications are also available in printed form.

# 2    Introduction

SISTEMA (the German acronym stands for "safety of controls on machines") is a free software tool not specific to any control product manufacturer that provides comprehensive support in the analysis of safety in accordance with EN ISO 13849-1 for the development and testing of safety-related parts of machine control systems. The tool, which runs on Microsoft Windows, enables users to model the structure of the safety-related parts of control systems (SRP/CS) based upon the specified "designated architectures", thereby permitting automated calculation of the reliability values at various levels of detail, including that of the attained Performance Level (PL).

Relevant parameters, such as the risk parameters for determining the required Performance Level ($PL_r$), the Category of the SRP/CS, measures against common-cause failures (CCFs) on multi-channel systems, the average component quality ($MTTF_D$) and the average test quality (DC) of components are entered step by step in input dialogs. Once the necessary data have been entered into SISTEMA, the results are computed and displayed. A practical advantage in use is that the effects of a given parameter change upon the system as a whole are displayed immediately in the user interface. The otherwise time-consuming consulting of tables and solving of formulae is performed almost entirely by the software. Users of SISTEMA can therefore vary parameter values with a minimum of effort in order to assess the effect of modifications. The final results are summarized in a report, which can be printed out.

User-friendly library functions complete SISTEMA's range of features. Almost all leading manufacturers of components provide SISTEMA libraries (SIS-lib) of their products' data. The first libraries became available in 2008; by the middle of 2016, 65 companies were already listed on the IFA's website with links to their libraries.

Libraries can be stored either on a local PC or centrally on a company's servers, from where they can be loaded. Users of SISTEMA can however create libraries of their own, in which they can manage parts of control systems that they have developed themselves, or frequently used components. This is possible only with SISTEMA libraries, however; VDMA libraries may be made available only by the manufacturer of the control products themselves.

Like those of many CAE[2] programs, SISTEMA's libraries are an essential factor for effective, efficient and error-free engineering. The engineering task in this context is the analysis of safety controls. Such analyses are however time-consuming in the early stages. For a once-off analysis, the reliability data could be read off a data sheet and typed into a SISTEMA project. Where numerous safety functions are used repeatedly in the same or slightly modified form, however, it must be possible for the analysis to be performed rapidly and automatically. Ideally, a complex part of a control system is defined, documented and verified only once. It is then archived in the SIS-lib and retrieved again in the future for re-use. This re-use in conjunction with the standardization of parts of control systems results in sustained time savings and avoidance of error.

The **provision of reliability parameter values for the manufacturers' products** ($PFH_D$[3], SIL, PL, $MTTF_D$, $B10_D$, etc.) is one of the typical applications of libraries and is one reason

---

[2] Computer-aided engineering

[3] This cookbook follows the new convention used in the third, revised edition of ISO 13849-1, published in 2015, in place of the "PFH" (without the index "D") used up until now. The new convention is "$PFH_D$" (refer also to 0).

why the SISTEMA library was defined. The database published in 2012 in VDMA Specification 66413 (see Annex B) was defined for the same purpose. An exclusive feature of the SISTEMA solution is the additional facility **for parts of control systems pre-engineered by the user to be saved together with all their components, their parameters, and the associated documentation** – in a similar way to the circuit diagram macros of CAE programs. Not only is the final result ($PFH_D$ and Performance Level) stored, but also the complete single or two-channel structure and the documentation of all parts contained in the system. This supports use in the field: such "macros" cannot always be used as-is, since a discrete component of a complex part of a control system must be substituted, added or removed. SISTEMA supports this redesign process, however with the SIS-libs.

Manufacturers who initially only intended to offer SIS-libs for their products can in addition use SIS-libs to distribute more complex parts of control systems and their components in the form of application examples (subsystems) to their customers.

SISTEMA was developed by the IFA on behalf of the former German Social Accident Insurance Institution in the printing and paper processing industry, with the use of the free software components stated in the readme file. The data structure and data fields of the SIS-lib were defined solely by the IFA; they are therefore proprietary. Only the IFA develops them further; it does so however taking account of proposals made by manufacturers, drawing upon experience from the field and technical progress (such as VDMA 66413).

The documentation of the SISTEMA data structure for SIS-lib and projects provided for developers is available from the IFA free of charge upon request. Manufacturers and users are therefore able to use the SISTEMA platform by developing interfaces to SISTEMA.

# 3     Library structure and interfaces to it

## 3.1    Interaction between manufacturers, users and the IFA

This section outlines the relationship between manufacturers, users and the IFA (Figure 1). For each activity (abbreviated by a capital letter), reference is made to the relevant chapter of this cookbook. Introductory information on the concept and user interface of the libraries can be found in Chapters 3, 4 and 5.

Manufacturers' libraries are first generated by the manufacturers (activity A, see Chapters 10 to 12). These libraries must be made available for use (usually by publication on the manufacturer's website). These pages can most easily be found via the IFA's list of links (activity B, Chapter 13).

Users wishing to make use of libraries first download them to PC from the manufacturer's website via this list of links (activity C, Chapter 6) and can then store them on their own servers (Chapter 6).

Figure 1: Interaction between manufacturers, users and the IFA
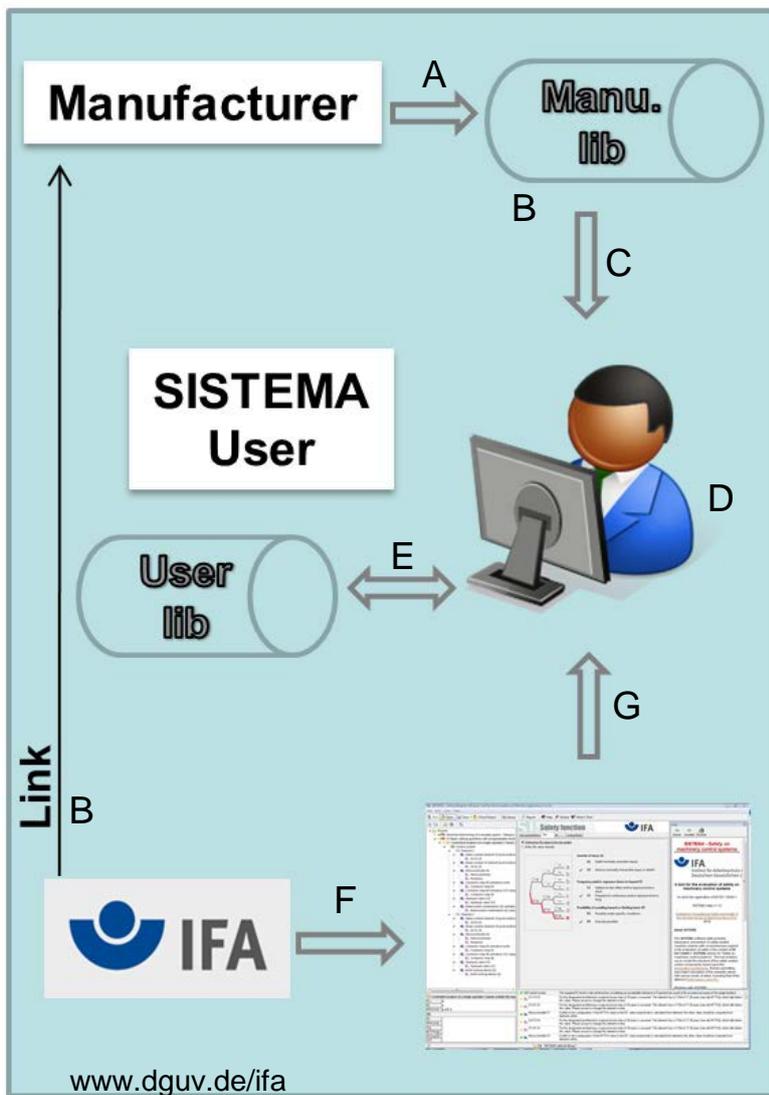
Application of the libraries (activity D) is described in Chapter 7 and 8. Further information for operators can be found in Chapter 9. Users of SISTEMA can also generate their own user libraries, as has already been described (activity E, Chapters 10 and 11).

The IFA develops SISTEMA (activity F) and hosts it for download on its website (activity G). Other enterprises or organizations are not permitted to host SISTEMA in this way.

## 3.2 Sources of reliability parameters

For the analysis of safety-related machine controls, parameters are required for standard components ($MTTF_D$, $\lambda_D$, $B_{10D}$ etc.) etc. and for parts of control systems (PL, SIL, $PFH_D$ etc.), which may also be safety components. An index of these abbreviations can be found in 0 of this cookbook. Annex C of EN ISO 13849-1 describes prioritization: preference is to be given to the component manufacturer's data. If this is not possible, typical parameters from standards or databases are used. These values can be obtained from a number of sources:

- The manufacturer's data sheets or certificates: this has been the typical source to date. These values can be copied and entered directly into the SISTEMA project. In order to simplify re-use and to prevent errors from occurring, they should be copied – following careful validation – into a SISTEMA user library. The values can also be entered directly in the tabs of the library.

- The manufacturer's electronic library: this is the ideal source. The manufacturer supplies a SIS-lib or a VDMA-lib; for use, the components/parts of the control system and their values are copied into the relevant SISTEMA project. Alternatively, the manufacturer's components that are frequently used can be copied into a dedicated SISTEMA user library.

- Values from standards (such as EN ISO 13849-1): these values can also be entered into a SISTEMA user library. The typical values of EN ISO 13849-1, Table C.1 and those of IEC 61508[4] are already stored in a SISTEMA input dialog, from where they can be called up directly (No SIS-lib, see Block/Element object, "$MTTF_D$" tab).

- Databases of components (such as SN 29500[5], IEC/TR 62380[6]): these values, too, can be imported into a SISTEMA user library. EN ISO 13849-1:2008, Tables C.2 to C.7 also provides typical values for semiconductors and passive electrical components. Further databases are cited in the bibliography of EN ISO 13849-1.

## 3.3 Differences between SISTEMA users' and manufacturers' libraries

Both users and manufacturers can therefore create SIS-libs and distribute them to third parties. The SIS-libs in these two cases are technically identical. Differences may arise between the way the SIS-libs are created; for large-scale databases, manufacturers will prefer other methods to the integrated editor (see Chapter 12). These methods are in principle available to users, however. Manufacturers generally write-protect their SIS-libs in order to prevent inadvertent manipulation (see Section 11.6). Write-protection is also recommended when SIS-libs are distributed within larger working groups or to customers/suppliers.

---

[4] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems.

[5] SN 29500 (calculation of the failure rates of electronic equipment)

[6] IEC 62380: Reliability Prediction

Differences typically arise with regard to the data objects: manufacturers usually generate SIS-libs with the data for their components (Figure 2). For re-use in user libraries, complex parts of control systems are more likely to be saved (Figure 3): in SISTEMA, these are "subsystems", comprising multiple components. Manufacturers may however equally supply SIS-libs comprising application examples (blocks or subsystems) together with their products. And as already described in Section 3.2, users can still create their own libraries containing component parameters when a manufacturer's SIS-lib for them does not exist.

Figure 2: Examples of components (PLCs, position switches, relays)
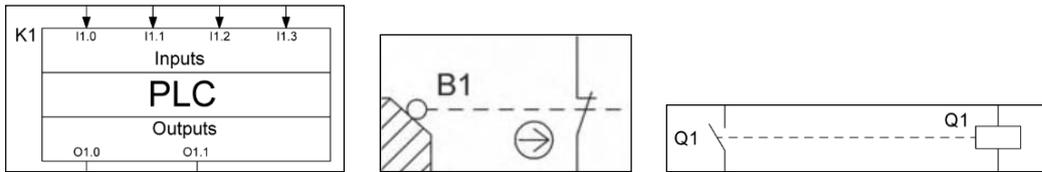


Figure 3: Example of a part of a control system comprising five components (taken from SISTEMA Cookbook 1)



⇑  Shown in actuated position
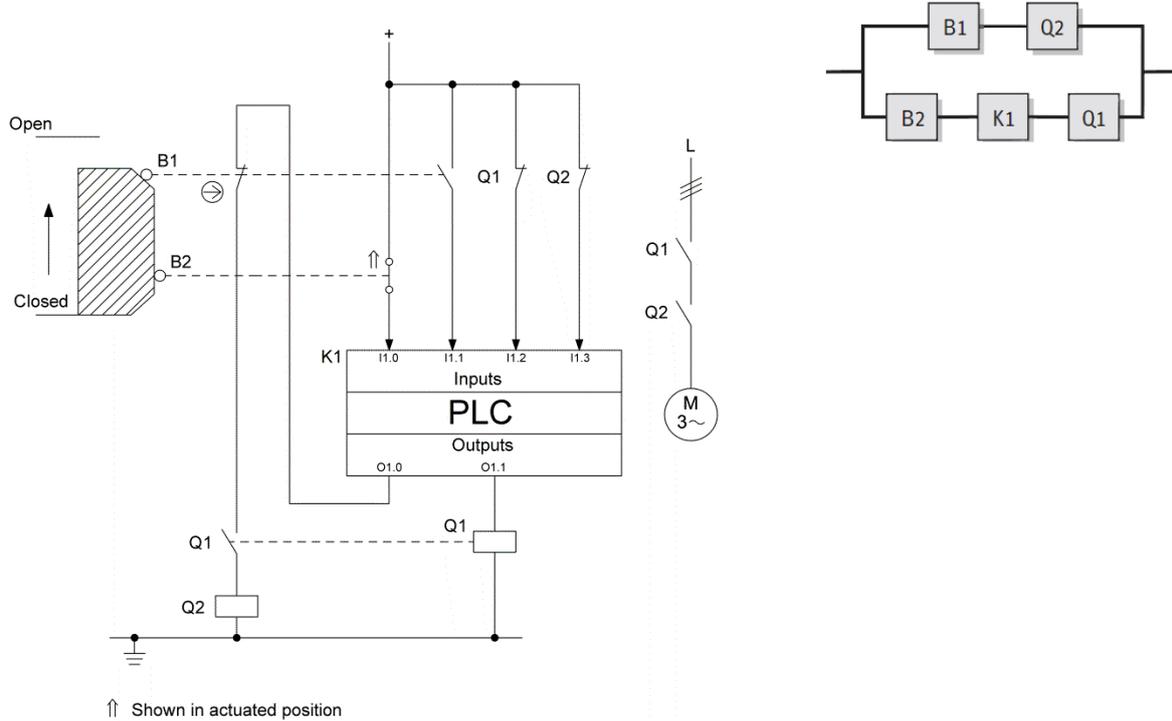
## 3.4   SISTEMA library database properties and structure

Three of the seven object types that can be managed by SISTEMA are possible for publication in the form of a library:

- The subsystem (SB) generally implements a "designated architecture" from the standard. For each Category, EN ISO 13849-1 recommends an architecture setting out requirements concerning the basic structure, the number of channels and the test equipment. SISTEMA enables the user to connect multiple subsystems in series in the context of a safety function.

- The block (BL) is on the one hand a function block in the context of a logical block diagram. It divides a channel into logical functional units (such as sensor, logic and main contactor). The block can however also constitute a single component.

- The element (EL) is the lowest hierarchical level of the objects. Elements are electronic, electromechanical, hydraulic, mechanical or pneumatic components of which a function block is composed.

The channel (CH)/test channel (TE) object is used only as a structuring component. Channels and test channels can therefore be managed in the library only as parts of a subsystem; they cannot be saved or loaded on their own.

Besides the channel and test channel, the following object types are also not supported by the library:

- The project (PR) represents the project file (*.ssm) itself, and generally refers to a machine or danger zone under analysis.

- The safety function (SF) is defined as a function of the machine under analysis failure of which can lead to a direct increase in the risk. It is implemented by one or more subsystems; it does not itself constitute a component however, but a functionality. It too cannot therefore be managed in the library. Safety functions can be archived as samples in (sample) projects, distributed and re-used.

Table 1 and Figure 4 show an overview of the objects stated, their rank in the hierarchy, and the facilities for their management in the library.

| Object type with abbreviation and symbol | Rank in the hierarchy | Management by means of the library |
|---|---|---|
| Project (PR) PR | 1 | No |
| Safety function (SF) SF | 2 | No |
| Subsystem (SB) SB | 3 | Yes |
| Channel (CH) CH | 4 | Indirect |
| Test channel (TE) TE | 4 | Indirect |
| Block (BL) BL | 5 | Yes |
| Element (EL) EL | 6 | Yes |

Table 1: Overview of SISTEMA object types

Figure 4: Hierarchical levels considered in SISTEMA
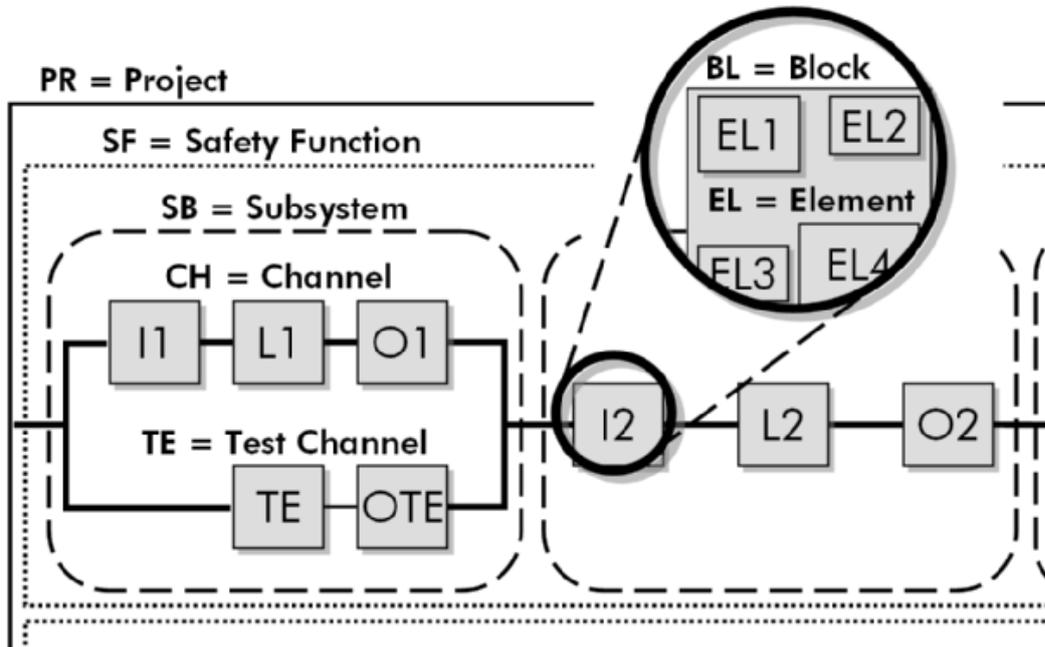


The help file of the software is installed together with SISTEMA and provides a comprehensive description of the objects. Further information is provided by the SISTEMA cookbooks.

SISTEMA offers manufacturers of components/parts of control systems the facility to create SISTEMA libraries of their own. The limits of the integrated library editor are however rapidly reached when the number of components is large. We therefore seek to support the component manufacturers by implementing further interfaces.

## 3.5   Interfaces to SISTEMA libraries

SISTEMA offers a number of interfaces for SIS-libs (upper part of Figure 5) and for project files which demonstrate the Performance Level of the safety functions.

The simplest interface to SIS-libs is provided by the integrated library editor (Figure 5, right-hand GUI). Once the user has become familiar with the user interface, values and information can be entered and verified directly in the objects of a SIS-lib without additional tools. The user interface and the various input methods are described in Chapters 4 and 11. For smaller libraries, this editor will be the tool of choice.

However, this interface does not allow content to be generated automatically from an existing body of data. The SQL interface to the database is more suitable in this case. SISTEMA provides an import interface for VDMA-libs, since this format is available only to product manufacturers.

Figure 5: Interfaces to SISTEMA
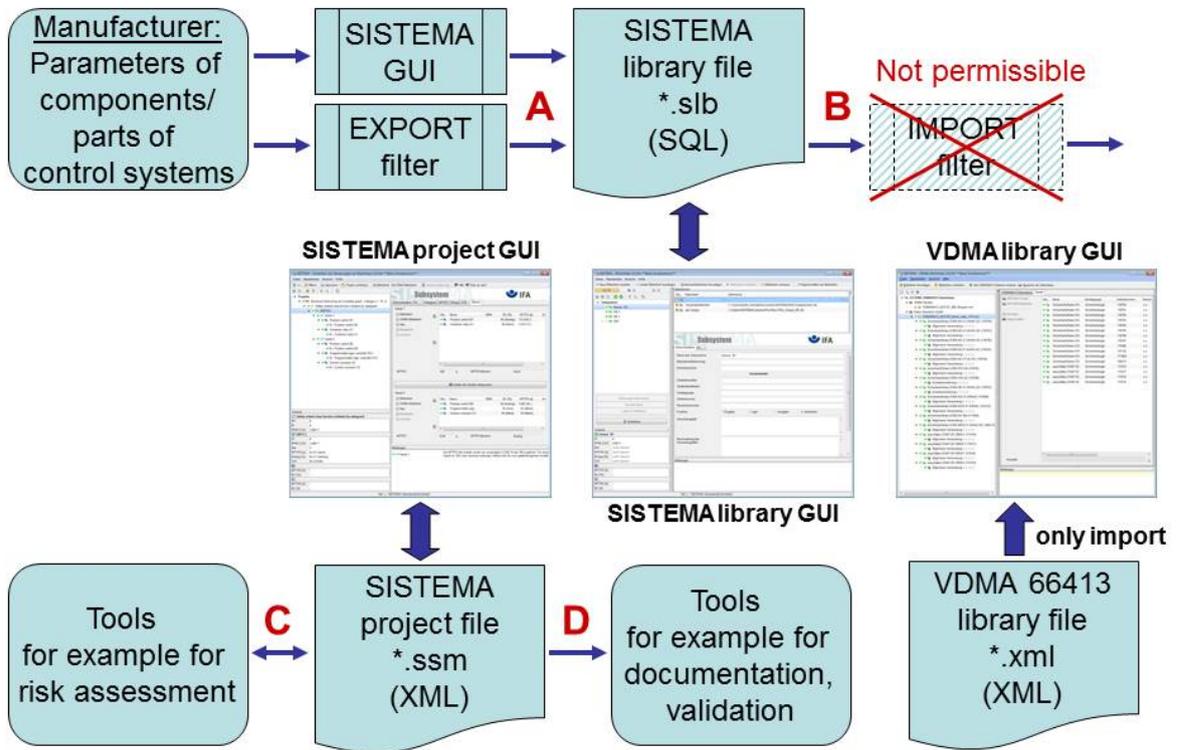


The Path B shown in Figure 5 – the development of an import filter for the conversion of SIS-libs – is not permissible.

# 4    The user interface for SISTEMA libraries

The simplest way of creating, editing and using SISTEMA libraries is by means of the library window integrated into SISTEMA's graphical user interface. Reference has already been made to the limits of the editor when library content is to be generated on a larger scale (see Section 3.5).

This chapter describes, in the first instance, calling up and closing of this window and the GUI structure of the SISTEMA library window. The detailed procedures for the various tasks involving libraries are described in Chapters 7 and 11.

## 4.1    Calling up and closing the SISTEMA library window

The SISTEMA library window is called up in the main SISTEMA window by means of:

- The "Edit -> Library" menu item or the Ctrl+B keyboard shortcut

- The "Library" button in the toolbar (Figure 6)

- The "Library" button in the various list views of the SISTEMA subsystems, channels, blocks or elements objects (for an example, see Figure 7 for blocks in Channel 1)


Figure 6: "Library" button in the toolbar of the main window
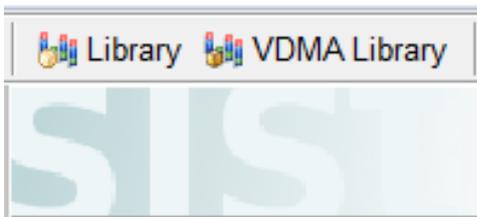


Figure 7: "Library" button in the blocks list view



**Important: a report cannot be generated in the main window whilst the SISTEMA library window is open, even if it has been minimized. The reverse also applies: if the report display is opened, the SISTEMA library window cannot be opened.**

The SISTEMA library window can be closed in a number of ways:

- By clicking on Windows' "Close" button of the window (top right-hand corner, white cross in a red field) or by means of the "Close Window" command in the Windows task bar

- By means of the "File -> Close" menu item or the Alt+F4 keyboard shortcut

- By means of the "Close" button in the centre-left of the library window (Figure 8)

Figure 8: "Close" button in the library window



## 4.2   Structure of the SISTEMA library window user interface

The user interface of the library window is similar in structure to the SISTEMA main window: the control space at the top edge of the window with its menu bars and toolbars can be used to control and configure the libraries. The centre of the window contains a workspace, a navigation pane with tree view (left), a message pane (bottom), and a context pane (bottom left). A help pane for context help, as in the SISTEMA main window, is not provided; the help file can however be opened separately ("Help -> Help" menu item). The control space and navigation pane differ from those in the SISTEMA main window.

The size of the areas, i.e. the position of the borders between them, can be adjusted by means of the left mouse button. The context and message panes can be hidden and unhidden via the menu or by keyboard shortcuts. The user interface arrangement of the complete library window is saved in the Windows registry when SISTEMA is closed, and is restored when SISTEMA is next launched. The factory settings can be restored by means of the SISTEMA configurator, which is launched separately (without SISTEMA running).

### 4.2.1   The control space

The control space provides the following information (Figure 9):

Figure 9: Control space of the library window
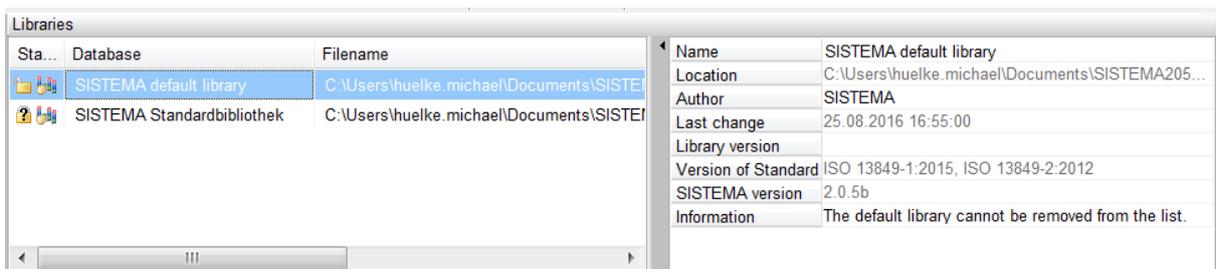


- "Libraries" list (left): contains all libraries previously opened. This list is stored in the Windows registry when SISTEMA is closed, and is read in again when SISTEMA is next launched. Selecting a list entry loads the library in question into RAM, thereby making its

objects accessible. Following closing of the library window, the last SIS-lib selected remains loaded in RAM, and is available again immediately when the window is opened again.

Each list entry displays the library name and the associated file path and file name. The write-protection status is indicated by a symbol (padlock) before the name:

- 🔒 Indicates write-protected libraries.

- 🔓 Indicates writeable libraries.

- ❓ Indicates that SISTEMA has not read out any information on the write-protection status of the library. Select the library in order to check its write-protection status.

- "Library properties" list (right): permits viewing and (in the case of non-write-protected libraries) configuration of the meta information for a loaded library. The following information is displayed: the name (to be determined by the user; this may differ from the file name and may also contain special characters), location, author (the Windows user name is the default), last change, library version (to be determined by the user), version of the standard, version of SISTEMA, and information on the library currently selected (to be determined by the user; see Section 11.10). This list can be unhidden and hidden by means of the left-pointing black triangle control element in the centre divider.

### 4.2.2   Navigation pane

The navigation pane (Figure 10) provides the following actions and control elements in five areas:

Figure 10: Navigation pane in the library window
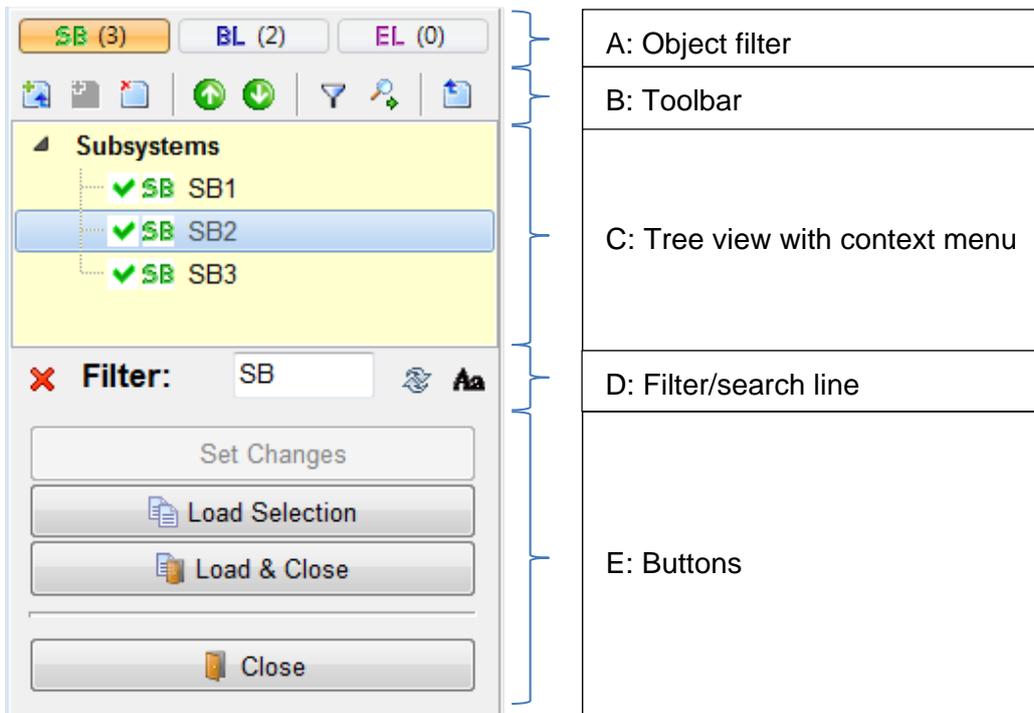
**Area A: Object filter**

Each library manages three different object types: subsystem, block and element. The object filter (Figure 11) at the top of the navigation pane is used to control which of the three types is to be displayed in the navigation pane. Click on one of the three buttons to select the object type. Only one type can be displayed at any one time. If no object of the selected type is saved in the selected library, the navigation pane remains empty. The number of objects present is shown in brackets after the object code SB, BL or EL.

Figure 11: Object filter for selection between the three object types
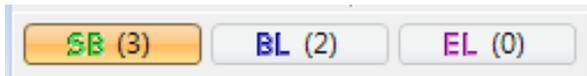


**Area B: Toolbar**

Square buttons can be found in the navigation pane between the object filter and the object tree. These buttons have the following actions, which are active depending upon the context:

-  "New subsystem/block/element":

-  "New": adds the selected object to a new, empty sub-object.

-  "Delete": deletes the selected object from the list.

-  "Move One Up": moves the selected object upwards one place in the list.

-  "Move One Down": moves the selected object downwards one place in the list.

-  "Show Filter Input": displays a line below the object tree showing control elements for filtering the object tree.

-  "Show Search Input": displays a line below the object tree showing control elements for searching in the object tree.

-  "One Up": moves the selected object up one level (in complex object structures)

**Area C: Tree view with context menu**

The objects in the tree view of the navigation pane are typically controlled by means of the mouse. The corresponding keyboard shortcuts are described in Section 4.3. Left-clicking with the mouse on one of the objects selects it for display/editing. Right-clicking on an object opens a context menu containing the following commands, which are active or not depending upon the context:

-  "Expand All": all objects below the selected object are displayed (the object tree expands).

-  "Collapse": the object tree below the selected object is minimized. The objects below the level concerned are no longer visible.

-  "New": inserts a new, empty object into the selected object list.

- "Load from VDMA library…": opens the VDMA library window. A VDMA 66413 use case can be loaded into the SIS-lib.

- "Delete": deletes the selected object from the list.

- "Load Selection": copies an object from the library into the project. The object is inserted as a lower-level object below the object currently selected in the project.

- "Load & Close": copies an object from the library into the project, and closes the library window.

- "Print Tree View…": a printer dialog is opened and the current view of the object tree (fully or partly maximized) is printed on the selected printer.

- "Cut": this command is not available in the navigation pane of the library window, in order to prevent objects from being deleted from a library inadvertently.

- "Copy": copies the selected object into the Windows clipboard.

- "Paste": inserts an object from the Windows clipboard, provided it is compatible as a lower-level object to the object currently selected.

- "Move One Up": moves the selected object upwards one place in the list.

- "Move One Down": moves the selected object downwards one place in the list.

- "Sort A-Z": sorts the list of objects in ascending alphanumeric order.

- "Sort Z-A": sorts the list of objects in descending alphanumeric order.

- "Filter": displays a line below the object tree showing control elements for filtering the object tree.

- "Search": displays a line below the object tree showing control elements for searching in the object tree.

**Area D: Filter/search line**

This area is visible only when the filter/search function is activated by the command referred to above. The line contains the following buttons:

- "Close": the line is hidden and the function terminated.

- "Filter/Search": a string that is to be filtered or searched for in the object names is entered in the field to the right. The function is launched with the Enter key. Following filtering, only objects with names containing the filter string are displayed. During searching, objects in the tree structure with names containing the search string are marked and can be selected in turn.

- "Automatically Update Filter": when the string in the input field is changed, the filter function is launched immediately, without the Enter key having to be pressed. The background of the button is dark when this option is active.

- **Aa** "Match Case": when this option is active, the case of the filter string is matched during filtering. The background of the button is dark.

- 🔍 "Show Next Match": during searching, the next object in the search tree after the current selection that contains the search string in its name is selected. The Cursor Down key executes the same command.

- 🔍 "Show Previous Match": during searching, the previous object in the search tree before the current selection that contains the search string in its name is selected. The Cursor Up key executes the same command.

**Area E: Buttons**

Large buttons are provided in the lower space for frequent actions:

- "Set Changes" button: saves the current changes in the library file. Active only if changes have been made.

- 📄 "Load Selection" button: copies the selected object from the library into the project. This button is available only if the following conditions are met:

   – In the Navigation pane, an object of the highest displayed hierarchical level (subsystem, block or element) has been marked with the mouse. This is the object to be loaded. In Figure 10 it is the selected subsystem "SB2".

   – In the SISTEMA main window, an object of the next hierarchical level up is marked. The object to be loaded can subsequently be copied into this object (a safety function for the selected "SB2").

   Loading creates a copy of the selected object and adds it to the selection in the main window. The library window remains open.

- 📋 "Load & Close" button: copies the selected object (as described in the previous action), then closes the library window.

- 🚪 "Close" button: closes the library window. A message draws attention to libraries that have not yet been saved.


Copying of objects between the object trees in the main window and the library window is also possible in both directions by drag and drop, i.e. by dragging an object with the left mouse button depressed.

A general rule is that objects may be inserted only at suitable points. The paste operation is not otherwise performed, or the "Paste" or "Load" commands are not active, but greyed out. The following constraints apply:

- An element can be inserted only into a block.

- A block can be inserted only into a channel/test channel.

- A subsystem can be inserted only into a safety function.

Where a library is write-protected, only the "Copy" operation is possible; "Paste", "Cut" and "Delete" operations are not possible.

### 4.2.3   The workspace

The workspace is located in the centre of the SISTEMA library window. The input dialogs for the object currently selected in the navigation pane are displayed here. The input dialogs for subsystems, channels, blocks and elements are the same as those in the SISTEMA main window.

### 4.2.4   The context pane

The context pane is displayed at the bottom left of the library window. It summarizes the parameters of the selected objects and also those of the higher-level objects. The context pane is the same as that in the SISTEMA main window, with the exception that safety functions cannot be displayed. The "View -> Context" menu item (Ctrl+K) hides and unhides this window.

### 4.2.5   The status of objects

Objects are marked with their current status before the object name in the various areas of the user interface. SISTEMA differentiates between three states:

*   ✔ A green tick indicates that no warnings, errors, inconsistencies or unsatisfied conditions are active for the object concerned. Information in the message pane should however be observed.

*   🟡 A yellow dot indicates that at least one warning is active for the object (see message pane).

*   ✖ A red cross indicates that the object exhibits at least one error, inconsistency, or unsatisfied condition (see message pane).

The status of an object is inherited by the object on the next level up in the hierarchy. The status with the highest priority takes precedence. The highest priority is the red status, the lowest the green status.

Should a library contain objects with red status, analysis is not possible or has not been completed successfully.

When objects have the yellow status, the user of the objects must observe the warning messages. The analysis has however been completed successfully.

### 4.2.6   The message pane

Like the main SISTEMA window, the message pane (Figure 12) contains information, warning and error messages concerning the selected object. The "View -> Message" menu item (Ctrl+M) hides and unhides this pane.

Figure 12: Message pane



Each message is marked by a symbol showing the status of the object (see Section 4.2.5). The message with the highest priority also determines the status of the object. The list in the message pane shows the messages for the selected object and for all objects on the levels below it. Messages relating directly to the selected object are highlighted by a yellow background. The messages for the lower-level objects have a white background. Double-click on a message to jump to the object concerned; this object then becomes the current selection. An object for which no messages exist also has green status, but is not shown in the message pane.

In the example (Figure 12), "Channel 1" contains a green message. "Channel 2" contains no message, but the objects "BL1" and "BL2" in Channel 2 have messages with yellow status. "Channel 2" thus has the overall status of yellow, which is shown in the navigation pane.

### 4.2.7   The toolbar

The toolbar (Figure 13) is located directly below the menu bar.

Figure 13: Toolbar of the library window



The toolbar provides rapid access to all SISTEMA library-related commands:

- "Create New Library": creates a new, empty SIS-lib and appends it to the list (see Section 11.1).

- "Add local Library": opens a file manager in which a SIS-lib present in the local file system can be inserted into the list (see Section 7.1).

- "Add Network Library": links a SIS-lib on the network server into the list. Further information on this function can be found in SISTEMA Cookbook 2 (see Section 7.2).

- "Close Library": removes the selected SIS-lib from the list. The SIS-lib is however not deleted (see Section 7.4).

- "Lock Library": saves a write-protected copy of the selected SIS-lib. Objects can be loaded from a write-protected SIS-lib; no modifications can however be made within the library (see Section 11.6).

- "Library properties": shows the properties of the selected SIS-lib in a separate window. Some properties can be edited (see Section 11.10).

### 4.2.8    The menu bar

The menu bar contains four menus: "File", "Edit", "View" and "Help". The majority of menu items have already been explained in the preceding sections.

The "View -> Context" and "View -> Message" menu items unhide and hide these windows. "Refresh" updates the complete object tree in the navigation pane.

The "Help -> Manufacturer Libraries" menu item opens the computer's standard browser and, if an Internet connection exists, calls up the IFA web page of hyperlinks to the manufacturers' libraries.

### 4.2.9    The status bar

Information on the content of the Windows clipboard is displayed on the left of the status bar at the bottom of the window (Figure 14). Should the clipboard contain objects, the associated symbol and object name are displayed.

A round symbol  may be displayed to the right of the "clipboard" display. If the mouse pointer is moved over this symbol, system messages and help information are displayed.

The SIS-lib currently selected is displayed in the middle of the status bar, as is the "padlock" indicating its write-protection status. Figure 14 shows an open padlock, indicating a library without write protection.

Figure 14: Status bar of the library window showing all elements
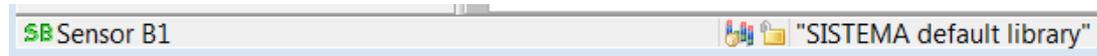


## 4.3    Keyboard shortcuts for the SISTEMA library window

SISTEMA functions can also be accessed by keyboard shortcuts. These appear adjacent to the items in the menu bar, and in the readme file. The readme file is opened in the workspace when the "Projects" folder (first line of the object tree) is clicked on in the navigation pane of the SISTEMA main window. General keyboard shortcuts for the main and library windows are shown in Table 2. Further keyboard shortcuts are available in the library window (Table 3).

| Shortcuts | Action | Shortcuts | Action | Shortcuts | Action |
|---|---|---|---|---|---|
| Ctrl+C | Copy | Ctrl+Ins | New (project or object) | Ctrl+B | Open library |
| Ctrl+Alt+C | Copy into library | Ctrl+Del | Delete (tree or object) | Ctrl+Alt+B | Load from selected library |
| Ctrl+V | Insert | Ctrl+N | New project | Ctrl+H | Unhide/hide message pane |
| Ctrl+X | Cut | Ctrl+L | Load project | Ctrl+K | Unhide/hide context pane |
| Ctrl+Cursor up | Move object upwards | Ctrl+S | Save project | Ctrl+O | Display option window |
| Ctrl+Cursor down | Move object downwards | Ctrl+Alt+S | Save project as ... | Ctrl+R | Show summary |
| Ctrl+Home | One level upwards | Ctrl+P | Print project | CTRL+I | Information on SISTEMA |

| Page up | One page up | Ctrl+E | Export as PDF | F1 | Help |
|---|---|---|---|---|---|
| Page down | One page down | Ctrl+F4 | Close project | F3 | "What's this?" help |
| Home | Jump to the beginning of the tree/list | Ctrl+Alt+F4 | Close all projects | Ctrl+F3 | Context help |
| End | Jump to the end of the tree/list | F2 | Edit the selected object name | Alt+F3 | Open wizard |
| | | | | Alt+F4 | Quit SISTEMA |
| | | | | F5 | Refresh view/tree |

Table 2: General keyboard shortcuts

| Keyboard shortcuts | Action |
|---|---|
| Ctrl+B | Add local library |
| Alt+B | Add network library |
| Ctrl+Alt+B | Create new library |
| Ctrl+F4 | Close library |
| Ctrl+N | Create new object in the tree |
| Ctrl+S | Set changes to the library |
| Ctrl+P | Lock library |
| Ctrl+I | Library information |
| Alt+F4 | Close library window |

Table 3: Additional keyboard shortcuts in the SISTEMA library window

# 5    User interface for VDMA libraries

This chapter describes calling up and closing of the library window for files in the VDMA Specification 66413 format ("VDMA libraries", see Annex B) and the graphical structure of the library window. The detailed operating processes for the various tasks performed with VDMA libraries are described in Chapter 8.

## 5.1    Calling up and closing the VDMA library window

The VDMA library window is called up in the SISTEMA main window by means of:

- The "Edit -> VDMA Library" menu item

- The "VDMA Library" button in the toolbar (Figure 15)

- The "VDMA Library" button in the various list views of the SISTEMA subsystem, channel, block or element objects (for an example, see Figure 16 for blocks in Channel 1)

Figure 15: "VDMA Library" button in the toolbar of the main window
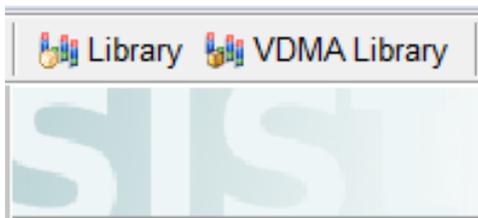


Figure 16: "VDMA Library" button in the blocks list view
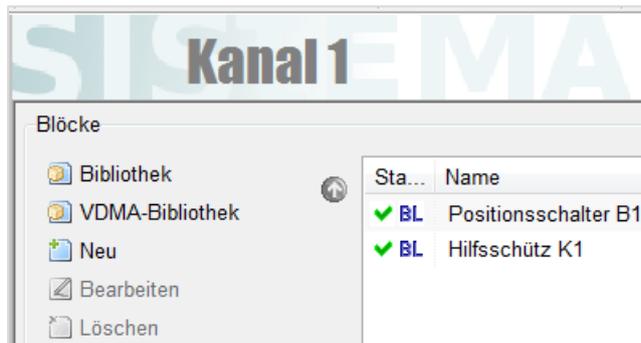


The VDMA library window can be closed in a number of ways:

- By clicking on the Windows' "Close" button of the window (top right-hand corner, white cross on a red field)

- By means of the "Close Window" command in the Windows task bar

- By means of the "File -> Close" menu item or the Alt+F4 keyboard shortcut

## 5.2    Structure of the VDMA library window

The VDMA library window is similar in structure to the SISTEMA library window: the menu and toolbar at the top of the window are used to control and configure the VDMA libraries.

The window also features a workspace in the middle, a navigation pane with tree view on the left, and a message pane at the bottom. The window has no context window, status bar or help window. The control space, workspace and navigation pane differ from those of the SISTEMA library window.

The size of the areas, i.e. the position of the dividers between them, can be adjusted by means of the left mouse button. The message pane can be hidden and unhidden by means of the menu or keyboard shortcuts. The position and layout of the entire library window are saved in the Windows registry when SISTEMA is closed, and recreated when it is next launched. The factory settings can be restored by means of the SISTEMA configurator, which is launched separately (without SISTEMA running).

### 5.2.1   Navigation pane

The navigation pane (Figure 17) provides the following actions and control elements in three areas:

Figure 17: Navigation pane in the VDMA library window
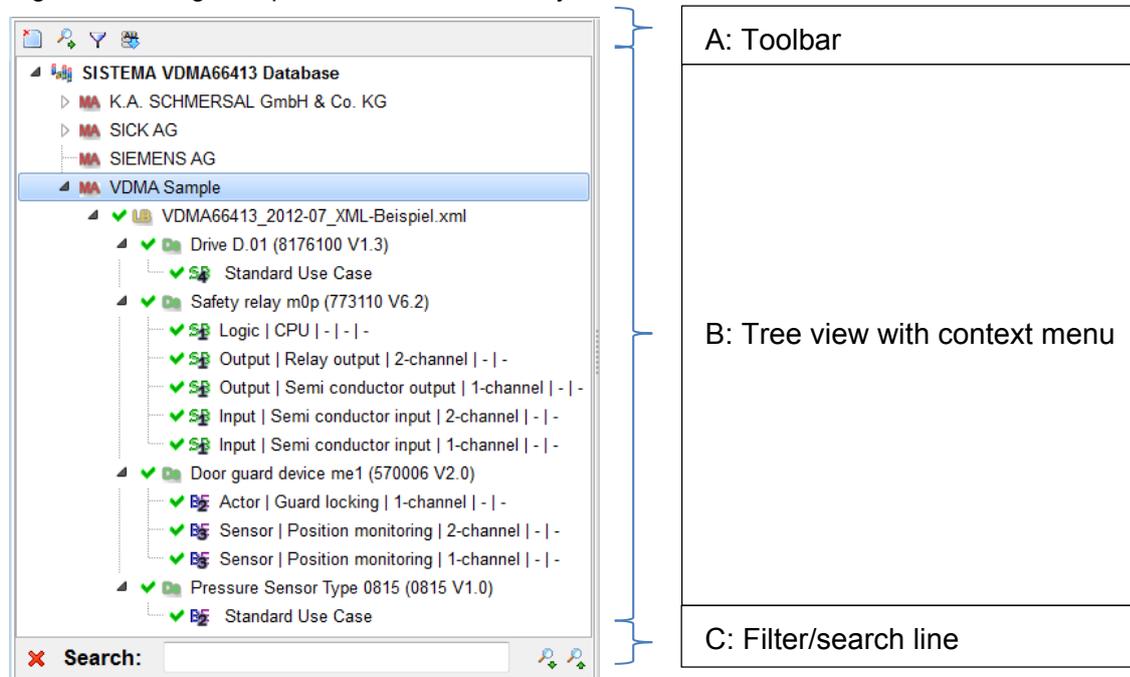


**Area A: Toolbar**

The navigation pane contains square buttons above the object tree. These are used to perform the following actions, which are active depending upon the context:

-  "Delete": deletes the selected manufacturer and all lower-level libraries/only the selected library from the list. Devices and use cases cannot be deleted.

-  "Show Search Input": displays a line below the object tree showing control elements for searching in the object tree.

-  "Show Filter Input": displays a line below the object tree showing control elements for filtering the object tree.

- ![icon] "Load data from all VDMA 66413 files": All libraries loaded in the object tree are loaded from their files. When SISTEMA is launched and the VDMA library window opened, all previously loaded manufacturers and libraries are displayed, but their content is loaded only upon request (individually, or with this command completely).

**Area B: Tree view with context menu**

Unlike in the SISTEMA library window, the list of manufacturers and their loaded VDMA libraries are displayed directly in the navigation pane in the form of a hierarchical structure. The hierarchy of the objects comprises the following object types (Figure 18):

Figure 18: Object types in the VDMA library window



- ![icon] "SISTEMA VDMA 66413 Database": this is the top-level object in the hierarchy, the root of the object tree. This object contains all manufacturers added in the navigation pane.

- ![icon] "Manufacturer": object type for manufacturers and suppliers of VDMA libraries. The manufacturer name is unique. The object contains all libraries from the manufacturer in question that have been added in the navigation pane.

- ![icon] "Library": object type for VDMA libraries. This shows the file name of the library. This object is also termed a "database". It contains all devices from the manufacturer in question listed in this library.

- ![icon] "Device": object type for devices in a VDMA library. It shows the device names. This object contains one or more use cases for this device (Figure 19). If devices are marked by the manufacturer as "archive data", their names are shaded in brown (Figure 20). These devices may no longer be available and should for example no longer be used for new projects. These archive data can be unhidden and hidden in the tree by means of an option in the "Edit -> Options -> VDMA Library" menu item.

Figure 19: One device with three use cases
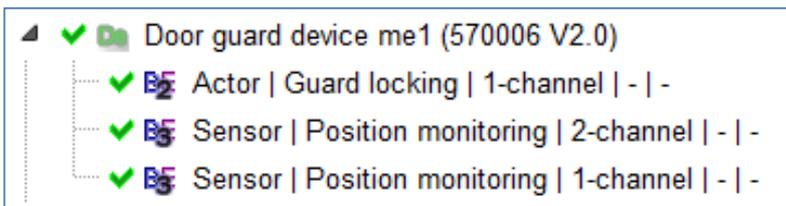
Figure 20: Two versions of one device (archive data with shaded in brown)
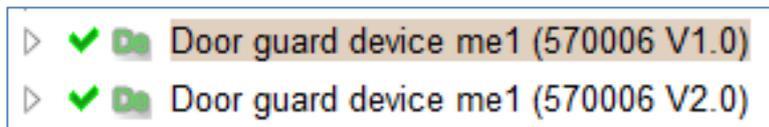


A use case defines the corresponding set of reliability values for the described application of the device. Each use case corresponds to one of the four device types defined in VDMA Specification 66413. These can be copied into a suitable SISTEMA object. Accordingly, a SISTEMA symbol with a digit for the device type appears in front of each use case of a device:

- Device types 1 and 4 correspond to the SISTEMA subsystem.

- Device type 2 can be used as a SISTEMA block or element for components not subject to wear.

- Device type 3 can be used as a SISTEMA block or element for components subject to wear.

The objects in the tree view of the navigation pane are typically controlled by means of the mouse. Left-clicking with the mouse on one of the objects selects it for display/editing. Right-clicking on an object opens a context menu containing the following commands, which are active or not depending upon the context:

- "Expand All": all objects below the selected object are displayed (the object tree expands fully).

- "Collapse All": the object tree below the selected object is minimized. The objects below the selected object are no longer visible.

- "Expand": only the objects below the selected object are displayed (the object tree expands partly).

- "Collapse": only the object tree below the selected object is minimized. The objects below the selected object are no longer visible.

- "Copy": copies the selected object into the Windows clipboard.

- "Remove manufacturer": removes the selected manufacturer and all lower-level libraries from the object tree.

- "Close library": closes the selected library and deletes the library object from the object tree.

- "SISTEMA Project": copies the selected use case to the previously selected location in the SISTEMA project. The command is effective only when the target of the copy operation has been properly selected.

- "SISTEMA Library": copies the selected use case into the previously selected SISTEMA library. Device types 1 and 4 are copied into the list of subsystems, device types 2 and 3 into the list of blocks.

- "Read Library:" if one or more libraries from a manufacturer have not yet been loaded after SISTEMA is launched, the action can be triggered by means of this command, specific to the manufacturer in question.

- "Reload Library": the data from the libraries of one manufacturer or all manufacturers are reloaded, depending upon the selected point on the object tree.

- "Sort A-Z": sorts the list of objects in ascending alphanumeric order.

- "Sort Z-A": sorts the list of objects in descending alphanumeric order.

- "Search": displays a line below the object tree showing control elements for searching in the object tree.

- "Filter": displays a line below the object tree showing control elements for filtering the object tree.

- "Properties": object-specific properties are displayed in a separate pane.


**Area C: Filter/search line**

This area is visible only when the filter/search function is activated by the command referred to above. The line contains the following buttons:

- "Close": the line is hidden and the function terminated.

- **Filter:** "Filter/Search": a string that is to be filtered or searched for in the object names is entered in the field to the right. The function is launched with the Enter key. Following filtering, only objects with names containing the filter string are displayed. During searching, objects in the tree structure with names containing the search string are marked and can be selected in turn.

  Filtering is possible only within a manufacturer. All other libraries are not analysed (their names remain in italics and shaded in yellow for the duration of the operation).

- "Automatically Update Filter": when the string in the input field is changed, the filter function is launched immediately, without the Enter key having to be pressed. The background of the button is then dark.

- **Aa** "Match Case": when this option is active, the case of the filter string is matched during filtering. The background of the button is dark.

- "Show Next Match:" during searching, the next object in the search tree after the current selection that contains the search string in its name is selected. The Cursor Down key executes the same command.

- "Show Previous Match:" during searching, the previous object in the search tree before the current selection that contains the search string in its name is selected. The Cursor Up key executes the same command.

### 5.2.2   The workspace

The workspace is located in the centre of the library window. The display dialogs for the object currently selected in the navigation pane appear here. Since the VDMA libraries are read-only, no inputs are possible. Any entries required ($n_{op}$, RDF, etc.) can be input only once a use case has been copied into the project. The text, graphics and values shown are explained in detail in VDMA Specification 66413 (see Annex B). When the use cases have been copied, terminology may change. For example, "TMT_1" becomes "Mission time" in SISTEMA.

Two tabs are shown in the workspace: on the left, details of the selected object, and on the right, the list of lower-level objects in multiple columns. Commands are available for the selected list objects as described above (see Figure 21 for an example).

Figure 21: List of use cases



### 5.2.3   Status of objects

Objects are marked with their current status before the object name in the various areas of the VDMA library window. SISTEMA recognizes three status settings:

- A green tick indicates that no warnings, errors, inconsistencies or unsatisfied conditions are active for the object concerned. Observe any information in the message pane, however.

- A yellow dot indicates that at least one warning is active for the object (see message pane).

- A red cross indicates that the object exhibits at least one error, inconsistency, or unsatisfied condition (see message pane).

The status of an object is inherited by the object on the next level up in the hierarchy. The status with the highest priority always takes precedence. The highest priority in this case is red status, the lowest green status.

Should a library contain objects with red status, the objects cannot be used.

When objects with yellow status are used, it is imperative for the user to observe the warning messages for the objects. The analysis has however then been completed successfully.

### 5.2.4   The message pane

Like the main SISTEMA window, the message pane displays information, warning and error messages concerning the selected object. The "View -> Messages" menu item hides and unhides this pane.

### 5.2.5   The toolbar

The toolbar (Figure 22) is located directly below the menu bar.

Figure 22: Toolbar of the VDMA library window



The toolbar provides rapid access to all commands relating to VDMA libraries:

- "Add Library": opens a file manager in which a library present in a local or remote file system can be inserted into the list (see Section 8.1).

- "Close Library": removes the selected library from the list. The library is however not deleted (see Section 8.2).

- "Read all Libraries": all libraries loaded in the navigation pane are reloaded.

- "Library Language": opens a dialog in which a VDMA 66413 language file can be loaded or closed (see Section 5.4).

### 5.2.6   The menu bar

The menu bar contains four menus: "File", "Edit", "View" and "Help". The majority of menu items have already been explained in the preceding sections.

The "Help -> Manufacturers' Libraries" menu item opens the computer's standard browser and, provided an Internet connection exists, calls up the IFA web page of hyperlinks to the manufacturers' libraries.

## 5.3   Keyboard shortcuts for the VDMA library window

General keyboard shortcuts for the main and library windows are shown in Table 2. Further keyboard shortcuts are available in the VDMA library window (Table 4).

| Shortcuts | Action |
|---|---|
| Ctrl+L | Add library |
| Ctrl+Del | Close library/remove manufacturer |
| Ctrl+Alt+F5 | Read all Libraries |
| Ctrl+F5 | Read Library |
| Shift+F5 | Reload Library |
| Ctrl+T | Library Language |
| Ctrl+S | SISTEMA Project |
| Ctrl+Alt+S | SISTEMA Library |
| Ctrl+C | Copy |
| Ctrl+U | Search |
| Ctrl+F | Filter |
| Ctrl+H | Messages |
| Alt+F4 | Close VDMA library window |

Table 4: Additional keyboard shortcuts in the VDMA library window

## 5.4    Languages of the VDMA libraries

The XML files of VDMA libraries may already contain several language versions of the text. The languages can be selected in the workspace of the library (Figure 23). All text is then displayed in the selected language and can be copied into SISTEMA.

Figure 23: Selection of a language in the dialog for VDMA libraries



In addition to the languages supplied with the library, a manufacturer may make further language versions available subsequently for a certain VDMA library in the form of a "language file". These language versions can be added by means of the "Library Language" command on the toolbar (see Section 5.2.5) for the library currently selected (Figure 24). SISTEMA then checks whether the language file matches the library; if not, an error message appears in the message pane.

Figure 24: Window for management of the supplementary VDMA language files
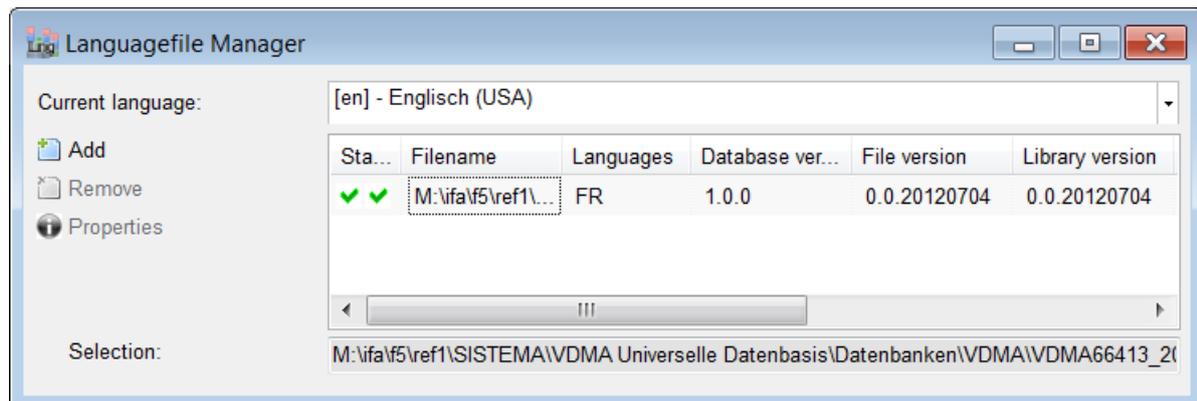


## 5.5    Options for VDMA libraries

The "Edit -> Options -> VDMA Library" menu item in the SISTEMA main window provides a number of options for control of the VDMA library window (Figure 25).

The checks of the VDMA 66413 files should always be activated. Manufacturers may disable these checks during testing of new libraries.

Figure 25: Options for VDMA libraries
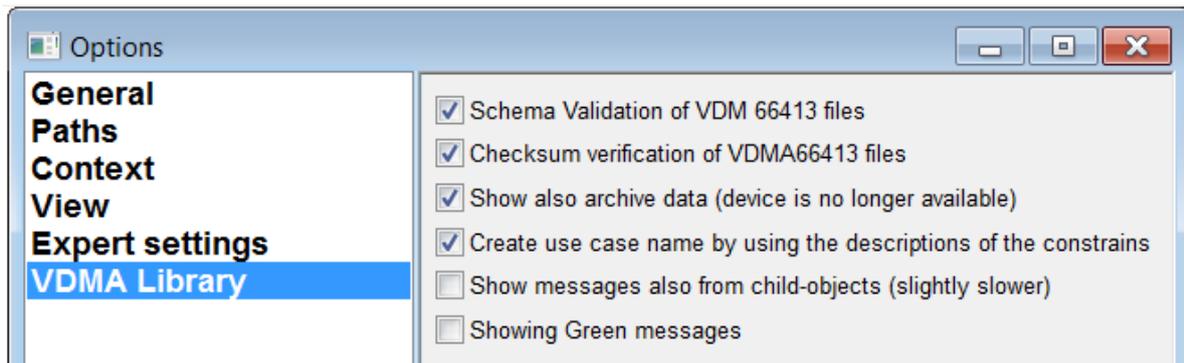
# 6    Obtaining SISTEMA libraries

This chapter describes how searches can be performed for manufacturers' SISTEMA librar-ies and how they can be obtained for use. The IFA does not host SIS-libs on its own servers; they must be obtained directly from the manufacturers. The IFA is not able to answer ques-tions concerning the content of manufacturers' libraries, whether they will be released by manufacturers, and if so when. Please direct such enquiries to the manufacturer or his sales agents.

The "SISTEMA standard library" shipped with SISTEMA does not contain any objects. It is intended solely to enable SISTEMA users to test the library functions, for example by creat-ing their own objects, etc. For programming reasons, this standard library cannot be removed from the list of libraries.

## 6.1    Information page at the IFA

The IFA's website contains two information pages with links to numerous manufacturers' SISTEMA libraries (Figure 26):

http://www.dguv.de/webcode/e92603 (English)

http://www.dguv.de/webcode/d92599 (German)

These pages list, in alphabetical order, the names of all companies providing SISTEMA libraries **that have been reported to the IFA**, together with links to the download areas of their SIS-libs. The procedure for downloading libraries or where applicable for registering for their use is not defined by the IFA, and varies from manufacturer to manufacturer. Not all manufacturers who make SIS-libs available are known to the IFA and linked to from the IFA website.

Figure 26: IFA web page providing information on manufacturers' libraries (as at 19 July 2016)

The web pages can also be accessed directly from the help menu in the SISTEMA library window (Figure 27) by means of the "Manufacturer Libraries" command. If the PC is online, its default browser is launched and the page loaded.

Figure 27: IFA web page of manufacturers' libraries shown in the browser
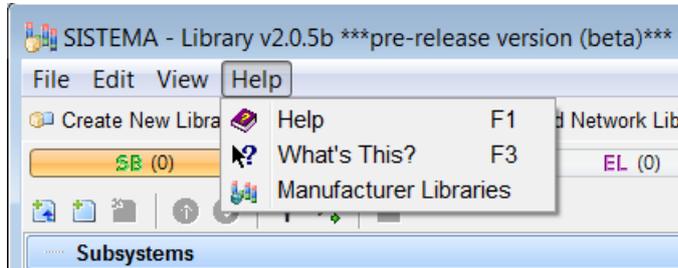


## 6.2    Downloading from manufacturers' websites and saving the SISTEMA libraries

The manufacturers' SIS-libs are generally hosted together with the product documentation in the form of an archive file, and can be downloaded to a local computer for use. Depending upon the volume of the data, a manufacturer may offer one or more SIS-libs. SIS-libs can generally be downloaded free of charge. The IFA charges no licence fees, either to manufacturers or to users, for use of the SIS-lib technology.

The standard folder on the local system for the saving and opening of SIS-libs is (example: Windows 7):

C:\User\<user name>\My Documents\SISTEMA\Libraries

The SIS-libs can however be saved in any desired location in the local file system of the SISTEMA user's computer, and opened again from there. A condition is that the SIS-lib can be opened by SISTEMA in a local folder (internal or external hard drive, plug-in memory media). The "Add local Library" command is used for this purpose (see Section 7.1). An attempt to open a library that is not saved locally, such as a library on a network drive, triggers an error message (Figure 28). The SIS-lib must be copied to the local computer if necessary.

Figure 28: Error message output when a library is opened that is not saved locally
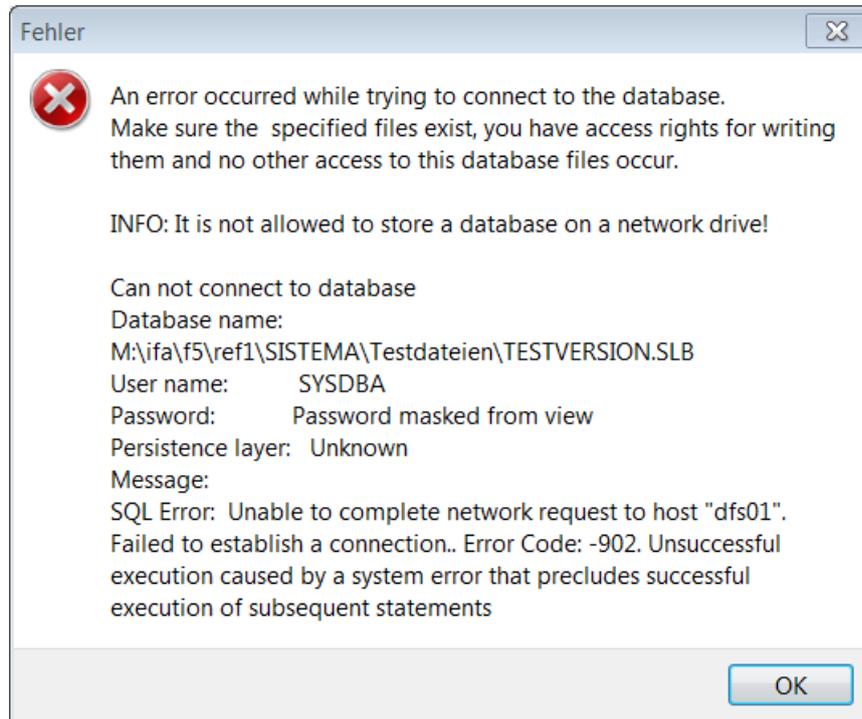


If the SIS-lib is not merely a file but is stored in an archive file together with other documents, the archive file must be unpacked with retention of the internal folder structure. The documents for the individual objects can then generally be opened directly from within the loaded SIS-lib ("Open" button to the right of the "Document" input field).

## 6.3    Differences between local and network libraries

SISTEMA was originally programmed exclusively as a client application for use by a single user on a PC. Up to and including Version 1.1.2, libraries had to be present on the local hard drive, since the simple "embedded" version of the "Firebird" open database technology was used for accessing these files. This "local" variant of a Firebird server assumes exclusive access to the file. This is not assured in the case of a network drive.

The Firebird server has been supported since SISTEMA Version 1.1.3. With this variant, the libraries are managed by a "true" database server that enables manufacturers' databases to be hosted centrally where they can be used by several people simultaneously on a network. A guide to this arrangement can be found in SISTEMA Cookbook 2. The internal database files for the SISTEMA installation can also be stored on this database server, besides the libraries. Transfer of the exclusive access entitlement to the Firebird server enables multiple instances to be launched in parallel on a computer, as a result of which terminal server-client hosting for example can be implemented. The IFA has also published SISTEMA Cookbook 3, which describes this mode and the necessary configuration.

## 6.4    Transferring SISTEMA libraries to a network server

Provided a database server has been configured as described in SISTEMA Cookbook 2 and access rights are granted, SIS-libs can be hosted centrally on a server. For this purpose, one person should receive access rights to the Firebird server, to which he or she should copy or move the desired SIS-libs. It is important that this person knows the local path for this access

entitlement. They also require the IP addresses or network name of the server. The access entitlement has the sole function of copying, moving and information exchange. SISTEMA does not require this entitlement, since access to the SIS-lib is provided via the Firebird server service.

These SIS-libs can now be accessed for use by means of the information provided (IP/host name and local path of the SIS-libs, e.g. *192.168.1.10:C:\SSM-databases\manufacturerXYa-SSM-DB.slb)* through the "Add Network Library" menu item.

SISTEMA supports management of these network libraries in a plain-text file; the complete address need therefore be entered only there. Any individual with access to the plain-text file can in turn read the address and then load the new SIS-lib directly from the database server (see Section 7.2).

# 7    Working with SISTEMA libraries

This chapter describes how SIS-libs already stored locally or on the network server can be opened and the objects copied from the SIS-libs into a project.

## 7.1    Opening local libraries

The command for loading SIS-libs stored on local data media into the library window is "Add local Library". This command opens a file manager (Figure 29) in which the folder and the SIS-lib can be selected and added to the list of libraries with the "Open" button. This results in the SIS-lib being loaded into RAM. The objects contained in the SIS-lib are displayed in the navigation pane, according to the selection made in the object filter (Figure 11).

Figure 29: File manager for the loading of SISTEMA libraries stored locally
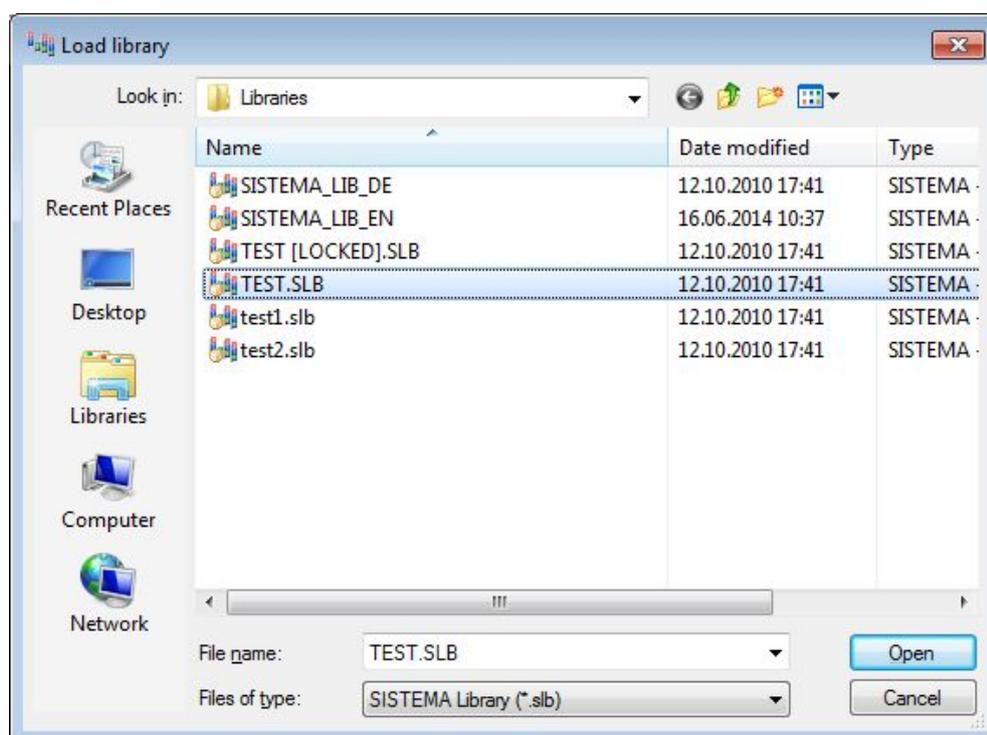


## 7.2    Opening network libraries

Network libraries have also been supported since SISTEMA Version 1.1.3. The required database parameters (IP-address or name of the database server and file name of the library) are made known by the responsible administrator in your company, generally in the form of a list in a plain-text file (see SISTEMA Cookbook 2, Section 1.4). Once the server PC hosting the libraries and the lists has been configured and a network connection is available to the server PC, the libraries can be opened. A description follows of the two options, i.e. entering a single library or a list of several libraries.

The SISTEMA library window contains the "Add Network Library" command. This command calls up a dialog (Figure 30) in which you can enter the library parameters, check the status of the library, and add it to the list of libraries. It can then be used in the same way as any other library. Please note the further information in Chapter 3 of SISTEMA Cookbook 2.

### 7.2.1   Reporting a single library

Enter the parameters for a single library on the server in the "Server name/IP-address" and "Database file name" fields under "Network database parameter" at the top of the "Add SISTEMA Network library" dialog (Figure 30). The complete database name required is generated automatically as the "Identified database name". After entering the parameters, use the "Check connection status of the library" button to test whether a connection to the desired library can be established and the library therefore actually used.

Only once the test of the "Identified database name" has been passed is the "Add" button enabled. Clicking on the "Add" button causes this entry to be added to the list of libraries and the library to be loaded.

Figure 30: Dialog for loading of network libraries



### 7.2.2   Reporting a list of libraries

A function also exists by which an entry can be made automatically. This is possible in the middle of the dialog under "Import database parameters (optional)", which enables you to import a list of library file names (Figure 30). The list required for this purpose should be supplied to you in the form of a plain-text file by the administrator of the Firebird server. Two locations are essentially possible for this plain-text file: either locally on the PC, or on a central drive accessible to all network users (refer to SISTEMA Cookbook 2, Section 1.4.1).

There are no constraints upon the file name of this plain-text file or the location in which it is stored.

Click on the button with three dots to the right of the "File name of the database list" field in order for this plain-text file containing the list of library file names to be read in via the file manager.

Once the file has been read in, an entry (i.e. a library) can be selected ("Selected database entry"). The database parameters of this library then appear in the associated "Server name/IP-address" and "Database file name" fields at the top of the dialog. The entries in the plain-text file can be listed according to three sort criteria in the "Sort By" area: 1) list order, 2) server name/IP address, or 3) database file name.

The availability of the library must now be checked (click on "Check connection status of the library"). If the library is available, clicking on the "Add" button causes it to be added to the list of SISTEMA libraries. Libraries from this list can be added only one at a time. The procedure must be repeated when several libraries are added from the list in the text file (select entry, check connection status, add). The database list need not be selected again each time. The window can be closed by means of the "Cancel" button.

## 7.3    Progress bar for the loading of libraries

Use the SISTEMA options ("Edit -> Options -> View -> Show progress bar" menu item) to activate a progress bar (Figure 31) which shows the progress of loading a library into RAM when it is opened. Display of the progress bar extends the duration of loading slightly.

Figure 31: Progress bar for the loading of libraries
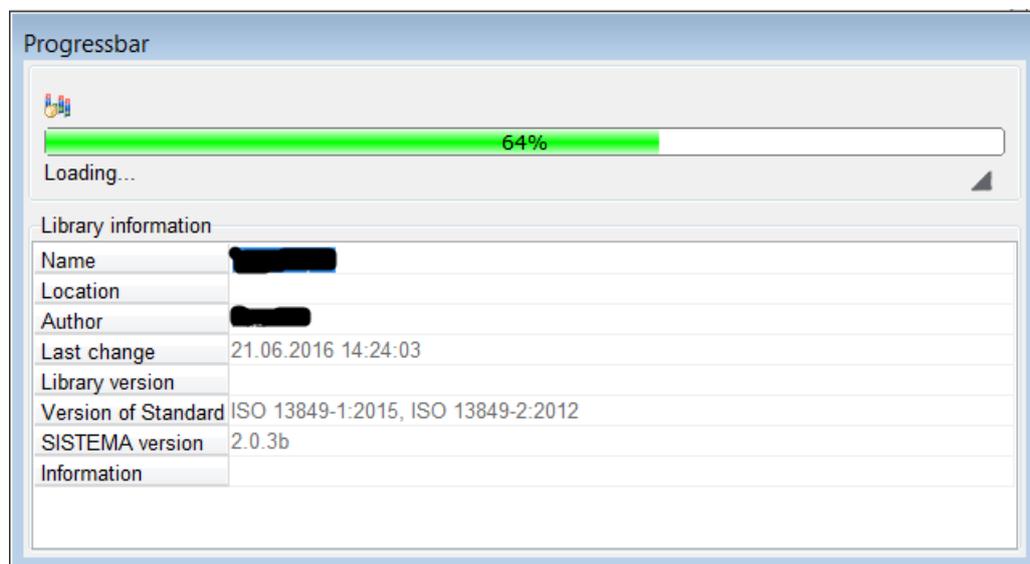


## 7.4    Closing libraries

The name and file name of each library that has been loaded by means of the functions referred to above are saved in the list of libraries. The list is stored in the registry each time SISTEMA is closed and loaded again when SISTEMA is next launched. Use the "Close Library" command ("File" menu, button, context menu or Ctrl+F4) to remove the selected library name from the list. This enables the list to be kept concise and clear. Removing library names from the list does not delete the library files.

## 7.5  Compatibility of libraries

The library properties also show the version of SISTEMA with which a library was created. It may be possible for the library to be opened by a different version of SISTEMA. Four possible scenarios exist:

* The library was created by an older version of the SISTEMA program, but the database definition is the same (the leading and middle version numbers are identical). No message is output. The library is loaded, since the data are compatible.

* The library was created by a more recent version of the SISTEMA program (e.g. 1.1.6) than the user's version (e.g. 1.1.5), but the database definition is the same (the leading and middle version numbers are identical). A message is output that the current SISTEMA version should be installed (Figure 32). The library is nevertheless loaded, since the data are compatible.

Figure 32: Version difference message: the library is more recent than the program
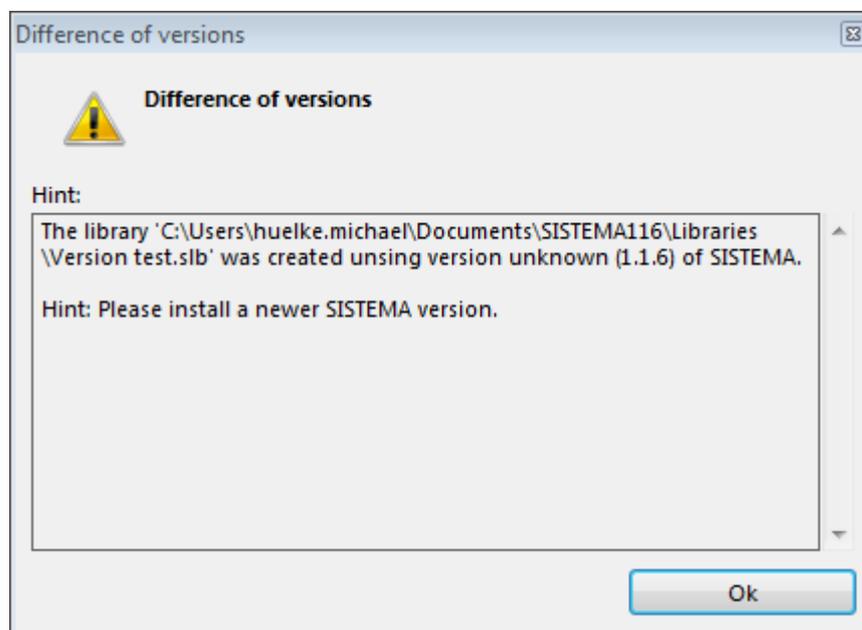


* The library was created by a **substantially** more recent version of SISTEMA than the user's version and the database definitions are not compatible (the leading or middle version numbers differ). A message is output that the current SISTEMA version must be installed. The library is not loaded.

* The library was created by a substantially older version (e.g. 1.0.4) of SISTEMA than the user's version (e.g. 1.1.5); the database definitions are not compatible (the leading or middle version numbers differ). In SISTEMA Versions 1.1.x, a message is output that an up-to-date library must be obtained (Figure 33). The library is not loaded. In SISTEMA Versions 2.x.x, attention is drawn to the version difference; the libraries are nevertheless loaded and the database format converted.

Figure 33: Version difference message: the library is not compatible with the program
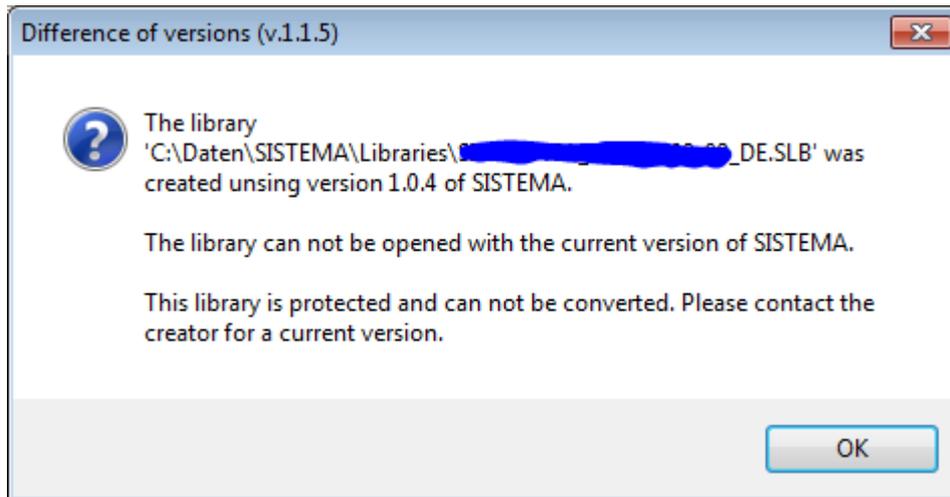


Should the database formats differ between the SISTEMA version used and the library to be loaded (thereby necessitating conversion), write-protected SIS-libs are not opened (Figure 33). The write protection is respected. The person responsible must first convert the original library, and then save it again with write protection. Further information on version control and compatibility can be found in Section 11.8.

## 7.6  Paths for linked documents

Objects in a library are often linked in addition to documents in a range of formats (PDF, XLS, DOC, JPEG, etc.). The "Document" input field in the various tabs is used to link these documents. Provided the SIS-lib and the linked documents are stored in an appropriate folder structure, the documents can be opened directly from within the loaded SIS-lib (for example: "Documentation" tab of the objects, "Open" button to the right of the "Document" input fields).

A frequently asked question when objects are inserted into a project is: how are the attached documents handled, and can they be copied into the project or stored elsewhere?

At present (Version 2.0.x), these documents are not automatically copied into a project along with an object. SISTEMA users must decide for themselves and take action if necessary. The form of linking is crucial. A URL (i.e. a web address, e.g. "http://www.sample_company.com/Document.pdf") may be available, or a relative path to a project folder (e.g. "..\sample_company\Document.pdf"). Absolute paths such as "L:\sample_company\Document.pdf" are unlikely to occur in manufacturers' libraries. These cases can be handled as follows:

- URL: either leave unchanged (the URL continues to be valid), or back up the document by loading it from the URL and copying it into the project folder. For the sake of completeness, then assign this document to the project by means of the "Document" input field. This ensures that the document can still be opened in the future even should the URL cease to be valid. The URL could in this case then be copied, for example into the "Documentation" field.

- Relative path: copy the manufacturers' documents into the project folder ("..\Documents\<relative path>") and check that the document can be opened correctly

from within the object in the project. Further information on relative paths can be found in the SISTEMA help under the "Options" menu item.

- Absolute path: copy the manufacturers' documents (e.g. "<document file name>") into the project folder, e.g. to "..\Documents\<manufacturer name>\<document file name>". Then modify the absolute address in the input field to the relative address, in this case to "..\<manufacturer's name>\<document file name>". Finally, check that the document can be opened correctly from within the object in the project.

## 7.7   Selecting an object type of a library

After opening a library and loading it into RAM, select in the object filter at the top of the navigation pane which of the three object types (subsystem, block or element) is to be displayed in the pane. Click on the object filter button for this purpose. Only one type can be displayed at any one time. The navigation pane remains empty if the selected library contains no objects of the selected type. In this case, change the type in the object filter. Further information can be found in Section 4.2.2.

## 7.8   Copying objects from libraries

The typical task involving the use of libraries is the copying of objects (subsystems, blocks or elements) into a SISTEMA project. This can be achieved by several different methods, which are described below. These methods do not enable an object to be moved, i.e. deleted from the SIS-lib. Objects can be deleted from the SIS-lib only by means of the "Delete" command, and then only if the SIS-lib is not write-protected.

A general rule is that objects may be inserted only at suitable points. The paste operation is not otherwise performed, or the "Paste" or "Load" commands are not active, but greyed out. The following constraints apply:

- An element can be inserted either into a block or into a channel/test channel and converted in the process into a block.

- A block can be inserted either into a channel/test channel or into a block and converted in the process into an element.

- A subsystem can be inserted only into a safety function.

If one or more objects of the same type are already present, the inserted object is appended by default to the last object of this type in the list. Only if drag and drop is used for copying (see Section 7.8.3) can the insertion point within a list be specified (by means of the mouse pointer).

### 7.8.1   Copying via the clipboard

An object can always be copied from a library and pasted into a project via the Windows clipboard, as follows: use the right mouse button to open the context menu for the object to be copied from the library, and select the "Copy" command. The "Cut" command is not available. The name of the object in the clipboard is displayed in the status bar. In the main menu, select the intended location of the desired project ("Channel 2" in Figure 34) and open the context menu again. The object (a block in this example) can then be inserted at the desired point by means of the "Paste" command. The object can be inserted multiple times at suitable locations, provided the clipboard is not overwritten. The "Paste" and "Copy" commands are also accessible through the Ctrl+V and Ctrl+C shortcuts.

Figure 34: Using the clipboard to insert a library object into a project
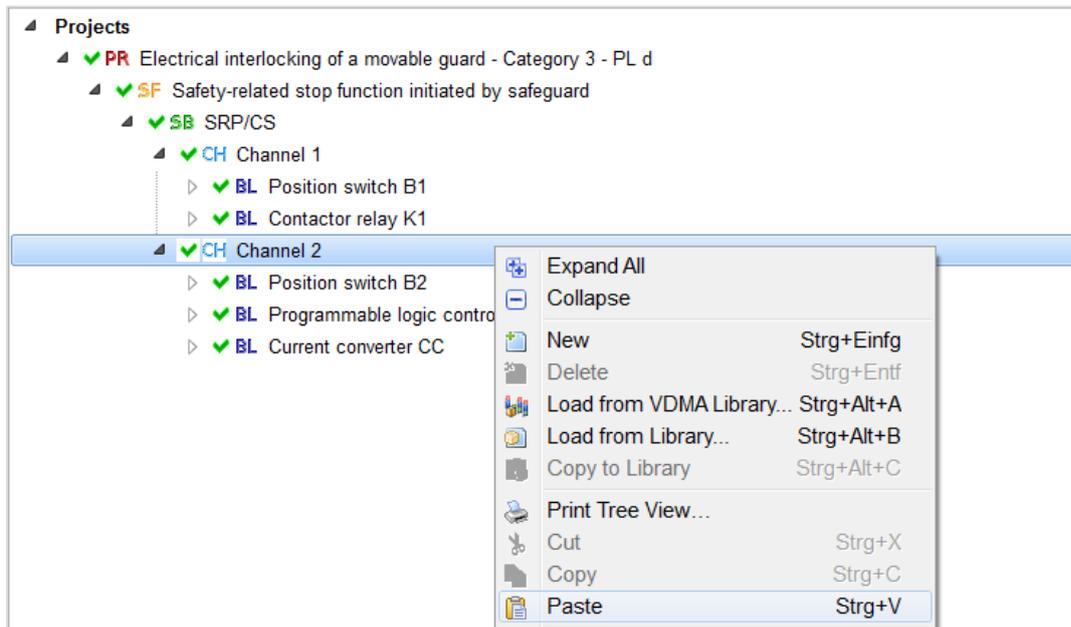


## 7.8.2   Load from Library

Figure 34 shows a further command in the context menu: "Load from Library". When this command is executed, the selected point (in this case "Channel 2") in the project is marked, and the library window opens. The desired library and the appropriate object can now be selected (the "Contactor relay K1" block in Figure 35). A right mouse click on this object opens the context menu, which now contains the commands "Load Selection" and "Load & Close". Both commands copy the object and insert it at the same time at the point marked beforehand (in this case, "Channel 2"). "Load & Close" closes the library window following this action. Conversely, it remains open after "Load Selection", and further objects (in this case, further blocks) can be selected from libraries and inserted at the point marked without the user having to toggle to the project.

Figure 35: Using the "Load Selection" command to insert a library object into a project
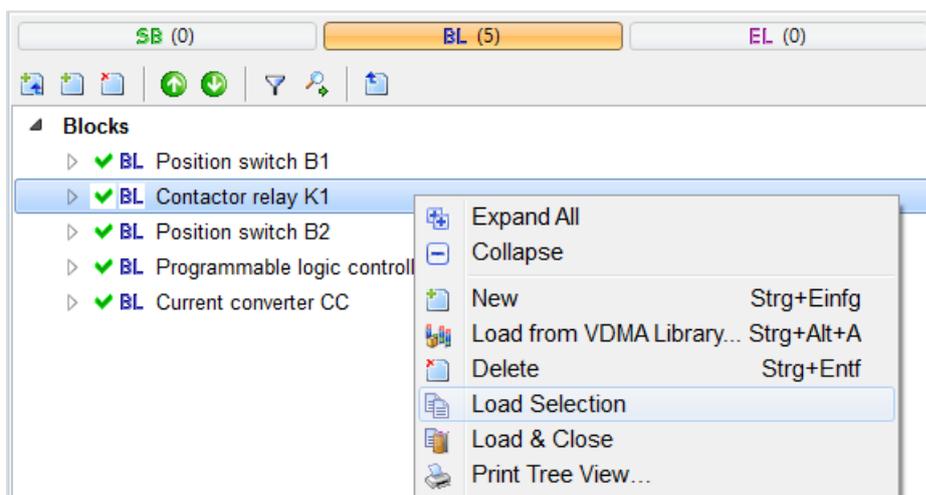


As an alternative to the context menu, the commands can also be executed by means of the "Load Selection" and "Load & Close" buttons below the navigation pane, as shown in

Figure 35. As an alternative to the "Load from Library" command in the context menu, it is sufficient to mark the target object and then to toggle to the library window. If the two windows are arranged adjacent to each other on the screen, this is the quicker method.

### 7.8.3   Copying with drag and drop

A fast alternative to use of the clipboard is to copy an object from a library using the familiar "drag and drop" function, as described in Section 11.2.5. In this case, select the object in the library window by means of the left mouse button, and keep the mouse button depressed. With the mouse button still depressed, drag the object to the main window in a project. Place the mouse pointer over the desired object folder in the project, then release the mouse button to drop the object. The object is appended to the list. Conversely, if the object is dropped onto an existing object in the list, the copied object is placed directly beneath it in the list. This enables the sort order in the list to be controlled. The object name is displayed adjacent to the mouse pointer as the object is dragged.

Drag and drop functions best when the two windows are located adjacent to each other on one or two screens. Alternatively, the "Alt+Tab" function can be used to toggle between the two windows with the mouse button depressed.

### 7.8.4   Order in the display and resorting objects

Once the objects have been copied into a project, their order in the list in the navigation pane can be modified. This can be achieved in three different ways, as described in Section 11.3:

- Select an object and use the commands "Move One Up/Down" (context menu, button, keyboard shortcut).

- Move the objects by means of drag and drop: the object to be sorted is placed above or below another object when dropped upon it, depending upon the latter's position in the list.

- Use the commands "Sort A-Z" or "Sort Z-A" to resort the object list.

## 7.9   Adding content to copied objects

When an object has been copied into a project, information must often be added in order to permit proper analysis of the object or safety function. Only the user is able to do this, since he or she is familiar with the control system concerned. Such information is for example the diagnostic coverage (DC), and in the case of components subject to wear, statement of the mean number of operations per year ($n_{op}$).

Even if the SIS-lib is write-protected, this information can still be entered by the user after the object has been copied into the project. It is essential that inadvertent changes to other values in the process be avoided.

To enable the new object to be identified in the project, an item designation, possibly an inventory number and further project-specific comments are also often added to the object name or the documentation.

## 7.10 Linking between SISTEMA projects and library objects

When an object is copied from a SIS-lib and inserted into a SISTEMA project, for example via the clipboard, the object and all its data are embedded in the project's object table. A link to the original library is not entered. Changes to the object in the project therefore have no effect upon the data in the SIS-lib. By the same token, subsequent modifications to the object in the SIS-lib have no influence upon the copy in the project.

At the present time (Version 2.0.x), SISTEMA is not able to check whether a more recent version of an object in a project exists in a SIS-lib. Nor is it possible to check in what projects a given object from a SIS-lib has been used.

# 8   Working with VDMA libraries

This chapter describes how libraries in the format of VDMA Specification 66413 ("VDMA-libs") can be opened and objects from them copied into a project or a SIS-lib. In contrast to the usual SIS-libs, it is irrelevant whether the VDMA-libs are stored locally or on a network PC. Nor does a database server need to be installed on the network PC.

## 8.1   Opening VDMA libraries

The command for loading VDMA-libs into the VDMA library window is "Add Library". This command opens a file manager (Figure 36) in which the folder and the VDMA-lib can be selected and added to the list of manufacturers and libraries with the "Open" button. At the same time, the VDMA-lib is loaded into memory and selected. The objects it contains are displayed in the navigation pane (Figure 17).

Figure 36: File manager for the loading of VDMA libraries
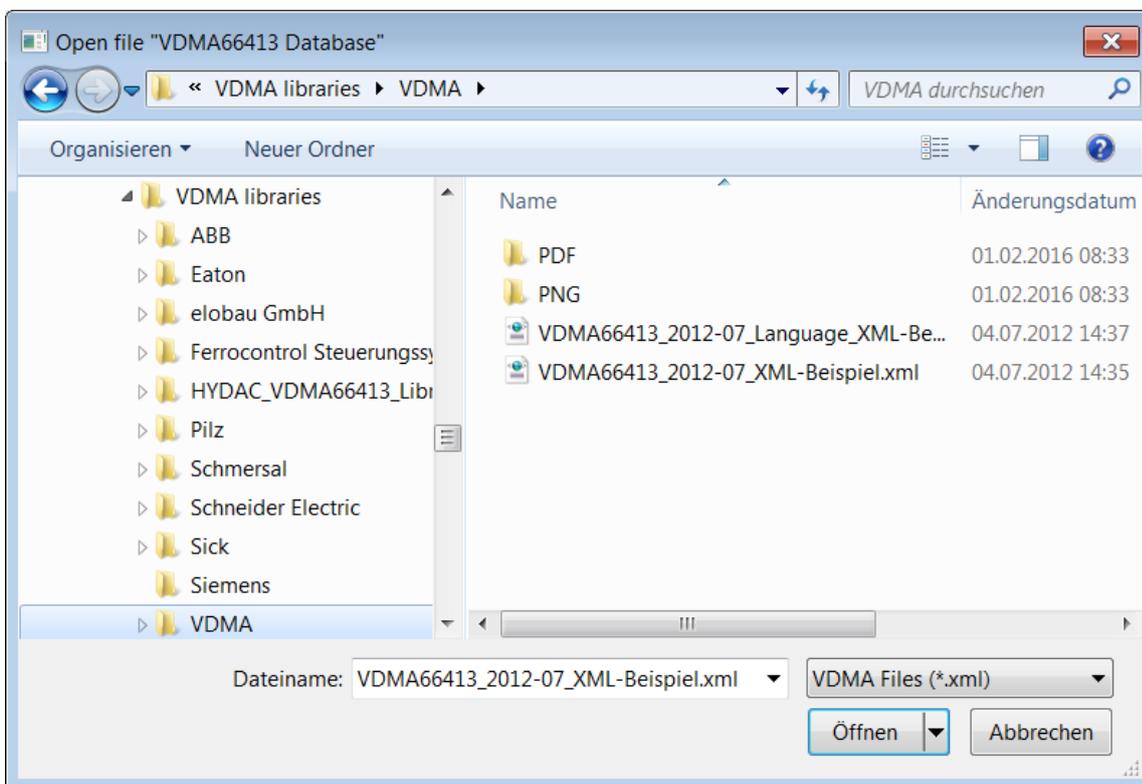


## 8.2   Closing VDMA libraries

The name and the file name of each VDMA-lib that has been loaded by means of the functions referred to above are saved in the list of manufacturers. The list is stored in the registry each time SISTEMA is closed and is loaded again when SISTEMA is next launched. The "Close Library" or "Close" command (in the "File" menu, button or context menu) deletes the selected library name from the list. This enables the list to be kept concise and clear. Removing library names from the list does not delete the library files.

## 8.3   Compatibility of VDMA libraries

The version of the VDMA Specification on which it is based is stated for each VDMA-lib in the "VDMA 66413 Version" data field in the workspace. Libraries must be adapted by the manufacturers when changes are made to definitions in the Specification. SISTEMA is also adapted accordingly and a new version issued.

## 8.4   Paths for linked documents

Objects in a VDMA library are often linked in addition to documents. The "Document" input field in the various tabs is used to link these documents. Clicking on the folder symbol to the right of the input field opens the document in the relevant application.

## 8.5   Copying objects from libraries

All devices in a VDMA-lib together with their use cases are always displayed in the navigation pane when the object tree is maximized (Figure 17 on Page 27). Each use case corresponds to a SISTEMA object type (subsystem, block or element, see Section 5.2.1, Area B).

The typical task involving the use of VDMA-libs is the copying of these objects into a SISTEMA project or alternatively into a SISTEMA library. This can be achieved by several different methods, which are described below. It is however not possible for an object to be moved, i.e. deleted from the SIS-lib. Objects cannot be deleted from VDMA-libs, which are always write-protected.

A general rule is that objects must be inserted at suitable points. The paste operation cannot otherwise be performed, and the "SISTEMA Project" or "SISTEMA Library" commands are not active, but greyed out. The following constraints apply:

- An element can be inserted either into a block or into a channel/test channel and converted in the process into a block.

- A block can be inserted either into a channel/test channel or into a block and converted in the process into an element.

- A subsystem can be inserted only into a safety function.

If an object of the same type already exists, the inserted object is always appended after the last object of this type in the list. Only if drag and drop is used for copying (see Section 8.5.3) can the insertion point within a list be specified (by means of the mouse pointer).

### 8.5.1   Copying via the clipboard

An object can always be copied from a VDMA-lib and pasted into a project or a SIS-lib via the Windows clipboard, as follows: use the right mouse button to open the context menu for the object to be copied from the library, and select the "Copy" command. The name of the object in the clipboard is displayed in the status bar of the main window. Then, in the main menu, select the intended location in the project, open the context menu again and insert the object at the desired point with the "Paste" command. The object can be inserted multiple times at suitable locations, provided the clipboard is not overwritten. The "Paste" and "Copy" commands can also be executed through the "Ctrl+V" and "Ctrl+C" shortcuts.

### 8.5.2   Loading from a VDMA library

When using the "Load from VDMA Library" command, mark the selected point in the project (e.g. a safety function) as the destination. The VDMA library window then opens. The desired library and the appropriate object can now be selected (in Figure 37, the subsystem: "Drive D.01, standard use case"). Right-clicking with the mouse on this object opens the context menu, which also contains the commands "SISTEMA Project" and "SISTEMA Library". Both commands copy the object and insert it at the point marked beforehand in the project, i.e. the currently selected SIS-lib. Further objects of the same type can be selected from libraries and inserted at the point marked without the user having to toggle to the project.

Figure 37: Using the "SISTEMA Project" command to insert a library object into a project
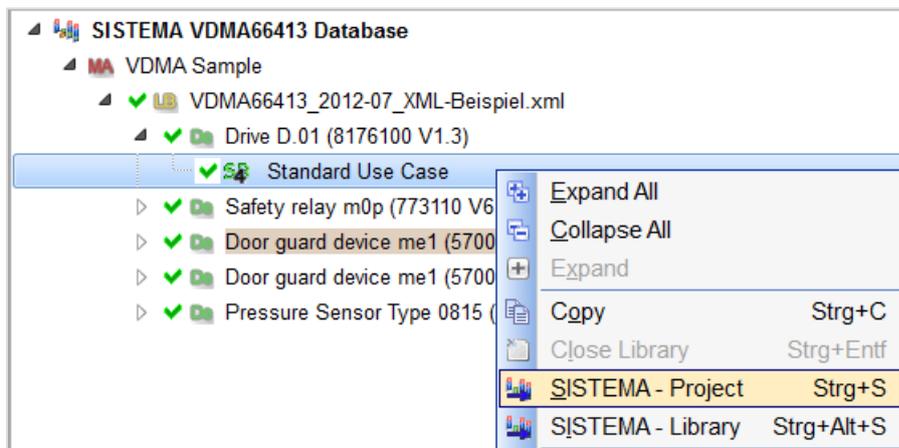


As an alternative to the context menu, the commands can also be executed by means of buttons adjacent to the list of use cases (Figure 21, left).

### 8.5.3   Copying with drag and drop

A fast alternative to use of the clipboard is to copy an object from a VDMA-lib using the familiar "drag and drop" function, as described in Section 11.2.5. Select the object (a use case) in the library window by means of the left mouse button. With the mouse button still depressed, drag the object to the main window in a project. Place the mouse pointer over the desired higher-level object in the project, then release the mouse button to drop the object. The object is appended to the list. Conversely, if the object is dropped onto an existing object of the same type in the list, the copied object is inserted into the list directly beneath the existing object. This enables the sort order in the list to be controlled. The object name is displayed adjacent to the mouse pointer as the object is dragged.

Drag and drop is most convenient when the two windows are arranged adjacent to each other on one or two screens. Alternatively, the "Alt+Tab" key combination can be used to toggle between the two windows with the mouse button depressed.

### 8.6   Adding content to copied objects

When a use case has been copied into a project in the form of an object, information must often be added in order to permit proper analysis of the object or safety function. Only the user is able to do this, since he or she is familiar with the control system in question. Such information concerns for example the diagnostic coverage (DC) values, and in the case of

components subject to wear, statement of the mean number of operations per year ($n_{op}$). The item designation and inventory number can also be entered if required.

Conversely, once copied into the project/SIS-lib, the device data and values can no longer be modified by the user.

## 8.7   Linking between SISTEMA projects and VDMA library objects

When an object is copied and inserted into a SISTEMA project, for example via the clipboard, the object and all its data are embedded in the project's object table. A link to the source library is not entered. Changes to the object in the project therefore have no effect upon the data in the VDMA-lib. By the same token, subsequent modifications to the object in the VDMA-lib or newer versions of the VDMA-lib have no influence upon the copy in the project.

At the present time (Version 2.0), SISTEMA is not able to check whether a more recent version of an object in a project exists in a VDMA-lib. Nor is it possible to check in what projects a given object from a VDMA-lib has been used.

# 9      Libraries for operators of machinery

Operators should ensure that they receive not only the SISTEMA project files and the summary for the machine control system, but also the manufacturers' original libraries. Extensions or modifications may make it necessary to update these projects. Safety-related components may be held as spare parts, and fitted. The corresponding (older) libraries are then required for these components.

# 10    Modelling of components/parts of control systems in SISTEMA libraries

This chapter describes how components and parts of control systems can be modelled as objects for SISTEMA libraries. It is intended both for manufacturers of products and for persons using SISTEMA and wishing to create their own user libraries. SISTEMA cannot however be used to create or edit libraries in the VDMA Specification 66413 format (see Annex B).

The possibilities described in this chapter are implemented in a SISTEMA library of examples accompanying SISTEMA Cookbook 5. The model library can be downloaded here:

http://www.dguv.de/webcode/e109249

Chapter 11 describes the steps for the input of objects into a library.

## 10.1 Using the SISTEMA objects

Distinction must be drawn between three object types: subsystems, blocks and elements. Components/parts of control systems can be modelled in objects in several different ways. Possible strategies are described below for the creation of a library. Annex D shows an example of a list in which typical components/parts of control systems are mapped to the object types.

SISTEMA is flexible in the modelling of components/parts of control systems in SISTEMA objects. Two essential alternatives exist:

1. The resulting parameters (e.g. PL, SIL, $PFH_D$, $MTTF_D$) of the components/parts of control systems are calculated not in SISTEMA, but by other means. These values can then be entered directly into the suitable SISTEMA objects (Figure 38, example SB1). This alternative is described in Section 10.1.1.

2. The resulting values are calculated in SISTEMA from a number of defined sub-objects (Figure 38, example SB2). The parameter values that can be defined only with knowledge of the application, such as $n_{op}$, can be left open. This alternative is described in Section 10.1.2.

Figure 38: Alternatives for the modelling of components/parts of control systems (example: subsystems)
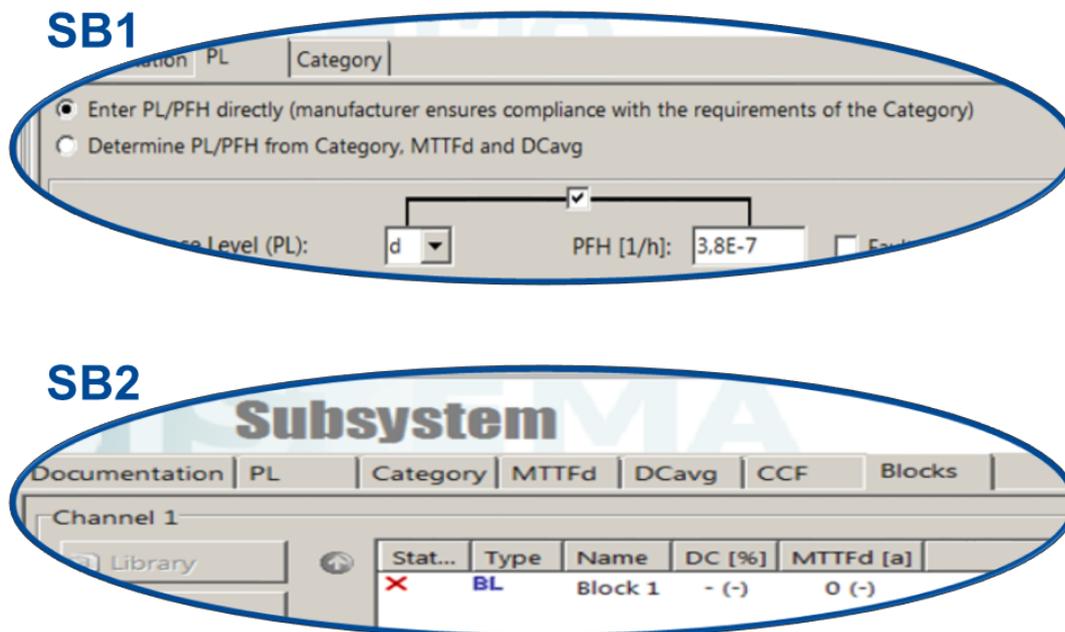


### 10.1.1 Direct input of parameters in SISTEMA objects

**Subsystems** represent single-channel or two-channel parts of control systems for which a PL or SIL with a $PFH_D$ value can already be stated following an analysis. They are referred to here as "encapsulated subsystems" (generally safety components such as safety PLCs or safety modules).

This enables these parts of control systems to be combined with further subsystems. Subsystems may also have a $PFH_D$ value of 0 [1/h] or an exclusion of dangerous component failure. These parameter values can be assigned in the "PL" tab. With reference to VDMA Specification 66413 (see 0 of this cookbook): the subsystem satisfies both the "device type 1" and the "device type 4" defined in the Specification (special case of $PFH_D$ = 0).

Examples of safety components constituting subsystems: electro-sensitive protective equipment, safety PLCs, safety modules, electric drives with integrated safe stop functions, etc.

Examples of parts of control systems constituting subsystems: position monitoring with actuated switches, control logic employing relays, contactors or combinations of frequency converter and mains contactor, combination of hydraulic or pneumatic valves, etc. Such combinations of standard components can all constitute single and two-channel Categories.

From SISTEMA Version 2 onwards, standard components can be modelled either as **blocks** or as **elements**. The input dialogs are the same.

**Blocks** represent standard components for which an $MTTF_D$ (or MTTF, MTBF, RDF or a failure rate $\lambda_D$) and a mission time $T_M$ can be stated by the manufacturer. These components are typically not subject to wear, i.e. the $MTTF_D$ is not dependent upon the number of operations. This is essentially the case for electronic, mechanical and hydraulic components. The diagnostic coverage (DC) for these components must typically be determined by the user and entered for the component in the project. With reference to VDMA Specification 66413: These components correspond to the "device type 2" defined in the specification.

Examples of standard components not subject to wear constituting blocks or elements: mechanical parts of control systems, standard light barriers, standard PLCs, standard frequency converters, hydraulic valves, electronic components such as integrated circuits, resistors, capacitors, etc.

Conversely, standard components subject to wear exist for which the manufacturer states a $B_{10D}$ (or $B_{10}$ and RDF) value and the mission time $T_M$. The $MTTF_D$ calculated for these components is dependent upon the number of operations $n_{op}$. This essentially applies to electro-mechanical and pneumatic components. The $n_{op}$ must typically be determined by the user and entered together with the DC in the project for these components. With reference to VDMA Specification 66413: These components correspond to the "device type 3" defined in the Specification. Components modelled as elements are grouped in a block during creation of a project, as a result of which a resulting $MTTF_D$ is calculated for the block.

Examples of standard components subject to wear constituting blocks or elements: switches, relays, contactors, pneumatic components, etc.

### 10.1.2 Calculation of parameters from assembled SISTEMA objects

The resulting parameters for a part of a control system can be calculated in the SISTEMA library window from a number of lower-level objects constituting the individual components of the part of the control system. The PL and $PFH_D$ of a subsystem for example are calculated from a number of blocks, and the component parameters ($MTTF_D$, $\lambda_D$, $B_{10D}$ etc.) must then be entered for the blocks. Alternatively, the resulting $MTTF_D$ of a block is calculated from several lower-level elements and their component parameters ($MTTF_D$, $\lambda_D$, $B_{10D}$ etc.). All these defined objects are then saved in the library, are visible to the user, and are copied together into a project.

The advantage is that application-specific values which have an influence upon the reliability can be added or modified later. For components subject to wear, this concerns first and foremost statement of the number of operations of the component, $n_{op}$. It also applies to the diagnostic coverage DC, if this can be determined only within the application in accordance with the test measures taken.

## 10.2 Data required for SISTEMA objects

The objects stated should contain the following parameters and information in the library:

**Data for subsystems:**

- Name

- Performance Level (PL) or Safety Integrity level (SIL) attained

- Average probability of a dangerous failure per hour ($PFH_D$)

- Category

- Mission time ($T_M$; 20 years is the default value)

- Application-specific conditions, for example for fault exclusion (statement in the "Documentation" field)

- Documentation/references or files/links (e.g. data sheets; statement in the "Documentation" field or as a link in the "Document" field)

**Data for blocks or elements:**

- Name

- Mean time to dangerous failure ($MTTF_D$ or $\lambda_D$); for elements alternatively $B_{10D}$. The appropriate $n_{op}$ must be determined by the machine manufacturer according to the application. As an alternative to calculating the $MTTF_D$ from the $B_{10D}$, the manufacturer can also state $MTTF_D$ benchmark values for typical $n_{op}$ ranges directly in the "Documentation" field (see Section 10.3.1).

- Where applicable: well-tried component (statement in the "Documentation" field), with reference to specific applications

- Diagnostic coverage (DC) which can be attained by built-in test facilities, such as positive mechanically linked contacts, including requirements for the interpretation measures (statement in the "Documentation" field)

- Mission time ($T_M$; 20 years is the default value)

- Application-specific conditions, for example for fault exclusions (statement in the "Documentation" field)

- Documentation/references or files/links (e.g. data sheets; statement in the "Documentation" field or as a link in the "Document" field)

## 10.3  Strategies for the creation of libraries

As described in Section 10.1, alternative structures of the SIS-lib are possible. These alternatives and the flexibility for the SIS-lib are described here for typical cases. A distinction is drawn here between individual components and parts of control systems (comprising multiple components). The alternatives can be found in the SISTEMA library of examples (refer to the beginning of Chapter 10).

The manufacturer does not generally enter a DC value for standard components; the default value of DC = 0% remains unchanged. The DC value must be added later in use according to the diagnosis performed in the project.

### 10.3.1  Components subject to wear

Three alternatives exist for the modelling of standard components subject to wear (for examples, refer to Annex D):

A    Manufacturers can model components subject to wear typically in a block or element, this being the only place in which the characteristic parameter $B_{10d}$ can be entered. At the same time, no value may be entered for $n_{op}$, since this value must be entered at a later stage when the copy of the object is used in the project concerned. An error message with the red cross appears in the SIS-lib ("Please enter a positive real number for the Nop value of this element."). From the manufacturer's perspective, the component is nevertheless fully modelled in the SIS-lib. The message, which cannot be overlooked, indicates that the $n_{op}$ must be entered here at a later stage. The $MTTF_D$ of the block or element is calculated correctly – not in the SIS-lib, but in the project.

B    The manufacturer enters the components with the $B_{10D}$ parameter in a block or element and supplements them with an estimated value for $n_{op}$ that in his view is typical or erring on the side of safety. The $MTTF_D$ of the element is already calculated in the SIS-lib for this parameter assignment. An error message is no longer output. Attention must how-

ever be drawn clearly to this parameter assignment in the documentation or in the name of the block or element, in order to prevent the component from being analysed incorrectly in use should different operating cycles ultimately apply in the application. The $n_{op}$ parameter can however be adapted to the application concerned, resulting in the $MTTF_D$ of the element being recalculated in the project.

C    The manufacturer enters the components in a block or element and determines the $MTTF_D$ of the block or element from a combination of the specified $B_{10D}$ and an assumed $n_{op}$. An error message is no longer output. Attention must however be drawn clearly to this parameter assignment in the documentation or in the name of the block or element, in order to prevent the component from being analysed incorrectly in use. An extension of this alternative is for the manufacturer to state several $MTTF_D$ values for typical $n_{op}$ ranges in the "Documentation" input field. These $MTTF_D$ values must be adapted according to the application concerned.

The manufacturer and the user must consider that the number of annual operations $n_{op}$ of a component is derived in part from the requirements of safety functions, but that process functions may also lead to operations resulting in wear. The $n_{op}$ value must reflect the total operations.

### 10.3.2 Components not subject to wear

A typical procedure exists for the modelling of standard components not subject to wear (for examples, refer to Annex D):

A    Modelling as a block or as an element with the characteristic parameter $MTTF_D$. These object types enable complex parts of control systems and further standard components to be assembled and analysed. The manufacturer does not enter a DC value; the default DC = 0% remains unchanged. The DC value must be entered later according to the diagnosis performed in the project.

B    Only in rare cases will such a standard component be modelled as a subsystem in Category B/1. The $MTTF_D$ must be entered directly in the "$MTTF_D$" tab of the subsystem. Diagnosis is not applicable. A subsystem of this kind can no longer be combined with further standard components to form a redundant part of a control system.

### 10.3.3 Combinations of components subject and not subject to wear

The manufacturer of a component comprising a combination of components subject to wear (such as relay outputs) and not subject to wear (such as electronics) can use the following alternatives to model the component (Figure 39, without alternative D):

A    As an element, block or subsystem with a single $MTTF_D$ or $PFH_D$ value specified by the manufacturer, which assumes a worst-case approach for $n_{op}$ for the aspect subject to wear. The worst-case assumption for $n_{op}$ (i.e. the maximum permissible switching frequency) is stated in the "Documentation" input field.

B    As a block/element with $MTTF_D$ input field or as a subsystem with $PFH_D$ input field. The $MTTF_D$/$PFH_D$ fields are not completed. This forces the appropriate value to be entered from the "Documentation" input field in use. Alternatively, the fields are completed with the poorest table value, i.e. the worst-case approach. In the "Documentation" input field, the table values for the $MTTF_D$ or $PFH_D$ are listed as a function of the $n_{op}$, for manual transfer to the input fields.

C    In the form of two (or more) elements connected in series, one with statement of the MTTF$_D$ (C-1 in the SISTEMA library of examples) and one with statement of the B$_{10D}$ (C-2), but without statement of the n$_{op}$. These must be entered later. Summation of the failure rates for these elements yields the sum for the combined component (the block). In the "Documentation" input field, it must be stated clearly that the elements of the block belong together and must not be used separately.

D    In the form of two (or more) subsystems connected in series, one encapsulated with statement of the PFH$_D$ (D-1 in the SISTEMA library of examples), the other not encapsulated (D-2), with statement of the B$_{10D}$ in the elements. The two probabilities of failure PFH$_D$ of these subsystems are summated for the combined component. In the "Documentation" input field, it must be stated clearly that the two subsystems belong together and must not be used separately.

Figure 39: Alternatives for combinations of components subject and not subject to wear
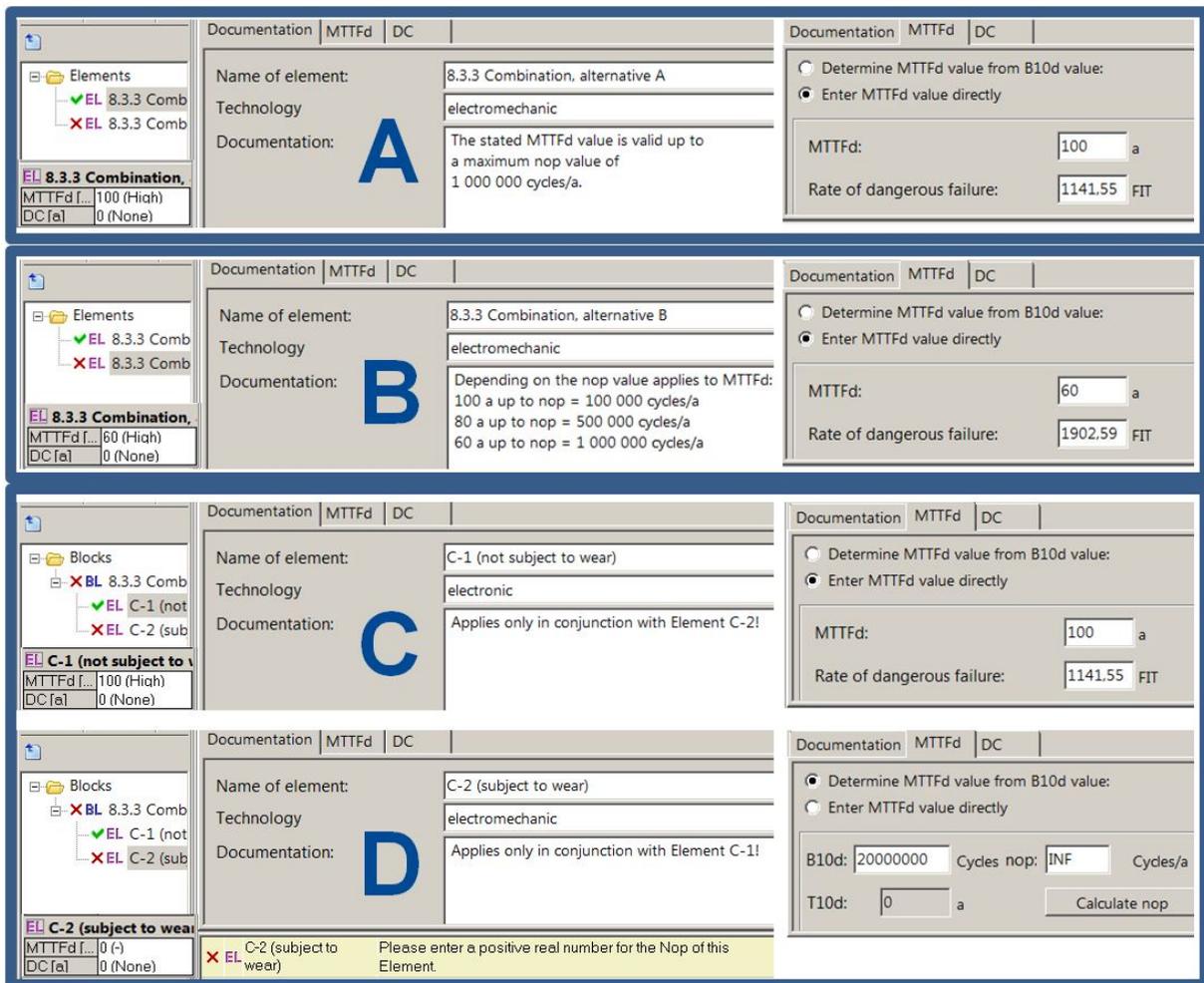


Further alternatives certainly exist; the manufacturer can select the model best suited to his scenario.

### 10.3.4 Parts of control systems comprising standard components (enter components individually)

Parts of control systems comprising multiple components are modelled as subsystems. All components are analysed by SISTEMA following input. A condition is that the components under analysis constitute one of the designated architectures. Components subject to wear and not subject to wear can be combined, even where they employ different technologies. Two essential alternatives exist here:

A   The manufacturer has added all the necessary component values and a specified application to the part of the control system. The circuit is specified in terms of the DC measures and the number of operations $n_{op}$. Users need enter no further parameters. An error message is not output.

B   Only for the components has the manufacturer added all necessary values to the part of the control system. In use, the DC measures and/or the CCF analysis and/or the number of operations $n_{op}$ must be added in consideration of the specific conditions of use. A number of error messages with the red cross appear (including "Please enter a positive real number for the $n_{op}$ value of this element."). The requirements imposed upon the Category may also have to be confirmed.

### 10.3.5 Parts of control systems produced from standard components (enter parameters individually)

These parts of control systems are generally modelled as encapsulated subsystems, and the resulting parameters that have already been calculated are entered directly. The individual components are not entered. Several alternatives exist:

A   PL/SIL, $PFH_D$ and the Category are available for the part of the control system. They are entered directly in the "PL" tab of the subsystem. No further parameters need be entered. An error message is not output.

B   A variant of alternative A is that the $PFH_D = 0$ [1/h]. The fault exclusion can be entered. The tick linking the input fields PL/SIL and $PFH_D$ must be removed for this purpose.

C   A further variant of alternative A is that the PL/SIL and $PFH_D$ do not lie within the same value range. The tick linking the two input fields must be removed in this case. The two values can then be entered separately from each other.

D   No $PFH_D$ values are available; the $MTTF_D$ (or fault exclusion) and $DC_{avg}$ values for the part of the control system are however available. These values are entered directly in the "$MTTF_D$" and "$DC_{avg}$" tabs of the subsystem. The option for this input function must be activated (Expert Setting -> Permit Direct Input of $MTTF_D/DC_{avg}$ in a subsystem). The CCF analysis must also be performed. The $PFH_D$ value is calculated from these values, as is the PL. It must then be checked whether the measures against failures caused by systematic faults (including software faults) satisfy the calculated PL; if not, the PL must be reduced.

### 10.3.6 Parts of control systems involving an encapsulated subsystem

Parts of control systems are typically made up of a number of standard components (with $MTTF_D$) in accordance with the designated architectures of EN ISO 13849-1. However, if "encapsulated" subsystems are used in one channel of a two-channel structure, the $MTTF_D$ – which is required for analysis – of this subsystem is not available. In order for parts

of control systems employing encapsulated subsystems to be modelled and analysed despite this, the corresponding $MTTF_D$ for one channel must instead be determined from the $PFH_D$ and PL values stated by the manufacturer.

The use of an encapsulated subsystem in Categories 2, 3 or 4 in only one channel is in fact not economically advantageous. Such circuits do however exist in practice. The treatment of these subsystems is described in SISTEMA Cookbook 4, Section 2, "When the designated architectures don't match".

# 11    Creation of SISTEMA libraries

This chapter describes the creation of SISTEMA libraries in the library window, and is intended both for the manufacturers of products, and for persons wishing to create their own SISTEMA user libraries. SISTEMA cannot however be used to create or edit libraries in the VDMA Specification 66413 format (see Annex B).

## 11.1  Creating a new library

Before components can be input into a SIS-lib in the library window, either an existing SIS-lib must be opened (see Sections 7.1/7.2) or a new SIS-lib created (menu bar or toolbar in the library window, see Sections 4.2.8/4.2.7). Creation of a new SIS-lib is described below. First execute the "Create New Library" command. The file manager opens. Select a suitable loca-tion in which to save the library; this must be on a local drive (e.g. C:\, D:\). Enter the file name of the new SIS-lib with the file extension ".slb" in the input line and click on the "Save" button (Figure 40), e.g. the file name "New Library.slb").

Figure 40: File manager dialog for creation of a SIS-lib
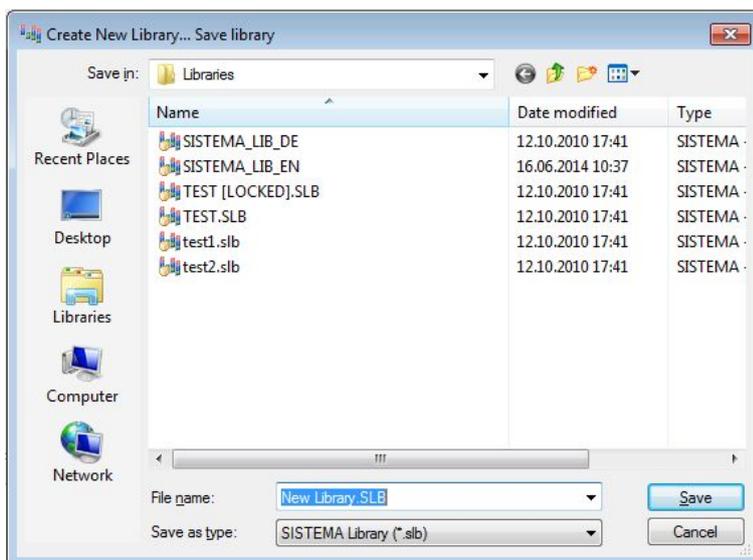


The file name of the new SIS-lib is added to the list of libraries. It can be changed by modifi-cation of the "Name" property on the right in the properties window. This name is displayed in the list (Figure 41, for example "New User Library"). Figure 41 also shows the difference between the database names (left) and the file names (right) of the two SIS-libs.

Figure 41: List of libraries showing a new SIS-lib



The SIS-lib selected from the list by means of the mouse is loaded into RAM and displayed in the navigation pane.

## 11.2 Inserting objects into a library

A number of methods are described below for the input of components/parts of control systems into a library in the form of subsystem, block or element objects.

### 11.2.1 Direct input of objects in the workspace

The most obvious way of inputting new objects is to enter them in the workspace of the library window. For this purpose, select the desired object type using the object filter at the top of the navigation pane (Figure 11, p. 19) and define a new object (by means of the menu bar, button, context menu or the Ctrl+Ins keyboard shortcut). The folder of the object (the first line in the object tree) must be selected beforehand by means of the mouse; the commands referred to are not otherwise active.

All tabs of the new object must then be completed, as is familiar from the SISTEMA main window.

### 11.2.2 Input from a project using the context menu

A new object may already have been created in the SISTEMA main window as part of a project. In this case, it does not need to be entered manually a second time as described in Section 11.2.1. Select the object to be copied in the main window and use the right mouse button to open the context menu (Figure 42, for the "Position switch B1" block). The "Copy to Library" command copies the object into the appropriate object folder of the currently selected library (see status bar). A message is output that this change to the library must be saved (see Section 11.4).

Figure 42: Using the context menu to copy an object from a project
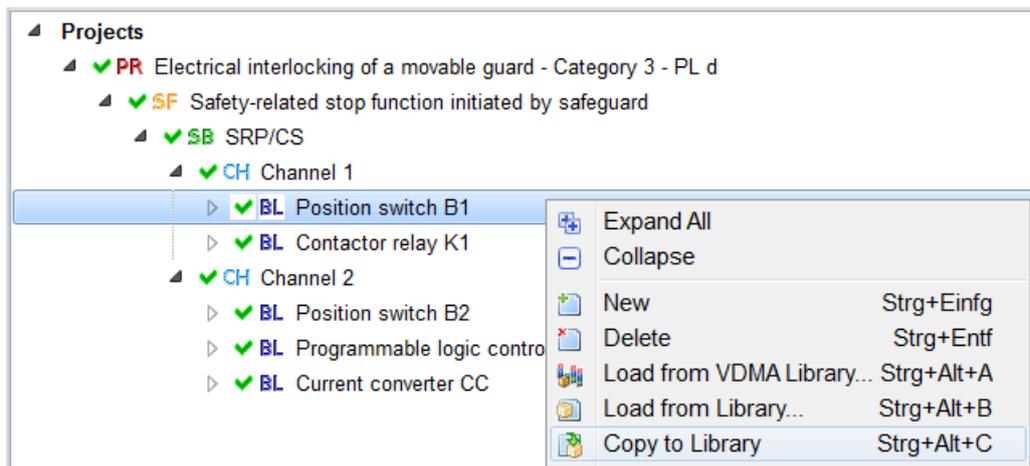


### 11.2.3 From VDMA-lib

Select the object to be copied (a use case) in the VDMA library window and use the right mouse button to open the context menu (Figure 43, for the standard use case). The "SISTEMA Library" command copies the object into the appropriate object folder of the currently selected SIS-lib (see status bar). The object data also remain write-protected in the SIS-lib.

Figure 43: Copying an object from a VDMA library into a SISTEMA library
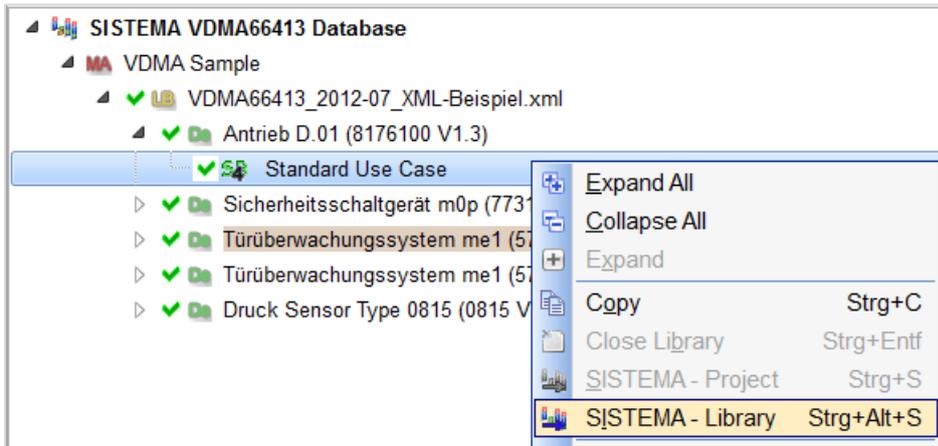


## 11.2.4  Input from a project or library by means of the clipboard

A finished object can always be copied or cut from a project by means of the Windows clipboard. In this case, open the context menu for the object to be copied and select the "Copy" or "Cut" command. Alternatively, an object can be copied from a different SISTEMA or VDMA library. The name of the object in the clipboard is displayed in the status bar. Select the appropriate object folder ("Blocks" in Figure 44) in the library window and use the right mouse button to open the context menu. Then use the "Paste" command to insert the object (in this case a block) into the desired SISTEMA library. An object cannot however be inserted into a VDMA library, owing to the write protection.

Figure 44: Inserting an object into the library using the clipboard
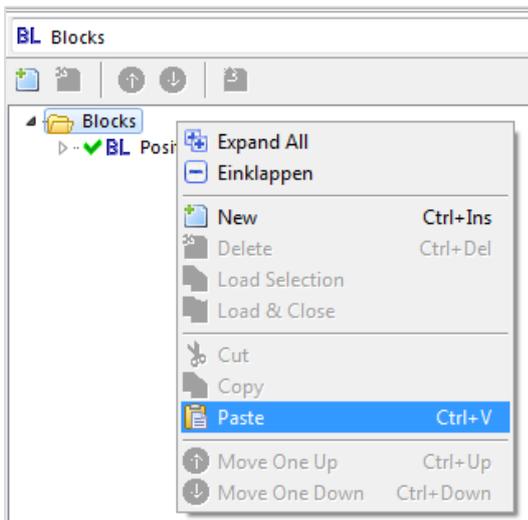


## 11.2.5  Input by drag and drop from a project or library

A quicker alternative to the clipboard is to use the familiar "drag and drop" function to copy an object from a project or library. In this case, select the object in the main window using the left mouse button, hold the Ctrl key, and keep the mouse button depressed. In a VDMA

library, a use case of a device must be selected. With the left mouse button still depressed, drag the object (the "Position switch B1" block in Figure 45) to the library window (dragging shown in the right-hand image). Position the mouse pointer for example over the object folder (which must first be adjusted as required), then release the mouse button to drop the object. The object is appended to the list. Conversely, if the object is dropped onto an existing object in the list, the copied object is placed directly below it. This provides control of the list sort order at the insertion stage. The object name is displayed adjacent to the mouse pointer as the object is dragged.

Figure 45: Moving objects from the project (left) to the library (right) by drag and drop
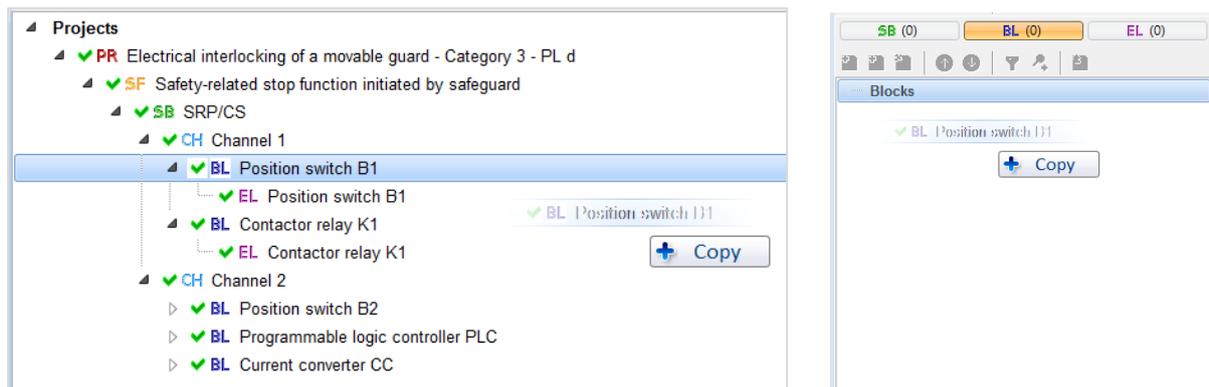


Drag and drop functions best when the two windows are located adjacent to each other on one or two screens. Alternatively, the "Alt+Tab" shortcut can be used to toggle between the two windows, even with the mouse button depressed.

## 11.3  Order in the display and resorting objects

The order of the objects in the list in the navigation pane can be modified and saved after they have been entered or copied into the library. This can be achieved in three different ways:

- Select an object and use the "Move One Up/Down" commands (context menu, button, keyboard shortcut, see Figure 46).

Figure 46: Using the context menu items or the buttons to sort the objects
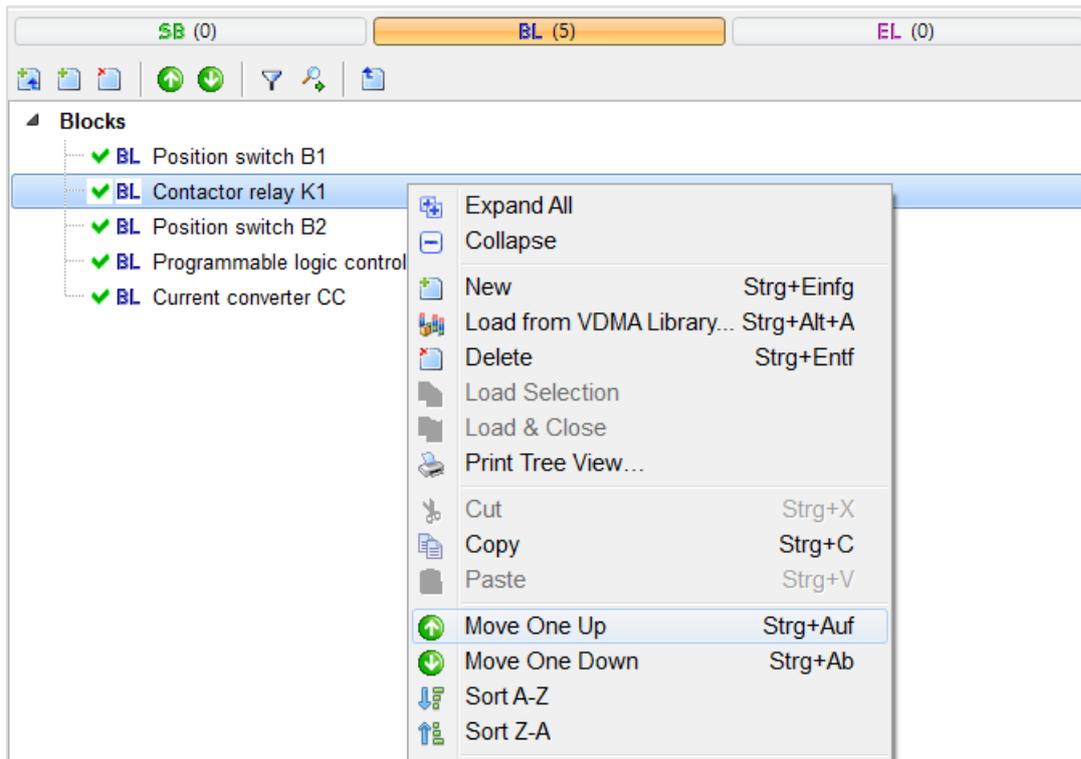


- Move the objects by means of drag and drop (as described in Section 11.2.5): when an object to be sorted is dropped onto another object, it is sorted above or below the latter, according to the latter's position in the list (in Figure 47, the "K1" block is placed **above** "B2").

Figure 47: Sorting objects by drag and drop
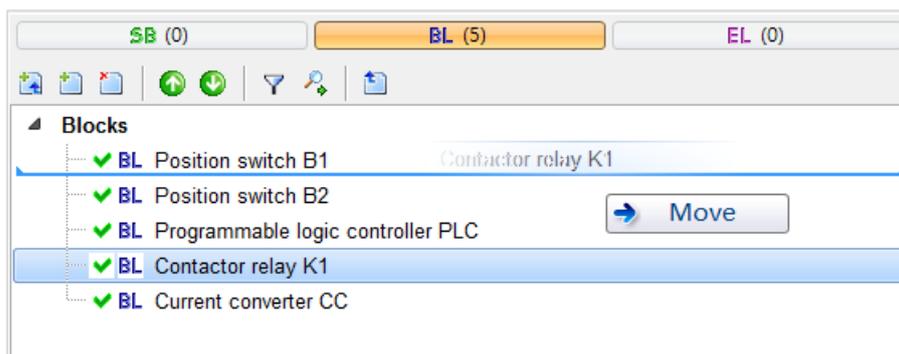


- Use the commands "Sort A-Z" and "Sort Z-A" in the context menu to resort the entire list, see Figure 46, bottom.

## 11.4 Saving changes in the library

When changes have been made to the library, the "Set Changes" button becomes active (Figure 48). To save the changes, click on this button, or use the menu bar or the Ctrl+S shortcut. After sorting, the button is deactivated again (greyed out).

Figure 48: "Set Changes" button
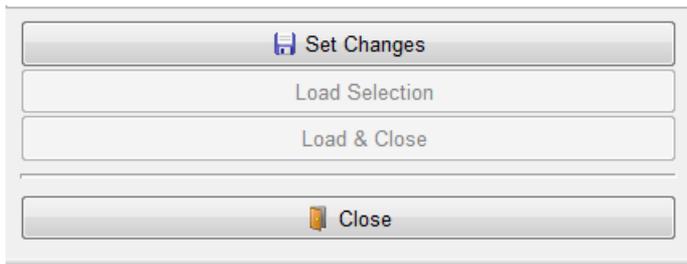


If the library window is closed without the changes having been saved, a prompt is output (Figure 49) and saving can still be performed ("Yes" button). Unwanted changes can be undone in this way ("No" button).

Figure 49: "Accept Changes" prompt when the library window is closed
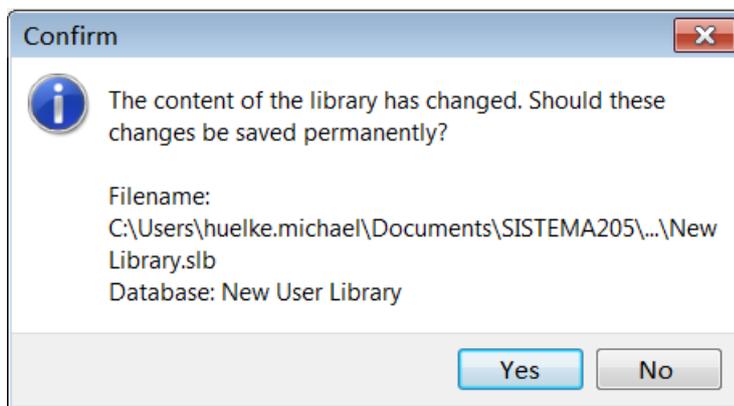


## 11.5 Recommendations for object names

In the list and tree views of SISTEMA, the objects are displayed and can be distinguished only by way of their object names. The names used for them in the SIS-lib should therefore contain all necessary **product-related** information such as the manufacturer's name, equipment name, variant, etc. Alternatively, SISTEMA Version 2 onwards has specific data fields for this information.

Names in SISTEMA may be up to 512 characters in length. In order to ensure compatibility with the database to VDMA 66413 (see Annex B of this cookbook), no more than 255 characters should be used in future. If the list or tree is not fully visible, the name is displayed in the pop-up view (when the mouse pointer is moved over the name). This name length is sufficient in practice.

## 11.6 Protecting libraries

### 11.6.1 Write-protection procedure

It is advisable to write-protect a SIS-lib before making it available to external users. This ensures that third parties cannot inadvertently modify the SIS-lib in SISTEMA and pass the modified version on under its original name. Manufacturers' SIS-libs are generally write-protected for this reason.

The procedure is as follows: create a duplicate of the SIS-lib (a second SLB file) and save it with a different name and file name. Write-protection of this duplicate is activated by SISTEMA. The result is therefore the original library, which is not write-protected, and its write-protected copy. The write-protection is implemented by attributes in the database and can be detected only by SISTEMA itself. SISTEMA indicates the write-protection status by a padlock symbol (Figure 50). A closed padlock indicates that the SIS-lib is write-protected.

**Important: the original file must not be deleted following write-protection, since the SIS-lib cannot then otherwise be modified. It is essential that you make backups of your original files.**

The write-protected duplicate of the SIS-lib can be opened in the library window, as in the case of the "New User Library [locked]" shown in Figure 50. The "[locked]" component of the name can be selected freely.

The objects in this library cannot be modified, not even by the person who write-protected the file! The objects can however be copied into SISTEMA projects or into other non-write-protected SIS-libs and modified there.

Figure 50: Padlock symbol indicating write-protection of SIS-libs
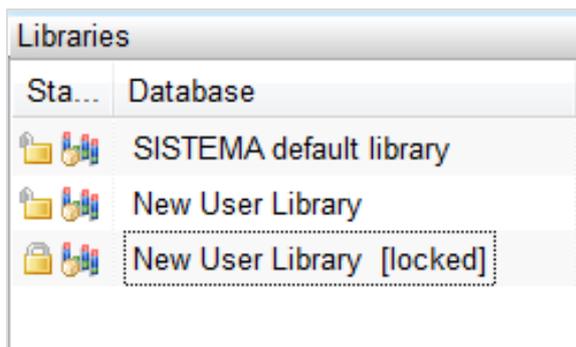


Write-protection has repercussions for the use of libraries created with older versions of SISTEMA. A write-protected library can be updated by reading-in of the non-write-protected original SIS-lib, which is then saved and write-protected following a (pseudo-)change. This is generally also possible even between major differences in the leading digit of the version number (e.g. between 1.1.6 and 2.0.0). Further information on version control and compatibility can be found in Sections 11.8 and 7.5.

### 11.6.2 Write-protecting libraries

To create a write-protected duplicate, select the original library in the list of libraries, then call up the "Lock Library" command from the menu or the toolbar. A dialog opens (Figure 51) in which a new library name can be defined. The original name with the "[locked]" suffix is proposed. The new name is however not subject to any restrictions other than that it must differ from the original name.

Figure 51: Input of the new library name for write-protected SIS-libs
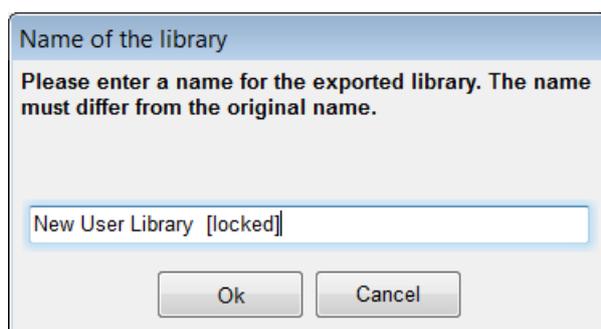


The write-protection procedure can still be aborted at this point by means of the "Cancel" button, in which case a new SIS-lib is not created.

Clicking on the "OK" button causes the new library name to be accepted, and a file manager opens (Figure 52) in which a new file name can be specified. The library name previously entered is proposed, with the addition of the "*.slb" extension. Any new file name can however still be selected here; the only requirement is that it differ from the original file name.

Figure 52: Input of the new file name for write-protected SIS-libs
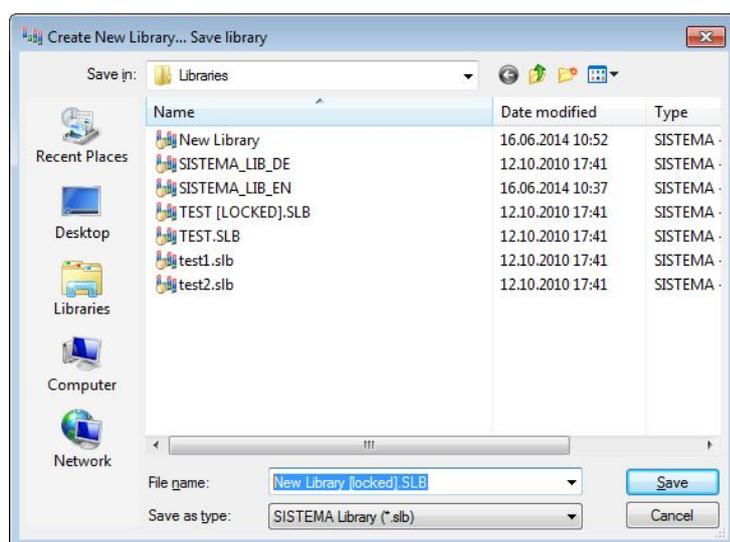


The write-protection procedure can still be aborted at this point by means of the "Cancel" button in the file manager, in which case a new SIS-lib is not created. Clicking on the "Save" button causes the new write-protected SIS-lib to be created, displayed at the end of the list of libraries, selected directly, and loaded into the navigation pane.

## 11.7  Documentation of objects

Objects in a library are often linked in addition to documents and further information concerning the application. These links are created by the "Document" input field in the various tabs of the objects. SISTEMA is not limited to PDF as a document format; other formats can be used for which the corresponding software is available.

Three forms of link exist: a URL (i.e. a web address such as "http://www.sample_company.de/Document.pdf") is possible, as is a relative path to the location of the library (e.g. "..\sample_company\Document.pdf"). Absolute paths such as "L:\sample_company\Document.pdf" are unlikely to be practical for manufacturers' libraries. The differences are as follows:

- URL: in order for a document to be opened, the local computer must be online and must be able to access the URL. This URL must also remain unchanged for the anticipated duration of use of the SIS-lib concerned, or it will no longer be possible to load the document. The advantage of this alternative is that documents do not need to be saved locally. Access to the documents is nevertheless possible at any time, even once the library objects have been copied to a project and the project has been passed on to further users, customers or suppliers.

- Relative path: the manufacturers' documents must be saved locally together with the SIS-lib. The local computer need not be online in order for a document to be opened, however. If a SISTEMA project created with these objects is passed to other parties, the documents linked to it should always be supplied with it, in order for these parties to be able to open the documents. Further information on relative paths can be found in the SISTEMA help under the "Options" menu item.

Only the link entered in the "Document" input field is embedded in the SIS-lib, and not the complete documents (files). If a URL is not stated, these external documents must therefore also be made available for the application in addition to the library. The documents are typically made available in the form of an archive file (for example in ZIP format) in which the folder structure of the documents corresponds to the path information in the "Document" input field.

Users who load this archive file onto their computers should unpack the files contained within it in the folder of their library, retaining the folder structure. This generally ensures that the documents retrieved from the object of a library can actually be opened. The opening of relative paths does not currently work in network libraries (Version 2.0): in this case, documents must be located and opened manually, or copied into a project.

At present (Version 2.0), documents are not automatically copied into a project along with the objects to which they relate. Users must therefore copy the documents themselves (see Section 7.6). It is generally sufficient to copy the document folder of the library into the project folder (by default, into the ".\documents" folder).

## 11.8 Version control and compatibility

SISTEMA projects and libraries bear a three-digit version number. The number is taken from the version of SISTEMA with which they were created or modified. Example: SISTEMA Version 1.1.6 was published in mid-2013. Projects and libraries created or modified with this version also receive the version number 1.1.6.

The significance of the three digits is as follows:

- The leading digit indicates major changes to the program concept and at the same time to the database structure.

- The middle digit generally changes when changes are made to the database structure. Program functionality may have been extended.

- The final digit is used for bug fixes and extended functionality; the database structure has **not** changed in this case.

Downward compatibility of libraries is assured only if the database structure has remained unchanged. As long as the database structure is identical (the leading and middle digits are unchanged), a SIS-lib from a more recent version (e.g. 1.1.4) can be read in and processed by an older version of SISTEMA (e.g. 1.1.2). A warning message is output in this case. If this loaded SIS-lib is modified and saved, the version number of the library changes to that of the SISTEMA version used (in this case, Version 1.1.2).

Section 7.5 describes the ramifications of the different cases of version differences when libraries are opened.

Upward compatibility is also assured when the database structure is extended: older versions of a SIS-lib are also loaded correctly by the current SISTEMA version, and in general no message is displayed. This is the case even where the database structures differ; in this case, the extended data fields of the newer version are added and populated with intelligent content (generally empty fields). Saving causes the version number of the library to be updated to that of the newer database.

If a SIS-lib, for example of a manufacturer, is write-protected, the version number cannot be changed. The SIS-lib can nevertheless be used and read.

Manufacturers wishing to create a SIS-lib are advised always to use the most recent version of SISTEMA. This makes the latest functions and bug fixes available. The SIS-libs created by the newer version can be used by older SISTEMA versions, provided they share the same database structure. Another strategy is to create the SIS-lib using a recent version, but then to open it with an older version of SISTEMA (with up to two version numbers' difference) and to save it again. This assigns the older version number to the SIS-lib, making it downward compatible without warning messages. The older SISTEMA versions are also available on the Internet for download:

http://www.dguv.de/webcode/e103328

As soon as a change in database structure or content is anticipated, the providers of SIS-libs (i.e. the manufacturers) registered with the IFA are informed and receive a beta version of the update, with which they can convert their SIS-libs. Conversion is a straightforward process. **First, create backups of your non-write-protected, original SIS-libs.** Load a copy of the non-write-protected original SIS-lib in the new SISTEMA version, and save it with new file and library names and with the new version number in the name. This causes the SIS-lib to be converted. Take care not to overwrite the previous SIS-lib. After conversion, you can create a write-protected file from the new SIS-lib (see Section 11.6) and publish this file again (see Chapter 13).

## 11.9  Conventions for the names and file names of libraries

This section describes a proposal for a structured and forward-thinking convention for the name and file name of the SIS-lib. Other conventions are possible; the IFA does not formulate binding rules in this respect. The choice of name must however consider requirements such as ease of comprehension, freedom from contradictions, etc., in order to prevent users from using an incorrect SIS-lib or objects from it for a project.

Many manufacturers and users will typically create multiple SIS-libs, as a function of the total number of their products/objects. Larger numbers of products/objects (over approx. 200) should be divided intelligently between several SIS-libs, in order to limit the size of the individual SIS-libs and therefore the time required to load them. The strategy for distribution of the objects between SIS-libs is not subject to constraints. Different language versions of the objects for example are generally distributed between multiple SIS-libs (one SIS-lib for

English, one for German, etc.). The SIS-libs can be further divided up by product group (one SIS-lib for sensors, one for actuators, etc.). In the course of time, multiple SISTEMA database versions of the SIS-lib will also exist in parallel (for example for Version 1.1.x, at some point in the future for 2.0.x, etc.). The providers of the SIS-libs will also have to manage multiple successive versions of their SIS-libs (1st version in 2013, 2nd version in 2014, etc.).

Distinction must be drawn between three name types:

- File name of the SIS-lib (can be modified, even when the SIS-lib itself is write-protected)

- Name of the SIS-lib: this is displayed in the list of libraries, in the properties window and in the status bar (cannot be changed when the SIS-lib is write-protected)

- Names of the objects in a SIS-lib (cannot be changed when the SIS-lib is write-protected)

There is no reason why the file name (without extension) and the name of the library should not be the same. In response to enquiries from manufacturers, the IFA recommends that the file names/names of manufacturers' libraries observe the following convention:

*<company name>_<product group>_<version number issued by the manufacturer, or statement of the year/month>_<SISTEMA version>_<language code>{.slb}*

For example for file names: *company_sensors_2013-11_V1.1.4_EN.slb*

The corresponding name: *company_sensors_2013-11_V1.1.4_EN*

Alternatively, for a small number of products, for a single SIS-lib and with a different version numbering system:

Example: *company_001_V1.1.4_EN.slb*

Comments on the components of the name:

- Name of the company: the most important component in the name, since the products can be pre-sorted in the folder by name of manufacturer. The name of the company is also important in the SIS-lib name when several SIS-libs are loaded in the list of libraries.

- Product group: where the number of objects is small (fewer than approximately 200), this component of the name can be omitted. In this case, the manufacturer will create only one SIS-lib per language and version. The product group can otherwise be geared to the manufacturer's preferences and product range. This component of the name then becomes the most important criterion for selection of a SIS-lib by the users.

- Manufacturer's version number: this differentiates between several different successive SIS-libs of the same product group. Including the year and month in the form YYYY-MM in the version number is advantageous, as this enables the SIS-libs to be sorted by date in the folder. Some manufacturers base their version numbering upon the SISTEMA version; the latter is however not incremented regularly.

- SISTEMA version: the version with which the SIS-lib was created. This version number is also displayed under the properties of the SIS-lib. Details can be found in Section 11.8.

- Language code: the Alpha 2 code of ISO 639-1[7] is proposed for this purpose. This is an international standard which defines codes for the names of the major living languages. The code uses two letters. EN stands for the English version, i.e. texts in English. The

---

[7] ISO 639-1: Codes for the representation of names of languages – Part 1: Alpha 2 code

majority of SIS-libs are still supplied with text in English only. Further information on the language codes can be found at:
http://www.iso.org/iso/home/standards/language_codes.htm.

This naming convention facilitates sorting of the SIS-libs, which is important during use, since a large number of files from multiple manufacturers generally have to be managed over the years. The various different naming conventions used by other providers of SIS-libs can be studied, such as those available in the IFA's list of links.

## 11.10  Further library properties

The library properties (Figure 53) of a SIS-lib serve as its profile. Default values are assigned when a new SIS-lib is created and some of them can be modified, provided the library has not been write-protected.

Figure 53: Library properties pane

| Name | New User Library |
|---|---|
| Location | C:\Users\huelke.michael\Documents\SISTEMA205\Libraries\New Library.slb |
| Author | SISTEMA |
| Last change | 02.09.2016 13:36:19 |
| Library version | 10.2 (User versioning) |
| Version of Standard | ISO 13849-1:2015, ISO 13849-2:2012 |
| SISTEMA version | 2.0.5b |
| Information | Detailed version history |

Table 5 characterizes the library properties.

| Property name | Meaning | Change possible in use? | Number of editable characters |
|---|---|---|---|
| Name | Name of the library | The file name (without path and extension) is specified when the library is created. | 512 |
| Location | Location at which the library is stored (path, file name with ".slb" extension) | No; is determined only by SISTEMA (is not stored in the library). | – |
| Author | Author of the library | The Windows user name is entered at creation. It can be changed. | 1024 |
| Last change | Timestamp of the last change | No. The value is updated only by SISTEMA, after each change. | – |
| Library version | Versioning of the library | Yes. The field is initially | 10 |

| Property name | Meaning | Change possible in use? | Number of editable characters |
|---|---|---|---|
| | by the user | empty. | |
| Version of the standard | Version of the standard with which this library was created | No | – |
| SISTEMA version | Software version with which this library was created | No | – |
| Information | Information on the library | Yes. The field is initially empty. | 2000 |

Table 5: Information on the library properties

For technical reasons, the number of editable characters has been reduced in SISTEMA Version 2 compared to versions 1.1.x. When existing SIS-libs are converted, attention is drawn to a possible shortening of textual content. The texts must be adjusted prior to conversion in this case.

The file name and path can be modified, for example in Windows Explorer, provided the SIS-lib is not loaded in the list. Modification is not possible within SISTEMA. The library window does not offer a function corresponding to the "Save As..." function in the main window.

To edit the properties in the display line directly, click on them with the mouse. Should you wish to edit the "Information" property using multiple lines, clicking in the line causes a button with three dots to appear on the right. Clicking on this button opens a small editor in a dedicated window (Figure 54).

Figure 54: Editor for the information line of the library
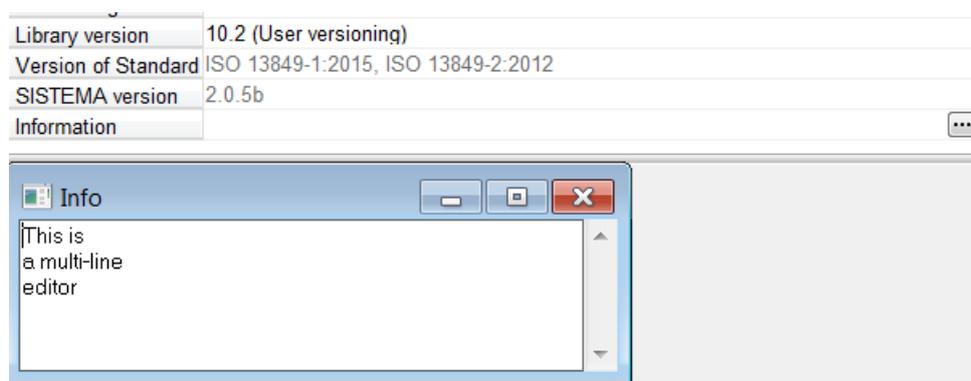


## 11.11 Multilingual libraries

Alternative text data fields (e.g. for names, documentation or a document) cannot be saved in SISTEMA objects; the creation of multilingual objects is therefore also not possible. Multiple languages for a product with a single set of parameters require multiple objects. Three strategies are possible here:

- These objects (e.g. three subsystems) could be stored together for multiple languages (e.g. English, German, French) in a single SIS-lib. Each object would contain text in one language only. In this case, the object name (for the subsystem) would then include the corresponding language code, enabling it to be selected more quickly.

- In use, it is doubtless more practicable for the relevant SIS-lib to be identified for the user's language and for all objects within it then to be used in the user's language. In this case, each object contains text only in the selected language.

- The text fields of an object contain all available languages (for example: text in English at the top, text in German below it). Non-language-specific names must then be used. This strategy is practical only when the number of languages used is small. If that is the case, it is however very practical, since when the SISTEMA user interface is switched to a different language and a summary is created, all texts in the "other" language are available.

Refer to Section 11.9 regarding the language codes.

# 12    Automatic generation of SISTEMA libraries

We offer manufacturers of components the facility to create their own SISTEMA libraries. The limits of the integrated library editor are however rapidly reached when the number of components is large. This chapter provides further information to manufacturers who are contemplating creating SIS-libs automatically.

## 12.1  Reference to the IFA's documentation for developers

The documentation for developers, which assists in the generation of SISTEMA libraries, is available free of charge upon request from the IFA. Please contact Dr Michael Huelke at sistema@dguv.de in this case.

Users of the documentation for developers require knowledge of SQL, XML, XSD and the use of SISTEMA. Owing to time constraints, the IFA is able to provide only very limited consulting and support beyond the documentation for developers.

The database's SQL interface is the primary option for the creation of libraries. Since the complex relationships between the data objects cannot all be defined by SQL constraints, it is to some extent the responsibility of the user of the interface to create valid data that SISTEMA is able to process without error.

Alternatively, the data can be imported from an XML file in the **SISTEMA format**. A hidden import function in the library editor enables library content in the form of an XML file to be imported. **Important: an XML file in the VDMA Specification 66413 format cannot be loaded into the SISTEMA library editor. The separate VDMA library window (Chapter** 5**) is used for this purpose from SISTEMA Version 2 onwards.**

We recommend that preference be given to SQL interface rather than the XML interface, since its stricter constraints make it less prone to error, and debugging is less problematic.

## 12.2  SQL interface

SISTEMA stores libraries in the file system in the form of files with the "*.slb" extension. This file format is used with a Firebird database.[8] The Firebird database can be addressed and edited by means of suitable tools employing the SQL standard (to ANSI SQL-99). SQL scripts enable generic library content to be created and modified. Use of this interface requires SQL expertise. The database's SQL interface is described in greater detail in the documentation for developers. It enables you to use SQL commands to populate a database created by SISTEMA. The scripts used in SISTEMA for creation of a new library are provided with the documentation for developers in the form of example files.

The annex of the documentation for developers also contains a sample SQL script that demonstrates the generation of data through this interface (when the examples are installed). Some SQL scripting tools enable SQL command sequences to be generated from existing database content. Should you be uncertain, we recommend that you begin by generating the desired library content using the integrated library editor, and then export the data to an SQL script by means of such a tool. This enables you to recognize easily the required form to be taken by the content.

---

[8] SISTEMA uses the embedded version of Firebird, which does not require installation of the server. Firebird databases have the file extension "*.fdb" as standard. SISTEMA uses the extension "*.slb" instead.

## 12.3  XML interface (nur für das SISTEMA-Format)

SISTEMA's integral library editor possesses a hidden interface for the import and export of XML files, but only in the SISTEMA format. Use of this interface requires XML expertise. The interface must first be enabled by the setting of a Windows registry key. This procedure activates two additional commands in the library's "Edit" menu: "Import..." and "Export...". The "Import..." command causes all content from the XML file to be added to the library currently loaded. Existing library entries are retained. The "Export..." command saves the entire library to an XML file. This facilitates the compilation of libraries.

The content of such an XML file can of course also be edited or created manually. Note in this case that the relational table structure of the SQL database is modelled by the XML structure. This also enables very large libraries to be divided into several small libraries.

SISTEMA checks the validity of the XML file (SSM projects and XML imports) against a rudimentary XML schema. This schema has the file name ssm.xsd and can be found in the SISTEMA program folder. Should an error be detected with reference to the schema when the file is parsed, the reason for the error and the line containing it are displayed. However, the schema detects only major errors in the data structure. Out-of-range errors of individual fields for example are not detected. These and other errors not detected during parsing are generally detected by the program logic when the data are being imported.

This XML interface to SISTEMA and enabling of it are described in greater detail in the documentation for developers. A further XML interface is available from SISTEMA Version 2 onwards for the "universal database" in accordance with VDMA Specification 66413 (Annex B). This will however support only the importing of data.

## 12.4  Microsoft Windows clipboard

SISTEMA uses the functions of the Microsoft Windows clipboard in order to copy objects. A dedicated clipboard format has been defined for this purpose. Other applications do not therefore generally respond to this content. Interprocess communication with other applications is however conceivable and could be developed. Information on the clipboard format will be provided by the IFA upon request.

The current content of the clipboard for a valid SISTEMA object is displayed in the status bar (Section 4.2.9). As soon as another application uses the clipboard, this display is cleared.

# 13    Publication of SISTEMA manufacturers' libraries

This chapter provides manufacturers of control products with information on creating SISTEMA libraries and publishing them for their customers. The IFA provides information on these manufacturers' libraries on a dedicated web page. The XML files in the VDMA Specification 66413 format are listed by the VDMA (see Annex B).

## 13.1  Terms of reference

Firstly, thank you for your willingness to make SISTEMA libraries available to your customers. This constitutes a major contribution to efficient, error-reducing analysis of safety-related control systems on machinery in accordance with EN ISO 13849-1.

The creation and distribution of the libraries remains the sole responsibility of the manufacturers concerned. The IFA places no particular requirements upon the libraries and does not inspect them. At the same time, it charges no fees to either suppliers or users of the data. Should you have a need for consent to create SISTEMA libraries for your products, we will be happy to grant it formally in response to an e-mail request (sistema@dguv.de).

## 13.2  Procedure for creating and publishing manufacturers' libraries

How are SISTEMA libraries created? **To begin with, you can create libraries directly in SISTEMA: create a new library, enter the data for each product, then write-protect the library (see Chapters 10 and 11).**

Examples are already available from many other manufacturers who are publishing SIS-libs. Besides the parameters, you can also attach documents and internet links to your online documents directly to the library.

Should you wish to export your data from a database automatically in the form of a SIS-lib, please request the database documentation from us with which you can develop an export tool (Chapter 12). This is generally worthwhile only for very large volumes of data, or when the library data are subject to frequent change.

Besides the organizational requirements for creating the SIS-lib, certain technical requirements must also be met. These include:

• Knowledge of the EN ISO 13849-1 and -2 control standards

• Availability of the required product parameters

• Experience in the use of SISTEMA

Creation of manufacturers' libraries for the first time often raises the following questions (Q), more comprehensive answers (A) to which can be found in the various sections of this cookbook:

• Q: What parameters and information on the products are required?
  A: See Chapter 10

• Q: What languages should be supported?
  A: In order to assure that the manufacturers' libraries are accessible to an international user base, at least an English version of the library should be created (Section 11.11).

• Q: What SISTEMA version should be used for creation of the library?
  A: The very latest version of SISTEMA should not be used for publication of the libraries,

owing to compatibility with older SISTEMA versions which may still be installed on users' systems (see Section 11.8).

- Q: What measures must be taken when new SISTEMA versions are released? Do the libraries need to be adapted?
  A: This is not necessary for each change in the final version number digit. However, if the database definitions have changed (leading or middle version number higher), the libraries should at least be converted with the new SISTEMA version (see Section 11.8). The manufacturers registered with the IFA (see below) receive advance information and access to a beta version, enabling them to make preparations in good time. For SISTEMA Version 2, the SIS-lib must either be recreated with this SISTEMA version, or have been converted from older versions.

- Q: What are the preferred naming conventions for file and library names?
  A: See Section 11.9

- Q: How many products can/should a library contain?
  A: Where the number of products is high (over approximately 200), it is urgently recommended that they be distributed between a number of libraries, for example by product group. Very large libraries have long loading times. From SISTEMA Version 2 onwards, search, sort and filter functions can be used to find products.

The contact for manufacturers is Dr Michael Huelke, who can be reached at: sistema@dguv.de. Please also let us know the names of individuals to be included in our e-mail mailing list for manufacturers publishing SIS-libs. We may occasionally send special information bulletins (for example concerning beta versions and changes to databases).

## 13.3  Legal information

**Copyright**: The intellectual property rights to the use and exploitation of SISTEMA libraries are normally held by the producer of the library concerned.

**Right of use**: The IFA grants third parties a non-exclusive right worldwide and in perpetuity to use the SISTEMA technology (such as the database format) solely for the creation of SISTEMA libraries and provision of public access to them. Any other use or exploitation, particularly modification, is subject to prior approval by the IFA (such as the importing and converting of SISTEMA libraries or the importing and exporting of SISTEMA project files).

**Any party using the SISTEMA technology to create SISTEMA libraries must state when making the libraries public, and particularly when providing public access to them on the Internet, that the libraries have been tested by the creator only for SISTEMA and may therefore be used only by SISTEMA.**

**Use of SISTEMA**: The legal information for the installed SISTEMA version and documentation apply. This information is displayed on the download page for SISTEMA, during the installation process, and subsequently in the SISTEMA readme file (which can be found in the installation folder).

**Exclusion of liability**

The software was developed with care and in line with current good scientific and industrial practice. It is made available to the user free of charge. The liability of the IFA/DGUV is thus limited to wrongful intent and gross negligence (Section 521 of the German Civil Code), and in the case of technical and legal faults, to the fraudulent concealment of faults (Sections 523 and 524 of the German Civil Code).

The IFA endeavours to keep its web site free of viruses. No guarantee can however be given that the software and information provided are virus-free. Users are therefore advised to take appropriate precautions of their own and to scan for viruses before downloading software, documentation or information.

**Licensing information**

SISTEMA is free software and may be copied and distributed freely. Modification of SISTEMA is not permitted. Please note however that SISTEMA makes use of further open-source software the use of which is covered by licenses of its own. Modified versions of these software components may be distributed only under the terms of the respective licences. The source code covered by the respective licences has been adopted without modification.

The "ZeosLib" library and the source code of the version used are also supplied with SISTEMA (refer to the "Standards/ZeosLib" sub-folder). Since the library is integrated dynamically (at runtime) in the form of the compiled bpl files, any user is free under the terms of the GNU LGPL to use a modified version of the library.

Copies of the respective licences can be found in the "Licences" sub-folder of the application.

## 13.4  Verifying/testing SISTEMA manufacturers' libraries

The creation and distribution of the libraries remains the sole responsibility of the manufacturers concerned. This also includes verification of the library data. Check that the data are also processed correctly with older SISTEMA versions.

The IFA imposes no particular requirements upon the libraries, nor does it inspect them. This also applies to the libraries to which links are provided on the IFA web pages (Section 13.6).

## 13.5  Publication by the manufacturer

The manufacturers' SIS-libs are generally hosted on the manufacturers' websites for download. Users may be required to register before downloading. Many manufacturers also provide a dedicated "SISTEMA page", containing a brief presentation of the program together with the SIS-libs that they offer. We will be happy to provide a screenshot of the SISTEMA user interface, should you require one for example for your download page. No SISTEMA logo is available, nor should such a logo be invented.

We would be pleased if you were to refer your customers who may not yet have installed SISTEMA themselves to our download pages by means of a link:

Link to the English pages: http://www.dguv.de/webcode/e34183

Link to the German pages: http://www.dguv.de/webcode/d11223

Hosting SISTEMA installation files on manufacturers' servers is however not permitted and would also not be constructive, given that updates are released regularly.

**Manufacturers should use their information channels to their customers in order to draw the latter's attention to new or modified SISTEMA libraries. Changes relevant to safety may have been made that must be considered during the program's use for analysis.**

## 13.6 Inclusion of SISTEMA manufacturers' libraries on IFA lists

We would like to refer visitors to our SISTEMA web pages to your libraries.

http://www.dguv.de/webcode/e92603 (English)

http://www.dguv.de/webcode/d92599 (German)

The company names of all providers are arranged alphabetically on these pages, in each case with a link to the download area of the SIS-libs on the manufacturers' websites. Inclusion is of course free of charge. Should you also wish to be included in this list, please send your links, the company name to be published, and of course your consent for their publication, to: sistema@dguv.de. Please also let us know the names of individuals to be included in our e-mail mailing list for manufacturers, for the event of enquiries concerning the links.

The links should point directly to the SIS-libs. Please do not therefore send global links, for example to your company's home page. The pages in question should already be live when you send us the links; we are keen to avoid publishing dead links. The IFA regularly scans the links in order to detect broken links quickly.

## 13.7 Amending list entries

Should changes be made to the links or the entries, it is sufficient to inform us by sending a brief e-mail to sistema@dguv.de. The web pages will then be updated immediately.

# 14 Annexes

## Annex A    Version history

This annex documents the changes in this cookbook that have resulted from further development of SISTEMA. The first version (1.0) of the cookbook is based upon SISTEMA Version 1.1.6, released in July 2013.

The following changes were made in Cookbook Version 2.0 (September 2016) for SISTEMA Versions 2.0.x:

- New Chapter 5: **User interface for VDMA libraries**

- New Chapter 8: **Working with VDMA libraries**

- Annex B: **VDMA Specification 66413** has been updated

- All screenshots have been updated with SISTEMA Version 2.0.5

- All chapters and annexes have been revised taking account of the functional changes in SISTEMA

- The abbreviations for the reliability values have been brought into line with EN ISO 13849-1:2016

- The distinction between libraries in the SISTEMA and VDMA Specification 66413 formats has been made clear in the text

## Annex B   VDMA Specification 66413

In July 2012, VDMA Specification 66413 with the title "Functional Safety – Universal data format for safety-related values of components or parts of control systems" was adopted. This document describes a universal database in XML format serving as a common basis for the exchange of parameters and information between machine manufacturers, equipment manufacturers, test bodies and suppliers of analysis tools in the area of functional safety. Representatives of manufacturers and users of safety-related control components and the IFA were involved in development of this VDMA Specification.

Further information (in German) and a list of manufacturers can be found on the VDMA website (www.vdma.org) under the search term "Einheitsblatt 66413".

In Version 2, the IFA has extended the SISTEMA application with the addition of an import interface for this universal database format. The interface is implemented as a new SISTEMA window. It enables an XML file to be opened in the window and a device to be selected with a set of values in the desired language. For further development of new data fields of the universal database, the existing SISTEMA database, the SISTEMA libraries and the input dialogs have been extended. The SISTEMA libraries for users and manufacturers will continue to be supported and the functionality of the SISTEMA library window to be developed further. A tool for the conversion of SISTEMA libraries to a corresponding XML file and an editor for the creation of XML files are not planned, however. Analysis of the XML data to EN 62061 is not implemented.

## Annex C   Index of abbreviations/equation symbols

| Type of parameter | Abbre-viation | Unit | Original term (EN ISO 13849-1) |
|---|---|---|---|
| Safety-related part of a control system | SRP/CS | – | **S**afety-**R**elated **P**art of a **C**ontrol **S**ystem |
| Component quality | $MTTF_D$ | **Year, a** | **M**ean **T**ime **t**o **d**angerous **F**ailure |
| Test quality (subsystem) | $DC_{avg}$ | **%** | **av**era**g**e **d**iagnostic **c**overage |
| Test quality (block, element) | DC | **%** | **D**iagnostic **C**overage |
| Simultaneous failure of redundant channels | CCF | – | **C**ommon-cause **f**ailure |
| Probability of failure | $PFH_D$ | **1/h** | **P**robability of a **d**angerous **F**ailure per **H**our |
| Specified value of the functional safety | $PL_r$ | – | **R**equired **P**erformance **L**evel |
| Actual value of the functional safety | PL | – | **P**erformance **L**evel |
| Category | Cat. | – | **Cat**egory |
| Duration in service | $T_M$ | **Year, a** | **M**ission **T**ime |
| Component quality (subject to wear) | $B_{10D}$ | **Cycles** | Number of cycles until **10**% of the components fail **d**angerously |
| Permissible duration in service (subject to wear) | $T_{10D}$ | **Year, a** | Mean **T**ime until **10**% of the components fail **d**angerously |
| Number of operations | $n_{op}$ | **Cycles/a** | **N**umber of **op**erations (average number of operations per year) |
| Ratio of dangerous failures | RDF | **%** | **R**atio of **D**angerous **F**ailures, see VDMA Specification 66413 |

# Annex D    List of examples: modelling of components as object types

This list, which is not exhaustive, shows how different typical components can generally be modelled as SISTEMA objects. Deviations may occur with certain technologies when the manufacturers state different parameters. For example: hydraulic components can be characterized by the $MTTF_D$ or by the $B_{10D}$. In the first case, the component can be modelled as a block or element, in the second only as an element (with analysis of the $B_{10D}$).

In the event of doubt, the manufacturer must be consulted.

| Component | Usual parameter for the reliability | Encapsulated subsystem (SB), block (BL), element (EL) or test equipment (TE) |
|---|---|---|
| Mechanical components | $PFH_D$/$MTTF_D$/$B_{10D}$ | SB/BL/EL |
| Hydraulic components (valves) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Pneumatic components (valves) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Pressure switches (hydraulic/pneumatic) | $MTTF_D$ or $B_{10D}$ | TE |
| Relays and contactor relays | $B_{10D}$ | BL/EL |
| Proximity switches (safety component) | $PFH_D$ | SB |
| Proximity switches (standard component) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Contactors | $B_{10D}$ | BL/EL |
| Position switches (positive-opening contact) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Position switches (make contact) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Position switches (with separate actuator, guard locking device) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Pushbuttons (positive-opening contact) | $B_{10D}$ | BL/EL |
| Pushbuttons (make contact) | $B_{10D}$ | BL/EL |
| Emergency-stop devices | $PFH_D$ or $B_{10D}$ | SB/BL/EL |
| Enabling switches | $PFH_D$ or $B_{10D}$ | SB/BL/EL |

| Component | Usual parameter for the reliability | Encapsulated subsystem (SB), block (BL), element (EL) or test equipment (TE) |
|---|---|---|
| Electro-sensitive protective equipment (safety components: laser scanner, light barrier, pressure-sensitive edge, etc.) | $PFH_D$ | SB |
| Rotary encoders (sin/cos) (safety component) | $PFH_D$ | SB |
| Rotary encoders (standard) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Distance measurement system (standard) | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Electromechanical safety modules, e.g. PNOZ | $PFH_D$ | SB |
| Safety PLCs | $PFH_D$ | SB |
| Standard PLCs | $MTTF_D$ or $B_{10D}$ | BL/EL |
| Microcontrollers | $MTTF_D$ | BL/EL |
| Optocouplers | $MTTF_D$ | BL/EL |
| Frequency converters with integral safety functions | $PFH_D$ | SB |
| Standard frequency converters (e.g. with servo disable) | $MTTF_D$ or $B_{10D}$ | BL/EL |