FAQ COMBIVIS studio 6

KEB

Project-Handling and versioning FAQ No.0002

Part	Version	Revision	Date	Status
en	6.2.3.0	003	2019-01-01	Released

Content

FAQ COMBIVIS studio 6

Introduction

This document describes general recommendations regarding project creation, layout and version maintenance with COMBIVIS studio 6.

COMBIVIS studio 6 is an automation platform. By using many different target systems, fieldbus devices, communication interfaces and topologies COMBIVIS studio 6 makes application development possible. It supports its user with pre-developed software modules and project templates.

COMBIVIS studio 6 offers an open and flexible device and library database. The user can expand the preinstalled range of functions individually by adding further fieldbus devices and software components or by using the integrated version management.

COMBIVIS studio 6 projects that have been verified during their initial operation and test phase are internally exactly defined. By archiving a project the user is able to freeze the current state and restore the project at any time.

Individual components can be changed with older or newer versions as needed.

Fundamentals

Versioning

Components are being identified by a 4-digit version code

<major version>.<specification version>.<release version>.<implementation version>

Major version	Introducing a new generation
Specification version	Introducing new specifications and concepts
Release version	Introducing new functions
Implementation version	Hotfixes/ Bug fixes

Among others the following main components pursue the above pattern:

Component	Example
COMBIVIS 6	6.2.3.0
Compiler (CODESYS CORE)	3.5.4.10
Project	1.0.0.25
Devices (PLCs, Fieldbus-Master/slaves),	3.5.4.10
Library	3.5.4.11

Updates

On a regular basis KEB delivers software and device updates together with expansions that can be downloaded via the integrated online update function or manually via www.keb.de.
Older versions remain on the programming system.

A new COMBIVIS studio 6 Setup installs a coherent set of versions, meaning compiler, devices and libraries are well-matched and verified.

Outdated device versions are not being installed to ensure a clear and performant device selection as well as the development with an optimal version set.

Thanks to the backwards compatibility an update of individual components is possible.
However, some aspects must be kept in mind:

Implementation Version [_._._.d]

Contains small bug fixes. The risk of incompatibility is very low.

Release Version [_._.c._]

New functions are being released. These functions are only usable with the at release active compiler version and in some circumstances effective across multiple devices.
For the use of new functions, it is necessary to project wise update the compiler, and every object, that regards this function, to its according version.

Specification Version [_.b._._]

New specifications or concepts are being released. "Mixed"-Projects with different specifications may cause compiler errors or lead to unexpected behavior!

For example: 3.3 PLC + 3.4 ETC Master + 3.5 EtherCAT Library

It is recommended to update compiler, devices and library simultaneously.

Major Version [a._._._]

A new software generation is being released. In some circumstances file formats are not compatible and have to be exchanged via the Import/ Export function.
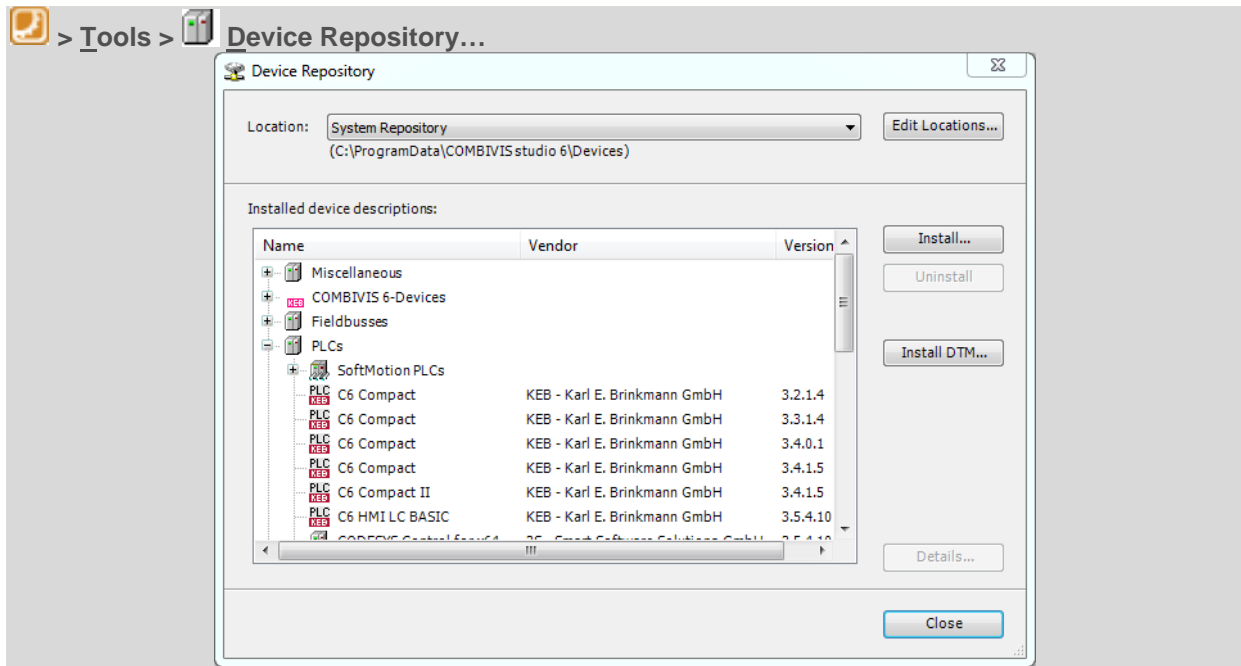
Repositories

Device-Repository

The Device-Repository is the local device database located on the user- pc.

> Tools > Device Repository…

All local installed devices and their versions can be managed here.

> Details…

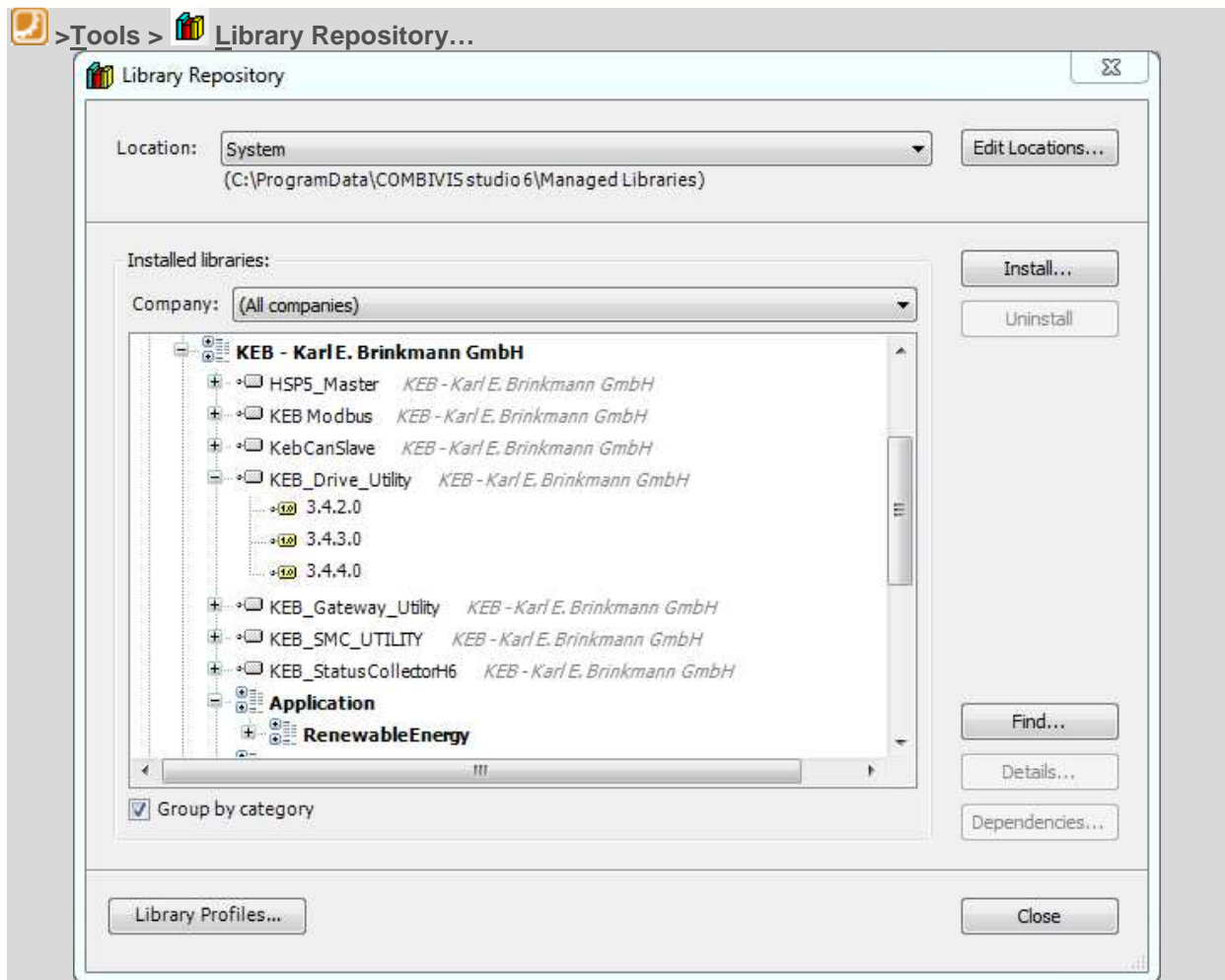Furthermore, it is possible to install additional devices:

> Install…

Library-Repository

The Library-Repository is local library database located on the user- pc.

>Tools > Library Repository…

All local installed libraries and their versions can be managed here.

> **Details…**

Furthermore, it is possible to install additional libraries:

> **Install…**

Library management

A library is a self-contained collection of POUs (Program Organization Unit), that can be used to enlarge the range of functions of a project or a superior library.

Advantages of libraries

- ✓ Source code protection
- ✓ Reusability
- ✓ Reproducibility through versioning
- ✓ Easy usability and integrated documentation

The available set of libraries is defined by the Library-Repository.

The available set of functions of each project is set by the Library Manager and the added libraries within.

Some system- or hardware specific libraries are added automatically (shaded in light grey)

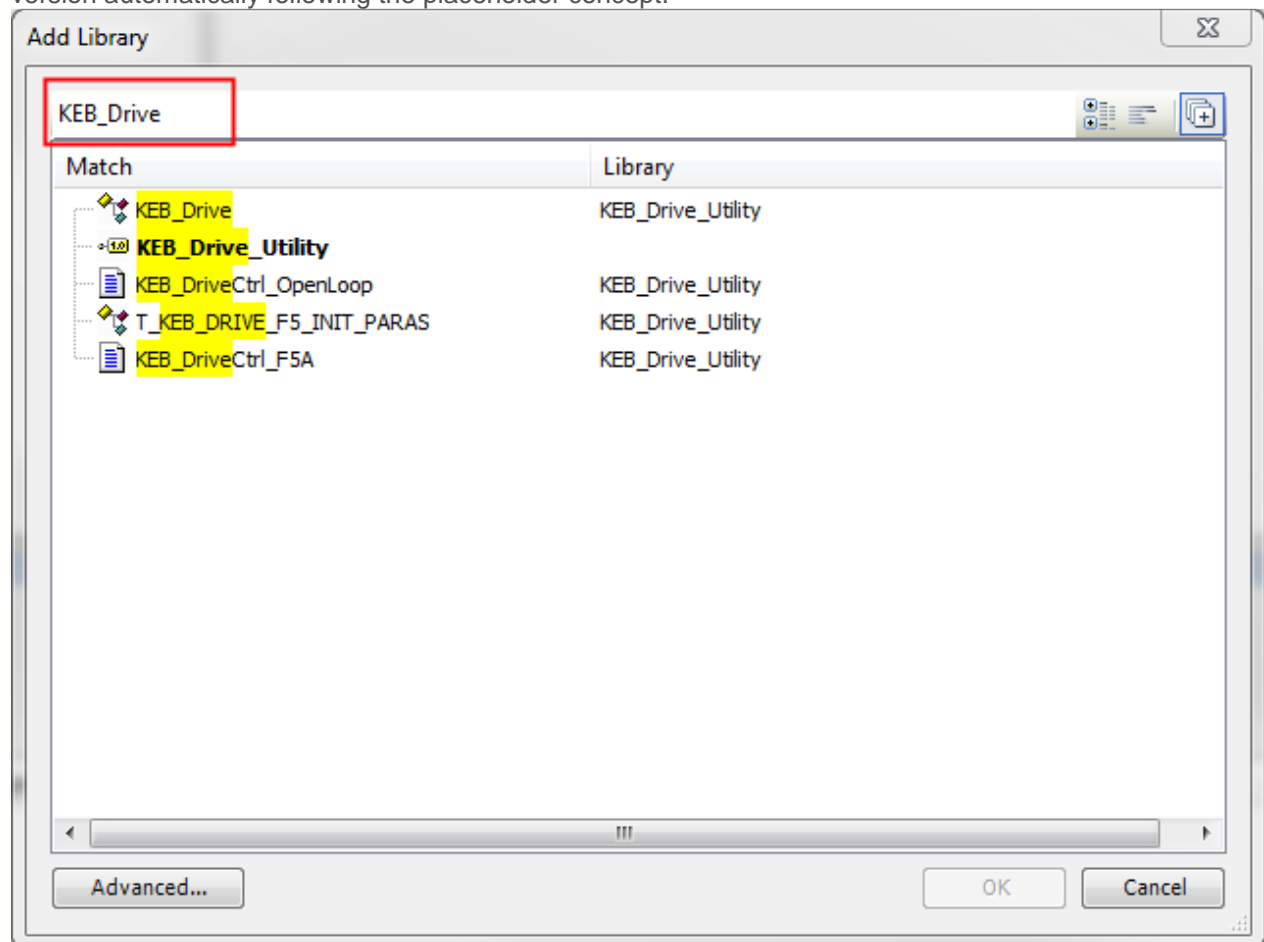**For example: IoDrvEtherCat-Library, while using an EtherCAT Master**

Add Library

Additional libraries can be added by the user.

> 📙 > 🏛 **Library Manager > Add Library**

The Add Library Assistant lists all POUs and libraries with a specific keyword and adds the correct version automatically following the placeholder concept.

Placeholder Concept

The library injection has been simplified since COMBIVIS studio 6 V3.5.4.10.

All libraries are being added via placeholder by default.
The manufacturer defines in the background, which library version is optimal for which target system and the set compiler version. The library version is being applied automatically.

If necessary, the placeholder version can be set to a different version manually.

FAQ COMBIVIS studio 6

Project Environment

The Project Environment dialogue checks the used versions in the project and informs the user about updates.

> Project > Project Environment…
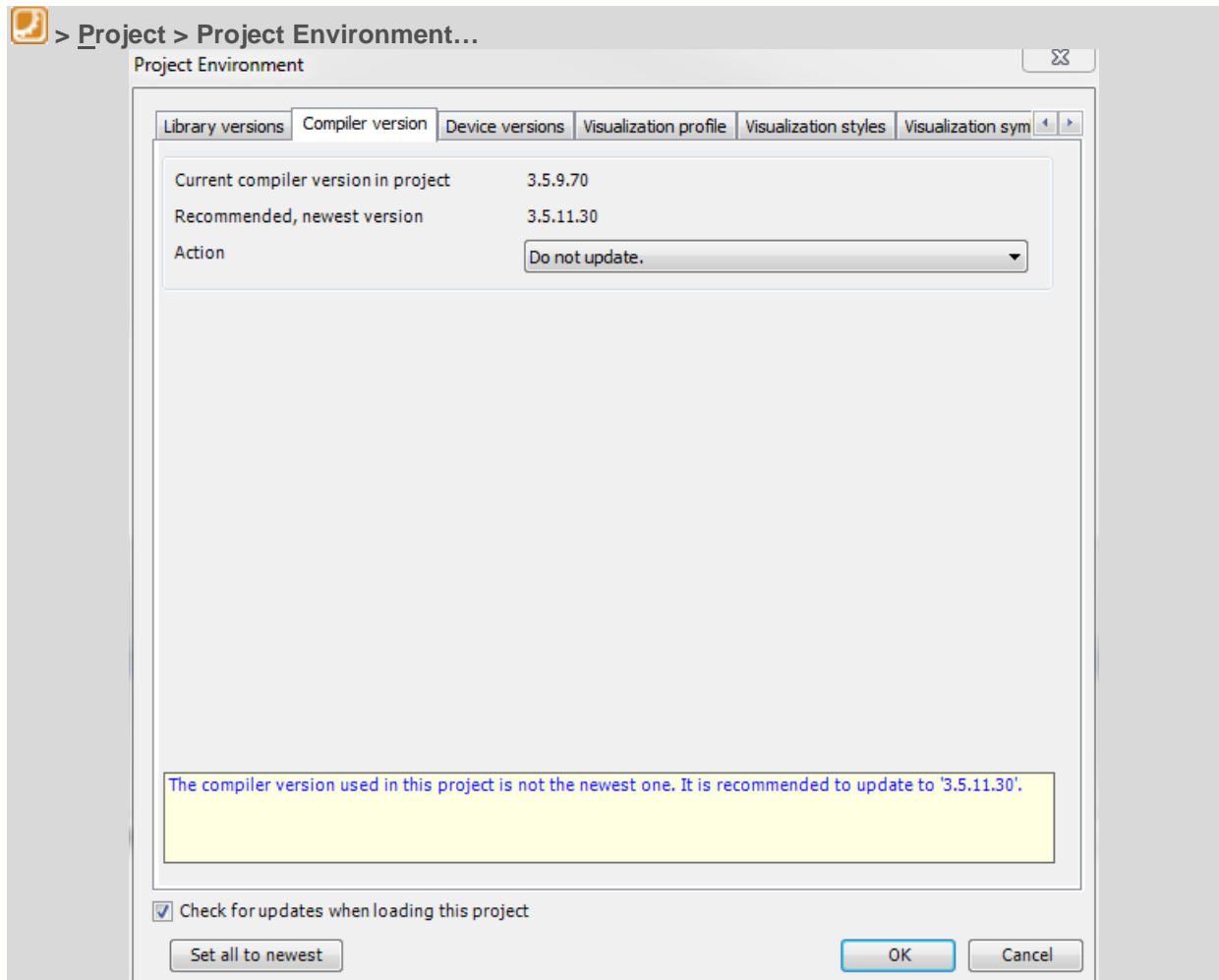
If a project has to be supplemented by new functions, the dialogue offers the possibility to update all objects at once.

> Set all to newest

If an already verified project, meaning an already on a machine running project has to be changed by uncritical source code lines, or if the project's only purpose is the monitoring of something, it is recommended to cancel the assistant. If preferred, the "Check for updates when loading this project"-function can be bypassed by disabling the Checkbox.
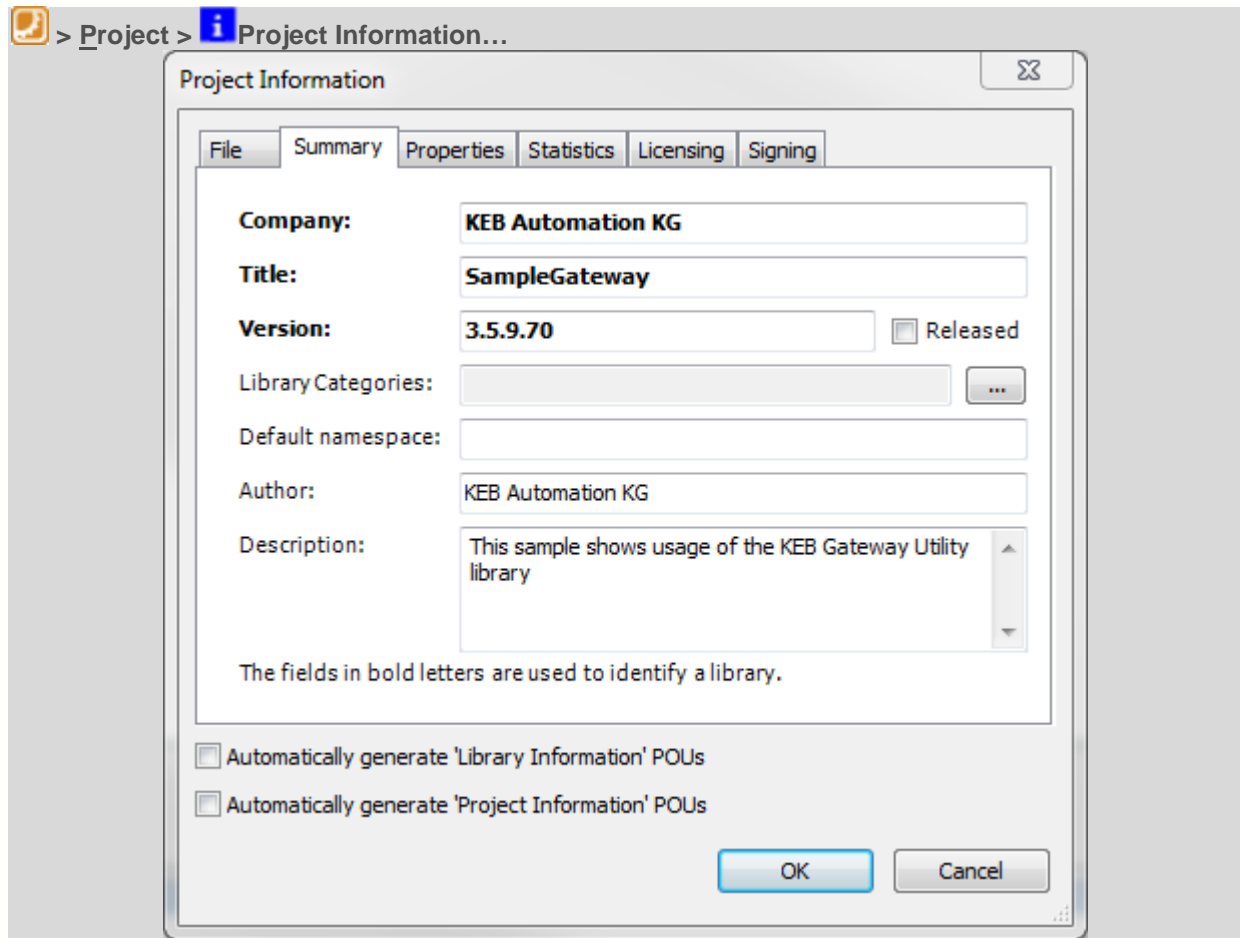
> Cancel
☐ **Check for updates when loading this project**

Project Information

Even projects can be marked by an explicit version number or additional information such as Company, Title, Version, Author and Descriptions.

The Project Information must be activated separately for each project.

> Project > Project Information…

The deposited information can be viewed online. The precise machine version is therefore able to be determined.

It is being recommended to create and maintain the Project Information for each project.

FAQ COMBIVIS studio 6

Recommendations for user

Creating a new project
Recommended procedure for new projects:

- Using the newest COMBIVIS studio version (see also www.keb.de)

> Help > Check for updated…

- Using the automatically generated standard project

> File > New Project > Standard PLC project

- Using sample projects as basis

> File > Open sample project…

Expanding an existing project
Recommended procedure to update projects or old machines:

Case A: Usage of new functions
New functions must be added to a project that requires a compiler, library or device update.
The whole project functionality must be re-verified. The update of all components to an equivalent version is recommended; thereto the Project Environment dialogue is being used.

> Project > Project Environment…

Additionally, the manual update of all fieldbus devices is recommended:
Right click on device > Update Device…

The project should be cleaned after a device update. Eventual precompile information that is no longer in use is being deleted and libraries are being reinitialized.

> Build > Clean All…

Case B: Source code changes
A project must be extended by various operations that solely regard the source code.
For example: Changing values of variables

For example: Adaption of a state machine

In that case, it is recommended to freeze all core components such as compiler, devices and libraries to its already verified state. The Project Environment dialogue that might be shown at each project opening should be canceled.

Project layout/ Programming convention

For a global consistent and optimal project and source code layout, KEB recommends the usage of „KEB Software Engineering Guideline for IEC 61131-3 libraries and applications".
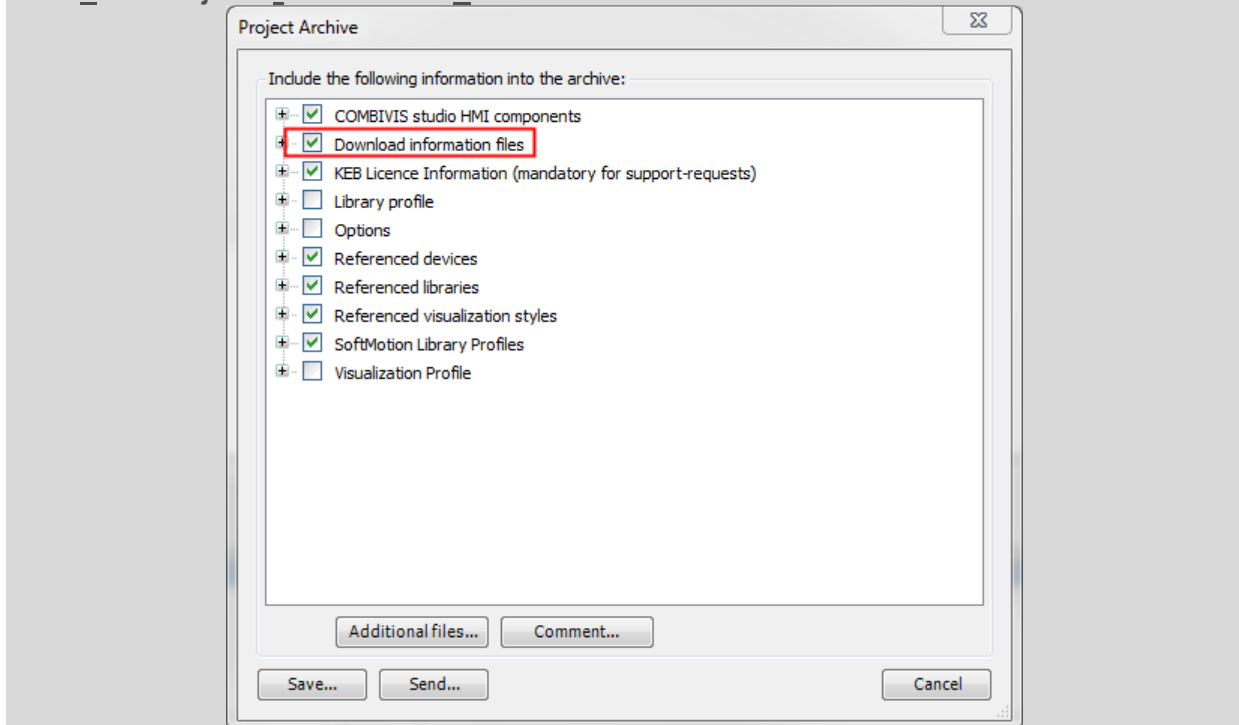
> **Help** > **Show "FAQ" documents…**

Data maintenance / After-Sales-Support

Project Archive

To safe a project entirely the Project Archive creation is necessary.

> **File** > **Project Archive** > **Save/Send Archive…**

A Project Archive consists among others of the following data in one file:
- Project files (IEC Code, POUs, etc.)
- All used library versions!
- All used device versions!

It is highly recommended to create a whole Project Archive after project acceptance and to save it on a failsafe medium. The Project Archive allows a full project restore, independent on the user's repository.

Login without compiling

A project archive should not be created until the project file has been downloaded to the target system or the creation of a boot application on the target system. Only if these steps are being followed the **download information files** can be saved into the archive. Without download information a login on a running machine without recompiling is not possible.

Source code download

The function Source download… enables the possibility to deposit a Project Archive directly onto a target system.

> **File > Source download…**

If this is done consistently, the correct project source code is directly stored on the machine and in case of service available.

This option can be enabled within the project for simplification purposes.

> **Project > Project Settings…> Source Download**
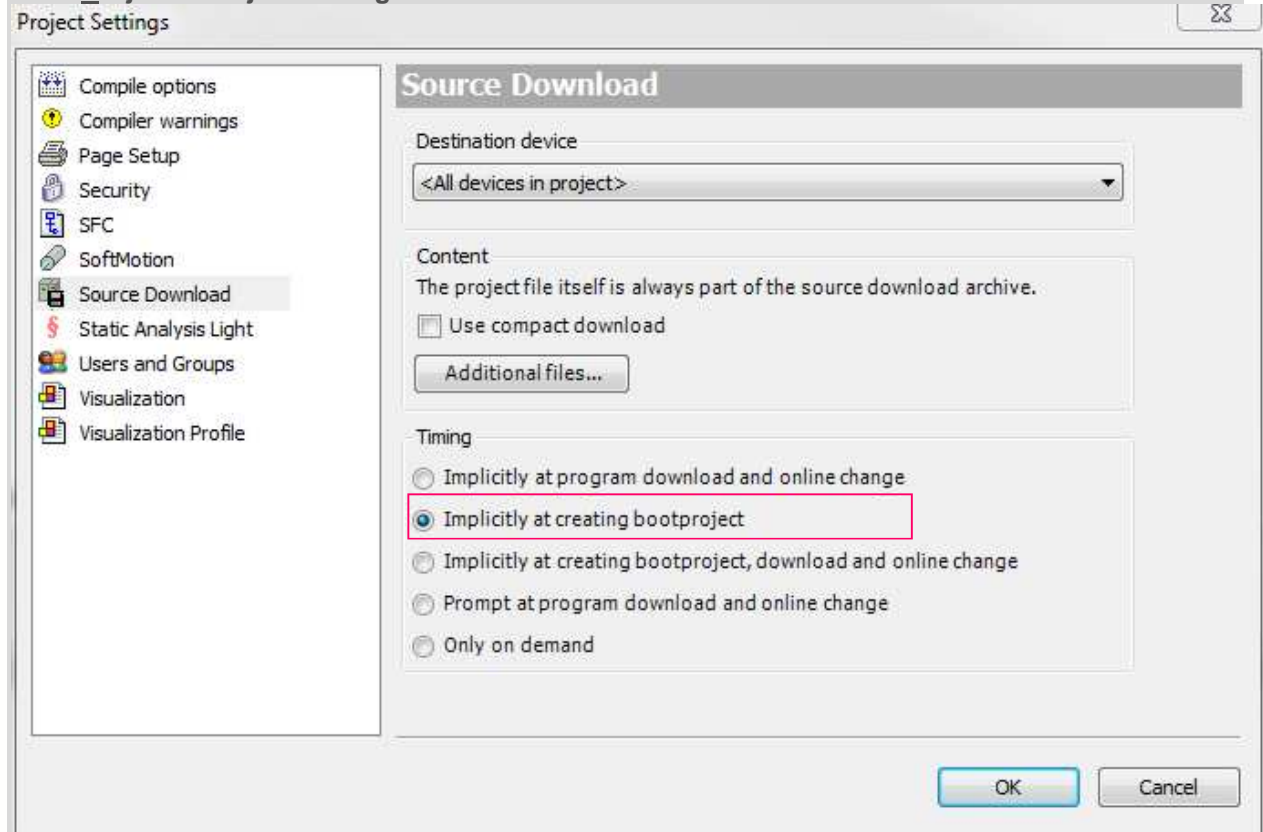
Disclaimer

KEB Automation KG reserves the right to change/adapt specifications and technical data without prior notification. The safety and warning reference specified in this manual is not exhaustive. Although the manual and the information contained in it is made with care, KEB does not accept responsibility for misprint or other errors or resulting damages. The marks and product names are trademarks or registered trademarks of the respective title owners.

The information contained in the technical documentation, as well as any user-specific advice in verbal or in written form are made to the best of our knowledge and information about the application. However, they are considered for information only without responsibility. This also applies to any violation of industrial property rights of a third-party.

Inspection of our units in view of their suitability for the intended use must be done generally by the user. Inspections are particular necessary, if changes are executed, which serve for the further development or adaption of our products to the applications (hardware, software or download lists). Inspections must be repeated completely, even if only parts of hardware, software or download lists are modified.

Application and use of our units in the target products is outside of our control and therefore lies exclusively in the area of responsibility of the user.

KEB Automation KG
Südstraße 38 • D-32683 Barntrup
fon: +49 5263 401-0 • fax: +49 5263 401-116
net: www.keb.de • mail: info@keb.de