



Platzhalter Mechanismus

FAQ Nr.0007

Part	Version	Revision	Datum	Status
de	6.2.3.0	001	2019-01-01	Released

Inhalt

Einleitung	2
Vorgeschichte	2
Methode ‚Feste Bibliotheksversion‘	2
Methode ‚Neueste Bibliotheksversion‘	3
Methode ‚Platzhalter‘	4
Ziel: Ein Projekt führt zum immer gleichen Maschinencode	5
Hintergrund: Definition der Auflösung für Platzhalter	6
Gerätebeschreibung der Steuerung	6
Library Profile	6
Überschreiben einer Platzhalter Definition	7
Benennungskonvention für Bibliotheken und Platzhalter	7
Fazit und Leitfaden	8
Disclaimer	9

Einleitung

In Combivis studio 6 ist es möglich, weit ineinander kaskadierte Bibliotheken zu erstellen, diese Möglichkeit wird durch KEB z.B. im Bereich der Bibliothekssammlung rund um den KEB_ChannelHandler auch verwendet. Es ist allerdings fast immer unbedingt erforderlich, dass sich verschiedene Bibliotheken auf gleiche Basisbibliotheken beziehen – insbesondere immer dann, wenn z.B. bibliotheksübergreifende Schnittstellen eingesetzt werden oder sich verschiedene Elemente in zwei Bibliotheken auf eine gemeinsame Basis in einer dritten Bibliothek beziehen.

Ist diese Voraussetzung der immer gleichen Basisversion nicht erfüllt kommt es zu Referenzierungsproblemen, da verschiedene Versionen einer einzigen Bibliothek per Definition nicht I/O kompatibel sein müssen und somit verschiedenen Schnittstellen innerhalb ihrer Interfaces und Bausteine aufweisen können bzw. dürfen.

Um dieses Problem in den Griff zu bekommen ist in Combivis ein so genanntes Platzhalterkonzept enthalten, das in diesem Dokument erläutert wird und dessen Hintergründe beleuchtet werden sollen. Dabei richtet sich dieses Dokument vornehmlich an Bibliotheksentwickler, die bereits ein gutes Grundwissen über die Möglichkeiten und Bedienkonzepte der IDE mitbringen.

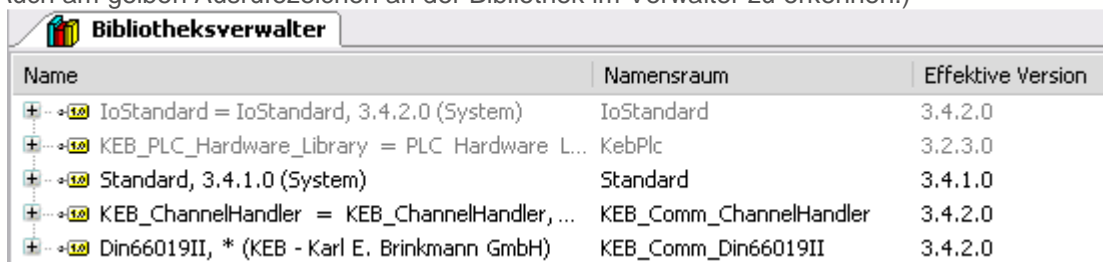
Vorgeschichte

Schon immer gab es in Combivis studio mehrere Möglichkeiten, eine IEC Bibliothek zum Projekt hinzuzufügen: Über eine feste, nicht veränderliche Auswahl der Bibliotheksversion, über eine Option, stets die neueste auf der jeweiligen IDE verfügbare Version zu verwenden (,*)' oder aber über einen so genannten Platzhalter (engl. Placeholder). Die verschiedenen Methoden rufen dabei unterschiedliche Effekte hervor, die nicht immer so vom Benutzer gewollt sind bzw. in ihrer ganzen Tragweite so nicht wahrgenommen werden:

Methode ‚Feste Bibliotheksversion‘

Bei dieser Methode legt der Entwickler eines Projektes oder einer Bibliothek sich schon bei der Erstellung fest, welche Version einer Bibliothek in diesem Projekt verwendet werden soll.

Eine Bibliothek, die mit einer festen Version in das Projekt eingefügt wurde, erkennt man an der Version unmittelbar hinter dem Bibliotheksnamen im Bibliotheksverwalter. Die in der Spalte ‚Effektive Version‘ angezeigte Version entspricht eben dieser oder bleibt leer, wenn die Version auf der IDE nicht verfügbar ist. (Auch am gelben Ausrufezeichen an der Bibliothek im Verwalter zu erkennen.)



Name	Namensraum	Effektive Version
IoStandard = IoStandard, 3.4.2.0 (System)	IoStandard	3.4.2.0
KEB_PLC_Hardware_Library = PLC Hardware L...	KebPlc	3.2.3.0
Standard, 3.4.1.0 (System)	Standard	3.4.1.0
KEB_ChannelHandler = KEB_ChannelHandler, ...	KEB_Comm_ChannelHandler	3.4.2.0
Din66019II, * (KEB - Karl E. Brinkmann GmbH)	KEB_Comm_Din66019II	3.4.2.0

Hier ist die Bibliothek ‚Standard‘ in der Version 3.4.1.0 fest hinzugefügt worden und die Version ist auf der IDE auch verfügbar.

Diese Version wird sich *nie* verändern; das hat mehrere Konsequenzen:

- Bestehende Referenzen auf Objekte dieser Bibliothek werden immer so erhalten bleiben, da sich die Bibliothek nicht verändern wird. (Vorausgesetzt man hält sich an die Konvention, dass jede Änderung in einer Bibliothek eine Erhöhung der Version zur Folge hat.)
- Die entsprechend hinzugefügte Version einer Bibliothek muss auf jeder IDE mit der das Projekt

bearbeitet oder die Bibliothek verwendet wird, verfügbar sein. (Um das sicher zu stellen sollte ein Projekt stets als Projektarchiv abgespeichert werden.)

- Sollen sich in einem komplexeren Aufbau, etwa einer Bibliothekslandschaft mit gemeinsamen Basisbibliotheken (vgl. KEB_ChannelHandler), verschiedene Bibliotheken auf eine gemeinsame Basis beziehen, müsste in jeder Version die gleiche feste Version verwendet werden. Ab einer gewissen Anzahl von Bibliotheken wird das nicht mehr praktisch umsetzbar.
- Sollte es eine neuere Version einer Basisbibliothek geben, in der etwa ein Bugfix enthalten ist, so müsste die feste Referenz auf die neue Version im Bibliotheksverwalter angepasst werden – und zwar für alle Bibliotheken, die die neue Version verwenden.

Methode ‚Neueste Bibliotheksversion‘

Hier entscheidet sich der Projektentwickler lediglich für die Art von Bibliothek, die er in seinem Projekt verwenden will. Die IDE wird zum Zeitpunkt des Öffnens des Projektes (oder im Falle einer Bibliothek, des Projektes, welches die Bibliothek verwendet) entscheiden, welche Version der Bibliothek aktuell die neueste ist, und ebendiese an allen Punkten in denen die Einfügemethode ‚Neueste‘ für die Bibliothek gewählt ist, einsetzen.

Eine Bibliothek, die mit der neuesten Version in das Projekt eingefügt wurde, erkennt man an dem Stern (*¹) unmittelbar hinter dem Bibliotheksnamen im Bibliotheksverwalter. Die in der Spalte ‚Effektive Version‘ angezeigte Version ist hier ein Verweis auf die tatsächlich verwendete Bibliotheksversion. Im vorigen Screenshot ist die Bibliothek ‚DIN66091II‘ über die Option ‚Neueste‘ zum Projekt hinzugefügt worden und die resultierende Version ist die 3.4.2.0.

Folgende Effekte werden/können bei dieser Methode auftreten:

- In einer Bibliothekslandschaft ist zumindest sichergestellt, dass verschiedene Bibliotheken dieselbe, wenn auch die Version betreffend noch nicht genau definierte Bibliothek verwenden werden. Aus diesem Grund war dies der Weg, der auch in den ChannelHandler Bibliotheken bis zu alle Versionen vor Stand 3.4.x angewendet wurde.
- Ein Update oder Bugfix in einer zu Grunde liegenden Basisbibliothek wird ebenfalls ohne Zutun des Projektentwicklers verwendet. Ein Anpassen der Referenzen in der Basisbibliothek entfällt.
- Wenn in einer IDE zu einem späteren Zeitpunkt eine neuere Version der Bibliothek verfügbar wird (etwa durch die Installation eines neuen Setups) so wird die IDE beim nächsten Öffnen des Projektes die neue Version verwenden. Das wird dazu führen, dass beim kompilieren unter Umständen ein anderer Maschinencode erzeugt wird, der ein Einloggen ohne Download unmöglich macht.
- Auch wenn ein Projekt als Projektarchiv gespeichert wurde (was die Konsistenz der Projektdaten ja gewährleisten soll) ist mit dieser Methode nicht sichergestellt, dass nicht doch in einer IDE auf der das Archiv entpackt wird, bereits eine neuere Version einer Bibliothek vorhanden ist und anstelle der im Archiv verwendet wird (siehe vorherigen Punkt.).
- Es ist zum Zeitpunkt des Erstellens eines Projektes nicht definiert, welcher Maschinencode erzeugt werden wird. Weiterhin ist es nicht möglich, die ursprünglichen, effektiven Versionen der verwendeten Bibliotheken in Erfahrung zu bringen!

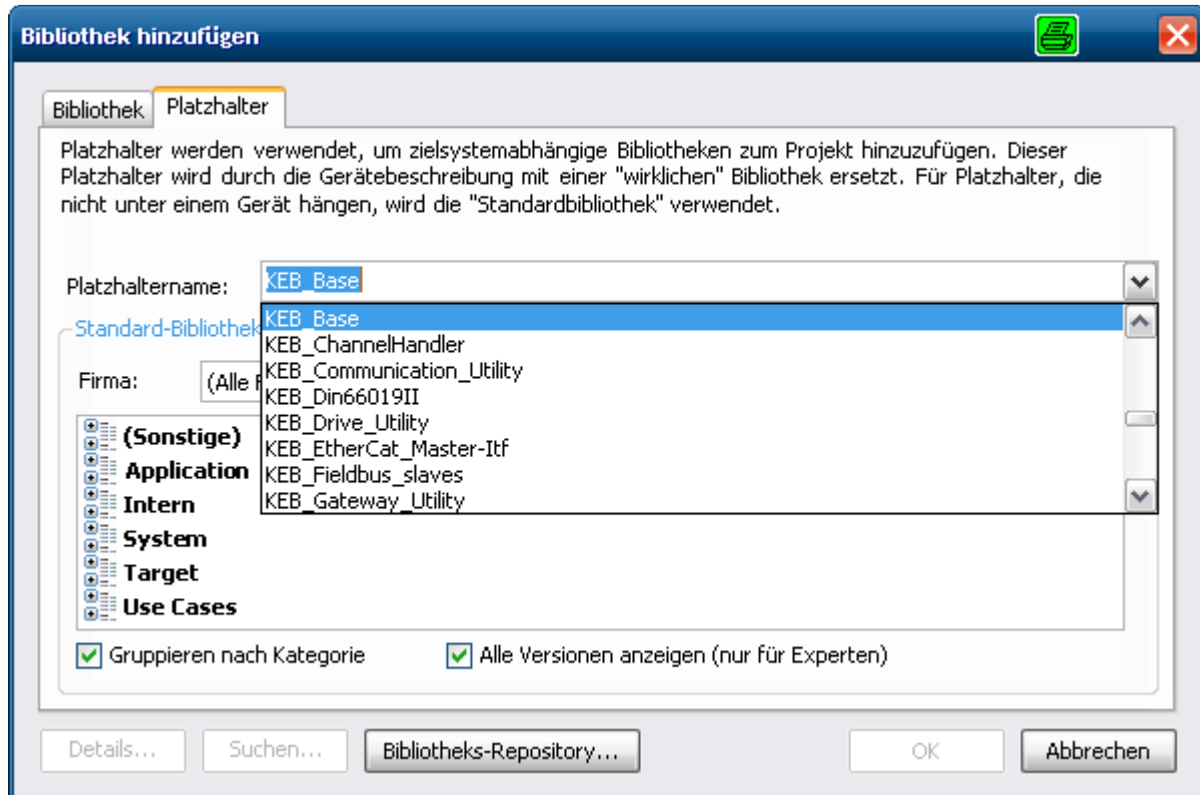
Methode ‚Platzhalter‘

Als dritte und letzte Möglichkeit kann eine Bibliothek auch als so genannter Platzhalter eingefügt werden. Auch dabei weiß der Entwickler zum Zeitpunkt der Erstellung einer Bibliothek noch nicht genau, welche Version der Bibliothek letztendlich angewendet werden wird - dies wird aber im Gegensatz zur Methode ‚Neueste Bibliotheksversion verwenden‘ nicht über die bloße Präsenz einer neueren Version sondern über die Definition der Platzhalter Bibliothek festgelegt. Hierfür gibt es zwei Möglichkeiten - siehe Hintergrund: Definition der Auflösung für Platzhalter. Für den Fall, dass bei der Entwicklung einer Bibliothek keine der Definitionen für den Platzhalter verfügbar ist, wählt der Entwickler beim Einfügen des Platzhalters eine Bibliotheksversion aus, die in diesem Fall verwendet werden soll - das sollte dann stets die neueste Version sein (‚*‘), um zumindest noch die Garantie zu haben, dass in diesem Falle dann alle Referenzen zumindest auf die gleiche Version zeigen.

Eine Bibliothek, die als Platzhalter in das Projekt eingefügt wurde, erkennt man am Platzhalternamen gefolgt von einem Gleichheitszeichen und der sich anschließenden effektiven Bibliothek, inklusive der Version. Die in der Spalte ‚Effektive Version‘ angezeigte Version sollte wiederum gleich sein. Im Screenshot ist die Bibliothek ‚KEB_ChannelHandler‘ über einen Platzhalter referenziert und dieser zeigt aktuell auf die Version 3.4.2.0.

Diese Möglichkeit vereint nun mehrere positive Eigenschaften der beiden vorher beschriebenen Methoden ineinander:

- Es ist über das Konzept der Platzhalter möglich, auch in unübersichtlichen Szenarien mit verschiedenen neuen Bibliotheken noch immer eine fest definierte Version eine Bibliothek zu verwenden.
- Referenzen von anderen Bibliotheken zeigen, wenn sie über denselben Platzhalter eingefügt werden, stets auf die gleiche Version (zumindest, wenn auch für den Fall eines nicht aufgelösten Platzhalters die neueste Version ausgewählt wurde.) Es gibt deshalb mit dem Konzept niemals das Problem, dass sich Bibliotheken auf verschiedene Versionen einer Basisbibliothek beziehen.
- Updates einer Basisbibliothek werden über die Definition des Platzhalters freigegeben. Änderungen der Referenzen in einer Bibliothek oder einem Projekt sind nicht erforderlich.



Soll eine Bibliothek über einen Platzhalter eingefügt werden, so muss im ‚Bibliothek hinzufügen‘ Dialog der Tab ‚Platzhalter‘ ausgewählt werden. Anschließend kann man den Platzhalter eingeben, oder aus einer Liste der auf der IDE bekannten Platzhalter auswählen. Anschließend wird noch die Standard Bibliothek, die verwendet werden soll, falls der Platzhalter in einem Bibliotheksprojekt nicht aufgelöst werden kann, ausgewählt. Hier sollte dann in aller Regel die neueste Version („*“) verwendet werden. (Die Auflösung eines Platzhalters gegen eine konkrete Version wird erst im jeweilige Zielprojekt erfolgen.)

Vorsicht: Es ist auch möglich, eine völlig andere Bibliothek als Standard Bibliothek anzugeben. Das ist aber so normalerweise nicht gewünscht.

Ziel: Ein Projekt führt zum immer gleichen Maschinencode

Um die langfristige Wartbarkeit eines IEC Projektes zu gewährleisten, ist es erforderlich, dass ein bestimmter Projektstand, einmal gespeichert immer wieder zum gleichen Maschinencode führt. Nur so ist die Möglichkeit gegeben, sich auch nach einer unter Umständen längeren Zeit in eine laufende Applikation (ohne Online Change oder gar Download) einzuloggen.

Dazu ist es zum einen notwendig, das Projekt in einem Projektarchiv abzuspeichern, um alle vom Projekt benötigten Bibliotheken auch zu einem späteren Zeitpunkt verfügbar zu haben.

Zum anderen ist es nur mit dem Konzept möglich, in dem Bibliotheken über Platzhalter referenziert werden:

Werden feste Versionen eingefügt, ist die Wartung von Bibliothekslandschaften sehr aufwändig bis, ab einer bestimmten Anzahl von Bibliotheken, unmöglich. Allerdings ist darüber gewährleistet, dass sich der erzeugte Code nicht ändert.

Werden, wie vorher beschrieben, Referenzen auf die neueste Version einer Bibliothek verwendet, bleiben auch umfangreiche Bibliothekslandschaften einfach wartbar. Allerdings ist es möglich, dass sich

ohne (bewusstes) Zutun des Anwenders Referenzen ändern und dadurch ein anderer Code erzeugt wird, was nicht erwünscht sein kann.

Die Lösung ist die Mischung beider Konzepte, bei der Verwendung von Platzhaltern werden die vorher genannten Vorteile der beiden anderen Optionen ineinander vereint. In den Bibliotheken werden nur Referenzen eingefügt, die es ermöglichen, neuere Basisbibliotheken zu verwenden, ohne die Bibliothek selbst zu ändern. Gleichzeitig werden durch die Auflösung der Platzhalter aber effektiv stets die gleichen Versionen verwendet (so lange das Gerät bzw. die Compilerversion gleich bleibt, siehe Hintergrund: Definition der Auflösung für Platzhalter). Es sollten Bibliotheken deshalb stets per Platzhalter in ein Projekt eingefügt werden.

Hintergrund: Definition der Auflösung für Platzhalter

Es existieren zwei Möglichkeiten, wie eingefügte Platzhalter zu effektiven Bibliotheken aufgelöst werden können. Dabei hat die Variante der Definition in der Gerätebeschreibung der Steuerung Priorität.

Gerätebeschreibung der Steuerung

Als erstes werden die in der Gerätebeschreibung hinterlegten Platzhalter Definitionen durchsucht. Typischerweise werden hier alle solchen Platzhalter definiert, die direkten Bezug zur Steuerungsfirma- bzw. Hardware haben, z.B. die Sys-Libs. Hintergrund ist hier, dass es sich bei diesen Bibliotheken sehr häufig um IEC ‚Frontends‘ zu Funktionen handelt, die auf eine C-Routine in der Firmware zurückgreifen. Nur der Firmware Entwickler kann somit entscheiden, welche Funktionen in der Firmware implementiert sind und wird deshalb die für die jeweilige FW geeigneten Version der Sys Libs in der Beschreibung der Steuerung hinterlegen.

Library Profile

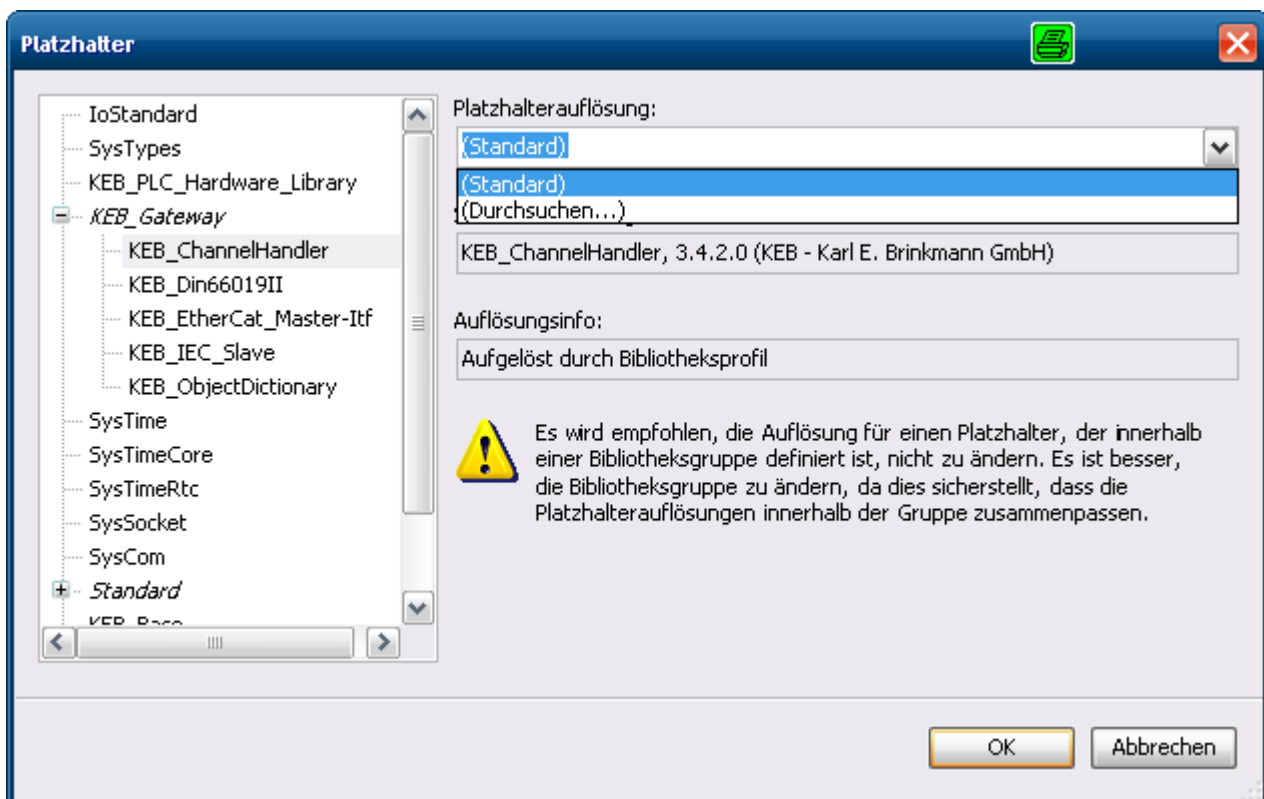
Über die vorher genannte Möglichkeit hinaus gibt es auch noch die Option, Platzhalter im so genannten Bibliotheksprofil der IDE anhand der im Projekt verwendeten Compilerversion (Projekt → Projekteinstellungen → Compiler) aufzulösen. Diese Methode ist vor allem für plattformunabhängige Bibliotheken sinnvoll, die nicht direkt auf der Hardware aufsetzen sondern bereits etwas abstrahiert sind, z.B. die KEB Utilitys oder die ChannelHandler Bibliotheken. Dabei legt dann immer die eingestellte Compilerversion die jeweilige Auflösung der Version fest, was dazu führt das für die Verwendung von neueren Bibliotheken (oder auch Bibliothekssätzen) die Compilerversion entsprechend aktualisiert werden muss. (Wobei aber die bestehenden Einschränkungen weiter bestehen, d.h. dass z.B. die Majorversion des Gerätes Y in 3.Y.x.x dieselbe wie die des compilers sein muss usw.)

Überschreiben einer Platzhalter Definition

Das Überschreiben eines Platzhalters kann zu unerwarteten Effekte führen und sollte deshalb nur in absoluten Ausnahmefällen erfolgen!

Obwohl die Platzhalter im Normalfall stets auf eine effektive Version zeigen kann es, z.B. zu Testzwecken oder aus anderen Gründen notwendig sein, die Definition eines Platzhalters manuell zu verändern.

Das ist in jedem Projekt über die Schaltfläche ‚Platzhalter...‘ möglich. Dabei öffnet sich dann zunächst eine Liste der im Projekt verwendeten Platzhalter und deren aktuelle Auflösung:



Hier unterscheidet die IDE zwischen ‚Aufgelöst durch Gerät‘, ‚Aufgelöst durch Bibliotheksprofil‘ und ‚Aufgelöst durch Platzhalterweiterleitung‘. Einige Platzhalter sind in Gruppen zusammengefasst, um eine bessere Gliederung der Liste zu erzielen, z.B. die Bibliotheken des KEB Gateways. Gruppen erkennt man an der kursiven Schrift und dem ‚+‘ vor dem Gruppennamen, was die Möglichkeit des Aufklappens der Gruppe optisch andeutet. Ist eine Definition nicht mehr Standard, so wird dieser Platzhalter und auch die entsprechende Gruppe fett dargestellt. Die manuelle Auswahl einer alternative Bibliothek erfolgt genauso, wie das hinzufügen einer Bibliothek zum Projekt – durch einen Klick auf Durchsuchen öffnet sich der Bibliotheksverwalter und die gewünschte Version kann ausgewählt werden.

Benennungskonvention für Bibliotheken und Platzhalter

KEB verwendet per Definition immer auch den Namensraum KEB für alle seine Bibliotheken und Platzhalter. Deshalb sind auch die bisher definierten Platzhalter alle im Format ‚KEB_[Bibliotheksname]‘ – werden neue Bibliotheken erstellt sollten diese auch stets dieses Benennungsschema einhalten, da es die Übersicht stark vereinfacht.

Fazit und Leitfaden

Das vorgestellte Konzept bietet eine durchgängige Möglichkeit, die Verwendung von Bibliotheken in allen Szenarien in sich konsistent und für den Endanwender einfach handhabbar zu machen. Die KEB Applikationsbibliotheken sind vollständig auf Platzhalter umgestellt und die folgenden Punkte sollen Anwendern Entscheidungshilfen geben, wie eine Bibliothek am besten eingefügt wird.

In einem normalen IEC Projekt...

- ...sollten Bibliotheken immer so eingefügt werden, dass unter allen Umständen die stets die gleiche Version verwendet wird. (Ein Projektstand – immer gleicher erzeugter Maschinencode)
 - Entweder werden feste Bibliotheksversionen eingefügt
 - Oder der Anwender nutzt, besonders für System- und andere hardware-abhängige Bibliotheken, die für die jeweilige Steuerung definierten Platzhalter. So ist einerseits sichergestellt, dass keine ungültige Funktionen im Projekt verwendet werden, die zu Fehlern beim Download des Codes führen würden und andererseits werden darüber auch bei einem späteren Update der Steuerung wieder automatisch die korrekten, freigegebenen Bibliotheken verwendet.
- ...sollten nur in absoluten Ausnahmen Referenzen über die Methode „Neueste Version verwenden“ (mit *) verwendet werden. Nur so ist es möglich, einen bestimmten Projektstand dauerhaft zu konservieren.

In einem Bibliotheksprojekt...

- ...werden weiter reichende Anforderungen an die intern verwendeten Bibliotheken gestellt: Die Version von zugrunde liegenden Basisbibliotheken muss...
 - ...eindeutig sein.
 - ...unter Umständen auch in der gleichen Version in anderen Bibliotheken verwendet werden.
- Dazu sollten in Bibliotheken alle Referenzen über Platzhalter eingefügt werden. Als Standard Bibliothek sollte, für die Auflösung des Platzhalters während der Bibliotheksentwicklung, die erwartete Version ausgewählt werden; meistens ist die Einfügeoption ‚Neueste‘ empfehlenswert.

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 particularly necessary, if changes are executed, which serve for the further development or adaptation 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