KBL Implementation Guidelines

The VDA / PSI recommendation Harness Description List (KBL) defines an information model, a data dictionary, and an XML schema derived from and compliant to the model. The intention of the model was to cover a wide range of use cases and application scenarios. For this reason the specification had to be kept generic in some degree and in some aspects. However, for specific scenarios and / or use cases a more detailed description on “how the different pieces fit together” is possible.

To avoid dialects in the different KBL implementations, further guidelines or recommendations are necessary. This collection of implementation guidelines contributes to the unambiguous interpretation of the KBL standard. For various wiring harness definition or electrical system aspects and scenarios the correct instantiation is shown and specific hints for correct usage are given.

Contributing and Proposals

If you find any bugs in the implementation guidelines or if you have a request for a specific topic, or if you would like to contribute your own tutorials please drop us an issue on the PROSTEP JIRA. If you don’t have an account there yet, see here for the procedure to get one.

Additional Resources

- KBL Implementation Guidelines (PDF-Version)

Latest Changes

The following table contains lately changed pages, sorted descending by last change.

Title	Latest Content Addition / Commit	Created	Changed
Sealed Ring Terminals	KBLFRM-631: Added guideline for ring terminals and overlapping wire protections. Latest Commit: Review notice removed after expiry of the objection period.	2020-05-29	2024-10-22
Specification of Splices	KBLFRM-1026: Added guideline for the distinction of parallel and end splices. Latest Commit: Review notice removed after expiry of the objection period.	2021-05-19	2024-10-22
3D B-Spline Representation	KBLFRM-923: Added clarification for the usage of B-Splines in KBL. Latest Commit: Improved display of incoming & outgoing relations. Added support for deprecation.	2020-05-04	2023-01-31
Supplier Parts and Supplier Part Selection	Latest Commit: Improved display of incoming & outgoing relations. Added support for deprecation.	2020-05-26	2023-01-31
Multiple Cavity Parts	KBLFRM-950: Added implementation guideline for multiple parts at a contact point. Latest Commit: Improved display of incoming & outgoing relations. Added support for deprecation.	2020-06-16	2023-01-31

1 3D B-Spline Representation

For historical reasons the documentation of the B_spline_curve is not clear and unfortunately leaves room for interpretation. This Implementation Guideline clarifies the relevant facts and describes the valid interpretations in the field.

1.1 Background

The following section contains a short wrap-up about NURBS (Non-uniform rational B-spline). The description in this section aims primarily at an informal understanding and not at a precise and 100% correct mathematical definition. It contains just enough information to understand the definitions in this guidline and is a summary from multiple sources. For more details check for the following links [1][2][3] from which this summary has been derived.

NURBS are a commonly used as a representation of surfaces and curves in computer-aided design and are part of numerous industry wide standards. For the KBL & VEC only the representation of curves (the 3D centerline of Segments) as NURBS is relevant. NURBS are representing a curve as a mathematical function. The appearance of the curve can be influenced by a set of parameters:

- **Degree d:** This is usally one of $\left [1,2,3,5 \right]$. Sometimes, there are references to the *order* of a NURBS, where *order* is $d + 1$. The *degree* defines the number of control points that influence any given point of the curve.
- **Control Points**: The control points define the shape curve. Each NURBS can have n control points, where $n > d$.
- **Weight**: Each control point can define an individual weight.
- **Knot vector**: The knot vector defines where and how the control points affect the NURBS curve. The number of knots is equal to $n + d + 1$. The values in the vector have a non-decreasing order. However, consecutive knots can have the same value, e.g. $(0,0,1,2,3,4,4)$ is a valid vector. A number of coinciding knots is sometimes referred to as a knot with a certain **multiplicity**. A knot where the *multiplicity* is equal to the *order* ($d+1$) is a **full** multiplicity knot.

1.1.1 Special Cases of NURBS

The NURBS (Non-Uniform Rational B-Spline) are the most common form. There are groups of NURBS that have special properties:

1. If all control points have the same weight ($w=1.0$) the B-Spline is called **non-rational**
2. If knot vector starts and ends with a **full** multiplicity knot the B-spline is called **clamped**. A *clamped* B-spline starts in the first and ends in the last control point.
3. **Uniform**: There are some slightly different interpretations about the defintion of *uniformity.* In general *uniform* refers to the distribution of the knot values in the knot vector. Some sources (e.g. [2]) define, that if the knot vector is clamped, all other knots have a multiplicity of one, and all knots

(values) have the same distance, the B-spline is called **uniform**. For example a NURBS with $d=4$ and with a knot vector $(0,0,0,0,0,1,2,3,4,5,5,5,5,5)$ would be *uniform*. Other sources (e.g. [4]) differentiate between *clamped uniform* and *unclamped uniform*:

- **Clamped uniform** would correspond to the defintion above,
- **Unclamped uniform** would require all knots to have a multiplicity of one, and all knots (values) to have the same distance (e.g. $(0,1,2,3,4,5,6,7,8,9,10,11,12)$).

1.2 Current Situation in the KBL

The intention of the KBL was, to keep the B-spline data model as simple as possible. Therefore the data model just contains the *control points* and the *degree*, assuming that all other parameters have an unambigious default when the set of valid NURBS are restricted to **Uniform non-rational B-Splines (UNRBS)**. This is the reason why the KBL model does not define a *weight* nor a *knot-vector*.

Unfortunately, the definition of the KBL was not as precise as it could have been. No concrete definition was made as to whether these are *clamped* or *unclamped* uniform B-Splines. At the moment implementations for both variants exist.

A subsequent restriction to one of the two variants was discussed in the relevant committees and considered impracticable. The reasons for this were, on the one hand, the large volume of existing data and, on the other hand, the non-trivial conversion process between the two variants, which makes it virtually impossible to implement it in practice.

1.3 Definition

The B_spline_curve in the KBL represents a **uniform non-rational B-spline** (either *clamped* or *unclamped*). When rendering 3D KBL data, the renderer has to use external knowledge to determine which variant is used.

Note: *Due to this fact, the B-Spline modeling in VEC version 1.2.0 and higher has been extended in a way so that all information of a NURBS can be represented.*

---

1. https://en.wikipedia.org/wiki/Non-uniform_rational_B-spline ↩
2. https://wiki.mcneel.com/de/rhino/nurbs ↩ ↩
3. https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/B-spline/bspline-curve.html ↩
4. Les Piegl, Wayne Tiller: The NURBS Book, Page 572 ↩

2 Supplier Parts and Supplier Part Selection

The Part in the KBL represents an identifier for the PDM information of a component in the context of a specific company. Therefore the Part has the mandatory key attributes *Part_number* and *Company_name*. The most common case in the KBL is, that the Part within a harness, represents an OEM component specification.

There are cases where this specification can be satisfied by components from different suppliers. The harness manufacturer has a freedom of choice between these components (e.g. for reasons of local availability). However, in the end the choosen alternative shall be documented. This implementation guideline defines how this scenario shall be represented in the KBL.

Basically this scenario is covered by the Alias_identification as it represents *"...an additional identifier that is used to identify the object of interest in a different context, either in another Organization, or in some other context."*. Therefore, the following applies:

- Alias_identification in the context of a Part shall define the **possible** supplier part numbers.
- Alias_identification in the context of a Connection_or_occurrence shall define the **selected** supplier part number.

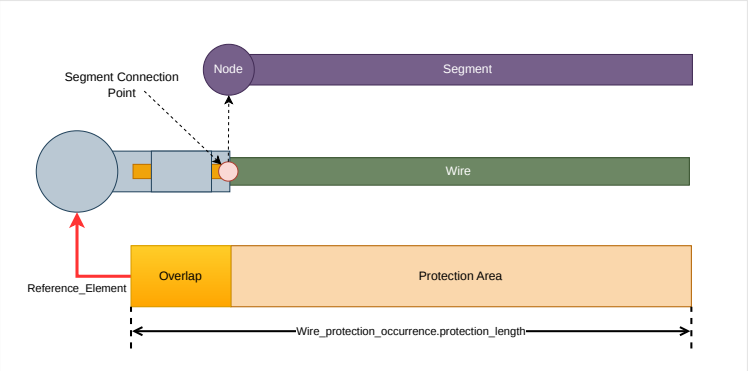
The following attribute mapping shall be used:

Attribute	Value - Definition
Alias_id	The part number of the component in the "other" context (the supplier part number).
Scope	Defines the scope where the part number is defined (the company name of the supplier).
Description	The type of the alias id. Despite the misleading naming of the attribute, this is not a human readable description, but the actual type / role of the alias id. For the scenario defined by this implementation guideline, the following values shall be used: • **supplier number** for possible supplier part numbers in the context of the part. • **selected supplier number** for selected supplier part numbers in the context of an occurrence.

3 Sealed Ring Terminals

Ring terminals are often installed with a wire protection (e.g. a shrinking tube) that is covering the crimp area (see the figure on the right). This implementation guideline defines how this scenario shall be represented in the KBL.

The necessity for a separate definition of this situation arises from the fact, that in the KBL wire protections are always placed on segments of the topology. For electrical end points (e.g. connectors or ring terminals) the last segment ends at a defined location, the *segment connection point* (a.k.a. bundle connection point). For ring terminals, the start of the crimp area is defined as segment connection point (the blue marker in the figure on the right). That means, that the wire protection (the yellow tube) is overlapping into an area that is outside of the topology (the red arrow).



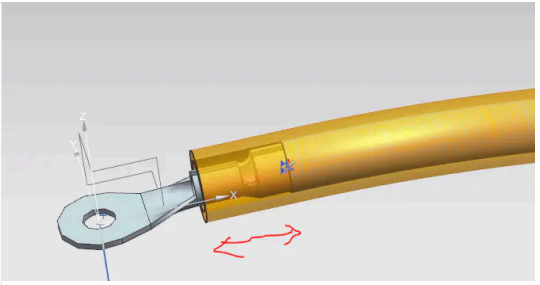


FIGURE 1: Sealed ring terminal with overlapping protection

FIGURE 2: Sealed ring terminal with overlapping protection (Sketch)

A regular Protection_area assigns a Wire_protection_occurrence onto a Segment. It starts and ends on locations defined by a value that relates to the curve parameter of the segment ($0.0 \leq x \leq 1.0$). A representation of the scenario by these means (e.g. a curve parameter $x < 0$) is not possible without simultaneously creating a representation that is highly open to interpretation, especially for other cases (e.g. "What does a curve parameter less than zero mean for segments that end at an intersection?").

Therefore the defined representation of this scenario is:

1. The Protection_area is defined as if it would end at the segment connection point.
2. The Wire_protection_occurrence uses the ring terminal as *reference_element* (possible with KBL 2.5 SR-1 and later).
3. The *protection_length* is the calculated length from the Protection_area plus the length of the overlapping area.

4 Multiple Cavity Parts

The Contact_point in the KBL allows the referencing of an unspecified number of Part_usage_selects in the role *Associated_parts*. This implementation guideline clarifies the valid use cases of multiplicities > 1.

For a standard contacting (the "90% case"), the *Associated_parts* association references a single Terminal_occurrence and an (optional) Cavity_seal_occurrence for sealed contacting situations.

Chapter 2.2 of the KBL Recommendation Document (V2.5) already defines:

> The Contact_point specifies a single contacting variant. This means that the contacting is manufactured, as specified in the KBL. Either all participants (Cavities, Terminals, Seals, Wires) set into a relationship by the Contact_point exist in a specific harness or none. There is no requirement, to filter the participants of a contacting situation with information derived from Module_configurations in order to create a manufacturing variant.
>
> The Contact_point represents a single potential. As a consequence, all cavities and wires referencing / being referenced by a Contact_point are short-circuited and have the same potential (even if the signals on the wires are named differently.

In other words, using the *Associated_parts* to represent a 150% contacting situation is not a valid approach in the KBL. However, there are still remaining situations, where more components than one Terminal_occurrence (and one Cavity_seal_occurrence) participate in a specific contacting situation. The representation of those cases in the KBL depends on the role that the components have in the contacting sitation.

- Multiple Terminal_occurrence and/or Cavity_seal_occurrence are valid, provided all the components of the respective class have a conducting (terminal) or sealing function and they always occur together (100%). Illustrations for such cases can be found on the right.
- All other components (e.g. wire fixations, damping elements) have to be represented as an Accessory_occurrence to the primary terminal component.

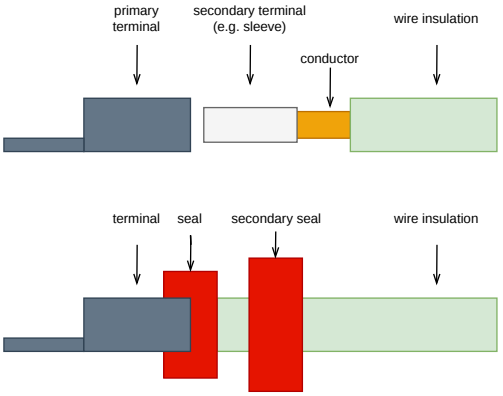


FIGURE 3: Illustration Multiple Terminals & Seals

The representation is illustrated the following figure. Reasons for choosing this representation where:

1. It is aligned with the representation in the VEC, which defines individual classifications for those components (WireEndAccessorySpecification & CavityAccessorySpecification)
2. Representing those "accessories" as *regular Associated_parts* would mislead existing established sanity checks (e.g. the existence of a Cavity_seal_occurrence would raise the question why only one contact in the connector was sealed.)

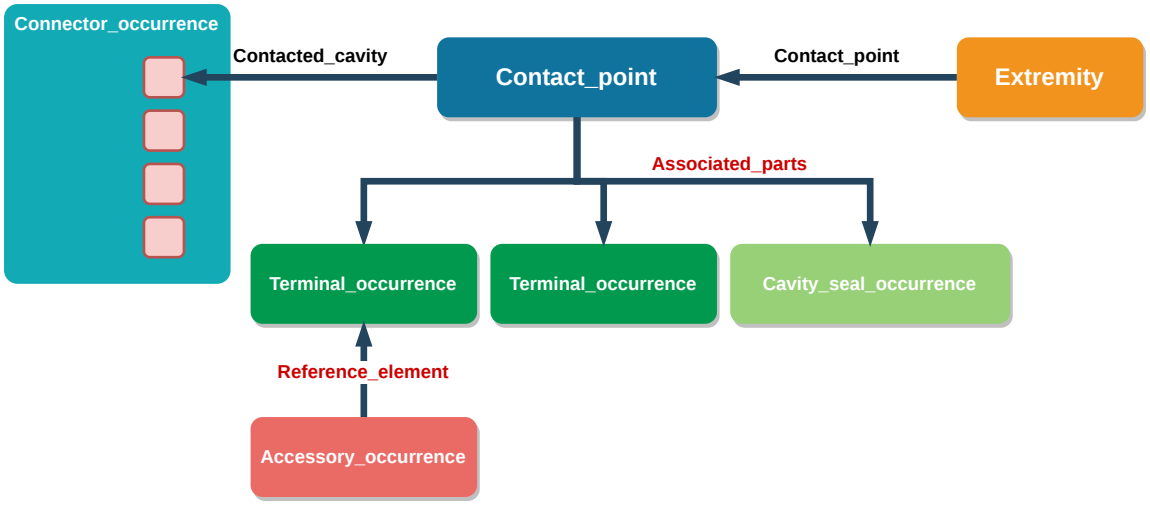


FIGURE 4: Schematic illustration of the contact point

5 Specification of Splices

For historical reasons, the representation of splices in the KBL is quite implicit. A splice is represented as a regular Connector_housing respectively a Connector_occurrence. It can be identified with the *Connector_occurrence.Usage* and/or *Connector_housing.Housing_type* (see KBLFRM-1025). Due to these generic approach the splice is currently missing some splice specific information on the occurrence level in the KBL model:

- **Type:** Defines if the splice is a parallel splice or an end splice. An end splice is a splice where all wires are inserted from the same side of the splice, a parallel splice has wires from both sides.
- **Seal State:** Defines if the splice should be sealed or not.

To overcome this limitation the following Installation_instructions shall be used on the Connector_occurrence

Instruction_type	Definition	Instruction_value (one of)
splice_type	Defines whether there are requirements for the implementation of the splice, and if so, what they are.	*[end_splice, inline_splice, unspec_splice]*
seal_state	Defines requirements for the implementation of the sealing of the splice	*[sealed, unsealed]*