

IDM-3D. Connection

Ekkehard Beier (Editor)

Version

Specification: IDM-3D.Connection 1.0

Document version: Final Version

Copyright: The copyright for the starting point model on which this specification is based is and remains with intelligentgraphics AG. This does not result in any restrictions on the right of use of this specification by third parties from the point of view of intelligentgraphics AG.

Specification

IDM-3D.Connection

IDM-3D.Connection is a part of the IDM-3D specification and describes the definition of attachment points, hereafter also 3D attachment points, which are used to connect products in the sense of IDM Living. Typical examples of product connections are:

- Symmetrical left-right connections, e.g. a cushion composition
- Asymmetric child connection, e.g. a headrest (as a separate product) on another product

Starting points as well as the relations between pairs of starting points are in principle dynamic and their parameters or validity are influenced by certain conditions:

- An intra-product rule influences the parameters of a connection point: Depending on the width of the selected left armrest, the width coordinate of the right connection point results. Depending on the selection of the feature 'with/without headrest support', the attachment point of a headrest is active or inactive.
- An explicit inter-product rule considers at least two products to parameterize their attachment points: Couch elements can only be concatenated if their seat height is equal.

- A generic inter-product rule is defined independently of concrete products: a product cannot be scheduled despite valid pairing because this would result in a collision with another product.

This results in a complex overall algorithm which cannot be described in full because, on the one hand, there can be different scenarios (purely commercial order entry, 2D planning based on abstracted symbols, 3D planning with realistic collision detection). On the other hand, such an overall algorithm is also part of a specific software solution and thus the subject of the competition.

In this respect, IDM Living and IDM-3D.Connection are only building blocks, albeit very important ones, for an ultimately specific overall algorithm:

- IDM Living describes the principle pairing of starting points.
- IDM Living defines connection vectors in 2D plan view symbols with a defined semantics and static or dynamic coordinates.
- IDM Living describes the nominal dimensions which, when applied to dynamic 2D plan view symbols, together with the connection vectors defined in them, give concrete coordinates.
- IDM-3D.Connection describes parametric starting points - and the pairings between them via generic key pairs or tags.

This results in an overlap of data for the area of pairings on the one hand, and for the area of coordinates on the other. In the area of pairings, IDM-Living is to be the leading system, which will possibly be refined by IDM-3D.Connection. It is also conceivable to use IDM-3D.Connection as a fast variant during interactive movements and to carry out a correspondingly time-consuming, usually server-based evaluation only when the movement is completed. For the overlap in the area of the coordinates, a definition is made below.

An IDM 3D.Connection record is preferably a complete set of the starting points of a project, where the assignment of a project to a manufacturer, a product series, a sub-product series, several product series [at commercial level] is not defined. For example, it may be useful to group starting points with reference to one (1) IDM Living record. Starting points are assigned to one (1) product.

There is a functional overlap between IDM-3D.Connection and IDM Living (incl. the SVGs), hereafter 2D attachment points. This is defined in this regard:

- In a 2D context, the 2D handles have priority. If they are missing and 3D points of interest are present, the latter are used.
- In the 3D context, the 3D attachment points have priority. If they are missing and 2D points of interest are present, the latter are used.

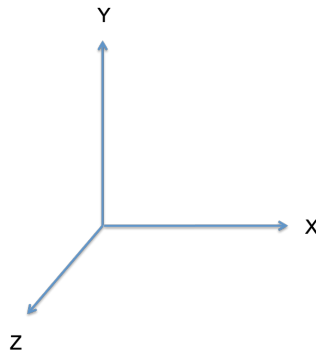
Reference

Starting points are defined relative to the product origin. This origin is generally: left, bottom and back - ignoring transitions through functions/animations¹.

On the other hand, other covers are recommended, especially for add-on parts such as headrests or footrests - usually on the axis of symmetry.

Height-neutral add-on parts should be modelled and only be moved upwards via a corresponding pair of starting points.

The reference system has the unit of measurement 1 meter. This means: 0.5 corresponds to 0.5 meter. The underlying coordinate system is defined as follows:



- x-axis: to the right [ascending values].
- y-axis: upwards
- z-axis: forward

The interpretation of the rotation results according to the right-hand rule.

Parametrics

The definition of starting points is in principle parametric. This means that only the actual configuration state ultimately determines the trigger point, which also includes its activation. Consequently, certain characteristics of a trigger point are either fixed values of the specified type (Boolean, number) or an expression whose evaluation results in a value of the specified type.

Expressions are specified as strings that are included in a block, e.g. "{ 0.4 * 0.5 }"². The content of the expression is interpreted as JavaScript (ECMA 5.1). The

¹ This specification is advantageous from the point of view of 3D modelling, especially for mass data creation. This does not contradict the use of an x-z centre point for the interactive rotation of products, since such centre points can or must also be determined dynamically (multi-product selection).

evaluation context shall provide a function $\$s$ that allows access to the configuration state.

Let the function $\$s$ be defined as follows:

```
 $\$s$ (key: string, default: string): string
```

Here, *key* is the identifier of the relevant characteristic and *default*, the value to be delivered if no value exists for this characteristic in the configuration state.

The runtime environment must provide the configuration state as a flat list, passing both feature identifiers and values as strings. Numeric values must be converted in the scientific notation, Boolean values as True or False.

Furthermore, user-defined functions can be declared to make expressions more compact. The functions must not cause any side effect, e.g. by modifying global variables or functions like $\$s$.

Starting point parameters

Parameters that are parametric are indicated by a $\{\}$. Numbers must be given in scientific notation.

Key: string

The parameter defines the primary key of the starting point in the context of a product. It is also used to define/establish point pairing.

The parameter is mandatory and is limited to 10 ASCII characters (A ... z, 0 ... 9, _, -).

Foreign: string

The parameter defines a mask for starting points that can be placed at this starting point. The character *** can be used to allow several keys. The mask `Links_*`, for example, allows the keys `Links_1` and `Links_2`.

The parameter is mandatory and is limited to 10 ASCII characters (A ... z, 0 ... 9, _, -). In addition, the character *** can be used.

Tags: string[]

² The first and last characters are therefore reserved for the block and thus also indicate the necessity of evaluating the expression - in contrast to a direct parsing of the target type.

The optional parameter defines a list of tags to express an additional fine-grained pairing of starting points. A tag is a free character string with the characters (A ... z, 0 ... 9, _, -). Two tags are identical if their normalization to lower case letters is identical.

The parameter is specified either as a string-serialized JSON array of strings or as a parametric expression whose evaluation must yield a string array.

Type: string

The parameter defines the type of the starting point. It is also used to define/establish the pairing of starting points.

The parameter is mandatory and must have one of the following values:

- Child - Starting point at which a product can be planned as an add-on element.
- Bottom - Starting point at which a product can be planned as a vertical neighbour. The starting point should be at the bottom when looking at the product.
- Left - Starting point at which a product can be planned as a horizontal neighbour. The starting point should be on the left when looking at the product from the front.
- Origin - Origin of the product in relation to a pair of starting points.
- Right - Starting point at which a product can be planned as a neighbour. The starting point should be on the right when looking at the product from the front.
- Top - Starting point at which a product can be planned as a vertical neighbour. The starting point should be at the top when looking at the product.

Reserved: Front, Back, Alignment, Minimum, Maximum

The starting point pairings at type level are then:

- Origin + Child - An asymmetric relationship in which manipulations of the Origin product affect the Child product rather than vice versa.
- Left + Right - A symmetrical relationship in which manipulations of the Left product affect the Right product as well as vice versa.
- Top + Bottom - A symmetrical relationship in which manipulations of the top product affect the bottom product as well as vice versa.

Position: {}

The parameter defines the position of the starting point relative to the origin of the product, which is usually but not necessarily left - back - down.

Position.X: number {}

The parameter defines the X coordinate of the starting point in meters. The default value is 0.

Position.Y: number {}

The parameter defines the Y coordinate of the starting point in meters. The default value is 0.

Position.Z: number {}

The parameter defines the Z coordinate of the starting point in meters. The default value is 0.

Rotation: {}

The parameter defines the position of the starting point relative to the origin of the product, which is usually but not necessarily left - back - down.

Rotation.X: number {}

The parameter defines the exclusive rotation around the X axis in degrees. The default value is 0.

Rotation.Y: number {}

The parameter defines the exclusive rotation around the Y axis in degrees. The default value is 0.

Rotation.Z: number {}

The parameter defines the exclusive rotation around the Z axis in degrees. The default value is 0.

Enabled: boolean {}

The parameter defines whether the starting point is active or inactive.

Representation: string

The parameter sets a visual representation for the starting point.

If necessary, visual representations can be parameterised in terms of colour. They must be excluded from the collision detection.

The following values are defined:

- Standard - Standard or default representation
- Ghost - At this starting point a so-called ghost representation of a product to be displayed there should be shown.
- Ghost2 - At this starting point a so-called ghost representation of a product to be displayed there should be shown, but a lightweight version should be used³.
- HotSpot - This is a pseudo attachment point to be used for special purposes.

Starting point pairing

Altogether, a starting point pairing results when:

- Key and foreign match, including wildcard semantics if necessary,
- there is at least one identical tag on both sides, as long as at least one starting point defines tags,
- the types fit as described above.
- both trigger points are activated,
- the starting points transformed into the global coordinate system
 - a position offset < 0.2 [m].
 - a rotation offset < 45 degreeshave (although these are indicative values that can be handled differently or adaptively).

References

[ECMA5.1] ECMAScript Language Specification, <https://262.ecma-international.org/5.1/>

[SCHEMA] <https://archive.intelligentgraphics.biz/schemas/idm-3d/connection.json>

³ In the field of shelf planning, forexample, Ghost can be used for a macro grid and Ghost2 for a micro grid.



History

IDM-3D.Connection 1.0 – 2023-10-06

Contact

Data Competence Center e. V.

Goebenstraße 4-10

32052 Herford

Info: www.dcc-moebel.org

Dr.-Ing. Olaf Plümer

E-mail: pluemer@dcc-moebel.org

Tel.: +49 52 21 / 12 65-37

PD Dr Ing habil Ekkehard Beier

E-mail: eb@intelligentgraphics.ag

Tel.: +49 36 77 / 66 73 520