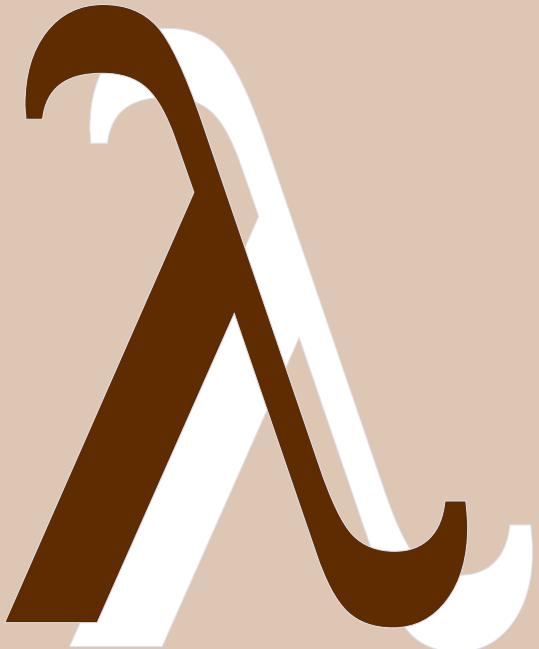
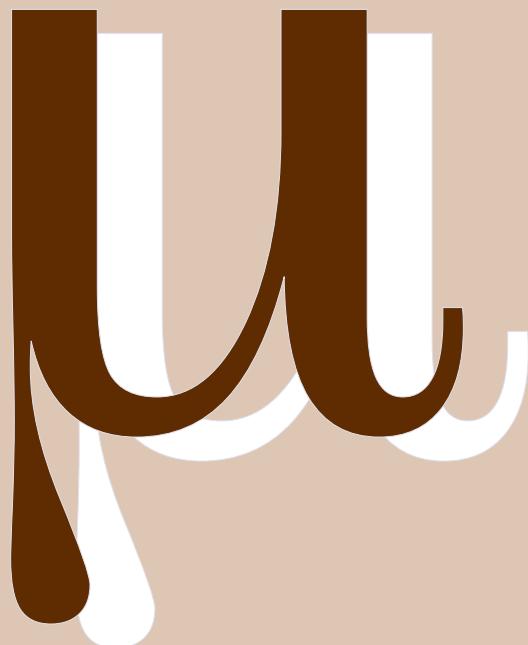


FAT | 286



xMCF

A Standard for Describing
Connections and Joints in the
Automotive Industry



xMCF

A Standard for Describing

Connections and Joints

in the Automotive Industry

FAT-AK25 Fügetechnik

Version 3.0

Documentation of Contents and File Format

Das Forschungsprojekt wurde mit Mitteln der Forschungsvereinigung Automobiltechnik e. V. (FAT) gefördert.

Table of Contents

1	INTRODUCTION	17
1.1	Motivation.....	17
1.2	MCF at Ford	17
1.3	From MCF to χ MCF - The Scope of the Document.....	17
2	DESIGN PRINCIPLES AND BASIC FEATURES OF XMCF.....	19
2.1	Design Principles.....	19
2.2	Idealization of Joints.....	20
2.3	Reconstruction of Joints from χ MCF	20
2.4	Description of Topology.....	20
2.5	χ MCF in the Development Processes	21
3	KEY-WORDS OF XML SPECIFICATION	24
3.1	Key-words	24
3.2	Operators	25
4	PARTS, PROPERTIES AND ASSEMBLIES	26
4.1	Parts	26
4.1.1	Part Labels.....	26
4.2	Properties.....	26
4.3	Assemblies.....	27
5	FILE STRUCTURE OF XMCF	28
5.1	Elements containing general information	28
5.1.1	Date.....	28
5.1.2	Version	29
5.1.3	Unit System	29
5.2	Application, User and Process Specific Data	30
5.2.1	User Specific Data <appdata>	30
5.2.2	Finite Element Specific Data <femdata>	32
5.3	Connection Data <connection_group/>	34
5.3.1	Connected Objects	35
5.3.1.1	Element <part/>	35
5.3.1.2	Element <assy/>	36
5.3.2	Contacts and Friction	37

5.3.2.1	Element < <i>contact_list</i> >	37
5.3.2.2	Element < <i>contact</i> >	37
5.3.2.3	Element < <i>partner</i> >	38
5.3.2.4	Element < <i>coefficients</i> >	38
5.3.2.5	Local Contact Properties	38
5.3.3	Joints	39
5.4	A Minimalistic Example of a XMCF file	40
5.5	XML Schema Definition	41
6	DATA COMMON TO ANY CONNECTION	42
6.1	Indices and their properties	42
6.2	Attribute label	42
6.3	Dimensions and Coordinates	42
6.4	Attribute quality_control	42
6.5	Custom Attributes list	43
6.6	Distinction between < <i>custom_attributes</i> > and < <i>appdata</i> >	48
7	0D CONNECTIONS	49
7.1	Generic Definitions	49
7.1.1	Identification	49
7.1.2	Location	49
7.1.3	Direction	50
7.1.4	Type Specification	51
7.2	Spot Welds	51
7.3	Robscans	52
7.4	Rivets	54
7.4.1	Blind Rivets	56
7.4.2	Self-Piercing Rivets	59
7.4.3	Solid Rivets	60
7.4.4	Swop Rivets	63
7.5	Threaded Connections: Bolts and Screws	65
7.5.1	Introduction	65
7.5.2	Contacts and Friction	66
7.5.3	Definition of element < <i>threaded_connection</i> >	69
7.5.4	Washer	72
7.5.5	Nut	72
7.5.6	Bolt	73
7.5.6.1	Possible Bolt and Screw Assemblies	77
7.5.7	Screw	79
7.5.7.1	Flow Drilled Screws (FDS)	80
7.6	Gum Drops	82

7.7 Clinches	83
7.8 Heat Stakes / Thermal Stakes	86
7.9 Clips/Snap Joints	88
7.10 Nails	91
8 1D CONNECTIONS	94
8.1 Generic Definitions	94
8.1.1 Identification	94
8.1.2 Location.....	94
8.1.3 Type Specification	95
8.2 Seam Welds.....	95
8.2.1 Description and Modeling Parameters	95
8.2.2 Seam Weld Definition Overview	96
8.2.3 Specific XML Realization.....	99
8.2.4 Generic Seam Weld Definition	99
8.2.4.1 Identification.....	99
8.2.4.2 Type Specification	100
8.2.4.3 Weld Position and Sheet Metal Parameters.....	102
8.2.4.3.1 Parameters Assigned to a Specific Sheet of the Flange.....	102
8.2.4.3.2 Welding Position.....	103
8.2.5 Butt Joint.....	108
8.2.5.1 Sheet Parameters	108
8.2.5.2 Weld Parameters	108
8.2.5.3 Attributes	108
8.2.5.4 Element “weld_position”.	108
8.2.5.5 Element “sheet_parameter”.....	110
8.2.6 Corner Weld	110
8.2.6.1 Sheet Parameters	110
8.2.6.2 Weld Parameters	111
8.2.6.3 Attributes	111
8.2.6.4 Element “weld_position”.	111
8.2.6.5 Element “sheet_parameter”.....	113
8.2.7 Edge Weld	113
8.2.7.1 Sheet Parameters	114
8.2.7.2 Weld Parameters	114
8.2.7.3 Attributes	114
8.2.7.4 Element “weld_position”.	114
8.2.7.5 Element “sheet_parameter”.....	115
8.2.8 I-Weld.....	116
8.2.8.1 Sheet Parameters	116
8.2.8.2 Weld Parameters	116
8.2.8.3 Attributes	116
8.2.8.4 Element “weld_position”.	117
8.2.8.5 Element “sheet_parameter”.....	117
8.2.9 Overlap Weld	118
8.2.9.1 Simple Overlap Weld	118
8.2.9.2 Single Sided Double Overlap Weld	119
8.2.9.3 Double Sided Double Overlap Weld	119
8.2.9.4 Attributes.....	120
8.2.9.5 Element “weld_position”.....	120
8.2.9.6 Element “sheet_parameter”.....	122
8.2.10 Y-Joint.....	123

8.2.10.1	Sheet Parameters	123
8.2.10.2	Weld Parameters	123
8.2.10.3	Attributes.....	123
8.2.10.4	Element “weld_position”.....	124
8.2.10.5	Element “sheet_parameter”.....	125
8.2.11	K-Joint.....	126
8.2.11.1	Sheet Parameters	126
8.2.11.2	Weld Parameters	126
8.2.11.3	Attributes.....	127
8.2.11.4	Element “weld_position”.....	127
8.2.11.5	Element “sheet_parameter”.....	129
8.2.12	Cruciform Joint.....	129
8.2.12.1	Sheet Parameters	130
8.2.12.2	Weld Parameters	130
8.2.12.3	Attributes.....	130
8.2.12.4	Element “weld_position”.....	130
8.2.12.5	Element “sheet_parameter”.....	133
8.2.13	Flared Joint.....	133
8.2.13.1	Attributes.....	134
8.2.13.2	Element “weld_position”.....	134
8.2.13.4	Element “sheet_parameter”.....	134
8.3	Adhesive Lines	135
8.4	Hemming Flanges	137
8.4.1	Introduction	137
8.4.2	Definition of element <hemming />	138
8.5	Sequence Connections.....	141
9	2D CONNECTIONS	144
9.1	Generic Definitions	144
9.1.1	Identification	144
9.1.2	Connection Face.....	144
9.1.3	Type Specification	146
9.2	Adhesive Faces	147
10	FUTURE EXTENSIONS	149
10.1	Additional parameters for spot and seam welds	149
10.2	Other relevant and new joint types	149
11	DISCLAIMER.....	150
12	REFERENCES.....	151

Foreword to χ MCF 3.0

Joining technologies play an important role in modern automotive structures whose designs are mainly determined by weight and cost. This leads to the application of a variety of materials with a wide spectrum of mechanical and physical properties and, hence, requires joining technologies specifically tailored to the joining partners. These joining technologies have to be cost-optimized and to deliver desired mechanical properties like high fatigue and crash strengths.

In order to characterize a specific joint completely, numerous attributes like geometry, process parameters, strengths etc. are necessary. Different divisions like CAD, CAE Manufacturing or different OEMs use often only a subset of these attributes. Therefore, each engineering function has been using their own way to describe joints leading to incompatibilities or gaps in data transfers. In order to allow a seamless data flow from CAD data creation through the various virtual and physical optimization & verification steps (CAE, Testing, Feasibility) and finally to manufacturing control processes a comprehensive standard for the characterization of joints is required.

More than 10 years ago, the AK25 (Working Group 25) "Joint Technologies" of the Research Association of German Automotive Industry (FAT/VDA) began to develop the standard χ MCF (Extended Master Connection File) for joints, in co-operation with several leading vendors for pre- and post-processors in CAE and fatigue software. While the focus was around a seamless process especially for fatigue prediction of welds in the beginning, meanwhile, the project has significantly grown into a cross-functional standard that can also support the definition and automated virtual builds of full vehicle assemblies.

χ MCF is aimed at describing the major attributes of a joint in a neutral way. Information contained in χ MCF can be shared by all participants in the development process (e.g. geometrical data for CAD, CAE and CAM, process parameters for CAM etc.). Once fully deployed, all application software uses the same χ MCF. No interface or data conversion software is required and there is no risk of information loss.

The documents χ MCF 2.1 and 3.0 are results of a project funded by FAT. χ MCF 2.1 comprises the definition of all joint types (welds, adhesives, bolts and screws etc.) which were already supported either by the Pre-processor Ansa (Beta CAE Systems) or Medina (T-Systems). χ MCF 3.0 is a further extension of χ MCF 2.1 with more detailed attributes & parameters and introduces also a new element `<custom_attributes>` by which users can enrich a specific joint with useful information which is not (yet) defined by the standard. All relevant joint types for practical application are included in χ MCF 3.0:

- 1) 0d-joints: spot welds, robscans, rivets, bolts and screws, gum drops, clinches, heat stakes/thermal stakes, clips/snap joints, nails.
- 2) 1d-joints: seam welds, adhesive lines, hemming flanges, sequence connections.
- 3) 2d-joints: adhesive faces.

The CAE engineers will benefit from the implementation of χ MCF 3.0 in the pre-processors for a fully automated assembly of FE-models using batch-meshing and auto-assembly techniques, or the implementation in fatigue solvers for accelerated fatigue evaluations of complex structures. It is hopeful that it will find a broad application in the complete process chain CAD-CAE-CAM.

Genbao Zhang, Wolfsburg, Germany
Matthias Weinert, Cologne, Germany
2016

List of Figures

Figure 1: Seam weld as 1-dimensional joint.....	20
Figure 2: Topological Relations between Parts and Assemblies	21
Figure 3: Product Structures Fitting to Previous Figure.	21
Figure 4: The Development Process.....	22
Figure 5: χ MCF as a Platform for Connection Information in the Complete Development Process	22
Figure 6: Weld line crossing tailored blank vs. weld line crossing physical gap.....	27
Figure 7: Robscans with Different Rotation Angles; Two of them Mirrored	52
Figure 8: Rivet head types.....	55
Figure 9: Cross Section of a blind rivet.....	57
Figure 10: Thick and Thin Assembling.....	58
Figure 11: Fastening Soft and Hard	58
Figure 12: Cross Section of a Self-Piercing Rivet	59
Figure 13: Self-piercing rivet setting apparatus.....	59
Figure 14: Dimensions of Solid Rivets	61
Figure 15: Clinch allowance of solid rivet.....	62
Figure 16: Cross section of a SWOP Rivet	63
Figure 17: Bolts and Screws.....	65
Figure 18: Different Screw Forms.....	65
Figure 19: Definition of Length and Head Sizes	66
Figure 20: Definition of lead, pitch and starts of a thread.	66
Figure 21: Bolt with welded nut	77
Figure 22: Bolt with free nut	77
Figure 23: Screw without nut	77
Figure 24: Welded stud with free nut	78
Figure 25: Plain stud	78
Figure 26: Process of Flow Drill Screwing.....	80
Figure 27: Measures of applied FDS.....	80
Figure 28: Pre-machined or clearance hole in FDS connection	81
Figure 29: Pilot hole on sheet metal	81
Figure 30: Schematic representation of the clinching operation.....	83
Figure 31: Clinch Joint Dimensions.....	83
Figure 32: TOX (left) and BTM's Tog-L-Loc system.....	84
Figure 33: Cross Section of a Heat Stake	86
Figure 34: A "Hairpin Clip".....	88
Figure 35: Internal and External Circlips	88

Figure 36: Clips Pushed into a Hole	89
Figure 37: Clips Sliding onto a Flat Surface	89
Figure 38: RIVTAC® Nail.....	91
Figure 39: Cross Section of a Nail, Connecting Two Sheets	91
Figure 40: Weld Line Changing from Y-Joint to Overlap-Joint	96
Figure 41: Longitudinal stiffener, top view	96
Figure 42: Seam weld types and attributes.....	98
Figure 43: χ MCF Structure of a Seam Weld (<i>connection_1d</i>)	99
Figure 44: Sheet Parameters vs. Weld Position Parameters.....	102
Figure 45: Welding Position of a Y-Joint.....	104
Figure 46: Welding Position vector direction and length.....	105
Figure 47: Butt Joint Sheet Layout	108
Figure 48: Butt Joint Weld parameters	108
Figure 49: Corner Weld Sheet Layout	110
Figure 50: Corner Weld Parameters.....	111
Figure 51: Edge Weld Sheet Layout.....	114
Figure 52: Edge Weld parameters.....	114
Figure 53: I-Weld Sheet Layout	116
Figure 54: I-Weld Parameters	116
Figure 55: Overlap Weld Sheet Layout.....	118
Figure 56: Overlap Weld Parameters	118
Figure 57: Single Sided Double Overlap Weld.....	119
Figure 58: Overlap Weld Parameters	119
Figure 59: Double Sided Double Overlap Weld.....	120
Figure 60: Parameters of Double Sided Double Overlap Weld	120
Figure 61: Y-Joint Sheet Layout	123
Figure 62: Parameters of Y-Joint Weld.....	123
Figure 63: K-Joint Sheet Layout.....	126
Figure 64: Parameters of K-Joint Weld.....	126
Figure 65: Cruciform Joint Sheet Layout	130
Figure 66: Parameters of Cruciform Joint	130
Figure 67: Flared Joint Sheet Layout	133
Figure 68: Parameters of Flared Joint Weld	133
Figure 69: The Three Regions of a Hemming	137
Figure 70: Path Changes and Width Changes in Hemming Flanges	137
Figure 71: Adhesive Path Differs from Root Path	138

Figure 72: Reinforcements need to be considered as Part of the Inner Panel	138
Figure 73: Sequence without margin	141
Figure 74: Sequence with margin and spacing.....	141
Figure 75: Margin relaxation	141
Figure 76: Spacing relaxation	141
Figure 77: Picture of an adhesive face	147

List of Tables

Table 1: Nested elements of element <xmcf/>	28
Table 2: XML-specification of <units/>	29
Table 3: XML-specification of <appdata>.....	31
Table 4: XML-specification of element <femdata>	33
Table 5: Nested elements of element <femdata>.....	33
Table 6: Attributes elements of element <femdata>.....	33
Table 7: Nested elements of element <CAE_DATA>	33
Table 8: Attributes of element <connection_group/>.....	34
Table 9: Nested elements of element <connection_group/>.....	34
Table 10: Nested elements of <connected_to>.....	35
Table 11: Attributes of element <part/>	35
Table 12: Attributes of element <assy/>	36
Table 13: Nested elements of element <contact_list/>	37
Table 14: Nested elements of element <contact>.....	37
Table 15: Attributes of element <partner/>	38
Table 16: Attributes of element <contact>.....	39
Table 17: Nested elements of element <connection_list>	39
Table 18: Nested elements of element <custom_attributes_list/>	44
Table 19: Attributes of <custom_attributes/> element	44
Table 20: Nested elements of element <custom_attributes/>	45
Table 21: Attributes of <string/> element.....	45
Table 22: Attributes of <real/> element	45
Table 23: Attributes of <integer/> element	45
Table 24: Attributes of <string_list/> element.....	45
Table 25: Attributes of <value/> element inside <string_list/>.....	46
Table 26: Attributes of <real_list/> element.....	46
Table 27: Attributes of <value> element inside <real_list/>.....	46
Table 28: Attributes of <int_list/> element.....	46
Table 29: Attributes of <value/> element inside <real_list/>.....	46
Table 30: Attributes of element <connection_0d/>.....	49
Table 31: Text values of element <loc>	50
Table 32: Attributes of elements <normal_direction/>& <tangential_direction/>.	50
Table 33: Nested elements of element <connection_0d/>.....	51
Table 34: Nested element of <connection_0d/> for <spotweld/>	51

Table 35: Attributes of element < <i>spotweld</i> >.....	51
Table 36: Nested element of < <i>connection_0d</i> > for < <i>robscan</i> >.....	53
Table 37: Attributes of element < <i>robscan</i> >	53
Table 38: Nested elements of element < <i>robscan</i> >	54
Table 39: < <i>connection_0d</i> > for < <i>rivet</i> >	55
Table 40: Attributes of element < <i>rivet</i> >.....	55
Table 41: Nested elements of element < <i>rivet</i> >.....	56
Table 42: Attributes of element < <i>blind</i> >	56
Table 43: Attributes of element < <i>self_piercing</i> >.....	60
Table 44: Pictures of all Solid Rivets.....	61
Table 45: Attributes of element < <i>solid</i> >	62
Table 46: Attributes of element < <i>swop</i> >	64
Table 47: Nested elements of element < <i>contact_list</i> >.....	67
Table 48: Attributes of element < <i>contact</i> >	67
Table 49: Nested elements of < <i>threaded_connection</i> >.....	69
Table 50: Attributes of element < <i>threaded_connection</i> >.....	70
Table 51: Nested elements of element < <i>threaded_connection</i> >	71
Table 52: Attributes of element < <i>washer</i> >.....	72
Table 53: Attributes of element < <i>nut</i> >.....	73
Table 54: Nested elements of element < <i>nut</i> >.....	73
Table 55: Attributes of element < <i>bolt</i> >	74
Table 56: Nested elements of element < <i>bolt</i> >	74
Table 57: Attributes of element < <i>screw</i> >	79
Table 58: Nested elements of element < <i>screw</i> >	79
Table 59: Attributes of element < <i>flow_drilled</i> >	81
Table 60: Nested elements of < <i>connection_0d</i> > for < <i>gumdrop</i> >	82
Table 61: Attributes of element < <i>gumdrop</i> >	82
Table 62: Nested elements of < <i>connection_0d</i> > for < <i>clinch</i> >	84
Table 63: Attributes of element < <i>clinch</i> >.....	84
Table 64: Nested elements of element < <i>clinch</i> >.....	85
Table 65: Nested elements of < <i>connection_0d</i> > for < <i>heat_stake</i> >.....	87
Table 66: Attributes of element < <i>heat_stake</i> >	87
Table 67: Nested elements of < <i>connection_0d</i> > for < <i>clip</i> >.....	89
Table 68: Attributes of element < <i>clip</i> >	89
Table 69: Nested elements of element < <i>clip</i> >	90

Table 70: Nested elements of <connection_0d/> for <nail/>.....	91
Table 71: Attributes of element <nail/>	92
Table 72: Nested elements of element <nail/>	93
Table 73: Nested elements of <loc_list>.....	94
Table 74: Attributes of element <loc/>.....	94
Table 75: Nested elements of element <connection_1d/>.....	95
Table 76: Attributes of element <connection_1d/>.....	99
Table 77: Nested elements of element <seamweld/>.....	100
Table 78: Attributes of element <subtype/>	101
Table 79: Nested elements of element <subtype/>	101
Table 80: Attributes of element <sheet_parameter/>	103
Table 81: Attributes of element <weld_position/>.....	104
Table 82: Default values of attribute "filler", dependent from attribute "technology"	107
Table 83: Parameters of Butt Joint Weld	108
Table 84: Attributes of element <weld_position/> for Butt Joint	109
Table 85: Attributes of element <sheet_parameter/> for Butt Joint	110
Table 86: Parameters of Corner Weld.....	111
Table 87: Attributes of element <weld_position/> for Corner Weld	112
Table 88: Values of Attribute <i>section</i>	112
Table 89: Values of Attribute <i>angle</i>	112
Table 90: Attributes of element <sheet_parameter/> for Corner Weld	113
Table 91: Parameters of Edge Weld.....	114
Table 92: Attributes of element <weld_position/> for Edge Weld	115
Table 93: Attributes of element <sheet_parameter/> for Corner Weld	116
Table 94: Parameters of I-Weld.....	116
Table 95: Attributes of element <weld_position/> for I Weld	117
Table 96: Attributes of element <sheet_parameter/> for I Weld.....	118
Table 97: Parameters of Overlap Weld	119
Table 98: Parameters of Single Sided Double Overlap Weld	119
Table 99: Parameters of Double Sided Double Overlap Weld	120
Table 100: Attributes of element <weld_position/> for Overlap Weld.....	121
Table 101: Attributes of element <sheet_parameter/> for Overlap Weld	122
Table 102: Parameters of Y-Joint	123
Table 103: Attributes of element <weld_position/> for Y Joint	124
Table 104: Value Dependency of Attribute <i>thickness</i>	125
Table 105: Attributes of element <sheet_parameter/> for Y-Joint.....	126

Table 106: Parameters of K-Joint	127
Table 107: Attributes of element < <i>weld_position</i> /> for K Joint.....	127
Table 108: Value Dependency of Attribute <i>thickness</i>	128
Table 109: Attributes of element < <i>sheet_parameter</i> /> for K Joint	129
Table 110: Parameters of Cruciform Joint	130
Table 111: Attributes of element < <i>weld_position</i> /> for Cruciform Joint.....	131
Table 112: Value Dependency of Attribute <i>thickness</i>	131
Table 113: Attributes of element < <i>sheet_parameter</i> /> for Cruciform Joint.....	133
Table 114: Parameters of Flared joint.....	133
Table 115: Attributes of element < <i>weld_position</i> /> for Flared Joint.....	134
Table 116: Attributes of element < <i>sheet_parameter</i> /> for Flared Joint	134
Table 117: Attributes of < <i>connection_1d</i> />	135
Table 118: Nested elements of < <i>connection_1d</i> />.....	135
Table 119: Attributes of element < <i>adhesive_line</i> />.....	135
Table 120: Attributes of < <i>connection_1d</i> /> for < <i>hemming</i> />	138
Table 121: Nested elements of < <i>connection_1d</i> /> for < <i>hemming</i> />	139
Table 122: Attributes of element < <i>hemming</i> />	139
Table 123: Nested elements of element < <i>hemming</i> />	139
Table 124: Attributes of element < <i>region</i> />.....	139
Table 125: Nested elements of element < <i>region</i> />.....	140
Table 126: Nested elements of < <i>connection_1d</i> /> for < <i>sequence_connnection_0d</i> />	142
Table 127: Nested elements of < <i>sequence_connnection_0d</i> />.....	143
Table 128: Attributes of element < <i>sequence_connnection_0d</i> />	143
Table 129: Attributes of < <i>connection_2d</i> />	144
Table 130: Nested elements of < <i>loc_list</i> >.....	145
Table 131: Attributes of element < <i>loc</i> />.....	145
Table 132: Nested elements of element < <i>face_list</i> >.....	145
Table 133: Attributes of element < <i>face</i> />	146
Table 134: Nested elements of < <i>connection_2d</i> />.....	146
Table 135: Attributes of element < <i>connection_2d</i> />.....	147
Table 136: Nested elements of element < <i>connection_2d</i> />.....	147
Table 137: Attributes of element < <i>adhesive_face</i> />.....	147

Members of the Group xMCF (2016)

Dr. Thomas Bruder (BMW Group)

Mr. Nikolaos Economidis (BETA CAE Systems)

Mr. Harald Fleischer (BMW Group)

Dr. Carsten Franke (PROSTEP AG)

Dr. Christian Gaier (Magna Powertrain, ENGINEERING CENTER STEYR GmbH & Co KG)

Mr. János Golumba (Ford Werke GmbH)

Dr. Michael Hack (Siemens Industry Software GmbH & Co. KG)

Mr. Daniel, Koenen (Volkswagen Osnabrück)

Mr. Peter Nuhn (Siemens Industry Software GmbH & Co. KG)

Dr. Georg Tröndle (Volkswagen AG)

Mr. Michael Tryfonidis (BETA CAE Systems)

Mr. Vincent Dampure (Altair Engineering GmbH)

Dr. Stephan Vervoort (Hottinger Baldwin Messtechnik GmbH)

Dr. Mathias Weinert (Ford Werke GmbH)

Dr. Genbao Zhang (speaker of the group, Volkswagen AG)

Financial Sponsors

The elaboration of this document (version 3.0) is financially supported by *VDA-FAT (Germany Automotive Association, Berlin)*

Version History of χ MCF

V 1.0 - June 2006 (P. Mikolaj†)

V 1.1 - September 2011 (P. Nuhn)

V 1.2 - October 2012 (P. Nuhn, G. Zhang, C. Franke)

V 2.0 - November 2013 – September 2014 (P. Nuhn, C. Franke, G. Zhang, T. Herrmann, J. Golumba)

Incorporating the decisions of working group meeting 30th October, 2012.

Incorporating the Feedbacks from 14th July, 2014 to 7th September, 2014.

V 2.1 - October 2014 - March 2015 (N. Economidis, C. Franke (editor in chief), J. Golumba, M. Tryfonidis, G. Zhang)

Adding following connection types:

- Adhesive points, lines and faces,
- Bolts and screws,
- General rivets and self-piercing rivets,
- Hemming flanges,
- Robscans,
- Sequence connections

Clarifying following connection types:

- Spotwelds

Applying changes suggested during meeting of 6th November, 2014 in Leinfelden.

V 3.0 - November 2015 - January 2016 (N. Economidis, C. Franke, J. Golumba)

Adding following connection types:

- Blind and solid rivets
- Flow drilled screws
- Clinches
- Heat stakes
- Clips / snap joints
- Nails

Incorporating the decisions of working group meeting 26th November, 2015.

Incorporating the Feedbacks from 26th November, 2015 to 14th January, 2016.

Adding following sections:

- 6.5 Distinction between <custom_attributes_list/> and <appdata/>

1 Introduction

1.1 Motivation

An automobile is a complex system like many other technical systems. It consists typically of thousands of individual parts which are assembled by joints. Depending on the involved materials and the manufacturing process, different joint types are often necessary in order to obtain an economical and reliable complete structure.

A wide range of joints are used in the automotive industry: From welded joints to screws, adhesives and mechanical joints like clinching etc. They differ from each other not only in their physical and mechanical properties but also in their geometrical shapes and manufacturing processes.

The development of a new car typically passes through the stages of design, simulation and testing until it reaches SOP (start of production) at the end. The information concerning the joints arises peu à peu during the progress of the development. The types of the required information may vary. For designers and CAE-engineers, geometrical information may be of primary interest whereas additional information like fillers etc. is important for prototyping and production, and so on.

An effective development process relies on the efficient management of connection information, which is extensive for a system like an automobile. This is only possible, if the involved information is standardized and mapped into a database or file. The advantage of a standard is evident: Typically, different systems like CAD, CAE and CAM are employed in the developing process. Based on a standard, there is no necessity to convert the same information from one system to the others which are often interested in only a part of the joint information. This prevents possible errors and reduces the development effort of the involved systems.

1.2 MCF at Ford

Up to the present day, there is still no standard which is capable to describe all the joint information needed in the development process. The companies use internal file formats to hold the joint information, which is typically in-house and incomplete. The **Master Connection File (MCF)** by Ford is one of these formats. The MCF format is based on the XML-standard and covers only few joint types (cf. [1]).

1.3 From MCF to xMCF - The Scope of the Document

Facing the difficulty that joints are represented or realized quite differently in different CAE tools, FAT-AK 25¹ made the proposal to develop a standard for connections and joints in cooperation with the vendors of CAE-software.

The evaluation of existing formats revealed that the MCF format by Ford was the most suitable basis for further developments and extensions. In order to distinguish it from the original Ford-MCF, the FAT-format was named the **Extended Master Connection File**, abbreviated as xMCF (read: chi-M-C-F) or xMCF (read: x-M-C-F).

¹ Working group 25 for joining technologies of the German Research Association of Automotive Technologies.

In 2005, the consortium decided to begin with the extension of MCF to seam welds. There were several reasons for this decision. First of all, the demand for the fatigue evaluation of seam welds was increasing rapidly. Furthermore, there are a wide variety of weld types with partly complex geometrical shapes. The proper description of these welds meant a big challenge. The successful treatment of seam welds would lay the foundation for the integration of any other joint type.

The current document provides a complete documentation of the spot weld and welded seams after some basic properties and features of χ MCF are explained. Further joints like rivets and adhesives etc. will be treated in the near future.

The current version of χ MCF does not cover all information relevant for the joints treated here. Thanks to the simple extensibility of χ MCF, additional information can be integrated on demand.

2 Design Principles and Basic Features of χ MCF

The Extended Master Connection File (χ MCF) is a container for connection information of a complex structure (here the focus is put on automobiles).

Typically, a complex structure consists of a lot of individual parts which are joined together. Unconnected parts are amorphous. Connections establish a topology between the parts. It is thus evident that any efficient database or container designed to gather connection information should be equipped with structures which are able to map this kind of topology between the parts.

Real development processes are complicated. The amount of connection information is huge. It is intended to promote χ MCF to become an industry standard in the long term. This demands certain rigorousness of χ MCF. On the other hand, some flexibility is desired in order to enable an easy integration of χ MCF into different processes. This makes clear that χ MCF needs a sophisticated design.

This chapter explains the design principles and some basic features of χ MCF, which are important for a proper understanding and straight-forward future extensions.

2.1 Design Principles

The design of χ MCF is guided by the following principles:

- 1) χ MCF should be able to *completely* and *unambiguously* describe all relevant connections/joints used in the automotive industry. These include spot welds, seam welds, rivets and adhesives, and so on.
- 2) It should be able to address all kind of processes, let it be in CAD, CAE and CAM, on the long run.
- 3) χ MCF contains *only* information relevant to connections. Hierarchical product structure, assembly sequence, part variants etc. are *not* subject of χ MCF. Such kind of information needs different vessels for propagation. However, χ MCF may *refer* to such “external” information, e. g. part codes.
This principle grants χ MCF’s flexibility for application to any kind of process variants, established at different automotive OEMs.
- 4) The format has to be flexible and easy to extend to any future joint types and applications.
- 5) χ MCF is built upon the industry standard XML.
- 6) Connection data are unique.
- 7) The content of χ MCF may be incomplete to a certain extend. This addresses the fact that new data is created and needs to be stored throughout the course of CAx processes, without changing its vessel.
- 8) χ MCF follows the max-min principle: It contains information as much as necessary, at the same time, as little as possible.
- 9) At any certain stage of any involved process, connectors can be reconstructed from χ MCF without loss of data or ambiguities.
- 10) The format description is kept compact. Elements are reused, whenever possible.
- 11) Application specific data can be stored in χ MCF even without standardization: χ MCF offers corresponding “empty” containers which can be assigned to any certain connector or to the complete collection / file.
- 12) Due to its simplicity and extensibility, χ MCF forms a good candidate for long-term archiving connector information.

Using XML deserves some comments. XML is by itself an industry standard and human readable. XML allows for contents getting certain structure which is the precondition to easily map the topology of connectivity of complex structures like automobiles.

2.2 Idealization of Joints

Different joints possess different characteristics. They may differ from each other by their geometrical shapes, mechanical properties like strengths for different loadings, manufacturing processes etc.

To allow an efficient description of joints, some simplifications and idealizations are necessary. The way chosen by χ MCF is to classify joints by their most basic and mandatory attribute, namely its geometrical dimensions. Thus, there are 0-, 1- and 2-dimensional joints in χ MCF.



Figure 1: Seam weld as 1-dimensional joint

A spot weld is treated as a 0-dimensional joint in χ MCF. In this way, a (an idealized) spot weld is geometrically described by its coordinate vector x and its diameter d as an additional attribute. Besides spot welds, there are more joints which can be treated as 0-dimensional.

A seam weld is a representative of 1-dimensional joints, see Figure above. It is characterized by a curve describing its spatial course and additional parameters (attributes) determining its sectional shape perpendicular to the curve. Details are referred to later chapters.

Similarly, adhesive joints are idealized as 2-dimensional surfaces. Details are subjects of future extensions.

2.3 Reconstruction of Joints from χ MCF

An important topic worthwhile to mention is the *reconstruction* of the joints. It is crucial that it is possible to reconstruct the corresponding joint in its idealized form uniquely by means of the introduced parameters and attributes. In case of spot weld, a unique reconstruction is possible by the coordinate vector x and the diameter d , plus the sheet thicknesses which by themselves are not a constituent of χ MCF (recall χ MCF contains only information relevant to joints), but of the corresponding CAD or CAE model.

2.4 Description of Topology

As mentioned before, a complex structure arises by connection of parts and sub-structures (assemblies). The connections introduce a topology between the individuals. In the present context, the description of the topological relations is not necessarily unique, a-priori. For example, the structure shown in Figure below could be described by the following sentences (alternatives)

- 1) Part (or Assembly) A is joined to Part B by the seam weld 1 along the curve l_1 and the spot weld 1 at the position x_1 ...
and
Part (or Assembly) A is connected to Part C by the adhesive AD_x in the area A_x , etc..

- 2) The seam weld 1 joins Part (or Assembly) A to Part B along the curve I_1
and
spot weld 1 connects Part (or Assembly) A to Part B at the position x_1 etc..

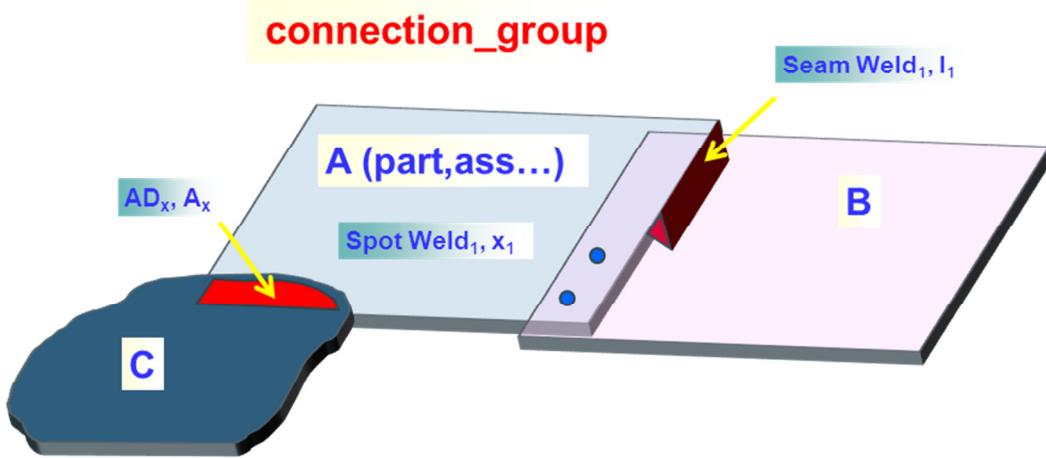


Figure 2: Topological Relations between Parts and Assemblies

The alternative 1) is adopted by χ MCF. The description is mapped into XML by using an element tagged `<connection_group>`. A `<connection_group>` comprises all joints which connect the same parts (or assemblies). Details are referred to later chapters. Here one of the merits of employing XML becomes apparent.

It is to mention that frequently more than two parts are joined together. A spot weld can e.g. join three sheets, a screw even more. Such situations have to be taken into account, too.

According to design principle 3), overall product structure cannot be reproduced from χ MCF. For example, any of the following product structures would equally fit to Figure 2:

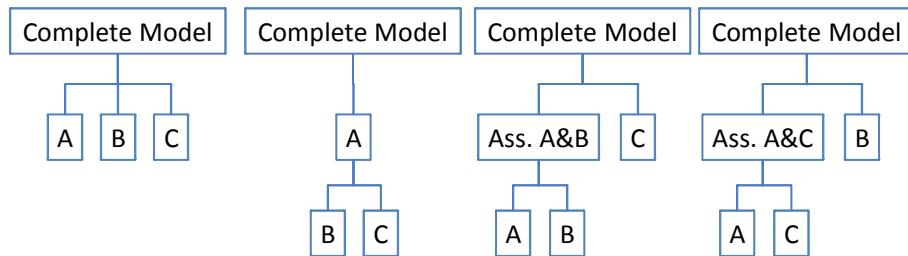


Figure 3: Product Structures Fitting to Previous Figure.

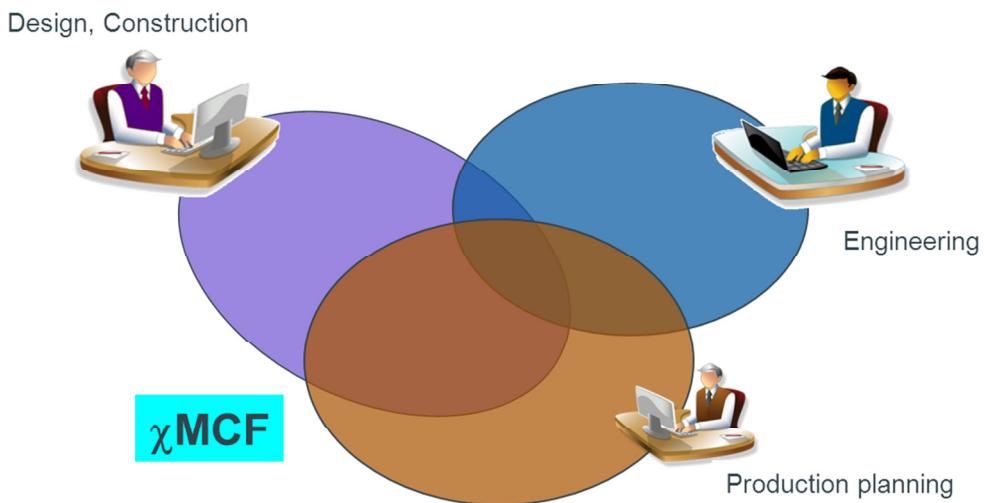
And this list is far from being complete.

2.5 χ MCF in the Development Processes

The typical development process is a long chain involving many (maybe overlapping) single steps like design, construction, prototyping, simulation, testing, production planning, etc., see Figure 4. Depending on the individual manufacturer considered, information of connections and joints arises at different stages of the process and comes from different parties (Figure 5). An efficient handling and management of this information can only be guaranteed by a (common) database/container which contains the information *uniquely*. This shall be guaranteed by a standard like χ MCF.



Figure 4: The Development Process



**Figure 5: xMCF as a Platform for Connection Information
in the Complete Development Process**

A careful examination of Figure 5 enables one to get more feeling and understanding about how the work with xMCF in a real process could look like: xMCF is a structured set which can be divided into several overlapping subsets. Each subset contains a part of connection information which is of interest for a certain party, e.g., simulation or planning. The intersection of all subsets contains information which is of interest for all other parties involved, e. g. coordinates and flange partners.

As mentioned before, the information contained in xMCF is not necessarily complete, at least not at an early stage of the development process. Rather its content grows while the process is advancing. Defining the individual joint and filling up the container thus build up a continuous process. As shown in Figure 5, connection information could be created by any of the involved parties (design, construction, engineering, planning, etc.). The common situation is that each party contributes part of information (geometrical, technological etc.) defining a specific joint. Merging of the partial information leads to the complete characterization of the joint. Therefore, xMCF is an ideal tool to enable this dynamic process since filling up xMCF means merging information.

Figure 5 makes also clear that connection information (full or partial) is available to everyone once it is defined and stored in xMCF. Thus, unnecessary duplication of effort is avoided automatically. Typically different parties work in different environments using different software tools. Provided all systems support xMCF, transfer of data from one format to another will be not necessary any more. This will save development cost and avoid loss of data caused by the transfer.

Information contained in xMCF can be used to automate many tasks in the development and thus to enhance efficiency:

- **Automatic CAE assembly**

Meanwhile most FE-preprocessors are able to mesh parts automatically in the batch-meshing mode. An automated assembly can be realized by the connection information contained in χ MCF.

- **Automatic Programming of Welding Robot**

Based on χ MCF, welding robots can be programmed automatically.

An essential feature of χ MCF is that it contains only information relevant to the joints. No data are included which are dependent on the process. Hence it is relatively easy to implement χ MCF into any real process. Depending on the application, it is possible to use χ MCF as a stand-alone database or integrate χ MCF into an even more comprehensive database.

3 Key-words of XML specification

3.1 Key-words

The carrier of information in a *xMCF* file is an element which can be equipped with some attributes and child elements. Elements and attributes are defined by their names (identifiers) and values (information).

By the XML standard, values assumed by elements can be distinguished by their types like boolean, float, double, string, date, etc. The same applies to attributes. The user can determine how elements and attributes are used (optional, required or prohibited). If necessary, the frequency of occurrence of elements with the given name (i.e., number of siblings of identical names) can be restricted (in XML schema, this is specified by the attributes *minOccurs* and *maxOccurs*).

In accordance with the XML standard (version 1.0²), the following key-words are used in the current document to characterize the elements and attributes:

- Type
- Value Space
- Default
- Use
- Multiplicity (corresponds to the attributes *minOccurs* and *maxOccurs* of the element *<xs:element>* of XML schema)
- Restrictions (corresponds to the element *restriction* of XML schema)

The type of the value of an element or attribute is specified by the key-word *Type*. The numerical ID of a property (attribute “pid”) of a *<part>* opening tag is typically an integer which is a built-in type of XML standard.

The most common types are:

- *xs:string*
- *xs:decimal*
- *xs:integer*
- *xs:float*
- *xs:boolean*
- *xs:date*
- *xs:time*

Note: The maximum number of decimal digits you can specify is 18.

However, only positive integers are usually used in this context. That means, the possible value of ID of the type *integer* is restricted. To specify the values which are assumable by an element or an attribute, the key-word *Value Space* (a set) is used. The Value Space can be given as an

² Up to now, only versions 1.0 and 1.1 of XML exist, where 1.1 is *not* widely used. Hence, most systems still create XML 1.0 files. (For differences see <http://www.w3.org/TR/xml11/#sec-xml11>.)

enumeration (a finite set) or a set defined explicitly. E.g. positive integer is symbolized by > 0 whereas a float between 0.0 and 1.0 is given by [0.0, 1.0], similar to mathematics.

Some elements and attributes obtain default values if they are not given explicitly in the xMCF file. The default values adopted are specified by the keyword *Default*.

In this document, the special type “alphanumeric” is frequently used for labels of parts and assemblies, which deserve a careful discussion. In the CAD world, a label is synonymous with the name of a part, a geometric object etc. Not only letters “[A-Za-z]”, but also numbers “[0-9]” and other special characters like “[_.\\$#\t]” and more are used for labels. Sometimes, first character is restricted to “[A-Za-z]”. Thus, it is difficult to give an exact definition for the type “alphanumeric” which would fit to the individual need. Fortunately, using XML’s “encoding” attribute, even non-ASCII characters can be handled easily, e.g. Arabic, Chinese, Cyrillic, Greek, Hebrew, etc. Nevertheless, as sort of general recommendation, labels should not start or end with white space.

The key-word *Use* specifies, whether an element or an attribute is optional, required or prohibited. The frequency of the occurrence of an element or attribute is defined by *Multiplicity* e.g. in the form: $\text{minOccurs} \leq \text{Multiplicity} \leq \text{maxOccurs}$. Any additional restrictions imposed on an element or an attribute are specified by the key-word *Restrictions*.

3.2 Operators

As explained above, the individual use of some elements or attributes may be optional. But some of them are coherent (thus in certain sense redundant). An important example is the label and pid of a part or an assembly. They represent the same part (except for e.g. tailored blanks). One can use the one or the other or both to identify a part.

4 Parts, Properties and Assemblies

xMCF describes, how parts, properties and assemblies are connected by joints in a pre-defined way. Hence, we need a clear understanding about what a part, property or assembly actually is in our context.

4.1 Parts

Parts are logical groupings of 3D objects, on first hand. Their objective is to provide a general nomenclature of the pieces which form a certain product. This nomenclature allows communications between all stakeholders of all involved processes.

Typically, it is assumed that parts do not disintegrate into several physical compounds.

Parts can be instantiated at different locations of a product, e. g. wheels in a car etc.

Parts can be mirrored at a symmetry plane of the model, e. g. front doors of a car.

Parts can contain other parts (sub-parts): A car, for instance is made of body in white, power train, doors and claps etc. A door is made of an outer sheet, an inner sheet, a window with its mechanics, some crash enforcements etc. The mechanics of a window are made of some guiding rails, an electric motor and so on.

Hence, in sense of graph theory, parts form a tree (if their instances are considered) or a directed, cycle free graph. Parts without sub-parts are called the “leaves” of this tree or graph.

If a part is mentioned in a list, not only its own content (e. g. finite elements) is addressed, but also all contents of its sub-parts and their children, down to the lowest level (leaves) of the part graph.

4.1.1 Part Labels

A part is uniquely identified by its *label*, up to ditto-parts. Connectors within a connection group that refers to ditto parts must be able to “detect” the “correct” part instance according to their respective geometrical location.

We assume that mirror parts have other part labels than their “base” parts.

Note: In most CAx processes, parts actually have two string attributes: One label describing the name and/or usage of a part in a human readable form, and another one used for indexing this item in the OEM’s “part store”. The latter one typically consists of only few characters (some 8 to 12, e. g.), resembles more to a number than to a name, and hence is not human readable. In our context, we refer to the latter one, if we say “part label”.

4.2 Properties

In CAE, properties are a concept for assigning physical behavior to a number of finite elements. Hence, any finite element can have at most one property. However, there frequently are elements without such properties (RBEs, masses, etc.). In most solvers, properties are uniquely identified by positive integers, so called property IDs or short: PIDs.

Even, if finite elements of different parts have same physical behavior (let's say, left and right wing of a car), they usually have assigned different PIDs. This can be considered as reminiscence to ancient times, when parts just have not been invented. PIDs were also used for administrative purposes, then.

However, for χ MCF, PIDs are just alternative, non-recursive means for addressing collections of elements.

One specific part frequently consists of one specific property (PID), only. However, there are important exceptions:

- A tailored blank is a metal sheet which consists of several pieces of simple sheets joined together. Both, the thicknesses and the materials of the individual sheets, may differ. Nevertheless, a tailored blank is one single part from the χ MCF point of view. Since one PID would not provide a name for the *complete* part, the part label has to be used, or else an assembly of several PIDs.
- Sometimes, a cast part can be treated with shell formulation in its thin areas, whereas solid elements (with different PIDs) are used in other areas.
- Due to e. g. stamping processes, physical behavior and thickness may vary even within one originally homogeneous sheet metal, requiring several PIDs for correct simulation.
- Occasionally, CAD parts containing several subparts with their PIDs are aggregated to one single CAE part, consequently still containing several PIDs.

4.3 Assemblies

In many CAx systems, parts containing sub-parts are called assemblies. The notion distinguishes them from leaves of the part tree or graph.

However, in χ MCF, an assembly is just a set of parts and/or properties, denoted by their part labels and PIDs. They do not need to possess any special relation respective to the part graph. The opposite is true: χ MCF-assemblies address situations, where specifying a single PID would address not enough, a high level part would address way too many elements and medium-size parts would not make the job.

On the other hand, this does not happen too often: If a weld line e. g. crosses property boundaries, these properties usually belong to the same tailored blank, hence the same part. If there would be a physical gap between the properties, welding would be applied to a single sheet across this gap, which causes new questions to the welding process:

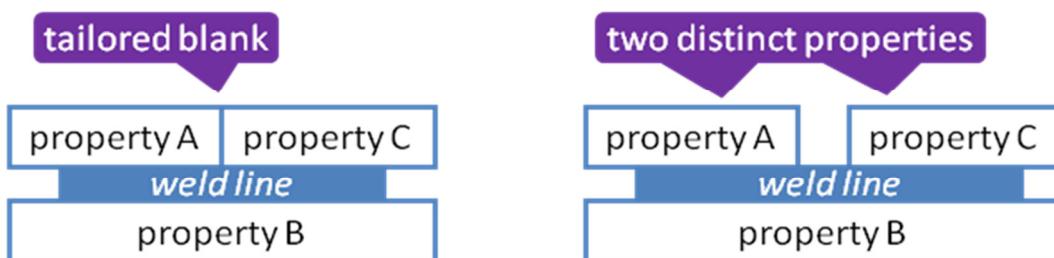


Figure 6: Weld line crossing tailored blank vs. weld line crossing physical gap

And even then: Due to geometrical proximity and usual assembly processes, it is very likely that properties A and C belong to the same part just one level above in part graph.

5 File Structure of χ MCF

As mentioned before, χ MCF is built upon XML. This eases χ MCF to possess a clear logical structure.

The root/document element of χ MCF is mandatorily named `<xmcf/>`³. The root element may contain the following types of child elements

- 1) Comments following the usual XML standard; hence not further discussed here.
- 2) Elements containing general information.
- 3) Variant declaration.
- 4) Groups of connection specific elements `<connection_group/>` of arbitrary number.
- 5) Element `<appdata>` containing data specific for individual applications
- 6) Element `<femdata>` containing finite element specific data.

5.1 Elements containing general information

χ MCF is equipped with the following elements for general information:

- `<date/>` optional
- `<version/>` mandatory
- `<units/>` optional

The root element `<xmcf/>` contains the following nested elements:

Nested Elements	Multiplicity	Status	Constraint / Remarks
date	0-1	Optional	-
version	1	Required	-
units	0-1	Optional	-
appdata	0-*	Optional	See section 5.2.1
femdata	0-*	Optional	See section 5.2.2
connection_group	0-*	Optional	See section 5.3

Table 1: Nested elements of element `<xmcf/>`

5.1.1 Date

The element `<date>` of the format “yyyy-mm-dd” specifies the date on which of the file is created. It follows norm ISO 8601, cf. http://en.wikipedia.org/wiki/ISO_8601.

³ Since V2.0 introduces significant changes, root element has been renamed from "mcf" to "xmcf" in order to avoid confusion with the "old" MCF-Format.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xmcf_3_0_0.xsd">
    <date> 2015-08-27 </date>
    <version> 3.0.0 </version>
    <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
    ...
</xmcf>
```

5.1.2 Version

The version code of the χ MCF standard upon which the current file is built must be specified by the element `<version>`.

The version code of χ MCF files following this document is 3.0.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=" xmcf_3_0_0.xsd">
    <date> 2015-08-27 </date>
    <version> 3.0.0 </version>
    <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
    ...
</xmcf>
```

5.1.3 Unit System

The unit system used by χ MCF is based upon the International System of Units (SI⁴) and specified by the element `<units>`. Both the base and the derived units are supported, including decimal prefixes.

Following non-SI units are allowed, additionally: Length [in] and [ft]; Mass [lb].

There is no need to declare units for dimensionless physical quantities, e. g. friction coefficients.

XML-specification of `<units>`:

Attribute	Use	Value Space	Default
length	Optional	"mm", "m", "in", "ft"	"mm"
angle	Optional	"deg", "rad"	"deg"
mass	Optional	"g", "kg", "t", "lb"	"kg"
force	Optional	"kN", "N"	"N"
time	Optional	"s", "min", "h"	"s"
torque	Optional	"Nm"	"Nm"

Table 2: XML-specification of `<units>`

⁴ Cf. <http://en.wikipedia.org/wiki/SI>.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xmcf_3_0_0.xsd">
    <date> 2015-08-27 </date>
    <version> 3.0.0 </version>
    <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
    ...
</xmcf>
```

5.2 Application, User and Process Specific Data

The user/application software can store additional information into a χ MCF file. In this way, flexibility is introduced which enables an easy integration of χ MCF into an existing development process.

The current χ MCF definition allows two such data elements:

- **<appdata>**
Contents has to be documented by the corresponding application or user. It is *no* official part of the χ MCF standard.
- **<femdata/>**
Contents is documented in FATXML [7] and hence does not need to be described, here.

5.2.1 User Specific Data <appdata>

<appdata/> is suitable for any user/application specific information and can be placed on root level (directly within <xMCF/> tag) and within any single connector (tags <connection_0d/>, <connection_1d/>, and <connection_2d/>). Additionally it is also allowed to define directly under element <connection_group/>.

<appdata/> must contain at least one nested element named after the application or user that is intended to interpret the data.

Content of <appdata/> is regarded to be “private property” of the corresponding application. However, in the sense of “best practices”, it is recommended, but not required,

- to place application specific tags into a separate namespace,
- to provide an XML schema for its content,
- to import, store and export <appdata> of 3rd party applications on block – to prevent loss of data in applications allowing export,
- for a FE pre-/post processor to offer means for editing 3rd party data “formally”, i.e. according to the corresponding XML schema, but regardless of the physical meaning of that data.

The user must be aware that different systems are likely to introduce the same physical parameter at the same time (inducted e. g. by a certain new emerging connecting method), but describe them in their own XML schemata with different element/attribute names.

A preprocessor does not have any chance to detect these equivalent parameters, then. Hence, it cannot prevent contradictions between different <appdata> blocks of the same χ MCF file.

As of November 2015, the following applications (in alphabetical order) have been registered:

- ANSA
- FEMFAT

- HyperMesh
- LMS Virtual.Lab
- MEDINA
- nCode
- SyncroFIT

The above given list does *not* imply that other application names are forbidden. Its only purpose is to reserve the registered names against inappropriate use.

XML-specification of `<appdata>`:

Nested Elements	Multiplicity	Status	Constraint
ANSA	1	Optional	-
FEMFAT	1	Optional	-
HyperMesh	1	Optional	-
LMSVirtualLab	1	Optional	-
MEDINA	1	Optional	-
NCODE	1	Optional	-
SyncroFIT	1	Optional	-

Table 3: XML-specification of `<appdata>`

Example A (`<appdata>` for MEDINA at root level):

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:MEDINA="http://servicenet.t-systems.com/medina/xMCF"
xsi:schemaLocation="http://servicenet.t-systems.com/medina/xMCF mcf_MEDINA.xsd"
xsi:noNamespaceSchemaLocation="xmcf.xsd">
    <date> 2014-08-07 </date>
    <version> 3.0.0 </version>
    <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
    <appdata>
        <MEDINA5 xmlns="http://servicenet.t-systems.com/medina/xMCF">
            <data_at_root>
                <version MEDINA="MEDINA 8.4.2 Maintenance Release (64 Bit)"/>
            ...
            </data_at_root>
        </MEDINA>
    </appdata>
    ...
</xmcf>
```

Example B (`appdata` for MEDINA at connection level):

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:MEDINA="http://servicenet.t-systems.com/medina/xMCF"
xsi:schemaLocation="http://servicenet.t-systems.com/medina/xMCF mcf_MEDINA.xsd"
xsi:noNamespaceSchemaLocation="xmcf.xsd">
    <date> 2014-08-07 </date>
    <version> 3.0.0 </version>
    <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
    ...
    <connection_group id="1">
        <connected_to>
```

⁵ MEDINA support for v3.0 is unforeseen.

```

    ...
</connected_to>
<connection_list>
    <connection_1d label="1000032">
        <loc_list>
            ...
        </loc_list>
        <seamweld>
            ...
        </seamweld>
        <appdata>
            <MEDINA xmlns="http://servicenet.t-systems.com/medina/xMCF">
                <data_at_connector>
                    ...
                </data_at_connector>
            </MEDINA>
        </appdata>
    </connection_1d>
</connection_list>
</connection_group>
</xmcf>

```

5.2.2 Finite Element Specific Data <femdata>

For the numerical simulation by finite element method, a joint can be discretized (realized) in different kinds and ways depending on the focus of the simulation (crash, fatigue etc.). It is thus frequently necessary to switch from one realization to another one. For this purpose, details of a specific realization may be of interest.

The optional `<femdata>` can be placed within any single connector⁶ (relevant tags are `<connection_0d/>`, `<connection_1d/>` and `<connection_2d/>`).

`<femdata>` references FEM-entities that are related to the connector in which it is placed. Its content, i.e. nested elements are specific to a single solver.

This solver naming should be taken from FATXML version 1.1.0 (as current version) which are the followings:

- PAM-CRASH
- LS-DYNA
- Permas
- ABAQUS
- RADIOSS
- Optistruct

And these should be extended by other also required solver names to enable wide usage of the standard:

- NASTRAN
- FEMFAT

XML-specification of `<femdata>`:

Nested Elements	Multiplicity	Status	Constraint
PAMCRASH	1	Optional	-
LSDYNA	1	Optional	-

⁶ Future xMCF versions may include `<femdata/>` at root level or `<connection_group/>` elements, but this is not allowed in V 3.0

PERMAS	1	Optional	-
ABAQUS	1	Optional	-
RADIOSS	1	Optional	-
OPTISTRUCT	1	Optional	-
NASTRAN	1	Optional	-
FEMFAT	1	Optional	-

Table 4: XML-specification of element <femdata>

Only the `<CAE_DATA>` tag, defined and documented in FATXML [7], is allowed as nested element of the child element of `<femdata>`.

<i>Nested Elements</i>	<i>Multiplicity</i>	<i>Status</i>	<i>Constraint / Remarks</i>
CAE_DATA	1-*	Required	As defined in [7].

Table 5: Nested elements of element <femdata>

The `CAE_DATA` element has only one attribute as defined in FATXML document version v1.1.0. This is the following:

<i>Attributes</i>	<i>Type</i>	<i>Status</i>	<i>Constraint</i>
VERSION	Numeric	Required	-

Table 6: Attributes elements of element <femdata>

The `CAE_DATA` element can have nested elements as defined in FATXML document version v1.1.0. These are the followings:

<i>Nested Elements</i>	<i>Multiplicity</i>	<i>Status</i>	<i>Constraint</i>
REPRESENTATION	1	Optional	-
COMMENT	1	Optional	-
CAE_PART_MEMBER	1-*	Required	-

Table 7: Nested elements of element <CAE_DATA>

The FATXML elements and attributes have the following semantics:

- ***VERSION***: It represents the CAE Variant in the xml code. The 0 value stands for original data from PDM System. If the CAE Part is being modified in a preprocessor then this attribute will be incremented.
- ***REPRESENTATION***: This element describes by text content within its nested element value the same PDM Part which is being represented by a specific FE modeling technique. E.g. for CRASH or NVH different mesh quality or used FE Element can be assumed.
- ***COMMENT***: This element can be used to make some comments regarding to the applied CAE data.
- ***CAE_PART_MEMBER***: To describe the CAE part members involved in the connection this element can be used to identify it by defining an `ID` and `ENTITY` elements. For further definition of `CAE_PART_MEMBER` see the document source website for FATXML [7].

Example A (example of `<femdata>` within a `<connection_0d/>` element):

```

<connection_0d label="My0dConnection_id_100000">
  ...
  <femdata>
    <NASTRAN>
      <CAE_DATA VERSION="1">
        <REPRESENTATION>
          ...
        </REPRESENTATION>
        <COMMENT>
          ...
        </COMMENT>
        <CAE_PART_MEMBER ID="1">
          <ENTITY>
            <TYPE>
              CQUAD
            </TYPE>
            <ID>
              12345-12356
            </ID>
          </ENTITY>
        </CAE_PART_MEMBER>
      </CAE_DATA>
    </NASTRAN>
  </femdata>
  ...
</connection_0d>

```

Similar to FATXML, χ MCF data can be embedded into solver decks by this means: Any receiving system can easily detect and remove discretization objects, created by a sending system, in order to substitute them by its own new discretization objects.

5.3 Connection Data `<connection_group/>`

`<connection_group/>` comprises the topological information about the involved parts and assemblies (Chapter 4), respectively. As explained in Section 2.4, joints are grouped together by the parts or assemblies which they commonly connect.

The topological relation (relation of neighbors) is defined by the child element `<connected_to>` whereas all involved joints are listed in the child element `<connection_list>` according to their types (see Section 2.2).

Each `<connection_group/>` is uniquely identified by a numeric identifier (`id`).

Note: From this, it follows that χ MCF files *cannot* be simply “pasted together” by use of a standard text editor.

XML-specification of `<connection_group/>`:

Attributes	Type	Status	Constraint
<code>id</code>	Numeric (integer)	Required	unique within a χ MCF file

Table 8: Attributes of element `<connection_group/>`

Nested Elements	Multiplicity	Status	Constraint
<code>connected_to</code>	1	Required	-
<code>connection_list</code>	1	Required	-
<code>contact_list</code>	0-1	Optional	-

Table 9: Nested elements of element `<connection_group/>`

Remarks:

- An empty or missing `<connected_to>` (meaning a connection according only to geometric neighborhood) is *not* allowed by the standard. However, systems may import such files with warnings.
- In addition to parts and properties, *no* other means (e.g. sets) for grouping objects are allowed.

5.3.1 Connected Objects

The basic objects which can be jointed together are parts and assemblies (see Chapter 4) which appear as nested elements `<part/>` and `<assy/>` of `<connected_to>`.

XML-specification of `<connected_to>`:

Nested Elements	Multiplicity	Status	Constraint
part	1 - *	At least <i>one</i> of these elements is required	-
assy	1 - *	-	-

Table 10: Nested elements of `<connected_to>`

5.3.1.1 Element `<part/>`

In xMCF, a part may refer to one CAx part or one CAE property, as well. However, if both attributes “label” and “pid” are present, the label governs.

It is described by the element `<part/>` and a numeric `index`, a `label` or a `pid` (property id), all provided as attributes.

The `index` needs to be unique only within the parent element `<connected_to>`. For specific connections, it is used as the matching index for the base sheet.

The attribute `index` of `<part/>` element is required only if the `part` element is being used as nested element under element `<connected_to>`. In case if the `<part/>` element is used within the element `<assy/>` then NO `index` is allowed to be present as attribute of the `<part/>` element.

XML-specification of `<part/>`:

Attributes	Type	Value Space	Status	Constraint
index	Integer	> 0	Required	Unique and required only within the parent element <code>connected_to</code>
label	Alphanumeric	Alphanumeric	Optional	Optional, if pid is present.
pid	Integer	> 0	Optional	Optional, if label is present.

Table 11: Attributes of element `<part/>`

Example A (only required attributes):

```
<connected_to>
  <part index="1" label="PART_7000400"/>
</connected_to>
```

Example B (within optional attributes):

```
<connected_to>
  <part index="1" label="PART_7000400" pid="3202132"/>
</connected_to>
```

5.3.1.2 Element <assy />

An assembly represents a sub-structure consisting of at least two `<part/>` elements. It is described by the element `<assy/>` with only the mandatory attribute `index`.

XML specification of element `<assy/>`:

Attributes	Type	Status	Constraint
index	Integer	Required	Unique within the parent element

Table 12: Attributes of element <assy/>

Example A (only <assy/> element within <connected_to> - full definition):

```
<connected_to>
  <assy index="42">
    <part label="sheet_steele_in_door_left_30_thickness2.70" pid="110013"/>
    <part label="sheet_steele_in_door_left_31_thickness2.75" pid="110099"/>
  </assy>
</connected_to>
```

Example B (<part/> and <assy/> elements within <connected_to> - full definition):

```
<connected_to>
  <part index="1" label="sheet_steele_in_door_left_32_thickness3.2" pid="3202132"/>
  <assy index="42">
    <part label="sheet_steele_in_door_left_30_thickness2.70" pid="110013"/>
    <part label="sheet_steele_in_door_left_31_thickness2.75" pid="110099"/>
  </assy>
</connected_to>
```

Example C (<part/> and <assy/> elements within <connected_to> - minimum definition):

```
<connected_to>
  <part index="1"/>
  <assy index="42">
    <part label="sheet_steele_in_door_left_30_thickness2.70"/>
    <part label="sheet_steele_in_door_left_31_thickness2.75"/>
  </assy>
</connected_to>

OR

<connected_to>
  <part index="1"/>
  <assy index="42">
    <part pid="110013"/>
    <part pid="110099"/>
  </assy>
</connected_to>
```

The body of an `<assy/>` tag equals that of a `<connected_to>` tag. But the meaning is different: All parts within one `<assy/>` tags are meant to constitute *the same* side/layer/partner of a flange, whereas all members of a `<connected_to>` tag are *different* sides/layers/partners of a flange.

Recursion, i.e. an `<assy/>` tag nested within another `<assy/>` tag, is not allowed.

5.3.2 Contacts and Friction

For many joint types like bolts, screws etc., friction between the jointed partners plays an important role for the manufacturing and the mechanical behavior of the joints in service.

In general, friction is a property of pairs of materials in contact. Normally it can be assumed that the friction property, here simply characterized by the static and kinetic friction coefficients, is homogenous. Nevertheless, friction properties must allow for local modification of an individual connection in order to enhance the service behavior.

In xMCF, friction coefficients for any combination of joint partners defined in `<connected_to>` can be specified by the element `<contact_partners/>` which is nested in the element `<contact_list/>`. Each part in contact is given by the element `<partner/>`. The static and kinetic friction coefficients are defined by the element `<coefficients/>`.

The friction property between the head of a bolt to jointed parts is specified, where the joint is defined.

5.3.2.1 Element `<contact_list/>`

Relevant contacts, which are possible between the flange partners of a `<connection_group/>`, are collected in a `<contact_list/>`.

XML specification of `<contact_list/>` element:

Nested Elements	Multiplicity	Status	Constraint
contact	1 - *	Required	Any set (= non-ordered pair) of physical contact partners must not appear more than once within a <code><contact_list/></code> .

Table 13: Nested elements of element `<contact_list/>`

The element `<contact_list/>` does not allow for any attributes.

5.3.2.2 Element `<contact/>`

The features or coefficients of a physical contact between flange partners are described by an element `<contact>`.

XML specification of `<contact/>` element:

Nested Elements	Multiplicity	Status	Constraint
partner	2	Required	-
coefficients	1	Required	-

Table 14: Nested elements of element `<contact>`

Ordering of `<contact>` elements within a `<contact_list/>` is irrelevant, since it is assumed that features of a physical contact are invariant under permutation of the two involved materials.

The element `<contact>` does not allow for any attributes.

5.3.2.3 Element *<partner>*

Each joint partner involved in a contact is specified by the element *<partner>*. It is referred to by an attribute label or pid. Only those labels or pids are allowed, which are listed in *<connected_to>*.

XML specification of *<partner>* element:

Attributes	Type	Value Space	Status	Constraints / Remarks
label	Alphanumeric	Alphanumeric	Optional	Optional, if pid is present.
pid	Integer	> 0	Optional	Optional, if label is present.

Table 15: Attributes of element *<partner>*

These attributes have following semantics:

- **label:** One of the labels mentioned in *<connected_to>* of the current *<connection_group>* (see section 5.3.1.1).
- **pid:** One of the pids mentioned in *<connected_to>* of the current *<connection_group>* (see section 5.3.1.1).

The element *<partner>* does not allow for any nested elements.

5.3.2.4 Element *<coefficients>*

Static and kinetic friction coefficients are defined by the attributes *static_friction* and *kinetic_friction* of an element *<coefficients>*, respectively.

Example

```

<connected_to>
  <part index="1" label="sheet_steele_in_door_left_32_thickness3.2" pid="3202132"/>
  <assy index="42">
    <part label="sheet_steele_in_door_left_30_thickness2.70" pid="110013"/>
    <part label="sheet_steele_in_door_left_31_thickness2.75" pid="110099"/>
  </assy>
</connected_to>
<contact_list>
  <contact>
    <partner label="sheet_steele_in_door_left_32_thickness3.2"/>
    <partner label="sheet_steele_in_door_left_30_thickness2.70"/>
    <coefficients static_friction="0.3" kinetic_friction=".25"/>
  </contact>
  <contact>
    <partner label="sheet_steele_in_door_left_31_thickness2.75"/>
    <partner pid="3202132"/>
    <coefficients static_friction="0.52" kinetic_friction=".25"/>
  </contact>
</contact_list>

```

The element *<coefficients>* does not allow for any nested elements.

5.3.2.5 Local Contact Properties

If necessary, local contact properties can be given within any element *<connection_0d>*, *<connection_1d>* or *<connection_2d>*, respectively (see section 5.3.3 Joints). In case of conflict, a local *<contact_list>* overrules the global one.

XML-specification of `<contact>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
static_friction	Floating point	0,+∞[Optional	-
kinetic_friction	Floating point	0,+∞[Optional	-

Table 16: Attributes of element `<contact>`

A `<connection_list>` must not be empty. That means, at least 1 connection has to be defined.

5.3.3 Joints

All the joints which connect the same set of objects (order does not matter) described in the element `<connected_to>` are listed in the element `<connection_list>`. There should be *only one* connection group for any distinct set of objects in a xMCF file.

As discussed in Sect. 2.2, xMCF differs between 0-, 1- and 2-dimensional joints which will be specified in detail in the following chapters. Thus, an element `<connection_list>` can comprise child elements `<connection_0d>`, `<connection_1d>` and `<connection_2d>` of arbitrary repetitions.

XML-specification of `<connection_list>`:

Nested Elements	Multiplicity	Status	Constraint
connection_0d	*	optional	-
connection_1d	*	optional	-
connection_2d	*	optional	-

Table 17: Nested elements of element `<connection_list>`

A `<connection_list>` must not be empty. That means, at least 1 connection has to be defined.

5.4 A Minimalistic Example of a xMCF file

In the following, an example shows how the xMCF xml file should look like:

```

<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:MEDINA="http://servicenet.t-systems.com/medina/xMCF"
      xsi:schemaLocation="http://servicenet.t-systems.com/medina/xMCF mcf_MEDINA.xsd"
      xsi:noNamespaceSchemaLocation="xmcf.xsd">
<!-- File Name: new_car.xml -->
<xmcf>
    <!-- some comments -->
    <date> 2016-01-11 </date>
    <version> 3.0.0 </version>
    <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
    <appdata> <!--appdata at root level -->
        <MEDINA xmlns="http://servicenet.t-systems.com/medina/xMCF">
            <data_at_root>
                <version MEDINA="MEDINA 8.4.2 Maintenance Release (64 Bit)"/>
                ...
            </data_at_root>
        </MEDINA>
    </appdata>
    ...
    <connection_group id="1">
        <connected_to>
            <part index="1" label="sheet_leftdoor_front_t=3" pid="20123213"/>
            <part index="2" label="assembly_leftdoor_hinge" pid="90123213"/>
        </connected_to>
        <appdata> <!--appdata at connection_group level -->
            <MEDINA xmlns="http://servicenet.t-systems.com/medina/xMCF">
                <data_at_connection_group>
                    ...
                </data_at_connection_group>
            </MEDINA>
        </appdata>
        <connection_list>
            <connection_0d>
                <femdata>
                    <NASTRAN>
                        <CAE_DATA xmlns="FATXML"/>
                        ...
                    </CAE_DATA>
                </NASTRAN>
            </femdata>
            ...
            </connection_0d>
            <connection_1d>
                <loc_list>
                    ...
                </loc_list>
                <seamweld>
                    ...
                </seamweld>
            <appdata>
                <MEDINA xmlns="http://servicenet.t-systems.com/medina/xMCF">
                    <data_at_connector>
                        ...
                    </data_at_connector>
                </MEDINA>
            </appdata>
            ...
            </connection_1d>
            <connection_2d>
                ...
            </connection_2d>
            ...
        </connection_list >
    </connection_group>
    ...
</xmcf>
```

⁷ `xmlns="FATXML"` is just an example. Usually, an XML namespace is determined by providing an URL.

5.5 XML Schema Definition

XML-Schema definition (XSD) will be published at a later time on VDA web server.

6 Data Common to any Connection

6.1 Indices and their properties

χ MCF provides several elements which are essentially ordered sets of the same data type (strings, integers or decimals). More precisely they are like lists or vectors. For example, the `<loc_list/>` for the coordinate list of a seam weld or the `<string_list/>` in the `<custom_attributes/>`. Often the order of the elements in a set is essential. E.g. the coordinates in the `<loc_list/>` for a seam weld define the weld in the space uniquely by their values and their explicit order in the list.

The current XML standard allows that several child elements with an identical name can share the common parent. However, it lacks a built-in mechanism to introduce a logical structure (like an order) in an xml-document. χ MCF resolves this problem by introducing an index (attribute) in such cases. Indices may play a twofold role: to distinguish from each other and to ensure a unique arrangement in the list. Usually an index may be strictly monotone increasing natural numbers. In some cases strictly monotone increasing real numbers may also be rationally. For example, the “loc_list” for the coordinate list of a seam weld can be indicated both by real numbers like arc length of the line or any increasing integer series.

Depending on the context the name of an index (attribute) may `index`, `v` or something else. They will always explicitly be declared at the appropriate places in the text.

6.2 Attribute `label`

Any connection should have an attribute called `label`, which identifies it throughout the entire CAE process. It is not necessary that these labels are unique: For instance, if a weld line is split into different parts at a certain step in the process (e.g.: when crossing holes in the structure), its components shall keep the `label` attribute. A system “way down” in the process (i.e. detached from any centralized naming authority) may create new connections with all the same label, e.g.: “0” or empty string.

The `label` may be composed of digits only, but it should not be confused with e.g.: a finite element’s ID. If desired, finite element IDs would have to be placed within some `<appdata>` element.

6.3 Dimensions and Coordinates

Connections may come in three different dimensions: `<connection_0d/>`, `<connection_1d/>` and `<connection_2d/>`.

Any connection must have *coordinates*. How many they are and how they are described depends on the connection’s dimension. Details are described in the following sections.

6.4 Attribute `quality_control`

Some connections are more relevant than others, e. g. with respect to crash safety. Hence, several levels of quality control are well established in manufacturing processes. For this reason, any connection can have an optional attribute `quality_control`. Since there is no general standard for such quality controls, χ MCF cannot define a set of possible values for this attribute. Hence, it must be of type **Alphanumeric**.

6.5 Custom Attributes list

It was mentioned in Chapter 2.1 that only information relevant to connections should be contained in χ MCF. Exceptions `<appdata>` and `<femdata/>` were introduced in Chapter 5.2. `<appdata>` and `<femdata/>` aim mainly at specific needs of application software. The internal structure of `<appdata>` itself is not standardized, may be very complex and depends on the specific software. The content can usually not be interpreted by other software systems.

There are frequently situations where a user of χ MCF wishes to introduce supplementary information (attributes) to enrich the standard attributes defined by χ MCF. In principle, the supplementary information could also be placed in an `<appdata>`-block, but with a substantial drawback, namely, its exchange between different commercial software tools will be difficult in case the tool specific internal structure is not documented.

With `<custom_attributes/>`, χ MCF provides an element which is simple in handling and flexible enough to meet many requirements. All descendants of `<custom_attributes/>` are key-value-pairs, following the same pattern `key ↔ value(s)`, with supported `value-type` $\in [int, real, string]^N$, where N is a positive integer:

$$(\text{value-type}) \text{ key} = \{\text{value1}, \text{value2}, \dots, \text{valueN}\}.$$

The case $N > 1$ is reminiscent of the `vector` or `list` from the STL of C++ and is called *list* in χ MCF.

In detail, the individual elements of `<custom_attributes/>` are of one of the following forms:

```

<int key="NameofIntValue"> value </int>

<int_list key="NameofIntListValue">
    <value index="1"> value1 </value>
    ...
    <value index="N"> valueN </value>
</int_list>

<real key="NameofRealValue"> value </real>
<real_list key="NameofRealListValue">
    <value index="1"> value1 </value>
    ...
    <value index="N"> valueN </value>
</real_list>

<string key="NameofStringValue"> value </string>
<string_list key="NameofStringListValue">
    <value index="1"> value1 </value>
    ...
    <value index="N"> valueN </value>
</string_list>.

```

I.e. the name of the elements specifies the `value-type` while the `value(s)` is/are hold in one or several element(s) `<value/>`. A list is signified by the suffix `_list`. All elements own the attribute `key`.

Often the `<custom_attributes/>` has an `owner` and is needed `for` a special purpose. For example, Mr. Brown needs for one and the same joint element an integer valued attribute named `priority`

which should assume different values for two applications "*Fatigue*" (1) and "*Statics*" (22). These could be specified in a `<custom_attributes_list/>` as follows:

Example

```
<custom_attributes_list>
    <custom_attributes owner="Mr Brown" for="Fatigue">
        <int key="priority"> 1 </int>
    </custom_attributes>
    <custom_attributes owner="Mr Brown" for="Statics">
        <int key="priority"> 22 </int>
    </custom_attributes>
</custom_attributes_list>
```

In the above example, the `owner` is in both cases "*Mr Brown*" while the applications can be distinguished by the attributes `for="Fatigue"` and `for="Statics"`, respectively.

The more general case that several `<custom_attributes/>` with different ownerships and for different purposes is considered by the element `<custom_attributes_list/>` with all the `<custom_attributes/>`'s as child-elements. No attributes are associated to the element `<custom_attributes_list/>`.

Existence of `<custom_attributes_list/>` inside a connection is optional. There can be up to one element inside each connection.

The `<custom_attributes_list/>` contains at least one of the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
custom_attributes	1 - *	Required	-

Table 18: Nested elements of element `<custom_attributes_list/>`

Existence of `<custom_attributes/>` inside `<custom_attributes_list/>` is required. There must be at least one element inside `<custom_attributes_list/>`.

XML specification of `<custom_attributes/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
owner	Alphanumeric	Alphanumeric	Required	Non-empty string
for	Alphanumeric	Alphanumeric	Optional	Non-empty string

Table 19: Attributes of `<custom_attributes/>` element

The attributes `owner` and `for` together of each `<custom_attributes/>` element must be unique within each `<custom_attributes_list/>`.

The `<custom_attributes/>` element may contain at least one of the following nested elements:

Nested Elements	Multiplicity	Status	Constraints / Remarks
string	0 - *	optional	At least one of these nested elements.
real	0 - *	optional	
integer	0 - *	optional	
string_list	0 - *	optional	
real_list	0 - *	optional	

int_list	0 - *	optional	
----------	-------	----------	--

Table 20: Nested elements of element `<custom_attributes/>`

The elements `<string/>`, `<real/>` and `<integer/>` are allowed to have the following data type assignments for their value:

- **string element:** alphanumeric that is covered by string data type in xsd, which can contain characters, line feeds, carriage returns, and tab characters.
- Remark:* If required to handle not needed items e.g. line feeds or tab characters another data type should be used in XML Schema Definition that is the so called normalized string data type. The normalized string data type is derived from the String data type. The normalized string data type also contains characters, but the XML processor will remove line feeds, carriage returns, and tab characters.)
- **real element:** floating point that is covered by decimal data type in xsd, which can contain a numeric value. The maximum number of decimal digits you can specify is 18.
 - **integer element:** integer that is covered by integer data type in xsd, which can contain a numeric value without a fractional component.

XML specification of `<string/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
key	Alphanumeric	Alphanumeric	Required	Non-empty string

Table 21: Attributes of `<string/>` element

XML specification of `<real/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
key	Alphanumeric	Alphanumeric	Required	Non-empty string

Table 22: Attributes of `<real/>` element

XML specification of `<integer/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
key	Alphanumeric	Alphanumeric	Required	Non-empty string

Table 23: Attributes of `<integer/>` element

XML specification of `<string_list/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
key	Alphanumeric	Alphanumeric	Required	Non-empty string

Table 24: Attributes of `<string_list/>` element

`<string_list/>` has the nested element:

Nested Elements	Type	Multiplicity	Status	Constraints / Remarks
value	Alphanumeric	1 - *	required	-

Where `<value/>` within `<string_list/>` is specified as:

Attributes	Type	Value Space	Status	Constraints / Remarks
index	Integer	>0	Required	unique within the parent element

Table 25: Attributes of `<value/>` element inside `<string_list/>`

XML specification of `<real_list/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
key	Alphanumeric	Alphanumeric	Required	Non-empty string

Table 26: Attributes of `<real_list/>` element

`<real_list/>` has the nested element:

Nested Elements	Type	Multiplicity	Status	Constraints / Remarks
value	Floating point	1 - *	Required	-

Where `<value/>` within `<real_list/>` is specified as:

Attributes	Type	Value Space	Status	Constraints / Remarks
index	integer	>0	Required	unique within the parent element

Table 27: Attributes of `<value/>` element inside `<real_list/>`

XML specification of `<int_list/>`:

Attributes	Type	Value Space	Status	Constraints / Remarks
key	Alphanumeric	Alphanumeric	Required	Non-empty string

Table 28: Attributes of `<int_list/>` element

`<int_list/>` has the nested element:

Nested Elements	Type	Multiplicity	Status	Constraints / Remarks
value	Integer	1 - *	Required	-

Where `<value/>` within `<int_list/>` is specified as:

Attributes	Type	Value Space	Status	Constraints / Remarks
index	integer	>0	Required	unique within the parent element

Table 29: Attributes of `<value/>` element inside `<real_list/>`

Remarks:

1. Values of `key`'s must be unique within their common parent element.
2. The order of the values in the corresponding list is identified by the numerical value of their attribute `index`, in ascending order, see 6.1 Indices and their properties.
3. In case of strings the whitespaces deserve extra mention: To avoid mistakes, whitespaces are *not* to be used at beginning and end of a string.

Example:

```
<custom_attributes_list>
    <custom_attributes owner="DepartmentA" for="Fatigue">
        <int key="priority"> 1 </int>
        <string key="used S-N curve">Steel_225_ISO</string>
        <real key="fatigue_limit"> 223.1 </real>
    </custom_attributes>
    <custom_attributes owner="DepartmentA" for="Statics">
        <int key="priority"> 2 </int>
    </custom_attributes>
    <custom_attributes owner="DepartmentB">
        <string key="priority">high</string>
        <real_list key="direction vector">
            <value index="1">10.3 </value>
            <value index="2"> -2.1</value>
            <value index="3">-1.5</value>
        </real_list>
        <string_list key="verifiedby" >
            <value index="1">john</value>
            <value index="2">Smith</value>
        </string_list>
    </custom_attributes>
</custom_attributes_list>
```

6.6 Distinction between `<custom_attributes/>` and `<appdata/>`

Besides the differences mentioned in Chapt. 6.5 there are further differences between `<appdata/>` and `<custom_attributes/>` in use and handling.

`<appdata/>` may be used on different levels of a χ MCF file:

- It may appear on root level (directly within `<xmcf/>` tag)
- and within any single connector (tags `<connection_0d/>`, `<connection_1d/>` and `<connection_2d/>`)

In contrast to this, `<custom_attributes_list/>` can only be used within any single connector, not at root level. There are good reasons for this:

Consider the common scenario, where many χ MCF files each containing connections of subsystems are to be read in an application. The application will have to deal with conflicts between root level data and conflicts with data at connector level.

- At root level (within `<xmcf/>` element), any application should be able to deal with conflicts of its own `<appdata/>`, because their nature is known by the application. On the other hand, the purpose of a possible `<custom_attributes/>` element is not known by the application. The application would therefore have to pass the task of resolving `<custom_attributes/>` conflicts to the engineer. This is undesirable.
- At the connector level (within `<connection_xd/>` elements), any application should be able to handle conflicts between connectors. This is because connectors are domain objects: both, the application and the engineer, are aware of the connectors' role and existence. So, both the application and the engineer can resolve connector conflicts if needed. After such a conflict has been resolved, there is no conflict of `<appdata/>` or `<custom_attributes/>` left to be solved by the engineer or the application, because these data have a limited scope; they live within the confines of the connector. This is very convenient.

7 0D connections

7.1 Generic Definitions

7.1.1 Identification

Each point connection is optionally identified by its `label`. This identification can be made at the element called `<connection_0d/>`.

The XML definitions of all 0D connections i.e. `<connection_0d/>` elements are containing the following attributes:

Attributes	Type	Status	Constraint
label	Alphanumeric	Optional	-
quality_control	Alphanumeric	Optional	See section 6.4 Attribute <code>quality_control</code>

Table 30: Attributes of element `<connection_0d/>`

Attribute “label”

The label defines the human readable identification of connection. It might contain a description of the connection or simply an index as an integer.

Example A (minimum definition):

```
<connection_list>
  <connection_0d>
    <loc>
      ...
    </loc>
    <spotweld>
      ...
    </spotweld>
  </connection_0d>
</connection_list>
```

Example B (within assigned text to `label`):

```
<connection_list>
  <connection_0d label="MySpotWeld">
    <loc>
      ...
    </loc>
    <spotweld>
      ...
    </spotweld>
  </connection_0d>
</connection_list>
```

7.1.2 Location

The definition of the connection location is described by the element `<loc>`. This element is nested below the parent element `<connection_0d/>`. It contains three values specifying the x, y and z coordinates of the location as text content.

<i>Text</i>	<i>Type</i>	<i>Value Space</i>	<i>Status</i>	<i>Constraint</i>
x	Floating point	($-\infty$, $+\infty$)	Required	-
y	Floating point	($-\infty$, $+\infty$)	Required	-
z	Floating point	($-\infty$, $+\infty$)	Required	-

Table 31: Text values of element <loc>

Example (with minimum definition for <connection_0d>):

```
<connection_0d>
  <loc> 2581.21 -708.408 31.6532 </loc>
  ...
</connection_0d>
```

7.1.3 Direction

The definition of connection directions, where applicable, is described by the elements <normal_direction/> and <tangential_direction/>. They both specify a direction vector. Lengths of both vectors are not relevant, as long as they are > 0.0 .

Their XML definition is the same. If some definition requires both vectors, then their names signify their usage:

Element <normal_direction/> denotes a direction of a local z axis.

Element <tangential_direction/> denotes the direction of an axis tangential to (base) part surface next to the point given in <loc>, giving locale x axis. Its orthogonalization⁹ relative to <normal_direction> must not vanish, i. e. both vectors must not be collinear.

If both elements are given, a right handed coordinate system is uniquely defined:

- Origin is in <loc>.
- z-axis is in direction of <normal_direction/>.
- x-axis is the orthogonalization of <tangential_direction/> relative to <normal_direction/>.
- y-axis is given by cross product¹⁰ z-axis \times x-axis.

XML specification of <normal_direction/> and <tangential_direction/> elements:

<i>Attributes</i>	<i>Type</i>	<i>Value Space</i>	<i>Status</i>	<i>Constraint</i>
X	Floating point	($-\infty$, $+\infty$)	Required	-
Y	Floating point	($-\infty$, $+\infty$)	Required	-
Z	Floating point	($-\infty$, $+\infty$)	Required	-

Table 32: Attributes of elements <normal_direction/> & <tangential_direction/>

Both elements do *not* allow for any nested elements.

Unless otherwise stated, direction elements are optional. However, if omitted, importing systems may use a geometric search for determining <normal_direction/>, using a spherical characteristic, which may not be very reliable. <tangential_direction/> can then only be guessed, implying a

⁹ See http://en.wikipedia.org/wiki/Gram%20Schmidt_process.

¹⁰ See http://en.wikipedia.org/wiki/Cross_product.

random orientation of the connection (e.g. a Robscan) in receiving system. Hence, it is recommended for a receiving system to issue a warning, at least.

Examples:

```
<normal_direction x="0.0" y="0.0" z="-1.0" />
<tangential_direction x="70.7" y="70.7" z="0.0" />
```

7.1.4 Type Specification

Each connection is identified by its type. The XML definitions of all OD connections are containing the following elements:

Nested Elements	Multiplicity	Status	Constraint
bolt	1	Optional	-
gumdrop	1	Optional	-
rivet	1	Optional	-
robscan	1	Optional	-
screw	1	Optional	-
spotweld	1	Optional	-
contact_list	1	Optional	See section 5.3.2.5.

Table 33: Nested elements of element `<connection_0d/>`

Note: Exactly *one* of the type elements (*bolt*, *gumdrop*, *rivet*, *robscan*, *screw*, or *spotweld*) must exist in `<connection_0d/>`. There is *no* default type.

7.2 Spot Welds

A spot weld is denoted by an element `<spotweld/>`. This element is described completely by its attribute and nested elements.

Nested Elements	Multiplicity	Status	Constraint
spotweld	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 34: Nested element of `<connection_0d/>` for `<spotweld/>`

XML specification of `<spotweld/>` with element `diameter`:

Attributes	Type	Value Space	Status	Constraint
diameter	Floating point	>0.0	Optional	-
technology	Selection	resistance, laser, projection, friction	Optional	-

Table 35: Attributes of element `<spotweld/>`

Attribute “diameter”

The diameter of a spotweld is specified by the attribute `diameter` for the child element of `<connection_0d/>`.

Attribute “technology”

The technology used to weld the connection can be specified for each of the spot welds of a connection separately.

This technology can be one of

- Resistance welding
- Laser welding
- Projection welding
- Friction welding

Example:

```
<connection_0d label="SW_left_Gh_2123921">
  <spotweld diameter="5.0"/>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

7.3 Robscans

A Robscan is a pattern of arbitrary shape, drawn onto the flange partners by a laser optic. Such a shape has a length and width significant larger than the diameter of the laser focus. The laser beam defines a local z-axis and is assumed to be perpendicular to the flange partners. However, the pattern can be rotated around this z-axis, and it can be mirrored at its x-axis. This is depicted in following figure:

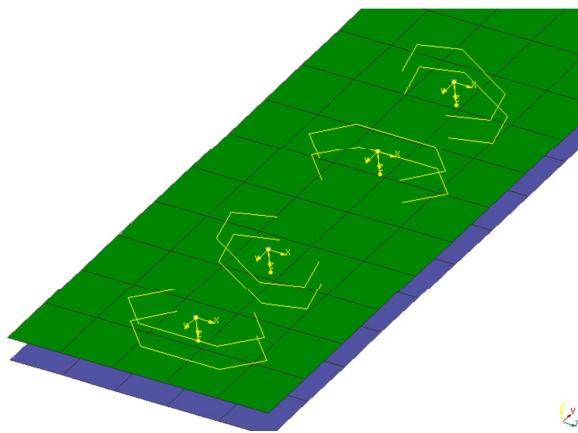


Figure 7: Robscans with Different Rotation Angles; Two of them Mirrored

The pattern of the bottom left Robscan is oriented with no rotation and no mirroring with respect to its own coordinate system (yellow). The next instance has 30° rotation. The two Robscans, top right in the figure, have a mirrored pattern; the uppermost having again 30° rotation.

There is a continuum of patterns for Robscans. Each one which shall be used at assembly line needs to be verified (by simulation plus test) in advance. This is expensive. Some implications are:

- Companies regard this information to be their own intellectual property.
- A pattern must not simply be stretched etc. It would need a new validation.
- Validated Robscan patterns are usually not part of distributions of FE processors.
- However, subcontractors shall have access to the position and "bounding box" of the Robscan, e. g.
- Hence, χ MCF definition shall contain some "abstract" data.
- FE processors may address the danger of inconsistency by taking both levels of information from the same configuration file. So it is at the responsibility of the companies' admins to have consistent data in that file.

Since the exact shape of the Robscan pattern is 3rd party intellectual property, it cannot be part of χ MCF definition. It is referred to by just a string attribute "pattern". Possible values of attribute "pattern" are *not* subject of standard: In general, they are very OEM specific. However, to provide a minimum amount of information, width and length of the pattern are given by attributes "pattern_width" and "pattern_length".

A Robscan is denoted by an element `<robscan/>`. This element is described completely by its attribute and nested elements.

Nested Elements	Multiplicity	Status	Constraint / Remarks
robscan	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 36: Nested element of `<connection_0d/>` for `<robscan/>`

XML specification of `<robscan/>` element:

Attributes	Type	Value Space	Status	Constraint / Remarks
base	Integer	> 0	Optional	-
pattern	Alphanumeric	Alphanumeric	Optional	Non-empty, if present.
gap	Floating point	≥ 0.0	Optional	-
width	Floating point	> 0.0	Optional	-
pattern_width	Floating point	> 0.0	Optional	-
pattern_length	Floating point	> 0.0	Optional	-
mirrored	Boolean	"false" (default), "true"	Optional	-
orientation_angle	Floating point	$[-180^\circ, 180^\circ]$	Optional	According to the unit of angles, defined in element <code><units/></code> .
filler_material	Alphanumeric	Alphanumeric	Optional	-

Table 37: Attributes of element `<robscan/>`

All attributes of element `<robscan/>` are optional for import to CAD or CAE processors. However, specific FE solvers may declare some of them to be mandatory.

General defaults are: false for boolean values, 0 for numeric values, "" for strings. However, these defaults are not always useful for CAE.

- **gap**: is the gap between both flange partners, probably defined by so-called "spacers".
- **width**: is the width of the laser beam.
- Width and length of the pattern are given by attributes **pattern_width** and **pattern_length**.
- **mirrored**: denotes, whether pattern has to be mirrored along its length-axis x, i. e. local y coordinate has to be inverted.
- **orientation_angle**: allows a rotation around z axis, following right-hands-rule. Angle is measured in the unit of angles, defined in element **<units/>**, within range [-180°, 180°]. -180° and +180° degree are regarded to be identical. (Ranges can be defined in equivalent range in other unit.)

Both parameters, **mirrored** and **orientation_angle** address optimization simulations: They allow to vary their parameters more easily, if it is just an angle and a boolean, compared with calculating completely new orientation vectors.

Nested Elements	Multiplicity	Status	Constraint
normal_direction	1	Optional	-
tangential_direction	1	Optional	-

Table 38: Nested elements of element <robscan/>

- Element **<normal_direction/>** denotes direction of laser beam, giving locale z axis.
- Element **<tangential_direction/>** denotes direction of length-axis, giving locale x axis.

<normal_direction> and **<tangential_direction>** elements are described in section 7.1.3.

Example:

```

<connection_0d label="1272360">
  <loc> 507 1 0.8 </loc>
  <robscan base="1" pattern="PatternNameGoesHere" gap="0.15" width="0.4"
    mirrored="false" pattern_width="5" pattern_length="12" orientation_angle="0">
    <normal_direction x="0" y="0" z="-1"/>    <!-- locale z axis -->
    <tangential_direction x="1" y="0" z="0"/> <!-- locale x axis -->
  </robscan>
  <appdata>
    ...
  </appdata>
</connection_0d>

```

7.4 Rivets

There are many types of rivets. If at some state of the model the specific type of rivet (e. g. SPR, Self-Piercing Rivet) is not known, then a generic rivet element should be used to capture just the necessary information, like direction, length and diameter.

A rivet is denoted by an element **<rivet/>**. This element is described completely by its attributes and nested elements.

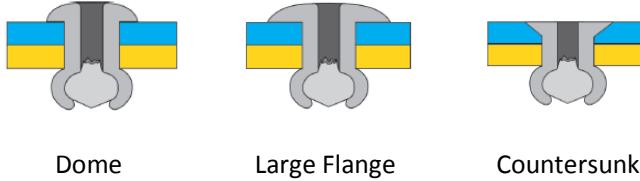
Nested Elements	Multiplicity	Status	Constraint / Remarks
rivet	0-1	Optional	-
loc	1	Required	-

Nested Elements	Multiplicity	Status	Constraint / Remarks
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 39: <connection_0d/> for <rivet/>

XML specification of <rivet/> element:

Attributes	Type	Value Space	Status	Constraint / Remarks
shaft_diameter	Floating point	> 0.0	Optional	-
length	Floating point	> 0.0	Optional	-
head_diameter	Floating point	> 0.0	Optional	-
head_height	Floating point	≥ 0.0	Optional	If at least one of them is specified $head_height + sink_size > 0$ is required.
head_type	Alphanumeric	Alphanumeric	Optional	-
sink_size	Floating point	≥ 0.0	Optional	-
strength_property_class	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	-

Table 40: Attributes of element <rivet/>

Dome

Large Flange

Countersunk

Source of image: <http://sfsintecusa.com/files/2011/09/Rivet-Brochure-Feb-2011.pdf>**Figure 8: Rivet head types**

- **shaft_diameter:** the diameter of the shaft of the (unmounted) rivet.
- **length:** is the overall length of the (unmounted) rivet itself.
- **head_diameter:** the diameter of the head of the (unmounted) rivet.
- **head_height:** the height of the head.
- **head_type:** description of head type ("dome", "countersunk" or "large_flange").
- **sink_size:** the size of the head that is sunk.
- **strength_property_class:** Strength according to ISO, EN, BSW, DIN, etc.

e.g.: **SAE J492 - Guide for Rivet Selection and Design Consideration**

- **part_code:** the part code of the rivet, as used e. g. in a PDM system. Frequently, it may be convenient to use the rivet norm (according to ISO, EN, BSW, DIN, ...) as part code.

If possible, a rivet should know the direction of fixation, i.e. possess a nested element <normal_direction>. However, this is not mandatory in order to allow for importing incomplete data. Direction sense of <normal_direction/> is from rivet head to foot, which element's definition can be found in section 7.1.3.

A `<rivet/>` is always placed into holes drilled before, whereas its subtype `<self_piercing/>` creates its own hole during placement.

Specific subtypes of rivets are defined by adding according nested elements, listed in following table:

Nested Elements	Multiplicity	Status	Constraint / Remarks
normal_direction	1	Optional	-
blind self_piercing solid swop	1	Optional	Maximum one of the listed elements.

Table 41: Nested elements of element `<rivet/>`

The subtypes are described in detail in the following sections.

Example:

```
<connection_0d label="RVT_2123921">
  ...
  <rivet shaft_diameter="5.0" head_diameter="8" length="3.5">
    <normal_direction x="0" y="0" z="3"/>
    <blind/>
    <loc> 1645.83 821.145 616.585 </loc>
    <appdata>
      ...
    </appdata>
  </connection_0d>
```

7.4.1 Blind Rivets

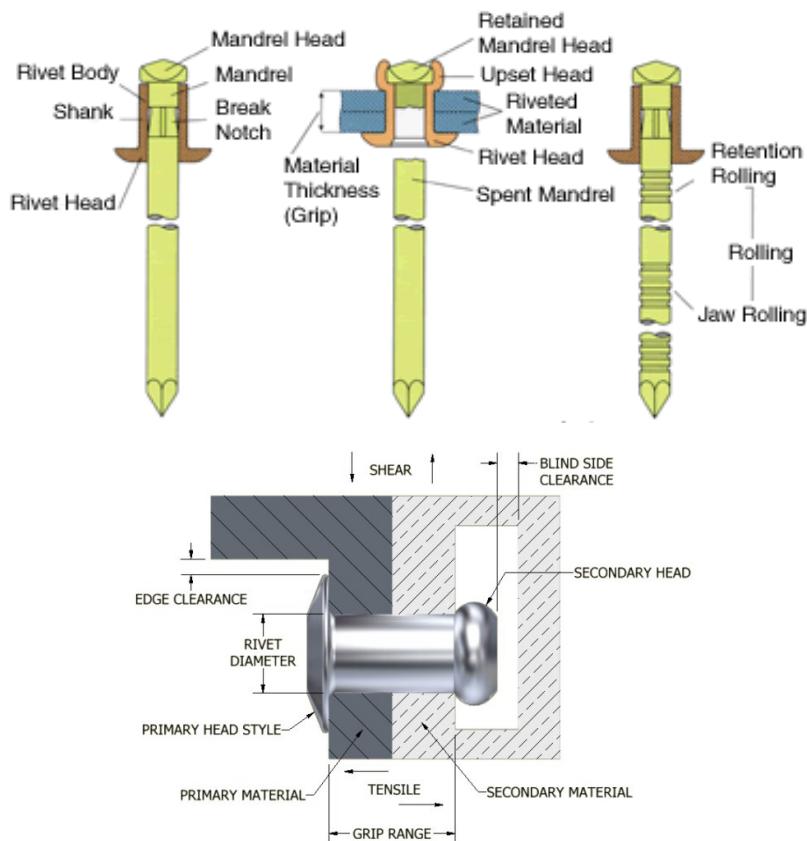
Blind rivets are one-sided rivets that require a pre-drilled hole. Blind rivets form their shape when the mandrel is pulled out from the rivet body. This action securely clamps the sheets together.

A blind rivet is denoted by a nested element `<blind/>` within `<rivet/>`. This element is described completely by its attributes and those of `<rivet/>`.

XML specification of `<blind/>` element:

Attributes	Type	Value Space	Status	Constraint / Remarks
min_grip	Floating point	> 0.0	Optional	-
max_grip	Floating point	> 0.0	Optional	greater equal to min_grip
clearance	Floating point	> 0.0	Optional	-
material	Alphanumeric	Alphanumeric	Optional	Material of the rivet body

Table 42: Attributes of element `<blind/>`



Source of image: <http://www.stanleyengineeredfastening.com/brands/pop/rivets/selection-factors>

Figure 9: Cross Section of a blind rivet

The pictures above describe what the attributes of `<rivet/>` and `<blind/>` correspond to:

- **`min_grip`, `max_grip`:** these two attributes collectively describe the effective grip range of the rivet. A blind rivet is engineered so that it can be used for a specific range of material thickness for which it provides proper joining between connected parts. This can be called as the blind rivet's grip range.
- ***Remark:*** In case of material thickness changes in connected parts might lead to other size of blind rivet as joining element!
- **`clearance`:** the blind rivet needs some clearance on the blind side, which is the side of mandrel head, when inserted into the holes, before it is applied.
- **`material`:** this attribute defines the applied material of the blind rivet body. Generally the applied rivet should be used with connected parts so that the connector rivet element has the same physical and mechanical properties as the components to be joined. (usual materials: Steel, Stainless Steel, Nickel Copper Alloy (Monel) Copper and several grades of Aluminum)

When a blind rivet is going to be applied to join 2 components which have different mechanical properties like one of them is thinner than the other or one of them is softer than the other, then the normal direction element will become more important to show the proper setting direction of the rivet as seen in the picture below:

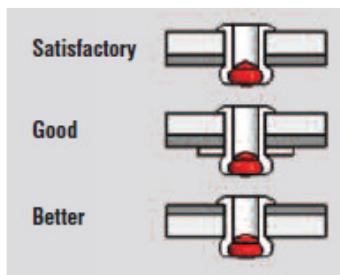


Figure 10: Thick and Thin Assembling

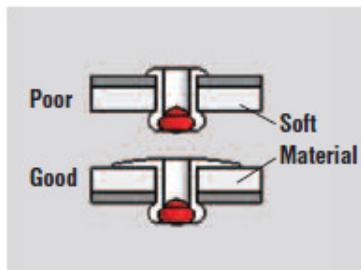


Figure 11: Fastening Soft and Hard

Example:

```
<connection_0d label="RVT_2123921">
    <rivet shaft_diameter="3.35" head_diameter="5.5" head_type="dome" length="4">
        <blind min_grip="3" max_grip="3.2" clearance="4.5" material="Steel"/>
        <normal_direction x="0.0" y="1.5" z="3.0"/>
    </rivet>
    <loc> 1645.83 821.145 616.585 </loc>
    <appdata>
        ...
    </appdata>
</connection_0d>
```

For further information about the Blind rivets you can check the following document:

<http://www.stanleyengineeredfastening.com/brands/pop/rivets>

7.4.2 Self-Piercing Rivets

A self-piercing rivet is a special kind of rivet which does not need a pre-drilled hole. Originally a hollow cylinder with a cap on one end, it deforms together with the material it is pushed into like sketched in following figure:

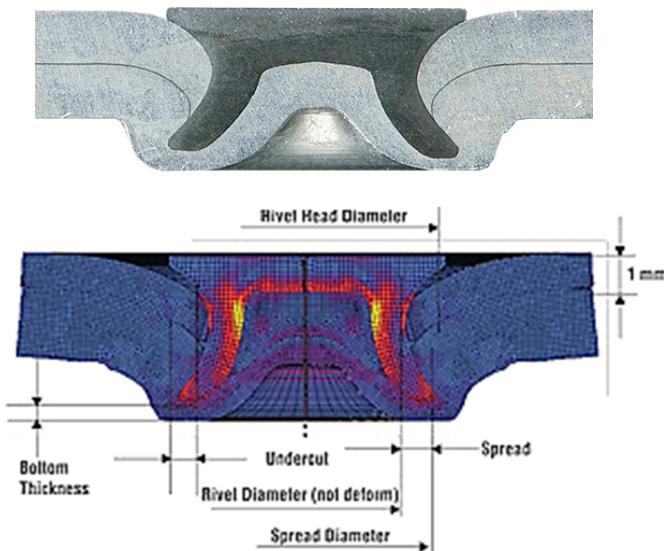
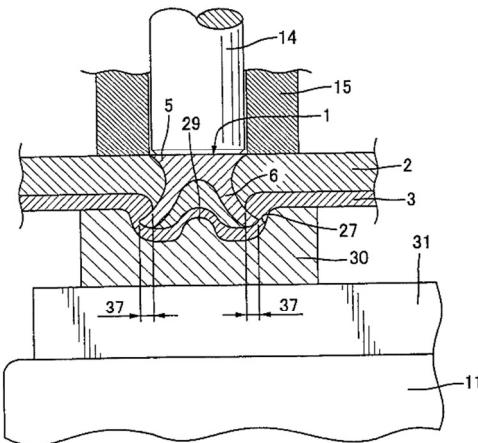


Figure 12: Cross Section of a Self-Piercing Rivet



Source of image: <http://www.google.com/patents/US7810231>

Figure 13: Self-piercing rivet setting apparatus

There is a wide range of such rivets available on the market. They can be used with different rivet dollies or dies (30) on the opposite side. Such combinations have to be chosen in accordance to the materials of the flange partners. Which combinations have been validated successfully is 3rd party intellectual property and hence cannot be part of xMCF definition. It is referred to by string attributes for rivet and die parameters. Possible values of these attribute are *not* subject of standard: In general, they are very OEM specific. However, to provide a minimum amount of information, some general geometric information is given by according attributes.

A self-piercing rivet is denoted by a nested element `<self_piercing/>` within `<rivet/>`. This element is described completely by its attributes and those of `<rivet>`. In especially, attributes "length", "head_diameter" and "shaft_diameter" are inherited from `<rivet>`.

XML specification of `<self_piercing/>` element:

Attributes	Type	Value Space	Status	Constraint
hardness	Floating point	> 0.0	Optional	-
head_label	Alphanumeric	Alphanumeric	Optional	-
shaft_label	Alphanumeric	Alphanumeric	Optional	-
die_label	Alphanumeric	Alphanumeric	Optional	-
die_diameter	Floating point	> 0.0	Optional	-
die_depth	Floating point	> 0.0	Optional	-

Table 43: Attributes of element `<self_piercing/>`

Attribute `hardness` is Vickers hardness HV of the rivet material.

All attributes of this connection are optional for import to CAD or CAE processors. However, specific FE solvers may declare some of them to be mandatory.

The `head`, `shaft` and die labels are very OEM specific. However, to provide a minimum amount of information, diameters of them plus depth of die are given.

Attribute `die_label` can be used to refer to a catalog entry. Then `die_diameter` and `die_depth` can be omitted in xMCF file, if their values are given in catalog.

One level higher, the entire rivet can refer to an item via attribute which refers to an item that is being used in OEM Specific PDM system. In this case, subtype definition is used from catalog, too, if present. The `<rivet/>` in xMCF file *must not* specify another subtype than the referred item from the PDM system!

General defaults for attributes are: 0 for numeric values, "" for strings. However, these defaults are not always useful for CAE.

Example:

```

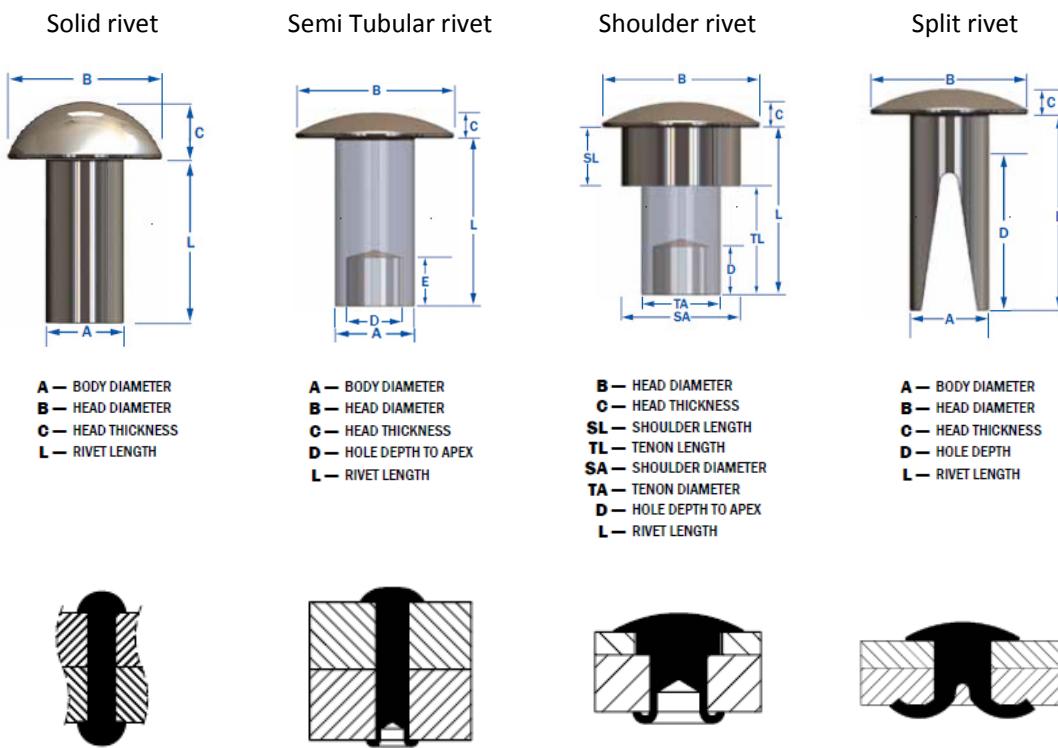
<connection_0d label="RVT_2123921">
    <rivet shaft_diameter="3.35" head_diameter="5.5" length="4">
        <normal_direction x="0" y="0" z="3"/>
        <self_piercing hardness="410" head_label="N00000002651" shaft_label="C"
            die_depth="2.5" die_label="DZ11x2,5-0,50" die_diameter="11" />
    </rivet>
    <loc> 1645.83 821.145 616.585 </loc>
    <appdata>
        ...
    </appdata>
</connection_0d>

```

7.4.3 Solid Rivets

Solid rivets require a pre-drilled hole. They can be found in many similar forms, with a cap on one end. The other end deforms when it is pushed from the other side.

The shafts of solid rivets are typically solid, but for all rivets that have similar shapes, this type will be used:



Source of image: http://www.rivet.com/Catalog_CompleteVersion/ImpactOnly-2-03-12.pdf

Table 44: Pictures of all Solid Rivets

The dimensions of all these rivets generalize into the following diagram:

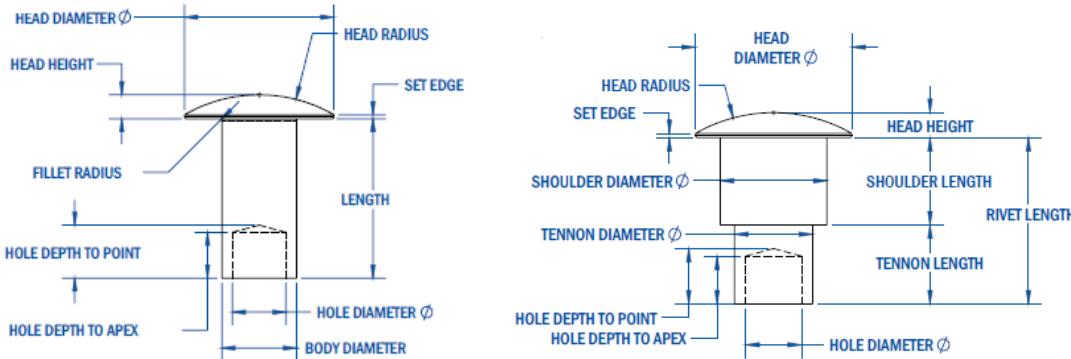


Figure 14: Dimensions of Solid Rivets

A solid rivet is denoted by a nested element `<solid/>` within `<rivet/>`. This element is described completely by its attributes and those of `<rivet/>`.

XML specification of `<solid/>` element:

Attributes	Type	Value Space	Status	Constraint
min_grip	Floating point	> 0.0	Optional	-
max_grip	Floating point	> 0.0	Optional	-
hole_diameter	Floating point	> 0.0	Optional	-
hole_depth	Floating point	> 0.0	Optional	-

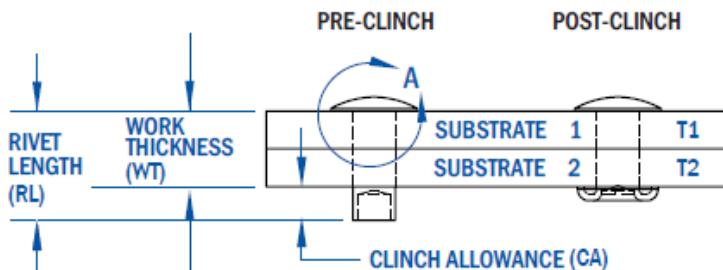
shoulder_diameter	Floating point	> 0.0	Optional	-
shoulder_length	Floating point	> 0.0	Optional	-
tennon_diameter	Floating point	> 0.0	Optional	-
tennon_length	Floating point	> 0.0	Optional	-

Table 45: Attributes of element <*solid*>**Recommendations:**

1. *hole_diameter* is defined with *hole_depth* and vice versa.
2. *tennon_diameter* exist only if *shoulder_diameter* is defined and vice versa.

The pictures above describe what the attributes of *<rivet>* and *<solid>* correspond to:

- *min_grip*, *max_grip*: these two attributes collectively describe the effective grip range.
- *hole_diameter*: this is the diameter of the hole of the tube. This value is provided in a supplier standard normally.
- *hole_depth*: this is a measure of the hole of the tube. There is no exact relation between *hole_depth* and grip range. Based on the supplier it might be a length calculation that could be result in an advised clinch allowance based on the work thickness calculated by the sum of the thicknesses of connected parts.

**Figure 15: Clinch allowance of solid rivet**

- *shoulder_diameter*, *shoulder_length*: the rivet's shoulder sizes. Note that shoulder length is typically measured next under the head.
- *tennon_diameter*, *tennon_length*: these attributes describe the secondary smaller shoulder sizes. A *tennon_diameter* should not exist without a primary *shoulder_diameter*.

If a *head_height* exists, *sink_size* will be 0, and vice versa. But there is no constraint in xMCF.

Example:

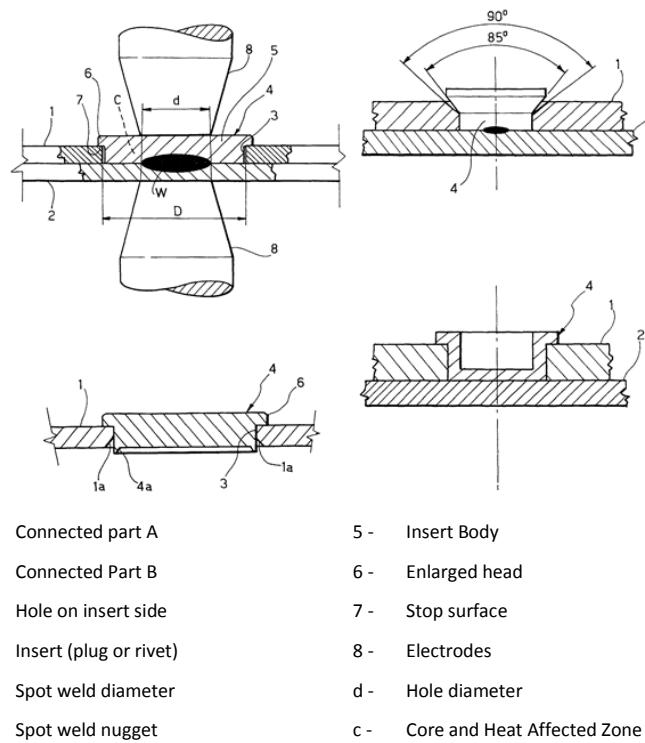
```

<connection_0d label="RVT_2123921">
  <loc> 1645.83 821.145 616.585 </loc>
  <rivet shaft_diameter="3.35" head_diameter="5.5" head_height="0.4" length="4">
    <solid min_grip="3" max_grip="3.2" hole_depth="0.8"
      shoulder_diameter="3.8" shoulder_length="1.2"/>
    <normal_direction x="0" y="1.5" z="3"/>
  </rivet>
  <appdata>
    ...
  </appdata>
</connection_0d>

```

7.4.4 Swop Rivets

“SWOP” stands for “Sheet Weld Opposed Plug”. The method is used to connect parts with spot welds in cases where one component material is not suitable to create any alloy with the other part’s material. Typically it is the case when aluminum and steel parts are going to be connected.



Source of image: <https://www.google.com.ar/patents/EP0967044A2?cl=en&hl=de>

Figure 16: Cross section of a SWOP Rivet

The following description is quoted from the Patent documented under **EP 0967044 A2**:

“A sheet (1) of a material which cannot be electrically welded is connected to a second sheet (2) of a weldable material by providing a through hole (3) in the first sheet having a transverse dimension substantially greater than the transverse dimension of the tips of the electrodes (8) which are used for carrying out the electric welding spot. Within the hole (3) there is provided an insert (4) of a material which can be electrically welded. The two electrodes (8) are applied so as to cause welding of the insert (4) to the second sheet (2). The insert (4) has a stop surface (7) which prevents the first sheet (1) from separating from the second sheet (2) after welding. The difference of the transverse dimensions of the tip of each electrode (8) and the insert (4) avoids any possible overheating up to the melting point of the material constituting the first sheet (1) while welding is carried out.”

Based on the description above one can imagine a wide range of insert shapes. Hence, they cannot be part of xMCF definition. The shape is referred by a string attribute `insert_shape`. The possible values of this attribute are *not* subject of the standard: in general, this is an entry which is very OEM specific. However, to provide a minimum amount of information some general geometric data are given by the attributes introduced below.

A swop rivet is denoted by a nested element `<swop/>` within `<rivet/>`. This element is described completely by its attributes and parent element attributes within `<rivet/>`. Especially the attributes `shaft_diameter`, `sink_size`, `length`, `head_diameter` and `head_height` are inherited from

<*rivet*> element. Other rivet parameters (e.g.: *length* or *shaft_diameter*) may be treated meaningless.

XML specification of <*swop*> element:

Attributes	Type	Value Space	Status	Constraint
insert_shape	Alphanumeric	Alphanumeric	Optional	-
insert_height	Floating point	> 0.0	Optional	-
spotweld_diameter	Floating point	> 0.0	Optional	-
spotweld_technology	Selection	resistance laser projection friction	Optional	-

Table 46: Attributes of element <*swop*>

All attributes of this connection type are optional for importing it into CAD or CAE application. Although, it can be that some FE pre-processor may declare some of them to be mandatory.

These attributes have the following semantics:

- ***insert_shape***: Identification of the applied insert shape. (In the illustrated example, the hole is circular, but it may have a polygonal shape in order to prevent relative rotation of the two sheets in case they were connected by a single framing spot.)
- ***insert_height***: Height of the (unmounted) insert.
- ***spotweld_diameter***: Diameter of the spot weld, see section 7.2 Spot Welds.
- ***spotweld_technology***: Technology of the spot weld, see section 7.2 Spot Welds.

The element of <*swop*> does not allow any nested elements:

Example:

```

<connection_0d label="RVT_2123921">
    <loc> 1645.83 821.145 616.585 </loc>
    <rivet head_diameter="8.5" head_height="0.9" hardness="410" shaft_diameter="5.4"
        sink_size="0.3" length="1.5" >
        <normal_direction x="0" y="0" z="3" />
        <swop insert_shape="cone_23" insert_height="1.8"
            spotweld_diameter="4.5" spotweld_technology="resistance" />
    </rivet>
    <appdata>
        ...
    </appdata>
</connection_0d>

```

7.5 Threaded Connections: Bolts and Screws

7.5.1 Introduction

Bolts and screws are probably the most well-known connection techniques, even within non-specialists. However, they do need a closer look at their details. This starts, but does not end with the differentiation between screws and bolts:

- A screw has a tapped bore.
- A bolt needs a nut.

On the other hand: What differentiates a nut from any other part, which carries a cut-in inner thread?

It is probably just the fact that nuts are standardized and combinations of bolts with fitting screws are available “ready-to-use” in most software systems as well as in real life.

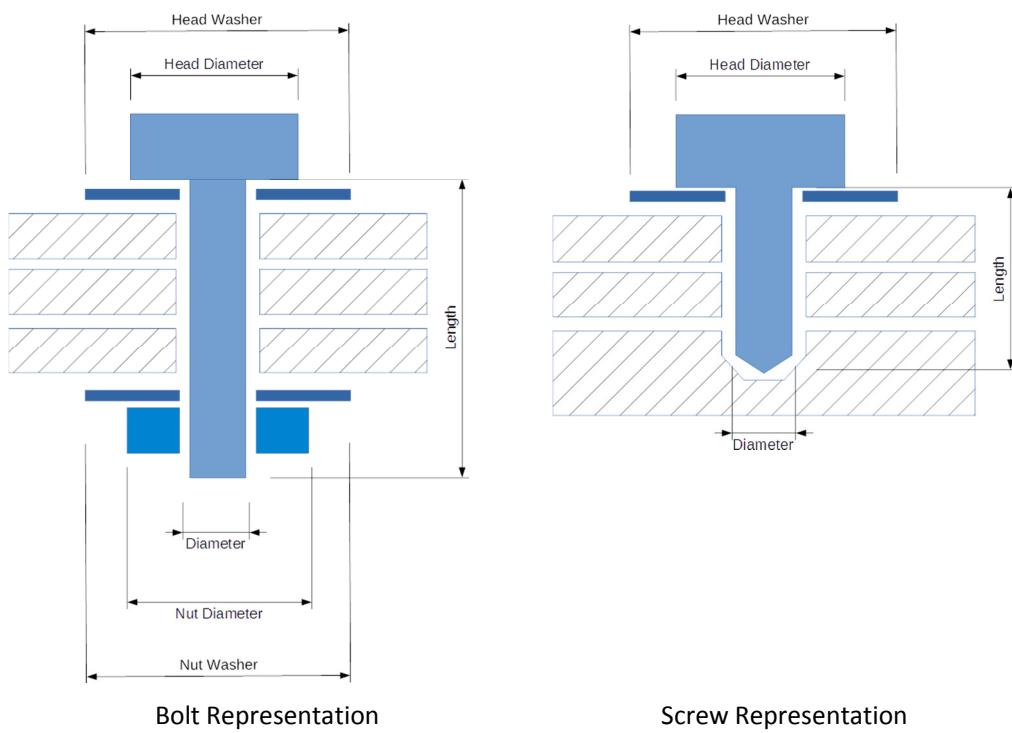
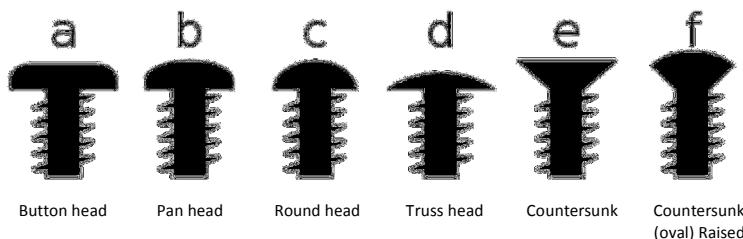


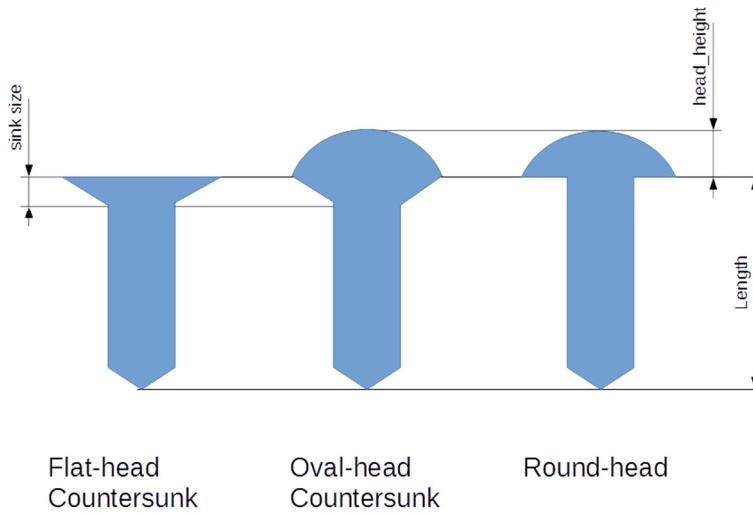
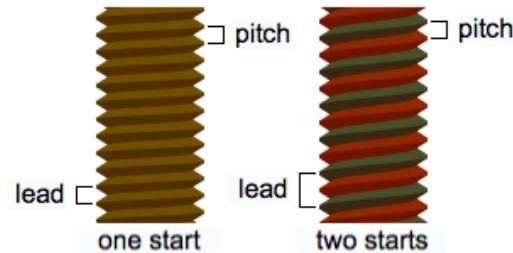
Figure 17: Bolts and Screws



Source of image: http://commons.wikimedia.org/wiki/File:Screw_head_types.svg.

Author published it under the “under the [Creative Commons Attribution-Share Alike 3.0 Unported](#) license”.

Figure 18: Different Screw Forms

**Figure 19: Definition of Length and Head Sizes**

Source of image: http://upload.wikimedia.org/wikipedia/commons/0/00/Lead_and_pitch.png.

Figure 20: Definition of lead, pitch and starts of a thread.

7.5.2 Contacts and Friction

Self-loosening of screws and bolts has to be prevented. Static friction, together with pretension, is a major means against self-loosening. However, kinetic (or dynamic) friction also has some meaning in CAE processes. Hence, both friction types need to be supported by χ MCF.

Friction occurs between any two objects in contact. In case of bolts and screws, these contacts are usually obtained between:

1. head and washer (if there is a “loose” washer, i.e. a washer not being fixed to the head or shaft)
2. washer (if there is one) and first connected part, or else
3. head and first connected part
4. part n and $n+1$, with $n=1 \dots N-1$ and $N := \text{cardinality of } \langle \text{connected_to} \rangle$
5. part N and “loose” washer (if there is one)
6. washer (if there is one) and nut, or else
7. part N and nut
8. screw/bolt thread (outer thread) and nut thread/cut thread (inner thread)

Consequently, χ MCF assigns friction attributes to

- a. heads and nuts, applying to their contacts to either washers or adjacent parts,
- b. washers, applying to their contacts to adjacent parts (*not* to head or nut),
- c. any contact $n=1 \dots N-1$ between each two adjacent parts,
- d. the screw thread (assuming that there always is exactly one item with an inner thread).

For heads (as a constituent part of a screw or bolt), nuts and washers, there are specific XML elements in χ MCF. According friction attributes are located, there.

In case of c. above of inter-part contacts is addressed by the following XML elements.

XML specification of `<contact_list>` element:

Nested Elements	Multiplicity	Status	Constraint
contact	1 - *	Required	Maximum no. of items: <i>cardinality of <connected_to></i> - 1

Table 47: Nested elements of element `<contact_list>`

The element `<contact_list>` does not allow for any attributes.

XML specification of `<contact/>` element:

Attributes	Type	Value Space	Status	Constraint
index	Integer	> 0	Optional	Must be one of the indices defined in <code><connected_to></code> , but not the last one of them. Any specific index may occur not more than once.
thread	Boolean	“false” (default) “true”	Optional	If “true”, attribute index is ignored.
static_friction	Floating point	> 0.0	Optional	At least one of these two friction coefficients has to be specified.
kinetic_friction	Floating point	> 0.0	Optional	See above.

Table 48: Attributes of element `<contact/>`

These attributes have following semantics:

- **index:** This `<contact/>` attribute refers to the contact between the flange partner with this index (see section 5.3.1.1) and the following one.
Notice that this constitutes the only place in χ MCF, where ordering (i. e. correct ascending numbering sequence) of part indices within `<connected_to>` element matters!
- **thread:** Value “true” indicates that the contact is between the outer thread of a screw/bolt and the inner thread of a nut/cut thread. Then the attribute `index` is ignored.
- **static_friction:** the static friction of this contact.
- **kinetic_friction:** the kinetic friction of this contact.

The element `<contact/>` does not allow for any nested elements.

Example A (with washers):

```
<connection_group id="1">
  <connected_to>
    <part index="1" label="PART_7000400"/>
    <part index="2" label="PART_7100100"/>
    <part index="5" label="PART_5000300"/>
    <part index="6" label="PART_5000800"/>
  </connected_to>
  <connection_list>
    <connection_0d label="135"> <!-- bolt with washers -->
      <loc> 84 60 10 </loc>
      <!-- Friction is "head to washer": -->
      <threaded_connection length="50" static_friction="0.8" >
        <normal_direction x="0" y="0" z="-10"/>
        <!-- Washer next to head with its friction to 1st part -->
        <washer outer_diameter="20" attached="false" static_friction="0.8" />
        <bolt>
          <!-- Friction is "nut to washer" -->
          <nut diameter="16." static_friction="0.8">
            <!-- Washer next to nut with its friction to last part -->
            <washer outer_diameter="25" attached="false" static_friction="0.8" />
          </nut>
        </bolt>
      <contact_list>
        <!-- Friction within thread -->
        <contact thread="true" static_friction="0.8" />
        <!-- Friction part 1 to 2 -->
        <contact index="1" static_friction="0.8" />
        <!-- Friction part 2 to 5 -->
        <!-- Assuming, indices 3 & 4 are not defined in <connected_to/> -->
        <contact index="2" static_friction="0.8" />
        <!-- Friction part 5 to last part -->
        <!-- Assuming there is exactly one more index after 5 in <connected_to/> -->
        <contact index="5" static_friction="0.8" />
      </contact_list>
    </threaded_connection>
  </connection_0d>
  </connection_list>
</connection_group>
```

Example B (without washers):

```

<connection_group id="1">
  <connected_to>
    <part index="1" label="PART_7000400"/>
    <part index="2" label="PART_7100100"/>
    <part index="5" label="PART_5000300"/>
    <part index="6" label="PART_5000800"/>
  </connected_to>
  <connection_list>
    <connection_0d label="135" > <!-- bolt with washers -->
      <loc> 84 60 10 </loc>
      <!-- Friction is "head to washer": -->
      <threaded_connection length="50" static_friction="0.8" >
        <normal_direction x="0" y="0" z="-10"/>
        <!-- Washer next to head with its friction to 1st part -->
        <washer outer_diameter="20" attached="false" static_friction="0.8" />
        <bolt>
          <!-- Friction is "nut to washer" -->
          <nut diameter="16." static_friction="0.8">
            <!-- Washer next to nut with its friction to last part -->
            <washer outer_diameter="25" attached="false" static_friction="0.8" />
          </nut>
        </bolt>
        <contact_list>
          <!-- Friction within thread -->
          <contact thread="true" static_friction="0.8" />
          <!-- Friction part 1 to 2 -->
          <contact index="1" static_friction="0.8" />
          <!-- Friction part 2 to 5 -->
          <!-- Assuming, indices 3 & 4 are not defined in <connected_to/> -->
          <contact index="2" static_friction="0.8" />
          <!-- Friction part 5 to last part -->
          <!-- Assuming there is exactly one more index after 5 in <connected_to/> -->
          <contact index="5" static_friction="0.8" />
        </contact_list>
      </threaded_connection>
    </connection_0d>
  </connection_list>
</connection_group>

```

7.5.3 Definition of element *<threaded_connection>*

Due to its similar characters, bolts and screws share a couple of common attributes. Hence, in order to reduce redundancy, they are subsumed beneath a common, more abstract XML element *<threaded_connection>*.

Nested Elements	Multiplicity	Status	Constraint / Remarks
threaded_connection	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 49: Nested elements of *<threaded_connection>*

Element “loc”

The syntax of this element is described in the corresponding section 7.1.2 Location.

Element “appdata”

This follows the same syntax as defined in section 5.2.1 User Specific Data *<appdata>*.

Element “threaded_connection”

XML specification of `<threaded_connection/>` element with the following attributes :

Attributes	Type	Value Space	Status	Constraints / Remarks
diameter	Floating point	> 0.0	Optional	
length	Floating point	> 0.0	Optional	-
thread_length	Floating point	> 0.0	Optional	length ≥ thread_length
head_diameter	Floating point	> 0.0	Optional	-
head_height	Floating point	≥ 0.0	Optional	If at least one of them is specified, <i>head_height + sink_size > 0</i> is required.
head_type	Alphanumeric	Alphanumeric	Optional	-
sink_size	Floating point	≥ 0.0	Optional	Usually, sink_size > 0 implies no washer.
pitch	Floating point	> 0.0	Optional	Not to be confused with “lead”.
lead	Floating point	> 0.0	Optional	In case of single-start threadform pitch is equal to lead. Default value is equal to pitch attribute!
torque	Floating point	> 0.0	Optional	-
angle	Floating point	> 0.0	Optional	-
pretension	Floating point	> 0.0	Optional	-
static_friction	Floating point	> 0.0	Optional	
kinetic_friction	Floating point	> 0.0	Optional	-
strength_property_class	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	-

Table 50: Attributes of element `<threaded_connection/>`

These attributes have following semantics:

- **diameter**: the diameter of the bolt or screw. It should be provided, since e. g. only few CAE simulation types can live without it.
- **length**: the length of the bolt or screw. Refer to Figure 19: Definition of Length and Head Sizes.
- **thread_length**: the length of the thread of the bolt or screw. Only needed in case of a partial-thread screw. In case of a full-thread screw, thread continues from tip to head, without a non-threaded area. Then, *thread_length = length – sink_size*.
- **head_diameter**: the diameter of the head of the bolt or screw
- **head_height**: the height of the head.
- **head_type**: Type of screw head, e. g. “outer hexagonal”, “flanged-hex/Phillips-head combi”, “external torx plus”. Since there is a wide variety and ever increasing of screw head types, alphanumeric type is appropriate for this attribute.
- **sink_size**: the size of the head that is sunk (for countersunk screws).

- ***pitch***: is the distance from the crest of one thread to the next.¹¹
- ***lead***: is the distance along the screw's axis that is covered by one complete rotation of the screw (360°). Lead and pitch are parametrically related, and the parameter that relates them, the number of starts (number of single thread), very often has a value of 1, in which case their relationship becomes equality. In general, lead is equal to S times pitch, in which S is the number of starts.
- ***torque***: The torque which should be applied when fastening the bolt or screw.
- ***angle***: The turning angle which should be applied when fastening the bolt or screw.
- ***pretension***: The pretension which is generated within the bolt or screw when fastening.
- ***static_friction***: The static friction between head and adjacent washer or part.
- ***kinetic_friction***: The kinetic friction between head and adjacent washer or part.
- ***strength_property_class***: Strength according to applied standard within a unique part supplier or OEM.
- ***part_code***: the part code of the bolt or screw, as used e. g. in a PDM system. Frequently, it may be convenient to use the screw norm (according to ISO, EN, BSW, DIN, ...) as part code.

Torque, pretension and angle interact as follows:

- Torque is only applied, if no pretension is given.
- Angle is only applied, if torque is given and no pretension is present.

For bolts as well as screws, it is recommended to provide the direction of fixation. Hence `<threaded_connection/>` offers following nested elements:

<i>Nested Elements</i>	<i>Multiplicity</i>	<i>Status</i>	<i>Constraint</i>
normal_direction	1	Optional	-
bolt screw	1	Required	Exactly one of these elements.
washer	1	Optional	-
contact_list	1	Optional	-

Table 51: Nested elements of element `<threaded_connection/>`

Element “normal_direction”

The direction of the bolt or screw is described by the element `<normal_direction/>` in form of an orientation vector. This is necessary to define the orientation of the bolt or screw and hence which end is considered to be the connection's head-side. The orientation sense of the bolt is *from head to nut* and of the screw is *from head to point*.

Refer to section 7.1.3 for syntax of element `<normal_direction/>`.

Elements `<bolt/>`, `<screw/>` and `<washer/>` are described in the following sections.

The nested element `<washer/>` refers to the washer next to the head of a screw or bolt.

The nested element `<contact_list>` is described in section 5.3.2.1.

All attributes of threaded connections are optional for import to CAD or CAE processors. However, specific FE solvers may declare some of them to be mandatory.

¹¹ For more details, see http://en.wikipedia.org/wiki/Screw_thread#Lead.2C_pitch.2C_and_starts.

General defaults are: 0 for numeric values, "" for strings. However, these defaults are not always useful for CAE.

7.5.4 Washer

Bolts and screws are frequently combined with washers. Hence, we define the XML element `<washer>`.

Attributes	Type	Value Space	Status	Constraints / Remarks
outer_diameter	Floating point	> 0.0	Optional	-
inner_diameter	Floating point	> 0.0	Optional	Usually NO inner diameter, if attached.
thickness	Floating point	> 0.0	Optional	-
attached	Boolean	"false" (default), "true"	Optional	-
static_friction	Floating point	> 0.0	Optional	-
kinetic_friction	Floating point	> 0.0	Optional	-
strength_property_class	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	NO part code, if attached.

Table 52: Attributes of element `<washer>`

These attributes have following semantics:

- **outer_diameter**: the outer diameter of the washer. If a washer is used at all, at least its diameter must be specified.
- **inner_diameter**: the inner or hole diameter of the washer
- **thickness**: the thickness of the washer
- **attached**: true, if and only if the washer is firmly attached to the screw head or nut, resp.
- **static_friction**: the static friction between this washer and its adjacent part (not head or nut)
- **kinetic_friction**: the kinetic friction between this washer and its adjacent part (not head or nut)
- **strength_property_class**: Strength according to applied standard within a unique part supplier or OEM.
- **part_code**: the part code of the washer, as used e. g. in a PDM system. Frequently, it may be convenient to use the washer norm as part code.

The element `<washer>` does not allow for any nested elements.

7.5.5 Nut

Any bolt requires a nut. But since nuts may have a number of own attributes, it is worth to define a separate XML element for them.

Attributes	Type	Value Space	Status	Constraints / Remarks
diameter	Floating point	> 0.0	Optional	-
height	Floating point	> 0.0	Optional	-

Attributes	Type	Value Space	Status	Constraints / Remarks
torque	Floating point	> 0.0	Optional	-
angle	Floating point	> 0.0	Optional	-
static_friction	Floating point	> 0.0	Optional	-
kinetic_friction	Floating point	> 0.0	Optional	-
clipped_to	Integer	> 0	Optional	-
fixed_to	Integer	> 0	Optional	-
strength_property_class	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	NO part code, if attached.

Table 53: Attributes of element <nut/>

These attributes have following semantics:

- *diameter*: the diameter of the nut
- *height*: the height of the nut.
- *torque*: The torque which should be applied when fastening the nut.
- *angle*: The turning angle which should be applied when fastening the nut.
- *static_friction*: The static friction between nut and adjacent washer or part.
- *kinetic_friction*: The kinetic friction between nut and adjacent washer or part.
- *clipped_to*: The nut is fixed with a clip or it is a clip itself. It is clipped to the flange partner with this index (see section 5.3.1.1). If attribute is missing, nut is not clipped. Nut and clip share a common part code, i. e. they are regarded to be one single part.
- *fixed_to*: The nut is firmly fixed (e.g. welded) to the flange partner with this index (see section 5.3.1.1). If attribute is missing, nut is not fixed.
- *strength_property_class*: Strength according to applied standard within a unique part supplier or OEM.
- *part_code*: the part code of the nut, as used e. g. in a PDM system. Frequently, it may be convenient to use the nut norm (according to ISO, EN, BSW, DIN, ...) as part code.

Usually nut *fixed_to* prohibits nut *clipped_to* and vice versa.

Usually nut *clipped_to* or *fixed_to* prohibits bolt *clipped_to* or *fixed_to* and vice versa.

There are other means of fixating nuts to sheets, as well, e.g. punching or riveting.

The element <nut/> allows following nested elements:

Nested Elements	Multiplicity	Status	Constraint
washer	1	Optional	-

Table 54: Nested elements of element <nut/>

The nested element <washer/> refers to the washer next to the nut of the bolt.

7.5.6 Bolt

A bolt connection is denoted by an element <bolt/>. This element is described completely by its attributes and nested elements.

Element "bolt"

For the `<bolt/>` element, the following attributes are allowed:

Attributes	Type	Value Space	Status	Constraint
clipped_to	Integer	> 0	Optional	-
fixed_to	Integer	> 0	Optional	-

Table 55: Attributes of element `<bolt/>`

- **`clipped_to`**: The head of the bolt is fixed with a clip to the flange partner with this index (see section 5.3.1.1). If attribute is missing, bolt is not clipped. Bolt and clip share a common part code, i.e. they are regarded to be one single part.
- **`fixed_to`**: The head of the bolt is fixed (eg. welded) to the flange partner with this index (see section 5.3.1.1). Also applies if there is no screw head at all, e.g. this bolt actually is stay bolt or stud. If attribute is missing, bolt is not fixed.

There is no "base" attribute for bolts, since this information can be derived from connection direction.

Usually bolt `fixed_to` prohibits bolt `clipped_to` and vice versa.

The element `<bolt/>` allows following nested elements:

Nested Elements	Multiplicity	Status	Constraint
nut	1	Required	Required, because no nut implies screw

Table 56: Nested elements of element `<bolt/>`

The nested element `<nut/>` refers to the bolt's nut. This, in turn, may contain a nested element `<washer/>`.

The nested element `<nut/>` is required by definition of `<bolt/>`. The nut itself (rsp. its `part_code` or property) must *not* be mentioned in element `<connected_to>` of the `<connection_group/>` containing the `<bolt/>`. This allows keeping other connection types (glue, rivets ...) in the same `<connection_group/>`.

Example A:

```

<connection_0d label="100532">
    <threaded_connection diameter="10.0" length="50.0"
        head_diameter="16.0" head_height="5" sink_size="3" diameterheight>
        <normal_direction> x="3.0" y="0.0" z="0.0" />
        <!--magnitude is irrelevant, direction sense is from head to nut-->
        <bolt>
            <nut diameter="16." height="5">
                <washer outer_diameter="20"/>
            </nut>
        </bolt>
        <washer outer_diameter="20">
    </threaded_connection>
    <loc> 1500.3 838.7 730.6 </loc>
    <appdata>
        ...
    </appdata>
</connection_0d>

```

Example B:

```
<connection_0d label="135">
  <threaded_connection diameter="10.0" length="50.0"
    head_diameter="16.0" head_height="5" thread_length="35"
    torque="80" angle="30" pretension="1200" part_code="M10x50 12.9" >
    <normal_direction x="0" y="0" z="-10"/>
    <!-- Washer next to head -->
    <washer outer_diameter="20" inner_diameter="10.3" thickness="1.5"
      attached="false" part_code="M10x20x1.5"/>
    <bolt fixed_to="1" >
      <nut diameter="16." height="5" static_friction="0.8">
        <!-- Washer firmly attached to nut -->
        <washer outer_diameter="25" thickness="1.5" attached="true"/>
      </nut>
    </bolt>
  </threaded_connection>
  <loc> 1500.3 838.7 730.6 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

Example C:

```
<connection_0d label="135">
  <threaded_connection length="50" diameter="10"
    head_diameter="16" head_height="5" thread_length="35"
    torque="80" angle="30" pretension="1200" part_code="M10x50 12.9">
    <normal_direction x="0" y="0" z="-10"/>
    <!-- Washer next to head -->
    <washer outer_diameter="20" inner_diameter="10.3" thickness="1.5"
      attached="true" part_code="M10x20x1.5"/>
    <bolt>
      <nut diameter="16." height="5" static_friction="0.8" clipped_to="4"/>
    </bolt>
  </threaded_connection>
  <loc> 1500.3 838.7 730.6 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

Example D (using every attribute, as many as possible):

```

<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="xmcf_3_0_0.xsd">
  <version> 3.0 </version>
  <date> 2016-01-08 </date>
  <units length="mm" angle="deg" mass="kg" force="N" torque="Nm" time="s"/>
  <connection_group id="1">
    <connected_to>
      <part index="1" label="PART_7000400"/>
      <part index="2" label="PART_7100100"/>
      <part index="5" label="PART_5000300"/>
      <part index="6" label="PART_5000800"/>
    </connected_to>
    <connection_list>
      <connection_0d label="135"> <!-- bolt with washers -->
        <loc> 84 60 10 </loc>
        <!-- Friction is "head to washer": -->
        <threaded_connection diameter="10" length="50" thread_length="26"
          head_diameter="16" head_height="6.4" head_type="hexagonal"
          sink_size="0" pitch="0.75" lead="1.5"
          torque="20" angle="35" pretension="180"
          static_friction="0.8" kinetic_friction="0.6"
          strength_property_class="8.8"
          part_code="M10x50 8.8">
          <normal_direction x="0" y="0" z="-10"/>
          <!-- Washer next to head with its friction to 1st part -->
          <washer outer_diameter="20" inner_diameter="10.4" thickness="1.25"
            attached="false"
            static_friction="0.8" kinetic_friction="0.6"
            strength_property_class="8.8"
            part_code="W20/10.4x1.25 8.8"/>
        <bolt>
          <!-- No Friction nut to washer, since washer is attached! -->
          <nut diameter="16." height="6.4"
            torque="20" angle="35"
            clipped_to="6"
            strength_property_class="8.8"
            part_code="N10 8.8">
            <!-- Washer attached to nut with its friction to last part -->
            <washer outer_diameter="25" attached="true" static_friction=".8"/>
          </nut>
        </bolt>
      </connection_0d>
      <contact_list>
        <!-- Friction between thread and nut: -->
        <contact thread="true" static_friction="0.8" kinetic_friction="0.6"/>
        <!-- Friction part 1 to 2: -->
        <contact index="1" static_friction="0.8"/>
        <!-- Friction part 2 to 5: -->
        <!-- Assuming, indices 3 & 4 are not defined in <connected_to/> -->
        <contact index="2" static_friction="0.8"/>
        <!-- Friction part 5 to last part: -->
        <contact index="5" static_friction="0.8"/>
      </contact_list>
    </threaded_connection>
  </connection_list>
</connection_group>
</xmcf>

```

7.5.6.1 Possible Bolt and Screw Assemblies

Altogether, there are following cases of assembly:

1. Bolt with welded nut (to the bottom sheet):

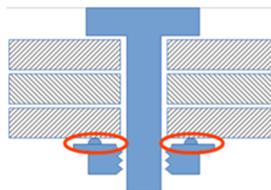


Figure 21: Bolt with welded nut

Example:

```
<connection_0d label="135">
  <threaded_connection diameter="10.0" length="50.0"
    head_diameter="16.0" head_height="5" thread_length="35"
    torque="80" angle="30" pretension="1200" part_code="M10x50 12.9" >
    <normal_direction x="0" y="0" z="-10"/>
    <!--No Washer in this case-->
    <bolt>
      <nut diameter="16." height="5" fixed_to="3" />
    </bolt>
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

2. Bolt with clipped nut (clipped to the bottom sheet): This is the *same*, only *fixed_to* is replaced by *clipped_to*.
3. Bolt with free nut (not clipped, nor welded to the bottom sheet):

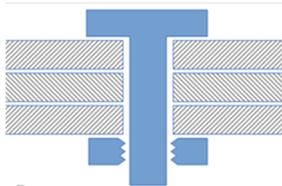


Figure 22: Bolt with free nut

(Since both, the screw and the nut are free, there is *no fixed_to* nor *clipped_to* attribute)

4. Screw without nut (screwed to the last sheet):

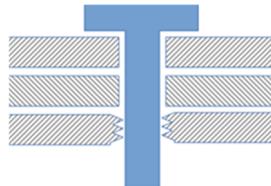


Figure 23: Screw without nut

Example:

```
<connection_0d label="135">
  <threaded_connection diameter="10.0" length="50.0"
    head_diameter="16.0" head_height="5" thread_length="35"
    torque="80" angle="30" pretension="1200" part_code="M10x50 12.9" >
    <normal_direction x="0" y="0" z="-10"/>
    <!--No Washer under bolt ehad in this case-->
    <bolt fixed_to="3" >
      <!--No Nut in this case-->
    </bolt>
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

5. Welded stud (with a free nut, of course):

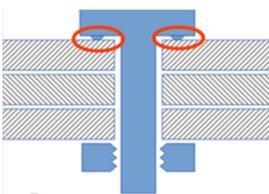


Figure 24: Welded stud with free nut

Example:

```
<connection_0d label="135">
  <threaded_connection diameter="10" length="50" head_diameter="16" head_height="5"
    thread_length="35" torque="80" angle="30" pretension="1200"
    part_code="M10x50 12.9" >
    <normal_direction x="0" y="0" z="-10"/>
    <!--No Washer in this case-->
    <bolt fixed_to="1" >
      <nut diameter="16." height="5" />
    </bolt>
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

6. Plain stud (with a nut on one end, screwed into a part on the opposite end):

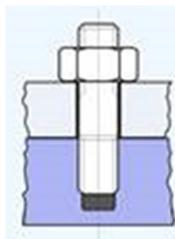


Figure 25: Plain stud

These studs are not a feature of xMCF version 3.0. They can be modeled according to case 4, but may become a topic of version 3.1.

In all cases the `<connected_to/>` element connects only the part codes or property IDs of the connected sheets.

7.5.7 Screw

A screw connection is denoted by an element `<screw/>`. This element is described completely by its attributes and nested elements.

Element “screw”

For the `<screw/>` element, the following attributes can be specified:

Attributes	Type	Value Space	Status	Constraint
base	Integer	> 0	Optional	-

Table 57: Attributes of element `<screw/>`

- `base`: the index (see section 5.3.1.1) of the flange partner, which is carrying the thread. If attribute is missing, the threaded part has to be derived from connection direction.

Specific subtypes of screws are defined by adding according nested elements, listed in following table:

Nested Elements	Multiplicity	Status	Constraint
flow_drilled	> 0	Optional	-

Table 58: Nested elements of element `<screw/>`

The subtypes are described in detail in sections below.

Example A (`screw` with no attributes):

```
<connection_0d label="100532" >
  <threaded_connection length="50." diameter="10"
    head_diameter="16." head_height="3" sink_size="4">
    <normal_direction x="3.0" y="0.0" z="0.0" />
    <!-- magnitude is irrelevant, direction sense is from head to point -->
    <screw /> <!-- Screw may come without any attributes -->
    <washer outer_diameter="20"/>
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

Example B (`screw` with “base” attribute with `washer`):

```
<connection_0d label="135">
  <threaded_connection length="50" diameter="10"
    head_diameter="16" head_height="5" thread_length="35">
    <normal_direction x="0" y="0" z="-10" />
    <washer outer_diameter="20" inner_diameter="10.3"/> <!--Washer next to head-->
    <screw base="5" />
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

Example C (*screw* with no attributes without *washer*):

```
<connection_0d label="135">
  <threaded_connection length="50" diameter="10"
    head_diameter="16" head_height="5" sink_size="1" thread_length="35" >
    <normal_direction x="0" y="0" z="-10"/>
    <screw base="5" />
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

7.5.7.1 Flow Drilled Screws (FDS)

Flow drilled screws are applied by a process called “friction drilling”. For details, see e.g.

http://en.wikipedia.org/wiki/Friction_drilling

and

<http://www.unique-design.co.uk/flow-drilling/>

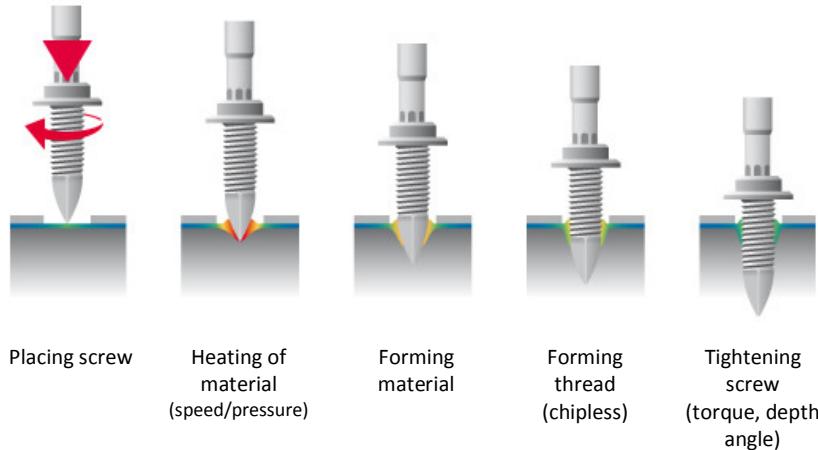
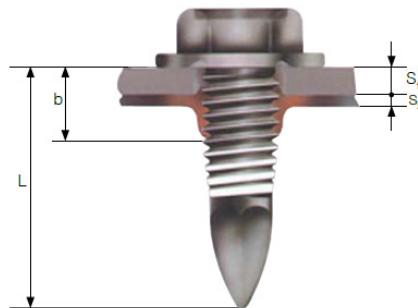


Figure 26: Process of Flow Drill Screwing



Source of image: http://www.ejot-avdel.se/sites/default/files/product/files/Brochure_EJOT_FDS_en.pdf

Figure 27: Measures of applied FDS

The application of such a connector element can be seen in the following video:

<https://www.youtube.com/watch?v=bnPBpN2y2FA>

The basic steps in the flow drill screw process consist of:

- a. Applying rotational velocity and pressure.

- b. Tool heats target sheet metal (or without pre-punching both sheet component) and melts through it.
- c. Screw thread tapping.
- d. Tightening the screw and applying proper torque to create the desired connection.

The FDS combines the tool with the screw: The screw itself drills its hole and shapes its thread.

Element “flow_drilled”

For the `<flow_drilled/>` element, the following attributes can be specified:

Attributes	Type	Value Space	Status	Constraint
pre_machined_hole_diameter	Floating point	≥ 0.0	Optional	-
pre_machined_hole_index	Integer	> 0	Optional	Exist only if <code><connected_to></code> properly filled out with parts to be connected.
pilot_hole_diameter	Floating point	≥ 0.0	Optional	Its definition depends on the applied FDS type.

Table 59: Attributes of element `<flow_drilled/>`

- ***pre_machined_hole_diameter***: In order to facilitate the penetration in the metal sheet of the tip of the Flow Drill Screw, a small hole may be machined in the sheet metal. Besides, when the penetration happens in the phase of material forming, a small portion of the formed part is flowing opposite to the fastening direction and creating a bulge (d_w) that has to be accommodated by the upper part’s clearance-hole (d_D). Default value is 0.0, which means “no pre-machined hole or clearance hole”.

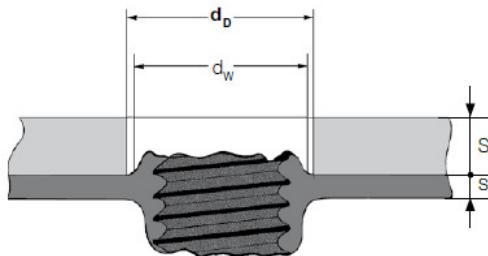


Figure 28: Pre-machined or clearance hole in FDS connection

- ***pre_machined_hole_index***: If $pre_machined_hole_diameter > 0.0$, then the hole is in the flange partner with index $pre_machined_hole_index$ (see section 5.3.1.1). If attribute is missing, this information is not (yet) available.
- ***pilot_hole_diameter***: This hole diameter (d_v) is defined in case of the applied FDS type requires a drilled hole on the sheet metal that is going to be formed during the process.

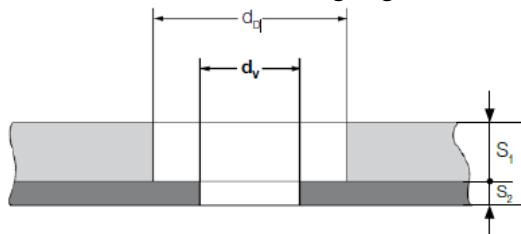


Figure 29: Pilot hole on sheet metal

The element `<flow_drilled/>` does not allow for any nested elements.

Example:

```

<connection_0d label="135">
    <threaded_connection length="50" diameter="10"
        head_diameter="16" head_height="5" sink_size="1" thread_length="35" >
        <normal_direction x="0" y="0" z="-10"/>
        <screw base="1">
            <flow_drilled pre_machined_hole_diameter="18.0"
                pre_machined_hole_index="1" pilot_hole_diameter="12.0" />
        </screw>
    </threaded_connection>
    <loc> 1500.3809 838.75885 730.6529 </loc>
    <appdata>
        ...
    </appdata>
</connection_0d>

```

7.6 Gum Drops

A gum drop, or adhesive point, is denoted by an element `<gumdrop/>`. This element is described completely by its attribute and nested elements.

Nested Elements	Multiplicity	Status	Constraint
gumdrop	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 60: Nested elements of `<connection_0d>` for `<gumdrop/>`

XML specification of `<gumdrop/>` with following attributes:

Attributes	Type	Value Space	Status	Constraint
diameter	Floating point	≥ 0.0	Optional	-
mass	Floating point	≥ 0.0	Optional	-
material	Alphanumeric	Alphanumeric	Optional	-

Table 61: Attributes of element `<gumdrop/>`

- **diameter**: The diameter of a gumdrop is specified by the attribute `diameter` for the child element of `<connection_0d>`. It specifies the diameter of the adhesive material *after* manufacturing.
- **mass**: This is the mass of the glue attached.
- **material**: the name of the adhesive material according to CAD/PDM. For CAE applications, another label from a reduced data base may be applicable. This is to be stored in `<appdata/>`.

Example:

```
<connection_0d label="SW_left_Gh_2123921">
  <!-- Assumed Unit system with mass attribute with value="kg" -->
  <gumdrop diameter="5.0" mass="0.0033" material="CAD_Material" />
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

7.7 Clinches

Clinching is a mechanical, cold forming fastening method to join sheet metal without additional components, using special tools to plastically form a mechanical interlock between the sheets.¹² In general, clinching is applied in case of lightweight metals because of they can be welded in poor quality or not at all. This joining technique can be cost-effective alternative to spot welding for specific steel structures. Such joints can be found on Air Conditioning Tube fixation or Air Bag Assemblies.

As result, the cross section of a clinch may look like this:

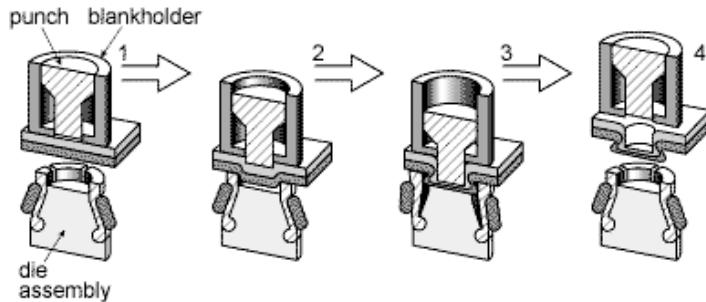


Figure 30: Schematic representation of the clinching operation

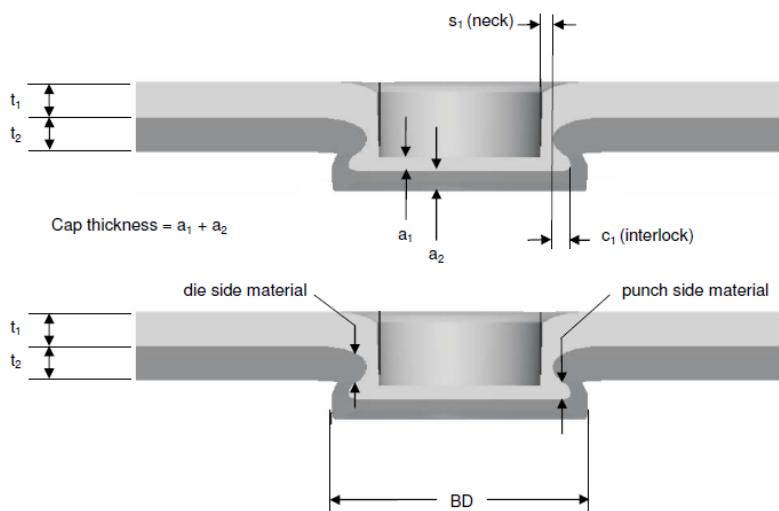


Figure 31: Clinch Joint Dimensions

¹² For details, see http://en.wikipedia.org/wiki/Clinching_%28metalworking%29

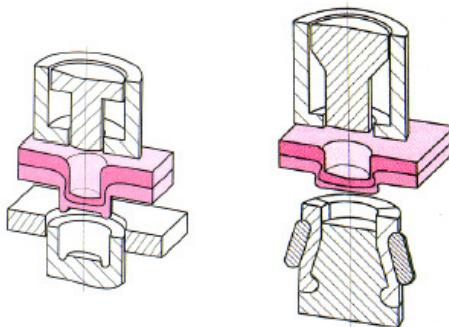


Figure 32: TOX (left) and BTM's Tog-L-Loc system¹³

One can imagine this cross section rotated around its vertical axis, giving a pan-shaped round clinch in 3 dimensions. Alternatively, this cross section could be the look at an open edge of two stacked sheets. The shape's height reduces, as we proceed "behind the paper", resulting in a wedge-shaped 3 dimensional contour.

Obviously, a wide range of geometrical shapes, produced by as many different tools, is possible. Hence, we cannot define an enumeration of all clinches, but must describe them by OEM specific alphanumeric names. Same is valid for the strength of the clinch, in terms of its strength class.

A clinch is denoted by an element `<clinch/>`. This element is described completely by its attributes and nested elements.

Nested Elements	Multiplicity	Status	Constraint
clinch	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 62: Nested elements of `<connection_0d/>` for `<clinch/>`

XML specification of `<clinch/>` element:

Attributes	Type	Value Space	Status	Constraint
clinch_type	Alphanumeric	Alphanumeric	Optional	-
strength_class	Alphanumeric	Alphanumeric	Optional	It is dependent from the applied punch diameter and part materials
shear_strength	Floating point	> 0.0	Optional	-
peel_strength	Floating point	> 0.0	Optional	-
button_diameter	Floating point	> 0.0	Optional	Dependent of punch diameter and sheet thicknesses
die_type	Alphanumeric	Alphanumeric	Optional	"round" or "rectangular"

Table 63: Attributes of element `<clinch/>`

- `clinch_type`: the alphanumeric name of the clinch. In this standard we will introduce and use for joint definition the 2 main systems which are TOX and BTM's Tog-L-Loc or

¹³ O.Hahn and A.Schulte, "Nutzung des festigkeitspotentials höherfesten stahlfeinbleche durch stanzniet- und clinchverbindungen", pp. 27-40, 1998

Lance-N-Loc¹⁴ system. The main difference is that the TOX system uses a fixed die whereas the BTM system employs an extending die (see Figure 32).

Remark: For further consideration the sub-process shape definitions can be introduced in this document that can be based on the data from:

<http://www.tox-uk.com/uk/products/joining-systems/tox-clinch-procedure.html>

- **strength_class:** the strength class name of the clinch. Due to the fact that the manufacturer of the applied clinching process has a specific tooling die diameter it can be defined the strength as 3 different classes. Such as:
 - Heavy Duty (HD) punches are 6.4mm/0.25" Ø and are used for thick material up to 0.35mm/0.135" thick. A HD joint is typically twice as strong as an equivalent MD joint.
 - Medium Duty (MD) punches are the most common and are approx. 4.6mm/0.18" Ø and are used for materials 0.20mm/0.075" to 0.025mm/0.010" thick.
 - Light Duty (LD) punches are 3.0mm/0.12" Ø and are used for thin materials up to 0.08mm/0.032" thick. LD joints are typically half as strong as a MD joint.
- **shear_strength:** Shear failure where the joint fails by shearing a hole in the punch side material. It is defined as maximum measured force during the test process.
- **peel_strength:** Pull failure in peeling test is where the joint pulls apart leaving a "male" and "female" parts. It is defined as maximum measured force during the test process.
- **punch_diameter:** The applied punch diameter to create this joint. The button diameter of the clinch seen in Figure 31 is shown by BD. The diameter may be defined as the maximum distance between the s-twists of the die side sheetFigure 31. According to its use, another definition may be more appropriate, though. As rule of thumb the following formula can be used: $D_{button} = d_{nom} \times 1.4$. Where d_{nom} is the punch diameter.
- **die_type:** The "round" dies (three and four blades) are used for drawable materials (like mild steel and aluminum). The "rectangular" dies (two blades) are used for hard materials (materials that do not draw very well) such as stainless steel.

If possible, a clinch should know the direction of fixation, i.e. possess a nested element `<normal_direction/>`. However, this is not mandatory in order to allow for importing incomplete data. Direction sense of `<normal_direction/>` is from punch to die, i.e. the direction in which metal is displaced. The element's definition can be found in section 7.1.3.

There is no "base" attribute for clinches, since this information can be derived from connection direction.

The element `<clinch/>` allows for following nested elements:

Nested Elements	Multiplicity	Status	Constraint
normal_direction	1	Optional	-

Table 64: Nested elements of element `<clinch/>`

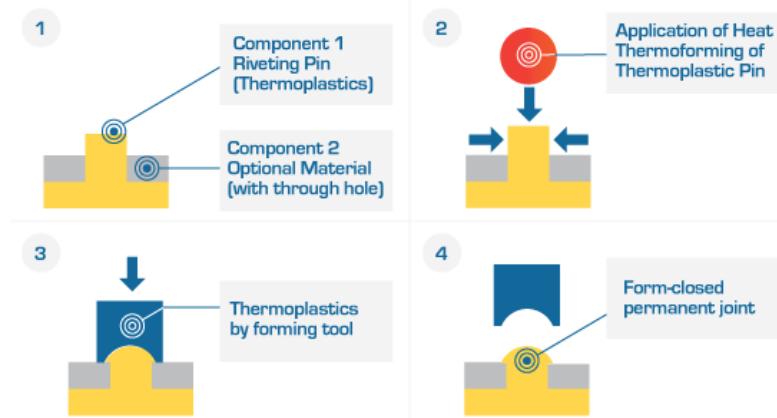
¹⁴ <http://www.btm-europe.de/en/tooling-system/lance-n-loc.html#how-it-works>

Example:

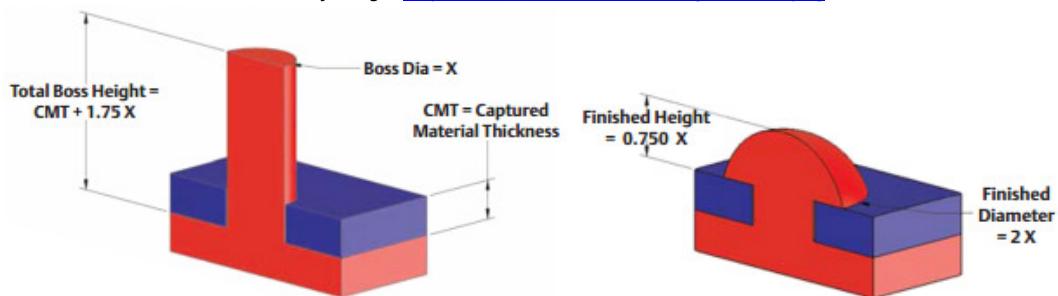
```
<connection_0d label="CLINCH_left_2123521">
    <!-- Unit definition and connected to is important for clinch -->
    <clinch clinch_type="TOX" button_diameter="3.0"
        strength_class="HD" shear_strength="890" peel_strength="356">
        <normal_direction x="0" y="0" z="-10"/>
    </clinch>
    <loc> 1645.83 821.145 616.585 </loc>
    <appdata>
        ...
    </appdata>
</connection_0d>
```

7.8 Heat Stakes / Thermal Stakes

Heat stakes are well known techniques to connect a shell-type part with a thermoplastic other part. For this reason, the thermoplastic part is manufactured with appropriate stakes.



Source of image: <http://www.bartec-dt.com/images/heat2.png>



Source of image:

<http://www.emersonindustrial.com/en-US/documentcenter/BransonUltrasonics/Plastic%20Joining/Non-Ultrasonics/Thermal%20Staking%20Design%20Guide%20pgs.pdf>

Figure 33: Cross Section of a Heat Stake

One can imagine this cross section rotated around its vertical axis, giving a round shape in 3 dimensions. This shape is most common, though not mandatory. Obviously, a wide range of geometrical shapes, produced by as many different tools, is possible.

Hence, we cannot define an enumeration of all heat stakes, but must describe them by OEM specific alphanumeric names (e.g.: flared, domed, knurled, hollow, flush etc.). Same is valid for the strength of the connection, in terms of its force-displacement diagram.

Heat stakes cannot be disassembled without irreversible damage to (at least) the thermoplastic part.

The element `<heat_stake/>` is described completely by its attributes and nested elements.

Nested Elements	Multiplicity	Status	Constraint
heat_stake	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 65: Nested elements of `<connection_0d/>` for `<heat_stake/>`

XML specification of `<heat_stake/>` element:

Attributes	Type	Value Space	Status	Constraint
heat_stake_type	Alphanumeric	Alphanumeric	Optional	-
strength	Floating point	> 0.0	Optional	-
diameter	Floating point	> 0.0	Optional	-
head_diameter	Floating point	> 0.0	Optional	diameter < hole_diameter
head_height	Floating point	≥ 0.0	Optional	-
void_diameter	Floating point	≥ 0.0	Optional	void_diameter < diameter
hole_diameter	Floating point	> 0.0	Optional	hole_diameter < head_diameter

Table 66: Attributes of element `<heat_stake/>`

- **heat_stake_type**: the alphanumeric name of the heat stake. (e.g.: domed, flared etc.)
- **strength**: the strength of the heat stake.
- **diameter**: The diameter of the heat stake, assuming a round/cylindrical shape.
- **head_diameter**: The diameter of the head of the heat stake after thermal forming, assuming the final shape is a round.
- **head_height**: the height of the head, created by the tool.
- **void_diameter**: The tool may form a hole/void within the stake. This is its diameter, assuming cylindrical shape.
- **hole_diameter**: Diameter of the hole(s) in the non-thermoplastic part(s).

If possible, a heat stake should know the direction of fixation, i. e. possess a nested element `<normal_direction/>`. However, this is not mandatory in order to allow for importing incomplete data. Direction sense of `<normal_direction/>` is from tool to thermoplastic part. The element's definition can be found in section 7.1.3.

There is no "base" attribute for heat stakes, since this information can be derived from connection direction.

The initial height of the stake (above base part) is not represented in xMCF: Before tool application, it can be derived from CAD data. After tool application (in final shape of the heat stake), this height has vanished.

Example:

```
<connection_0d label="HEAT_STAKE_521">
    <heat_stake heat_stake_type="domed" diameter="3.0"
        head_diameter="6.0" head_height="2.25">
        <normal_direction x="0" y="0" z="-10"/>
    </heat_stake>
    <loc> 1645.83 821.145 616.585 </loc>
    <appdata>
        ...
    </appdata>
</connection_0d>
```

7.9 Clips/Snap Joints

In general, a clip is a fastener with an elastic component. Pushed onto a firm counterpart, this elastic component causes the clip to hook onto that part. It depends on the type of the clip, whether it can be removed without being destroyed.

A wide and ever increasing variety of clinches is in practical use.

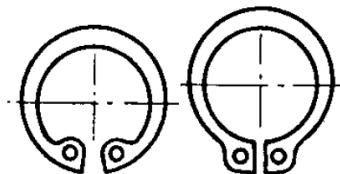
Examples:

- A so-called “Terry Clip” consists of a cylindrical metal band with a gap. Opening the gap, it snaps onto a tube. Frequently, there are means for fastening a screw etc. on the opposite side of the gap.
- A “Hairpin Clip” is similar to a “Terry Clip”, but uses some wire instead of a metal band.
- An “R-Clip” is similar to a “Hairpin Clip”. One of its legs is straight and suitable for inserting into a drilled hole of an axle.
- Circlips (also known as a C-Clip, Seeger ring, snap ring or Jesus clip) are used to secure some item against sliding on an axle.
- Another sort of clips is snapped into a hole in a sheet metal. Its other side is shaped to hold a certain item, such as a cable, a panel etc.
- Other clips slide onto a flat surface.



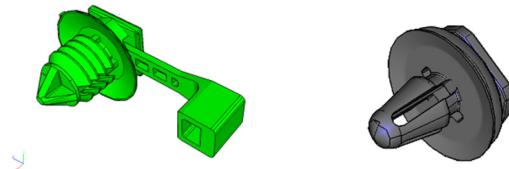
Source of image: http://en.wikipedia.org/wiki/File:Hairpin_clip.png

Figure 34: A "Hairpin Clip"



Source of image: http://commons.wikimedia.org/wiki/File:Circlips_interieur.png

Figure 35: Internal and External Circlips



Source of images: Ford Werke GmbH

Figure 36: Clips Pushed into a Hole



Source of images: Ford Werke GmbH

Figure 37: Clips Sliding onto a Flat Surface

A clip is denoted by an element `<clip/>` and described completely by its attributes and nested elements.

Nested Elements	Multiplicity	Status	Constraint
clip	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 67: Nested elements of `<connection_0d/>` for `<clip/>`

XML specification of `<clip/>` element:

Attributes	Type	Value Space	Status	Constraint
clip_type	Alphanumeric	Alphanumeric	Optional	-
attachment_type	Alphanumeric	Alphanumeric	Optional	-
hole_diameter	Floating point	≥ 0.0	Optional	-
hole_length	Floating point	≥ 0.0	Optional	hole_length > 0 implies hole_diameter > 0
pin_diameter	Floating point	≥ 0.0	Optional	-
pin_width	Floating point	≥ 0.0	Optional	pin_width > 0 implies pin_diameter > 0
pin_length	Floating point	≥ 0.0	Optional	-
strap_length	Floating point	≥ 0.0	Optional	-
clipped_to	Integer	> 0.0	Optional	-
material	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	-

Table 68: Attributes of element `<clip/>`

- **clip_type**: the alphanumeric name of the clip, e.g. at FORD: "STRAP 5-45X8X.9-4.1 PNL".
- **attachment_type**: the description, how the clip is fastened, e.g. "push into round hole".

- **hole_diameter**: If the clip is pushed into a hole, this attribute describes the diameter of that mating hole. If the hole is not round, the minimum diameter is meant. Default value is 0.0, which means “no hole”.
- **hole_length**: If the clip is pushed into a *non-round* hole, this attribute describes the maximum diameter of that hole. Default value is 0.0, which means “no hole or round hole”.
- **pin_diameter**: If the clip is pushed into a hole, this attribute describes the diameter of the clip’s pin. If the hole is not round, the minimum diameter is meant. Default value is 0.0, which means “no hole”.
- **pin_width**: If the clip is pushed into a *non-round* hole, this attribute describes the maximum diameter of the clip’s pin. Default value is 0.0, which means “no hole or round hole”.
- **pin_length**: If the clip is pushed into a hole, this attribute describes the length of the clip’s pin. Default value is 0.0, which means “no hole”.
- **strap_length**: If the clip carries a strap (cf. to Figure 15: Clips Pushed into a Hole, picture a.), this attribute describes the length of that strap. Default value is 0.0, which means “no strap”.
- **clipped_to**: The clip is clipped to the flange partner with this index (see section 5.3.1.1). If attribute is missing, this information is not (yet) available.
- **material**: the material of the clip.
- **part_code**: the part code of the clip, as used e. g. in a PDM system.

There is no **base** attribute for clips, since this information is hold by attribute **clipped_to**.

If possible, a clip should know the direction of fixation, i. e. possess a nested element **<normal_direction/>**. However, this is not mandatory in order to allow for importing incomplete data. Direction sense of **<normal_direction/>** is from tool to the flange partner given by attribute **clipped_to**.

Element **<tangential_direction/>** denotes direction of (one) maximum clip diameter, perpendicular to **<normal_direction/>**. This gives the local x axis. The **<normal_direction/>** and **<tangential_direction/>** elements are described in section 7.1.3.

The element **<clip/>** allows for following nested elements:

Nested Elements	Multiplicity	Status	Constraint
normal_direction	1	Optional	-
tangential_direction	1	Optional	-

Table 69: Nested elements of element **<clip/>**

Example:

```

<connection_0d label="CLIP_1001">
  <clip clipped_to="1" attachement_type="push into round hole" hole_diameter="8.0"
        hole_length="12.0" pin_diameter="10.0" pin_length="10.0" material="polyamid">
    <normal_direction x="0" y="0" z="-10"/>
    <tangential_direction x="0" y="10" z="0"/>
  </clip>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>

```

7.10 Nails

Nailing is a rather old joining method. However, with optimized nail shapes and high velocity application, it still addresses modern requirements, especially if non-steel materials are involved.



Source of image: <http://www.boellhoff.de/files/jpg2/RIVTAC-Alu-Hybrid-low.jpg>

Figure 38: RIVTAC® Nail

The components, which are connected by this type of connector, may consist of steel, aluminum, magnesium or plastic.



Source of image: <http://www.boellhoff.de>

Figure 39: Cross Section of a Nail, Connecting Two Sheets

A nail is denoted by an element `<nail/>`. This element is described completely by its attributes and nested elements.

Nested Elements	Multiplicity	Status	Constraint / Remarks
nail	0-1	Optional	-
loc	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

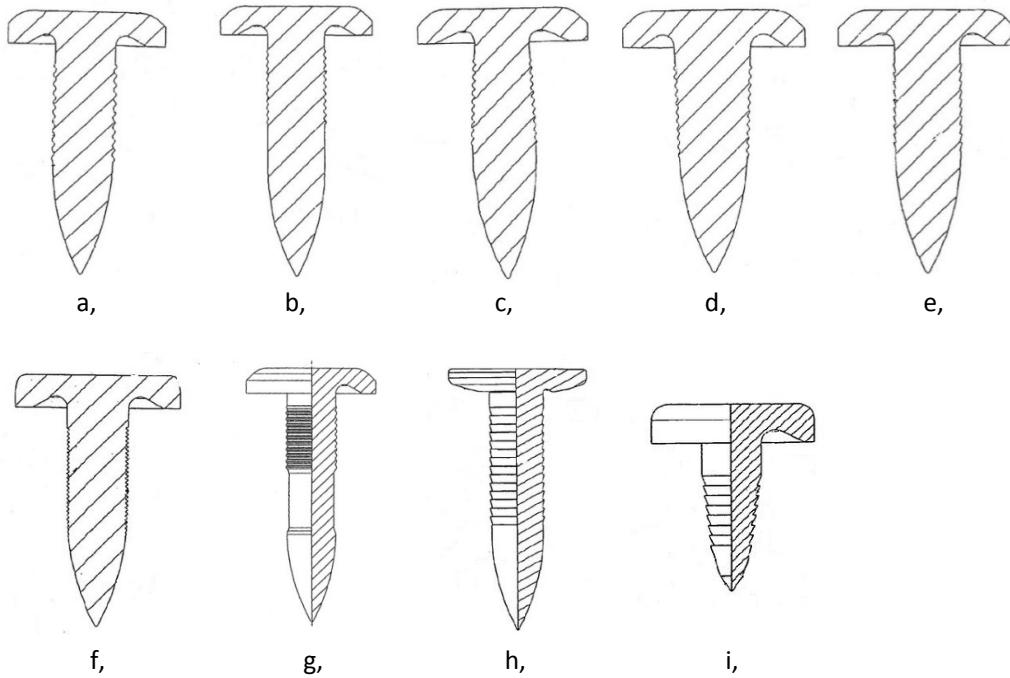
Table 70: Nested elements of `<connection_0d/>` for `<nail/>`

XML specification of `<nail/>` element:

Attributes	Type	Value Space	Status	Constraint / Remarks
nail_type	Alphanumeric	Alphanumeric	Optional	-
shaft_diameter	Floating point	> 0.0	Optional	-
length	Floating point	> 0.0	Optional	-
cylinder_length	Floating point	> 0.0	Optional	-
head_diameter	Floating point	> 0.0	Optional	-
head_height	Floating point	> 0.0	Optional	-
shear_strength	Floating point	> 0.0	Optional	Dependency from sheet thicknesses
peel_strength	Floating point	> 0.0	Optional	Dependency from sheet thicknesses
material	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	-

Table 71: Attributes of element `<nail/>`

- **`nail_type`**: the alphanumeric name of the nail¹⁵. (Naming convention based on supplier nail codes)



- **`shaft_diameter`**: the diameter of the shaft of the (unmounted) nail.
- **`length`**: is the overall length of the nail.
- **`cylinder_length`**: the length of the cylindrical part of the nail shaft.
- **`head_diameter`**: the diameter of the head of the nail.
- **`head_height`**: the height of the nail head.

¹⁵ <http://www.google.com/patents/EP1926918B1?cl=en>

- ***shear_strength***: Shear failure where the joint fails by shearing a hole in the cover part side material. It is defined as maximum measured force during the test process.
- ***peel_strength***: Pull failure in peeling test is where the joint i.e. nail and cover sheet pull apart leaving the base sheet part. It is defined as maximum measured force during the test process.
- ***material***: the material of the nail.
- ***part_code***: the part code of the nail, as used e. g. in a PDM system. Frequently, it may be convenient to use the nail norm (according to ISO, EN, BSW, DIN etc.) as part code.

There is no "base" attribute for nails, since this information can be derived from connection direction.

If possible, a `<nail/>` should know the direction of fixation, i.e. possess a nested element `<normal_direction/>`. However, this is not mandatory in order to allow for importing incomplete data. Direction sense of `<normal_direction/>` is from nail head to tip. The element's definition can be found in section 7.1.3.

The element `<nail/>` allows for following nested elements:

<i>Nested Elements</i>	<i>Multiplicity</i>	<i>Status</i>	<i>Constraint</i>
normal_direction	1	Optional	-
tangential_direction	1	Optional	-

Table 72: Nested elements of element `<nail/>`

Example:

```

<connection_0d label="NAIL_100">
  <nail shaft_diameter="10.0" length="26.0" head_diameter="15.0" material="steel"
    shear_strength="5.20" peel_strength="5.0"> <!-- unit def. for Force is kN -->
    <normal_direction x="0" y="0" z="-10"/>
  </nail>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>

```

8 1D connections

8.1 Generic Definitions

8.1.1 Identification

For identifying 1D connections, the same rules apply as for 0D connections, see section 7.1.1 identification.

8.1.2 Location

The definition of the connection line is described as a series of points and thus split into segments. All other curves can also be represented with this type of representation by adding necessary points and thus approximating to the needed accuracy.

The connection line may consist of more than one section. The sections need not to be joined to each other. This is to simulate gaps along the application of a seam or an adhesive, due to crossing another weld, or an obstacle, like a hole in the connected sheets.

This feature could also be used in cases where a seam weld changes its `<subtype/>`, seen in Figure 40: Weld Line Changing from Y-Joint to Overlap-Joint.

The χ MCF specifies the order of line sections, as well as the order of the locations within each section.

Element “loc_list”

The list of locations for the definition of the connection line is stored in the element `<loc_list>`. This element contains nested elements `<loc/>` defining the location of a point of the connection line in space. These locations have to be ordered so that the line defined by the ordered list of locations specifies the connection line.

No additional attributes are associated to the element `<loc_list>`.

The `<loc_list>` element has the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
loc	1-*	Required	-

Table 73: Nested elements of `<loc_list>`

Element “loc”

Each location specified by the element `<loc/>` contains three values specifying the x, y and z coordinates of the location.

The attributes associated to the element `<loc/>` are:

Attributes	Type	Status	Constraint
v	Numeric	Required	-

Table 74: Attributes of element `<loc/>`

The attribute `v` is used as surrogate index to ensure proper ordering. The values are NOT related to the attribute `u` used in the `<weld_position/>` element.

The `<loc/>` with the minimum value of "v" marks the start of a seam weld and `max(v)` is used to mark the end. The reason for that is some manufacturing techniques are not "symmetric" regarding both ends of a connection line.

Example A (a connection line with a single section):

```
<loc_list>
    <loc v="0" > 2581.21 -708.408 31.6532 </loc> <!-- first point -->
    <loc v="0.1" > 2581.42 -708.357 35.2816 </loc>
    <loc v="2.22"> 2581.05 -708.302 39.0643 </loc> <!-- last point -->
</loc_list>
```

Example B (a connection line consisting of two disjoint sections):

```
<loc_list index="1"> <!-- first section -->
    <loc v="0" > 2581.21 -708.408 31.6532 </loc> <!-- first point -->
    <loc v="1" > 2581.42 -708.357 35.2816 </loc>
    <loc v="2.22"> 2581.05 -708.302 39.0643 </loc> <!-- last point -->
<loc_list index="2"> <!-- second section -->
    <loc v="1" > 2581.05 -708.302 40.3340 </loc> <!-- first point -->
    <loc v="2.1"> 2581.05 -708.302 48.5300 </loc> <!-- last point -->
</loc_list>
```

8.1.3 Type Specification

Each connection is identified by its type. The XML definitions of all 1D connections are containing the following elements:

Nested Elements	Multiplicity	Status	Constraint / Remarks
seamweld	1	Optional	-
adhesive_line	1	Optional	-
hemming	1	Optional	-
sequence_connection_0d	1	Optional	-
contact_list	1	Optional	See section 5.3.2.5.

Table 75: Nested elements of element `<connection_1d/>`

Note: Only one of the elements (`seamweld`, `adhesive_line`, `hemming`, `sequence_connection_0d`) must exist in a `<connection_1d>`. If none of the type elements exist, then this will default to `<seamweld/>`.

8.2 Seam Welds

8.2.1 Description and Modeling Parameters

To be able to use the xMCF file as a description for seam welds in the process it is necessary to use the modeling described in this document.

The description of seam welds made up from different modeling types is handled in the way that these welds are split up into separate seam welds each of them containing the specific information representing the intended modeling.

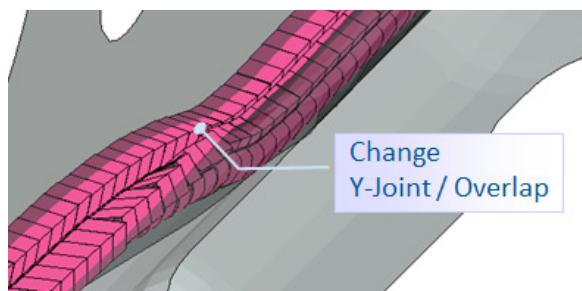
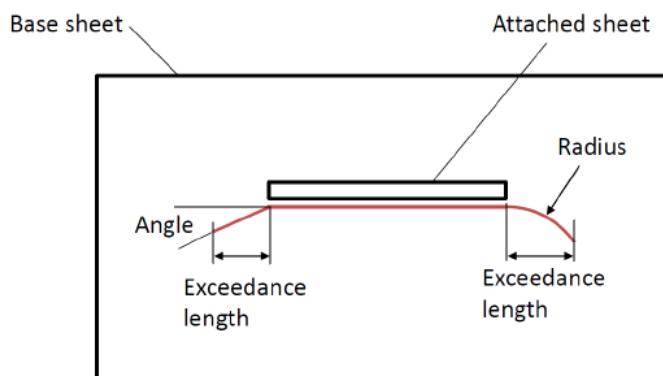


Figure 40: Weld Line Changing from Y-Joint to Overlap-Joint

This assures that a seam weld definition only represents one cross section with the welding parameters for all the welded sides.

There is a demand for handling weld lines exceeding the actual contact polygon of the involved parts, which is presented in the following figure:



Source of image: Dr. Thomas Bruder, BMW Group.

Figure 41: Longitudinal stiffener, top view

Standard conform polygons may well exceed the contact area. However, xMCF version 3.0 does not state anything about the physical meaning or the implications for CAE and CAM. Hence, later versions of xMCF may specify details about what should happen with exceeding parts of a weld line in CAE and CAM. In CAE, for example, hexahedron or tetrahedron could be generated on the exceeding polygons, if their height or thickness is provided.

Remark: It is well known that several welding technologies produce material structures which are oriented. In especially, there is a difference between the start and the end of a weld line. xMCF knows about the orientation of a weld line and hence can distinguish between start and end. But it does not yet provide means to transport details about what is the difference between both, neither for CAE nor CAM.

8.2.2 Seam Weld Definition Overview

The weld definition depends on the type of the weld. For each of the different types the parameters and their meaning can be different. The detailed description can be found in the next sections describing each weld type separately.

The table shown below provides an overview over the current seam weld types and their parameters.

For each of the weld types it provides the following information:

- Type of the weld
- Number of weld positions for the type
- Supported technology

- Valid weld sections
- Required parameters
- Optional parameters with their default values
- Section drawing related to the weld type

For the given combinations of weld type, technology and section the parameters and the section drawings are provided. The section drawings do not show the specific sections possible for a technology.

The sheet parameters describing the sheet thickness in the following document sections are not part of the χ MCF file contents. They are used in the weld specific sections to describe parameters stored in the χ MCF file and their relations.

The variety is to be handled by the application using the χ MCF file inside the process. All the information stored for the weld together with the model is sufficient to determine the specific type of connection.

Weld Type	# of Weld Positions	Welding Technology	Section	Weld Parameter			Layout
				Required	Optional	Fixed	
<i>Butt Joint</i>	1	Arc	I	w idth	-	-	
	1		V	w idth	-	-	
	1		U	w idth	-	-	
	2		X	w idth	-	-	
	1		Y	w idth	-	-	
	1		Radius	w idth	-	-	
<i>Corner Weld</i>	1-2	Arc	Fillet	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	1-2		HV	thickness i.e. throat	gap=0 angle=45	penetration=1	
	1-2		U	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
<i>Double Corner Weld</i>	2-4	Arc	Fillet	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	2-4		HV	thickness i.e. throat	gap=0 angle=45	penetration=1	
	2-4		U	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
<i>Edge Weld</i>	1	Arc	I	w idth	gap=0	-	
	1		V	w idth	gap=0	-	
	1		U	w idth	gap=0	-	
<i>I-Weld</i>	1	Laser	-	w idth	gap=0	-	
	1	Arc	U	w idth	gap=0	-	
<i>Overlap Weld</i>	1	Arc	-	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	2	Arc	-	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	2	Arc	-	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
<i>Y-Joint</i>	1-2	Arc	Fillet	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	1-2		HV	thickness i.e. throat	gap=0 angle=45	penetration=1	
	1-2		HY	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
<i>K-Joint</i>	2-3	Arc	Fillet	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	2-3		HV	thickness i.e. throat	gap=0 angle=45	penetration=1	
	2-3		HY	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
<i>Cruciform-Joint</i>	2-4	Arc	Fillet	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
	2-4		HV	thickness i.e. throat	gap=0 angle=45	penetration=1	
	2-4		HY	thickness i.e. throat	penetration=0 gap=0 angle=45	-	
<i>Flared Joint</i>	1	Arc	-	w idth	gap=0	-	

Figure 42: Seam weld types and attributes

8.2.3 Specific XML Realization

This part of the XML structure describes the data stored for each of the seam welds. This includes the details necessary to describe each connection in depth.

Inside the XML definition of the seam weld each of the welds related to a connection is stored in a separate weld position inside the specific subtype definition.

```

- <connection_list>
  - <connection_1d label="1000032">
    - <loc_list>
      <doc v="0">68 0 0</loc>
      <doc v="1">88 0 0</loc>
    </loc_list>
    - <seamweld>
      - <butt_joint base="1" technology="resistance" section="Y" filler="yes">
        <weld_position u="1" x="0" y="6.12323e-17" z="1" width="3" />
      </butt_joint>
    </seamweld>
    - <appdata>
      - <MEDINA xmlns="http://servicenet.t-systems.com/medina/xMCF">
        - <data_at_connector>
          - <original_loc_list>
            <doc u="0">68 4 10</loc>
            <doc u="1">88 4 10</loc>
          </original_loc_list>
          <connection_data adjust_limit="1" weld_position_id="1" max_projection_distance="10" />
          <administrative_data element_label="Weldline_Overlap-Join" part_tree_position="fixed" connector_property_id="1000032" />
        </data_at_connector>
      </MEDINA>
    </appdata>
  </connection_1d>
</connection_list>
```

Figure 43: xMCF Structure of a Seam Weld (*connection_1d*)

8.2.4 Generic Seam Weld Definition

8.2.4.1 Identification

Each seam weld is optionally identified by its *label*. The XML definition at *connection_1d* level contains the following attributes:

Attributes	Type	Status	Constraint / Remarks
label	Alphanumeric	Optional	-
quality_control	Alphanumeric	Optional	See section 6.4 Attribute <i>quality_control</i>

Table 76: Attributes of element *<connection_1d>*

Attribute “label”

The label defines the human readable identification of the seam weld connection.

Example:

```

<connection_list>
  <connection_1d label="MyWeldLine">
    <loc_list>
      ...
    </loc_list>
    <seamweld>
      ...
    </seamweld>
    <appdata>
      ...
    </appdata>
  </connection_1d>
</connection_list>
```

8.2.4.2 Type Specification

Each seam weld is identified by main type of the weld and described more precisely by its subtype. This means there is a general category that includes several subcases. Detailed information can be seen under definition of element main type and subtype.

Definition of main type

The element main type for seam welding always has the value `seamweld`. This is located directly below the `<connection_1d/>` element. It is used to define the connection as general as it can be.

The XML definition of seam weld main type contains the following nested elements:

<i>Nested Elements</i>	<i>Multiplicity</i>	<i>Status</i>	<i>Constraint</i>
butt_joint	1	Optional	-
corner_weld	1	Optional	-
edge_weld	1	Optional	-
i_weld	1	Optional	-
overlap_weld	1	Optional	-
y_joint	1	Optional	-
k_joint	1	Optional	-
cruciform_joint	1	Optional	-
flared_joint	1	Optional	-

Table 77: Nested elements of element `<seamweld/>`

Example A (main type as `seamweld`):

```
<connection_1d>
  <seamweld>
    ...
  </seamweld>
</connection_1d>
```

Note: The differentiator for the specific seam weld is stored as value in the subtype element which is described below.

Definition of subtype

Different kinds of welds are distinguished through the definition of a subtype of the seam weld.

Valid values for the subtype element are:

- `butt_joint`
- `corner_weld`
- `edge_weld`
- `i_weld`
- `overlap_weld`
- `y_joint`
- `k_joint`
- `cruciform_joint`
- `flared_joint`

Each subtype element can contain the following attributes:

Attributes	Type	Value Space	Status	Constraint
base	Integer	> 0	Optional	-
technology	Selection	resistance arc laser friction brazing	Optional	-

Table 78: Attributes of element <subtype/>

Each subtype element contains the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
weld_position	1 - *	Optional	-
sheet_parameter	1 - *	Optional	-

Table 79: Nested elements of element <subtype/>

Note: The number of elements of <*weld_position*> is dependent on the specific subtype.

Attribute “base”

The attribute *base* defines the index of the base sheet for the weld. It references the attribute *index* inside the element *<part>* of the *<connected_to>* element. This could be useful when the angle of the weld itself is not symmetrical between the welded sheet and the base sheet. That means it is crucial to be identified to which sheet part the angle is being measured.

Attribute “technology”

The technology used to weld the connection can be specified for each of the welds of a connection separately.

This technology can be one of

- Resistance welding
- Arc welding
- Energy beam welding (e.g. laser)
- Friction welding
- Brazing (Not allowed for I-Welds, for technical reasons.)

Additionally to the technology there is a specification for each of the weld positions whether the welding introduces additional material (attribute *filler*).

The attribute *technology* defines the welding technology used for its subtype.

Possible values are:

- *resistance*
- *arc*
- *laser* (Energy beam / Laser)
- *friction*
- *brazing*

Example A (main type as `seamweld` and subtype as `butt_joint`):

```
<connection_1d>
  <seamweld>
    <butt_joint base="1" technology="resistance">
      ...
      <weld_position ... />
      <sheet_parameter ... />
      ...
    </butt_joint>
  </seamweld>
</connection_1d>
```

8.2.4.3 Weld Position and Sheet Metal Parameters

We have to collect and put into separate groups the parameters that can be observed in terms of welding processes. Some of the used and measured parameters are related to the involved sheet metal parts, describing the thickness of the sheet and the applied sheet angle between two sheet metal parts.

On the other hand, we can distinguish the parameters that are mentioned in terms of the welding process has been made i.e. the weld itself is present. The detailed description of these parameters can be seen for Sheet Parameters in chapter 8.2.4.3.1 and for Weld Position Parameters in chapter 8.2.4.3.2.

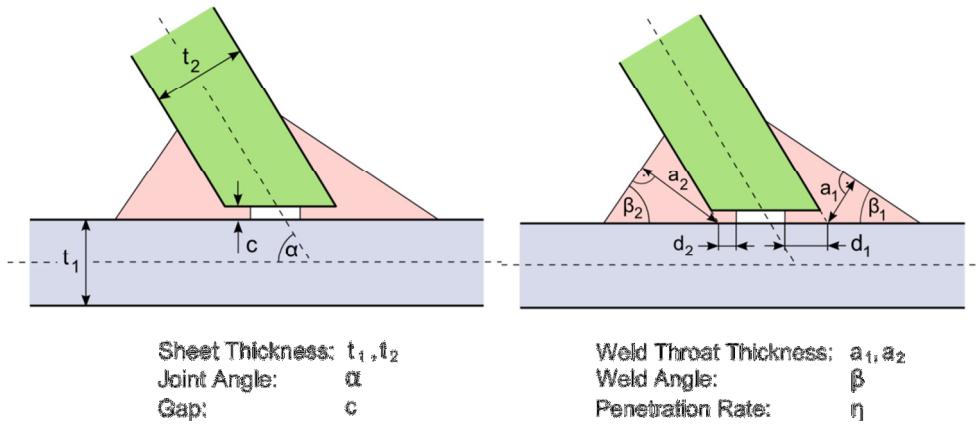


Figure 44: Sheet Parameters vs. Weld Position Parameters

8.2.4.3.1 Parameters Assigned to a Specific Sheet of the Flange

In a welded connection there are different kinds of parameters that have to be assigned to either welded sheet metal or the created weld itself. Thus we can group and put all those parameters under two elements directly under the parent subtype element. These are the `<sheet_parameter/>` and the `<weld_position/>`.

Element "sheet_parameter"

The element `<sheet_parameter/>` describes the sheet in order to identify the correct sheet when multiple sheets are connected. Furthermore it defines as attributes the corresponding gap applied between the welded sheet and the base sheet, i.e. in general the applied gap between the welded sheets involved in the welding process.

It is defined using the following attributes:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to < <i>part</i> > index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 80: Attributes of element <*sheet_parameter*>

Attribute “index”

The value of the attribute *index* must be referenced to the Part index. The *index* needs to be unique only within the parent element <*connected_to*>. For specific connections, it is used as the matching index for the subjected welded sheet.

Attribute “gap”

The value of the attribute *gap* is numerical in the range $(0, \infty)$. It defines the distance between the base and the connected sheet.

Attribute “sheet_thickness”

The value of the attribute *sheet_thickness* is numerical in the range $(0, \infty)$. It defines the CAD related input for the thickness measure of the connected sheet (in the example in Figure 44 this is t_2). In case of more than 1 welded sheet exist see the definition example in 8.2.11.5.

Attribute “sheet_angle”

The value of the attribute *sheet_angle* is numerical in the range $[0, 360]$. It defines the angle between the base sheet and the connected sheet middle lines.

Example A (within each *attribute*):

```

<connection_1d>
  <seamweld>
    <corner_weld base="1" technology="resistance">
      <weld_position .../>
      <sheet_parameter index="2" gap="1.0" sheet_thickness="1.5" sheet_angle="90">
    </corner_weld>
  </seamweld>
</connection_1d>

```

8.2.4.3.2 Welding Position

The position of the welding on the seam weld is specified by an orientation vector pointing from the weld root into the side where the welding takes place (see Figure 45).

The origin of this orientation vector is located directly on the connection line. The position on the connection line is determined by a fraction in the range $[0, 1]$ of the complete line. The fraction is applied to the length of the connection line measured as sum of all segment lengths in space.

A connection can be welded at different positions. This is depending on the seam weld type and can be between two and five positions (by combining K-Joint with a Y-Joint). Each position represents a welding performed from one side of the structure.

Details for each seam weld type are described inside the specific chapter (e.g. see 8.2.5).

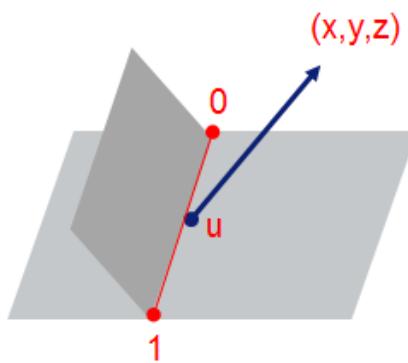


Figure 45: Welding Position of a Y-Joint

Primary and Secondary Sides

For weld definitions needing a specific side the orientation vector defines the primary side. All other sides are named secondary side not specifying any precedence on them.

Element “*weld_position*”

The element `<weld_position/>` describes the location of the weld relative to the connection line specified in `loc_list`. It is defined using the following attributes:

Attributes	Type	Status	Constraint / Remarks
u	Numeric	Required	$0 \leq u \leq 1$
x	Numeric	Required	-
y	Numeric	Required	-
z	Numeric	Required	-
reference	Boolean	Optional	“false”
section	Selection	Optional	-
thickness	Numeric	Optional	Value only for specific weld types
width	Numeric	Optional	Value only for specific weld types
angle	Numeric	Optional	-
filler	Selection	Optional	-
filler_material	Alphanumeric	Optional	-
shape	Selection	Optional	-
penetration	Numeric	Optional	$0 \leq \text{penetration} \leq 1$

Table 81: Attributes of element `<weld_position/>`

Depending on subtype the attributes of the element `<weld_position/>` are different. Each of the subtype is supporting its specific combination of attributes. Description of the specific combination can be found in the specific weld section below.

Example A (within each *attribute*):

```
<connection_1d>
  <seamweld>
    <corner_weld base="1" technology="resistance">
      <weld_position u="0.2" x="1" y="0" z="1"
                    reference="true"
```

```

        section="HY"
        thickness="0.5"
        width="3.0"
        angle="45"
        filler="yes"
        filler_material=" E7018-X"
        shape="straight"
        penetration="0.6"/>
    <sheet_parameter index="2" gap="1.0"/>
</corner_weld>
</seamweld>
</connection_1d>

```

Attributes "u", "x", "y", "z"

The attribute *u* specifies the relative location on the connection line defined in *loc_list*. Value *u=0* represents the first location of the connection line matching the element *loc* specified with the lowest value for the attribute *u*. Value *u=1* represents the last location of this line matching the element *loc* with highest value for the attribute value *u*. Values in between are specifying the point located at the specified fraction of the line measure in summed up lengths of the segments of the connection line in space.

The attributes *x*, *y*, *z* are specifying the direction vector in global coordinate system into the quadrant of the welding. The origin of this vector is defined by *u* and the *loc_list*.

The length of the vector has no specific meaning, only the direction is used. However, it should be sufficiently long to be unambiguous as it is presented in Figure 46.

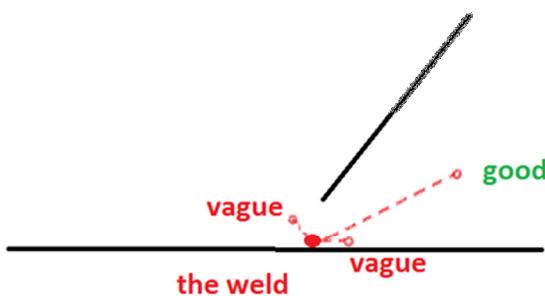


Figure 46: Welding Position vector direction and length

Attribute "reference"

The attribute *reference* specifies this weld position to be the reference for welds that need such a reference. In case of butt-welds or cruciform joints this is needed to specify a specific side for one of the attributes (see there).

Attribute "section"

The attribute *section* defines the geometry section of the weld. The different section types that can be used inside the definition of seam welds are listed here. The description here denotes the principles of the sections. Details of the interpretation on the different weld type can be found in the corresponding section for each of the weld types.

In most cases the sections "Fillet", "HV" and "HY" are used in seam weld connections when the head of a sheet is welded on a base sheet. Connections putting two sheet heads together are mostly using the section types "I", "V", "X" and "Y".

Allowed values are:

- *I*
- *V*
- *U*
- *X*
- *Y*
- *HV*
- *HY*
- *Fillet*
- *Radius*

Section "I"

The section "I" describes the filling of the weld normally on the head sides of a connection. The section is filled completely and may be welded from one or two sides.

Section "V"

The section "V" describes the one-sided filling of the weld with welding material looking like a "V". The weld filling provides full penetration.

Section "U"

The section "U" describes the one-sided filling of the weld with welding material looking like a "U". The penetration in most cases is less than full penetration.

Section "X"

The section "X" describes the filling of a two-side weld with welding material looking like an "X". The weld filling provides full penetration.

Section "Y"

The section "Y" describes the one-sided filling of the weld with welding material looking like a "v". Only a part of the gap between the welded sheets is filled thus there is no full penetration.

Section "HV"

The section "HV" describes the filling of a one-sided weld with a full penetration. The welded sheet is normally be phased to take full advantage of the full penetration.

Section "HY"

The section "HY" describes again a filling of a one-side weld but the penetration is only partial. In common cases the welded sheet is phased partially to take again advantage of the penetration at that area.

Section "Fillet"

The section "Fillet" describes a one-sided welding placed on the outside of the welded sheets. Depending on the sheet thicknesses there might be a penetration.

Section "Radius"

The section "Radius" describes a special case where the welding material looks like a circle but not filling the complete gap between the welded sheets. In most cases there is no full penetration.

Section "Laser"

The section "Laser" describes the filling of a laser welding affected zone. The result can be a complete weld-through on one sheet or the filling of a gap due to laser heating influence.

Attribute "thickness"

The value of the attribute `thickness` is a numerical value in the range of $(0, \infty)$. It describes the distance between the weld root and the weld surface. It is used for to describe the throat thickness of the weld.

Attribute "width"

The value of the attribute `width` is a numerical value in the range of $(0, \infty)$.

Attribute "angle"

The value of the attribute `angle` is a numerical value. This attribute of the `<weld_position/>` element describes the angle between the weld face and the base sheet face.

Attribute "filler"

The attribute `filler` specifies whether the welding is performed using filling material. This is the case for resistance or arc welding but not for laser welding.

The allowed values are:

- `yes`
- `no`

According to above rule on filling material, the default values are depending on the attribute value of `technology` of the element subtype:

Attribute value "technology"	Default value "filler"
resistant	yes
arc	yes
laser	no

Table 82: Default values of attribute "filler", dependent from attribute "technology"

Attribute "filler_material"

The attribute `filler_material` specifies the applied material during the welding process.

Attribute "shape"

The attribute `shape` defines the shape of the weld throat. Allowed values are:

- `straight`
- `convex`
- `concave`

Independent of the shape, the weld position attributes (a-measure, weld angle etc.) are taken with respect to the `straight` line. In fact, the shape is just a hint to specific solvers. It does *not* provide an exact definition whether convex or concave mean e.g. "segment of a circle", "parabolic" etc., nor how big the deviation from straight shape is.

Attribute "penetration"

The value of the attribute `penetration` is a numerical value in the range $[0; 1]$. The value describes the ratio between the thickness and the penetration of the sheets. Value of 0 means no penetration, value of 1 represents complete penetration.

Note: The attribute `penetration` of a `<weld_position/>` holds for all sheets connected by this `<weld_position/>` (e.g. important for K-joints).

Note: If all `<weld_position/>` at the same welded sheet have a sum of penetration ≥ 1 there is no open (unfilled) gap between the base sheet and the welded sheet.

8.2.5 Butt Joint

The principles of the modeling of Butt Joints for xMCF are described in this section. A Butt Joint describes a connection between two sheets welded at their forehead side.

The XML definition of a Butt Joint supports up to two weld positions. Each of the weld positions is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.5.1 Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1 Thickness of welded sheet

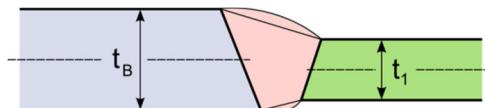


Figure 47: Butt Joint Sheet Layout

8.2.5.2 Weld Parameters

The parameters of the weld are described below:

- b_1 Width of the weld at primary side
- b_2 Width of the weld at secondary side
- e_1 Reinforcement of the weld at primary side
- e_2 Reinforcement of the weld at secondary side

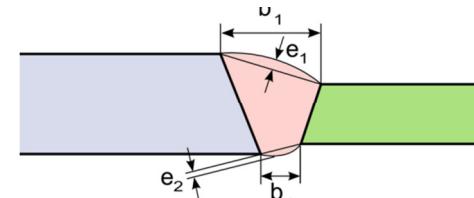


Figure 48: Butt Joint Weld parameters

Inside the xMCF File the following parameters can be specified:

Parameter	xMCF-Key	Multiplicity	Value Range	Status	Default Value
b	width	1 – 2	≥ 0	Optional	-
e	-	(1 – 2)	(≥ 0)	(Optional)	(0)

Table 83: Parameters of Butt Joint Weld

Note: The reinforcement is currently not defined as `attribute` in the version 3.0 document!

8.2.5.3 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute `base`.

Attribute “technology”

The value for the attribute `technology` can be specified using the following values:

- `resistance`
- `arc`
- `laser` (Energy beam / Laser)

8.2.5.4 Element “weld_position”

For the element `<weld_position/>`, the following attributes can be specified for the Butt Joint:

Attributes	Type	Status	Constraint
u	Numeric	Required	$0 \leq u \leq 1$

Attributes	Type	Status	Constraint
x	Numeric	Required	-
y	Numeric	Required	-
z	Numeric	Required	-
reference	Boolean	Optional	“false”
section	Selection	Optional	-
width	Numeric	Optional	-
filler	Selection	Optional	-
filler_material	Alphanumeric	Optional	-

Table 84: Attributes of element <weld_position/> for Butt Joint**Attributes “u, x, y, z, reference”**

Detailed definition can be found in section 8.2.4.3.2 Welding Position.

Attribute “section”

Valid values for the attribute **section** of a Butt Joint are:

- I
- U
- V
- X
- Y
- Radius

Attribute “width”

The attribute value **width** specifies the width of the weld.

Attribute “filler”

Valid values for the attribute **filler** can be:

- yes
- no

Note: Depending on the technology the default value can be different (see in Generic Seam Weld Definition section under attribute **filler**).

Attribute “filler_material”

The attribute **filler_material** specifies the applied material during the welding process.

Example A (within only necessary attributes):

```
<seamweld>
  <butt_joint base="1" technology="arc">
    <weld_position u="0.2" x="1" y="0" z="1"
      section="X"
      width="1.5"/>
    <sheet_parameter ... />
  </butt_joint>
</seamweld>
```

Example B (within every *attribute*):

```
<seamweld>
  <butt_joint base="1" technology="arc">
    <weld_position u="0.2" x="1" y="0" z="1"
      reference="true"
      section="X"
      width="1.5"
      filler="yes"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </butt_joint>
</seamweld>
```

8.2.5.5 Element “sheet_parameter”

For the element `<sheet_parameter/>`, the following attributes can be specified for the Butt Joint:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <code><part></code> index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 85: Attributes of element `<sheet_parameter/>` for Butt Joint

Example A (within only required *attributes*):

```
<seamweld>
  <butt_joint base="1" technology="arc">
    <weld_position u="0.2" x="1" y="0" z="1" ... />
    <sheet_parameter index="2" gap="0" sheet_thickness="1.5" sheet_angle="180" />
  </butt_joint>
</seamweld>
```

8.2.6 Corner Weld

The principles of the modeling of corner welds for xMCF are described in this section. A corner weld describes a connection between two or three sheets welded together.

The XML definition of a Corner Weld supports up to four positions. Each of the weld positions is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.6.1 Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1 Thickness of welded sheet
- c Gap between base sheet and welded sheet
- v Misalignment of welded sheet

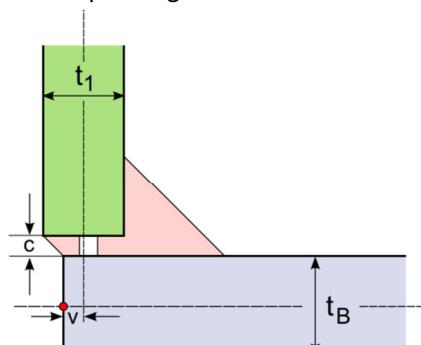


Figure 49: Corner Weld Sheet Layout

8.2.6.2 Weld Parameters

The parameters of the welds are the same for all of the potential welds on the connection:

- a_i Thickness of the weld (a-value, throat)
- d_i Depth of the penetration
- β_i Weld angle

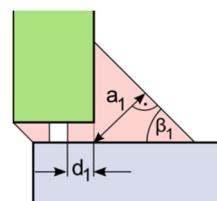


Figure 50: Corner Weld Parameters

For the penetration the ratio η_i of the penetration depth to the sheet thickness is specified inside the xMCF file.

This is computed by $\eta_i = \frac{d_i}{t_j} \sin \alpha_j$ where variable i is specifying the weld index and variable j is defined by the sheet index of the welded sheet related to the weld. (α_j in case of a Corner Weld is 90° and therefore $\sin \alpha_j = 1$.)

Inside the xMCF File the following parameters can be specified:

Parameter	xMCF-Key	Multiplicity	Value Range	Status	Default Value
a	thickness	1 – 2	≥ 0	Optional	
β	angle	0 – 2	≥ 0	Optional	45 [deg]
η	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

Table 86: Parameters of Corner Weld

All other parameters are provided by the model itself.

8.2.6.3 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute **base**.

Attribute “technology”

The value for the attribute **technology** can be specified using the following values:

- **resistance**
- **arc**
- **laser** (Energy beam / Laser)

8.2.6.4 Element “weld_position”

For the element `<weld_position>` the following attributes can be specified for the Corner Weld:

Attributes	Type	Status
<code>u</code>	Numeric	Required
<code>x</code>	Numeric	Required
<code>y</code>	Numeric	Required
<code>z</code>	Numeric	Required
<code>reference</code>	Boolean	Optional

section	Selection	Optional
thickness	Numeric	Optional
angle	Numeric	Optional
shape	Selection	Optional
penetration	Numeric	Optional
filler	Selection	Optional
filler_material	Alphanumeric	Optional

Table 87: Attributes of element <weld_position/> for Corner Weld**Attributes “u, x, y, z, reference”**

Detailed definition can be found in section 8.2.4.3.2 Welding Position.

Attribute “section”

Valid values for the attribute *section* of a corner weld are:

- *HV*
- *U*
- *Fillet*

Attribute “thickness”

The attribute *thickness* specifies the thickness (a-value, throat) of the weld. Depending on the section this is required, optional or not allowed:

<i>Attribute value “section”</i>	<i>Attribute “thickness”</i>
HV	Optional
U	Not allowed
Fillet	Required

Table 88: Values of Attribute section**Attribute “angle”**

The attribute *angle* specifies the angle of the weld relative to the base sheet. Depending on the section this is optional or not allowed:

<i>Attribute value “section”</i>	<i>Attribute “angle”</i>
HV	Optional
U	Not allowed
Fillet	Required

Table 89: Values of Attribute angle**Attribute “shape”**

The attribute *shape* defines the shape of the weld throat.

Attribute “penetration”

The attribute *penetration* specifies the degree of penetration resulting from the welding.

Attribute “filler”

Valid values for the attribute filler can be:

- *yes*
- *no*

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute *filler*).

Attribute “filler_material”

The attribute *filler_material* specifies the applied material during the welding process.

Example A (within each *attribute*):

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0" x="0" y="1" z="0"
      reference="no"
      section="Fillet"
      thickness="1.5"
      angle="30"
      shape="concave"
      penetration="0.5"
      filler="yes"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </corner_weld>
</seamweld>
```

8.2.6.5 Element “sheet_parameter”

For the element `<sheet_parameter>`, the following attributes can be specified for the Corner Weld:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <code><part></code> index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 90: Attributes of element `<sheet_parameter>` for Corner Weld

Example A (within only required *attributes*):

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0" x="0" y="1" z="0" ... />
    <sheet_parameter index="2" gap="0" sheet_thickness="1.5" sheet_angle="90" />
  </corner_weld>
</seamweld>
```

8.2.7 Edge Weld

The principles of the modeling of edge welds for xMCF are described in this section. An Edge Weld describes a connection between two sheets welded at their forehead side.

The XML definition of an Edge Weld supports one position. The weld position is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.7.1 Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1 Thickness of welded sheet
- c Gap between base and welded sheet

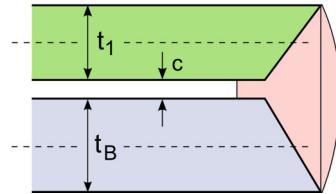


Figure 51: Edge Weld Sheet Layout

8.2.7.2 Weld Parameters

The parameters of the weld are described below:

- b Width of the weld
- e Reinforcement

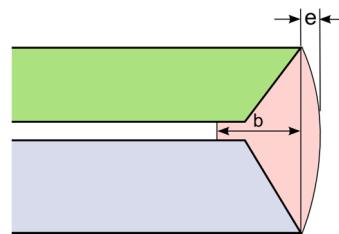


Figure 52: Edge Weld parameters

Parameter	xMCF-Key	Multiplicity	Value Range	Status	Default Value
b	width	1	≥ 0	Optional	-
c	gap	0 – 1	≥ 0	Optional	0
e	-	0 – 1	≥ 0	Optional	0

Table 91: Parameters of Edge Weld

Note: The reinforcement is currently not defined as `attribute` in the version 3.0 document!

8.2.7.3 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute `base`.

Attribute “technology”

The value for the attribute `technology` can be specified using the following values:

- `resistance`
- `arc`
- `laser` (Energy beam / Laser)

8.2.7.4 Element “weld_position”

For the element `<weld_position/>` the following attributes can be specified for the Edge Weld:

Attributes	Type	Status
<code>u</code>	Numeric	Required
<code>x</code>	Numeric	Required
<code>y</code>	Numeric	Required
<code>z</code>	Numeric	Required
<code>reference</code>	Boolean	Optional
<code>section</code>	Selection	Optional

width	Numeric	Optional
filler	Selection	Optional
filler_material	Alphanumeric	Optional

Table 92: Attributes of element <weld_position/> for Edge Weld**Attributes “u, x, y, z, reference”**

Detailed definition can be found in section 8.2.4.3.2 Welding Position.

Attribute “section”

Valid values for the attribute *section* of a edge weld are:

- I
- V
- U

Attribute “width”

The attribute *width* specifies the width of the weld.

Attribute “filler”

Valid values for the attribute *filler* can be:

- yes
- no

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute *filler*).

Attribute “filler_material”

The attribute *filler_material* specifies the applied material during the welding process.

Example A (within each optional attribute):

```
<seamweld>
  <edge_weld base="1" technology="arc">
    <weld_position u="1" x="1" y="1" z="0"
      reference="no"
      section="V"
      width="2"
      filler="yes"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </edge_weld>
</seamweld>
```

8.2.7.5 Element “sheet_parameter”

For the element <sheet_parameter/>, the following attributes can be specified for the Corner Weld:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <part> index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 93: Attributes of element <sheet_parameter/> for Corner Weld

Example A (within only required *attributes*):

```
<seamweld>
  <edge_weld base="1" technology="resistance">
    <weld_position u="1" x="1" y="1" z="0" ... />
    <sheet_parameter index="2" gap="0" sheet_thickness="1.5" sheet_angle="90" />
  </edge_weld>
</seamweld>
```

8.2.8 I-Weld

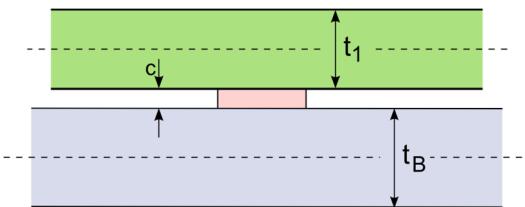
The principles of the modeling of I-welds for xMCF are described in this section. An I-Weld describes a connection between two sheets welded together.

The XML definition of an I-Weld supports one weld position. The weld position is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.8.1 Sheet Parameters

The parameters to describe the connection are:

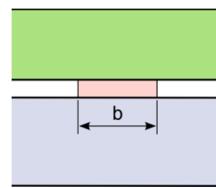
- t_B Thickness of base sheet
- t_1 Thickness of welded sheet
- c Gap between base and welded sheet

**Figure 53: I-Weld Sheet Layout**

8.2.8.2 Weld Parameters

The parameters of the weld are described below:

- b Width of the weld

**Figure 54: I-Weld Parameters**

Parameter	xMCF-Key	Multiplicity	Value Range	Status	Default Value
b	width	1	≥ 0	Optional	-

Table 94: Parameters of I-Weld

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

8.2.8.3 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute `base`.

Attribute “technology”

The value for the attribute `technology` can be specified using the following values:

- `resistance`
- `arc`
- `laser` (Energy beam / Laser)

8.2.8.4 Element “weld_position”

For the element `<weld_position/>` the following attributes can be specified for the I-Weld:

Attributes	Type	Status
u	Numeric	Required
x	Numeric	Required
y	Numeric	Required
z	Numeric	Required
reference	Boolean	Optional
width	Numeric	Optional
filler	Selection	Optional
filler_material	Alphanumeric	Optional

Table 95: Attributes of element `<weld_position/>` for I Weld

Attributes “u, x, y, z, reference”

Detailed definition can be found in section 8.2.4.3.2 Welding Position.

Attribute “width”

The attribute `width` specifies the width of the weld.

Attribute “filler”

Valid values for the attribute `filler` can be:

- `yes`
- `no`

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute `filler`).

Attribute “filler_material”

The attribute `filler_material` specifies the applied material during the welding process.

Example A (within each attribute):

```

<seamweld>
  <i_weld base="1" technology="laser">
    <weld_position u="0" x="1" y="1" z="1"
      reference="no"
      width="1.0"
      filler="no"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </i_weld>
</seamweld>

```

8.2.8.5 Element “sheet_parameter”

For the element `<sheet_parameter/>`, the following attributes can be specified for the I Weld:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to < <i>part</i> > index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 96: Attributes of element <sheet_parameter/> for I Weld**Example A (within only required attributes):**

```

<seamweld>
    <i_weld base="1" technology="laser">
        <weld_position u="0" x="1" y="1" z="1" ... />
        <sheet_parameter index="2" gap="0" sheet_thickness="1.5" sheet_angle="0"/>
    </i_weld>
</seamweld>

```

8.2.9 Overlap Weld

The principles of the modeling of overlap welds for xMCF are described in this section. An Overlap Weld describes a connection between two or three sheets welded together.

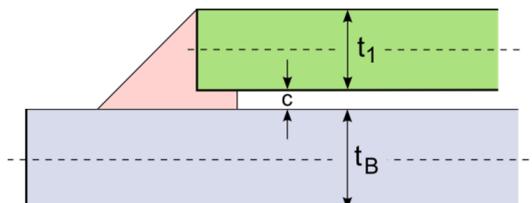
The XML definition of an Overlap Weld supports up to three weld positions. Each of the weld positions is specified using the element <weld_position/> with the corresponding attributes and nested elements inside the subtype definition.

8.2.9.1 Simple Overlap Weld

Sheet Parameters

The parameters to describe the connection are:

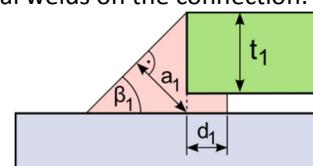
- t_B Thickness of base sheet
- t_1 Thickness of welded sheet
- c Gap between base and welded sheet

**Figure 55: Overlap Weld Sheet Layout**

Weld Parameters

The parameters of the welds are the same for all of the potential welds on the connection:

- a_1 Thickness of the weld (a-value, throat)
- d_1 Depth of the penetration
- β_1 Weld angle

**Figure 56: Overlap Weld Parameters**

For the penetration, the ratio η_1 of the penetration depth to the sheet thickness is specified inside the xMCF file.

This is computed by $\eta_1 = \frac{d_1}{t_1}$, where t_1 is the thickness of the attached sheet (green in above figure),

not of the base sheet.

Inside the xMCF File the following parameters can be specified:

Parameter	χ MCF-Key	Multiplicity	Value Range	Status	Default Value
a	thickness	1	≥ 0	Optional	-
β	angle	0 – 1	≥ 0	Optional	45 [deg]
η	penetration	0 – 1	$0 \leq \eta \leq 1$	Optional	0

Table 97: Parameters of Overlap Weld

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

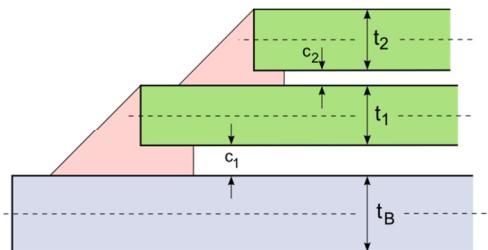
8.2.9.2 Single Sided Double Overlap Weld

The Single Sided Double Overlap Weld is represented by a stacked welding.

Sheet Parameters

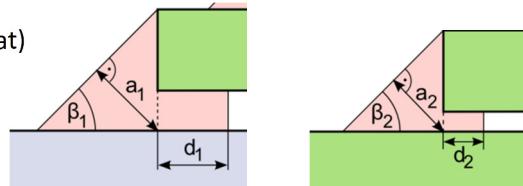
The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1, t_2 Thicknesses of welded sheets
- c_1, c_2 Gaps between base and welded sheets

**Figure 57: Single Sided Double Overlap Weld**

The parameters of the welds are the same for all of the welds on the connection:

- a_i Thickness of the weld (a-value, throat)
- d_i Depth of the penetration
- β_i Weld angle

**Figure 58: Overlap Weld Parameters**

For the penetration the ratio η_i of the penetration depth to the sheet thickness is specified inside the χ MCF file.

This is computed by $\eta_i = \frac{d_i}{t_j} \sin \alpha_j$ where index i is specifying the weld index and index j is defined by the sheet index of the welded sheet related to the weld.

Inside the χ MCF File the following parameters can be specified:

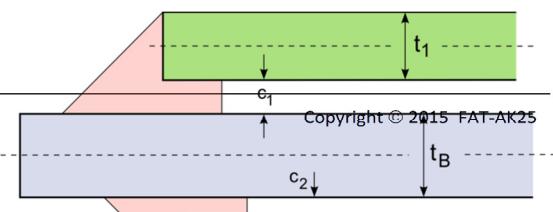
Parameter	χ MCF-Key	Multiplicity	Value Range	Status	Default Value
a	thickness	2	≥ 0	Optional	-
β	angle	0 – 2	≥ 0	Optional	45 [deg]
η	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

Table 98: Parameters of Single Sided Double Overlap Weld

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

8.2.9.3 Double Sided Double Overlap Weld

A Double Sided Double Overlap Weld can have the welds on both sides of the base sheet.



Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1, t_2 Thicknesses of welded sheets
- c_1, c_2 Gaps between base and welded sheets

Weld Parameters

The parameters of the welds are the same for all of the welds on the connection:

- a_i Thickness of the weld (a-value, throat)
- d_i Depth of the penetration
- β_i Weld angle

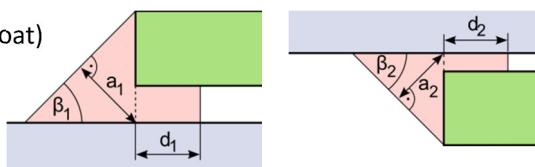


Figure 59: Double Sided Double Overlap Weld

For the penetration the ratio η_i of the penetration depth to the sheet thickness is specified inside the xMCF file.

This is computed by $\eta_i = \frac{d_i}{t_j} \sin \alpha_j$, where index i is specifying the weld index and index j is defined by

the sheet index of the welded sheet related to the weld.

Inside the xMCF File the following parameters can be specified:

Parameter	xMCF-Key	Multiplicity	Value Range	Status	Default Value
a	thickness	2	≥ 0	Optional	-
β	angle	0 – 2	≥ 0	Optional	45 [deg]
η	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

Table 99: Parameters of Double Sided Double Overlap Weld

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

8.2.9.4 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute `base`.

Attribute “technology”

The value for the attribute `technology` can be specified using the following values:

- `resistance`
- `arc`
- `laser` (Energy beam / Laser)

8.2.9.5 Element “weld_position”

For the element `<weld_position>` the following attributes can be specified for the Overlap Weld:

Attributes	Type	Status
base	Numeric	Optional
u	Numeric	Required
x	Numeric	Required
y	Numeric	Required
z	Numeric	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Numeric	Optional
angle	Numeric	Optional
shape	Selection	Optional
penetration	Numeric	Optional
filler	Selection	Optional
filler_material	Alphanumeric	Optional

Table 100: Attributes of element <weld_position/> for Overlap Weld**Attributes “u, x, y, z, reference”**

Detailed definition can be found in section 8.2.4.3.2 Welding Position.

Attribute “base”

For this type of weld the **base** sheet can be specified also inside the element `<weld_position/>`. This is necessary in the case of a stacked welding with two welded sheets.

Attribute “section”

The only valid value currently for the attribute **section** of an Overlap Weld is:

- **Fillet**

Note: This value is the default if the **section** attribute is not specified.

Attribute “thickness”

The attribute **thickness** specifies the thickness (a-value, throat) of the weld.

Attribute “angle”

The attribute **angle** specifies the angle of the weld relative to the base sheet.

Attribute “shape”

The attribute **shape** defines the shape of the weld throat.

Attribute “penetration”

The attribute **penetration** specifies the degree of penetration resulting from the welding.

Attribute “filler”

Valid values for the attribute **filler** can be:

- **yes**

- no

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute *filler*).

Attribute “*filler_material*”

The attribute *filler_material* specifies the applied material during the welding process.

Example A (within each **attribute**, except **base** within **<weld_position/>**):

```

<seamweld>
  <overlap_weld base="1" technology="resistance">
    <weld_position u="0" x="0" y="0" z="1"
      reference="no"
      section="Fillet"
      thickness="1.5"
      angle="30"
      shape="concave"
      penetration="0.5"
      filler="yes"
      filler_material="E7018-X"/>
    <sheet_parameter ... />
  </overlap_weld>
</seamweld>

```

8.2.9.6 Element “*sheet_parameter*”

For the element **<sheet_parameter/>** the following attributes can be specified for the Overlap Weld:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <part> index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 101: Attributes of element *<sheet_parameter/>* for Overlap Weld

Example A (within only required **attributes**):

```

<seamweld>
  <overlap_weld base="1" technology="resistance">
    <weld_position u="0" x="0" y="0" z="1" .../>
    <sheet_parameter index="2" gap="1.0" sheet_thickness="1.5" sheet_angle="0"/>
  </overlap_weld>
</seamweld>

```

8.2.10 Y-Joint

The principles of the modeling of Y-joints for χ MCF are described in this section. A Y-Joint describes a connection between two or three sheets. The Y-Joint defines a connection between a welded sheet and a base sheet. There are two potential welds that can be specified for this type of connection. The parameters for each of the welds can be described separately.

The XML definition of a Y-Joint supports up to three weld positions. Each of the weld positions is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.10.1 Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1 Thickness of welded sheet
- α Sheet angle of welded sheet
- c Gap between base and welded sheet

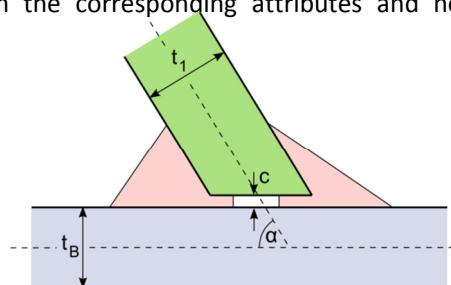


Figure 61: Y-Joint Sheet Layout

8.2.10.2 Weld Parameters

The parameters of the welds are the same for all of the four potential welds on the connection:

- a_i Thickness of the weld (a -value, throat)
- d_i Depth of the penetration
- β_i Weld angle

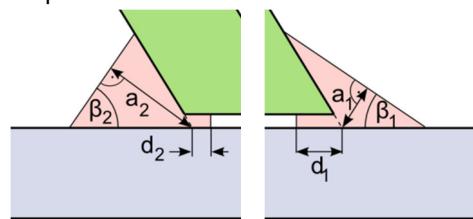


Figure 62: Parameters of Y-Joint Weld

For the penetration the ratio η_i of the penetration depth to the sheet thickness is specified inside the χ MCF file.

This is computed by $\eta_i = \frac{d_i}{t_j} \sin \alpha_j$ where index i is specifying the weld index and index j is defined by

the sheet index of the welded sheet related to the weld.

Inside the χ MCF File only a subset can be specified:

Parameter	χ MCF-Key	Multiplicity	Value Range	Status	Default Value
a	thickness	1 – 2	≥ 0	Optional	-
β	angle	0 – 2	≥ 0	Optional	45 [deg]
η	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

Table 102: Parameters of Y-Joint

8.2.10.3 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute `base`.

Attribute “technology”

The value for the attribute `technology` can be specified using the following values:

- *resistance*
- *arc*
- *laser* (Energy beam / Laser)

8.2.10.4 Element “weld_position”

For the element <*weld_position*> the following attributes can be specified for the Y-Joint:

Attributes	Type	Status
base	Numeric	Optional
u	Numeric	Required
x	Numeric	Required
y	Numeric	Required
z	Numeric	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Numeric	* see attribute description
angle	Numeric	* see attribute description
penetration	Numeric	* see attribute description
filler	Selection	Optional
filler_material	Alphanumeric	Optional
shape	Selection	Optional

Table 103: Attributes of element <*weld_position*> for Y Joint

Attributes “u, x, y, z, reference”

Detailed definition can be found in section 8.2.4.3.2 Welding Position.

Attribute “base”

For this type of weld the *base* sheet can be specified also inside the element <*weld_position*>. This is necessary in the case of a stacked welding with two welded sheets.

Attribute “section”

The attribute *section* can be absent in the case of attribute value *technology="laser"* inside element subtype.

Valid values for the attribute *section* (if present) of a Y-Joint are:

- *Fillet*
- *HV*
- *HY*

Attribute “thickness”

The attribute *thickness* specifies the thickness (a-value, throat) of the weld. Depending on the *section* this is required, optional or not allowed:

Attribute value “section”	Attribute “thickness”
HV	Optional

HY	Not allowed
Fillet	Required

Table 104: Value Dependency of Attribute *thickness***Attribute "angle"**

The attribute `angle` specifies the angle of the weld relative to the base sheet.

Attribute "penetration"

The attribute `penetration` specifies the degree of penetration resulting from the welding.

Attribute "shape"

The attribute `shape` defines the shape of the weld throat.

Attribute "filler"

Valid values for the attribute filler can be:

- `yes`
- `no`

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute `filler`).

Attribute "filler_material"

The attribute `filler_material` specifies the applied material during the welding process.

Example A (within each `attribute`, except `base` within `<weld_position/>`):

```

<seamweld>
  <y_joint base="1" technology="resistance">
    <weld_position u="0.5" x="1" y="0" z="1"
      reference="no"
      section="HY"
      thickness="0.5"
      angle="30"
      penetration="0.5"
      filler="yes"
      filler_material=" E7018-X"
      shape="concave"/>
    <weld_position u="0.2" x="-1" y="0" z="1"
      reference="no"
      section="HY"
      thickness="0.5"
      angle="45"
      penetration="0.5"
      filler="yes"
      filler_material=" E7018-X"
      shape="concave"/>
    <sheet_parameter ... />
  </y_joint>
</seamweld>

```

8.2.10.5 Element "sheet_parameter"

For the element `<sheet_parameter/>`, the following attributes can be specified for the Y-Joint:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <code><part></code> index attribute
gap	Numeric	Optional	Default value is 0

Attributes	Type	Status	Constraint / Remarks
sheet_thickness	Numeric	Optional	-
sheet_angle	Numeric	Optional	-

Table 105: Attributes of element <sheet_parameter/> for Y-Joint**Example A (within only required attributes):**

```
<seamweld>
  <y_joint base="1" technology="resistance">
    <weld_position u="0.2" x="1" y="0" z="1" .../>
    <sheet_parameter index="2" gap="1.0" sheet_thickness="1.5" sheet_angle="180"/>
  </y_joint>
</seamweld>
```

8.2.11 K-Joint

The K-Joint connects two welded sheets from the same side to a base sheet.

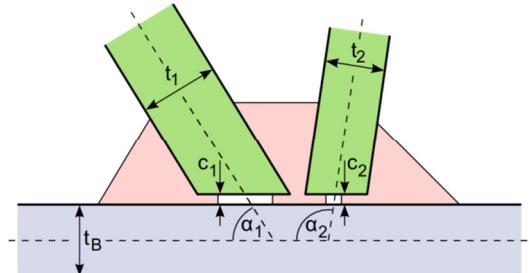
There are four potential welds that can be specified for this type of connection. The parameters for each of the welds can be described separately.

The XML definition of a K-Joint supports up to three weld positions. Each of the weld positions is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.11.1 Sheet Parameters

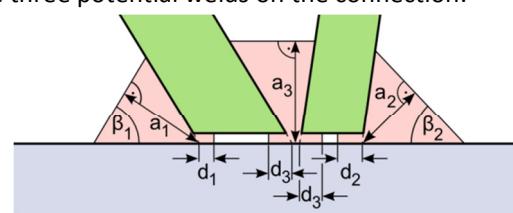
The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1, t_2 Thickness of welded sheet
- α_1, α_2 Sheet angle of welded sheet
- c_1, c_2 Gap between base and welded sheet

**Figure 63: K-Joint Sheet Layout**

The parameters of the welds are the same for all of the three potential welds on the connection:

- a_i Thickness of the weld (a-value, throat)
- d_i Depth of the penetration
- β_i Weld angle

**Figure 64: Parameters of K-Joint Weld**

For the penetration the ratio η_i of the penetration depth to the sheet thickness is specified inside the xMCF file.

This is computed by $\eta_i = \frac{d_i}{t_j} \sin \alpha_j$ where index i is specifying the weld index and index j is defined by the sheet index of the welded sheet related to the weld.

The following parameters can be specified for the K-Joint:

Parameter	<i>xMCF-Key</i>	Multiplicity	Value Range	Status	Default Value
a	thickness	1 – 3	≥ 0	Optional	-
β	angle	0 – 2	≥ 0	Optional	45 [deg]
η	penetration	0 – 3	$0 \leq \eta \leq 1$	Optional	0

Table 106: Parameters of K-Joint

The penetration of the 3rd weld connection (d_3) is assumed to be equal on both welded sheet. There is only one value to be specified.

8.2.11.3 Attributes

Attribute "base"

The index for the base sheet is specified using the attribute *base*.

Attribute "technology"

The value for the attribute *technology* can be specified using the following values:

- *resistance*
- *arc*
- *laser* (Energy beam / Laser)

8.2.11.4 Element "weld_position"

For the element <*weld_position*> the following attributes can be specified for the K-Joint:

Attributes	Type	Status
base	Numeric	Optional
u	Numeric	Required
x	Numeric	Required
y	Numeric	Required
z	Numeric	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Numeric	* see attribute description
angle	Numeric	* see attribute description
penetration	Numeric	* see attribute description
filler	Selection	Optional
filler_material	Alphanumeric	Optional
shape	Selection	Optional

Table 107: Attributes of element <*weld_position*> for K Joint

Attribute "base"

For this type of weld the *base* sheet can be specified also inside the element <*weld_position*>. This is necessary in the case of a stacked welding with two welded sheets.

Attribute “section”

The attribute `section` can be absent in the case of attribute value `technology="laser"` inside element subtype.

Valid values for the attribute `section` (if present) of a K-Joint are:

- `Fillet`
- `HV`
- `HY`

Attribute “thickness”

The attribute `thickness` specifies the thickness (a-value, throat) of the weld. Depending on the section this is required, optional or not allowed:

Attribute value “section”	Attribute “thickness”
HV	Optional
HY	Not allowed
Fillet	Required

Table 108: Value Dependency of Attribute `thickness`

Attribute “angle”

The attribute `angle` specifies the angle of the weld relative to the base sheet. The weld angle of a center weld of a K-Joint is assumed to be parallel to the base sheet (i.e. 0°).

Attribute “penetration”

The attribute `penetration` specifies the degree of penetration resulting from the welding.

Attribute “shape”

The attribute `shape` defines the shape of the weld throat.

Attribute “filler”

Valid values for the attribute filler can be:

- `yes`
- `no`

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute `filler`).

Attribute “filler_material”

The attribute `filler_material` specifies the applied material during the welding process.

Example A (within each `attribute`, except `base` within `<weld_position/>`):

```
<seamwweld>
  <k_joint base="2" technology="resistance">
    <weld_position u="1.0" x="2" y="0" z="1"
      reference="yes"
      penetration="3.0"
      thickness="1.4"
      angle="15"
      section="HV"
      filler="yes"
      filler_material=" E7018-X"
```

```

        shape="straight" />
<weld_position u="0.0" x="1" y="0" z="2"
    reference="yes"
    penetration="3.0"
    thickness="1.1"
    angle="90"
    section="HV"
    filler="yes"
    filler_material=" E7018-X"
    shape="straight" />
<weld_position u="1.0" x="-2" y="0" z="1"
    reference="yes"
    penetration="3.0"
    thickness=".5"
    angle="30"
    section="HV"
    filler="yes"
    filler_material=" E7018-X"
    shape="straight" />
<sheet_parameter ... />
<sheet_parameter ... />
</k_joint>
</seamweld>

```

8.2.11.5 Element “sheet_parameter”

For the element `<sheet_parameter/>`, the following attributes can be specified for the K Joint:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <code><part></code> index attribute
gap	Numeric	Required	-
thickness	Numeric	Required	-
sheet_angle	Numeric	Required	-

Table 109: Attributes of element `<sheet_parameter/>` for K Joint

Example A (within only required `attributes`):

```

<seamweld>
  <k_joint base="2" technology="resistance">
    <weld_position u="1.0" x="2" y="0" z="1" .../>
    <weld_position u="0.0" x="1" y="0" z="2" .../>
    <weld_position u="1.0" x="-2" y="0" z="1" .../>
    <sheet_parameter index="1" gap="1.5" thickness="1.5" sheet_angle="45"/>
    <sheet_parameter index="3" gap="1.0" thickness="1.5" sheet_angle="30"/>
  </k_joint>
</seamweld>

```

8.2.12 Cruciform Joint

The cross joint connects two welded sheets from different sides to a base sheet.

There are four potential welds that can be specified for this type of connection. The parameters for each of the welds can be described separately.

The XML definition of a Cruciform Joint supports up to four weld positions. Each of the weld positions is specified using the element `<weld_position/>` with the corresponding attributes and nested elements inside the subtype definition.

8.2.12.1 Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1, t_2 Thickness of welded sheet
- α_1, α_2 Sheet angle of welded sheet
- c_1, c_2 Gap between base and welded sheet

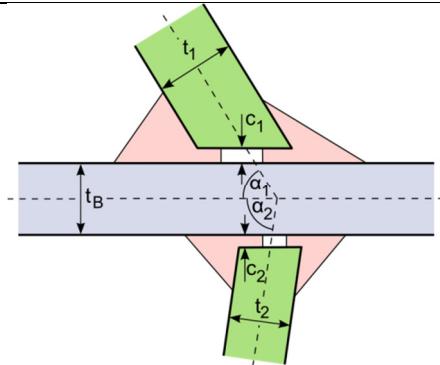


Figure 65: Cruciform Joint Sheet Layout

8.2.12.2 Weld Parameters

The parameters of the welds are the same for all of the four potential welds on the connection:

- a_i Thickness of the weld (a-value, throat)
- d_i Depth of the penetration
- β_i Weld angle

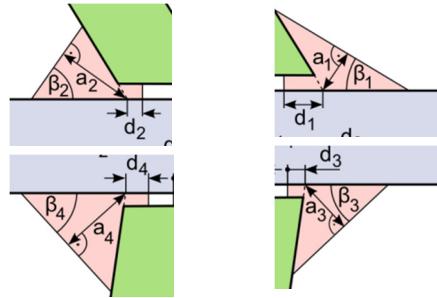


Figure 66: Parameters of Cruciform Joint

For the penetration the ratio η_i of the penetration depth to the sheet thickness is specified inside the xMCF file.

This is computed by $\eta_i = \frac{d_i}{t_j} \sin \alpha_j$ where index i is specifying the weld index and index j is defined by

the sheet index of the welded sheet related to the weld.

The following parameters can be specified for the Cruciform Joint:

Parameter	xMCF-Key	Multiplicity	Value Range	Status	Default Value
a	thickness	2 – 4	≥ 0	Optional	-
β	angle	0 – 4	≥ 0	Optional	45 [deg]
η	penetration	0 – 4	$0 \leq \eta \leq 1$	Optional	0

Table 110: Parameters of Cruciform Joint

8.2.12.3 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute `base`.

Attribute “technology”

The value for the attribute `technology` can be specified using the following values:

- `resistance`
- `arc`
- `laser` (Energy beam / Laser)

8.2.12.4 Element “weld_position”

For the element `<weld_position>` the following attributes can be specified for the Cruciform Joint:

Attributes	Type	Status
base	Numeric	Optional
u	Numeric	Required
x	Numeric	Required
y	Numeric	Required
z	Numeric	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Numeric	* see attribute description
angle	Numeric	* see attribute description
penetration	Numeric	* see attribute description
filler	Selection	Optional
filler_material	Alphanumeric	Optional
shape	Selection	Optional

Table 111: Attributes of element `<weld_position/>` for Cruciform Joint**Attribute "base"**

For this type of weld the `base` sheet can be specified also inside the element `<weld_position/>`. This is necessary in the case of a stacked welding with two welded sheets.

Attribute "section"

The attribute `section` can be absent in the case of attribute value `technology="laser"` inside element subtype.

Valid values for the attribute `section` (if present) of a cross joint are:

- `Fillet`
- `HV`
- `HY`

Attribute "thickness"

The attribute `thickness` specifies the thickness (a-value, throat) of the weld. Depending on the section this is required, optional or not allowed:

Attribute value "section"	Attribute "thickness"
HV	Optional
HY	Not allowed
Fillet	Required

Table 112: Value Dependency of Attribute `thickness`**Attribute "angle"**

The attribute `angle` specifies the angle of the weld relative to the base sheet.

Attribute “penetration”

The attribute `penetration` specifies the degree of penetration resulting from the welding¹⁶.

Attribute “shape”

The attribute `shape` defines the shape of the weld throat.

Attribute “filler”

Valid values for the attribute filler can be:

- `yes`
- `no`

Note: Depending on the technology the default value can different (see in Generic Seam Weld Definition section under attribute `filler`).

Attribute “filler_material”

The attribute `filler_material` specifies the applied material during the welding process.

Example A (within each `attribute`, except `base` within `<weld_position/>`):

```

<seamweld>
  <cruciform_joint base="1" technology="arc">
    <weld_position u="0.2" x="1" y="0" z="1"
      thickness="3.0"
      penetration="0.8"
      section="HY"
      angle="30"
      reference="yes"
      filler="yes"
      filler_material=" E7018-X"
      shape="straight" />
    <weld_position u="0.4" x="-1" y="0" z="-1"
      thickness="4.0"
      penetration="0.4"
      section="HY"
      angle="45"
      reference="yes"
      filler="yes"
      filler_material=" E7018-X"
      shape="straight" />
    <weld_position u="0.6" x="-1" y="0" z="1"
      thickness="5.0"
      penetration="0.8"
      section="HY"
      angle="50"
      reference="yes"
      filler="yes"
      filler_material=" E7018-X"
      shape="straight" />
    <weld_position u="0.8" x="1" y="0" z="-1"
      thickness="6.0"
      penetration="0.4"
      section="HY"
      angle="75"
      reference="yes"
      filler="yes"
      filler_material=" E7018-X"
      shape="straight" />
    <sheet_parameter ... />
    <sheet_parameter ... />
  </cruciform_joint>
</seamweld>

```

¹⁶ The attribute penetration of a `<weld_position/>` holds for all sheets connected by this `<weld_position/>` (e. g. important for K-Joints).

8.2.12.5 Element “sheet_parameter”

For the element `<sheet_parameter/>`, the following attributes can be specified for the Cruciform Joint:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <code><part></code> index attribute
gap	Numeric	Required	-
thickness	Numeric	Required	-
sheet_angle	Numeric	Required	-

Table 113: Attributes of element `<sheet_parameter/>` for Cruciform Joint

Example A (within only required attributes):

```

<seamweld>
  <cruciform_joint base="1" technology="arc">
    <weld_position u="0.2" x="1" y="0" z="1" .../>
    <weld_position u="0.4" x="-1" y="0" z="-1" .../>
    <weld_position u="0.6" x="-1" y="0" z="1" .../>
    <weld_position u="0.8" x="1" y="0" z="-1" .../>
    <sheet_parameter index="2" gap="1.5" thickness="1.5" sheet_angle="90"/>
    <sheet_parameter index="3" gap="1.0" thickness="1.5" sheet_angle="90"/>
  </cruciform_joint>
</seamweld>

```

8.2.13 Flared Joint

Sheet Parameters

The parameters to describe the connection are:

- t_B Thickness of base sheet
- t_1 Thickness of welded sheet
- c Gap between base and welded sheet

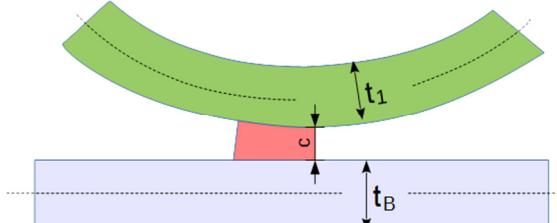


Figure 67: Flared Joint Sheet Layout

Weld Parameters

The parameters of the welds are described below:

- b width of the weld

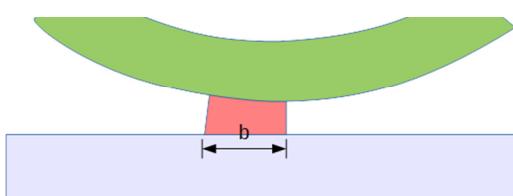


Figure 68: Parameters of Flared Joint Weld

The following parameter can be specified for the Flared Joint:

Parameter	XMCF-Key	Multiplicity	Value Range	Status	Default Value
b	Width	1	≥ 0	Optional	-

Table 114: Parameters of Flared joint

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

8.2.13.1 Attributes

Attribute “base”

The index for the base sheet is specified using the attribute base.

Attribute “technology”

The value for the attribute technology can be specified using the following values:

- arc

8.2.13.2 Element “weld_position”

For the element `<weld_position/>` the following attributes can be specified for the Flared-Joint:

Attributes	Type	Status
u	Numeric	Required
x	Numeric	Required
y	Numeric	Required
z	Numeric	Required
reference	Boolean	Optional
width	Numeric	Optional

Table 115: Attributes of element `<weld_position/>` for Flared Joint

Attributes “u, x, y, z, reference”

Detailed definition can be found in section Welding Position.

Attribute “width”

The attribute `width` specifies the width of the weld.

Example A (within each attribute):

```
<seamweld>
  <flared_joint base="1" technology="arc">
    <weld_position u="0" x="1" y="1" z="1"
      reference="no"
      width="1.0" />
    <sheet_parameter ... />
  </flared_joint >
</seamweld>
```

8.2.13.4 Element “sheet_parameter”

For the element `<sheet_parameter/>`, the following attributes can be specified for the Flared Joint:

Attributes	Type	Status	Constraint / Remarks
index	Numeric	Required	It must be referenced to <code><part></code> index attribute
gap	Numeric	Optional	Default value is 0
sheet_thickness	Numeric	Optional	-

Table 116: Attributes of element `<sheet_parameter/>` for Flared Joint

Example A (within only required *attributes*):

```
<seamweld>
  <flared_joint base="1" technology="arc">
    <weld_position u="0" x="1" y="1" z="1" ... "/>
    <sheet_parameter index="2" gap="0" sheet_thickness="1.5"/>
  </flared_joint >
</seamweld>
```

8.3 Adhesive Lines

An adhesive line is denoted by an element `<adhesive_line/>`. This element is described completely by its attributes and nested elements.

XML specification of `<connection_1d/>` with attribute `label`:

Attributes	Type	Value Space	Status	Constraint / Remarks
label	Alphanumeric	Alphanumeric	Optional	-
quality_control	Alphanumeric	Alphanumeric	Optional	See section 6.4 Attribute <code>quality_control</code>

Table 117: Attributes of `<connection_1d/>`

Nested Elements	Multiplicity	Status	Constraint / Remarks
adhesive_line	0-1	Optional	-
loc_list	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 118: Nested elements of `<connection_1d/>`

Element “adhesive_line”

For the `<adhesive_line/>` element, the following attributes can be specified:

Attributes	Type	Value Space	Status	Constraint
base	Integer	> 0	Optional	-
width	Floating point	≥ 0.0	Optional	-
thickness	Floating point	≥ 0.0	Optional	-
material	Alphanumeric	Alphanumeric	Optional	-

Table 119: Attributes of element `<adhesive_line/>`

- **base**: the index of the flange partner, which the robot applies the adhesive to, before the flange partners are fitted together.
- **width**: the width of the adhesive
- **thickness**: the height of the adhesive
- **material**: the name of the adhesive material according to CAD/PDM. For CAE applications, another label from a reduced data base may be applicable. This is to be stored in `<appdata/>`, then.

All attributes of this connection are optional for import to CAD or CAE processors. However, specific FE solvers may declare some of them to be mandatory.

General defaults are: 0 for numeric values, "" for strings. However, these defaults are not always useful for CAE.

Element "loc_list"

This follows the same syntax as defined in section 8.1.2 Location.

Element "appdata"

This follows the same syntax as defined in section 5.2.1 User Specific Data <appdata>.

Example A (without *base* definition):

```
<connection_1d label="100006">
  <adhesive_line width="5" thickness="2" material="CAD_Material"/>
  <!-- material is optional -->
  <loc_list>
    <loc v="1"> 2169.300 -489.495 1773.936 </loc>
    <loc v="2"> 2165.593 -480.000 1790.221 </loc>
    <loc v="3"> 2165.593 480.000 1790.221 </loc>
    <loc v="4"> 2169.302 489.495 1773.937 </loc>
  </loc_list>
  <appdata>
    ...
  </appdata>
</connection_1d>
```

Example B (with *base* definition):

```
<connection_1d label="1544256">
  <adhesive_line base="2" width="1" thickness="1" material="CAD_test_Mat"/>
  <loc_list>
    <loc v="0.0"> 501 1.03333 3.33332 </loc>
    <loc v="0.5"> 502 1.03333 3.33332 </loc>
    <loc v="1.0"> 503 1.03333 3.33332 </loc>
  </loc_list>
  <appdata>
    ...
  </appdata>
</connection_1d>
```

8.4 Hemming Flanges

8.4.1 Introduction

A hemming involves rolling over a metal sheet onto itself, to reinforce an edge and improve appearance. In automotive engineering, the hemming process also involves adhering other metal sheets within the rolled one.

A hemming involves a path around which the outer metal sheet is rolled over. This is called the *hemming root*.

The hemming also consists of three regions, where glue is applied and which are compressed during hem formation.

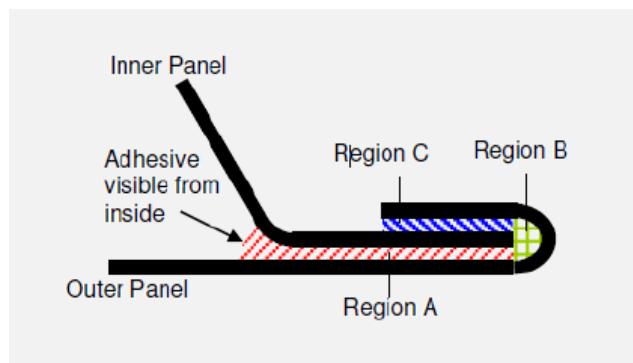


Figure 69: The Three Regions of a Hemming

The solution described below addresses certain features in hemming design:

- The path of the hemming root does not need to coincide with the paths of the adhesive.
- Each region may have a different filling percentage. This is mainly to prevent spillage, but also to maximize contacting overlap.
- Reinforcements may exist in the *Inner Panel*.

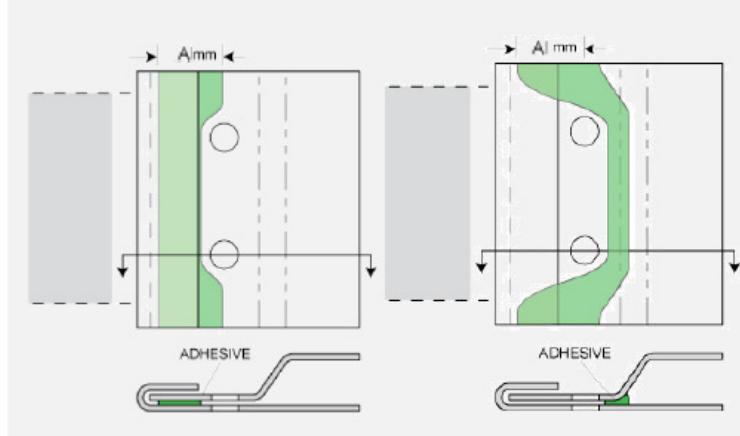


Figure 70: Path Changes and Width Changes in Hemming Flanges

Width and path sometimes change to avoid obstacles, like holes.

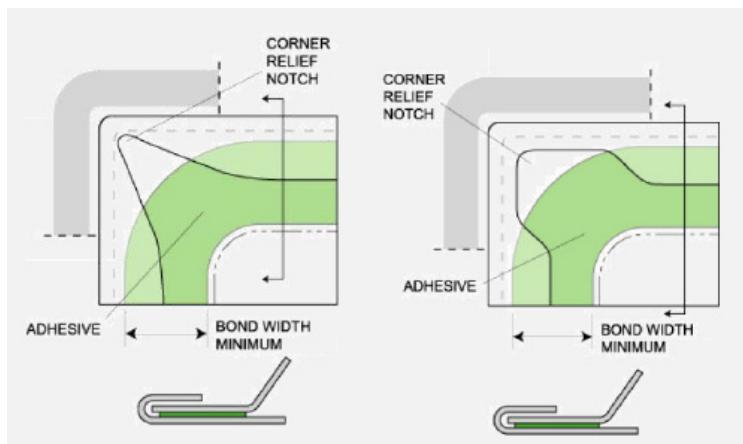


Figure 71: Adhesive Path Differs from Root Path

Adhesive generally follows inner routes around corners.

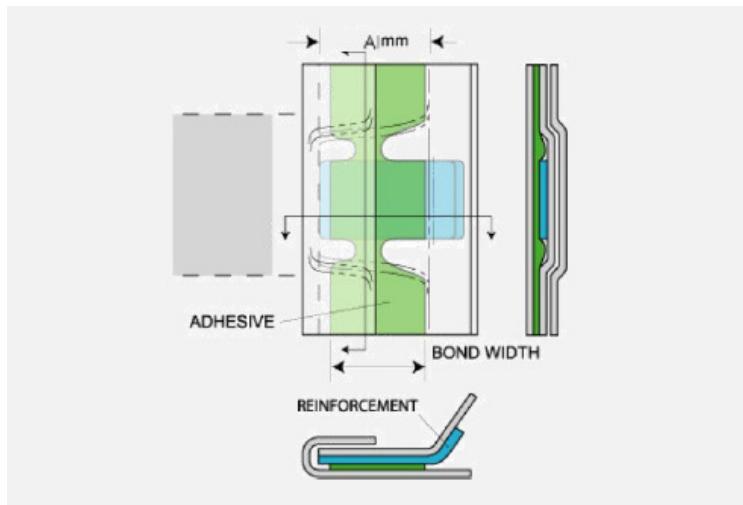


Figure 72: Reinforcements need to be considered as Part of the Inner Panel

Reinforcements need to be considered as part of the Inner Panel, and glued accordingly.

To address the features above, the hemming is treated as a composite connection. This allows for separate paths between the hemming root and the adhesive of each region.

8.4.2 Definition of element `<hemming/>`

XML specification of `<connection_1d/>` with attribute `label`:

Attributes	Type	Value Space	Status	Constraint / Remarks
label	Alphanumeric	Alphanumeric	Optional	-
quality_control	Alphanumeric	Alphanumeric	Optional	See section 6.4 Attribute <code>quality_control</code>

Table 120: Attributes of `<connection_1d/>` for `<hemming/>`

Nested Elements	Multiplicity	Status	Constraint / Remarks
hemming	0-1	Optional	-
loc_list	1	Required	-

Nested Elements	Multiplicity	Status	Constraint / Remarks
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 121: Nested elements of `<connection_1d/>` for `<hemming/>`**Element “loc_list”**

This is the path of the *hemming root*. It follows the same syntax as defined in section 8.1.2 Location.

Element “appdata”

This follows the same syntax as defined in section 5.2.1 User Specific Data `<appdata>`.

Element “hemming”

For the `<hemming/>` element, the following attributes can be specified:

Attributes	Type	Value Space	Status	Constraint
folded_width	Floating point	> 0.0	Optional	-
folded_part	Numeric	-	Optional	Index of the folded sheet

Table 122: Attributes of element `<hemming/>`

- **folded_width**: This is the measure of the width of the folded metal sheet. It is different from the width of the adhesive which may optionally exist.
- **folded_part**: refers to the index of the part that is folded for this kind of connection, as defined in 5.3.1.1 Element `<part/>`.

Its definition is similar to “base” attribute of `<seamwelds/>` in section 8.2.4.2 Type Specification. The usage of adhesive can be specified by the optional nested elements, `<region>`, below.

The three regions of the hemming can be described in the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
region	0-3	Optional	-

Table 123: Nested elements of element `<hemming/>`**Element “region”**

For `<region/>` element, the following attributes can be specified:

Attributes	Type	Value Space	Status	Constraint
label	Alphanumeric	"A", "B", or "C"	Required	-
fill_percentage	Floating point	[0.0, 100.0]	Optional	-

Table 124: Attributes of element `<region/>`

This element defines adhesion properties of region A, B, or C.

- **label**: this is an identifier of the hemming region, according to Figure 69 “The Three Regions of a Hemming”. Only values "A", "B" and "C" are meaningful.
- **fill_percentage**: target hem fill for this region.

The adhesive of hemming regions "A" and "C" can be described in the following nested elements:

Nested Elements	Multiplicity	Status	Constraint / Remarks
connection_1d	1	Optional	Exactly one of these elements. It must either contain an <code><adhesive_line/></code> or an <code><adhesive_face/></code> .
connection_2d			

Table 125: Nested elements of element `<region/>`

The usage of adhesives in the `<region/>` is described in sections 8.3 Adhesive Lines and 9.2 Adhesive Faces.

Note: Region "B" is not expected to contain an adhesive line or face. The definition is left open for future extensions.

Example A (main type as `<hemming/>`):

```

<connected_to>
    <part index="1" label="outer hood panel"/>
    <assembly index="23">
        <part index="2" label="inner hood panel"/>
        <part index="3" label="reinforcement"/>
    </assembly>
</connected_to>
<connection_1d label="100574">
    <loc_list>  <!-- hemming root's path -->
        <loc v="1"> 2169.300 -489.495 1773.936 </loc>
        <loc v="2"> 2165.593 -480.000 1790.221 </loc>
        <loc v="3"> 2165.593 480.000 1790.221 </loc>
        <loc v="4"> 2169.302 489.495 1773.937 </loc>
    </loc_list>
    <appdata>
        ...
    </appdata>
    <hemming folded_width="5" folded_part="1">
        <region_A fill_percentage="50">
            <connection_1d label="100574 region A adhesive">
                <adhesive_line base="1" width="4" thickness="1" material="CAD_test_Mat"/>
                <loc_list>  <!-- adhesive's path -->
                    ...
                </loc_list>
                <appdata>  <!-- adhesive's appdata -->
                    ...
                </appdata>
            </connection_1d>
        </region_A>
        <region_B fill_percentage="100"/>
        <region_C top_index="23" bottom_index="1" fill_percentage="100">
            <connection_2d label="100574 region C adhesive">
                <adhesive_face thickness="1" material="CAD_test_Mat"/>
                <loc_list>  <!-- adhesive face's nodes -->
                    ...
                </loc_list>
                <face_list>  <!-- adhesive's facets -->
                    ...
                </face_list>
                <appdata>  <!-- adhesive's appdata -->
                    ...
                </appdata>
            </connection_2d>
        </region_C>
    </hemming>
</connection_1d>

```

8.5 Sequence Connections

A sequence connection represents a set of 0d connections collectively described, using the definition of a connection line. The connections are uniformly distributed along the line, at a given density. This type of modeling allows for easy optimization of the number of connections along a line.

The distribution of connections is described by *spacing* and *margin*. Spacing is a mandatory dimension.



Figure 73: Sequence without margin

An optional *margin* value allows space to be left from each side.

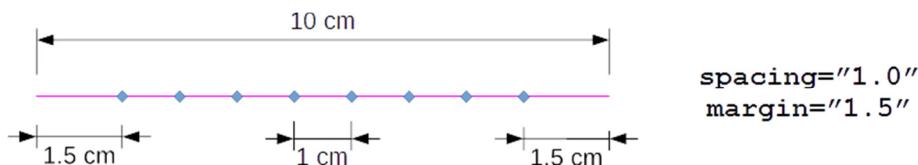


Figure 74: Sequence with margin and spacing

The default value for *margin* is 0.

However, there are cases where the *spacing* and *margin* do not add up to exactly the length of the line. In this case, either the *margin* or the *spacing* may be relaxed:

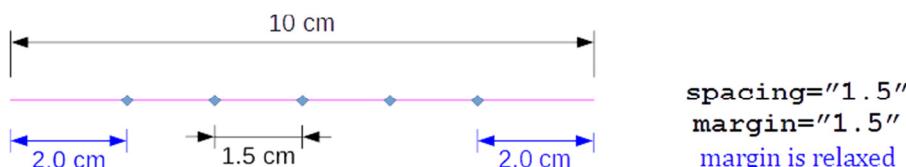


Figure 75: Margin relaxation

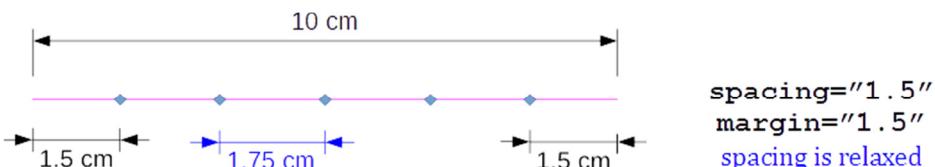


Figure 76: Spacing relaxation

To decide which case is required, one has to give **priority** either to *spacing* or to *margin*.

- When priority is given to *spacing*, the *margin* can be slightly stretched to a greater value, so that the maximum number of connections can fit using the given spacing.
 - If $2 \times \text{margin}$ is greater than the line length, one connection is placed at the middle of the line.
- When priority is given to *margin*, *spacing* can be slightly squeezed or stretched (such that Δ_{spacing} is minimal).

A *<loc_list>* is necessary for this type of connection.

Example A (with minimum definition for `<sequence_connection_0d/>`):

```

<connection_1d label="11000">
  <sequence_connection_0d spacing="30.0">
    <spotweld/>
  </sequence_connection_0d>
  <loc_list>
    ...
  </loc_list>
  <appdata>
    ...
  </appdata>
</connection_1d>

```

Example B (full definition for `<sequence_connection_0d/>`):

```

<connection_1d label="11000">
  <sequence_connection_0d spacing="30.0" margin="1.0" priority="spacing">
    <gumdrop diameter="4.0" mass="10." material="CAD_Material"/>
  </sequence_connection_0d>
  <loc_list>
    ...
  </loc_list>
  <appdata>
    ...
  </appdata>
</connection_1d>

```

To define the type of 0d-connection elements that this connection line describes, any of the connection_0d types can be nested in the `<sequence_connection_0d>` element.

Example C (definition of a `<sequence_connection_0d/>` of `<spotweld/>`):

```

<connection_1d label="11000">
  <sequence_connection_0d spacing="30.0" margin="1.0" priority="spacing">
    <spotweld diameter="6"/>
  </sequence_connection_0d>
  <loc_list>
    ...
  </loc_list>
  <appdata>
    ...
  </appdata>
</connection_1d>

```

XML specification of `<connection_1d/>` in case of `<sequence_connection_0d/>`:

Nested Elements	Multiplicity	Status	Constraint / Remarks
sequence_connection_0d	0-1	Optional	-
loc_list	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 126: Nested elements of `<connection_1d/>` for `<sequence_connection_0d/>`

The XML definition of a `<sequence_connection_0d/>` may contain any of the following 0d connection types:

Nested Elements	Multiplicity	Status	Constraint
spotweld	1	Optional	-
gumdrop	1	Optional	-

Table 127: Nested elements of `<sequence_connection_0d/>`

Remark: nesting 0d elements with directions (such as rivet, screws, robscans) would be impossible with this definition. Note: Only *one* of the nested elements (spotweld, or gumdrop) must exist. If all are missing, then this will default to spotweld.

XML specification of `<sequence_connection_0d/>`:

Attributes	Type	Value Space	Status	Constraint
spacing	Floating point	≥ 0.0	Required	-
margin	Floating point	≥ 0.0	Optional	Default value is 0.0
priority	Selection	{"spacing", "margin"}	Optional	Default value is "spacing"

Table 128: Attributes of element `<sequence_connection_0d/>`

9 2D connections

9.1 Generic Definitions

9.1.1 Identification

Each face connection is optionally identified by its `label`. The XML definition at `<connection_2d>` level contains the following attributes:

Attributes	Type	Status	Constraint / Remarks
label	Alphanumeric	Optional	-
quality_control	Alphanumeric	Optional	See section 6.4 Attribute <code>quality_control</code>

Table 129: Attributes of `<connection_2d>`

Attribute “label”

The label defines the human readable identification of the connection.

Example (typical definition):

```

<connection_list>
  <connection_2d label="adh_patch">
    <loc_list>
      ...
    </loc_list>
    <face_list>
      ...
    </face_list>
    <adhesive_face>
      ...
    </adhesive_face>
  </connection_2d>
</connection_list>

```

9.1.2 Connection Face

The definition of the connection face is described using tessellations. Each tessellation is a set of facets. The facets refer to 3 or 4 points, also described in the same level. Faces of any curvature can be represented by adding more points to the tessellations to obtain the needed accuracy.

The facets do not have any sense of order. The facets refer to the points via an index to the corresponding points, to avoid data duplication. The index is valid only within one certain `<connection_2d>`. Hence, it can start with e. g. 1 every time again.

Element “loc_list”

The list of locations for the definition of the connection face is stored in the element `<loc_list>`. This element contains nested elements `<loc>` defining the location of a point of the connection line in space. These locations have to be uniquely identifiable so that the facets can refer to them.

No additional attributes are associated to the element `<loc_list>`.

The `<loc_list>` element has the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
loc	3-*	Required	-

Table 130: Nested elements of `<loc_list>`

Element "loc"

Each location specified by the element `<loc>` contains three values specifying the x, y and z coordinates of the location.

The attributes associated to the element `<loc>` are:

Attributes	Type	Status	Constraint / Remarks
v	Numeric (integer)	Required	Unique within the parent element <code><connection_2d></code>

Table 131: Attributes of element `<loc>`

The attribute `v` is used to ensure unique identification. The index values must be unique within the `<connection_2d>` element.

Example:

```
<loc_list>
  <loc v="1">  2581.21 -708.408 31.6532  </loc>
  <loc v="2">  2581.42 -708.357 35.2816  </loc>
  <loc v="3">  2581.05 -708.302 39.0643  </loc>
</loc_list>
```

Element "face_list"

The list of facets for the definition of the connection face is stored in the element `<face_list>`. This element contains nested elements `face` defining tessellated facets of the connection face in space. These facets are in no particular order.

No additional attributes are associated to the element `<face_list>`.

The `face_list` element has the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
face	1-*	Required	-

Table 132: Nested elements of element `<face_list>`

Element "face"

Each location specified by the element `<face>` contains *four* values specifying each vertex of the facet, using the `<loc>` identifier, `v`.

Attribute (Vertex)	Type	Status	Constraint
v1	Integer	Required	Must correspond to an v in a loc from loc_list
v2	Integer	Required	Must correspond to an v in a loc from loc_list
v3	Integer	Required	Must correspond to an v in a loc from loc_list
v4	Integer	Optional	Must correspond to an v in a loc from loc_list

Table 133: Attributes of element <face/>

- To represent a quadrangular facet, four distinct vertex indices must be supplied.
- To represent a triangular facet, three distinct vertex indices must be supplied.

Example A: (with minimum definition for <connection_2d/>)

```

<loc_list>
  <loc v="1"> 2001.557 14.435 1736.898 </loc>
  <loc v="2"> 1994.802 14.435 1734.247 </loc>
  <loc v="3"> 1994.790 0.0436 1734.256 </loc>
  <loc v="4"> 2001.547 0.0545 1736.911 </loc>
  <loc v="5"> 2008.298 14.435 1739.550 </loc>
  <loc v="6"> 2008.336 28.784 1739.524 </loc>
</loc_list>
<face_list>
  <face v1="1" v2="2" v3="3" v4="4"/>      <!-- quadrangular facet -->
  <face v1="1" v2="5" v3="6"/>            <!-- triangular facet -->
</face_list>

```

9.1.3 Type Specification

Each connection is identified by its type. The XML definition of 2D connections contains the following nested elements:

Nested Elements	Multiplicity	Status	Constraint
adhesive_face	1	Optional	-

Table 134: Nested elements of <connection_2d/>

Note: Only *one* of the type elements <adhesive_face/> must exist in <connection_2d/>. If none of the type elements exist, then this will default to **adhesive_face**.

9.2 Adhesive Faces

A die-cut adhesive is denoted by an element `<adhesive_face/>`.



Figure 77: Picture of an adhesive face

XML specification of `<connection_2d/>` with attribute `label`:

Attributes	Type	Value Space	Status	Constraint / Remarks
label	Alphanumeric	Alphanumeric	Optional	-
quality_control	Alphanumeric	Alphanumeric	Optional	See section 6.4 Attribute <code>quality_control</code>

Table 135: Attributes of element `<connection_2d/>`

Nested Elements	Multiplicity	Status	Constraint / Remarks
adhesive_face	0-1	Optional	-
loc_list	1	Required	-
face_list	1	Required	-
appdata	0-1	Optional	-
custom_attributes_list	0-1	Optional	See section 6.5 Custom Attributes list

Table 136: Nested elements of element `<connection_2d/>`

For the `<adhesive_face/>` element, the following attributes can be specified:

Attributes	Type	Value space	Status	Constraint
base	Integer	> 0	Optional	-
thickness	Floating point	≥ 0.0	Optional	-
material	Alphanumeric	-	Optional	-

Table 137: Attributes of element `<adhesive_face/>`

- **base**: the index of the flange partner, on which the adhesive is applied to, before the flange partners are fitted together.
- **thickness**: denotes the thickness of the adhesive between the sheets.
- **material**: is an optional label that denotes which material is to be used for the adhesive.

Example (main type as <adhesive_face/>):

```
<connection_2d>
    <adhesive_face thickness="2.0" material="CAD_Material"/>
    <loc_list>
        <loc v="1"> 2001.557 14.435 1736.898 </loc>
        <loc v="2"> 1994.802 14.435 1734.247 </loc>
        <loc v="3"> 1994.790 0.0436 1734.256 </loc>
        <loc v="4"> 2001.547 0.0545 1736.911 </loc>
        <loc v="5"> 2008.298 14.435 1739.550 </loc>
        <loc v="6"> 2008.336 28.784 1739.524 </loc>
    </loc_list>
    <face_list>
        <face v1="1" v2="2" v3="3" v4="4"/>      <!-- quadrangular facet -->
        <face v1="1" v2="5" v3="6"/>            <!-- triangular facet -->
    </face_list>
    <appdata>
        ...
    </appdata>
    <custom_attributes_list>
        ...
    </custom_attributes_list>
</connection_2d>
```

10 Future extensions

So far, only the above mentioned connection types with the corresponding parameters are described, which cover mainly the applications of CAD and CAE. However, χ MCF is designed for the use in the complete development process and should be able to cover all major joint types thus two important extensions remain to be undertaken.

10.1 Additional parameters for spot and seam welds

For prototyping and manufacturing (CAM) additional parameters and information, like the type and the manufacturer of the welding device, the current density required in case of spot welds etc., may be relevant and needed. These parameters are not included in the present document yet. Their definitions will happen in the near future by the corresponding experts.

10.2 Other relevant and new joint types

It can be expected that increasingly new joint types will arise due to the advance of the technological development.

As mentioned before, χ MCF is open for any new joint type which will come and be of relevance for the technical application.

11 Disclaimer

This document contains, cites or refers to material which may be subject to trademark laws or copy right protection in one or more jurisdictions. All product and company names are trademarks™ or registered trademarks® of their respective holders. Their use does not imply any affiliation with or endorsement by them.

In especially, texts or graphics cited from Wikipedia are subject to their individual copy right (e. g. Creative Commons Attribution-Share alike 3.0 **Unported** License or GNU Free Documentation License).

Detailed information about the individual copy right are found at the respective URL, mentioned with the citation.

12 References

- [1] *Managing Connections using the Master Connection File*
Brian E. Huf, Ford Motor Company, **2001**
- [2] *Classification of Seam Welds.ppt*
Dr. Shicheng Zhang, Daimler AG, July **2005**
- [3] *Seam-Weld Types and Fatigue Relevant Parameter Sets for MCF Standard – Draft*
Gaier/Hofwimmer, Engineering Center Steyr GmbH & Co KG, July **2006**
- [4] *χMCF pilot in ANSA*
BETA CAE Systems S.A., February **2008**
- [5] *The Extended Master Connection File χMCF as a Transfer Standard of Seamweld Connection Definition (Slide set)*
Peter Mikolaj[†]/Carsten Franke, MSC Software/T-Systems, June 19th **2006**
- [6] *First Proposal for The Extended Master Connection File (χMCF) as a Transfer Standard of Seam-weld Connection Definition (Proposal for FAT AK 25 Fügetechnik), Revision 2,*
Peter Mikolaj[†], MSC.Software, Alzenau, June 27th **2006**
- [7] *FATXML-Format Version V1.2*
Schulte-Frankenfeld, VDA FAT-AK27, March **2016**
http://www.vda.de/de/publikationen/publikationen_downloads/index.html
- [8] FAT-AK25 Fügetechnik, "χMCF Extended Master Connection File: A Standard for Describing Connections and Joints in the Automotive Industry, Version 2.0", Berlin, September **2014**.
http://www.vda.de/de/publikationen/publikationen_downloads/index.html

Bisher in der FAT-Schriftenreihe erschienen (ab 2010)

Nr.	Titel
227	Schwingfestigkeitsbewertung von Nahtenden MSG-geschweißter Dünbleche aus Stahl, 2010
228	Systemmodellierung für Komponenten von Hybridfahrzeugen unter Berücksichtigung von Funktions- und EMV-Gesichtspunkten, 2010
229	Methodische und technische Aspekte einer Naturalistic Driving Study, 2010
230	Analyse der sekundären Gewichtseinsparung, 2010
231	Zuverlässigkeit von automotive embedded Systems, 2011
232	Erweiterung von Prozessgrenzen der Bonded Blank Technologie durch hydromechanische Umformung, 2011
233	Spezifische Anforderungen an das Heiz-Klimasystem elektromotorisch angetriebener Fahrzeuge, 2011
234	Konsistentes Materialmodell für Umwandlung und mechanische Eigenschaften beim Schweißen hochfester Mehrphasen-Stähle, 2011
235	Makrostrukturelle Änderungen des Straßenverkehrslärms, Auswirkung auf Lästigkeit und Leistung, 2011
236	Verbesserung der Crashsimulation von Kunststoffbauteilen durch Einbinden von Morphologiedaten aus der Spritzgießsimulation, 2011
237	Verbrauchsreduktion an Nutzfahrzeugkombinationen durch aerodynamische Maßnahmen, 2011
238	Wechselwirkungen zwischen Dieselmotortechnik und -emissionen mit dem Schwerpunkt auf Partikeln, 2012
239	Überlasten und ihre Auswirkungen auf die Betriebsfestigkeit widerstandspunktgeschweißter Feinblechstrukturen, 2012
240	Einsatz- und Marktpotenzial neuer verbrauchseffizienter Fahrzeugkonzepte, 2012
241	Aerodynamik von schweren Nutzfahrzeugen - Stand des Wissens, 2012
242	Nutzung des Leichtbaupotentials von höchstfesten Stahlfeinblechen durch die Berücksichtigung von Fertigungseinflüssen auf die Festigkeitseigenschaften, 2012
243	Aluminiumschaum für den Automobileinsatz, 2012
244	Beitrag zum Fortschritt im Automobilleichtbau durch belastungsgerechte Gestaltung und innovative Lösungen für lokale Verstärkungen von Fahrzeugstrukturen in Mischbauweise, 2012
245	Verkehrssicherheit von schwächeren Verkehrsteilnehmern im Zusammenhang mit dem geringen Geräuschniveau von Fahrzeugen mit alternativen Antrieben, 2012
246	Beitrag zum Fortschritt im Automobilleichtbau durch die Entwicklung von Crashabsorbern aus textilverstärkten Kunststoffen auf Basis geflochtener Preforms und deren Abbildung in der Simulation, 2013
247	Zuverlässige Wiederverwendung und abgesicherte Integration von Softwarekomponenten im Automobil, 2013
248	Modellierung des dynamischen Verhaltens von Komponenten im Bordnetz unter Berücksichtigung des EMV-Verhaltens im Hochvoltbereich, 2013
249	Hochspannungsverkopplung in elektronischen Komponenten und Steuergeräten, 2013
250	Schwingfestigkeitsbewertung von Nahtenden MSG-geschweißter Feinbleche aus Stahl unter Schubbeanspruchung, 2013

- 251 Parametrischer Bauraum – synchronisierter Fahrzeugentwurf, 2013
- 252 Reifenentwicklung unter aerodynamischen Aspekten, 2013
- 253 Einsatz- und Marktpotenzial neuer verbrauchseffizienter Fahrzeugkonzepte – Phase 2, 2013
- 254 Qualifizierung von Aluminiumwerkstoffen für korrosiv beanspruchte Fahrwerksbauteile unter zyklischer Belastung (Salzkorrosion), 2013
- 255 Untersuchung des Rollwiderstands von Nutzfahrzeugreifen auf echten Fahrbahnen, 2013
- 256 Naturalistic Driving Data, Re-Analyse von Daten aus dem EU-Projekt euroFOT, 2013
- 257 Ableitung eines messbaren Klimasummenmaßes für den Vergleich des Fahrzeugklimas konventioneller und elektrischer Fahrzeuge, 2013
- 258 Sensitivitätsanalyse rollwiderstandsrelevanter Einflussgrößen bei Nutzfahrzeugen, Teile 1 und 2, 2013
- 259 Erweiterung des Kerbspannungskonzepts auf Nahtübergänge von Linienschweißnähten an dünnen Blechen, 2013
- 260 Numerische Untersuchungen zur Aerodynamik von Nutzfahrzeugkombinationen bei realitätsnahen Fahrbedingungen unter Seitenwindeinfluss, 2013
- 261 Rechnerische und probandengestützte Untersuchung des Einflusses der Kontaktwärmeübertragung in Fahrzeugsitzen auf die thermische Behaglichkeit, 2013
- 262 Modellierung der Auswirkungen verkehrsbedingter Partikelanzahl-Emissionen auf die Luftqualität für eine typische Hauptverkehrsstraße, 2013
- 263 Laserstrahlschweißen von Stahl an Aluminium mittels spektroskopischer Kontrolle der Einschweißtiefe und erhöhter Anbindungsbreite durch zweidimensional ausgeprägte Schweißnähte, 2014
- 264 Entwicklung von Methoden zur zuverlässigen Metamodellierung von CAE Simulations-Modellen, 2014
- 265 Auswirkungen alternativer Antriebskonzepte auf die Fahrdynamik von PKW, 2014
- 266 Entwicklung einer numerischen Methode zur Berücksichtigung stochastischer Effekte für die Crash-simulation von Punktschweißverbindungen, 2014
- 267 Bewegungsverhalten von Fußgängern im Straßenverkehr - Teil 1, 2014
- 268 Bewegungsverhalten von Fußgängern im Straßenverkehr - Teil 2, 2014
- 269 Schwingfestigkeitsbewertung von Schweißnahtenden MSG-geschweißter Feinblechstrukturen aus Aluminium, 2014
- 270 Physiologische Effekte bei PWM-gesteuerter LED-Beleuchtung im Automobil, 2015
- 271 Auskunft über verfügbare Parkplätze in Städten, 2015
- 272 Zusammenhang zwischen lokalem und globalem Behaglichkeitsempfinden: Untersuchung des Kombinationseffektes von Sitzheizung und Strahlungswärmeübertragung zur energieeffizienten Fahrzeugklimatisierung, 2015
- 273 UmCra - Werkstoffmodelle und Kennwertermittlung für die industrielle Anwendung der Umform- und Crash-Simulation unter Berücksichtigung der mechanischen und thermischen Vorgeschichte bei hochfesten Stählen, 2015
- 274 Exemplary development & validation of a practical specification language for semantic interfaces of automotive software components, 2015
- 275 Hochrechnung von GIDAS auf das Unfallgeschehen in Deutschland, 2015
- 276 Literaturanalyse und Methodenauswahl zur Gestaltung von Systemen zum hochautomatisierten Fahren, 2015
- 277 Modellierung der Einflüsse von Porenmorphologie auf das Versagensverhalten von Al-Druckgussteilen mit stochastischem Aspekt für durchgängige Simulation von Gießen bis Crash, 2015

- 278 Wahrnehmung und Bewertung von Fahrzeugaußengeräuschen durch Fußgänger in verschiedenen Verkehrssituationen und unterschiedlichen Betriebszuständen, 2015
- 279 Sensitivitätsanalyse rollwiderstandsrelevanter Einflussgrößen bei Nutzfahrzeugen – Teil 3, 2015
- 280 PCM from iGLAD database, 2015
- 281 Schwere Nutzfahrzeugkonfigurationen unter Einfluss realitätsnaher Anströmbedingungen, 2015
- 282 Studie zur Wirkung niederfrequenter magnetischer Felder in der Umwelt auf medizinische Implantate, 2015
- 283 Verformungs- und Versagensverhalten von Stählen für den Automobilbau unter crashartiger mehrachsiger Belastung, 2016
- 284 Entwicklung einer Methode zur Crashsimulation von langfaserverstärkten Thermoplast (LFT) Bauteilen auf Basis der Faserorientierung aus der Formfüllsimulation, 2016
- 285 Untersuchung des Rollwiderstands von Nutzfahrzeugreifen auf realer Fahrbahn, 2016
- 286 xMCF - A Standard for Describing Connections and Joints in the Automotive Industry, 2016

Impressum

Herausgeber	FAT Forschungsvereinigung Automobiltechnik e.V. Behrenstraße 35 10117 Berlin Telefon +49 30 897842-0 Fax +49 30 897842-600 www.vda-fat.de
ISSN	2192-7863
Copyright	Forschungsvereinigung Automobiltechnik e.V. (FAT) 2016

VDA | Verband der
Automobilindustrie

FAT | Forschungsvereinigung
Automobiltechnik

Behrenstraße 35
10117 Berlin
www.vda.de
www.vda-fat.de