

## **χMCF (xMCF) — A Description and Data Standard for Connection and Joining Data in Structural Systems**

# CD stage

### **Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

## Contents

Foreword.....	xiii
Introduction.....	xiv
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions .....	1
4 Design Principles and Basic Features of $\chi$ MCF .....	1
4.1 Introduction.....	1
4.2 Design Principles .....	1
4.3 Idealization of Joints.....	2
4.4 Reconstruction of Joints from $\chi$ MCF.....	3
4.5 Description of Topology .....	3
4.6 $\chi$ MCF in the Development Processes .....	4
5 Keywords of XML specification .....	5
5.1 Keywords .....	5
6 Parts, Properties and Assemblies .....	6
6.1 General.....	6
6.2 Parts.....	6
6.2.1 General .....	6
6.2.2 Part Labels.....	7
6.2.3 Part Instances .....	7
6.3 Properties.....	7
6.4 Assemblies.....	7
7 File Structure of $\chi$ MCF .....	8
7.1 General.....	8
7.2 Elements containing general information .....	8
7.2.1 General .....	8
7.2.2 Date.....	9
7.2.3 Time.....	9
7.2.4 Version .....	9
7.2.5 Unit System .....	10
7.3 Application, User and Process Specific Data .....	10
7.3.1 General .....	10
7.3.2 User Specific Data <appdata/>.....	11
7.3.3 Finite Element Specific Data <femdata/>.....	12
7.4 Connection Data <connection_group/> .....	13
7.4.1 General .....	13
7.4.2 Connected Objects .....	14
7.4.3 Contacts and Friction .....	19
7.4.4 Joints.....	21
7.5 A Minimalistic Example of a $\chi$ MCF file.....	22
7.6 XML Schema Definition .....	23
8 Data Common to any Connection .....	23
8.1 Indices and their properties .....	23
8.2 Connection Referencing .....	23
8.2.1 Need for Referencing.....	23
8.2.2 Attribute label.....	23
8.2.3 Attribute ident.....	24

8.3 Dimensions and Coordinates .....	24
8.4 Attribute <code>quality_control</code> .....	24
8.5 Custom Attributes list .....	24
8.6 Distinction between <code>&lt;custom_attributes/&gt;</code> and <code>&lt;appdata/&gt;</code> .....	29
8.6.1 General .....	29
8.6.2 Needs of different process roles, addressed by <code>&lt;custom_attributes/&gt;</code> and <code>&lt;appdata/&gt;</code> .....	29
8.6.3 Needs of different applications, addressed by <code>&lt;custom_attributes/&gt;</code> and <code>&lt;appdata/&gt;</code> .....	29
8.6.4 Different levels of <code>&lt;custom_attributes/&gt;</code> and <code>&lt;appdata/&gt;</code> within $\chi$ MCF data model .....	29
9 OD connections .....	30
9.1 Generic Definitions .....	30
9.1.1 Identification .....	30
9.1.2 Location .....	31
9.1.3 Direction .....	32
9.1.4 Type Specification .....	33
9.2 Spot Welds .....	33
9.2.1 General .....	33
9.2.2 Attribute "diameter" .....	34
9.2.3 Attribute "technology" .....	34
9.3 Robscans .....	34
9.4 Rivets .....	37
9.4.1 General .....	37
9.4.2 Blind Rivets .....	39
9.4.3 Self-Piercing Rivets .....	41
9.4.4 Solid Rivets .....	43
9.4.5 Swop Rivets .....	45
9.4.6 Clinch Rivet Studs .....	46
9.5 Threaded Connections: Bolts and Screws .....	47
9.5.1 Introduction .....	47
9.5.2 Contacts and Friction .....	49
9.5.3 Definition of element <code>&lt;threaded_connection/&gt;</code> .....	51
9.5.4 Washer .....	54
9.5.5 Nut .....	55
9.5.6 Bolt .....	56
9.5.7 Screw .....	61
9.6 Gum Drops .....	64
9.7 Clinches .....	65
9.8 Heat Stakes / Thermal Stakes .....	68
9.9 Clips/Snap Joints .....	70
9.10 Nails .....	73
9.11 Rotation Joints .....	75
9.11.1 General .....	75
9.11.2 ROTAV .....	76
10 1D connections .....	78
10.1 Generic Definitions .....	78
10.1.1 Identification .....	78
10.1.2 Location .....	78
10.1.3 Intermittent Connection Lines .....	79
10.1.4 Type Specification .....	85
10.2 Seam Welds .....	85
10.2.1 Description and Modelling Parameters .....	85

10.2.2 Seam Weld Definition Overview.....	86
10.2.3 Specific XML Realization .....	88
10.2.4 Generic Seam Weld Definition .....	88
10.2.5 Butt Joint.....	97
10.2.6 Corner Weld.....	100
10.2.7 Edge Weld.....	106
10.2.8 I-Weld.....	108
10.2.9 Overlap Weld.....	111
10.2.10 Y-Joint.....	117
10.2.11 K-Joint.....	121
10.2.12 Cruciform Joint.....	126
10.2.13 Flared Joint.....	130
10.3 Adhesive Lines.....	133
10.3.1 Element "adhesive_line" .....	133
10.3.2 Element "loc_list" .....	133
10.3.3 Element "appdata" .....	133
10.3.4 Element "femdata" .....	134
10.4 Hemming Flanges .....	134
10.4.1 General .....	134
10.4.2 Element <hemming/> .....	136
10.4.3 Element "loc_list" .....	136
10.4.4 Element "appdata" .....	136
10.4.5 Element "femdata" .....	136
10.4.6 Element "hemming" .....	137
10.5 Sequence Connections .....	139
11 2D connections.....	141
11.1 Generic Definitions .....	141
11.1.1 Identification.....	141
11.1.2 Connection Face .....	141
11.1.3 Type Specification .....	143
11.2 Adhesive Faces.....	144
12 Future extensions .....	145
12.1 General .....	145
12.2 Additional parameters for spot and seam welds .....	145
12.3 Other relevant and new joint types.....	145
Annex A (informative) Derivation of Formulae used for Regular Intermittent Welds .....	146
Annex B (informative) Federative use of $\chi$ MCF with ISO 10303-242 .....	148
B.1 General principles .....	148
B.2 Cross-References between ISO 10303-242 and $\chi$ MCF .....	149
Annex C (informative) History .....	151
Bibliography .....	152

## List of Figures

Figure 1 — Seam weld as 1-dimensional joint .....	2
Figure 2 — Topological Relations between Parts and Assemblies .....	3

Figure 3 — Product Structures Fitting to Previous Figure .....	3
Figure 4 — The Development Process .....	4
Figure 5 — $\chi$ MCF as a Platform for Connection Data in the Complete Development Process ..	4
Figure 6 — Weld line crossing tailored blank vs. weld line crossing physical gap.....	8
Figure 7 — Special Stacking Topologies.....	17
Figure 8 — Robscans with Different Rotation Angles; Two of them Mirrored.....	35
Figure 9 — Rivet head types (Dome, Large Flange, Countersunk) .....	38
Figure 10 — Blind rivet – key attributes.....	40
Figure 11 — Assembly Recommendations for Blind Rivets.....	41
Figure 12 — Cross section of a self-piercing rivet & riveting machine .....	41
Figure 13 — Pictures of characteristic rivet types before and after mounting. ....	43
Figure 14 — Key dimensions of solid rivets .....	43
Figure 15 — Relation of working thickness (T1+T2) to max and min values of grip .....	44
Figure 16 — Cross section of a SWOP Rivet .....	45
Figure 17 — Clinch Rivet Studs: Threaded variant and Ball stud .....	47
Figure 18 — Bolts and Screws.....	48
Figure 19 — Different Screw Forms.....	48
Figure 20 — Definition of Length and Head Sizes.....	49
Figure 21 — Definition of lead, pitch and starts of a thread.....	49
Figure 22 — Bolt with welded nut.....	59
Figure 23 — Bolt with free nut .....	60
Figure 24 — Screw .....	60
Figure 25 — Welded stud with free nut .....	60
Figure 26 — Plain stud .....	61
Figure 27 — Process of Flow Drill Screwing.....	62
Figure 28 — Measures of applied FDS.....	63
Figure 29 — FDS connection with pre-machined clearance hole .....	63
Figure 30 — Pilot hole on sheet metal .....	64
Figure 31 — Clinch Joint Dimensions.....	66
Figure 32 — Two example clinch systems [5] (TOX (left) and BTM's Tog-L-Loc system).....	66
Figure 33 — Heat Stakes: Process steps and Design dimensions .....	68
Figure 34 — A "Hairpin Clip" .....	70
Figure 35 — Internal and External Circlips .....	71
Figure 36 — Clips Pushed into a Hole .....	71
Figure 37 — Clips Sliding onto a Flat Surface.....	71
Figure 38 — Cross Section of a nail joint connecting two Sheets .....	73
Figure 39 — Key measures of a nail & examples of different nail types .....	74

Figure 40 — Process of Rotation Joining (ROTAV) [7] .....	76
Figure 41 — ROTAV connecting aluminium and steel sheets [7] .....	76
Figure 42 — Terminology of a regular intermittent weld .....	80
Figure 43 — Regular intermittent weld with first spacing and last spacing.....	80
Figure 44 — Irregular intermittent welds.....	81
Figure 45 — <corner_weld/> with <regular_segments/> and “Required” attributes only ..	83
Figure 46 — Regular single sided welding (a <corner_weld/> with <regular_segments/> and all attributes).....	84
Figure 47 — Staggered welding.....	84
Figure 48 — Definition of a <corner_weld/> with <segment_list/> .....	85
Figure 49 — Seam weld types and attributes .....	87
Figure 50 — $\chi$ MCF Structure of a Seam Weld (connection_1d) .....	88
Figure 51 — Sheet Parameters vs. Weld Position Parameters.....	91
Figure 52 — Welding Position of a Y-Joint.....	93
Figure 53 — Welding Position vector direction and length .....	94
Figure 54— Butt Joint Sheet Layout .....	97
Figure 55— Butt Joint Weld parameters.....	98
Figure 56— Corner Weld Sheet Layout.....	101
Figure 57— Corner Weld Parameters .....	101
Figure 58 — Corner Weld Sheet Layout.....	102
Figure 59 — Double Corner Weld Parameters.....	102
Figure 60 — Edge Weld Sheet Layout .....	106
Figure 61 — Edge Weld parameters .....	106
Figure 62 — I-Weld Sheet Layout.....	109
Figure 63 — I-Weld Parameters .....	109
Figure 64 — Overlap Weld Sheet Layout.....	112
Figure 65 — Overlap Weld Parameters .....	112
Figure 66 — Single Sided Double Overlap Weld.....	113
Figure 67 — Overlap Weld Parameter Details for lower (left) and upper (right) Weld Section .....	113
Figure 68 — Double-Sided Double Overlap Weld .....	114
Figure 69 — Parameters of Double-Sided Double Overlap Weld (left side: upper section; right side: lower section) .....	114
Figure 70 — Y-Joint Sheet Layout.....	118
Figure 71 — Parameters of Y-Joint Weld.....	118
Figure 72 — K-Joint Sheet Layout .....	122
Figure 73 — Parameters of K-Joint Weld .....	122
Figure 74 — Cruciform Joint Sheet Layout .....	126

Figure 75 — Parameters of Cruciform Joint .....	126
Figure 76 — Flared Joint Sheet Layout .....	130
Figure 77 — Flared Joint Sheet Layout .....	131
Figure 78 — The Three Regions of a Hemming .....	134
Figure 79 — Path Changes and Width Changes in Hemming Flanges .....	135
Figure 80 — Adhesive Path Differs from Root Path .....	135
Figure 81 — Reinforcements need to be considered as Part of the Inner Panel .....	136
Figure 82 — Sequence without margin .....	139
Figure 83 — Sequence with margin and spacing .....	139
Figure 84 — Margin relaxation .....	139
Figure 85 — Spacing relaxation .....	139
Figure 86 — Picture of a sealing or adhesive face .....	144
Figure A. 1 — 'length', 'spacing', 'first_spacing' and 'last_spacing' are the terms needed to define a regular intermittent weld. ....	146
Figure A. 2 — A regular intermittent weld with 'n' segments and 'n-1' spacings between segments.....	146
Figure B. 1— References in STEP file to related $\chi$ MCF file .....	149

## List of Tables

Table 1 — Nested elements of element <code>&lt;xmcf/&gt;</code> .....	9
Table 2 — XML-specification of <code>&lt;units/&gt;</code> .....	10
Table 3 — Nested elements of the child element of <code>&lt;femdata/&gt;</code> .....	12
Table 4 — Attributes of element <code>&lt;connection_group/&gt;</code> .....	13
Table 5 — Nested elements of element <code>&lt;connection_group/&gt;</code> .....	14
Table 6 — Nested elements of <code>&lt;connected_to/&gt;</code> .....	14
Table 7 — Attributes of element <code>&lt;part/&gt;</code> .....	15
Table 8 — Attributes of element <code>&lt;assy/&gt;</code> .....	15
Table 9 — Nested elements of <code>&lt;stacking/&gt;</code> .....	18
Table 10 — Attributes of <code>&lt;stacking&gt;</code> .....	18
Table 11 — Attributes of <code>&lt;level&gt;</code> .....	18
Table 12 — Nested elements of element <code>&lt;contact_list/&gt;</code> .....	20
Table 13 — Nested elements of element <code>&lt;contact/&gt;</code> .....	20
Table 14 — Attributes of element <code>&lt;partner/&gt;</code> .....	20



Table 15 — Attributes of element <coefficients/>.....	21
Table 16 — Nested elements of element <connection_list/>.....	22
Table 17 — Nested elements of element <custom_attributes_list/>.....	25
Table 18 — Attributes of <custom_attributes/> element .....	26
Table 19 — Nested elements of element <custom_attributes/>.....	26
Table 20 — Attributes of <string/> element.....	26
Table 21 — Attributes of <real/> element.....	26
Table 22 — Attributes of <integer/> element.....	27
Table 23 — Attributes of <string_list/> element.....	27
Table 24 — Nested elements of <string_list/> element.....	27
Table 25 — Attributes of <value/> element inside <string_list/>.....	27
Table 26 — Attributes of <real_list/> element .....	27
Table 27— Nested element of <real_list/> element.....	27
Table 28 — Attributes of <value> element inside <real_list/> .....	27
Table 29 — Attributes of <int_list/> element.....	28
Table 30 — Nested elements of <int_list/> element .....	28
Table 31 — Attributes of <value/> element inside <real_list/> .....	28
Table 32 — Attributes of element <connection_0d/>.....	30
Table 33 — Text values of element <loc/>.....	31
Table 34 — Attributes of elements <normal_direction/> & <tangential_direction/>.....	32
Table 35 — Nested elements of element <connection_0d/>.....	33
Table 36 — Nested elements of <connection_0d/> for <spotweld/> .....	33
Table 37 — Attributes of element <spotweld/>.....	33
Table 38 — Nested elements of element <spotweld/> .....	34
Table 39 — Nested elements of <connection_0d/> for <robscan/>.....	35
Table 40 — Attributes of element <robscan/>.....	36
Table 41 — Nested elements of element <robscan/>.....	37
Table 42 — Nested elements of <connection_0d/> for <rivet/>.....	37
Table 43 — Attributes of element <rivet/>.....	38
Table 44 — Nested elements of element <rivet/>.....	39
Table 45 — Attributes of element <blind/>.....	40
Table 46 — Attributes of element <self_piercing/>.....	42
Table 47 — Attributes of element <solid/>.....	44
Table 48 — Attributes of element <swop/>.....	46
Table 49 — Attributes of element <clinch_rivet_stud/> .....	47

Table 50 — Nested elements of <connection_0d/> for <threaded_connection/>.....	52
Table 51 — Attributes of element <threaded_connection/>.....	52
Table 52 — Nested elements of element <threaded_connection/>.....	54
Table 53 — Attributes of element <washer/>.....	54
Table 54 — Attributes of element <nut/>.....	55
Table 55 — Nested elements of element <nut/>.....	56
Table 56 — Attributes of element <bolt/>.....	56
Table 57 — Nested elements of element <bolt/> .....	57
Table 58 — Attributes of element <screw/> .....	61
Table 59 — Nested elements of element <screw/>.....	61
Table 60 — Attributes of element <flow_drilled/>.....	63
Table 61 — Nested elements of <connection_0d> for <gumdrop/>.....	64
Table 62 — Attributes of element <gumdrop/>.....	65
Table 63 — Nested elements of element <gumdrop/>.....	65
Table 64 — Nested elements of <connection_0d/> for <clinch/>.....	66
Table 65 — Attributes of element <clinch/>.....	67
Table 66 — Nested elements of element <clinch/> .....	68
Table 67 — Nested elements of <connection_0d/> for <heat_stake/> .....	69
Table 68 — Attributes of element <heat_stake/> .....	69
Table 69 — Nested elements of element <heat_stake/>.....	70
Table 70 — Nested elements of <connection_0d/> for <clip/>.....	71
Table 71 — Attributes of element <clip/>.....	71
Table 72 — Nested elements of element <clip/> .....	72
Table 73 — Nested elements of <connection_0d/> for <nail/>.....	73
Table 74 — Attributes of element <nail/>.....	73
Table 75 — Nested elements of element <nail/> .....	75
Table 76 — Nested elements of <connection_0d/> for <rotation_joint/>.....	75
Table 77 — Attributes of element <rotation_joint/>.....	75
Table 78 — Nested elements of element <rotation_joint/> .....	76
Table 79 — Attributes of element <rotav/> .....	77
Table 80 — Attributes of element <loc_list/>.....	78
Table 81 — Nested elements of <loc_list/> .....	78
Table 82 — Attributes of element <loc/>.....	79
Table 83 — Attributes of element <segment/>.....	81
Table 84 — Attributes of element <regular_segments/>.....	81

Table 85 — Nested elements of element <connection_1d/> .....	85
Table 86 — Nested elements of element <seamweld/> .....	88
Table 87 — Attributes of element <subtype/>.....	89
Table 88 — Nested elements of element <subtype/>.....	90
Table 89 — Attributes of element <sheet_parameter/>.....	91
Table 90 — Nested elements of element <subtype/>.....	93
Table 91 — Attributes of element <weld_position/>.....	93
Table 92 — Default values of attribute "filler", dependent from attribute "technology" .....	96
Table 93 — Parameters of Butt Joint Weld.....	98
Table 94 — Attributes of element <weld_position/> for Butt Joint.....	99
Table 95 — Attributes of element <sheet_parameter/> for Butt Joint.....	100
Table 96 — Parameters of Simple Corner Weld .....	102
Table 97 — Parameters of Double Corner Weld.....	103
Table 98 — Attributes of element <weld_position/> for Corner Weld.....	103
Table 99 — Values of Attribute section.....	104
Table 100 — Values of Attribute angle .....	104
Table 101 — Attributes of element <sheet_parameter/> for Corner Weld.....	105
Table 102 — Parameters of Edge Weld.....	106
Table 103 — Attributes of element <weld_position/> for Edge Weld .....	107
Table 104 — Attributes of element <sheet_parameter/> for Corner Weld.....	108
Table 105 — Parameters of I-Weld .....	109
Table 106 — Attributes of element <weld_position/> for I Weld.....	110
Table 107 — Attributes of element <sheet_parameter/> for I Weld .....	111
Table 108 — Parameters of Overlap Weld .....	112
Table 109 — Parameters of Single Sided Double Overlap Weld.....	114
Table 110 — Parameters of Double-Sided Double Overlap Weld.....	115
Table 111 — Attributes of element <weld_position/> for Overlap Weld .....	115
Table 112 — Attributes of element <sheet_parameter/> for Overlap Weld.....	117
Table 113 — Parameters of Y-Joint.....	118
Table 114 — Attributes of element <weld_position/> for Y Joint .....	119
Table 115 — Value Dependency of Attribute thickness.....	120
Table 116 — Attributes of element <sheet_parameter/> for Y-Joint.....	121
Table 117 — Parameters of K-Joint.....	122
Table 118 — Attributes of element <weld_position/> for K Joint.....	123
Table 119 — Value Dependency of Attribute thickness.....	124
Table 120 — Attributes of element <sheet_parameter/> for K Joint.....	125

Table 121 — Parameters of Cruciform Joint .....	127
Table 122 — Attributes of element <weld_position/> for Cruciform Joint .....	127
Table 123 — Value Dependency of Attribute thickness .....	128
Table 124 — Attributes of element <sheet_parameter/> for Cruciform Joint.....	130
Table 125 — Parameters of Flared joint.....	131
Table 126 — Attributes of element <weld_position/> for Flared Joint .....	131
Table 127 — Attributes of element <sheet_parameter/> for Flared Joint .....	132
Table 128 — Nested elements of <connection_1d/>.....	133
Table 129 — Attributes of element <adhesive_line/>.....	133
Table 130 — Nested elements of <connection_1d/> for <hemming/>.....	136
Table 131 — Attributes of element <hemming/>.....	137
Table 132 — Nested elements of element <hemming/>.....	137
Table 133 — Attributes of element <region/>.....	137
Table 134 — Nested elements of element <region/>.....	138
Table 135 — Nested elements of <connection_1d/> for <sequence_connection_0d/> .....	141
Table 136 — Nested elements of <sequence_connection_0d/>.....	141
Table 137 — Attributes of element <sequence_connection_0d/> .....	141
Table 138 — Nested elements of <loc_list/>.....	142
Table 139 — Attributes of element <loc/>.....	142
Table 140 — Nested elements of element <face_list/>.....	143
Table 141 — Attributes of element <face/> .....	143
Table 142 — Nested elements of <connection_2d/>.....	143
Table 143 — Nested elements of element <connection_2d/> .....	144
Table 144 — Attributes of element <adhesive_face/>.....	144
 Table B. 1 — Cross-Reference Table between ISO 10303-242 and $\chi$ MCF .....	 150

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at [www.iso.org/patents](http://www.iso.org/patents). ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by the *German Association of the Automotive Industry (VDA), FAT-AK25 Fügetechnik* and was adopted by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

## Introduction

ISO 8329 ( $\chi$ MCF) aims at describing connections or joints related to mechanical systems or structures. The demand for such a standard has grown from the observation that modern PLM systems - while working well with part information, like geometry, material, weight etc. - are lacking a consistent handling of logical and process related connection information (like parts being connected, orientation of point connections, assembly process parameter etc.).

PLM workstreams need to include connection data to automate development processes and enable seamless data flows between engineering functions.  $\chi$ MCF is intended to be the “language” that is understood and used by the various tools to exchange connection data along the development chain.

The initial motivation to develop this standard has come from the automotive industry (Annex C). However, there is no element in the standard that limits it to this industry. It is clearly targeted to support virtual development processes for mechanical systems or structures in any industrial area.

Regardless of the actual industry, complex technical systems such as vehicles, planes, ships etc. typically consist of thousands of individual parts which are assembled by joints. Depending on the involved materials and the manufacturing processes, a wide range of joining types are used within an individual technical structure or system. It ranges from welds, bolt connections, adhesives, rivets, clips, etc. The efficient and reliable data management of this connection data is not only required for the actual design & verification process (CAD & CAE) but also for manufacturing planning and even cost estimations. Various design, material and manufacturing parameters are required to be managed for each connection.

The available details for connections or joints grow along the development process. At different stages (concept phase, detailed design, verification, manufacturing planning, ...) and in different functions (CAD, CAE, Manufacturing, ...) data will be added and consumed. So, a database for connection data is required. But also, the tools adding or extracting data need to understand the data structure and use a common description language.  $\chi$ MCF, defined in this document, serves as this language.

The advantage is evident, integrating connection data into the PLM structure and using a common language ( $\chi$ MCF) for data exchange, avoid data conversions or re-generations and, hence, avoid inconsistencies and flaws during system development.

# χMCF (xMCF) - Extended Master Connection File

(Standard for Describing Connections and Joints in Structural Systems)

## 1 Scope

This document specifies XML definitions that are being used to describe data and information related to connections or joints in mechanical systems or structures.

The following is within the scope of this document:

- Description & explanation of XML definitions for logical or process related data or other properties of a connection.

The following are outside the scope of this document:

- Geometry of fasteners or other parts;
- Handling of χMCF data in PDM-, SDM- and other data management systems.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- Not applicable. —

## 3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>;
- IEC Electropedia: available at <https://www.electropedia.org/>.

## 4 Design Principles and Basic Features of χMCF

### 4.1 Introduction

The Extended Master Connection File (χMCF) is a container for connection information of a complex structure. A complex structure consists of individual parts which are joined together. Connections establish a topology between the parts. Therefore, a database or container designed to gather connection information should be equipped with data structures which reflect this topology between the parts.

χMCF is intended to define an industry standard for the exchange of connection data between different CAx tools along development process steps. Design principles for χMCF are required to keep the standard as lean as possible on one hand, but also enable use case dependent adjustments.

This section explains the design principles and basic features of χMCF.

### 4.2 Design Principles

The design of χMCF is guided by the following principles:

- 1) χMCF should be able to *completely* and *unambiguously* describe all relevant connections/joints used in the automotive or other industries. This includes amongst others spot welds, seam welds, rivets, adhesives;

- 2)  $\chi$ MCF should be able to address all kind of CAx processes;
- 3)  $\chi$ MCF contains *only* information relevant to connections. Hierarchical product structure, assembly sequence, part variants etc. are *not* subject of  $\chi$ MCF. Such kind of information needs different vessels for propagation. However,  $\chi$ MCF may *refer* to such "external" information, for example part codes. This principle provides the flexibility to use  $\chi$ MCF in any development process variant, established at different companies;
- 4)  $\chi$ MCF has to be flexible and easy to extend to any future joint types and applications;
- 5)  $\chi$ MCF is built upon the industry standard XML;
- 6) Connection data in  $\chi$ MCF have to be unique;
- 7) The content of  $\chi$ 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)  $\chi$ MCF follows the max-min principle: it contains information as much as necessary, at the same time, as little as possible;
- 9)  $\chi$ MCF shall enable a reconstruction of connectors at any certain stage in the involved processes without loss of data or ambiguities;
- 10) The  $\chi$ MCF format description is kept compact. Elements are reused, whenever possible;
- 11)  $\chi$ MCF offers "empty" containers which can be assigned to any certain connector, to a collection of connectors or even to the complete file. This allows to incorporate application specific data before or without standardization;
- 12)  $\chi$ MCF forms a good candidate for long-term archiving connector information due to its simplicity and extensibility.

XML has been selected since it is by itself an industry standard and human readable. XML allows to build a data structure which describes the connection topology of complex structures like automobiles or planes.

### 4.3 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  $\chi$ 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  $\chi$ MCF.



**Figure 1 — Seam weld as 1-dimensional joint**

A spot weld is treated as a 0-dimensional joint in  $\chi$ 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 can be modelled as 2-dimensional surfaces.



#### 4.4 Reconstruction of Joints from $\chi$ 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  $\mathbf{x}$  and the diameter  $d$ , plus the sheet thicknesses which by themselves are not a constituent of  $\chi$ MCF (recall  $\chi$ MCF contains only information relevant to joints), but of the corresponding CAD or CAE model.

#### 4.5 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. There are many ways to describe the topological relations. The following example demonstrates the way  $\chi$ MCF adopted to describe the topology:

- Part (or Assembly) A is joined to Part B by the seam weld 1 along the curve  $l_1$  and the spot welds at positions  $x_i$  and;
- Part (or Assembly) A is connected to Part C by the adhesive  $AD_x$  in the area  $A_x$ , et cetera.

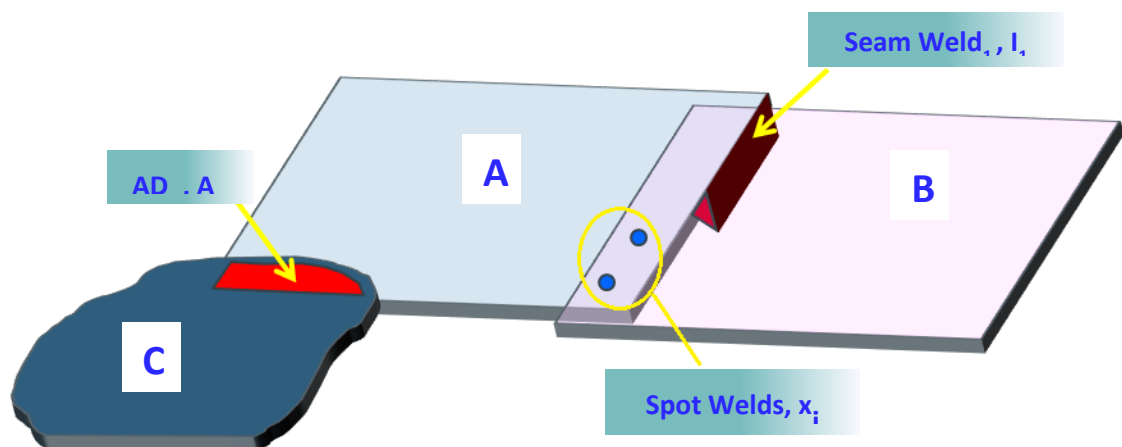


Figure 2 — Topological Relations between Parts and Assemblies

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.

Frequently, more than two parts are joined. A spot weld can, for instance, join three sheets, a screw even more. Such situations have to be considered, too.

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

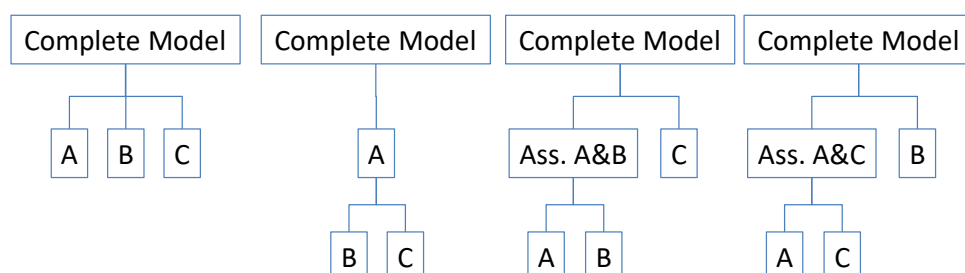


Figure 3 — Product Structures Fitting to Previous Figure.

And this list is far from being complete.

#### 4.6 $\chi$ 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  $\chi$ MCF.



Figure 4 — The Development Process

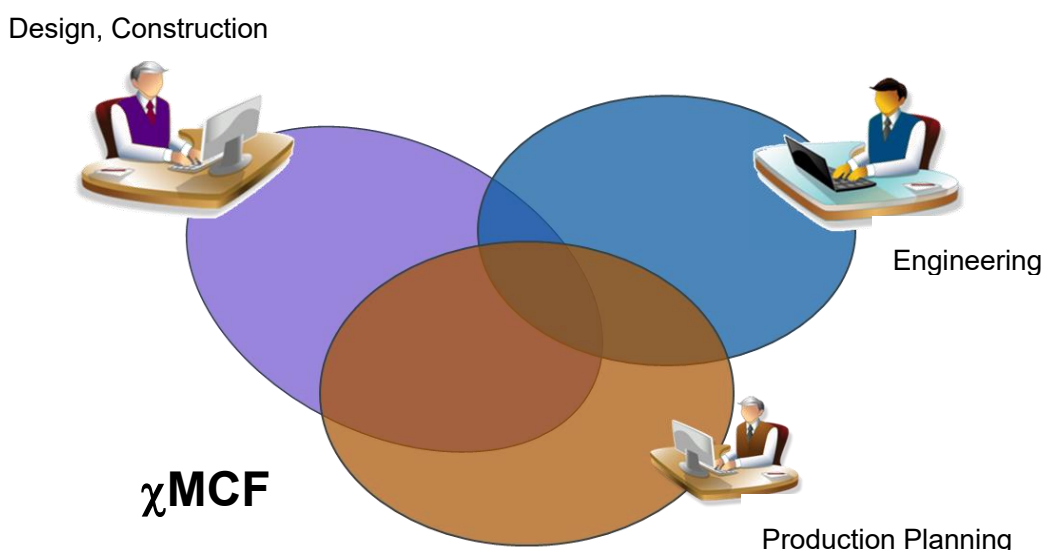


Figure 5 —  $\chi$ MCF as a Platform for Connection Data in the Complete Development Process

A careful examination of Figure 5 enables one to get more feeling and understanding about how the work with  $\chi$ MCF in a real process could look like:  $\chi$ MCF 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, for instance 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  $\chi$ MCF 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,  $\chi$ MCF is an ideal tool to enable this dynamic process since filling up  $\chi$ MCF 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  $\chi$ MCF. Thus, unnecessary duplication of effort is avoided automatically. Typically, different parties work in different environments using different software tools. Provided all systems

support  $\chi$ MCF, transfer of data from one format to another will be not necessary anymore. This will save development cost and avoid loss of data caused by the transfer.

Information contained in  $\chi$ MCF 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  $\chi$ MCF;
- Automatic Programming of Welding Robot  
Based on  $\chi$ MCF, welding robots can be programmed automatically.

An essential feature of  $\chi$ 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  $\chi$ MCF into any real process. Depending on the application, it is possible to use  $\chi$ MCF as a stand-alone database or integrate  $\chi$ MCF into an even more comprehensive database.

## 5 Keywords of XML specification

### 5.1 Keywords

The carrier of information in a  $\chi$ MCF 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 (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 1.0) the following keywords 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).

NOTE 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>.)

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 for instance is 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 values of the ID (type integer) have to be 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 enumeration (a finite set), or an explicitly defined set. For example, 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  $\chi$ MCF 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 "[-.\$#\pm]" 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, 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`, that is in the form:  $\text{minOccurs} \leq \text{Multiplicity} \leq \text{maxOccurs}$ . By convention, when `Use` is optional, `minOccurs` is 0. Any additional restrictions imposed on an element, or an attribute are specified by the key-word `Restrictions`.

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). For instance, the label and pid of a part or an assembly represent the same part (except for e. g. tailored blanks) and one can use one or the other or both to identify a part.

## 6 Parts, Properties and Assemblies

### 6.1 General

$\chi$ MCF 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.

### 6.2 Parts

#### 6.2.1 General

Parts are logical groupings of 3D objects. Their objective is to provide a general nomenclature of the pieces which form a certain product. This nomenclature allows communications between all stake holders 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): For instance, a car is made of body in white, power train, doors 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.

### 6.2.2 Part Labels

A part is uniquely identified by its *label*, up to ditto-parts. Connectors within a connection group that refers to ditto parts shall 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 have two string attributes: One label describing the name and 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, when we use the term "part label".

### 6.2.3 Part Instances

Instances of parts, also known as ditto-parts, typically have the same *label* as their "base" parts. Stating their *instance* makes such parts uniquely distinguishable, without resort to their geometrical location. Stating an *instance* without a *part* is meaningless, however.

## 6.3 Properties

In CAE, properties are a concept for assigning physical behaviour to several 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 the same physical behaviour (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  $\chi$ 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  $\chi$ 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 behaviour 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.

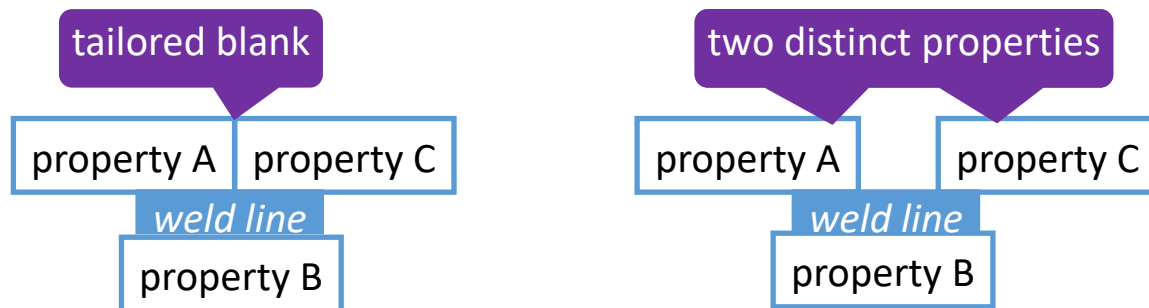
## 6.4 Assemblies

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  $\chi$ MCF, an assembly is just a set of parts, 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:  $\chi$ 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.

## 7 File Structure of $\chi$ MCF

### 7.1 General

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

The root/document element of  $\chi$ MCF is mandatorily named `<xmcf/>`. 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.

**NOTE** Since  $\chi$ MCF V2.0 introduced significant changes, the root element has been renamed from "mcf" to "xmcf" in order to avoid confusion with the "old" MCF-Format.

### 7.2 Elements containing general information

#### 7.2.1 General

$\chi$ 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 shown in Table 1.

**Table 1 — Nested elements of element <xmcf/>**

Nested Elements	Multiplicity	Use	Constraint / Remarks
date	1	Optional	-
version	1	Required	-
units	1	Optional	-
appdata	1-*	Optional	See section 7.3.2
femdata	1-*	Optional	See section 7.3.3
connection_group	1-*	Optional	See section 7.4

### 7.2.2 Date

The element <date/> of the format "yyyy-mm-dd" specifies the date on which the file was created. It follows ISO 8601 series, cf. [1].

#### EXAMPLE

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xmcf_3_1_1.xsd">
  <date> 2015-08-27 </date>
  <version> 3.1.1 </version>
  <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
  ...
</xmcf>
```

### 7.2.3 Time

The element <time/> of the format "hh:mm:ss±hh:mm" specifies the time on which the file was created. It follows ISO 8601 series, cf. [1].

Time element may exist only if date element exists.

#### EXAMPLE

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xmcf_3_1_1.xsd">
  <date> 2023-04-13 </date>
  <time> 15:34:05-01:00 </time>
  <version> 3.1.1 </version>
  <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
  ...
</xmcf>
```

### 7.2.4 Version

The version code of the  $\chi$ MCF standard upon which the current file is built shall be specified by the element <version/>.

The version code of  $\chi$ MCF files following this document is 3.1.1.

#### EXAMPLE

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xmcf_3_1_1.xsd">
```

```

<date> 2015-08-27 </date>
<time> 15:34:05-01:00 </time>
<version> 3.1.1 </version>
<units length="mm" angle="rad" mass="kg" force="N" time="s"/>
...
</xmcf>

```

## 7.2.5 Unit System

The unit system used by  $\chi$ MCF is based upon the International System of Units (SI) 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/>` are:

**Table 2 — 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"
angular_speed	Optional	"rad/s", "Hz", "kHz", "rpm"	"Hz"

## EXAMPLE

```

<?xml version="1.0" encoding="UTF-8" ?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xmcf_3_1_1.xsd">
  <date> 2015-08-27 </date>
  <version> 3.1.1 </version>
  <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
  ...
</xmcf>

```

## 7.3 Application, User and Process Specific Data

### 7.3.1 General

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

The current  $\chi$ 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  $\chi$ MCF standard;
- `<femdata/>`  
Contents is documented in FATXML [2] and hence does not need to be described, here.



### 7.3.2 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/> shall contain at least one nested element named after the application or user that is intended to interpret the data. In the examples A and B, the associated application is MEDINA, hence the nested element is <MEDINA/>.

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.

The user shall 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 χMCF file.

#### 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_3_1_1.xsd">
  <date> 2014-08-07 </date>
  <version> 3.1.1 </version>
  <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
  <appdata>
    <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>
  ...
</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_3_1_1.xsd">
  <date> 2014-08-07 </date>
  <version> 3.1.1 </version>
  <units length="mm" angle="rad" mass="kg" force="N" time="s"/>
  ...
  <connection_group id="1">
    <connected_to>
      ...
    </connected_to>
    <connection_list>
      <connection_1d>
        <loc_list>
          ...
        </loc_list>
      ...
    </connection_list>
  </connection_group>
  ...
</xmcf>
```

```

    <seamweld>
      ...
    </seamweld>
    <appdata>
      <MEDINA xmlns="http://servicenet.t-systems.com/medina/xMCF">
        <data_at_connector>
          ...
        </data_at_connector>
      </MEDINA>
    </appdata>
  </connection_id>
</connection_list>
</connection_group>
</xmcf>

```

### 7.3.3 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 connector (relevant tags are <connection\_0d/>, <connection\_1d/> and <connection\_2d/>).

χMCF versions 3.1 or later allow to contain <femdata/> at root level, but this is not allowed in V 3.0 and below. <femdata/> is not allowed on <connection\_group/> level in any case.

<femdata/> references FEM-entities that are related to the connector in which it is placed. Its content or the referenced entities respectively, are specific to a single solver.

Usually, referencing is done by solver specific entity IDs, which have no meaning outside the context of a specific finite element model. If, for example, element IDs in this model get renumbered, a χMCF file referencing such element IDs becomes detached and needs to be re-created.

**Conclusion:** A χMCF file containing <femdata/> always refers to *one specific* solver deck.

This solver naming should be taken from the current FATXML version. Examples are the following:

- PAM-CRASH;
- LS-DYNA;
- RADIOSS;
- OPTISTRUCT;
- NASTRAN;
- PERMAS;
- ABAQUS.

Only <entity/> (Table 3) is allowed as a nested element of the child element of <femdata/>. Its definition and documentation follow <ENTITY/>, the corresponding element in FATXML [2].

**Table 3 — Nested elements of the child element of <femdata/>**

Nested Elements	Multiplicity	Use	Constraint / Remarks
entity	1-*	Required	Corresponds to element <ENTITY/>, defined in [2].

For further definition of ENTITY see the document source website for FATXML [2].

EXAMPLE A <femdata/> within a <connection\_0d/> element

```
<connection_0d>
...
<femdata>
  <NASTRAN>
    <entity>
      <TYPE>
        CQUAD
      </TYPE>
      <ID>
        12345-12356
      </ID>
    </entity>
  </NASTRAN>
</femdata>
...
</connection_0d>
```

Like FATXML,  $\chi$ 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.

### Reasoning about <femdata/>

<femdata/> element can be used versatile for different use cases – even for yet unknown ones. This makes it hard to define exact semantics.

Specific agreements, for example between preprocessor and solver/postprocessor can be made to support specific use cases.

## 7.4 Connection Data <connection\_group/>

### 7.4.1 General

<connection\_group/> comprises the topological information about the involved parts and assemblies (section 6), respectively. As explained in section 4.5, joints are grouped together by the parts or assemblies which they commonly connect.

The topological relation (relation of neighbours) 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 4.3).

Each <connection\_group/> is uniquely identified by a numeric identifier (id).

From this, it follows that  $\chi$ MCF files *cannot* be simply "pasted together" by use of a standard text editor.

XML-specification of <connection\_group/> is shown in Table 4 and Table 5 — Nested elements of element <connection\_group/>.

**Table 4 — Attributes of element <connection\_group/>**

Attributes	Type	Use	Constraint
id	Integer	Required	unique within a $\chi$ MCF file

**Table 5 — Nested elements of element <connection\_group/>**

Nested Elements	Multiplicity	Use	Constraint
connected_to	1	Required	-
connection_list	1	Required	-
contact_list	1	Optional	-

An empty or missing <connected\_to/> means a connection according to geometric neighbourhood, alone. However, if <connected\_to/> is present, it shall be *complete*, that means no additional connection partners are to be searched. (Searching for a geometric neighbourhood may give different results, depending on the algorithm employed. To avoid ambiguities, no connections with missing <connected\_to> should reach the solver. Therefore, <connected\_to> should be filled by the pre-processor.).

In addition to parts and properties, no other means (such as sets) for grouping objects are allowed.

## 7.4.2 Connected Objects

### 7.4.2.1 General

The basic objects which can be jointed together are parts and assemblies (see section 6) which appear as nested elements <part/> and <assy/> of <connected\_to/>. The XML-specification of <connected\_to/> is shown in Table 6.

**Table 6 — Nested elements of <connected\_to/>**

Nested Elements	Multiplicity	Use	Constraint
part	1 - *	Optional	-
assy	1 - *	Optional	-

### 7.4.2.2 Element <part/>

In χMCF, a part may refer to one CAX part or one CAE property, as well.

It is described by the element <part/> and a numeric *index*, a *label* (part code), a *pid* (property id) or *pname* (property name), all provided as attributes. However, if both attributes "label" and "pid" or "label" and "pname" are present, the label governs.

Although most solvers use numbers as identifiers, ABAQUS uses names as identifiers. To identify a property, only one of pid or pname is sufficient. If both identifiers are present, they have to be equivalent. Rationale for allowing presence of both identifiers is the case that the same mesh, and hence same properties, are used in both, NASTRAN and ABAQUS. Then, it would be good to have a χMCF file with both in, PIDs and property names. On solver side, this would cause no confusion since NASTRAN would ignore the property name and Abaqus the PID. The responsibility to keep both primary keys unique and consistent resides on pre-processor side. Upon import of χMCF to a pre-processor, inconsistent property keys shall cause an error.

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 *index* is NOT allowed as attribute of the <part/> element. XML-specification of <part/> is shown in Table 7.

Table 7 — Attributes of element `<part/>`

Attributes	Type	Value Space	Use	Constraint
index	Integer	> 0	Required	Unique and required only within the parent element <code>&lt;connected_to/&gt;</code>
label	Alphanumeric	Alphanumeric	Optional	At least label, pid, or pname must exist.
pid	Integer	> 0	Optional	
pname	Alphanumeric	Alphanumeric	Optional	
instance	Alphanumeric	non-empty	Optional	label must exist if instance is used.

EXAMPLE A `<part/>` with required attributes only (pid or pname could be used alternatively to label)

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

EXAMPLE B `<part/>` with optional use of label and pid

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

EXAMPLE C `<part/>` with pname to identify a part or property

```
<connected_to>
  <part index="1" pname="P3202132 Thin Shell Property"/>
</connected_to>
```

EXAMPLE D `<part/>` using a label and an instance to identify a part

```
<connected_to>
  <part index="1" label="PART_WHEEL_900" instance="4"/>
</connected_to>
```

### 7.4.2.3 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`. The XML specification of element `<assy/>` is shown in Table 8.

Table 8 — Attributes of element `<assy/>`

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

## EXAMPLE A Full definition of <assy/> element within <connected\_to>

```
<connected_to>
  <assy index="42">
    <part label="PART_7000400" pid="110013"/>
    <part label="PART_7000800" pid="110099"/>
  </assy>
</connected_to>
```

## EXAMPLE B Full definition of the combined use of <part/> and <assy/> elements within <connected\_to/>

```
<connected_to>
  <part index="1" label="PART_9004400" pid="3202132"/>
  <assy index="42">
    <part label="PART_7000400" pid="110013"/>
    <part label="PART_7000800" pid="110099"/>
  </assy>
</connected_to>
```

## EXAMPLE C Usage of the attribute instance for a <part/> within an <assy/> element

```
<connected_to>
  <part index="1" label="PART_9004400" pid="3202132"/>
  <assy index="42">
    <part label="PART_7000400" instance="2" pid="110013"/>
    <part label="PART_7000800" pid="110099"/>
  </assy>
</connected_to>
```

## EXAMPLE D Minimum definition for the combined use of <part/> and <assy/> elements within <connected\_to>

```
<connected_to>
  <part index="1" label="PART_9004400"/>
  <assy index="42">
    <part label="PART_7000400"/>
    <part label="PART_7000800"/>
  </assy>
</connected_to>
```

OR

```
<connected_to>
  <part index="1" pid="3202132"/>
  <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, such as an <assy/> tag nested within another <assy/> tag, is not allowed.

## 7.4.2.4 Special Topological situations

### 7.4.2.4.1 Stacking

The aim of the `<connection_group/>` element is to group up all the joints that connect the same parts. Therefore, `<connected_to/>` contains each part connected in the joint only once. However, it may be important to explicitly define in which order some parts of the group are connected.

This includes the following scenarios:

- the stacking order of the connected parts may be important;
- some parts may be involved more than once in the same joint (self-connected joint), where either:
  - each part involved in a self-connected joint more than once is known individually, or;
  - just the *number* of parts involved in a self-connected joint is known;
  - or a combination of the two sub-scenarios above.

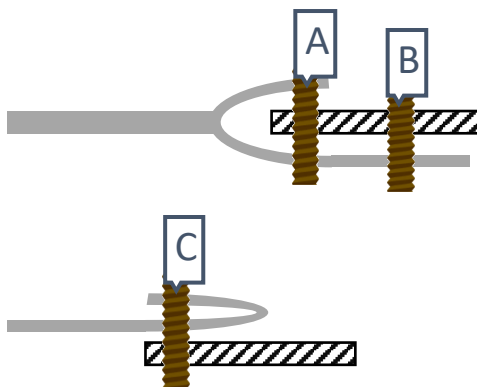


Figure 7 — Special Stacking Topologies

In Figure 7, all joints, A, B, C, exist within the same `<connection_group/>`, but each joint is connected in a different way.

```
<connection_group>
  <connected_to>
    <part index="1" label="PART_7000800"/> <!-- grey part in figure -->
    <part index="2" label="PART_7000400"/> <!-- hatched part in figure -->
  </connected_to>
</connection_group>
```

For joints A and C, the number of flanges connected is more than the number of parts in `<connected_to/>`. Between joints A and C, the flanges feature the same parts, but in a different order.

To store this information for each case, the `<stacking/>` element comes to use.

### 7.4.2.4.2 Element `<stacking/>`

`<stacking/>` may dictate list of flanges/sheets involved in a joint, as well as their order. Alternatively, `<stacking/>` may indicate the number of flanges/sheets of a joint, without defining which are the parts that are connected more than once. Table 9 shows the nesting element of `<stacking/>` and Table 10 its attribute.

**Table 9 — Nested elements of <stacking/>**

Nested Elements	Multiplicity	Use	Constraint
level	1 - *	Optional	-

**Table 10 — Attributes of <stacking>**

Attributes	Type	Value Space	Use	Constraints / Remarks
nr_levels	Integer	> 0	Optional	if <code>nr_levels</code> exists, no <code>&lt;level/&gt;</code> elements are allowed in <code>&lt;stacking/&gt;</code> .  <code>nr_levels</code> has to be greater than the number of nested elements of <code>&lt;connected_to/&gt;</code>

The attribute meaning is:

- `nr_levels` dictates the number of flanges/sheets connected by the joint.

The element `<level/>` within `<stacking/>` is specified as shown in Table 11.

**Table 11 — Attributes of <level>**

Attributes	Type	Value Space	Use	Constraint
order	Integer	> 0	Required	Unique only within the parent element <code>&lt;stacking/&gt;</code>
part_index	Integer		Required	

The attribute meanings are:

- `part_index`: The flange partner with this index (see section 7.4.2.2). The part of the flange is referenced by the attribute `index` inside the element `<part/>` or `<assy/>` of the `<connected_to/>` element;
- `order`: indicates the position of a flange relative to other flanges.

The order of the levels in the stacking list is identified by the numerical value of their attribute `order`, in ascending order. Hence, indices shall be unique within one stacking list. The restriction `nr_levels` must be greater than the number of nested elements of `<connected_to/>` implies that `nr_levels` attribute can only be used for self-connected joints.

**EXAMPLE A** The situations in may be described using `<level/>` elements in order to explicitly define the stacking of the part flanges involved.

```
<connection_group>
```

```
<connected_to>
```

```
<part index="1" label="PART_7000800"/>
```

```
<part index="2" label="PART_7000400"/>
```

```
</connected_to>
```

```
<!-- grey part in figure -->
```

```
<!-- hatched part in figure -->
```



```

<connection_list>
  <connection_0d label="A">
    <stacking>
      <level order="1" part_index="1"/> <!-- grey part in figure -->
      <level order="2" part_index="2"/> <!-- hatched part in figure -->
      <level order="3" part_index="1"/> <!-- grey part in figure -->
    </stacking>
    ...
  </connection_0d>

  <connection_0d label="B">
    <stacking>
      <level order="1" part_index="2"/> <!-- hatched part in figure -->
      <level order="2" part_index="1"/> <!-- grey part in figure -->
    </stacking>
    ...
  </connection_0d>

  <connection_0d label="C">
    <stacking>
      <level order="1" part_index="1"/> <!-- grey part in figure -->
      <level order="2" part_index="1"/> <!-- grey part in figure -->
      <level order="3" part_index="2"/> <!-- hatched part in figure -->
    </stacking>
    ...
  </connection_0d>
</connection_list>

</connection_group>

```

EXAMPLE B may be expressed using the `nr_levels` attribute, which simply states how many flanges of the `<connected_to>` parts are involved in each joint.

```

<connection_group>

  <connected_to>
    <part index="1" label="PART_7000800"/> <!-- grey part in figure -->
    <part index="2" label="PART_7000400"/> <!-- hatched part in figure -->
  </connected_to>

  <connection_list>
    <connection_0d label="A">
      <stacking nr_levels="3"/> <!-- "hatched", "grey" and one of "hatched"/"grey" -->
      ...
    </connection_0d>

    <connection_0d label="B">
      ... <!-- "hatched", "grey" in any order -->
    </connection_0d>

    <connection_0d label="C">
      <stacking nr_levels="3"/> <!-- "hatched", "grey" and one of "hatched"/"grey" -->
      ...
    </connection_0d>
  </connection_list>

</connection_group>

```

## 7.4.3 Contacts and Friction

### 7.4.3.1 General

For many joint types like bolts, screws etc., friction between the jointed partners plays an important role for the manufacturing and the mechanical behaviour 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 shall allow for local modification of an individual connection to enhance the service behaviour.

In  $\chi$ MCF, friction coefficients for any combination of joint partners defined in `<connected_to/>` can be specified by the element `<contact/>` 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.

#### 7.4.3.2 Element `<contact_list/>`

Relevant contacts, which are possible between the flange partners of a `<connection_group/>`, are collected in a `<contact_list/>`. The XML specification of `<contact_list/>` element is shown in Table 12.

**Table 12 — Nested elements of element `<contact_list/>`**

Nested Elements	Multiplicity	Use	Constraint
contact	1 - *	Required	Any set (= non-ordered pair) of physical contact partners is not allowed to appear more than once within a <code>&lt;contact_list/&gt;</code> .

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

#### 7.4.3.3 Element `<contact/>`

The features or coefficients of a physical contact between flange partners are described by an element `<contact/>`. The XML specification of the element `<contact/>` is shown in Table 13.

**Table 13 — Nested elements of element `<contact/>`**

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

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.

#### 7.4.3.4 Element `<partner/>`

Each joint partner involved in a contact is specified by the element `<partner/>`. Only the first level parts/assemblies which are listed in `<connected_to/>`, are allowed. XML specification of `<partner/>` element are shown in Table 14.

**Table 14 — Attributes of element `<partner/>`**

Attributes	Type	Value Space	Use	Constraints / Remarks
part_index	Integer		Required	

The attribute has following meaning:

- `part_index`: The flange partner with this index (see section 7.4.2.2). The part of the flange is referenced by the attribute `index` inside the element `<part/>` or `<assy/>` of the `<connected_to/>` element.

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

#### 7.4.3.5 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="PART_9004400" pid="3202132"/>
  <assy index="42">
    <part label="PART_7000400" pid="110013"/>
    <part label="PART_7000800" pid="110099"/>
  </assy>
</connected_to>
<contact_list>
  <contact>
    <partner part_index="1"/>
    <partner part_index="42"/>
    <coefficients static_friction="0.3" kinetic_friction=".25"/>
  </contact>
</contact_list>
```

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

#### 7.4.3.6 Local Contact Properties

If necessary, local contact properties can be given within any element `<connection_0d/>` or `<connection_1d/>`, respectively (see section 7.4.4 Joints). In case of conflict, a local `<contact_list/>` overrules the global one.

NOTE: `<connection_2d/>` is not relevant for the currently known use cases and was therefore intentionally not included in the list.

XML-specification of `<coefficients/>` are shown in Table 15.

**Table 15 — Attributes of element `<coefficients/>`**

Attributes	Type	Value Space	Use	Constraints / Remarks
<code>static_friction</code>	Floating point	$[0, \infty[$	Optional	-
<code>kinetic_friction</code>	Floating point	$[0, \infty[$	Optional	-

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

#### 7.4.4 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  $\chi$ MCF file.

As discussed in Sect. 4.3,  $\chi$ MCF 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/>`:

**Table 16 — Nested elements of element `<connection_list/>`**

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

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

## 7.5 A Minimalistic Example of a $\chi$ MCF file

In the following, an example shows how the  $\chi$ MCF xml file should look like.

### EXAMPLE

```
<?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_3_1_1.xsd">

  <!-- some comments -->
  <date> 2016-01-11 </date>
  <version> 3.1.1 </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 ...>
    <connected_to>
      <part index="1" label="PART_8000880" pid="20123213"/>
      <part index="2" label="PART_8100340" 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>
            ...
          </NASTRAN>
        </femdata>
        ...
      </connection_0d>
      <connection_1d>
        <loc_list>
          ...
        </loc_list>
        <seamweld>
          ...
        </seamweld>
      </connection_1d>
      <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>

```

## 7.6 XML Schema Definition

The XML Schema definition (XSD) can be found in computer-interpretable form at the following URL:

<https://standards.iso.org/iso/pas/8329/ed-1/en/>

## 8 Data Common to any Connection

### 8.1 Indices and their properties

χMCF 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. For instance, 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. χMCF 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.

### 8.2 Connection Referencing

#### 8.2.1 Need for Referencing

Any connection should have a way of referring to it, since its shape and dimensions may vary along the design process. Typically, connections are referred to by assigning them IDs and labels.

#### 8.2.2 Attribute `label`

Any connection should have an attribute called `label`, which identifies it throughout the entire CAx process, maybe even the complete product lifecycle including manufacturing. 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 (if there are interfering holes in the structure, for example), its components shall keep the `label` attribute. A system "way down" in the process (detached from any centralized naming authority) may create new connections with all the same label such as "0" or empty string.

The `label` may be composed of digits only, but it should not be confused with other IDs such as a finite element's ID. If desired, finite element IDs would have to be placed within some `<appdata/>` element.

### 8.2.3 Attribute `ident`

For systems or processes that use integers for referring to connections, the attribute `ident` is provided. In contrast to alphanumeric labels, integers are easy to generate and simple to shift when grouping is needed. This allows for unique identification, detached from a centralized naming authority, in case a connection is split, inserted, or duplicated.

`ident` can be used together with `label` as alternative ways of referring to a connection, bridging the gap between tools that work with integers only and tools that use labels only.

`ident` is a positive integer and unique within the  $\chi$ MCF file.

#### EXAMPLE

```
<connection_list>
  <connection_0d label="SPOT_3490" ident="3490">
    <loc> ... </loc>
    <spotweld/>
  </connection_0d>
</connection_list>
```

## 8.3 Dimensions and Coordinates

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

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

### 8.4 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,  $\chi$ MCF cannot define a set of possible values for this attribute. Hence, it must be of type `Alphanumeric`.

## 8.5 Custom Attributes list

It was mentioned in section 4.2 that only information relevant to connections should be contained in  $\chi$ MCF. Exceptions `<appdata/>` and `<femdata/>` were introduced in section 7.3. `<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  $\chi$ MCF wishes to introduce supplementary information (attributes) to enrich the standard attributes defined by  $\chi$ 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/>`,  $\chi$ 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*  $\leftrightarrow$  *value(s)*, with supported *value-type*  $\in [\text{int}, \text{real}, \text{string}]^N$ , where  $N$  is a positive integer:

(*value-type*) *key* = {*value1*, *value2*, ..., *valueN*}.

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

This means that 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:

**Table 17 — Nested elements of element `<custom_attributes_list/>`**

Nested Elements	Multiplicity	Use	Constraint
custom_attributes	1 - *	Required	-

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/>`:

**Table 18 — Attributes of `<custom_attributes/>` element**

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

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

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

**Table 19 — Nested elements of element `<custom_attributes/>`**

Nested Elements	Multiplicity	Use	Constraints / Remarks
string	1 - *	optional	At least one of these nested elements.
real	1 - *	optional	
int	1 - *	optional	
string_list	1 - *	optional	
real_list	1 - *	optional	
int_list	1 - *	optional	

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.
  - (If required to handle not needed items such as line feeds or tab characters another data type should be used in XML Schema Definition which is the 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/>`:

**Table 20 — Attributes of `<string/>` element**

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

XML specification of `<real/>`:

**Table 21 — Attributes of `<real/>` element**

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



XML specification of `<integer/>`:

**Table 22 — Attributes of `<integer/>` element**

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

XML specification of `<string_list/>`:

**Table 23 — Attributes of `<string_list/>` element**

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

`<string_list/>` has the nested element:

**Table 24 — Nested elements of `<string_list/>` element**

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

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

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

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

XML specification of `<real_list/>`:

**Table 26 — Attributes of `<real_list/>` element**

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

`<real_list/>` has the nested element:

**Table 27— Nested element of `<real_list/>` element**

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

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

**Table 28 — Attributes of `<value>` element inside `<real_list/>`**

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

XML specification of <int\_list/>:

**Table 29 — Attributes of <int\_list/> element**

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

<int\_list/> has the nested element:

**Table 30 — Nested elements of <int\_list/> element**

Nested Elements	Type	Multiplicity	Use	Constraints / Remarks
value	integer	1 - *	Required	-

Where <value/> within <int\_list/> is specified as:

**Table 31 — Attributes of <value/> element inside <real\_list/>**

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

#### Remarks:

1. Values of key have to 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. Hence, indices have to be unique within one list;
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>
```

## 8.6 Distinction between `<custom_attributes/>` and `<appdata/>`

### 8.6.1 General

At first glance, `<custom_attributes/>` and `<appdata/>` seem to address similar purpose or even to be redundant. This is deceptive, as evidenced by the following subsections.

### 8.6.2 Needs of different process roles, addressed by `<custom_attributes/>` and `<appdata/>`

In context of  $\chi$ MCF, at least two different roles can be clearly identified: The programmer of an application and the engineer using this application. The programmer needs to store extra data that are specific to the application. The engineer needs to store extra data that are specific to the process that the connections are involved.

As its name implies, `<appdata/>` is used to store application-specific data, whose structure and purpose is known only by the application itself, respectively applies to this application alone. The software vendor can choose to standardize and publish the format of this data in order to allow other applications to port data to it or may choose to use `<appdata/>` as a private storage of internal state.

`<custom_attributes/>` represent OEM- or process-specific data, whose purpose is known by the engineers, but may not be known by the application.

Engineers store connection-related information in `<custom_attributes/>`. Engineers choose which attributes they need to store and designate the corresponding data in `<custom_attributes/>`.

Applications store auxiliary data in `<appdata/>`. These data possibly may be data that the engineers do not know of. `<appdata/>` may include information about the internal state of the application specific data model.

Engineers know the purpose and representation of `<custom_attributes/>`. The software may not know what each custom attribute represents, but it shall nevertheless be able to transport these data unchanged, or to offer a (generic) GUI for accessing it.

### 8.6.3 Needs of different applications, addressed by `<custom_attributes/>` and `<appdata/>`

`<appdata/>` may be used as means of intercommunication between different applications. In this case, the format of `<appdata/>` needs to be standardized and published by the `<appdata/>` owner. Of course, information stored in `<appdata/>` does not necessarily need to be handled or maintained by 3<sup>rd</sup> party software. Therefore, `<appdata/>` should be considered as data that can be disregarded or thrown away by a 3<sup>rd</sup> party software. Hence, applications shall not rely on preservation of `<appdata/>`. Data corruption or crash has to be avoided if data from `<appdata/>` gets lost.

Internal structure of `<custom_attributes_list/>` is completely standardized, whereas internal structure of `<appdata/>` is arbitrary and can for instance be described by a software specific XML schema. (But this is optional.) Hence, `<custom_attributes_list/>` cannot be used as flexible as `<appdata/>`, but its content is easier to be preserved across system boundaries.

### 8.6.4 Different levels of `<custom_attributes/>` and `<appdata/>` within $\chi$ MCF data model

`<appdata/>` may be used on different levels of a  $\chi$ 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, but not at root level. There are good reasons for this:

Consider the common scenario, where many  $\chi$ 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.

## 9 0D connections

### 9.1 Generic Definitions

#### 9.1.1 Identification

##### 9.1.1.1 General

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

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

**Table 32 — Attributes of element `<connection_0d/>`**

Attributes	Type	Use	Constraint
label	Alphanumeric	Optional	-
ident	Integer	Optional	positive, unique within a $\chi$ MCF file
quality_control	Alphanumeric	Optional	See section 8.4 Attribute <code>quality_control</code>

##### 9.1.1.2 Attribute "label"

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

EXAMPLE A Minimum definition of a 0d connection without label

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

```
</connection_0d>
</connection_list>
```

EXAMPLE B Definition of a 0d connection with label

```
<connection_list>
  <connection_0d label="SPOT_3490">
    <loc>
      ...
    </loc>
    <spotweld>
      ...
    </spotweld>
  </connection_0d>
</connection_list>
```

9.1.1.3 Attribute "ident"

The attribute `ident` provides an alternative identification to the connection. The value of `ident` is a positive integer and unique within the  $\chi$ MCF file.

EXAMPLE

```
<connection_list>
  <connection_0d ident="3490">
    <loc>
      ...
    </loc>
    <spotweld>
      ...
    </spotweld>
  </connection_0d>
</connection_list>
```

9.1.2 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.

Table 33 — Text values of element `<loc/>`

Text	Type	Value Space	Use	Constraint
x	Floating point	$(-\infty, +\infty)$	Required	-
y	Floating point	$(-\infty, +\infty)$	Required	-
z	Floating point	$(-\infty, +\infty)$	Required	-

EXAMPLE

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

### 9.1.3 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, but must be  $> 0$ .

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 orthogonalization relative to `<normal_direction/>` is not allowed to vanish, that means both vectors are not allowed to 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 product z-axis  $\times$  x-axis.

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

**Table 34 — Attributes of elements `<normal_direction/>` & `<tangential_direction/>`**

Attributes	Type	Value Space	Use	Constraint
x	Floating point	$(-\infty, +\infty)$	Required	-
y	Floating point	$(-\infty, +\infty)$	Required	-
z	Floating point	$(-\infty, +\infty)$	Required	-

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 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.

#### EXAMPLE

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

### 9.1.4 Type Specification

Each connection is identified by its type. The XML definitions of all 0D connections are containing the following elements:

**Table 35 — Nested elements of element <connection\_0d/>**

Nested Elements	Multiplicity	Use	Constraint
clinch	1	Optional	-
clip	1	Optional	-
heat_stake	1	Optional	-
nail	1	Optional	-
gumdrop	1	Optional	-
rivet	1	Optional	-
robscan	1	Optional	-
rotation_joint	1	Optional	-
spotweld	1	Optional	-
threaded_connection	1	Optional	-
contact_list	1	Optional	See section 7.4.3.6.
stacking	1	Optional	See section 7.4.2.4

Exactly one of the type elements (clinch, clip, heat\_stake, gumdrop, nail, rivet, robscan, spotweld, or threaded\_connection) need to exist in <connection\_0d/>. There is no default type.

## 9.2 Spot Welds

### 9.2.1 General

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

**Table 36 — Nested elements of <connection\_0d/> for <spotweld/>**

Nested Elements	Multiplicity	Use	Constraint
spotweld	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of <spotweld/> with element diameter:

**Table 37 — Attributes of element <spotweld/>**

Attributes	Type	Value Space	Use	Constraint
diameter	Floating point	> 0.0	Optional	-

Attributes	Type	Value Space	Use	Constraint
technology	Selection	resistance, laser, projection, friction	Optional	-

### 9.2.2 Attribute "diameter"

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

### 9.2.3 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.

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

**Table 38 — Nested elements of element `<spotweld/>`**

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

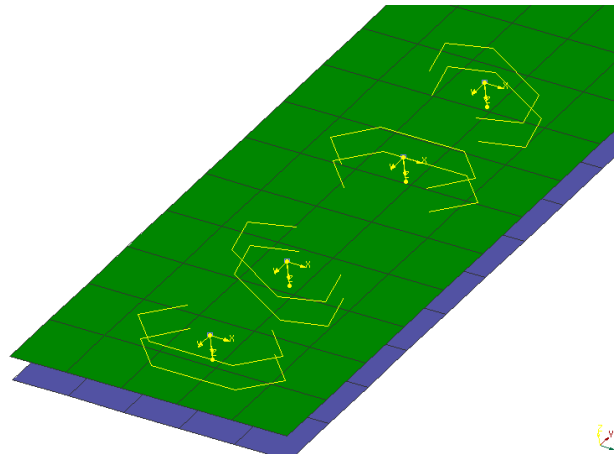
#### EXAMPLE

```
<connection_0d label="SPOT_Left_Gh_2123921">
  <spotweld diameter="5.0"/>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

## 9.3 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 significantly 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 8 — 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 shall 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 may have access to the position and "bounding box" of the Robscan;
- hence,  $\chi$ 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 3<sup>rd</sup> party intellectual property, it cannot be part of  $\chi$ 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.

**Table 39 — Nested elements of `<connection_0d/>` for `<robscan/>`**

Nested Elements	Multiplicity	Use	Constraint / Remarks
robscan	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of `<robscan/>` element:

**Table 40 — Attributes of element `<robscan/>`**

Attributes	Type	Value Space	Use	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°, 180°]	Optional	According to the unit of angles, defined in element <code>&lt;units/&gt;</code> .
filler_material	Alphanumeric	Alphanumeric	Optional	-

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`: this defines the gap between both flange partners;
- `width`: this is the width of the laser beam;
- Width and length of the pattern are given by attributes `pattern_width` and `pattern_length`;
- `mirrored`: this denotes, whether pattern has to be mirrored along its length-axis x, i. e. local y coordinate has to be inverted;
- `orientation_angle`: this defines 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 another 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.

**Table 41 — Nested elements of element <robscan/>**

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

Additional explanations for the directions are:

- Element <normal\_direction/> denotes direction of laser beam, giving local z axis;
- Element <tangential\_direction/> denotes laser moving direction, giving local x axis.

<normal\_direction/> and <tangential\_direction/> elements are described in section 9.1.3.

#### EXAMPLE

```
<connection_0d label="RSC_1272360">
  <loc> 507 1 0.8 </loc>
  <robscan base="1" pattern="KL_ST" 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>
```

## 9.4 Rivets

### 9.4.1 General

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.

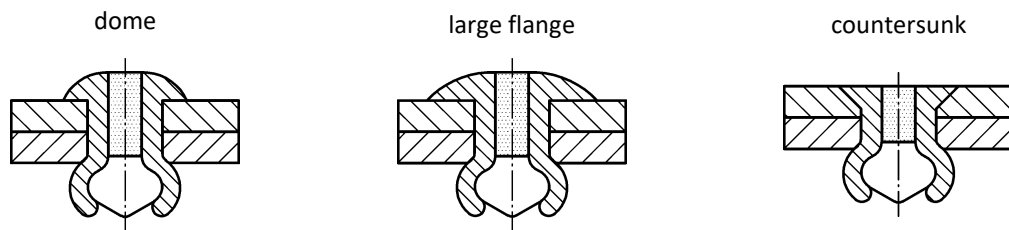
**Table 42 — Nested elements of <connection\_0d/> for <rivet/>**

Nested Elements	Multiplicity	Use	Constraint / Remarks
rivet	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of <rivet/> element:

**Table 43 — Attributes of element <rivet/>**

Attributes	Type	Value Space	Use	Constraint / Remarks
hardness	Floating point	> 0.0	Optional	-
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	-



**Figure 9 — Rivet head types (Dome, Large Flange, Countersunk)**

The following list explains the attributes:

- **hardness**: Vickers hardness HV of the rivet material. (Attribute hardness was moved from element <self\_piercing/> to element <rivet/> with χMCF version 3.1.);
- **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, SAE etc., e.g. [3].
- **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, hence, 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 9.1.3.

A <tangential\_direction/> can be provided for rivets that are not axis-symmetric and require a special orientation.

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 nested elements, listed in following table:

**Table 44 — Nested elements of element `<rivet/>`**

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

The subtypes are described in detail in the following sections.

**EXAMPLE A** Example for a (axisymmetric) rivet connection that uses only the `<normal_direction/>`

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

**EXAMPLE B** Example for a rivet connection that requires also the `<tangential_direction/>`

```
<connection_0d label="RVT_2123922">
  ...
  <rivet shaft_diameter="5.0" head_diameter="8" length="3.5">
    <normal_direction x="0" y="0" z="3"/>
    <tangential_direction x="3" y="0" z="0"/>
  </rivet>
  <loc> 1645.83 -821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

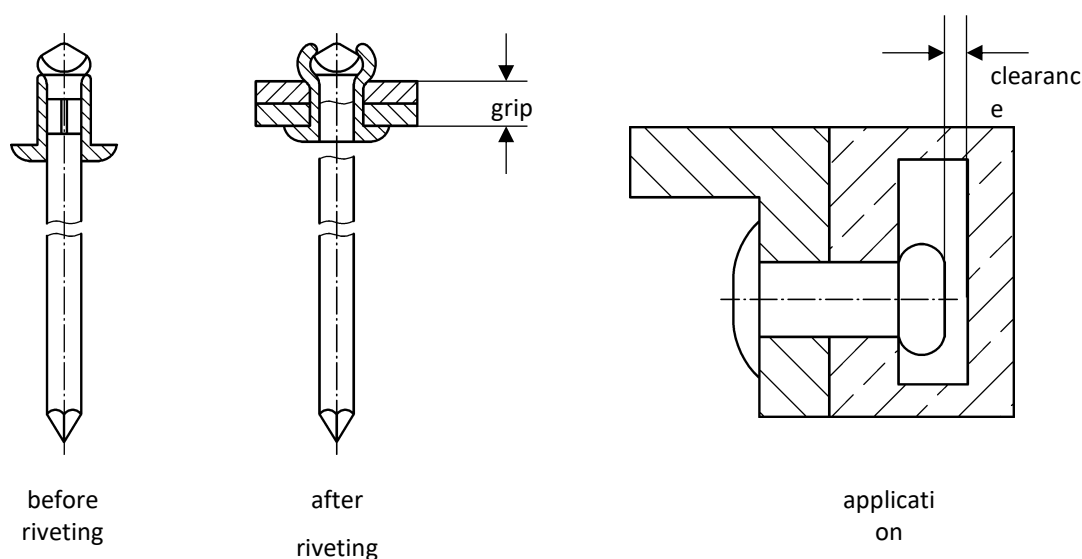
## 9.4.2 Blind Rivets

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:

**Table 45 — Attributes of element <blind/>**

Attributes	Type	Value Space	Use	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



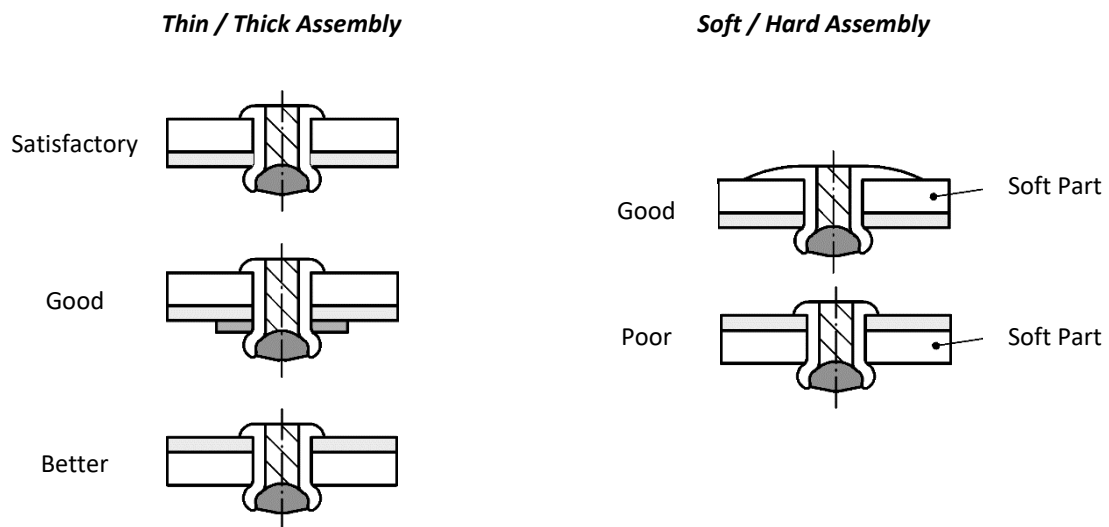
**Figure 10 — Blind rivet – key attributes**

Figure 10 describes 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;
- **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 Aluminium.

**NOTE** In case of material thickness changes in connected parts might lead to other size of blind rivet as joining element!

When a blind rivet is 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 Figure 11 — Assembly Recommendations for Blind Rivets



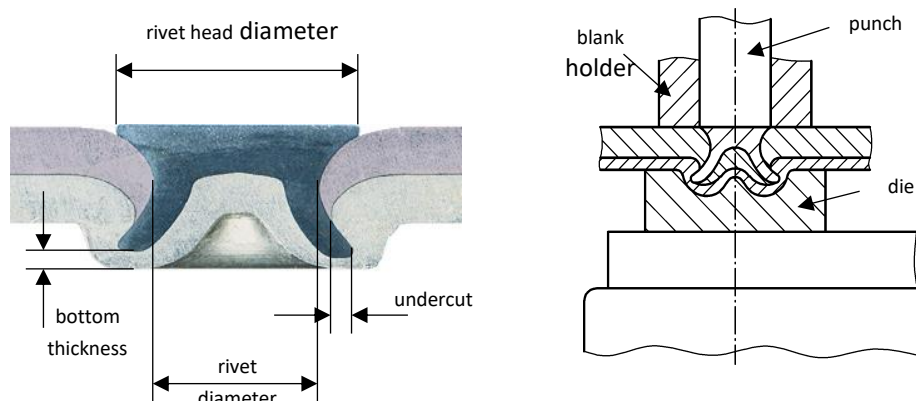
**Figure 11 — Assembly Recommendations for Blind Rivets**

#### 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>
```

### 9.4.3 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 & riveting machine**

There is a wide range of such rivets available on the market. They can be used with different rivet dies to optimize the riveting process. Such combinations have to be chosen in accordance with the materials of the flange partners. This is 3<sup>rd</sup> party intellectual property and hence cannot be part of  $\chi$ MCF 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 related 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:

**Table 46 — Attributes of element `<self_piercing/>`**

Attributes	Type	Value Space	Use	Constraint
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	-

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 catalogue entry. Then, `die_diameter` and `die_depth` can be omitted in  $\chi$ MCF file if their values are given in catalogue.

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 catalogue, too, if present. The `<rivet/>` in  $\chi$ MCF file is *not* allowed to 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.

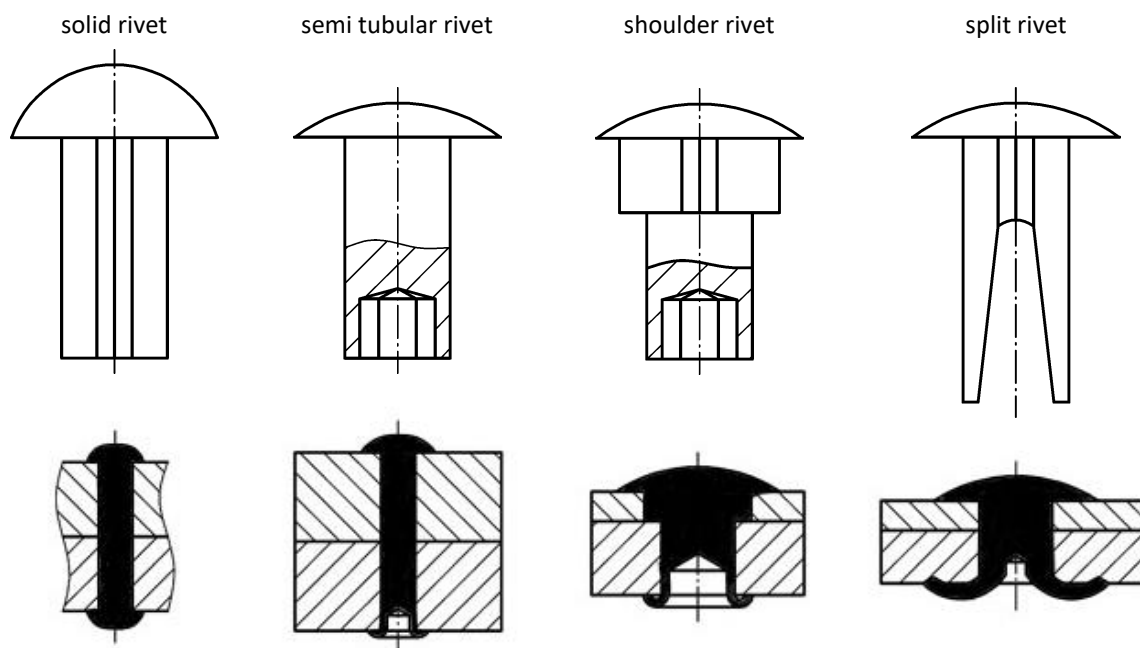
#### EXAMPLE

```
<connection_0d label="RVT_2123921">
  <rivet shaft_diameter="3.35" head_diameter="5.5" length="4" hardness="410">
    <normal_direction x="0" y="0" z="3"/>
    <self_piercing head_label="N000000002651" 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>
```



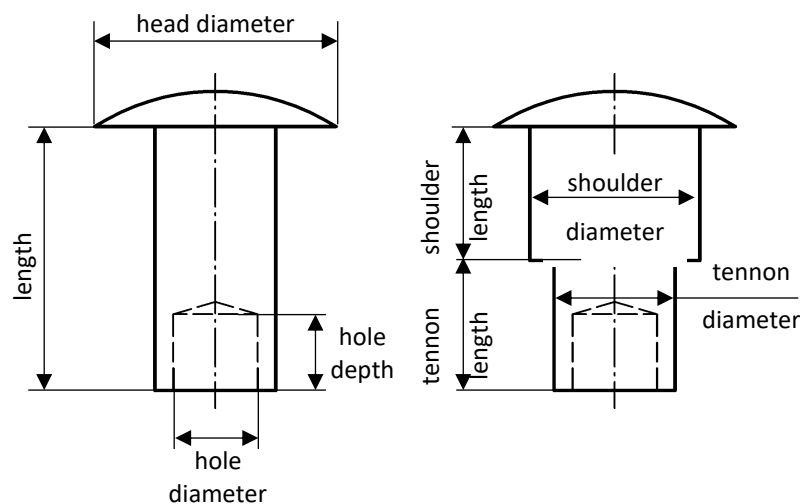
#### 9.4.4 Solid Rivets

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



**Figure 13 — Pictures of characteristic rivet types before and after mounting.**

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



**Figure 14 — Key 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:

**Table 47 — Attributes of element <solid/>**

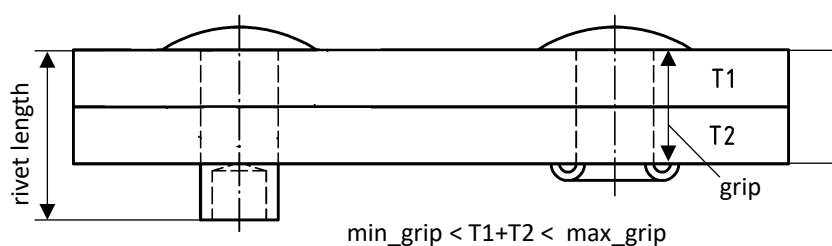
Attributes	Type	Value Space	Use	Constraint
min_grip	Floating point	> 0.0	Optional	-
max_grip	Floating point	> 0.0	Optional	max_grip ≥ min_grip
hole_diameter	Floating point	> 0.0	Optional	-
hole_depth	Floating point	> 0.0	Optional	-
shoulder_diameter	Floating point	> 0.0	Optional	-
shoulder_length	Floating point	> 0.0	Optional	-
tennon_diameter	Floating point	> 0.0	Optional	-
tennon_length	Floating point	> 0.0	Optional	-

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 — Relation of working thickness (T1+T2) to max and min values of grip**

- 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 χMCF.

EXAMPLE

```
<connection_0d label="RVT_2123921">
  <loc> 1645.83 821.145 616.585 </loc>
```

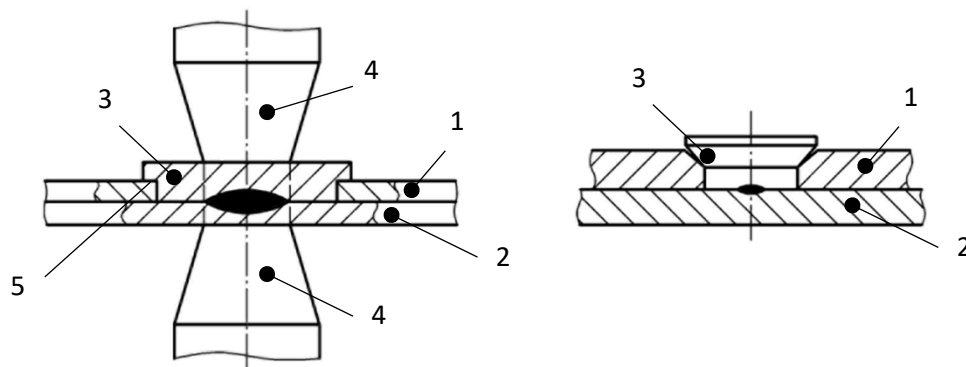
```

<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>

```

### 9.4.5 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 aluminium and steel parts are going to be connected.



- 1 – non-weldable part/material
- 2 – weldable part/material
- 3 – insert or plug
- 4 – electrodes
- 5 – stop surface

**Figure 16 — Cross section of a SWOP Rivet**

The following description is quoted from the Patent documented under **EP 0967044 A2** [4]:

"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 in the first sheet having a transverse dimension substantially greater than the transverse dimension of the tips of the electrodes (4) which are used for carrying out the electric welding spot. Within the hole there is provided an insert (3) of a material which can be electrically welded. The two electrodes are applied so as to cause welding of the insert to the second sheet. The insert has a stop surface (5) 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 and the insert 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  $\chi$ MCF 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 (such as: `length` or `shaft_diameter`) may be treated meaninglessly.

XML specification of <swop/> element:

**Table 48 — Attributes of element <swop/>**

Attributes	Type	Value Space	Use	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	-

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 9.2 Spot Welds;
- spotweld\_technology: Technology of the spot weld, see section 9.2 Spot Welds.

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

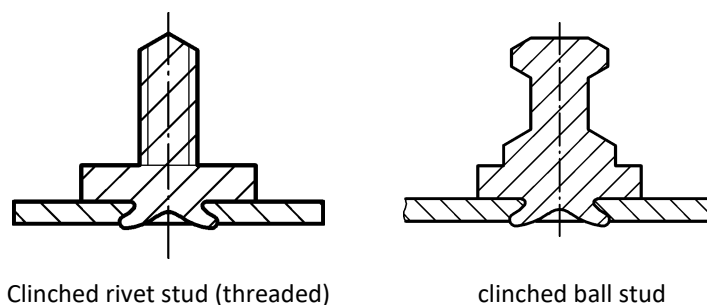
#### EXAMPLE

```
<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>
```

#### 9.4.6 Clinch Rivet Studs

A Clinch Rivet Stud (Clinchrietbolzen, or CNB) is fixed to the base metal sheet, typically by cold forming. The fastening method does not need additional components. Special tools are used to plastically form a mechanical interlock between the pin and the sheet.

One or more panels, typically of different material, are attached to the stud and fastened using a counterpart (a coarse nut, or a Tucker plastic nut).



**Figure 17 — Clinch Rivet Studs: Threaded variant and Ball stud**

A clinch rivet stud is denoted by a nested element `<clinch_rivet_stud/>` within `<rivet/>`. This element is described completely by the attributes of both XML elements. The attributes `shaft_diameter`, `length`, and `part_code` are inherited from `<rivet/>` element.

XML specification of `<clinch_rivet_stud/>` element:

**Table 49 — Attributes of element `<clinch_rivet_stud/>`**

Attributes	Type	Value Space	Use	Constraint
<code>press_in_force</code>	Floating point	<code>&gt; 0.0</code>	Optional	-

All attributes of this connection type are optional for importing it into CAD or CAE application.

These attributes have the following semantics:

- `press_in_force`: The force used to clinch the stud into the base sheet.  
For the unit, see section 7.2.5 Unit System;

The element of `<clinch_rivet_stud/>` does not allow any nested elements.

Direction sense of `<normal_direction/>` is towards the base sheet, where the rivet penetrates the metal.

#### EXAMPLE

```
<connection_0d label="CNB_2123921">
  <loc> 1645.83 821.145 616.585 </loc>
  <rivet shaft_diameter="4.0" length="6.0" >
    <normal_direction x="0" y="0" z="3" />
    <clinch_rivet_stud press_in_force="2000"/>
  </rivet>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

## 9.5 Threaded Connections: Bolts and Screws

### 9.5.1 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:

- Bolts are for the assembly of unthreaded components, with the aid of a nut;
- Screws are used in components which contain their own thread, and the screw may even cut its own internal thread into them.

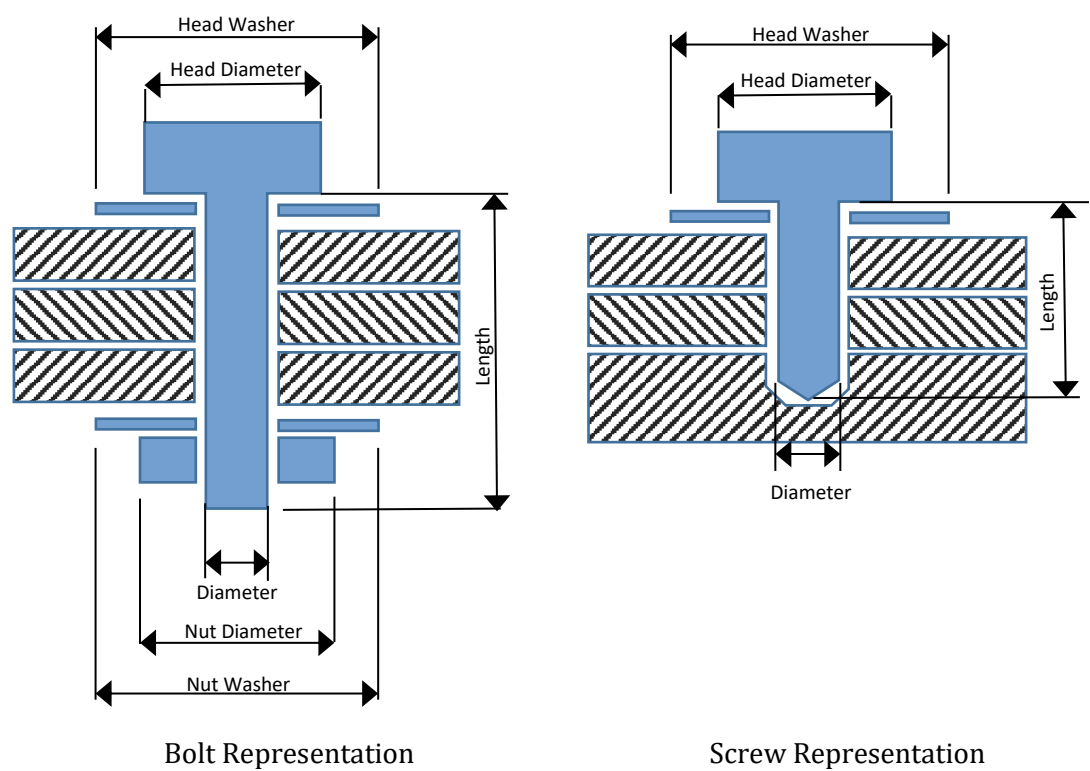
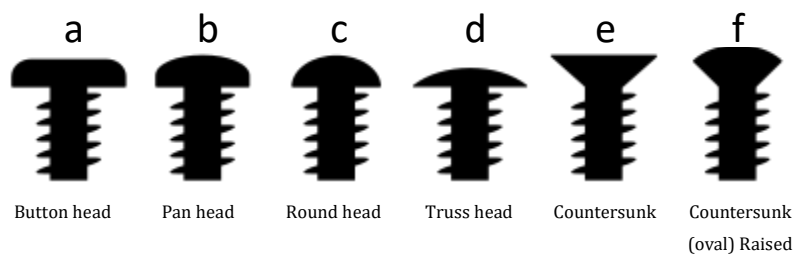
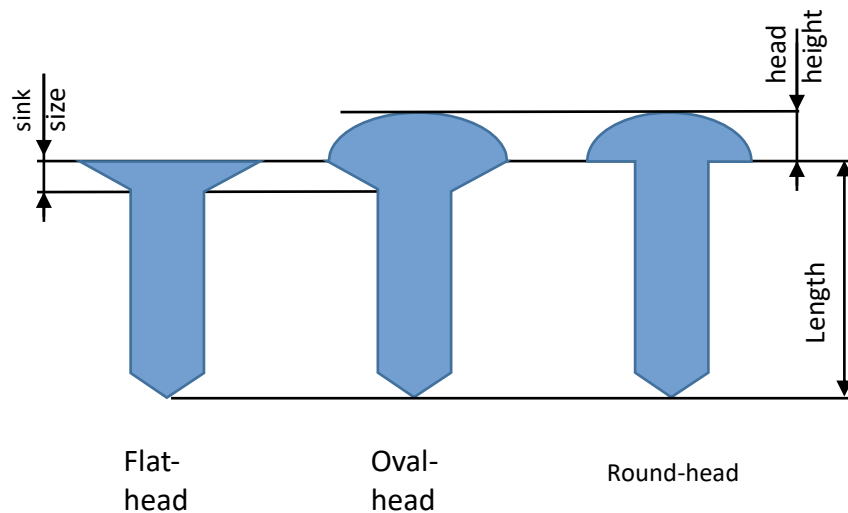


Figure 18 — Bolts and Screws

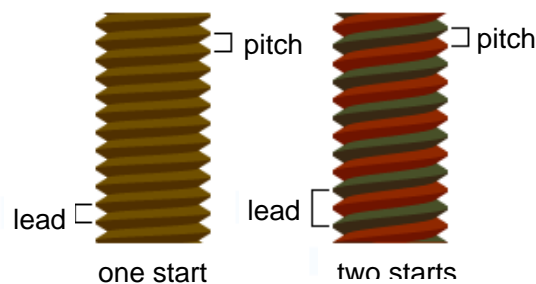


Source of image: [http://commons.wikimedia.org/wiki/File:Screw\\_head\\_types.svg](http://commons.wikimedia.org/wiki/File:Screw_head_types.svg).  
Author published it under the " [Creative Commons Attribution-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/) license".

Figure 19 — Different Screw Forms



**Figure 20 — Definition of Length and Head Sizes**



Source of image: Wikimedia

**Figure 21 — Definition of lead, pitch and starts of a thread.**

### 9.5.2 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  $\chi$ 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 a washer is loose, that means 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. between the connected sheets;
5. last connected part and "loose" washer (if there is one);
6. washer (if there is one) and nut, or else;
7. last connected part and nut;
8. screw and cut thread, or bolt thread and nut thread.

Consequently,  $\chi$ 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 between each two adjacent parts;
- d. the thread.

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

Case c. above, of inter-part contacts, is addressed by sections 5.3.2 (Contacts and Friction) and 7.4.3.6 (Local Contact Properties).

Case d. above, of the thread contact, is addressed in section 9.5.3 (Definition of element `<threaded_connection/>`).

#### EXAMPLE A Bolted Joint with washer definition

```
<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="BOLT_135"> <!-- bolt with washers -->
      <loc> 84 60 10 </loc>
      <!-- Friction is "head to washer": -->
      <threaded_connection length="50"
        static_friction="0.8"
        thread_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>
  </threaded_connection>

  <contact_list> <!-- Local Contact definition, according to 5.3.2.5 -->
    <contact>
      <partner part_index="1"/>
      <partner part_index="2"/>
      <coefficients static_friction="0.8"/>
    </contact>
    <contact>
      <partner part_index="2"/>
      <partner part_index="5"/>
      <coefficients static_friction="0.8"/>
    </contact>
    <contact>
      <partner part_index="5"/>
      <partner part_index="6"/>
      <coefficients static_friction="0.8"/>
    </contact>
  </contact_list>
</connection_0d>
</connection_list>
</connection_group>
```



**EXAMPLE B** Bolted Joint without washer definition but with global & local contact definition, and thread contact. Local contacts override global contacts.

```
<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>

  <contact_list>    <!-- Global Contact Properties, for the whole connection_group -->
    <contact>
      <partner part_index="1"/>
      <partner part_index="2"/>
      <coefficients static_friction="0.8"/>
    </contact>
    <contact>
      <partner part_index="2"/>
      <partner part_index="5"/>
      <coefficients static_friction="0.8"/>
    </contact>
    <contact>
      <partner part_index="5"/>
      <partner part_index="6"/>
      <coefficients static_friction="0.8"/>
    </contact>
  </contact_list>

  <connection_list>
    <connection_0d label="BOLT_135"> <!-- bolt without washers -->
      <loc> 84 60 10 </loc>
      <!-- Friction "head to first part" and "thread to nut": -->
      <threaded_connection length="50"
        static_friction="0.8"
        thread_static_friction="0.8">

        <bolt>
          <!-- Friction is "nut to last part" -->
          <nut diameter="16." static_friction="0.8"/>
        </bolt>
      </threaded_connection>

      <contact_list>    <!-- Local Contact definition, according to 5.3.2.5 -->
        <contact>
          <partner part_index="1"/>
          <partner part_index="2"/>
          <coefficients static_friction="0.9"/>
        </contact>
      </contact_list>

    </connection_0d>
  </connection_list>
</connection_group>
```

### 9.5.3 Definition of element <threaded\_connection/>

#### 9.5.3.1 General

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/>.

**Table 50 — Nested elements of <connection\_0d/> for <threaded\_connection/>**

Nested Elements	Multiplicity	Use	Constraint / Remarks
threaded_connection	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

**9.5.3.2 Element “loc”**

The syntax of this element is described in the corresponding section 9.1.2 Location.

**9.5.3.3 Element “appdata”**

This follows the syntax as defined in section 7.3.2 User Specific Data <appdata/>.

**9.5.3.4 Element “femdata”**

This follows the syntax as defined in section 7.3.3 Finite Element Specific Data <femdata/>.

**9.5.3.5 Element “threaded connection”**

XML specification of <threaded\_connection/> element with the following attributes:

**Table 51 — Attributes of element <threaded\_connection/>**

Attributes	Type	Value Space	Use	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 &gt; 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, thread form pitch is equal to lead. Default value is equal to pitch attribute!
Torque	Floating point	> 0.0	Optional	-

Attributes	Type	Value Space	Use	Constraints / Remarks
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	-
thread_static_friction	Floating point	> 0.0	Optional	
thread_kinetic_friction	Floating point	> 0.0	Optional	
strength_property_class	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	-

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 20;
- 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,  $\text{thread\_length} = \text{length} - \text{sink\_size}$ ;
- 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. (For more details, see e. g. English Wikipedia article "Screw thread", section "Lead, pitch, and starts");
- 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;
- thread\_static\_friction: The static friction between screw and cut thread, or bolt thread and nut thread;
- thread\_kinetic\_friction: The kinetic friction between screw and cut thread, or bolt thread and nut thread;

- `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:

**Table 52 — Nested elements of element `<threaded_connection/>`**

Nested Elements	Multiplicity	Use	Constraint
<code>normal_direction</code>	1	Optional	-
<code>tangential_direction</code>	1	Optional	-
<code>bolt</code> <code>screw</code>	1	Required	Exactly one of these elements.
<code>washer</code>	1	Optional	-

#### 9.5.3.6 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 9.1.3 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.

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.

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

#### 9.5.4 Washer

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

**Table 53 — Attributes of element `<washer/>`**

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

Attributes	Type	Value Space	Use	Constraints / Remarks
kinetic_friction	Floating point	> 0.0	Optional	-
strength_property_class	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	NO part code, if attached.

These attributes have following semantics:

- `outer_diameter`: the outer diameter of the washer. If a washer is used at all, at least its diameter has to 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;
- `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.

### 9.5.5 Nut

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

**Table 54 — Attributes of element `<nut/>`**

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

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 clinched, or it is a clip itself. It is clipped to the flange partner with this index (see section 7.4.2.2). 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 by welding or clinching to the flange partner with this index (see section 7.4.2.2). 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, such as punching or riveting.

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

**Table 55 — Nested elements of element `<nut/>`**

Nested Elements	Multiplicity	Use	Constraint
Washer	1	Optional	-

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

## 9.5.6 Bolt

### 9.5.6.1 General

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

### 9.5.6.2 Element "bolt"

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

**Table 56 — Attributes of element `<bolt/>`**

Attributes	Type	Value Space	Use	Constraint
<b>clipped_to</b>	Integer	> 0	Optional	-
<b>fixed_to</b>	Integer	> 0	Optional	-

The following list explains the attributes:

- `clipped_to`: The head of the bolt is fixed with a clip to the flange partner with this index (see section 7.4.2.2). If attribute is missing, bolt is not clipped. Bolt and clip share a common part code, which means they are regarded to be one single part.
- `fixed_to`: The head of the bolt is fixed (for instance, welded) to the flange partner with this index (see section 7.4.2.2). This also applies if there is no screw head at all, that means that this bolt actually is a fixed bolt, or a stud. If attribute is missing, the 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:

**Table 57 — Nested elements of element `<bolt/>`**

Nested Elements	Multiplicity	Use	Constraint
nut	1	Optional	

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 the definition of a `<bolt/>`. The nut itself (respectively its `part_code` or `property`) is not allowed to 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="BOLT_100532">
  <threaded_connection diameter="10.0" length="50.0"
    head_diameter="16.0" head_height="5" sink_size="3">
    <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="BOLT_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"/>
  </threaded_connection>
</loc> 1500.3 838.7 730.6 </loc>
<appdata>
  ...
</appdata>
</connection_0d>
```

```

    <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="BOLT_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 is part of the head, so it cannot have part code -->
    <washer outer_diameter="20" inner_diameter="10.3" thickness="1.5"
      attached="true"/>
  <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 Bolted joint with maximum parameter usage

```

<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<xmcf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="xmcf_3_1_1.xsd">
  <version> 3.1.1 </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="BOLT_135"> <!-- bolt with washers -->
        <loc> 84 60 10 </loc>
        <!-- Friction between "head to washer" and " thread and nut ": -->
        <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"
          thread_static_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>
</threaded_connection>

<contact_list>      <!-- friction between adjacent flange partners -->
  <contact>
    <partner part_index="1"/>
    <partner part_index="2"/>
    <coefficients static_friction="0.8"/>
  </contact>
  <contact>
    <partner part_index="2"/>
    <partner part_index="5"/>
    <coefficients static_friction="0.8"/>
  </contact>
  <contact>
    <partner part_index="5"/>
    <partner part_index="6"/>
    <coefficients static_friction="0.8"/>
  </contact>
</contact_list>

</connection_0d>
</connection_list>
</connection_group>
</xmcf>

```

### 9.5.6.3 Possible Bolt and Screw Assemblies

Altogether, there are following cases of assembly:

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

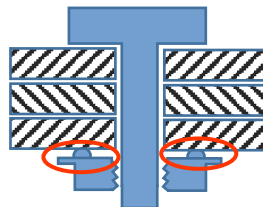


Figure 22 — Bolt with welded nut

#### EXAMPLE

```

<connection_0d label="BOLT_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 23 — Bolt with free nut

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

4. Screw (screwed to the last sheet):

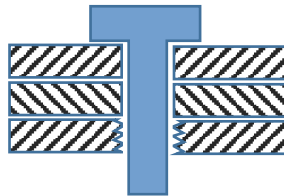


Figure 24 — Screw

#### EXAMPLE

```
<connection_0d label="SCREW_139">
  <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"/>
    <screw base="3"/>
  </threaded_connection>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

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

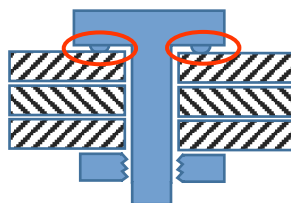


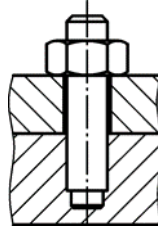
Figure 25 — Welded stud with free nut

#### EXAMPLE

```
<connection_0d label="BOLT_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>
```

```
</appdata>
</connection_0d>
```

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



**Figure 26 — Plain stud**

These studs are not a feature of  $\chi$ MCF version 3.1. They can be modelled according to case 4 but may become a topic of version 3.2.

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

## 9.5.7 Screw

### 9.5.7.1 General

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

### 9.5.7.2 Element "screw"

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

**Table 58 — Attributes of element `<screw/>`**

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

- `base`: the index (see section 7.4.2.2) 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 related nested elements, listed in following table:

**Table 59 — Nested elements of element `<screw/>`**

Nested Elements	Multiplicity	Use	Constraint
flow_drilled	1 - *	Optional	-

The subtypes are described in detail in sections below.

#### EXAMPLE A Screw without attributes

```
<connection_0d label="SCREW_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 and with washer)

```
<connection_0d label="SCREW_100532">
  <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>
```

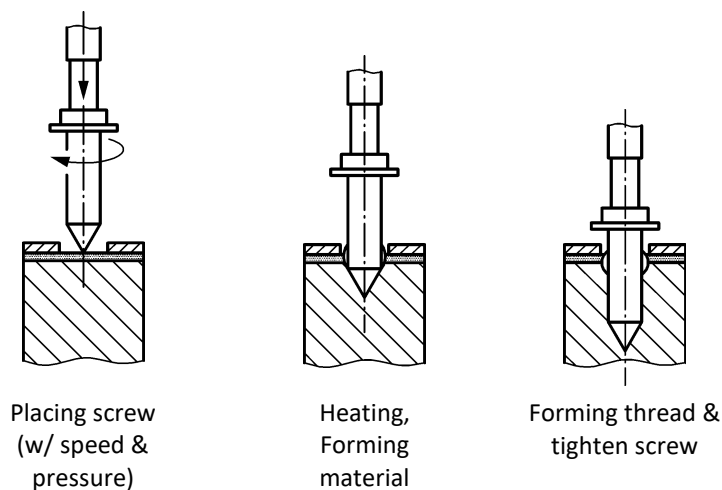
#### EXAMPLE C Screw with attributes but without washer

```
<connection_0d label="SCREW_100532">
  <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>
```

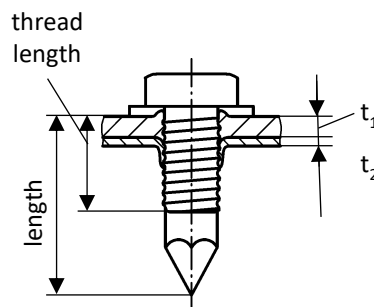
### 9.5.7.3 Flow Drilled Screws (FDS)

#### 9.5.7.3.1 General

Flow drilled screws are applied by a process called “friction drilling”.



**Figure 27 — Process of Flow Drill Screwing**



**Figure 28 — Measures of applied FDS**

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

- Applying rotational velocity and pressure;
- Tool heats target sheet metal (or without pre-punching both sheet component) and melts through it;
- Screw thread tapping;
- 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.

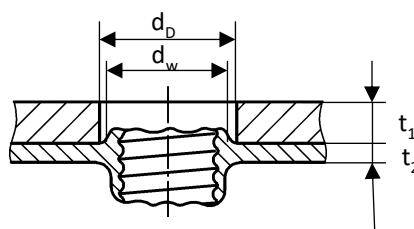
#### 9.5.7.3.2 Element "flow\_drilled"

For the <flow\_drilled/> element, the following attributes can be specified:

**Table 60 — Attributes of element <flow\_drilled/>**

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

- `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 clearance-hole ( $d_b$ ). Default value is 0.0, which means "no pre-machined hole or clearance hole".



**Figure 29 — FDS connection with pre-machined clearance hole**

- `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 7.4.2.2). 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.

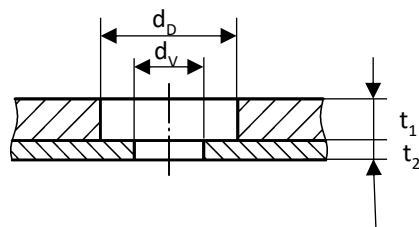


Figure 30 — Pilot hole on sheet metal

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

## EXAMPLE

```
<connection_0d label="FDS_96930">
  <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>
```

## 9.6 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.

Table 61 — Nested elements of `<connection_0d>` for `<gumdrop/>`

Nested Elements	Multiplicity	Use	Constraint
<code>gumdrop</code>	1	Optional	-
<code>loc</code>	1	Required	-
<code>appdata</code>	1	Optional	-
<code>femdata</code>	1	Optional	-
<code>custom_attributes_list</code>	1	Optional	See section 8.5 Custom Attributes list

XML specification of <gumdrop/> with following attributes:

**Table 62 — Attributes of element <gumdrop/>**

Attributes	Type	Value Space	Use	Constraint
diameter	Floating point	>= 0.0	Optional	-
mass	Floating point	>= 0.0	Optional	-
material	Alphanumeric	Alphanumeric	Optional	-

The following list explains the attributes:

- 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/>.

The element <gumdrop/> allows for following nested elements:

**Table 63 — Nested elements of element <gumdrop/>**

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

#### EXAMPLE

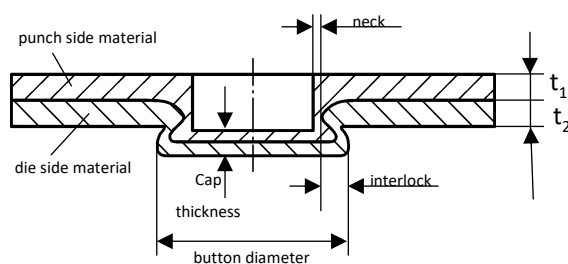
```
<connection_0d label="DROP_2123921">
  <!-- Assumed Unit system with mass attribute with value="kg" -->
  <gumdrop diameter="5.0" mass="0.0033" material="CAD_Material" />
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

## 9.7 Clinches

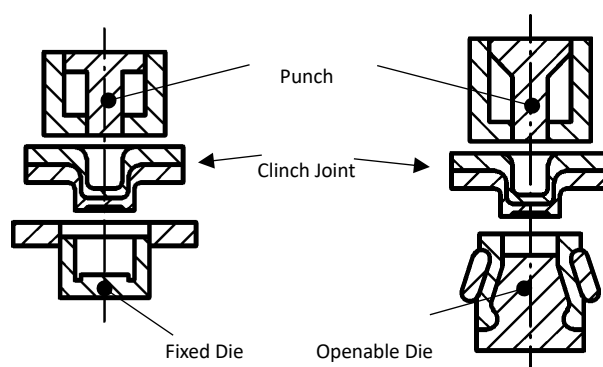
Clinching is a mechanical, cold forming fastening method to join sheet metal without additional components, using special tools to plastically form a mechanical interlock between the sheets.

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 31 — Clinch Joint Dimensions**



**Figure 32 — Two example clinch systems [5] (TOX (left) and BTM's Tog-L-Loc system)**

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 shall 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.

**Table 64 — Nested elements of `<connection_0d/>` for `<clinch/>`**

Nested Elements	Multiplicity	Use	Constraint
clinch	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list



XML specification of <clinch/> element:

**Table 65 — Attributes of element <clinch/>**

Attributes	Type	Value Space	Use	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"

The following list explains the attributes:

- **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 Lance-N-Loc [5] system. The main difference is that the TOX system uses a fixed die whereas the BTM system employs an extending die (see Figure 32). For more process and system details, refer to the documentation and website information of the specific clinch equipment supplier;
- **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 15 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;
- **button\_diameter**: The applied button diameter to create this joint. 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 aluminium). 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, hence, 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, which represents the direction in which metal is displaced. The element's definition can be found in section 9.1.3.

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

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

Table 66 — Nested elements of element &lt;clinch/&gt;

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

## 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>

```

## 9.8 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.

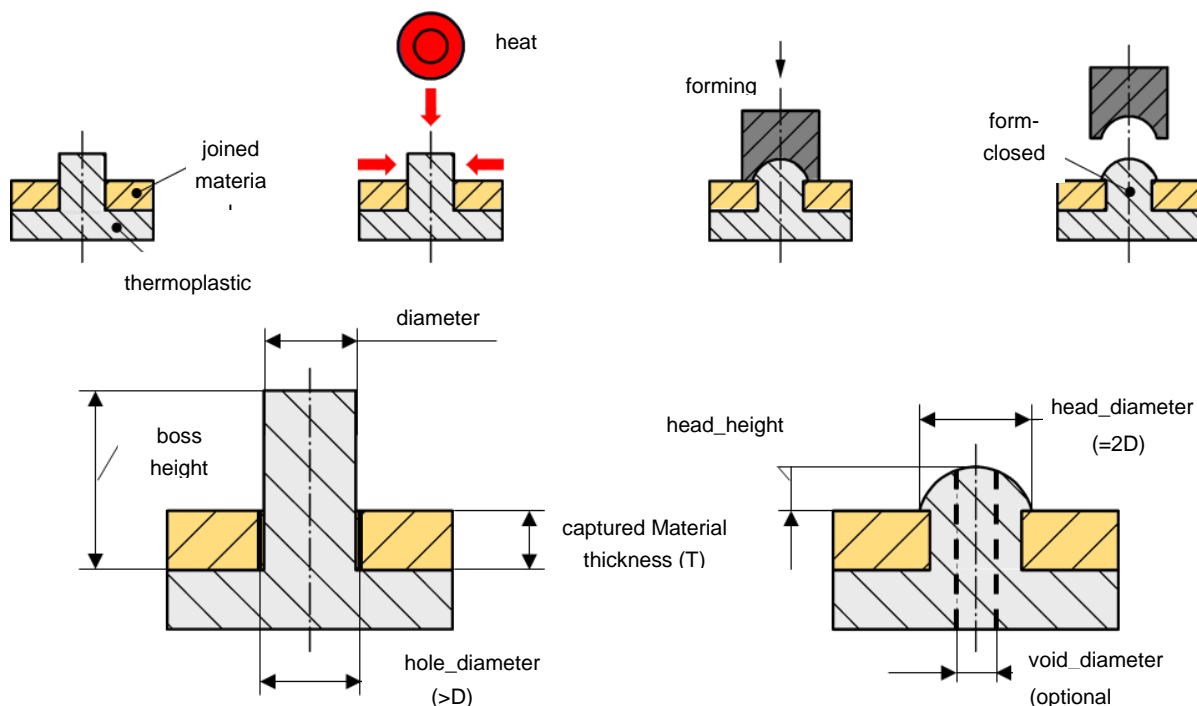


Figure 33 — Heat Stakes: Process steps and Design dimensions

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 shall describe them by OEM specific alphanumeric names (such as 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.

**Table 67 — Nested elements of `<connection_0d/>` for `<heat_stake/>`**

Nested Elements	Multiplicity	Use	Constraint
heat_stake	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of `<heat_stake/>` element:

**Table 68 — Attributes of element `<heat_stake/>`**

Attributes	Type	Value Space	Use	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

The following list explains the attributes:

- `heat_stake_type`: the alphanumeric name of the heat stake (such as 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 9.1.3.

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  $\chi$ MCF: Before tool application, it can be derived from CAD data. After tool application (in final shape of the heat stake), this height has vanished.

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

**Table 69 — Nested elements of element `<heat_stake/>`**

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

#### EXAMPLE

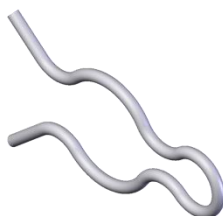
```
<connection_0d label="HEAT_STAKE_521">
  <heat_stake heat_stake_type="domed" diameter="3.0"
    head_diameter="6.0" head_height="2.25">
    <normal_direction x="0" y="0" z="-10"/>
  </heat_stake>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

## 9.9 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.

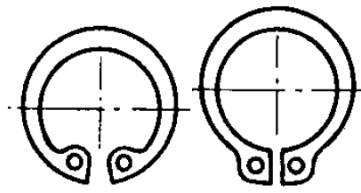
A wide and ever-increasing variety of clinches is in practical use. Examples are:

- 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 et cetera;
- Other clips slide onto a flat surface.



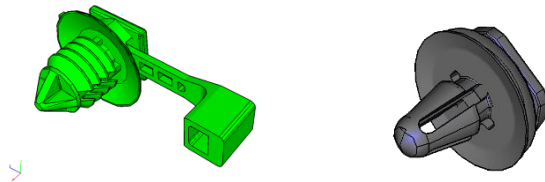
Source of image: Wikipedia

**Figure 34 — A "Hairpin Clip"**

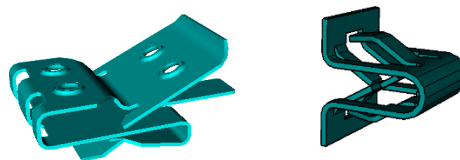


Source of image: Wikimedia

**Figure 35 — Internal and External Circlips**



**Figure 36 — Clips Pushed into a Hole**



**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.

**Table 70 — Nested elements of `<connection_0d/>` for `<clip/>`**

Nested Elements	Multiplicity	Use	Constraint
clip	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of `<clip/>` element:

**Table 71 — Attributes of element `<clip/>`**

Attributes	Type	Value Space	Use	Constraint
clip_type	Alphanumeric	Alphanumeric	Optional	-
attachment_type	Alphanumeric	Alphanumeric	Optional	-
hole_diameter	Floating point	$\geq 0.0$	Optional	-
hole_length	Floating point	$\geq 0.0$	Optional	hole_length > 0 implies hole_diameter > 0
pin_diameter	Floating point	$\geq 0.0$	Optional	-
pin_width	Floating point	$\geq 0.0$	Optional	pin_width > 0 implies pin_diameter > 0

Attributes	Type	Value Space	Use	Constraint
pin_length	Floating point	$\geq 0.0$	Optional	-
strap_length	Floating point	$\geq 0.0$	Optional	-
clipped_to	Integer	$> 0$	Optional	-
material	Alphanumeric	Alphanumeric	Optional	-
part_code	Alphanumeric	Alphanumeric	Optional	-

The following list explains the attributes:

- clip\_type: the alphanumeric name of the clip, for instance "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 36 — Clips Pushed into a Hole, left picture.), 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 7.4.2.2). 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 9.1.3.

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

**Table 72 — Nested elements of element `<clip/>`**

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

## EXAMPLE

```

<connection_0d label="CLIP_1001">
  <clip clipped_to="1" attachment_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>

```

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

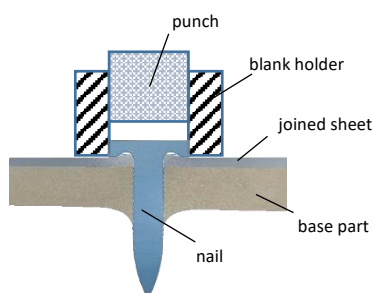


Figure 38 — Cross Section of a nail joint connecting two Sheets

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

Table 73 — Nested elements of `<connection_0d/>` for `<nail/>`

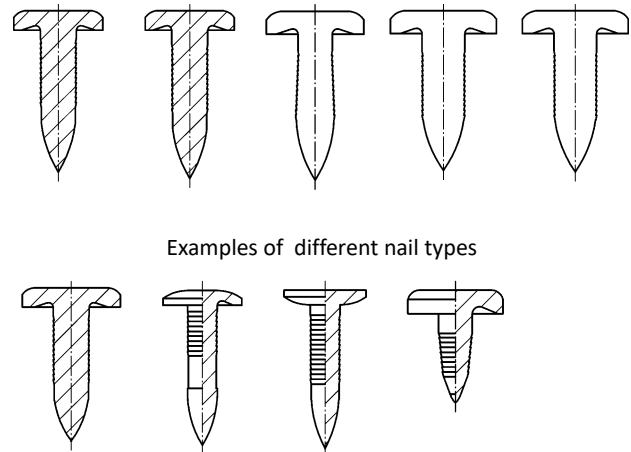
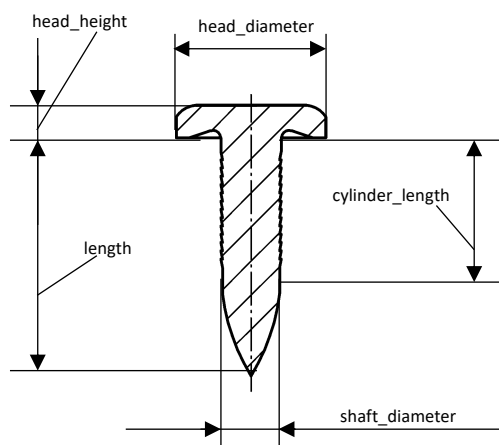
Nested Elements	Multiplicity	Use	Constraint / Remarks
nail	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of `<nail/>` element:

Table 74 — Attributes of element `<nail/>`

Attributes	Type	Value Space	Use	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	-

Attributes	Type	Value Space	Use	Constraint / Remarks
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	-



**Figure 39 — Key measures of a nail & examples of different nail types**

The following list explains the attributes:

- **nail\_type**: the alphanumeric name of the nail. (Naming convention based on supplier nail codes); for more details see [6];
- **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;
- **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, that is 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, hence, 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 9.1.3.



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

**Table 75 — Nested elements of element `<nail/>`**

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

#### EXAMPLE

```

<connection_0d label="NAIL_100">
  <nail shaft_diameter="10.0" length="26.0" head_diameter="15.0" material="steel"
shear_strength="5200" peel_strength="5000">
    <normal_direction x="0" y="0" z="-10"/>
  </nail>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>

```

## 9.11 Rotation Joints

### 9.11.1 General

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

**Table 76 — Nested elements of `<connection_0d/>` for `<rotation_joint/>`**

Nested Elements	Multiplicity	Use	Constraint / Remarks
rotation_joint	1	Optional	-
loc	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

XML specification of `<rotation_joint/>` element:

**Table 77 — Attributes of element `<rotation_joint/>`**

Attributes	Type	Value Space	Use	Constraint / Remarks
diameter	Floating point	> 0.0	Optional	-

The following list explains the attribute:

- diameter: the diameter of the shaft of the rotation joint.

If possible, a rotation joint should know the direction of fixation, hence, 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 the joint's head to point, which element's definition can be found in section 9.1.3.

Specific subtypes of rotation joints are defined by adding related nested elements, listed in following table:

Table 78 — Nested elements of element <rotation\_joint/>

Nested Elements	Multiplicity	Use	Constraint / Remarks
normal_direction	1	Optional	-
tangential_direction	1	Optional	-
rotav	1	Required	

The subtypes are described in detail in the following sections.

EXAMPLE

```
<connection_0d label="RJ_2123921">
...
  <rotation_joint diameter="3.0">
    <normal_direction x="0" y="0" z="3"/>
    <rotav/>
  </rotation_joint>
  <loc> 1645.83 821.145 616.585 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

9.11.2 ROTAV

ROTAVs are suitable for steel-aluminium connections. Joining of two or three sheets are possible. High grade steel sheets can be used. A description of this technology can be found in [7].

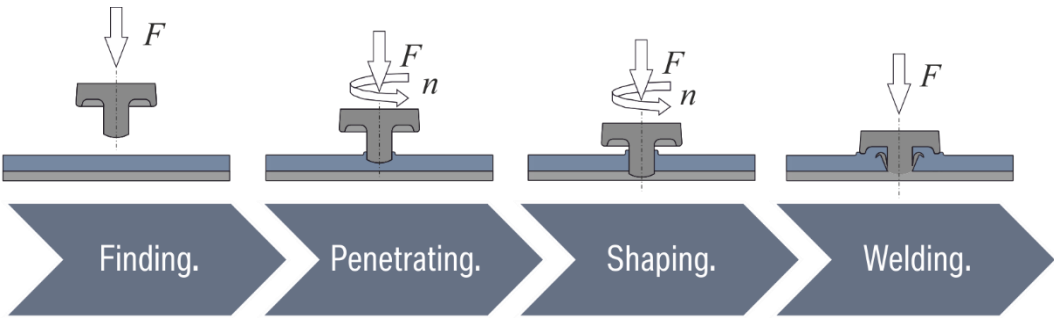


Figure 40 — Process of Rotation Joining (ROTAV) [7]

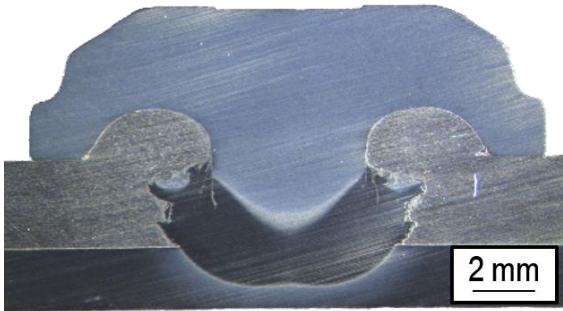


Figure 41 — ROTAV connecting aluminium and steel sheets [7]

The basic steps in the ROTAV process consist of:

- a. Applying rotational velocity and pressure to the ROTAV plug;
- b. ROTAV plug penetrates the soft aluminium sheet;
- c. ROTAV plug heats base sheet metal (or without pre-punching, both sheet components) and melts through it;
- d. Compression is applied to the ROTAV arrangement to finish the desired connection.

A ROTAV connection is denoted by an element `<rotav/>`.

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

**Table 79 — Attributes of element `<rotav/>`**

Attributes	Type	Value Space	Use	Constraint
rotational_speed	Floating point	$\geq 0.0$	Optional	-
compression_force	Floating point	$\geq 0.0$	Optional	-

The following list explains the attributes:

- `rotational_speed`: In order to facilitate the penetration in the metal sheet of the tip of the ROTAV, it is rotated at a high speed;
- `compression_force`: In order to achieve the fastening properties, the ROTAV is compressed with a vertical force.

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

**EXAMPLE A** Minimum definition of a “Rotav” connection

```
<connection_0d label="ROTAV_96930">
  <rotation_joint>
    <rotav/>
  </rotation_joint>
  <loc> 1500.3809 838.75885 730.6529 </loc>
</connection_0d>
```

**EXAMPLE B** Maximum definition of a “Rotav” connection

```
<connection_0d label="ROTAV_96930">
  <rotation_joint diameter="4.0">
    <rotav rotational_speed="1500000" compression_force="10000"/>
    <normal_direction x="0" y="0" z="-10"/>
  </rotation_joint>
  <loc> 1500.3809 838.75885 730.6529 </loc>
  <appdata>
    ...
  </appdata>
</connection_0d>
```

## 10 1D connections

### 10.1 Generic Definitions

#### 10.1.1 Identification

For identifying 1D connections, the same rules apply as for 0D connections, see section 9.1.1 identification.

#### 10.1.2 Location

##### 10.1.2.1 General

The definition of the connection line is one or multiple polylines. Each of the polylines is described as a series of points (vertices). All other curves can also be represented with this type of representation by adding necessary points and thus approximating to the needed accuracy.

The polylines do not need 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.

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

##### 10.1.2.2 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.

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

**Table 80 — Attributes of element `<loc_list/>`**

Attributes	Type	Use	Constraint
index	Integer	Optional	Required only if there are more than one <code>loc_list</code> elements in the <code>&lt;connection_1d/&gt;</code> .

A connection line with sharp corners can be expressed by a series of `<loc_list/>` elements. In this case, the `<loc_list/>` order is indicated by the `index` attribute.

NOTE Curves with sharp corners (for example, 90° angles) are not typically represented by a single curve in CAD systems. Using multiple `<loc_list/>` elements is suitable for representing such cases.

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

**Table 81 — Nested elements of `<loc_list/>`**

Nested Elements	Multiplicity	Use	Constraint
loc	1-*	Required	-

##### 10.1.2.3 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:

**Table 82 — Attributes of element `<loc/>`**

Attributes	Type	Use	Constraint
v	Floating point	Required	-

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 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>
<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>
```

## 10.1.3 Intermittent Connection Lines

### 10.1.3.1 General

Intermittent connection lines are connection lines, which are fixed only at certain *segments* along their total length. Here, the word *segment* shall not be confused with polygon edges, which were named segments in section 10.1.2, too. The gaps between the *segments* are called *spacings* to avoid confusion with the gap between the connected parts. The benefit of intermittent connection lines compared with individual connection lines is the reduction of administrative overhead.

Intermittent connection lines were introduced with  $\chi$ MCF version 3.1.1 and are only applicable to *seam welds*, currently.

The *total length* of a connection line is the length of the `<loc_list/>` polygon. That is, the total length contains the lengths of *both*, the *segments* and the *spacings* between, before and after segments.

The `<loc_list/>` polygon only *approximates* exact geometry. This can lead to unavoidable deviations between the length of both, and hence to the exact positions of segments, especially next to the end of a connection line. Thus, the reliable definition of intermittent connection lines requires a certain *accuracy* of this polygon. Additionally, the parameters describing the segmentation shall be *consistent* in the sense that the segmentation is feasible both, geometrically and with respect to manufacturing. It is *not* within

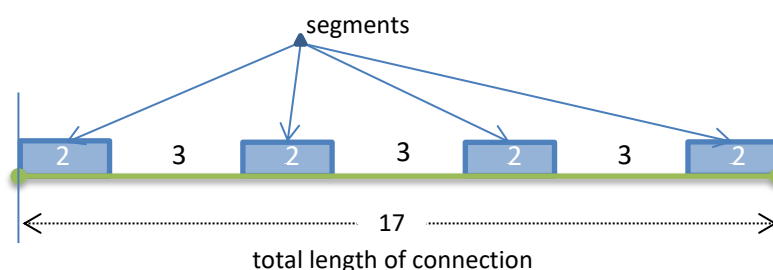
the scope of the  $\chi$ MCF format to take these responsibilities since additional external information would be required.

From applications like durability and fatigue, it is known that the beginnings and ends of a weld line are most crucial. Hence, it shall be guaranteed as far as possible that there exist *complete* segments. But in the end, it is the responsibility of the  $\chi$ MCF creating system that chopped final segments do not occur.

Therefore, following rules apply:

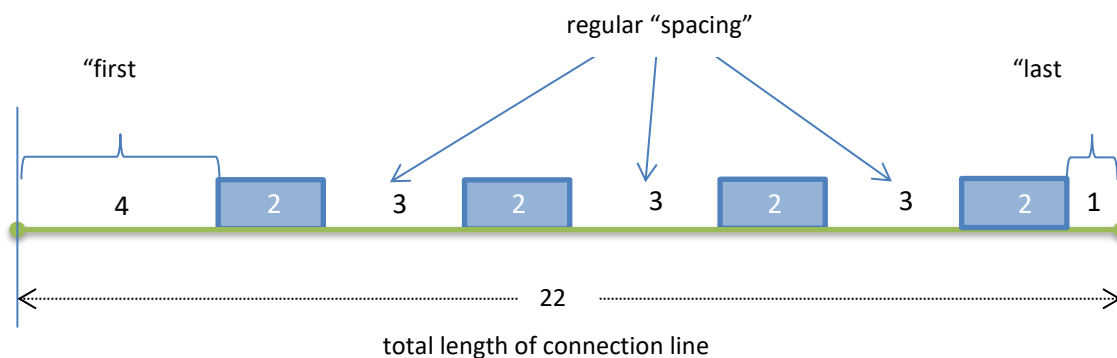
1. *Master rule*: The creating system alone is responsible for accurate and consistent definition of the segments;
2. If it is required that any segment length (especially first or last) deviates from other segment lengths, a <segment\_list/> has to be used. <regular\_segments/> are *not* intended to provide this feature;
3. Excess of segments at the end of a seam weld is not allowed.

### 10.1.3.2 Terminology:



**Figure 42 — Terminology of a regular intermittent weld**

In the example above, the connection line has a '**total length**' of 17.0. Its '**number of segments**' is 4. Each segment is of '**length**' 2.0. The welded segments have a '**spacing**' of 3.0. Note that the first and last segments match the start and end of the connection line.



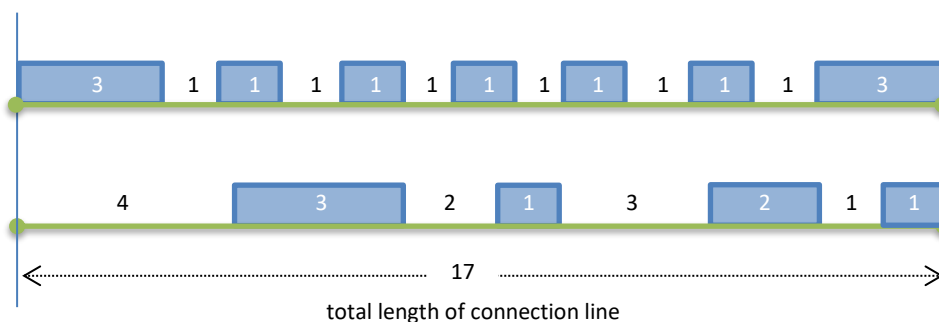
**Figure 43 — Regular intermittent weld with first spacing and last spacing**

In the above diagram, the welded segments have a special '**first spacing**' of 4.0 and a '**last spacing**' of 1.0, at the beginning and end of the connection line, respectively. Note that '**spacing**' is the gap between successive welds, in contrast with the gap at the begin and end of the connection line.

The '**density**'  $d$  of the welded portion of the weld is defined as

$$d := \frac{l}{l + s}$$

where  $l$  is the length, and  $s$  is the spacing, as defined above. For the example above, the density of the welded line is  $2/5$ .



**Figure 44 — Irregular intermittent welds**

The intermittent welds in the above diagram, are *not* regular ones. Therefore, they are treated as `<segment_list/>`, where each segment is described separately. When all welded segments have the same length and all gaps between segments have the same spacing, the connection can be represented as sequence of `<regular_segments/>`.

**Summary:** For the description of an intermittent connection line, following variants are available:

- `<segment_list/>`: All segments are specified *individually* with start and end given in curve length parameters of the `<loc_list/>` polygon;
- `<regular_segments/>`: All segments have *identical length*. All spacings have *identical length* except for a *first spacing* at the beginning of the `<loc_list/>` polygon (that is before the first segment) and a *last spacing* at the end of the `<loc_list/>` polygon (that is after the last segment).

The element `<segment_list/>` can only be used *mutually exclusively* to `<regular_segments/>` element. That means, only *one* of these elements may occur in one `<weld_position/>` element.

XML specification of `<segment_list/>` element:

The `<segment_list/>` element does *not* have any attributes, but at least one nested element `<segment/>`.

XML specification of `<segment/>` element (with  $L_{total}$  := total length of the `<loc_list/>` polygon):

**Table 83 — Attributes of element `<segment/>`**

Attributes	Type	Value Space	Use	Constraint
from	Floating point	$\geq 0$	Required	from < to
to	Floating point	$> 0$	Required	

If there are more than one segments in `<segment_list/>`, it is required that all segments  $s_n$  can be arranged in a way that  $s_n.to < s_{n+1}.from$ .

XML specification of `<regular_segments/>` element (with  $L_{total}$  := length of the `<loc_list/>` polygon and  $n$  := number of segments, both positive):

**Table 84 — Attributes of element `<regular_segments/>`**

Attributes	Type	Value Space	Use	Constraint
num_segments	Integer	$> 0$	Required	
length	Floating point	$> 0$	Required	
spacing	Floating point	$> 0$	Required	

Attributes	Type	Value Space	Use	Constraint
first_spacing	Floating point	$\geq 0.0$ (default: 0)	Optional	
last_spacing	Floating point	$\geq 0.0$ (default: 0)	Optional	
keep	Selection	spacing, length, density (default)	Optional	<p>If both attributes are missing, default of "max_percentage_of_compensation" is used.</p> <p>Only one of "max_absolute_compensation" or "max_percentage_of_compensation" may be specified.</p>
max_percentage_of_compensation	Floating point	$> 0.0$ and $\leq 100.0$ (default: 1.0)	Optional	
max_absolute_compensation	Floating point	$\geq 0.0$	Optional	

Description of `<regular_segments/>` requires parameters with specific semantics, as listed in Table 84:

- `num_segments`: Prescribed number of welded segments;
- `length`: Prescribed length of every segment;
- `spacing`: Prescribed length of any *inner* spacing, a spacing between two segments;
- `first_spacing`: Length of the spacing before the first segment, if any;
- `last_spacing`: Length of the spacing after the last segment, if any;
- `keep`: Strategy about how to cope with the case that all prescribed segments and spacings together differ from the total length of the `<loc_list/>` polygon;
- `max_percentage_of_compensation`: The maximum allowable deviation, as a percentage, of the resulting size of length or spacing over its prescribed size. A warning has to be issued, if adjusted value deviates from prescribed value by more than `max_percentage_of_compensation`. Valid range is from 0.0 to 100.0 percent;
- `max_absolute_compensation`: The maximum allowed deviation, in length units, of the difference between the resulting size of length or spacing and its prescribed size. A warning has to be issued, if adjusted value deviates from prescribed value by more than `max_absolute_compensation`.

Semantics of the different possible values of `keep` parameter:

- `spacing`: Spacing between segments is kept. Length is adjusted;
- `length`: Segment lengths are kept. Spacing between segments is adjusted;
- `density`: Effective density  $d$  is kept. This implies that both, segment lengths and spacing absorb the change proportionally, but `first_spacing` and `last_spacing` remain unchanged.

In any case, the number of segments is kept unchanged.



### 10.1.3.3 Formulae for adjusting the segment sizes according to the total length of the connection line

The welded segments in a connection line are spread over the **free area** between the margins. The size of the free area is given by (Details are given in Annex A):

$$L = L_{total} - m_{first} - m_{last} \quad \text{where} \quad \begin{cases} L_{total} : \text{the total length of the polyline} \\ m_{first} : \text{first\_spacing} \\ m_{last} : \text{last\_spacing} \end{cases}$$

The **number of segments**,  $n$ , is given by attribute `num_segments`.

NB. the **number of spacings** is always  $n-1$ .

— When **keep** = "length", the **adjusted spacing** is calculated with this formula:

$$\bar{s} = \frac{L - nl}{n - 1}$$

— When **keep** = "spacing", the **adjusted length** is calculated with this formula:

$$\bar{l} = \frac{L - (n - 1)s}{n}$$

— When **keep** = "density", the **adjusted length** and **adjusted spacing** are given by these formulae:

$$\bar{l} = \frac{dL}{n - 1 + d}$$

and

$$\bar{s} = \frac{(1 - d)L}{n - 1 + d}$$

where

$d$ : *prescribed density*, calculated by  $d = \frac{l}{l+s}$ .

EXAMPLE A `<corner_weld/>` with `<regular_segments/>` and "Required" attributes only.

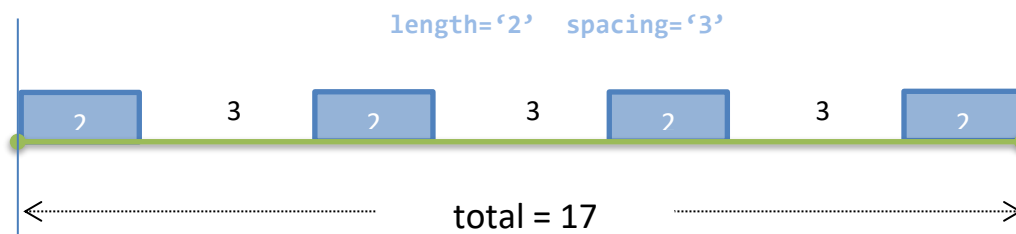


Figure 45 — `<corner_weld/>` with `<regular_segments/>` and "Required" attributes only

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0.2" x="1" y="0" z="1">
      <regular_segments num_segments="4" length="2" spacing="3"/>
    </weld_position>
    <sheet_parameter ... />
  </corner_weld>
</seamweld>
```

EXAMPLE B Regular single sided welding (a `<corner_weld/>` with `<regular_segments/>` and all attributes)

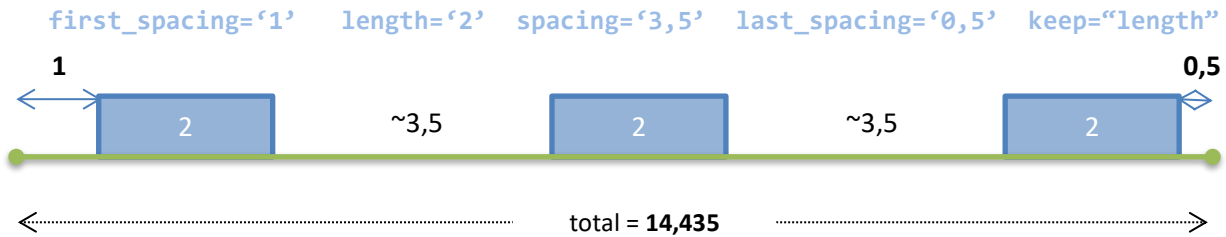


Figure 46 — Regular single sided welding (a <corner\_weld/> with <regular\_segments/> and all attributes)

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0.2" x="1" y="0" z="1">
      <regular_segments
        num_segments="3"
        first_spacing="1.0" last_spacing="0.5" length="2.0" spacing="3.5"
        keep="length" max_absolute_compensation="0.2"/>
      </weld_position>
      <sheet_parameter ... />
    </corner_weld>
  </seamweld>

...

<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0.2" x="1" y="0" z="1">
      <regular_segments
        num_segments="3"
        first_spacing="1.0" last_spacing="0.5" length="2.0" spacing="3.5"
        keep="length" max_percentage_of_compensation="3.0"/>
      </weld_position>
      <sheet_parameter ... />
    </corner_weld>
  </seamweld>
```

EXAMPLE C Staggered welding (a <corner\_weld/> welded from both sides in alternating sequence, with two <regular\_segments/> for the two <weld\_position/>s)

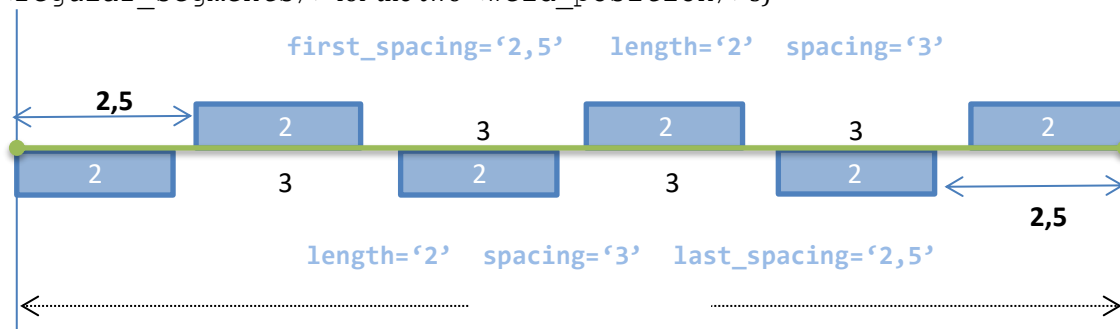


Figure 47 — Staggered welding

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0.2" x="1" y="0" z="1">
      <regular_segments
        first_spacing="2.5" num_segments="3" spacing="3.0" length="2.0"/>
      </weld_position>
    <weld_position u="0.5" x="-1" y="0" z="1">
      <regular_segments
        last_spacing="2.5" num_segments="3" spacing="3.0" length="2.0"/>
      </weld_position>
      <sheet_parameter ... />
    </corner_weld>
  </seamweld>
```

EXAMPLE D Definition of a <corner\_weld/> with <segment\_list/>

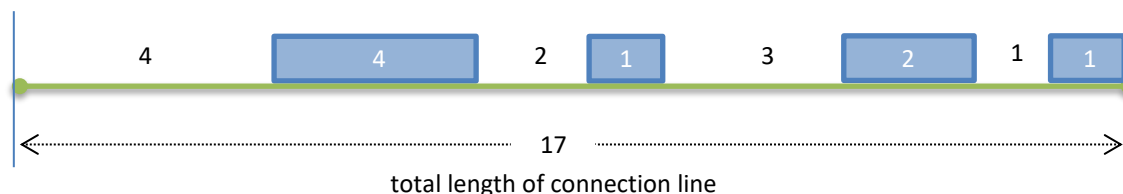


Figure 48 — Definition of a <corner\_weld/> with <segment\_list/>

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0.2" x="1" y="0" z="1">
      <segment_list>
        <segment from="4.0" to="7.0" />
        <segment from="9.0" to="10.0" />
        <segment from="13.0" to="15.0" />
        <segment from="16.0" to="17.0" />
      </segment_list>
    </weld_position>
    <sheet_parameter ... />
  </corner_weld>
</seamweld>
```

NOTE The order of <segment/> lines is arbitrary since segments are not allowed to overlap.

#### 10.1.4 Type Specification

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

Table 85 — Nested elements of element <connection\_1d/>

Nested Elements	Multiplicity	Use	Constraint / Remarks
seamweld	1	Optional	-
adhesive_line	1	Optional	-
hemming	1	Optional	-
sequence_connection_0d	1	Optional	-
contact_list	1	Optional	See section 7.4.3.6.
stacking	1	Optional	See section 7.4.2.4

Only one of the elements (seamweld, adhesive\_line, hemming, sequence\_connection\_0d) shall exist in a <connection\_1d/>. If none of the type elements exist, then this will default to <seamweld/>.

## 10.2 Seam Welds

### 10.2.1 Description and Modelling Parameters

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

The description of seam welds made up from different modelling 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 modelling.

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

**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.  $\chi$ MCF 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.

### 10.2.2 Seam Weld Definition Overview

The weld definition depends on the type of the weld. For each of the different weld 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;
- Widely used weld sections for the respective weld type  
(other section are generally enabled by the standard, but feasibility and compatibility have to be ensured by the designer);
- 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  $\chi$ MCF file contents. They are used in the weld specific sections to describe parameters stored in the  $\chi$ MCF file and their relations.

The variety is to be handled by the application using the  $\chi$ 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.

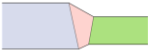
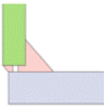


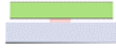



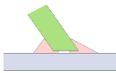


Weld Type	# Weld Positions	Welding Technology	Section	Weld Parameter			Layout
				Required	Optional	Fixed	
Butt Weld	1		I	width	-	-	
			V	width	-	-	
			U	width	-	-	
			X	width	-	-	
			Y	width	-	-	
			Radius		-	-	
Corner Weld	1-2	Fillet	Fillet	thickness	penetration=0, gap=0, angle=45	-	
			HV	thickness	gap=0, angle=45	penetration=1	
			U	thickness	penetration=0, gap=0, angle=45		
	2-4	Fillet	Fillet	thickness	penetration=0, gap=0, angle=45	-	
			HV	thickness	gap=0, angle=45	penetration=1	
			U	thickness	penetration=0, gap=0, angle=45		
Edge Weld	1		I	width	gap=0	-	
	1		V			-	
	1		U			-	
I-Weld	1	Laser	-	width	gap=0	-	
	1	Fillet	U			-	
Overlap Weld	1	Fillet	-	thickness	penetration=0, gap=0, angle=45	-	
	2	Fillet	-	thickness	penetration=0, gap=0, angle=45	-	
	2	Fillet	-	thickness	penetration=0, gap=0, angle=45	-	
Y-Joint	1-2	Fillet	Fillet	thickness	penetration=0, gap=0, angle=45	-	
			HV	thickness	gap=0, angle=45	penetration=1	
			HY	thickness	penetration=0, gap=0, angle=45	-	
K-Joint	2-3	Fillet	Fillet	thickness	penetration=0, gap=0, angle=45	-	
			HV	thickness	gap=0, angle=45	penetration=1	
			HY	thickness	penetration=0, gap=0, angle=45	-	
Cross-Joint	2-4	Fillet	Fillet	thickness	penetration=0, gap=0, angle=45	-	
			HV	thickness	gap=0, angle=45	penetration=1	
			HY	thickness	penetration=0, gap=0, angle=45	-	

Figure 49 — Seam weld types and attributes

### 10.2.3 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>
      <loc v="0">68 0 0</loc>
      <loc 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>
            <loc u="0">68 4 10</loc>
            <loc 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 50 — xMCF Structure of a Seam Weld (connection\_1d)

### 10.2.4 Generic Seam Weld Definition

#### 10.2.4.1 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.

##### 10.2.4.1.1 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:

Table 86 — Nested elements of element `<seamweld/>`

Nested Elements	Multiplicity	Use	Constraint
butt_joint	1	Optional	-
corner_weld	1	Optional	-
edge_weld	1	Optional	-
i_weld	1	Optional	-

Nested Elements	Multiplicity	Use	Constraint
overlap_weld	1	Optional	-
y_joint	1	Optional	-
k_joint	1	Optional	-
cruciform_joint	1	Optional	-
flared_joint	1	Optional	-

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.

#### 10.2.4.1.2 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 (not be confused with cross section "I", cf. section 10.2.4.4!);
- overlap\_weld;
- y\_joint;
- k\_joint;
- cruciform\_joint;
- flared\_joint.

Each subtype element can contain the following attributes:

**Table 87 — Attributes of element <subtype/>**

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

Each subtype element contains the following nested elements:

**Table 88 — Nested elements of element <subtype/>**

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

NOTE The number of elements of <weld\_position/> is dependent on the specific subtype.

#### 10.2.4.1.3 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.

#### 10.2.4.1.4 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 (laser, for example);
- Friction welding;
- Brazing (not allowed for I-Welds, for technical reasons).

In addition 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 Main type as seamweld with butt\_joint as subtype

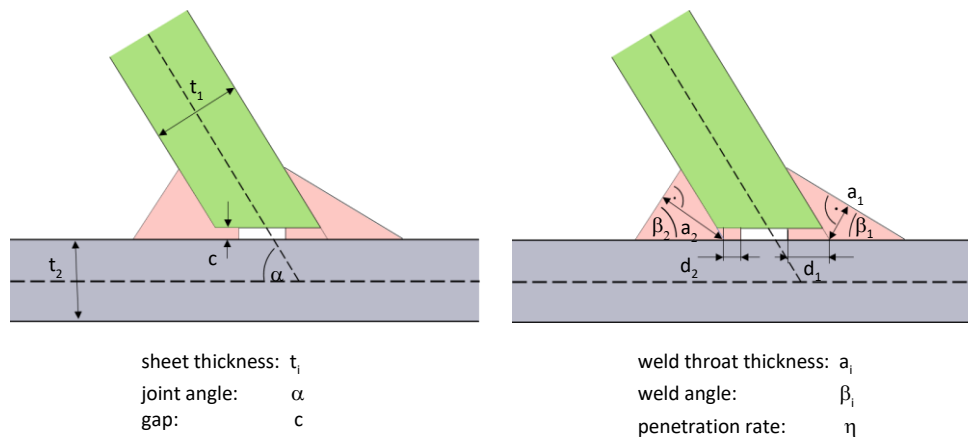
```
<connection_id>
  <seamweld>
    <butt_joint base="1" technology="resistance">
      ...
      <weld_position ... />
      <sheet_parameter ... />
      ...
    </butt_joint>
  </seamweld>
</connection_id>
```



### 10.2.4.2 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 to sheet metal parts.

On the other hand, we can distinguish the parameters that are mentioned in terms of the welding process related to the weld itself. The detailed description of these parameters can be seen for Sheet Parameters in section 10.2.4.3 and for Weld Position Parameters in section 10.2.4.4.



**Figure 51 — Sheet Parameters vs. Weld Position Parameters**

### 10.2.4.3 Parameters Assigned to a Specific Sheet of the Flange

#### 10.2.4.3.1 General

In a welded connection there are different kinds of parameters that must 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/>`.

#### 10.2.4.3.2 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 an attribute the corresponding gap applied between the welded sheet and the base sheet, which is in general the applied gap between the welded sheets involved in the welding process.

It is defined using the following attributes:

**Table 89 — Attributes of element `<sheet_parameter/>`**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <code>&lt;part/&gt;</code> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

#### 10.2.4.3.3 Attribute "index"

The value of the attribute `index` shall 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.

#### 10.2.4.3.4 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.

#### 10.2.4.3.5 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 51 this is  $t_2$ ). In case of more than 1 welded sheet exist see the definition example in 10.2.11.6.

#### 10.2.4.3.6 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

```
<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>
```

### 10.2.4.4 Welding Position

#### 10.2.4.4.1 Basic Definitions

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 52).

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 section (see also 10.2.5).

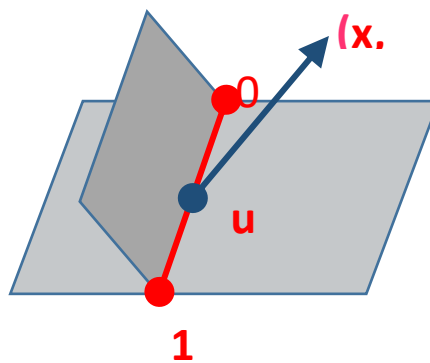


Figure 52 — Welding Position of a Y-Joint

#### 10.2.4.4.2 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.

#### 10.2.4.4.3 Element "weld\_position"

The element `<weld_position/>` describes the location of the weld relative to the connection line specified in `loc_list`.

Each `<weld_position/>` element can contain the following nested elements:

Table 90 — Nested elements of element `<subtype/>`

Nested Elements	Multiplicity	Use	Constraint
segment_list	0 - 1	Optional	mutually exclusive - For details, see section 10.1.3 Intermittent Connection Lines.
regular_segments	0 - 1	Optional	

The element `<weld_position/>` is defined using the following attributes:

Table 91 — Attributes of element `<weld_position/>`

Attributes	Type	Use	Constraint / Remarks
base	Integer	Optional	Value only for specific weld types
u	Floating point	Required	$0 \leq u \leq 1$
x	Floating point	Required	-
y	Floating point	Required	-
z	Floating point	Required	-
reference	Boolean	Optional	"false"
section	Selection	Optional	-
thickness	Floating point	Optional	Value only for specific weld types
width	Floating point	Optional	Value only for specific weld types
angle	Floating point	Optional	-
filler	Selection	Optional	-

Attributes	Type	Use	Constraint / Remarks
filler_material	Alphanumeric	Optional	-
shape	Selection	Optional	-
penetration	Floating point	Optional	$0 \leq \text{penetration} \leq 1$

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

```

<connection_ld>
  <seamweld>
    <corner_weld base="1" technology="resistance">
      <weld_position u="0.2" x="1" y="0" z="1"
        reference="true"
        section="HV"
        thickness="0.5"
        angle="45"
        filler="yes"
        filler_material=" E7018-X"
        shape="straight"
        penetration="0.6"/>
      <sheet_parameter index="2" gap="1.0"/>
    </corner_weld>
  </seamweld>
</connection_ld>

```

#### 10.2.4.4.4 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 53.

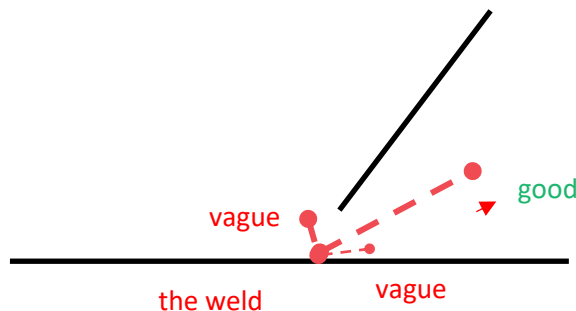


Figure 53 — Welding Position vector direction and length

#### 10.2.4.4.5 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).

#### 10.2.4.4.6 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".

Widely used values are:

- I;
- V;
- U;
- X;
- Y;
- HV;
- HY;
- Fillet;
- Radius.

#### 10.2.4.4.7 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.

**Remark:** Section "I" is not be confused with seam weld subtype "`i_weld`" (cf. section 10.2.4.1 Type Specification)!

#### 10.2.4.4.8 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.

#### 10.2.4.4.9 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.

#### 10.2.4.4.10 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.

#### 10.2.4.4.11 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.

**10.2.4.4.12 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.

**10.2.4.4.13 Section "HY"**

The section "HY" describes 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.

**10.2.4.4.14 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.

**10.2.4.4.15 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.

**10.2.4.4.16 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.

**10.2.4.4.17 Attribute "width"**

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

**10.2.4.4.18 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.

**10.2.4.4.19 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:

**Table 92 — Default values of attribute "filler", dependent from attribute "technology"**

Attribute value "technology"	Default value "filler"
resistance	Yes
arc	Yes
laser	No

#### 10.2.4.4.20 Attribute "filler\_material"

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

#### 10.2.4.4.21 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 for instance a "segment of a circle", "parabolic" etc., nor how big the deviation from straight shape is.

#### 10.2.4.4.22 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/>` (for example important for K-joints). If all `<weld_position/>` at the same welded sheet have a sum of penetration  $\geq 1$ , there is no open (unfilled) gap between the base sheet and the welded sheet.

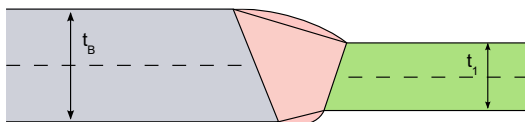
### 10.2.5 Butt Joint

#### 10.2.5.1 General

The principles of the modelling of Butt Joints for  $\chi$ MCF 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.

#### 10.2.5.2 Sheet Parameters

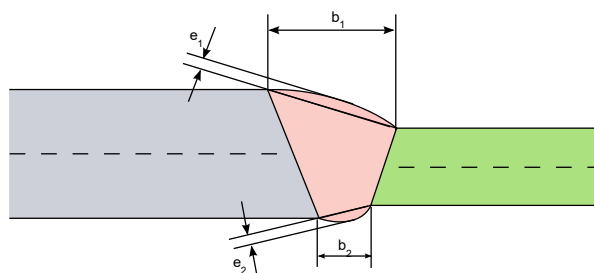


The parameters to describe the connection are:

- $t_B$                       Thickness of base sheet;
- $t_1$                       Thickness of welded sheet.

**Figure 54— Butt Joint Sheet Layout**

### 10.2.5.3 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 55— Butt Joint Weld parameters**

Inside the  $\chi$ MCF File the following parameters can be specified:

**Table 93 — Parameters of Butt Joint Weld**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
b	width	1 – 2	$\geq 0$	Optional	-
e	-	(1 – 2)	( $\geq 0$ )	(Optional)	(0)

NOTE The reinforcement is currently not defined as attribute in  $\chi$ MCF version 3.1.

### 10.2.5.4 Attributes

#### 10.2.5.4.1 Attribute "base"

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

#### 10.2.5.4.2 Attribute "technology"

The following list explains the attributes. The value for the attribute `technology` can be specified using the following values:

- `resistance`;
- `arc`;
- `laser` (energy beam / laser) ;
- `friction`;
- `brazing`.

### 10.2.5.5 Element "weld\_position"

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



**Table 94 — Attributes of element <weld\_position/> for Butt Joint**

Attributes	Type	Use	Constraint
u	Floating point	Required	$0 \leq u \leq 1$
x	Floating point	Required	-
y	Floating point	Required	-
z	Floating point	Required	-
reference	Boolean	Optional	"false"
section	Selection	Optional	-
width	Floating point	Optional	-
filler	Selection	Optional	-
filler_material	Alphanumeric	Optional	-

**10.2.5.5.1 Attributes "u, x, y, z, reference"**

Detailed definition can be found in section 10.2.4.4 Welding Position.

**10.2.5.5.2 Attribute "section"**

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

- I (not be confused with seam weld subtype "`i_weld`" (cf. section 10.2.4.1!) ;
- U;
- V;
- X;
- Y;
- Radius.

**10.2.5.5.3 Attribute "width"**

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

**10.2.5.5.4 Attribute "filler"**

Valid values for the attribute `filler` can be:

- yes;
- no.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

**10.2.5.5.5 Attribute "filler\_material"**

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

EXAMPLE A <weld\_position/> with required attributes only

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

```
</butt_joint>
</seamweld>
```

EXAMPLE B `<weld_position/>` with all attributes

```
<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>
```

### 10.2.5.6 Element "sheet\_parameter"

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

**Table 95 — Attributes of element `<sheet_parameter/>` for Butt Joint**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <code>&lt;part/&gt;</code> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

EXAMPLE `<sheet_parameter/>` with all 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>
```

## 10.2.6 Corner Weld

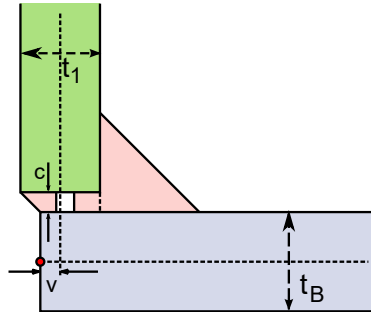
### 10.2.6.1 General

The principles of the modelling of corner welds for  $\chi$ MCF 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.

### 10.2.6.2 Simple Corner Weld

#### 10.2.6.2.1 Sheet Parameters

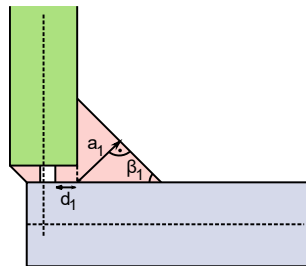


The parameters to describe the connection are:

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

**Figure 56— Corner Weld Sheet Layout**

#### 10.2.6.2.2 Weld Parameters



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

- $a_i$  Thickness of the weld (a-value, throat);
- $d_i$  Depth of the penetration;
- $\beta_i$  Weld angle.

**Figure 57— Corner Weld Parameters**

For the penetration, the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ MCF 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. ( $\alpha_j$  in case of a Corner Weld is  $90^\circ$  and therefore  $\sin \alpha_j = 1$ .)

Inside the  $\chi$ MCF File the following parameters can be specified:

Table 96 — Parameters of Simple Corner Weld

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	1 – 2	$\geq 0$	Optional	
$\beta$	angle	0 – 2	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

All other parameters are provided by the model itself.

10.2.6.3 Double Corner Weld

10.2.6.3.1 Sheet Parameters

The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1, t_2$  Thicknesses of welded sheet;
- $c_1, c_2$  Gaps between base sheet and welded sheet;
- $v$  Misalignment of welded sheet.

10.2.6.3.2 Weld Parameters

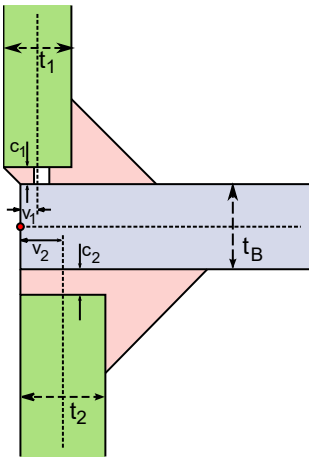


Figure 58 — Corner Weld Sheet Layout

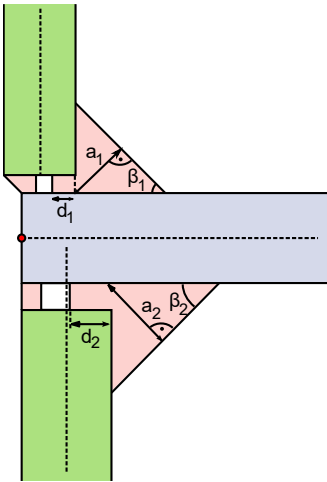


Figure 59 — Double Corner Weld Parameters

The parameters of the welds are the same for all the potential welds on the connection (applies to both figures above):

- $a_i$  Thickness of the weld (a-value, throat);
- $d_i$  Depth of the penetration;
- $\beta_i$  Weld angle.

For the penetration, the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ MCF 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. ( $\alpha_j$  in case of a Corner Weld is  $90^\circ$  and therefore  $\sin \alpha_j = 1$ .)

Inside the  $\chi$ MCF File the following parameters can be specified:

**Table 97 — Parameters of Double Corner Weld**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	1 – 2	$\geq 0$	Optional	
$\beta$	angle	0 – 2	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

All other parameters are provided by the model itself.

## 10.2.6.4 Attributes

### 10.2.6.4.1 Attribute "base"

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

### 10.2.6.4.2 Attribute "technology"

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

- resistance;
- arc;
- laser (energy beam / laser) ;
- friction;
- brazing.

### 10.2.6.5 Element "weld\_position"

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

**Table 98 — Attributes of element `<weld_position/>` for Corner Weld**

Attributes	Type	Use
u	Floating point	Required
x	Floating point	Required
y	Floating point	Required
z	Floating point	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Floating point	Optional
angle	Floating point	Optional
shape	Selection	Optional

Attributes	Type	Use
penetration	Floating point	Optional
filler	Selection	Optional
filler_material	Alphanumeric	Optional

#### 10.2.6.5.1 Attributes "u, x, y, z, reference"

Detailed definition can be found in section 10.2.4.4 Welding Position.

#### 10.2.6.5.2 Attribute "section"

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

- HV;
- U;
- Fillet.

#### 10.2.6.5.3 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:

**Table 99 — Values of Attribute `section`**

Attribute value "section"	Attribute "thickness"
HV	Optional
U	Not allowed
Fillet	Required

#### 10.2.6.5.4 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:

**Table 100 — Values of Attribute `angle`**

Attribute value "section"	Attribute "angle"
HV	Optional
U	Not allowed
Fillet	Required

#### 10.2.6.5.5 Attribute "shape"

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

#### 10.2.6.5.6 Attribute "penetration"

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

#### 10.2.6.5.7 Attribute "filler"

Valid values for the attribute filler can be:

- yes;
- no.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

#### 10.2.6.5.8 Attribute "filler\_material"

The attribute filler\_material specifies the applied material during the welding process.

EXAMPLE Definition of a corner\_weld with all attributes for the weld\_position

```
<seamweld>
  <corner_weld base="1" technology="resistance">
    <weld_position u="0" x="0" y="1" z="0"
      reference="false"
      section="Fillet"
      thickness="1.5"
      angle="30"
      shape="concave"
      penetration="0.5"
      filler="yes"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </corner_weld>
</seamweld>
```

#### 10.2.6.6 Element "sheet\_parameter"

For the element <sheet\_parameter/>, the following attributes can be specified for the Corner Weld:

**Table 101 — Attributes of element <sheet\_parameter/> for Corner Weld**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <part/> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

EXAMPLE

```
<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>
```

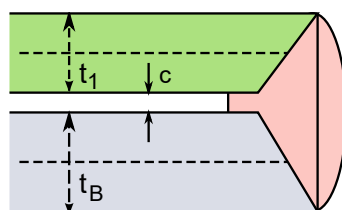
## 10.2.7 Edge Weld

### 10.2.7.1 General

The principles of the modelling of edge welds for  $\chi$ MCF 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.

### 10.2.7.2 Sheet Parameters

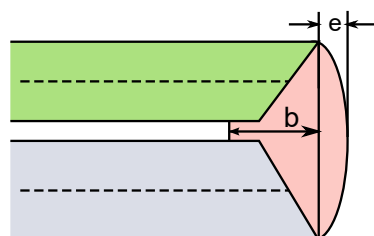


The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1$  Thickness of welded sheet;
- $c$  Gap between base and welded sheet.

**Figure 60 — Edge Weld Sheet Layout**

### 10.2.7.3 Weld Parameters



The parameters of the weld are described below:

- $b$  Width of the weld;
- $e$  Reinforcement.

**Figure 61 — Edge Weld parameters**

The following parameters can be specified for the edge weld:

**Table 102 — Parameters of Edge Weld**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
b	width	1	$\geq 0$	Optional	-
c	gap	0 – 1	$\geq 0$	Optional	0
e	-	0 – 1	$\geq 0$	Optional	0

NOTE The reinforcement is currently not defined as attribute in  $\chi$ MCF version 3.1.



### 10.2.7.4 Attributes

#### 10.2.7.4.1 Attribute "base"

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

#### 10.2.7.4.2 Attribute "technology"

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

- `resistance`;
- `arc`;
- `laser` (energy beam / laser);
- `friction`;
- `brazing`.

### 10.2.7.5 Element "weld\_position"

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

**Table 103 — Attributes of element `<weld_position/>` for Edge Weld**

Attributes	Type	Use
<code>u</code>	Floating point	Required
<code>x</code>	Floating point	Required
<code>y</code>	Floating point	Required
<code>z</code>	Floating point	Required
<code>reference</code>	Boolean	Optional
<code>section</code>	Selection	Optional
<code>width</code>	Floating point	Optional
<code>filler</code>	Selection	Optional
<code>filler_material</code>	Alphanumeric	Optional

#### 10.2.7.5.1 Attributes "u, x, y, z, reference"

Detailed definition can be found in section 10.2.4.4 Welding Position.

#### 10.2.7.5.2 Attribute "section"

Valid values for the attribute `section` of an edge weld are:

- `I` (not be confused with seam weld subtype `"i_weld"` (cf. section 10.2.4.1!));
- `V`;
- `U`.

#### 10.2.7.5.3 Attribute "width"

The attribute `width` specifies the width of the weld.

#### 10.2.7.5.4 Attribute "filler"

Valid values for the attribute filler can be:

- yes;
- no.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

#### 10.2.7.5.5 Attribute "filler\_material"

The attribute filler\_material specifies the applied material during the welding process.

#### EXAMPLE

```
<seamweld>
  <edge_weld base="1" technology="arc">
    <weld_position u="1" x="1" y="1" z="0"
      reference="false"
      section="V"
      width="2"
      filler="yes"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </edge_weld>
</seamweld>
```

#### 10.2.7.6 Element "sheet\_parameter"

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

**Table 104 — Attributes of element `<sheet_parameter />` for Corner Weld**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <code>&lt;part /&gt;</code> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

#### EXAMPLE

```
<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>
```

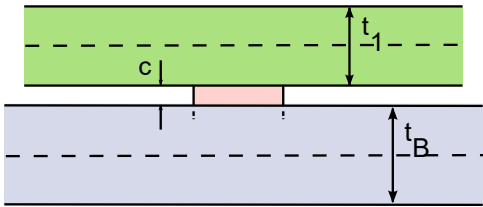
### 10.2.8 I-Weld

#### 10.2.8.1 General

The principles of the modelling of I-welds for  $\chi$ MCF 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.

10.2.8.2 Sheet Parameters

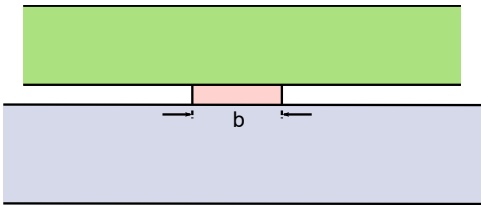


The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1$  Thickness of welded sheet;
- $c$  Gap between base and welded sheet.

Figure 62 — I-Weld Sheet Layout

10.2.8.3 Weld Parameters



The parameters of the weld are described below:

- $b$  Width of the weld.

Figure 63 — I-Weld Parameters

The following parameter can be specified for the I-weld:

Table 105 — Parameters of I-Weld

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
b	width	1	$\geq 0$	Optional	-

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

10.2.8.4 Attributes

10.2.8.4.1 Attribute "base"

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

**10.2.8.4.2 Attribute "technology"**

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

- `resistance`;
- `arc`;
- `laser` (energy beam / laser);
- `friction`;
- `brazing`.

**10.2.8.5 Element "weld\_position"**

For the element `<weld_position/>` the following attributes can be specified for the I-Weld:

**Table 106 — Attributes of element `<weld_position/>` for I Weld**

Attributes	Type	Use
<code>u</code>	Floating point	Required
<code>x</code>	Floating point	Required
<code>y</code>	Floating point	Required
<code>z</code>	Floating point	Required
<code>reference</code>	Boolean	Optional
<code>width</code>	Floating point	Optional
<code>filler</code>	Selection	Optional
<code>filler_material</code>	Alphanumeric	Optional

**10.2.8.5.1 Attributes "u, x, y, z, reference"**

Detailed definition can be found in section 10.2.4.4 Welding Position.

**10.2.8.5.2 Attribute "width"**

The attribute `width` specifies the width of the weld.

**10.2.8.5.3 Attribute "filler"**

Valid values for the attribute `filler` can be:

- `yes`;
- `no`.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

**10.2.8.5.4 Attribute "filler\_material"**

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

**EXAMPLE**

```
<seamweld>
  <i_weld base="1" technology="laser">
    <weld_position u="0" x="1" y="1" z="1"
      reference="false"
```

```

        width="1.0"
        filler="no"
        filler_material=" E7018-X"/>
    <sheet_parameter ... 7>
</i_weld>
</seamweld>

```

### 10.2.8.6 Element "sheet\_parameter"

For the element <sheet\_parameter/>, the following attributes can be specified for the I Weld:

**Table 107 — Attributes of element <sheet\_parameter/> for I Weld**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <part/> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-

#### EXAMPLE

```

<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"/>
  </i_weld>
</seamweld>

```

## 10.2.9 Overlap Weld

### 10.2.9.1 General

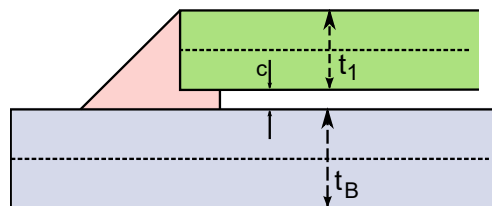
The principles of the modelling of overlap welds for  $\chi$ MCF are described in this section. An Overlap Weld describes a connection between two or three sheets welded together.

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.

NOTE: Overlap welds with *four* sheets have been observed. However, they are not explicitly depicted in this document.

## 10.2.9.2 Simple Overlap Weld

### 10.2.9.2.1 Sheet Parameters

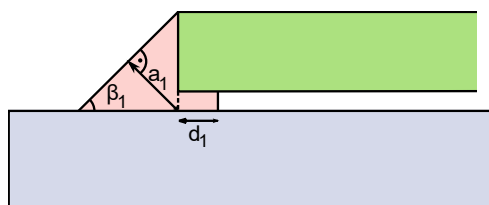


The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1$  Thickness of welded sheet;
- $c$  Gap between base and welded sheet.

**Figure 64 — Overlap Weld Sheet Layout**

### 10.2.9.2.2 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;
- $\beta_1$  Weld angle.

**Figure 65 — Overlap Weld Parameters**

For the penetration, the ratio  $\eta_1$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ MCF 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  $\chi$ MCF File the following parameters can be specified:

**Table 108 — Parameters of Overlap Weld**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	1	$\geq 0$	Optional	-
$\beta$	angle	0 – 1	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 1	$0 \leq \eta \leq 1$	Optional	0

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

### 10.2.9.3 Single Sided Double Overlap Weld

The Single Sided Double Overlap Weld is represented by a stacked welding.

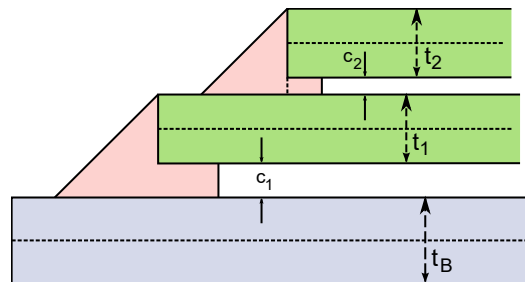


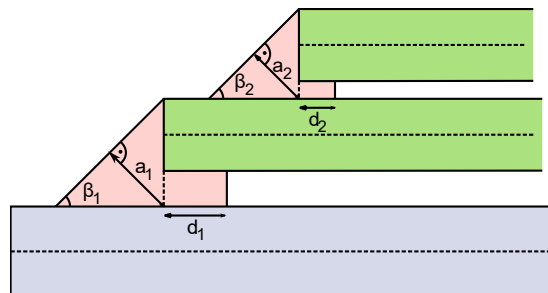
Figure 66 — Single Sided Double Overlap Weld

#### 10.2.9.3.1 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.

#### 10.2.9.3.2 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;
- $\beta_i$  Weld angle.

Figure 67 — Overlap Weld Parameter Details for lower (left) and upper (right) Weld Section

For the penetration the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ 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  $\chi$ MCF File the following parameters can be specified:

Table 109 — Parameters of Single Sided Double Overlap Weld

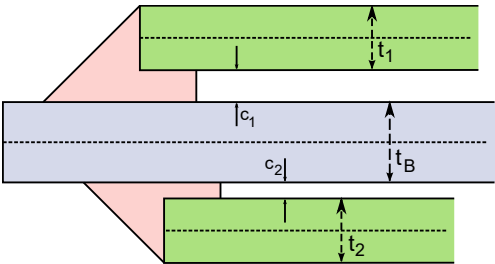
Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	2	$\geq 0$	Optional	-
$\beta$	angle	0 – 2	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

10.2.9.4 Double-Sided Double Overlap Weld

A Double-Sided Double Overlap Weld can have the welds on both sides of the base sheet.

10.2.9.4.1 Sheet Parameters

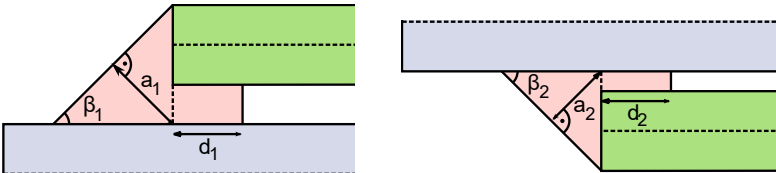


The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1, t_2$  Thicknesses of welded sheets;
- $c_1, c_2$  Gaps between base and welded sheets.

Figure 68 — Double-Sided Double Overlap Weld

10.2.9.4.2 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;
- $\beta_i$  Weld angle.

Figure 69 — Parameters of Double-Sided Double Overlap Weld (left side: upper section; right side: lower section)



For the penetration, the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ 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  $\chi$ MCF File the following parameters can be specified:

**Table 110 — Parameters of Double-Sided Double Overlap Weld**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	2	$\geq 0$	Optional	-
$\beta$	angle	0 – 2	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

## 10.2.9.5 Attributes

### 10.2.9.5.1 Attribute "base"

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

### 10.2.9.5.2 Attribute "technology"

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

- `resistance`;
- `arc`;
- `laser` (energy beam / laser);
- `friction`;
- `brazing`.

## 10.2.9.6 Element "weld\_position"

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

**Table 111 — Attributes of element `<weld_position/>` for Overlap Weld**

Attributes	Type	Use
base	Integer	Optional
u	Floating point	Required
x	Floating point	Required
y	Floating point	Required
z	Floating point	Required
reference	Boolean	Optional
section	Selection	Optional

Attributes	Type	Use
thickness	Floating point	Optional
angle	Floating point	Optional
shape	Selection	Optional
penetration	Floating point	Optional
filler	Selection	Optional
filler_material	Alphanumeric	Optional

#### 10.2.9.6.1 Attributes "u, x, y, z, reference"

Detailed definition can be found in section 10.2.4.4 Welding Position.

#### 10.2.9.6.2 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.

#### 10.2.9.6.3 Attribute "section"

The only valid value currently for the attribute `section` of an Overlap Weld is:

— `Fillet`.

This value is the default if the `section` attribute is not specified.

#### 10.2.9.6.4 Attribute "thickness"

The attribute `thickness` specifies the thickness (a-value, throat) of the weld.

#### 10.2.9.6.5 Attribute "angle"

The attribute `angle` specifies the angle of the weld relative to the base sheet.

#### 10.2.9.6.6 Attribute "shape"

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

#### 10.2.9.6.7 Attribute "penetration"

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

#### 10.2.9.6.8 Attribute "filler"

Valid values for the attribute `filler` can be:

- `yes`;
- `no`.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

#### 10.2.9.6.9 Attribute "filler\_material"

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

EXAMPLE Definition of <weld\_position/> with all attributes except base

```
<seamweld>
  <overlap_weld base="1" technology="resistance">
    <weld_position u="0" x="0" y="0" z="1"
      reference="false"
      section="Fillet"
      thickness="1.5"
      angle="30"
      shape="concave"
      penetration="0.5"
      filler="yes"
      filler_material=" E7018-X"/>
    <sheet_parameter ... />
  </overlap_weld>
</seamweld>
```

### 10.2.9.7 Element "sheet\_parameter"

For the element <sheet\_parameter/> the following attributes can be specified for the Overlap Weld:

**Table 112 — Attributes of element <sheet\_parameter/> for Overlap Weld**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <part/> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

EXAMPLE Definition of <sheet\_parameter/> including optional parameters

```
<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>
```

## 10.2.10 Y-Joint

### 10.2.10.1 General

The principles of the modelling of Y-joints for  $\chi$ 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.

NOTE The two most common welding positions are shown in Figure 70. The third welding position would be from underneath the base sheet, using a laser.

### 10.2.10.2 Sheet Parameters

The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1$  Thickness of welded sheet;
- $\alpha$  Sheet angle of welded sheet;
- $c$  Gap between base and welded sheet.

### 10.2.10.3 Weld Parameters

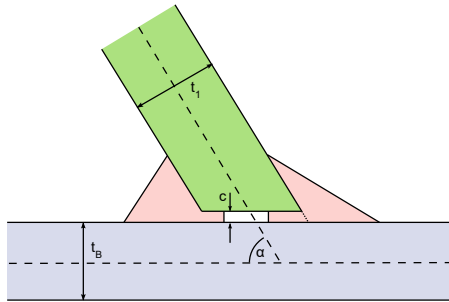


Figure 70 — Y-Joint Sheet Layout

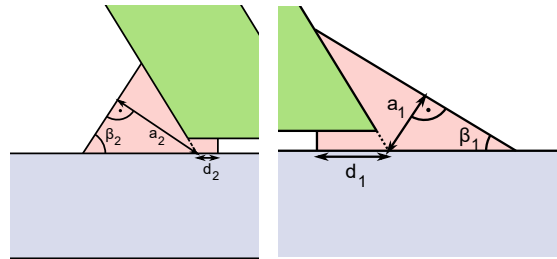


Figure 71 — Parameters of Y-Joint Weld

The parameters of the welds are the same for all of the four potential welds on the connection (applies to both figures above):

- $a_i$  Thickness of the weld (a-value, throat);
- $d_i$  Depth of the penetration;
- $\beta_i$  Weld angle.

For the penetration, the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ 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  $\chi$ MCF File only a subset can be specified:

Table 113 — Parameters of Y-Joint

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	1 – 2	$\geq 0$	Optional	-
$\beta$	angle	0 – 2	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 2	$0 \leq \eta \leq 1$	Optional	0

### 10.2.10.4 Attributes

#### 10.2.10.4.1 Attribute "base"

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

#### 10.2.10.4.2 Attribute "technology"

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

- resistance;
- arc;
- laser (energy beam / laser);
- friction;
- brazing.

#### 10.2.10.5 Element "weld\_position"

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

**Table 114 — Attributes of element `<weld_position/>` for Y Joint**

Attributes	Type	Use
base	Integer	Optional
u	Floating point	Required
x	Floating point	Required
y	Floating point	Required
z	Floating point	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Floating point	* See attribute description
angle	Floating point	* See attribute description
penetration	Floating point	* See attribute description
filler	Selection	Optional
filler_material	Alphanumeric	Optional
shape	Selection	Optional

##### 10.2.10.5.1 Attributes "u, x, y, z, reference"

Detailed definition can be found in section 10.2.4.4 Welding Position.

##### 10.2.10.5.2 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.

##### 10.2.10.5.3 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.

**10.2.10.5.4 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:

**Table 115 — Value Dependency of Attribute `thickness`**

Attribute value "section"	Attribute "thickness"
HV	Optional
HY	Not allowed
Fillet	Required

**10.2.10.5.5 Attribute "angle"**

The attribute `angle` specifies the angle of the weld relative to the base sheet.

**10.2.10.5.6 Attribute "penetration"**

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

**10.2.10.5.7 Attribute "shape"**

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

**10.2.10.5.8 Attribute "filler"**

Valid values for the attribute `filler` can be:

- `yes`;
- `no`.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

**10.2.10.5.9 Attribute "filler\_material"**

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

Example Definition of a Y-joint with all parameters for two `<weld_positions/>`

```

<seamweld>
  <y_joint base="1" technology="resistance">
    <weld_position u="0.5" x="1" y="0" z="1"
      reference="false"
      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="false"
      section="HY"
      thickness="0.5"
      angle="45"
      penetration="0.5"
      filler="yes"
      filler_material=" E7018-X"
      shape="concave"/>
    <sheet_parameter ... />
  </y_joint>
</seamweld>

```

### 10.2.10.6 Element "sheet\_parameter"

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

**Table 116 — Attributes of element `<sheet_parameter/>` for Y-Joint**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <code>&lt;part/&gt;</code> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

#### EXAMPLE

```
<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>
```

## 10.2.11 K-Joint

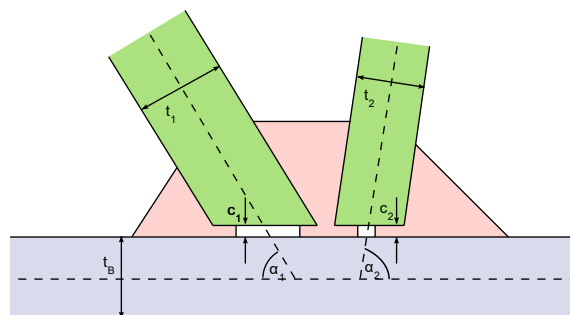
### 10.2.11.1 General

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 three most common welding positions are shown in Figure 72. The fourth weld position would be from underneath the base sheet, using a laser.

The XML definition of a K-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.

## 10.2.11.2 Sheet Parameters

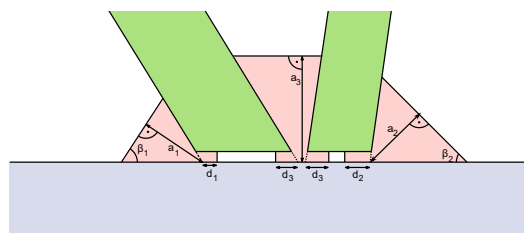


The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1, t_2$  Thickness of welded sheet;
- $\alpha_1, \alpha_2$  Sheet angle of welded sheet;
- $c_1, c_2$  Gap between base and welded sheet.

Figure 72 — K-Joint Sheet Layout

## 10.2.11.3 Weld Parameters



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;
- $\beta_i$  Weld angle.

Figure 73 — Parameters of K-Joint Weld

For the penetration, the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ 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.

The following parameters can be specified for the K-Joint:

Table 117 — Parameters of K-Joint

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	1 – 3	$\geq 0$	Optional	-
$\beta$	angle	0 – 2	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 3	$0 \leq \eta \leq 1$	Optional	0

The penetration of the 3<sup>rd</sup> weld connection ( $d_3$ ) is assumed to be equal on both welded sheets. There is only one value to be specified.



### 10.2.11.4 Attributes

#### 10.2.11.4.1 Attribute "base"

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

#### 10.2.11.4.2 Attribute "technology"

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

- `resistance`;
- `arc`;
- `laser` (energy beam / laser);
- `friction`;
- `brazing`.

### 10.2.11.5 Element "weld\_position"

#### 10.2.11.5.1 Attributes "u, x, y, z, reference"

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

**Table 118 — Attributes of element `<weld_position/>` for K Joint**

Attributes	Type	Use
<code>base</code>	Integer	Optional
<code>u</code>	Floating point	Required
<code>x</code>	Floating point	Required
<code>y</code>	Floating point	Required
<code>z</code>	Floating point	Required
<code>reference</code>	Boolean	Optional
<code>section</code>	Selection	Optional
<code>thickness</code>	Floating point	See attribute description
<code>angle</code>	Floating point	See attribute description
<code>penetration</code>	Floating point	See attribute description
<code>filler</code>	Selection	Optional
<code>filler_material</code>	Alphanumeric	Optional
<code>shape</code>	Selection	Optional

Detailed definition can be found in section 10.2.4.4 Welding Position.

#### 10.2.11.5.2 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.

**10.2.11.5.3 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.

**10.2.11.5.4 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:

**Table 119 — Value Dependency of Attribute `thickness`**

Attribute value "section"	Attribute "thickness"
HV	Optional
HY	Not allowed
Fillet	Required

**10.2.11.5.5 Attribute "angle"**

The attribute `angle` specifies the angle of the weld relative to the base sheet. The weld angle of a centre weld of a K-Joint is assumed to be parallel to the base sheet (this means 0°).

**10.2.11.5.6 Attribute "penetration"**

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

**10.2.11.5.7 Attribute "shape"**

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

**10.2.11.5.8 Attribute "filler"**

Valid values for the attribute `filler` can be:

- yes;
- no.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

**10.2.11.5.9 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>
  <k_joint base="2" technology="resistance">
    <weld_position u="1.0" x="2" y="0" z="1"
      reference="true"
      penetration="0.5"
      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="true"
        penetration="0.5"
        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="true"
        penetration="0.6"
        thickness=".5"
        angle="30"
        section="HV"
        filler="yes"
        filler_material=" E7018-X"
        shape="straight" />
<sheet_parameter ... />
<sheet_parameter ... />
</k_joint>
</seamweld>

```

#### 10.2.11.6 Element "sheet\_parameter"

For the element <sheet\_parameter/>, the following attributes can be specified for the K Joint:

**Table 120 — Attributes of element <sheet\_parameter/> for K Joint**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <part/> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

#### EXAMPLE

```

<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" sheet_thickness="1.5" sheet_angle="45"/>
    <sheet_parameter index="3" gap="1.0" sheet_thickness="1.5" sheet_angle="30"/>
  </k_joint>
</seamweld>

```

## 10.2.12 Cruciform Joint

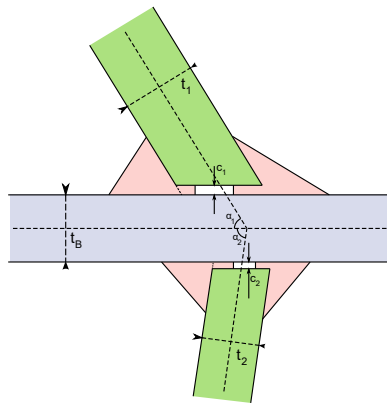
### 10.2.12.1 General

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.

### 10.2.12.2 Sheet Parameters

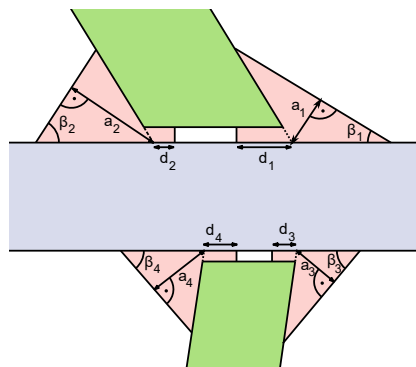


The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1, t_2$  Thickness of welded sheet;
- $\alpha_1, \alpha_2$  Sheet angle of welded sheet;
- $c_1, c_2$  Gap between base and welded sheet.

**Figure 74 — Cruciform Joint Sheet Layout**

### 10.2.12.3 Weld Parameters



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

- $a_i$  Thickness of the weld (a-value, throat);
- $d_i$  Depth of the penetration;
- $\beta_i$  Weld angle.

**Figure 75 — Parameters of Cruciform Joint**

For the penetration, the ratio  $\eta_i$  of the penetration depth to the sheet thickness is specified inside the  $\chi$ MCF file. This is computed by  $\eta_i = d_i/t_j \cdot \sin \alpha_j$  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:

**Table 121 — Parameters of Cruciform Joint**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
a	thickness	2 – 4	$\geq 0$	Optional	-
$\beta$	angle	0 – 4	$\geq 0$	Optional	45 [deg]
$\eta$	penetration	0 – 4	$0 \leq \eta \leq 1$	Optional	0

#### 10.2.12.4 Attributes

##### 10.2.12.4.1 Attribute "base"

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

##### 10.2.12.4.2 Attribute "technology"

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

- resistance;
- arc;
- laser (energy beam / laser);
- friction;
- brazing.

#### 10.2.12.5 Element "weld\_position"

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

**Table 122 — Attributes of element `<weld_position/>` for Cruciform Joint**

Attributes	Type	Use
base	Integer	Optional
u	Floating point	Required
x	Floating point	Required
y	Floating point	Required
z	Floating point	Required
reference	Boolean	Optional
section	Selection	Optional
thickness	Floating point	* See attribute description
angle	Floating point	* See attribute description
penetration	Floating point	* See attribute description
filler	Selection	Optional

Attributes	Type	Use
filler_material	Alphanumeric	Optional
shape	Selection	Optional

#### 10.2.12.5.1 Attributes "u, x, y, z, reference"

Detailed definition can be found in section 10.2.4.4 Welding Position.

#### 10.2.12.5.2 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.

#### 10.2.12.5.3 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.

#### 10.2.12.5.4 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:

**Table 123 — Value Dependency of Attribute `thickness`**

Attribute value "section"	Attribute "thickness"
HV	Optional
HY	Not allowed
Fillet	Required

#### 10.2.12.5.5 Attribute "angle"

The attribute `angle` specifies the angle of the weld relative to the base sheet.

#### 10.2.12.5.6 Attribute "penetration"

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

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

#### 10.2.12.5.7 Attribute "shape"

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

#### 10.2.12.5.8 Attribute "filler"

Valid values for the attribute filler can be:

- yes;
- no.

Depending on the technology, the default value can differ, cf. sec. 10.2.4.4.19 Attribute "filler".

#### 10.2.12.5.9 Attribute "filler\_material"

The attribute filler\_material specifies the applied material during the welding process.

EXAMPLE Definition of a cruciform\_joint with all parameters of 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="true"
      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="true"
      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="true"
      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="true"
      filler="yes"
      filler_material=" E7018-X"
      shape="straight" />
    <sheet_parameter ... />
    <sheet_parameter ... />
  </cruciform_joint>
</seamweld>
```

10.2.12.6 Element "sheet\_parameter"

For the element <sheet\_parameter/>, the following attributes can be specified for the Cruciform Joint:

Table 124 — Attributes of element <sheet\_parameter/> for Cruciform Joint

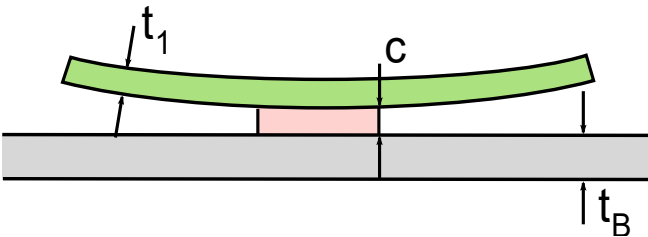
Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <part/> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-
sheet_angle	Floating point	Optional	-

EXAMPLE Definition of a cruciform\_joint with only required weld\_position parameters

```
<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" sheet_thickness="1.5" sheet_angle="90"/>
    <sheet_parameter index="3" gap="1.0" sheet_thickness="1.5" sheet_angle="90"/>
  </cruciform_joint>
</seamweld>
```

10.2.13 Flared Joint

10.2.13.1 Sheet Parameters



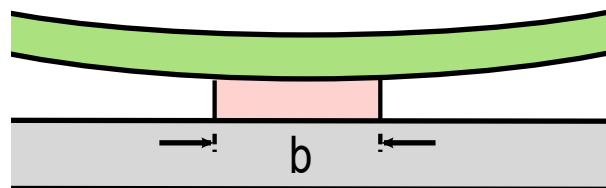
The parameters to describe the connection are:

- $t_B$  Thickness of base sheet;
- $t_1$  Thickness of welded sheet;
- $c$  Gap between base and welded sheet.

Figure 76 — Flared Joint Sheet Layout



### 10.2.13.2 Weld Parameters



The parameters of the welds are described below:

- $b$  width of the weld.

**Figure 77 — Flared Joint Sheet Layout**

The following parameter can be specified for the Flared Joint:

**Table 125 — Parameters of Flared joint**

Parameter	$\chi$ MCF-Key	Multiplicity	Value Range	Use	Default Value
$b$	width	1	$\geq 0$	Optional	-

All other parameters are provided by the model itself and are partially used to specify parameters of the weld.

### 10.2.13.3 Attributes

#### 10.2.13.3.1 Attribute "base"

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

#### 10.2.13.3.2 Attribute "technology"

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

- resistance;
- arc;
- laser (energy beam / laser);
- friction;
- brazing.

### 10.2.13.4 Element "weld\_position"

#### 10.2.13.4.1 General

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

**Table 126 — Attributes of element `<weld_position/>` for Flared Joint**

Attributes	Type	Use
$u$	Floating point	Required
$x$	Floating point	Required
$y$	Floating point	Required

Attributes	Type	Use
z	Floating point	Required
reference	Boolean	Optional
width	Floating point	Optional

#### 10.2.13.4.2 Attributes "u, x, y, z, reference"

Detailed definition can be found in section 10.2.4.4 Welding Position.

#### 10.2.13.4.3 Attribute "width"

The attribute `width` specifies the width of the weld.

EXAMPLE `<flared_joint/>` with all parameters of `weld_position`

```
<seamweld>
  <flared_joint base="1" technology="arc">
    <weld_position u="0" x="1" y="1" z="1"
      reference="false"
      width="1.0" />
    <sheet_parameter ... />
  </flared_joint >
</seamweld>
```

#### 10.2.13.5 Element "sheet\_parameter"

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

**Table 127 — Attributes of element `<sheet_parameter/>` for Flared Joint**

Attributes	Type	Use	Constraint / Remarks
index	Integer	Required	It shall be referenced to <code>&lt;part/&gt;</code> index attribute
gap	Floating point	Optional	Default value is 0
sheet_thickness	Floating point	Optional	-

EXAMPLE `<flared_joint/>` with all parameters of `weld_position`

```
<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>
```

### 10.3 Adhesive Lines

An adhesive line is denoted by an element `<adhesive_line/>`. This element is described completely by its attributes and nested elements.

**Table 128 — Nested elements of `<connection_1d/>`**

Nested Elements	Multiplicity	Use	Constraint / Remarks
adhesive_line	1	Optional	-
loc_list	1-*	Required	See section 10.1.2 loc_list
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

#### 10.3.1 Element "adhesive\_line"

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

**Table 129 — Attributes of element `<adhesive_line/>`**

Attributes	Type	Value Space	Use	Constraint
base	Integer	> 0	Optional	-
width	Floating point	>= 0.0	Optional	-
thickness	Floating point	>= 0.0	Optional	-
material	Alphanumeric	Alphanumeric	Optional	-

The following list explains the attributes:

- `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.

#### 10.3.2 Element "loc\_list"

This follows the syntax as defined in section 10.1.2 Location.

#### 10.3.3 Element "appdata"

This follows the syntax as defined in section 7.3.2 User Specific Data `<appdata/>`.

### 10.3.4 Element "femdata"

This follows the syntax as defined in section 7.3.3 Finite Element Specific Data <femdata/>.

EXAMPLE A Definition of an adhesive line without base attribute

```
<connection_1d label="ADH_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 Definition of an adhesive line with base attribute

```
<connection_1d label="ADH_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>
```

## 10.4 Hemming Flanges

### 10.4.1 General

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.

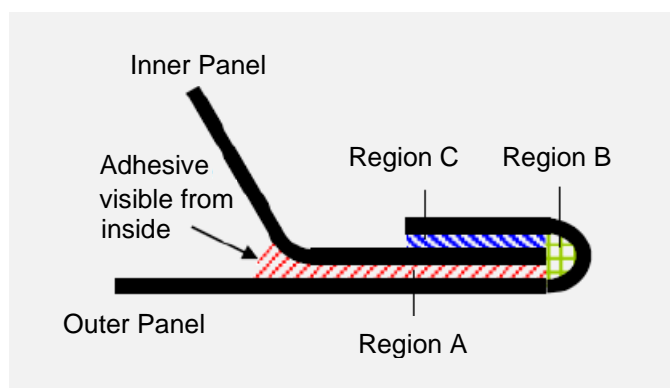
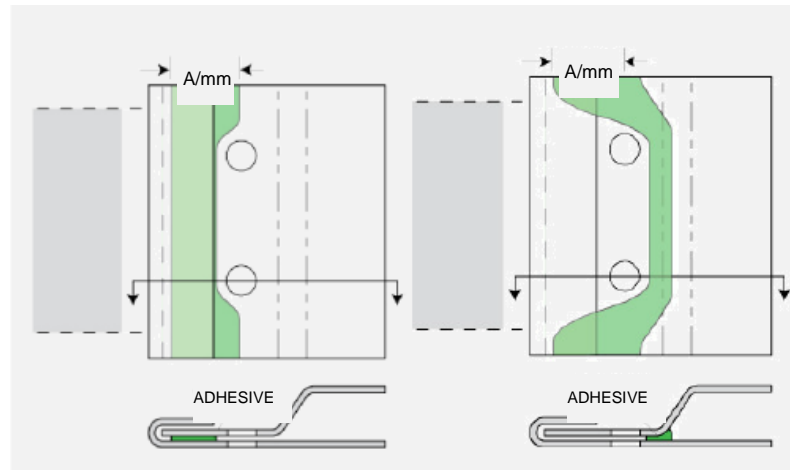


Figure 78 — The Three Regions of a Hemming

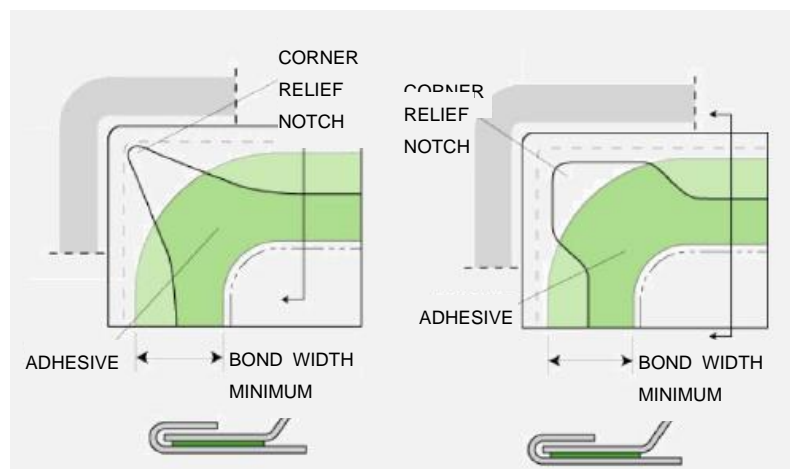
The solution described below addresses certain features in hemming design:

- 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 79 — Path Changes and Width Changes in Hemming Flanges**

Width and path sometimes change to avoid obstacles, like holes.



**Figure 80 — Adhesive Path Differs from Root Path**

Adhesive generally follows inner routes around corners.

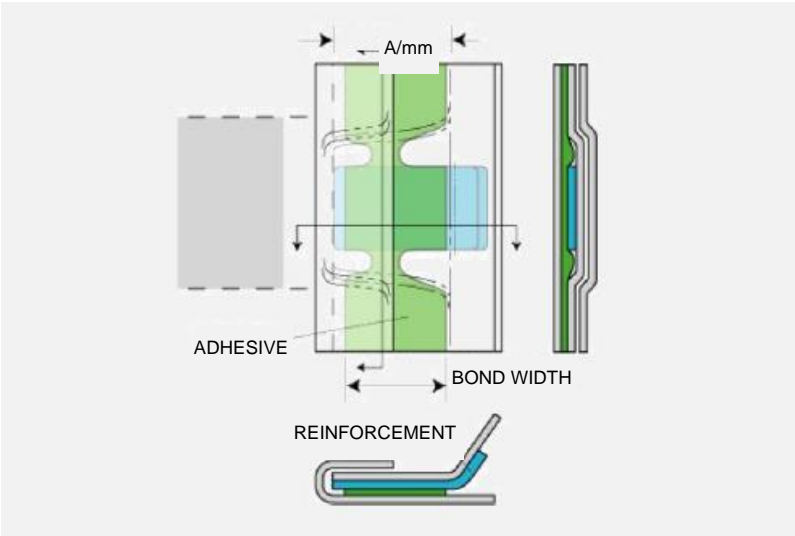


Figure 81 — Reinforcements need to be considered as Part of the Inner Panel

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.

10.4.2 Element <hemming/>

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

Table 130 — Nested elements of <connection\_1d/> for <hemming/>

Nested Elements	Multiplicity	Use	Constraint / Remarks
hemming	1	Optional	-
loc_list	1-*	Required	See section 10.1.2 loc_list
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

10.4.3 Element "loc\_list"

This is the path of the *hemming root*. It follows the syntax as defined in section 10.1.2 Location.

10.4.4 Element "appdata"

This follows the syntax as defined in section 7.3.2 User Specific Data <appdata/>.

10.4.5 Element "femdata"

This follows the syntax as defined in section 7.3.3 Finite Element Specific Data <femdata/>.

## 10.4.6 Element "hemming"

### 10.4.6.1 General

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

**Table 131 — Attributes of element <hemming/>**

Attributes	Type	Value Space	Use	Constraint
folded_width	Floating point	> 0.0	Optional	-
folded_part	Integer	-	Optional	Index of the folded sheet

The following list explains the attributes:

- `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 7.4.2.2 Element <part/>.

Its definition is similar to "base" attribute of <seamwelds/> in section 10.2.4.1 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:

**Table 132 — Nested elements of element <hemming/>**

Nested Elements	Multiplicity	Use	Constraint
region	1-3	Optional	-

### 10.4.6.2 Element "region"

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

**Table 133 — Attributes of element <region/>**

Attributes	Type	Value Space	Use	Constraint
label	Alphanumeric	"A", "B", or "C"	Required	-
fill_percentage	Floating point	[0.0, 100.0]	Optional	-
top_index	Integer	> 0	Optional	refers to <part/> index attribute
bottom_index	Integer	> 0	Optional	refers to <part/> index attribute

This element defines adhesion properties of region A, B, or C.

- `label`: this is an identifier of the hemming region, according to Figure 78. Only values "A", "B" and "C" are meaningful;
- `fill_percentage`: target hem filling for this region;
- `top_index`: the index (see section 7.4.2.2) where the region's adhesive connects to;
- `bottom_index`: the index (see section 7.4.2.2) where the region's adhesive connects to.

Existence of `top_index` and `bottom_index` is meaningful only if adhesive element is specified, especially when the hemming involves more than 2 flange partners.

The order of `top_index` and `bottom_index` is not important. However, if they are not specified, the corresponding adhesive is free to select any of the hemming's flange partners. The adhesive will guess which are the relevant partners, using its position.

The adhesive of hemming regions "A" and "C" can be described in the following nested elements:

**Table 134 — Nested elements of element `<region/>`**

Nested Elements	Multiplicity	Use	Constraint / Remarks
<code>connection_1d</code> <code>connection_2d</code>	1	Optional	Exactly one of these elements. It has to either contain an <code>&lt;adhesive_line/&gt;</code> or an <code>&lt;adhesive_face/&gt;</code> .

The usage of adhesives in the `<region/>` is described in sections 10.3 Adhesive Lines and 11.2 Adhesive Faces.

Region "B" is not expected to contain an adhesive line or face. The definition is left open for future extensions.

**EXAMPLE** Definition of a `<hemming/>` connection

```

<connected_to>
  <part index="1" label="PART_7000400"/> <!-- outer hood panel -->
  <assy index="23">
    <part label="PART_5000300"/> <!-- inner hood panel -->
    <part label="PART_5000800"/> <!-- reinforcement -->
  </assy>
</connected_to>
<connection_1d label="HMG_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 label="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>
    <region label="B" fill_percentage="100"/>
    <region label="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>
      </connection_2d>
    </region>
  </hemming>
</connection_1d>

```



```
<face_list> <!-- adhesive's facets -->
...
</face_list>
<appdata> <!-- adhesive's appdata -->
...
</appdata>
</connection_2d>
</region>
</hemming>
</connection_1d>
```

10.5 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 modelling 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.

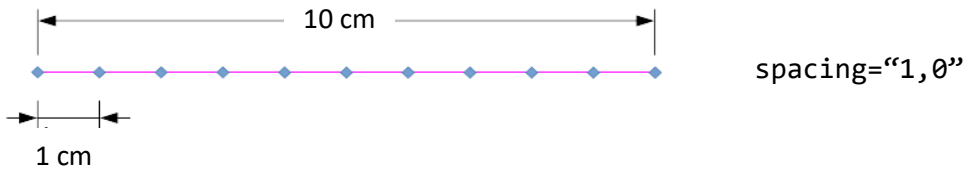


Figure 82 — Sequence without margin

An optional `margin` value allows space to be left from each side.

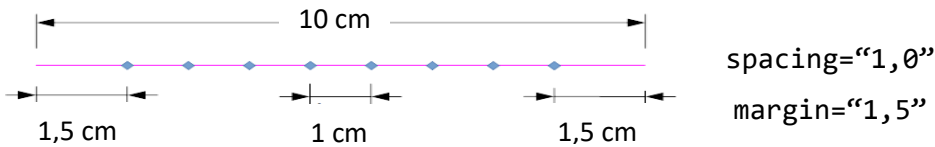


Figure 83 — Sequence with margin and spacing

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:

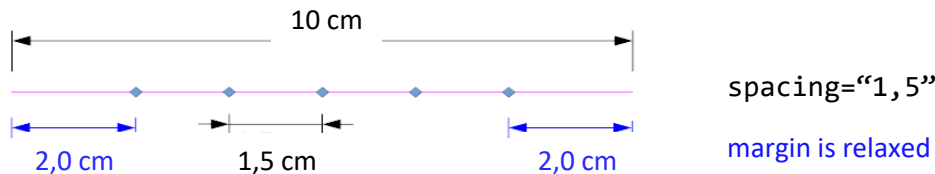


Figure 84 — Margin relaxation

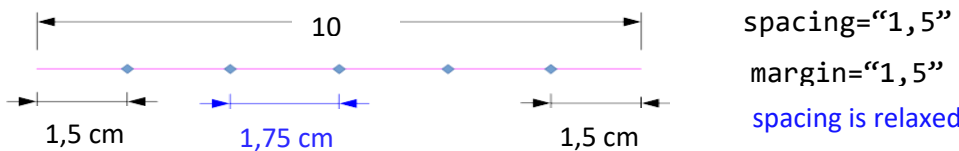


Figure 85 — Spacing relaxation

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  $\Delta_{\text{spacing}}$  is minimal).

A `<loc_list/>` is necessary for this type of connection.

Example A Minimum definition for `<sequence_connection_0d/>`

```
<connection_1d label="SPOT_LINE_11000">
  <sequence_connection_0d spacing="30.0">
    <spotweld/>
  </sequence_connection_0d>
  <loc_list>
    ...
  </loc_list>
  <appdata>
    ...
  </appdata>
</connection_1d>
```

Example B Complete definition for `<sequence_connection_0d/>`

```
<connection_1d label="DROP_LINE_33000">
  <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/>` with a diameter of 6mm

```
<connection_1d label="SPOT_LINE_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/>:

**Table 135 — Nested elements of <connection\_1d/> for <sequence\_connection\_0d/>**

Nested Elements	Multiplicity	Use	Constraint / Remarks
sequence_connection_0d	1	Optional	-
loc_list	1-*	Required	See section 10.1.2 loc_list
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

The XML definition of a <sequence\_connection\_0d/> may contain any of the following 0d connection types:

**Table 136 — Nested elements of <sequence\_connection\_0d/>**

Nested Elements	Multiplicity	Use	Constraint
spotweld	1	Optional	-
gumdrop	1	Optional	-

Nesting 0d elements with directions (such as rivet, screws, robscans) would be impossible with this definition.

Only one of the nested elements (spotweld or gumdrop) shall exist. If all are missing, then this will default to spotweld.

XML specification of <sequence\_connection\_0d/>:

**Table 137 — Attributes of element <sequence\_connection\_0d/>**

Attributes	Type	Value Space	Use	Constraint
spacing	Floating point	≥ 0.0	Optional	-
margin	Floating point	≥ 0.0	Optional	Default value is 0.0
priority	Selection	{"spacing", "margin"}	Optional	Default value is "spacing"

## 11 2D connections

### 11.1 Generic Definitions

#### 11.1.1 Identification

For identifying 2D connections, the same rules apply as for 0D connections, see section 9.1.1 identification.

#### 11.1.2 Connection Face

##### 11.1.2.1 General

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.

### 11.1.2.2 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:

**Table 138 — Nested elements of `<loc_list/>`**

Nested Elements	Multiplicity	Use	Constraint
loc	3-*	Required	-

### 11.1.2.3 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:

**Table 139 — Attributes of element `<loc/>`**

Attributes	Type	Use	Constraint / Remarks
v	Integer	Required	Unique within the parent element <code>&lt;connection_2d/&gt;</code>

The attribute v is used to ensure unique identification. The index values shall 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:

**Table 140 — Nested elements of element `<face_list/>`**

Nested Elements	Multiplicity	Use	Constraint
face	1-*	Required	-

#### 11.1.2.4 Element "face"

Each location specified by the element `<face/>` contains *four* values specifying each vertex of the facet, using the `<loc/>` identifier, *v*.

**Table 141 — Attributes of element `<face/>`**

Attribute (Vertex)	Type	Use	Constraint
<b>v1</b>	Integer	Required	shall correspond to a <b>v</b> in a <code>loc</code> from <code>loc_list</code>
<b>v2</b>	Integer	Required	shall correspond to a <b>v</b> in a <code>loc</code> from <code>loc_list</code>
<b>v3</b>	Integer	Required	shall correspond to a <b>v</b> in a <code>loc</code> from <code>loc_list</code>
<b>v4</b>	Integer	Optional	shall correspond to a <b>v</b> in a <code>loc</code> from <code>loc_list</code>

The following list explains the attributes:

- 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 Minimum definition for a `<connection_2d/>` facets

```
<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>
```

#### 11.1.3 Type Specification

Each connection is identified by its type. The XML definition of 2D connections contains the following nested elements:

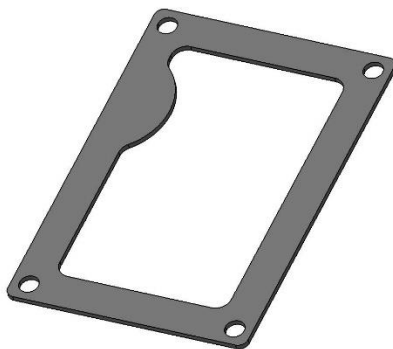
**Table 142 — Nested elements of `<connection_2d/>`**

Nested Elements	Multiplicity	Use	Constraint
adhesive_face	1	Optional	-
stacking	1	Optional	See section 7.4.2.4

Only one of the type elements `<adhesive_face/>` shall exist in `<connection_2d/>`. If none of the type elements exist, then this will default to `adhesive_face`.

## 11.2 Adhesive Faces

A die-cut adhesive is denoted by an element `<adhesive_face/>`.



**Figure 86 — Picture of a sealing or adhesive face**

An adhesive face connection is denoted by an element `<adhesive_face/>`. This element is described completely by its attributes and nested elements.

**Table 143 — Nested elements of element `<connection_2d/>`**

Nested Elements	Multiplicity	Use	Constraint / Remarks
adhesive_face	1	Optional	-
loc_list	1	Required	-
face_list	1	Required	-
appdata	1	Optional	-
femdata	1	Optional	-
custom_attributes_list	1	Optional	See section 8.5 Custom Attributes list

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

**Table 144 — Attributes of element `<adhesive_face/>`**

Attributes	Type	Value space	Use	Constraint
base	Integer	$> 0$	Optional	-
thickness	Floating point	$\geq 0.0$	Optional	-
material	Alphanumeric	-	Optional	-

The following list explains the attributes:

- **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** Definition of an adhesive face connection

```
<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>

```

## 12 Future extensions

### 12.1 General

So far, only the abovementioned connection types with the corresponding parameters are described, which cover mainly the applications of CAD and CAE. However,  $\chi$ 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.

### 12.2 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.

### 12.3 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,  $\chi$ MCF is open for any new joint type which will come and be of relevance for the technical application.

Annex A  
(informative)

Derivation of Formulae used for Regular Intermittent Welds

The regular intermittent weld is defined using with these labelled terms:

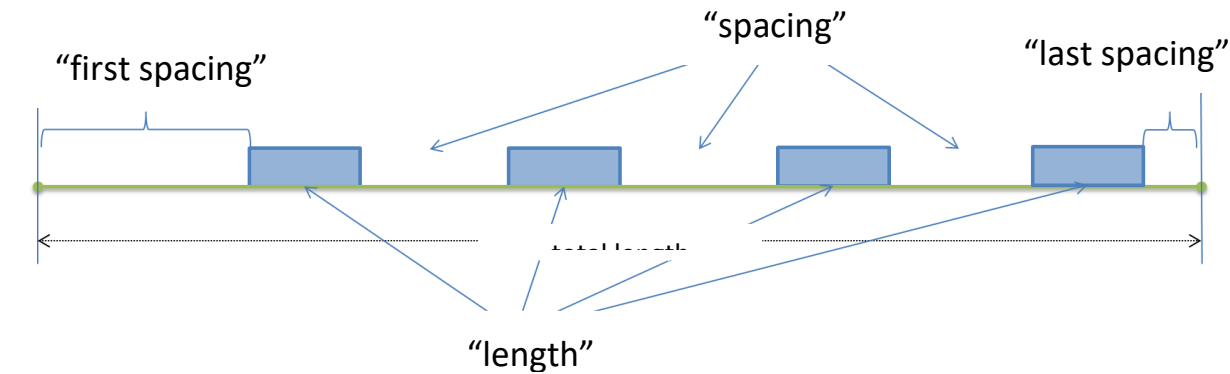


Figure A. 1 — 'length', 'spacing', 'first\_spacing' and 'last\_spacing' are the terms needed to define a regular intermittent weld.

For simplicity, we shall refer to them as:

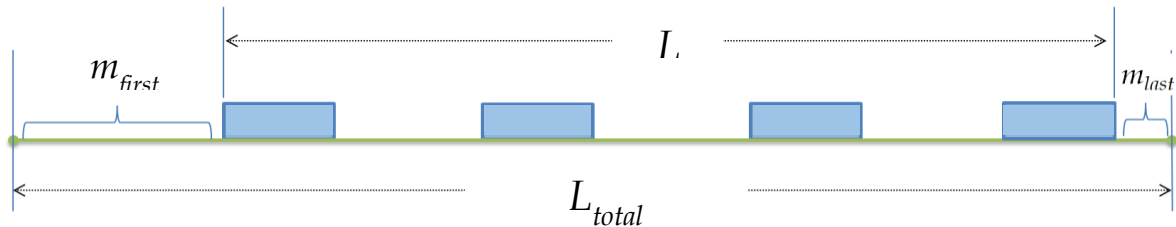


Figure A. 2 — A regular intermittent weld with 'n' segments and 'n-1' spacings between segments.

where:

$L_{total}$	The "total length" of the of the <loc_list> polyline in $\chi$ MCF.
$m_{first}$	"first_spacing"
$m_{last}$	"last_spacing"
$l$	the prescribed "length"
$s$	the prescribed "spacing"
$n$	"num_segments" defines the number of segments. $n - 1$ is the number of spacings between the segments.

Consider a connection line of total length  $L_{total}$ . The effective welded length,  $L$ , is:

$$L = L_{total} - m_{first} - m_{last}$$



The density,  $d$ , of the welded segments is defined as:

$$d = \frac{l}{l + s} \quad (\text{Annex A.1})$$

The effective length,  $L$ , can be calculated by adding the segments and the spacings:

$$L = nl + (n - 1)s \quad (\text{Annex A.2})$$

$n$  is the prescribed number of segments that fit in the connection line. The line is a polyline approximation in  $\chi$ MCF. As a result of this, the *prescribed* length and spacing may not fit in the approximated curve. Therefore, the length or spacing or both are *adjusted*.

There are 3 strategies for adjusting the length and spacing:

- keep length – adjust the spacing:  
from (Annex A.2):

$$\begin{aligned} L &= nl + (n - 1)\bar{s} && \text{where } \bar{s}: \text{adjusted spacing;} \\ \Rightarrow \bar{s} &= \frac{L - nl}{n - 1} && (\text{Annex A.3}) \end{aligned}$$

- keep spacing – adjust the length:  
from (Annex A.2):

$$\begin{aligned} L &= n\bar{l} + (n - 1)s && \text{where } \bar{l}: \text{adjusted length;} \\ \Rightarrow \bar{l} &= \frac{L - (n - 1)s}{n} && (\text{Annex A.4}) \end{aligned}$$

- keep density – adjust length and spacing:

$$\begin{aligned} (\text{Annex A.1}) \text{ becomes: } & d = \frac{\bar{l}}{\bar{l} + \bar{s}} \\ \Rightarrow & \bar{s} = \left( \frac{1 - d}{d} \right) \bar{l} && (\text{Annex A.5}) \end{aligned}$$

$$\begin{aligned} (\text{Annex A.2}) \text{ becomes: } & L = n\bar{l} + (n - 1)\bar{s} \\ \Rightarrow & L = n\bar{l} + (n - 1) \left( \frac{1 - d}{d} \right) \bar{l} && \text{substituted (Annex A.5)} \end{aligned}$$

$$\begin{aligned} \Rightarrow dL &= dn\bar{l} + (n - 1)(1 - d)\bar{l} && \text{multiplied by } d \\ &= (dn + (n - 1)(1 - d))\bar{l} && \text{factored by } \bar{l} \\ &= (dn + n - nd - 1 + d)\bar{l} && \text{expanded product} \\ &= (n - 1 + d)\bar{l} \end{aligned}$$

$$\Rightarrow \bar{l} = \frac{dL}{n - 1 + d} \quad (\text{Annex A.6})$$

$$\text{and } \bar{s} = \frac{(1 - d)L}{n - 1 + d} \quad (\text{Annex A.7})$$

## **Annex B** (informative)

### **Federative use of $\chi$ MCF with ISO 10303-242**

#### **B.1 General principles**

According to the widespread use of ISO 10303-242 (cf. [8]), it is important to describe the federated use of  $\chi$ MCF together with ISO 10303-242.

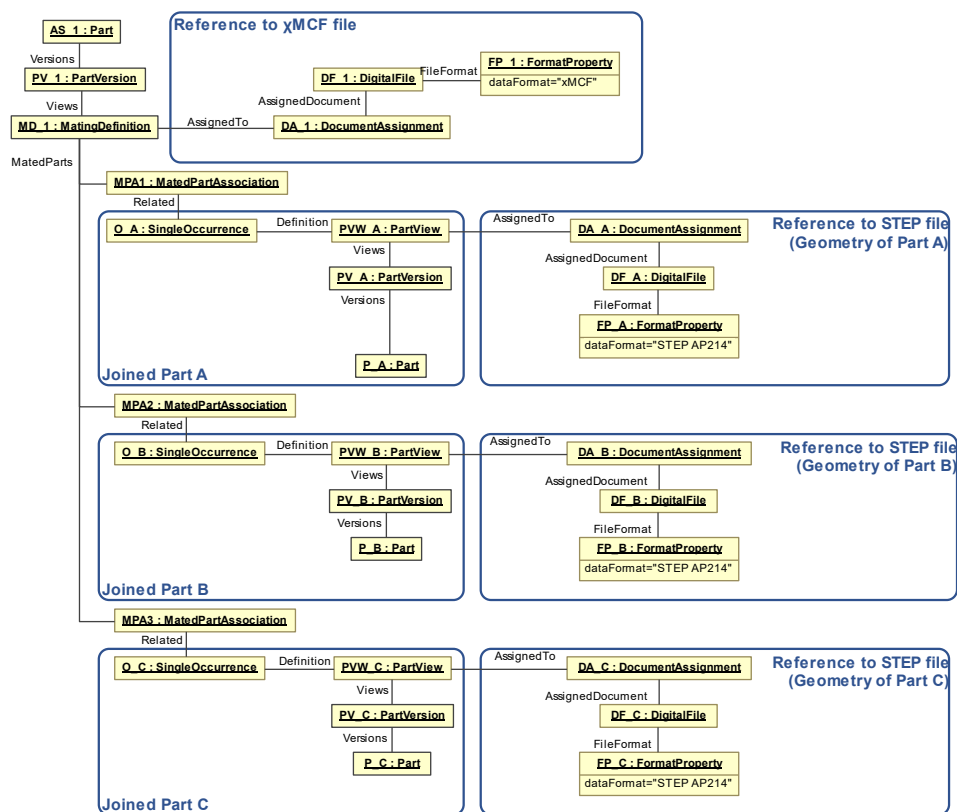
Following general principles apply to the federated use of  $\chi$ MCF together with ISO 10303-242:

1. Both standard definitions stay unchanged. Federated use shall be described by recommendations, only;
2. Clearly defined and delimited roles are assigned to both standards;
3. Redundancies shall be avoided as far as possible;
4. In case of unavoidable redundancies, there shall be no inconsistencies within the set of federatively used files.

These general principles are implemented by following regulations:

- A. ISO 10303-242 contains the usual PLM-type information. Especially, it references the relevant files (let it be CAD native or standard, visualization or  $\chi$ MCF) and defines the location in space, where their content shall be instantiated (geometric transformations);
- B.  $\chi$ MCF contains geometrical (position, orientation, length, ...), technical information of connecting elements, and the lists of the parts connected, only;
- C. In ISO 10303-242, the element "MatedPartAssociation" is necessary for technical reasons. The list of the part numbers of connected parts is mandatory within it. It shall be identical to `<connected_to/>` in  $\chi$ MCF up to sequence;
- D. Consequently, nested  $\chi$ MCF element `<assy/>` cannot be used. Furthermore, attributes "pid" and "pname" of nested `<part/>` element cannot be used;
- E.  $\chi$ MCF files are referenced from ISO 10303-242 by means of "external reference".

NOTE: In general,  $\chi$ MCF files are handled quite similar to CAD files, for example if it comes to configuration or version management.



**Figure B. 1— References in STEP file to related  $\chi$ MCF file**

Figure B. 1 depicts how the STEP AP 242 file refers to the  $\chi$ MCF file for an example assembly which was taken from Figure 2. The complete assembly named “AS\_1” consists of 3 parts (“P\_A”, P\_B”, P\_C”) which are logically linked via “MatingDefinition” in the STEP file. The  $\chi$ MCF-file which contains all the detailed joining information is external to the STEP AP242 file and is referenced by “DocumentAssignment”. The 3 parts of the assembly are associated to the assembly by 3 entries of “MatingAssociation”. The actual geometry of the 3 parts is defined in separate files external to the AP242 file by 3 entries “DocumentAssignment”. In summary, the AP242 file contains the logical structure of the assembly, whereas the detailed physical design information, such as part geometries and details of the connections or joints are described in external files which are referenced by the AP242 file.

## B.2 Cross-References between ISO 10303-242 and $\chi$ MCF

Both standards,  $\chi$ MCF and ISO 10303-242, contain elements which on first glance may potentially match. However, there is *only one* pair of matching elements, as is explained by following table:

Cross-Reference Table between ISO 10303-242 and  $\chi$ MCF

**Table B. 1 — Cross-Reference Table between ISO 10303-242 and  $\chi$ MCF**

$\chi$ MCF	ISO 10303-242	Comments
connection_group	MatingDefinition	MatingDefinition points to part version of assembly, which is irrelevant for $\chi$ MCF. Hence, there is no correlation between both XML elements.
+ connected_to	MatedPartAssociation	MatedPartAssociation contains geometric transformation, hence is necessary. List of part codes is mandatory within it.
+ connection_list	n/a	No corresponding entity in ISO 10303-242.
+ connection_[012]d	MatedPartRelationship	Semantics of both XML elements does not match exactly. They are just similar. MatedPartRelationship is not relevant for $\chi$ MCF use cases.
+ + connection_[012]d_type	Mating_Type	ISO 10303-242 defines the connection type as attribute within MatedPartRelationship, which is not relevant (see above). “connection_[012]d_type” is just a placeholder for a specific name, such as “spotweld”, “rivet”, “seamweld”, ...
+ + + loc	location	<loc/> in $\chi$ MCF is nested in <connection_[012]d/>. ISO 10303-242 entity is not relevant, since $\chi$ MCF is master for location.
units	Unit	Unit system used by the file. However, they do not need to be coincident (for instance, one could be in m, the other one in inches).
date	TimeStamp in header element	Date on which the file is created. Does not need to be coincident.
version	Encoded in XML name space	Version code of the standard used. These XML elements are not related.

**Any not mentioned entity of either standard does not map to or interact with an entity of the other standard.**

## Annex C (informative)

### History

Facing the difficulty that joints were represented quite differently in different CAE tools, The VDA Research Association for Automotive Technology (FAT) working group FAT-AK 25 (=Working Group 25) started to develop a standard for connections & joints in cooperation with CAE software vendors. The working group 25 for joining technologies is a working group of the German Research Association of Automotive Technologies (FAT). The FAT is a department of the German Association of the Automotive Industry (VDA),

The evaluation of existing formats revealed that the **Master Connection File (MCF)** by Ford [9] was the most suitable basis for future developments and extensions. This original MCF format is based on the XML-standard but covers only few joint types and parameters [9]. In order to distinguish from the original Ford-MCF, the FAT-format was named the **Extended Master Connection File**, abbreviated as " $\chi$ MCF" (read: chi-M-C-F) or xMCF (read: x-M-C-F).

In 2005, the working group decided to begin with the extension of MCF to seam welds. There were several reasons for this decision. First, the demand for the fatigue evaluation of seam welds was increasing rapidly. Furthermore, there were and still are a wide variety of weld types with partly complex geometrical shapes [10]. The proper description of these welds meant a big challenge. The successful treatment of seam welds laid out the foundation for the integration of any other joint type.

This document is based on the most recent VDA/FAT standard "xMCF – A Standard for Describing Connections & Joints in Mechanical Systems (Version 3.1)" (see [11]). The version of  $\chi$ MCF described in this document has a strong and stable structure but has probably not covered all potential joining types and parameters. Thanks to the simple extensibility of  $\chi$ MCF, additional information can be integrated on demand. In addition, customizable elements allow to adopt new joining technologies or parameters before formal implementation in future editions of ISO 8329. Older versions of the standard ([12], [13]) can be found on the VDA website but are for reference only.

## Bibliography

- [1] International Standards Organization, ISO 8601: Data elements and interchange formats – Information interchange – Representation of dates and times, Geneva: ISO.
- [2] N. Schulte-Frankenfeld, "FATXML-Format Version V1.2 R3," VDA FAT-Ak27, Berlin, 2020.
- [3] SAE International — Fasteners Committee, J492\_201611 — Guide for Rivet Selection and Design Consideration, Warrendale (Pennsylvania) & Troy (Michigan): SAE International, 2016, p. 14.
- [4] P. Garnero and V. Marchetto, "A method for resistance electric spot welding of a first sheet of non weldable material to a second sheet of weldable metal material". European Patent EP0967044A2, 23 10 1999.
- [5] O. Hahn and A. Schulte, "Nutzung des Festigkeitspotentials höherfesten Stahlfeinbleche durch Stanzniet- und Clinchverbindungen," 1998.
- [6] T. Draht and G. Meschut, "Method of producing a nail connection, and nail for this purpose". European Patent EP1926918B1, 16 01 2007.
- [7] T. Ziegler, "Joinability of light-weight components using riveted friction-welded joints.," in *Joining in Car Body Engineering*, Bad Nauheim, 2019.
- [8] International Standards Organization, ISO 10303-242: Industrial automation systems and integration — Product data representation and exchange — Part 242: Application protocol: Managed model-based 3D engineering, Geneva: ISO.
- [9] B. E. Huf, "Managing Connections using the Master Connection File," Ford Motor Co., Dearborn, 2001.
- [10] S. Zhang, "Classification of Seam Welds," Daimler AG, Stuttgart, 2005.
- [11] FAT-AK25, "χMCF Extended Master Connection File: A Standard for Describing Connections and Joints in the Automotive Industry, Version 3.1," VDA FAT-AK25, Berlin, 2020.
- [12] FAT-AK25, "χMCF Extended Master Connection File: A Standard for Describing Connections and Joints in the Automotive Industry, Version 2.0," VDA FAT-AK25, Berlin, 2014.
- [13] FAT-AK25, "χMCF Extended Master Connection File: A Standard for Describing Connections and Joints in the Automotive Industry, Version 3.0," VDA FAT-AK25, Berlin, 2016.
- [14] B. C. Systems, "χMCF pilot in ANSA," Beta CAE System S.A., Thessaloniki, 2008.
- [15] C. Gaier and K. Hofwimmer, "Seam-Weld Types and Fatigue Relevant Parameter Sets for NCF Standard," Magna Engineering Center Steyr GmbH & Co KG, Steyr, 2006.
- [16] P. Mikolaj, "First Proposal for The Extended Master Connection File (χMCF) as a Transfer Standard of Seamweld Connection Definition," MSC.Software, Alzenau, 2006.