

# Documentation

## ASSIST User Guide

Robert Hilbrich, Michael Behrisch

DLR German Aerospace Center  
Institute of Transportation Systems

Berlin



DLR

Deutsches Zentrum  
für Luft- und Raumfahrt  
German Aerospace Center



**DLR German Aerospace Center**

Institute of Transportation Systems

Prof. Dr. Karsten Lemmer

Rutherfordstraße 2

12489 Berlin

Tel: +49 531 295-3401

Robert Hilbrich

Tel: +49 30 67055-582

Fax: +49 30 67055-291

E-Mail: robert.hilbrich@dlr.de

**Document Identification:**

Title . . . . .	ASSIST User Guide
Subject . . . . .	Documentation
Author(s) . . . . .	Robert Hilbrich, Michael Behrisch
Filename . . . . .	user-guide.tex
Last saved on . . . . .	8th September 2015

**Document History:**

Version 0.1 . . . . .	Initial Version	18.08.2015
-----------------------	-----------------	------------



# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Usage</b>	<b>3</b>
<b>3. Input Specification</b>	<b>5</b>
<b>4. Constraint Semantics</b>	<b>7</b>
<b>A. Input Specification Grammar</b>	<b>A</b>



# 1. Introduction





## 2. Usage



## 3. Input Specification



## 4. Constraint Semantics



# A. Input Specification Grammar

In Listing A.1 the grammar for the input specification language is depicted. Please refer to this Listing when a specification contains syntax errors.

```
1 grammar ch.hilbri.assist.mappingdsl.MappingDSL with org.eclipse.xtext.common.Terminals

3 import "ch.hilbri.assist.datamodel.model"
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

5 AssistModel:
7   'Global' '{'
8     (
9       ('Name' '=' systemName=STRING ';' )?
10        &
11        ('Compatible Interface Types' '{' (compatibleTypes+=CompatibleTypeEntry)* '}' )?
12         &
13         ( cableWeightData=CableWeightData )?
14         &
15         ( protectionLevelData=ProtectionLevelData )?
16     )
17   '}',

18 compartments+=Compartment+

19 ('Interfaces' '{'
20   (eqInterfaces+=EqInterface)+
21 '}' )?

22 ('InterfaceGroups' '{'
23   (
24     eqInterfaceGroups+=EqInterfaceGroup |
25     eqInterfaceGroups+=EqInterfaceGroupWithCombinedDefinition
26   )*
27 '}' )?

28 ('Restrictions' '{'
29   (
30     dislocalityRelations += DislocalityRelation |
31     colocalityRelations += ColocalityRelation |
```

```

35         validDeployments+=ValidDeployment          |
        invalidDeployments+=InvalidDeployment
37     )*
39     '}'?
41 ;

41 ProtectionLevelData:
43     'Protection Level Restrictions' '{' (protectionLevelEntries+=ProtectionLevelEntry)+ '}'
45 ;

45 ProtectionLevelEntry:
47     'RDC.Location' '=' rdcLocation=STRING 'And' 'Equipment.EmhZone1' '=' emhZone1=STRING '->'
        ' protectionLevel+=ProtectionLevelType (' protectionLevel+=ProtectionLevelType)* ' ';
49 ;

51 CompatibleloTypeEntry:
        eqInterfaceloType=STRING '->' pinInterfaceloTypes+=STRING (' pinInterfaceloTypes+=STRING)* ' ';
53 ;

55 CableWeightData:
        'Cable Weights' '{' (cableWeightEntries+=CableWeightEntry)+ '}'
57 ;

59 CableWeightEntry:
        (eqInterfaceloType=STRING | defaultEntry?='default') '=' weight=Double ' ';
61 ;

63

65 Compartment:
        'Compartment' name=ID '{'
67     (
        ('Manufacturer' '=' manufacturer=STRING ' ;')? &
69     ('PowerSupply' '=' powerSupply=STRING ' ;')? &
        ('Side' '=' side=STRING ' ;')? &
71     ('Zone' '=' zone=STRING ' ;')? &
        (rdcs+=RDC+)
73     )
        '}'
75 ;

77 RDC:
        'RDC' name=ID '{'
79     (
        ('Manufacturer' '=' manufacturer=STRING ' ;')? &
81     ('PowerSupply' '=' powerSupply=STRING ' ;')? &

```



```

83         ('Type'                '=' rdcType=STRING ';' )?           &
        ('ESS'                  '=' ess=STRING ';' )?               &
85         ('Location'           '=' location=STRING ';' )?          &
        ('ResourceX'            '=' resourceX=SIGNEDINT ';' )?       &
        ('ResourceY'            '=' resourceY=SIGNEDINT ';' )?       &
87         ('ResourceZ'          '=' resourceZ=SIGNEDINT ';' )?       &
        (connectors+=Connector+)                                     &
89         ('Pins' '{' connectedPins+=InternallyConnectedPinEntry+ '}' )?
    )
91 '}',
;
93 InternallyConnectedPinEntry:
95     pins+=[ AvailableEqInterface | QualifiedName]
    ', ' pins+=[ AvailableEqInterface | QualifiedName]
97     (' ', ' pins+=[ AvailableEqInterface | QualifiedName] )?
    'are' 'connected' ';',
99 ;
101 Connector:
103     'Connector' name=ID '{'
        ( availableEqInterfaces +=AvailableEqInterface)*
105     '}',
;
107 AvailableEqInterface :
109     ('Name' '=' name=ID)? eqInterfaceType=STRING '=' count=INT ('with protection level'
        protectionLevel=ProtectionLevelType)? ';',
;
111 enum ProtectionLevelType:
113     NONE = 'None' |
        L1 = 'L1' |
115     L2 = 'L2' |
        L3 = 'L3' |
117     L4 = 'L4' |
        L5 = 'L5' |
119     L6 = 'L6' |
        L7 = 'L7' |
121     L8 = 'L8'
;
123
125 /* SOFTWARE */
127 EqInterfaceGroup:
    'Group' name=ID '{'
129     (

```

```

131         eqInterfaces+=[EqInterface] |
        implicitMemberDefinitions+=ImplicitEqInterfaceMemberDefinition
133     )
    (
135         ', ' (
            eqInterfaces+=[EqInterface] |
            implicitMemberDefinitions+=ImplicitEqInterfaceMemberDefinition
137         )
    ) *
139     '}',
141
143     (
145         'without' '{'
            (
147                 withoutEqInterfaces+=[EqInterface] |
                withoutImplicitMemberDefinitions+=ImplicitEqInterfaceMemberDefinition
149             )
            (
151                 ', ' (
                    withoutEqInterfaces+=[EqInterface] |
                    withoutImplicitMemberDefinitions+=
                        ImplicitEqInterfaceMemberDefinition
153                 )
            ) *
155         '}',
157     ) ?
    ';;'
159
161 ImplicitEqInterfaceMemberDefinition :
    'interfaces with' entries+=ImplicitEqInterfaceMemberDefinitionAttributesAndValues ('and' entries
        +=ImplicitEqInterfaceMemberDefinitionAttributesAndValues)*
163 ;
165 ImplicitEqInterfaceMemberDefinitionAttributesAndValues:
    attribute=ImplicitEqInterfaceMemberDefinitionAttribute '=' value=STRING
167 ;
169 enum ImplicitEqInterfaceMemberDefinitionAttribute:
    NAME      = 'Name'      |
171    SYSTEM    = 'System'    |
    SUBATA     = 'SubAta'    |
173    LINENAME  = 'LineName'  |
    WIRINGLANE = 'WiringLane'|
175    GRPINFO   = 'GrpInfo'   |
    ROUTE      = 'Route'     |

```

```

177     PWSUP1      = 'PwSup1'          |
        EMHZONE1 = 'EmhZone1'        |
179     IOTYPE      = 'Type'            |
        RESOURCE  = 'Resource'        |
181     RESOURCE_X = 'ResourceX'        |
        RESOURCE_Y = 'ResourceY'      |
183     RESOURCE_Z = 'ResourceZ'
;

185 EqInterfaceGroupWithCombinedDefinition:
187     'Group' name=ID 'combines' combinedGroups+=[EqInterfaceGroup] ('and' combinedGroups+=[
        EqInterfaceGroup])* ',';
;

189 EqInterface:
191     'Interface' name=ID '{'
193         ('System'      '=' system=STRING  ';' )?      &
        ('SubAta'       '=' subAta=STRING  ';' )?      &
195         ('Resource'    '=' resource=STRING ';' )?      &
        ('LineName'     '=' lineName=STRING ';' )?      &
197         ('WiringLane'  '=' wiringLane=STRING ';' )?    &
        ('GrpInfo'      '=' grpInfo=STRING  ';' )?      &
199         ('Route'       '=' route=STRING   ';' )?      &
        ('PwSup1'       '=' pwSup1=STRING   ';' )?      &
201         ('EmhZone1'    '=' emhZone1=STRING ';' )?      &
        ('Type'         '=' ioType=STRING   ';' )?      &
203         ('ResourceX'   '=' resourceX=SIGNEDINT ';' )?  &
        ('ResourceY'    '=' resourceY=SIGNEDINT ';' )?  &
205         ('ResourceZ'   '=' resourceZ=SIGNEDINT ';' )?
    )
207     '}' ;

209 /* RELATIONEN */
211 enum HardwareArchitectureLevelType :
213     COMPARTMENT = 'Compartment' |
        RDC       = 'RDC'         |
215     CONNECTOR = 'Connector'
;

217 DislocalityRelation :
219     eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup])* '
        dislocal up to' hardwareLevel=HardwareArchitectureLevelType ';' ;

221 ColocalityRelation :

```

```

223     eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup])* 'on
        same' hardwareLevel=HardwareArchitectureLevelType ';'
225 ;
ValidDeployment:
227     'Valid for' eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[
        EqInterfaceOrGroup])*
        ('is'|'are') '{'
229         ( hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition)
        (',' (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition) )*
231     '}' ',';
233 ;
InvalidDeployment:
235     'Invalid for' eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[
        EqInterfaceOrGroup])*
        ('is'|'are') '{'
237         ( hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition)
        (',' (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition) )*
239     '}' ',';
241 ;
DeploymentImplicitDefinition :
243     'connectors with' entries+=DeploymentImplicitDefinitionAttributeAndValue ('and' entries+=
        DeploymentImplicitDefinitionAttributeAndValue)*
245 ;
DeploymentImplicitDefinitionAttributeAndValue:
247     attribute = DeploymentImplicitDefinitionAttribute '=' value=STRING
249 ;
enum DeploymentImplicitDefinitionAttribute:
251     COMPARTMENT_NAME      = 'Compartment.Name'      |
        COMPARTMENT_MANUFACTURER = 'Compartment.Manufacturer' |
253     COMPARTMENT_POWERSUPPLY = 'Compartment.PowerSupply' |
        COMPARTMENT_SIDE      = 'Compartment.Side'      |
255     COMPARTMENT_ZONE      = 'Compartment.Zone'      |

257     RDC_NAME              = 'RDC.Name'              |
        RDC_MANUFACTURER     = 'RDC.Manufacturer'     |
259     RDC_POWERSUPPLY       = 'RDC.PowerSupply'       |
        RDC_SIDE              = 'RDC.Side'             |
261     RDC_TYPE              = 'RDC.Type'              |
        RDC_ESS               = 'RDC.ESS'              |
263     RDC_RESOURCE_X        = 'RDC.ResourceX'         |

```

```
265   RDC_RESOURCE_Y      = 'RDC.ResourceY'      |
      RDC_RESOURCE_Z      = 'RDC.ResourceZ'      |
267   CONNECTOR_NAME      = 'Connector.Name'
      ;
269   /* MISC */
271   MetricParameter:
      name=STRING '=' value=INT ' ';
273   ;

275   Double returns ecore::EDouble:
      '-'? INT? '.'? INT
277   ;

279   QualifiedName:
      ID ('.' ID)*
281   ;

283   SIGNEDINT returns ecore::Elint :
      '-'? INT;
```

Listing A.1: **ASSIST Input Specification Grammar**