

Common Warehouse Metamodel (CWM) Specification

Volume 4. Extensions XML, IDL and DTD

*Proposal to the OMG ADTF RFP:
Common Warehouse Metadata Interchange (CWMI)*

Joint Submission

<i>International Business Machines Corporation</i>	<i>Oracle Corporation</i>
<i>Unisys Corporation</i>	<i>UBS AG</i>
<i>NCR Corporation</i>	<i>Genesis Development Corporation</i>
<i>Hyperion Solutions Corporation</i>	<i>Dimension EDI</i>

Supported by:

<i>Deere & Company</i>	<i>InLine Software</i>
<i>Sun Microsystems, Inc.</i>	<i>Aonix</i>
<i>Hewlett-Packard Company</i>	<i>Hitachi, Ltd.</i>
<i>Data Access Technologies</i>	

*OMG Document ad/2000-01-11
February 11, 2000*

Copyright 1999, IBM Corporation
Copyright 1999, Unisys Corporation
Copyright 1999, NCR Corporation
Copyright 1999, Hyperion Solutions
Copyright 1999, Oracle Corporation
Copyright 1999, UBS AG
Copyright 1999, Genesis Development Corporation
Copyright 1999, Dimension EDI

The companies listed above hereby grant a royalty-free license to the Object Management Group, Inc. (OMG) for worldwide distribution of this document or any derivative works thereof, so long as the OMG reproduces the copyright notices and the below paragraphs on all distributed copies.

The material in this document is submitted to the OMG for evaluation. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. The companies listed above shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. The information contained in this document is subject to change without notice.

This document contains information which is protected by copyright. All Rights Reserved. Except as otherwise provided herein, no part of this work may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without the permission of one of the copyright owners. All copies of this document must include the copyright and other information contained on this page.

The copyright owners grant member companies of the OMG permission to make a limited number of copies of this document (up to fifty copies) for their internal use as part of the OMG evaluation process.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013.

CORBA, OMG, and Object Request Broker are trademarks of Object Management Group.

1. CWMX Rendered in XML 5

2. CWM Interfaces as CORBA IDL 197

- 2.1 ER.idl 198
- 2.2 CORBAIDLDataTypes.idl 205
- 2.3 COBOLData.idl 211
- 2.4 DMSII.idl 225
- 2.5 Essbase.idl 243
- 2.6 Express.idl 254
- 2.7 IMSDatabase.idl 269
- 2.8 InformationSet.idl 300
- 2.9 InformationReporting.idl 309

3. CWMX Document Types Definitions 313

- 3.1 CWMX.dtd 314
- 3.2 ER.dtd 315
- 3.3 CORBAIDLDataTypes.dtd 320
- 3.4 COBOLData.dtd 325
- 3.5 DMSII.dtd 334
- 3.6 Essbase.dtd 347
- 3.7 Express.dtd 356
- 3.8 IMSDatabase.dtd 364
- 3.9 InformationSet.dtd 383
- 3.10 InformationReporting.dtd 387

1 CWMX Rendered in XML

This chapter contains the CWM Extension (CWMX) metamodels expressed in XML. These metamodels are used to generate the interfaces and XML document types in chapters 2 and 3 of this volume.

The XML is shown below as an update to the CWM XML rather than as a separate XML document so that links to CWM and UML model elements can be expressed as simple XMI references. To add CWMX metamodels to the CWM XML in volume 2, chapter 1, make the two additions shown below.

First, add the following XMI.model elements into the XMI.header section after the CWM XMI.model elements.

```
<XMI.model xmi.name='org.omg.cwm.foundation.ER' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.foundation.CORBAIDLDataTypes' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.resource.COBOLData' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.resource.DMSII' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.resource.Essbase' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.resource.Express' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.resource.IMSDatabase' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.analysis.InformationSet' xmi.version='1.0'/>
<XMI.model xmi.name='org.omg.cwm.analysis.InformationReporting' xmi.version='1.0'/>
```

Second, add the following into the end of the XMI.content section.

```
<Model.Package xmi.id='a1768'>
  <Model.ModelElement.name>ER</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a1769'>
      <Model.ModelElement.name>Entity</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a112'/> <!-- Foundation.Core.Class -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
      </Model.Namespace.contents>
    </Model.Class>
    <Model.Class xmi.id='a1770'>
      <Model.ModelElement.name>NonuniqueKey</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
```

```

    <Model.Class xmi.idref='a788'/> <!-- KeysIndexes.Index -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1771'>
  <Model.ModelElement.name>CandidateKey</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a782'/> <!-- KeysIndexes.UniqueKey -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1772'>
  <Model.ModelElement.name>ForeignKey</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a786'/> <!-- KeysIndexes.KeyRelationship -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a1773'>
      <Model.ModelElement.name>relationshipEnd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XMI.field>0</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>false</XMI.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1774'/> <!-- ER.RelationshipEnd -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a1775'/> <!--
          ER.ForeignKeyImplements.relationshipEnd -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1776'>
  <Model.ModelElement.name>Domain</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a103'/> <!-- Foundation.Core.Classifier -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1777'>
      <Model.ModelElement.name>validationRule</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1778'>
  <Model.ModelElement.name>default</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a1779'>
  <Model.ModelElement.name>baseType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a103'/> <!-- Foundation.Core.Classifier -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1780'/> <!-- ER.DomainBaseType.baseType -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1781'>
  <Model.ModelElement.name>Attribute</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a128'/> <!-- Foundation.Core.Attribute -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1782'>
  <Model.ModelElement.name>Relationship</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='false'/>
<Model.GeneralizableElement.isLeaf xmi.value='false'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a180'/> <!-- Foundation.Core.AssociationClass -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1774'>
  <Model.ModelElement.name>RelationshipEnd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a108'/> <!-- Foundation.Core.AssociationEnd -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1783'>
      <Model.ModelElement.name>delete</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a1784'>
      <Model.ModelElement.name>update</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a1785'>
      <Model.ModelElement.name>insert</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```



```

    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a1786'>
    <Model.ModelElement.name>foreignKey</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1772'/> <!-- ER.ForeignKey -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a1787'/> <!-- ER.ForeignKeyImplements.foreignKey -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1788'>
  <Model.ModelElement.name>ModelLibrary</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a712'/> <!-- Model_Management.Subsystem -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1789'>
  <Model.ModelElement.name>Model</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a711'/> <!-- Model_Management.Model -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1790'>
  <Model.ModelElement.name>SubjectArea</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a707'/> <!-- Model_Management.Package -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>

```

```

    </Model.Namespace.contents>
  </Model.Class>
  <Model.Class xmi.id='a1791'>
    <Model.ModelElement.name>PrimaryKey</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false' />
    <Model.GeneralizableElement.isLeaf xmi.value='false' />
    <Model.GeneralizableElement.isAbstract xmi.value='false' />
    <Model.GeneralizableElement.visibility xmi.value='public_vis' />
    <Model.Class.isSingleton xmi.value='false' />
    <Model.GeneralizableElement.supertypes>
      <Model.Class xmi.idref='a1771' /> <!-- ER.CandidateKey -->
    </Model.GeneralizableElement.supertypes>
  </Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Association xmi.id='a1792'>
  <Model.ModelElement.name>ForeignKeyImplements</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true' />
  <Model.GeneralizableElement.isLeaf xmi.value='true' />
  <Model.GeneralizableElement.isAbstract xmi.value='false' />
  <Model.GeneralizableElement.visibility xmi.value='protected_vis' />
  <Model.Association.isDerived xmi.value='false' />
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1775'>
      <Model.ModelElement.name>relationshipEnd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true' />
      <Model.AssociationEnd.aggregation xmi.value='none' />
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true' />
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1774' /> <!-- ER.RelationshipEnd -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1787'>
      <Model.ModelElement.name>foreignKey</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true' />
      <Model.AssociationEnd.aggregation xmi.value='none' />
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true' />
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1772' /> <!-- ER.ForeignKey -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1793'>
  <Model.ModelElement.name>DomainBaseType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true' />
  <Model.GeneralizableElement.isLeaf xmi.value='true' />
  <Model.GeneralizableElement.isAbstract xmi.value='false' />
  <Model.GeneralizableElement.visibility xmi.value='public_vis' />
  <Model.Association.isDerived xmi.value='false' />
  <Model.Namespace.contents>

```

```

<Model.AssociationEnd xmi.id='a1780'>
  <Model.ModelElement.name>baseType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a103'/> <!-- Foundation.Core.Classifier -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a1794'>
  <Model.ModelElement.name>domain</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1776'/> <!-- ER.Domain -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a1795'>
  <Model.ModelElement.name>Expressions</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a820'/> <!-- Expressions -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a1796'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a1797'>
  <Model.ModelElement.name>KeysIndexes</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a781'/> <!-- KeysIndexes -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a1798'>
  <Model.ModelElement.name>Model_Management</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a706'/> <!-- Model_Management -->
  </Model.Import.importedNamespace>

```

```

    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Tag xmi.id='a1799'>
    <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.foundation</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.cwm.x.foundation</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1768'/> <!-- ER -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1800'>
    <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.foundation</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.java.cwm.x.foundation</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1768'/> <!-- ER -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1801'>
    <Model.ModelElement.name>org.omg.xmi.namespace=CWMXER</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>CWMXER</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1768'/> <!-- ER -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1802'>
    <Model.ModelElement.name>org.omg.mof.idl_substitute_name=defaultValue</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>defaultValue</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Attribute xmi.idref='a1778'/> <!-- ER.Domain.default -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1803'>
    <Model.ModelElement.name>org.omg.mof.java_substitute_name=defaultValue</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>defaultValue</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Attribute xmi.idref='a1778'/> <!-- ER.Domain.default -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1804'>
    <Model.ModelElement.name>org.omg.mof.idl_substitute_name=ErAttribute</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>

```

```

        <XML.any xmi.type='string'>ErAttribute</XML.any>
    </Model.Tag.values>
</Model.Tag.elements>
    <Model.Class xmi.idref='a1781'/> <!-- ER.Attribute -->
</Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1805'>
    <Model.ModelElement.name>org.omg.mof.java_substitute_name=ErAttribute</
        Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
        <XML.any xmi.type='string'>ErAttribute</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
        <Model.Class xmi.idref='a1781'/> <!-- ER.Attribute -->
    </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1806'>
    <Model.ModelElement.name>org.omg.mof.idl_substitute_name=ErRelationship</
        Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
        <XML.any xmi.type='string'>ErRelationship</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
        <Model.Class xmi.idref='a1782'/> <!-- ER.Relationship -->
    </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1807'>
    <Model.ModelElement.name>org.omg.mof.java_substitute_name=ErRelationship</
        Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
        <XML.any xmi.type='string'>ErRelationship</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
        <Model.Class xmi.idref='a1782'/> <!-- ER.Relationship -->
    </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1808'>
    <Model.ModelElement.name>org.omg.mof.idl_substitute_name=ErModel</
        Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
        <XML.any xmi.type='string'>ErModel</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
        <Model.Class xmi.idref='a1789'/> <!-- ER.Model -->
    </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1809'>
    <Model.ModelElement.name>org.omg.mof.java_substitute_name=ErModel</
        Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
        <XML.any xmi.type='string'>ErModel</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
        <Model.Class xmi.idref='a1789'/> <!-- ER.Model -->
    </Model.Tag.elements>
</Model.Tag>
</Model.Namespace.contents>
</Model.Package>

```

```

<Model.Package xmi.id='a1810'>
  <Model.ModelElement.name>CORBAIDLDataTypes</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a1811'>
      <Model.ModelElement.name>IDLType</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a114'/> <!-- Foundation.Core.DataType -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
        <Model.Attribute xmi.id='a1812'>
          <Model.ModelElement.name>typeCode</Model.ModelElement.name>
          <Model.ModelElement.annotation></Model.ModelElement.annotation>
          <Model.Feature.scope xmi.value='instance_level'/>
          <Model.Feature.visibility xmi.value='public_vis'/>
          <Model.StructuralFeature.multiplicity>
            <XML.field>1</XML.field> <!-- lower -->
            <XML.field>1</XML.field> <!-- upper -->
            <XML.field>>false</XML.field> <!-- is_ordered -->
            <XML.field>>false</XML.field> <!-- is_unique -->
          </Model.StructuralFeature.multiplicity>
          <Model.StructuralFeature.isChangeable xmi.value='true'/>
          <Model.Attribute.isDerived xmi.value='false'/>
          <Model.TypedElement.type>
            <Model.DataType xmi.idref='a1813'/> <!-- CORBAIDLDataTypes.TypeCode -->
          </Model.TypedElement.type>
        </Model.Attribute>
      </Model.Namespace.contents>
    </Model.Class>
    <Model.Class xmi.id='a1814'>
      <Model.ModelElement.name>ArrayType</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
        <Model.Attribute xmi.id='a1815'>
          <Model.ModelElement.name>length</Model.ModelElement.name>
          <Model.ModelElement.annotation></Model.ModelElement.annotation>
          <Model.Feature.scope xmi.value='instance_level'/>
          <Model.Feature.visibility xmi.value='public_vis'/>
          <Model.StructuralFeature.multiplicity>
            <XML.field>1</XML.field> <!-- lower -->
            <XML.field>1</XML.field> <!-- upper -->
            <XML.field>>false</XML.field> <!-- is_ordered -->
            <XML.field>>false</XML.field> <!-- is_unique -->
          </Model.StructuralFeature.multiplicity>
          <Model.StructuralFeature.isChangeable xmi.value='true'/>
          <Model.Attribute.isDerived xmi.value='false'/>
          <Model.TypedElement.type>
            <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
          </Model.TypedElement.type>
        </Model.Attribute>
      </Model.Namespace.contents>
    </Model.Class>
  </Model.Namespace.contents>
</Model.Package>

```

```

<Model.Reference xmi.id='a1816'>
  <Model.ModelElement.name>elementType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1817'/> <!--
      CORBAIDLDataTypes.ArrayElementType.elementType -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1818'>
  <Model.ModelElement.name>ExceptionType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1819'>
  <Model.ModelElement.name>SequenceType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1820'>
      <Model.ModelElement.name>length</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
  <Model.Reference xmi.id='a1821'>
    <Model.ModelElement.name>elementType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a1822'/> <!--
        CORBAIDLDataTypes.SequenceElementType.elementType -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1823'>
  <Model.ModelElement.name>WstringType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1824'>
      <Model.ModelElement.name>length</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1825'>
  <Model.ModelElement.name>StringType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1826'>
      <Model.ModelElement.name>length</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>

```



```

        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
        <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
    </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1827'>
    <Model.ModelElement.name>FixedType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.Class.isSingleton xmi.value='false'/>
    <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
    </Model.GeneralizableElement.supertypes>
    <Model.Namespace.contents>
        <Model.Attribute xmi.id='a1828'>
            <Model.ModelElement.name>digits</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.Feature.scope xmi.value='instance_level'/>
            <Model.Feature.visibility xmi.value='public_vis'/>
            <Model.StructuralFeature.multiplicity>
                <XML.field>1</XML.field> <!-- lower -->
                <XML.field>1</XML.field> <!-- upper -->
                <XML.field>>false</XML.field> <!-- is_ordered -->
                <XML.field>>false</XML.field> <!-- is_unique -->
            </Model.StructuralFeature.multiplicity>
            <Model.StructuralFeature.isChangeable xmi.value='true'/>
            <Model.Attribute.isDerived xmi.value='false'/>
            <Model.TypedElement.type>
                <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
            </Model.TypedElement.type>
        </Model.Attribute>
        <Model.Attribute xmi.id='a1829'>
            <Model.ModelElement.name>scale</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.Feature.scope xmi.value='instance_level'/>
            <Model.Feature.visibility xmi.value='public_vis'/>
            <Model.StructuralFeature.multiplicity>
                <XML.field>1</XML.field> <!-- lower -->
                <XML.field>1</XML.field> <!-- upper -->
                <XML.field>>false</XML.field> <!-- is_ordered -->
                <XML.field>>false</XML.field> <!-- is_unique -->
            </Model.StructuralFeature.multiplicity>
            <Model.StructuralFeature.isChangeable xmi.value='true'/>
            <Model.Attribute.isDerived xmi.value='false'/>
            <Model.TypedElement.type>
                <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
            </Model.TypedElement.type>
        </Model.Attribute>
    </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1830'>
    <Model.ModelElement.name>Alias</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>

```

```

<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  <Model.Class xmi.idref='a736'/> <!-- DataTypes.TypeAlias -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1831'>
  <Model.ModelElement.name>StructType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1832'>
  <Model.ModelElement.name>UnionType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a730'/> <!-- DataTypes.Union -->
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1833'>
  <Model.ModelElement.name>EnumType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a742'/> <!-- DataTypes Enumeration -->
    <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.DataType xmi.id='a1813'>
  <Model.ModelElement.name>TypeCode</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcTypeCode/>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
<Model.Association xmi.id='a1834'>

```

```

<Model.ModelElement.name>ArrayElementType</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='true'/>
<Model.GeneralizableElement.isLeaf xmi.value='true'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a1817'>
    <Model.ModelElement.name>elementType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a1835'>
    <Model.ModelElement.name>arrayType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='false'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1814'/> <!-- CORBAIDLDataTypes.ArrayType -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1836'>
  <Model.ModelElement.name>SequenceElementType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1822'>
      <Model.ModelElement.name>elementType</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1811'/> <!-- CORBAIDLDataTypes.IDLType -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.AssociationEnd xmi.id='a1837'>

```

```

    <Model.ModelElement.name>sequence</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='false'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1819'/> <!-- CORBAIDLDataTypes.SequenceType -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a1838'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a1839'>
  <Model.ModelElement.name>DataTypes</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a728'/> <!-- DataTypes -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Tag xmi.id='a1840'>
  <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.resource</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>org.omg.cwm.x.resource</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a1810'/> <!-- CORBAIDLDataTypes -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1841'>
  <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.resource</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>org.omg.java.cwm.x.resource</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a1810'/> <!-- CORBAIDLDataTypes -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1842'>
  <Model.ModelElement.name>org.omg.xmi.namespace=CWMXIDL</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>CWMXIDL</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>

```

```

    <Model.Package xmi.idref='a1810'/> <!-- CORBAIDLDataTypes -->
  </Model.Tag.elements>
</Model.Tag>
</Model.Namespace.contents>
</Model.Package>
<Model.Package xmi.id='a1843'>
  <Model.ModelElement.name>COBOLData</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a1844'>
      <Model.ModelElement.name>COBOLField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1089'/> <!-- Record.Field -->
      </Model.GeneralizableElement.supertypes>
    </Model.Class>
    <Model.Attribute xmi.id='a1845'>
      <Model.ModelElement.name>level</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a1846'>
      <Model.ModelElement.name>signKind</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a1847'/> <!-- COBOLData.SignKindType -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a1848'>
      <Model.ModelElement.name>isFiller</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1849'>
  <Model.ModelElement.name>isJustifiedRight</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1850'>
  <Model.ModelElement.name>isBlankWhenZero</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1851'>
  <Model.ModelElement.name>isSynchronized</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1852'>
  <Model.ModelElement.name>picture</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1853'>
  <Model.ModelElement.name>occursLower</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1854'>
  <Model.ModelElement.name>occursUpper</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1855'>
  <Model.ModelElement.name>indexName</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1856'>
  <Model.ModelElement.name>isExternal</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1857'>
  <Model.ModelElement.name>isGlobal</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a1858'>
  <Model.ModelElement.name>dependingOnField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1859'/> <!--
      COBOLData.OccursDependingOn.dependingOnField -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1860'>
  <Model.ModelElement.name>occurringField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1861'/> <!--
      COBOLData.OccursDependingOn.occurringField -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1862'>
  <Model.ModelElement.name>occursKeyInfo</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```



```

<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>-1</XML.field> <!-- upper -->
  <XML.field>true</XML.field> <!-- is_ordered -->
  <XML.field>true</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a1863'/> <!-- COBOLData.OccursKey -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a1864'/> <!--
    COBOLData.OccursKeyField.occursKeyInfo -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1865'>
  <Model.ModelElement.name>occursKeyFieldInfo</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1863'/> <!-- COBOLData.OccursKey -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1866'/> <!--
      COBOLData.OccuringKeyInfo.occursKeyFieldInfo -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1867'>
  <Model.ModelElement.name>redefinedByField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1868'/> <!-- COBOLData.Redefines.redefinedByField
      -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1869'>
  <Model.ModelElement.name>redefinedField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```

```

<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a1870'/> <!-- COBOLData.Redefines.redefinedField -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1871'>
  <Model.ModelElement.name>paddedFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1873'/> <!-- COBOLData.PaddingField.paddedFD -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1874'>
  <Model.ModelElement.name>linageInfo</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1875'/> <!-- COBOLData.LinageInfo -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1876'/> <!-- COBOLData.LinageField.linageInfo -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1877'>
  <Model.ModelElement.name>dependingFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1878'/> <!-- COBOLData.FDDepending.dependingFD -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1879'>
  <Model.ModelElement.name>statusFD</Model.ModelElement.name>

```

```

</Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>-1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>true</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a1880'/> <!-- COBOLData.FDStatusID.statusFD -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1881'>
  <Model.ModelElement.name>firstRenames</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1882'/> <!-- COBOLData.Renames -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1883'/> <!-- COBOLData.RenamesFirst.firstRenames -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1884'>
  <Model.ModelElement.name>thruRenames</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1882'/> <!-- COBOLData.Renames -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1885'/> <!-- COBOLData.RenamesThru.thruRenames -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1886'>
  <Model.ModelElement.name>relativeFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```

```

    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a1887'/> <!--
        COBOLData.RelativeOffsetField.relativeFD -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1888'>
  <Model.ModelElement.name>Usage</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a114'/> <!-- Foundation.Core.DataType -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1872'>
  <Model.ModelElement.name>COBOLFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1099'/> <!-- Record.RecordFile -->
    <Model.Class xmi.idref='a112'/> <!-- Foundation.Core.Class -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1889'>
      <Model.ModelElement.name>organization</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a1890'/> <!-- COBOLData.FileOrganization -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a1891'>
      <Model.ModelElement.name>accessMode</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a1892'/> <!-- COBOLData.AccessType -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1893'>
  <Model.ModelElement.name>isOptional</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1894'>
  <Model.ModelElement.name>reserveAreas</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1895'>
  <Model.ModelElement.name>assignTo</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1896'>
  <Model.ModelElement.name>codeSetLit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1897'>
  <Model.ModelElement.name>blockSizeUnit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a1898'/> <!-- COBOLData.BlockKind -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1899'>
  <Model.ModelElement.name>minBlocks</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1900'>
  <Model.ModelElement.name>maxBlocks</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1901'>
  <Model.ModelElement.name>minRecords</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1902'>
  <Model.ModelElement.name>maxRecords</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1903'>
  <Model.ModelElement.name>labelKind</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a1904'/> <!-- COBOLData.LabelKind -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1905'>
  <Model.ModelElement.name>isExternal</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1906'>
  <Model.ModelElement.name>isGlobal</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1907'>
  <Model.ModelElement.name>padLiteral</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a1908'>
  <Model.ModelElement.name>fileSection</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1909'/> <!-- COBOLData.FileSection -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1910'/> <!-- COBOLData.FileSectionFD.fileSection -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1911'>
  <Model.ModelElement.name>linagelInfo</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>4</XML.field> <!-- upper -->
    <XML.field>>true</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1875'/> <!-- COBOLData.LinagelInfo -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1912'/> <!-- COBOLData.LineageInfoField.linagelInfo -
    ->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1913'>
  <Model.ModelElement.name>statusID</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->

```



```

    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1914'/> <!-- COBOLData.FDStatusID.statusID -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1915'>
  <Model.ModelElement.name>dependsOn</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1916'/> <!-- COBOLData.FDDepending.dependsOn --
    >
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1917'>
  <Model.ModelElement.name>padField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1918'/> <!-- COBOLData.PaddingField.padField -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1919'>
  <Model.ModelElement.name>relativeField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1920'/> <!--
    COBOLData.RelativeOffsetField.relativeField -->

```

```

    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1882'>
  <Model.ModelElement.name>Renames</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a105'/> <!-- Foundation.Core.Feature -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Reference xmi.id='a1921'>
  <Model.ModelElement.name>firstField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1922'/> <!-- COBOLData.RenamesFirst.firstField -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1923'>
  <Model.ModelElement.name>thruField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1924'/> <!-- COBOLData.RenamesThru.thruField -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1925'>
  <Model.ModelElement.name>Section</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a103'/> <!-- Foundation.Core.Classifier -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>

```

```

<Model.Reference xmi.id='a1926'>
  <Model.ModelElement.name>record</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>true</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1093'> <!-- Record.RecordDef -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1927'> <!-- COBOLData.SectionRecord.record -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1928'>
  <Model.ModelElement.name>COBOLFDIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a788'> <!-- KeysIndexes.Index -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  <Model.Attribute xmi.id='a1929'>
    <Model.ModelElement.name>isAlternate</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1875'>
  <Model.ModelElement.name>LinageInfo</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  <Model.Attribute xmi.id='a1930'>
    <Model.ModelElement.name>value</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>

```

```

<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1931'>
  <Model.ModelElement.name>type</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a1932'/> <!-- COBOLData.LinageInfoType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a1933'>
  <Model.ModelElement.name>cobolFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1934'/> <!-- COBOLData.LineageInfoField.cobolFD -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a1935'>
  <Model.ModelElement.name>cobolField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1936'/> <!-- COBOLData.LinageField.cobolField -->
  </Model.Reference.referencedEnd>
</Model.Reference>

```

```

</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1937'>
  <Model.ModelElement.name>ReportWriterSection</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1925'> <!-- COBOLData.Section -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1938'>
  <Model.ModelElement.name>WorkingStorageSection</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1925'> <!-- COBOLData.Section -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1909'>
  <Model.ModelElement.name>FileSection</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1925'> <!-- COBOLData.Section -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a1939'>
      <Model.ModelElement.name>cobolFD</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'>
      <Model.Feature.visibility xmi.value='public_vis'>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>true</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1872'> <!-- COBOLData.COBOLFD -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a1940'> <!-- COBOLData.FileSectionFD.cobolFD -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1941'>
  <Model.ModelElement.name>LinkageSection</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>

```

```

<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1925'/> <!-- COBOLData.Section -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1863'>
  <Model.ModelElement.name>OccursKey</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a1942'>
      <Model.ModelElement.name>isAscending</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Reference xmi.id='a1943'>
      <Model.ModelElement.name>occursKeyField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a1944'/> <!--
          COBOLData.OccuringKeyInfo.occursKeyField -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
    <Model.Reference xmi.id='a1945'>
      <Model.ModelElement.name>occursKeyOf</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>

```

```

</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a1946'/> <!--
    COBOLData.OccursKeyField.occursKeyOf -->
</Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.DataType xmi.id='a1847'>
  <Model.ModelElement.name>SignKindType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeEnum>
      <XMI.CorbaTcEnum xmi.tcName='SignKindType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='sk_unsigned'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='sk_leadingSign'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='sk_trailingSign'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='sk_leadingSepSign'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='sk_trailingSepSign'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeEnum>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a1890'>
  <Model.ModelElement.name>FileOrganization</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeEnum>
      <XMI.CorbaTcEnum xmi.tcName='FileOrganization' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='fo_unspecified'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='fo_indexed'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='fo_relative'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='fo_sequential'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeEnum>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a1892'>
  <Model.ModelElement.name>AccessType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeEnum>
      <XMI.CorbaTcEnum xmi.tcName='AccessType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='at_unspecified'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='at_dynamic'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='at_random'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='at_sequential'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeEnum>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>

```

```

    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a1898'>
  <Model.ModelElement.name>BlockKind</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='BlockKind' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='bk_records'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='bk_characters'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a1904'>
  <Model.ModelElement.name>LabelKind</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='LabelKind' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='lk_unspecified'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='lk_standard'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='lk_omitted'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a1932'>
  <Model.ModelElement.name>LinageInfoType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='LinageInfoType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='li_linage'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='li_linageFooting'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='li_linageTop'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='li_linageBottom'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.Association xmi.id='a1947'>
  <Model.ModelElement.name>OccursDependingOn</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>

```



```

<Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a1859'>
    <Model.ModelElement.name>dependingOnField</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a1861'>
    <Model.ModelElement.name>occurringField</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1948'>
  <Model.ModelElement.name>FDStatusID</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1914'>
      <Model.ModelElement.name>statusID</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1880'>
      <Model.ModelElement.name>statusFD</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>

```

```

    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1949'>
  <Model.ModelElement.name>LinageField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1936'>
      <Model.ModelElement.name>cobolField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1876'>
      <Model.ModelElement.name>linageInfo</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1875'/> <!-- COBOLData.LinageInfo -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1950'>
  <Model.ModelElement.name>LineageInfoField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1912'>
      <Model.ModelElement.name>linageInfo</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>

```

```

<Model.AssociationEnd.aggregation xmi.value='none'/>
<Model.AssociationEnd.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>4</XML.field> <!-- upper -->
  <XML.field>true</XML.field> <!-- is_ordered -->
  <XML.field>true</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a1875'/> <!-- COBOLData.LinagelInfo -->
</Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a1934'>
  <Model.ModelElement.name>cobolFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='composite'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1951'>
  <Model.ModelElement.name>FileSectionFD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1940'>
      <Model.ModelElement.name>cobolFD</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>true</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1910'>
      <Model.ModelElement.name>fileSection</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>

```

```

    <Model.Class xmi.idref='a1909'/> <!-- COBOLData.FileSection -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1952'>
  <Model.ModelElement.name>Redefines</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1870'>
      <Model.ModelElement.name>redefinedField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1868'>
      <Model.ModelElement.name>redefinedByField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1953'>
  <Model.ModelElement.name>FDDDepending</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1916'>
      <Model.ModelElement.name>dependsOn</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>

```

```

    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a1878'>
    <Model.ModelElement.name>dependingFD</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1954'>
  <Model.ModelElement.name>OccursKeyField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1864'>
      <Model.ModelElement.name>occursKeyInfo</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>true</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1863'/> <!-- COBOLData.OccursKey -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1946'>
      <Model.ModelElement.name>occursKeyOf</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1955'>
  <Model.ModelElement.name>OccuringKeyInfo</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='true'/>
<Model.GeneralizableElement.isLeaf xmi.value='true'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a1944'>
    <Model.ModelElement.name>occursKeyField</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a1866'>
    <Model.ModelElement.name>occursKeyFieldInfo</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1863'/> <!-- COBOLData.OccursKey -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1956'>
  <Model.ModelElement.name>PaddingField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1873'>
      <Model.ModelElement.name>paddedFD</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a1918'>
    <Model.ModelElement.name>padField</Model.ModelElement.name>

```

```

</Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.AssociationEnd.isNavigable xmi.value='true'/>
<Model.AssociationEnd.aggregation xmi.value='none'/>
<Model.AssociationEnd.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
</Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1957'>
  <Model.ModelElement.name>RenamesFirst</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1922'>
      <Model.ModelElement.name>firstField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1883'>
      <Model.ModelElement.name>firstRenames</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1882'/> <!-- COBOLData.Renames -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1958'>
  <Model.ModelElement.name>RenamesThru</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>

```

```

<Model.AssociationEnd xmi.id='a1924'>
  <Model.ModelElement.name>thruField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a1885'>
  <Model.ModelElement.name>thruRenames</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1882'/> <!-- COBOLData.Renames -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1959'>
  <Model.ModelElement.name>RelativeOffsetField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1920'>
      <Model.ModelElement.name>relativeField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1844'/> <!-- COBOLData.COBOLField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1887'>
      <Model.ModelElement.name>relativeFD</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
      </Model.AssociationEnd.multiplicity>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```



```

        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1872'/> <!-- COBOLData.COBOLFD -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a1960'>
  <Model.ModelElement.name>SectionRecord</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1927'>
      <Model.ModelElement.name>record</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>true</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1093'/> <!-- Record.RecordDef -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a1961'>
      <Model.ModelElement.name>section</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1925'/> <!-- COBOLData.Section -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a1962'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a1963'>
  <Model.ModelElement.name>Record</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a1088'/> <!-- Record -->
  </Model.Import.importedNamespace>

```

```

    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a1964'>
    <Model.ModelElement.name>KeysIndexes</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a781'/> <!-- KeysIndexes -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Tag xmi.id='a1965'>
    <Model.ModelElement.name>org.omg.xmi.namespace=CWMXCOBOL</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>CWMXCOBOL</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1843'/> <!-- COBOLData -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1966'>
    <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1843'/> <!-- COBOLData -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1967'>
    <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.java.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1843'/> <!-- COBOLData -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1968'>
    <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=sk_</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>sk_</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.DataType xmi.idref='a1847'/> <!-- COBOLData.SignKindType -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a1969'>
    <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=fo_</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>fo_</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>

```

```

    <Model.DataType xmi.idref='a1890'/> <!-- COBOLData.FileOrganization -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1970'>
  <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=at_</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>at_</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.DataType xmi.idref='a1892'/> <!-- COBOLData.AccessType -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1971'>
  <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=bk_</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>bk_</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.DataType xmi.idref='a1898'/> <!-- COBOLData.BlockKind -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1972'>
  <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=lk_</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>lk_</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.DataType xmi.idref='a1904'/> <!-- COBOLData.LabelKind -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a1973'>
  <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=li_</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>li_</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.DataType xmi.idref='a1932'/> <!-- COBOLData.LinageInfoType -->
  </Model.Tag.elements>
</Model.Tag>
</Model.Namespace.contents>
</Model.Package>
<Model.Package xmi.id='a1974'>
  <Model.ModelElement.name>DMSII</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a1975'>
      <Model.ModelElement.name>Database</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    </Model.Class>
  </Model.Namespace.contents>
</Model.Package>

```

```

<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
  <Model.Class xmi.idref='a707'/> <!-- Model_Management.Package -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a1976'>
    <Model.ModelElement.name>isLogical</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a1977'>
    <Model.ModelElement.name>guardFile</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a1978'>
    <Model.ModelElement.name>source</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1979'>
  <Model.ModelElement.name>Remap</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>

```

```

<Model.Class xmi.idref='a1093'/> <!-- Record.RecordDef -->
<Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
<Model.Attribute xmi.id='a1980'>
  <Model.ModelElement.name>isRequiredAll</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1981'>
  <Model.ModelElement.name>isReadOnlyAll</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1982'>
  <Model.ModelElement.name>isGivingException</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1983'>
  <Model.ModelElement.name>selectCondition</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.Class xmi.idref='a6'/> <!-- Foundation.Data_Types.Expression -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a1984'>
  <Model.ModelElement.name>structure</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
</Model.TypedElement.type>
  <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
</Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1985'/> <!-- DMSII.RemappedStructure.structure -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1986'>
  <Model.ModelElement.name>DataSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
    <Model.Class xmi.idref='a1093'/> <!-- Record.RecordDef -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
  <Model.Attribute xmi.id='a1987'>
    <Model.ModelElement.name>isGlobal</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
  </Model.Attribute>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1988'>
  <Model.ModelElement.name>organization</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1989'>
  <Model.ModelElement.name>reorganize</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1990'>
  <Model.ModelElement.name>isRequiredAll</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a1991'>
  <Model.ModelElement.name>partitioningSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1993'/> <!-- DMSII.DataSetPartitionSet.partitioningSet -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1994'>
  <Model.ModelElement.name>DataItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>

```

```

<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1089'/> <!-- Record.Field -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a1995'>
    <Model.ModelElement.name>nullValue</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a1996'>
    <Model.ModelElement.name>isRequired</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a1997'>
    <Model.ModelElement.name>size</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a1998'>
    <Model.ModelElement.name>scaleFactor</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>

```



```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a1999'>
  <Model.ModelElement.name>isSigned</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2000'>
  <Model.ModelElement.name>occurs</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2001'>
  <Model.ModelElement.name>isVirtual</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2002'>
  <Model.ModelElement.name>virtualExpression</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2003'>
  <Model.ModelElement.name>isKanji</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2004'>
  <Model.ModelElement.name>ccsVersion</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2005'>
  <Model.ModelElement.name>isGemcosLiteral</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2006'>
  <Model.ModelElement.name>isGemcosData</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27' /> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2007'>
  <Model.ModelElement.name>isGemcosSSN</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27' /> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2008'>
  <Model.ModelElement.name>isGemcosDBSN</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27' /> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2009'>
  <Model.ModelElement.name>isComsProgram</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27' /> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2010'>
  <Model.ModelElement.name>isComsID</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />

```

```

<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2011'>
  <Model.ModelElement.name>isComsLocator</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2012'>
  <Model.ModelElement.name>isComsOutpQ</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2013'>
  <Model.ModelElement.name>fieldBit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>48</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2014'/> <!-- DMSII.FieldBit -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2015'/> <!-- DMSII.FieldBits.fieldBit -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2016'>
  <Model.ModelElement.name>occursDataItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```

```

<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
<Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2017'/> <!-- DMSII.OccursDepending.occursDataItem
-->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2018'>
<Model.ModelElement.name>occurringDataItem</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
<XMI.field>0</XMI.field> <!-- lower -->
<XMI.field>-1</XMI.field> <!-- upper -->
<XMI.field>>false</XMI.field> <!-- is_ordered -->
<XMI.field>>true</XMI.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
<Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2019'/> <!--
DMSII.OccursDepending.occuringDataItem -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2020'>
<Model.ModelElement.name>keyDataSet</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
<XMI.field>0</XMI.field> <!-- lower -->
<XMI.field>-1</XMI.field> <!-- upper -->
<XMI.field>>false</XMI.field> <!-- is_ordered -->
<XMI.field>>true</XMI.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
<Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2021'/> <!-- DMSII.KeyDataItem.keyDataSet -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2022'>
<Model.ModelElement.name>structure</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
<XMI.field>0</XMI.field> <!-- lower -->
<XMI.field>1</XMI.field> <!-- upper -->
<XMI.field>>false</XMI.field> <!-- is_ordered -->
<XMI.field>>false</XMI.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
<Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2023'/> <!-- DMSII.DataItemStructure.structure -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2024'>

```

```

<Model.ModelElement.name>remapItem</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>true</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2025'/> <!-- DMSII.RemapItem -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2026'/> <!-- DMSII.RemapItems.remapItem -->
</Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2027'>
  <Model.ModelElement.name>VariableFormatPart</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
    <Model.Class xmi.idref='a1093'/> <!-- Record.RecordDef -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2028'>
  <Model.ModelElement.name>vfLabel</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2029'>
  <Model.ModelElement.name>selectCondition</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a13'/> <!-- Foundation.Data_Types.BooleanExpression -->
  </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>

```

```

</Model.Class>
<Model.Class xmi.id='a2030'>
  <Model.ModelElement.name>SetStructure</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a788'/> <!-- KeysIndexes.Index -->
    <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2031'>
  <Model.ModelElement.name>duplicates</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a1992'>
  <Model.ModelElement.name>Set</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2030'/> <!-- DMSII.SetStructure -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2032'>
  <Model.ModelElement.name>setType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2033'>
  <Model.ModelElement.name>reorganize</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>

```

```

    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2034'>
  <Model.ModelElement.name>partitioningSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2035'/> <!-- DMSII.SetPartitionSet.partitioningSet -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2036'>
  <Model.ModelElement.name>partitionedSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2037'/> <!-- DMSII.SetPartitionSet.partitionedSet -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2038'>
  <Model.ModelElement.name>keyDataItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>true</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2039'/> <!-- DMSII.KeyDataItem.keyDataItem -->
  </Model.Reference.referencedEnd>
</Model.Reference>

```



```

<Model.Reference xmi.id='a2040'>
  <Model.ModelElement.name>partitionedDataSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1986'> <!-- DMSII.DataSet -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2041'> <!--
      DMSII.DataSetPartitionSet.partitionedDataSet -->
    </Model.AssociationEnd>
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2042'>
  <Model.ModelElement.name>Access</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2030'> <!-- DMSII.SetStructure -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2043'>
  <Model.ModelElement.name>Subset</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1992'> <!-- DMSII.Set -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2044'>
  <Model.ModelElement.name>AutomaticSubset</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2043'> <!-- DMSII.Subset -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  <Model.Attribute xmi.id='a2045'>
    <Model.ModelElement.name>condition</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>

```

```

    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a13'/> <!-- Foundation.Data_Types.BooleanExpression -->
  </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2046'>
  <Model.ModelElement.name>KeyItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a793'/> <!-- KeysIndexes.IndexedFeature -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2047'>
      <Model.ModelElement.name>collation</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2025'>
  <Model.ModelElement.name>RemapItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1089'/> <!-- Record.Field -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2048'>
      <Model.ModelElement.name>occurs</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2049'>
  <Model.ModelElement.name>isRequired</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2050'>
  <Model.ModelElement.name>isHidden</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2051'>
  <Model.ModelElement.name>isReadOnly</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2052'>
  <Model.ModelElement.name>isGivingException</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>

```

```

<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2053'>
  <Model.ModelElement.name>isVirtual</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2054'>
  <Model.ModelElement.name>virtualExpression</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2055'>
  <Model.ModelElement.name>dataItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2056'/> <!-- DMSII.RemapItems.dataItem -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2014'>
  <Model.ModelElement.name>FieldBit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>

```

```

<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Reference xmi.id='a2057'>
    <Model.ModelElement.name>dataItem</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2058'/> <!-- DMSII.FieldBits.dataItem -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2059'>
  <Model.ModelElement.name>Remark</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2060'>
      <Model.ModelElement.name>text</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2061'>
  <Model.ModelElement.name>PhysicalDatabase</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a707'/> <!-- Model_Management.Package -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>

```

```

    </Model.Namespace.contents>
  </Model.Class>
  <Model.Class xmi.id='a2062'>
    <Model.ModelElement.name>PhysicalDataSet</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.Class.isSingleton xmi.value='false'/>
    <Model.GeneralizableElement.supertypes>
      <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
    </Model.GeneralizableElement.supertypes>
    <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2063'>
  <Model.ModelElement.name>DASDLComment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a888'/> <!-- BusinessInformation.Description -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2064'>
  <Model.ModelElement.name>PhysicalSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2065'>
  <Model.ModelElement.name>PhysicalDataSetOverride</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a105'/> <!-- Foundation.Core.Feature -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2066'>
  <Model.ModelElement.name>PhysicalSetOverride</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a105'/> <!-- Foundation.Core.Feature -->
  </Model.GeneralizableElement.supertypes>

```

```

</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2067'>
  <Model.ModelElement.name>PhysicalAccessOverride</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a105'/> <!-- Foundation.Core.Feature -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2068'>
  <Model.ModelElement.name>DASDLProperty</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2069'>
      <Model.ModelElement.name>text</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Reference xmi.id='a2070'>
      <Model.ModelElement.name>owner</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a2071'/> <!-- DMSII.DASDLPropertyOwner.owner -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>

```

```

<Model.Association xmi.id='a2072'>
  <Model.ModelElement.name>OccursDepending</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2019'>
      <Model.ModelElement.name>occurringDataItem</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2017'>
      <Model.ModelElement.name>occursDataItem</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2073'>
  <Model.ModelElement.name>KeyDataItem</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2039'>
      <Model.ModelElement.name>keyDataItem</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>true</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```



```

<Model.AssociationEnd xmi.id='a2021'>
  <Model.ModelElement.name>keyDataSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2074'>
  <Model.ModelElement.name>SetPartitionSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2037'>
      <Model.ModelElement.name>partitionedSet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2035'>
      <Model.ModelElement.name>partitioningSet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2075'>
  <Model.ModelElement.name>DataSetPartitionSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>

```

```

<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a1993'>
    <Model.ModelElement.name>partitioningSet</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1992'/> <!-- DMSII.Set -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2041'>
    <Model.ModelElement.name>partitionedDataSet</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a1986'/> <!-- DMSII.DataSet -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2076'>
  <Model.ModelElement.name>FieldBits</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2058'>
      <Model.ModelElement.name>dataItem</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2015'>
      <Model.ModelElement.name>fieldBit</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->

```

```

        <XML.field>48</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
        <Model.Class xmi.idref='a2014'/> <!-- DMSII.FieldBit -->
    </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2077'>
    <Model.ModelElement.name>DataItemStructure</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='true'/>
    <Model.GeneralizableElement.isLeaf xmi.value='true'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.Association.isDerived xmi.value='false'/>
    <Model.Namespace.contents>
        <Model.AssociationEnd xmi.id='a2023'>
            <Model.ModelElement.name>structure</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>
                <XML.field>0</XML.field> <!-- lower -->
                <XML.field>1</XML.field> <!-- upper -->
                <XML.field>>false</XML.field> <!-- is_ordered -->
                <XML.field>>false</XML.field> <!-- is_unique -->
            </Model.AssociationEnd.multiplicity>
            <Model.AssociationEnd.isChangeable xmi.value='true'/>
            <Model.TypedElement.type>
                <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
            </Model.TypedElement.type>
        </Model.AssociationEnd>
        <Model.AssociationEnd xmi.id='a2078'>
            <Model.ModelElement.name>dataitem</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>
                <XML.field>0</XML.field> <!-- lower -->
                <XML.field>-1</XML.field> <!-- upper -->
                <XML.field>>false</XML.field> <!-- is_ordered -->
                <XML.field>>true</XML.field> <!-- is_unique -->
            </Model.AssociationEnd.multiplicity>
            <Model.AssociationEnd.isChangeable xmi.value='true'/>
            <Model.TypedElement.type>
                <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
            </Model.TypedElement.type>
        </Model.AssociationEnd>
    </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2079'>
    <Model.ModelElement.name>DASDLPropertyOwner</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='true'/>
    <Model.GeneralizableElement.isLeaf xmi.value='true'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.Association.isDerived xmi.value='false'/>
    <Model.Namespace.contents>
        <Model.AssociationEnd xmi.id='a2071'>
            <Model.ModelElement.name>owner</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>

```

```

<Model.AssociationEnd.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
</Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2080'>
  <Model.ModelElement.name>dasdlProperty</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>true</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2068'/> <!-- DMSII.DASDLProperty -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2081'>
  <Model.ModelElement.name>RemapItems</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2026'>
      <Model.ModelElement.name>remapItem</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2025'/> <!-- DMSII.RemapItem -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2056'>
      <Model.ModelElement.name>dataItem</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a1994'/> <!-- DMSII.DataItem -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2082'>
  <Model.ModelElement.name>RemappedStructure</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a1985'>
      <Model.ModelElement.name>structure</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a118'/> <!-- Foundation.Core.StructuralFeature -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2083'>
    <Model.ModelElement.name>remap</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1979'/> <!-- DMSII.Remap -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a2084'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2085'>
  <Model.ModelElement.name>Expressions</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a820'/> <!-- Expressions -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2086'>
  <Model.ModelElement.name>Model_Management</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a706'/> <!-- Model_Management -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2087'>
    <Model.ModelElement.name>Record</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a1088'/> <!-- Record -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2088'>
    <Model.ModelElement.name>KeysIndexes</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a781'/> <!-- KeysIndexes -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2089'>
    <Model.ModelElement.name>BusinessInformation</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a850'/> <!-- BusinessInformation -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Tag xmi.id='a2090'>
    <Model.ModelElement.name>org.omg.xmi.namespace=CWMXDMSII</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>CWMXDMSII</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1974'/> <!-- DMSII -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2091'>
    <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1974'/> <!-- DMSII -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2092'>
    <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.java.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a1974'/> <!-- DMSII -->
    </Model.Tag.elements>
  </Model.Tag>

```

```

</Model.Tag>
<Model.Tag xmi.id='a2093'>
  <Model.ModelElement.name>org.omg.mof.idl_substitute_name=dmsiiSetType</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.idl_substitute_name</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>dmsiiSetType</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Attribute xmi.idref='a2032'> <!-- DMSII.Set.setType -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a2094'>
  <Model.ModelElement.name>org.omg.mof.java_substitute_name=dmsiiSetType</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.java_substitute_name</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>dmsiiSetType</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Attribute xmi.idref='a2032'> <!-- DMSII.Set.setType -->
  </Model.Tag.elements>
</Model.Tag>
</Model.Namespace.contents>
</Model.Package>
<Model.Package xmi.id='a2095'>
  <Model.ModelElement.name>Essbase</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a2096'>
      <Model.ModelElement.name>Alias</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'>
      <Model.GeneralizableElement.isLeaf xmi.value='false'>
      <Model.GeneralizableElement.isAbstract xmi.value='false'>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'>
      <Model.Class.isSingleton xmi.value='false'>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1129'> <!-- Multidimensional.DimensionedObject -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
      </Model.Namespace.contents>
    </Model.Class>
    <Model.Class xmi.id='a2097'>
      <Model.ModelElement.name>Application</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'>
      <Model.GeneralizableElement.isLeaf xmi.value='false'>
      <Model.GeneralizableElement.isAbstract xmi.value='false'>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'>
      <Model.Class.isSingleton xmi.value='false'>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a707'> <!-- Model_Management.Package -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
      </Model.Namespace.contents>
    </Model.Class>
    <Model.Class xmi.id='a2098'>
      <Model.ModelElement.name>Comment</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'>
      <Model.GeneralizableElement.isLeaf xmi.value='false'>

```

```

<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2099'>
  <Model.ModelElement.name>Consolidation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2100'>
  <Model.ModelElement.name>CurrencyConversion</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2101'>
  <Model.ModelElement.name>DataStorage</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2102'>
  <Model.ModelElement.name>Database</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1140'/> <!-- Multidimensional.Schema -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2103'>
      <Model.ModelElement.name>isCurrency</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```



```

<Model.StructuralFeature.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2104'>
  <Model.ModelElement.name>outline</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2105'/> <!-- Essbase.Outline -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2106'/> <!-- Essbase.DatabaseOwnsOutline.outline -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2107'>
  <Model.ModelElement.name>Dimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1127'/> <!-- Multidimensional.Dimension -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2108'>
  <Model.ModelElement.name>type</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2109'/> <!-- Essbase.DimensionType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2110'>
  <Model.ModelElement.name>isDense</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>

```

```

    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2111'>
  <Model.ModelElement.name>outline</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2105'/> <!-- Essbase.Outline -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2112'/> <!--
      Essbase.OutlineReferencesDimensions.outline -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2113'>
  <Model.ModelElement.name>Formula</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2114'>
  <Model.ModelElement.name>Generation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2115'>
  <Model.ModelElement.name>ImmediateParent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>

```

```

<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2116'>
  <Model.ModelElement.name>Level</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2117'>
  <Model.ModelElement.name>Partition</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='true'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1330'/> <!-- Olap.CubeRegion -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  <Model.Attribute xmi.id='a2118'>
    <Model.ModelElement.name>isSource</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2119'>
    <Model.ModelElement.name>isShared</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2120'>

```

```

    <Model.ModelElement.name>formula</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a821'/> <!-- Expressions.ExpressionNode -->
    </Model.TypedElement.type>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2121'>
  <Model.ModelElement.name>LinkedPartition</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2117'/> <!-- Essbase.Partition -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2122'>
  <Model.ModelElement.name>MemberName</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2123'>
  <Model.ModelElement.name>OLAPServer</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a939'/> <!-- SoftwareDeployment.DataManager -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2105'>
  <Model.ModelElement.name>Outline</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>

```

```

<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a55'/> <!-- Foundation.Core.Namespace -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2124'>
    <Model.ModelElement.name>aliasTableName</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a2125'>
    <Model.ModelElement.name>database</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2102'/> <!-- Essbase.Database -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2126'/> <!-- Essbase.DatabaseOwnsOutline.database
        -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  <Model.Reference xmi.id='a2127'>
    <Model.ModelElement.name>dimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2107'/> <!-- Essbase.Dimension -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2128'/> <!--
        Essbase.OutlineReferencesDimensions.dimension -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2129'>
  <Model.ModelElement.name>ReplicatedPartition</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>

```

```

<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a2117'/> <!-- Essbase.Partition -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2130'>
  <Model.ModelElement.name>TimeBalance</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2131'>
  <Model.ModelElement.name>TransparentPartition</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2117'/> <!-- Essbase.Partition -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2132'>
  <Model.ModelElement.name>TwoPassCalculation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2133'>
  <Model.ModelElement.name>UDA</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2134'>
  <Model.ModelElement.name>VarianceReporting</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.GeneralizableElement.isRoot xmi.value='false'/>
<Model.GeneralizableElement.isLeaf xmi.value='false'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.DataType xmi.id='a2109'>
  <Model.ModelElement.name>DimensionType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='DimensionType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='ess_none'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='ess_accounts'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='ess_time'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='ess_country'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='ess_currencyPartition'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='ess_attribute'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.Association xmi.id='a2135'>
  <Model.ModelElement.name>DatabaseOwnsOutline</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2126'>
      <Model.ModelElement.name>database</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XMI.field>1</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>false</XMI.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2102'/> <!-- Essbase.Database -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2106'>
      <Model.ModelElement.name>outline</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XMI.field>1</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>false</XMI.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2105'/> <!-- Essbase.Outline -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2136'>
  <Model.ModelElement.name>OutlineReferencesDimensions</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2128'>
      <Model.ModelElement.name>dimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XMI.field>0</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>true</XMI.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2107'/> <!-- Essbase.Dimension -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2112'>
      <Model.ModelElement.name>outline</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XMI.field>1</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>false</XMI.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2105'/> <!-- Essbase.Outline -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a2137'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2138'>
  <Model.ModelElement.name>Expressions</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a820'/> <!-- Expressions -->
  </Model.Import.importedNamespace>

```



```

</Model.Import>
<Model.Import xmi.id='a2139'>
  <Model.ModelElement.name>Multidimensional</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'>
  <Model.Import.isClustered xmi.value='false'>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a1126'> <!-- Multidimensional -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2140'>
  <Model.ModelElement.name>Model_Management</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'>
  <Model.Import.isClustered xmi.value='false'>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a706'> <!-- Model_Management -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2141'>
  <Model.ModelElement.name>SoftwareDeployment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'>
  <Model.Import.isClustered xmi.value='false'>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a915'> <!-- SoftwareDeployment -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2142'>
  <Model.ModelElement.name>Olap</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'>
  <Model.Import.isClustered xmi.value='false'>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a1323'> <!-- Olap -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Tag xmi.id='a2143'>
  <Model.ModelElement.name>org.omg.xmi.namespace=CWMXESSB</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>CWMXESSB</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a2095'> <!-- Essbase -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a2144'>
  <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.resource</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>org.omg.cwm.x.resource</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a2095'> <!-- Essbase -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a2145'>
  <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.resource</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>org.omg.java.cwm.x.resource</XML.any>

```

```

    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2095'/> <!-- Essbase -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2146'>
    <Model.ModelElement.name>org.omg.xmi.enumerationUnprefix=ess_</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.enumerationUnprefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>ess_</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.DataType xmi.idref='a2109'/> <!-- Essbase.DimensionType -->
    </Model.Tag.elements>
  </Model.Tag>
</Model.Namespace.contents>
</Model.Package>
<Model.Package xmi.id='a2147'>
  <Model.ModelElement.name>Express</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a2148'>
      <Model.ModelElement.name>Dimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='true'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1127'/> <!-- Multidimensional.Dimension -->
      </Model.GeneralizableElement.supertypes>
    </Model.Namespace.contents>
  </Model.Namespace.contents>
  </Model.Class>
  <Model.Class xmi.id='a2149'>
    <Model.ModelElement.name>Database</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.Class.isSingleton xmi.value='false'/>
    <Model.GeneralizableElement.supertypes>
      <Model.Class xmi.idref='a1140'/> <!-- Multidimensional.Schema -->
    </Model.GeneralizableElement.supertypes>
  </Model.Namespace.contents>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2150'>
  <Model.ModelElement.name>Conjoint</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2151'>

```

```

<Model.ModelElement.name>searchAlgorithm</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XMI.field>1</XMI.field> <!-- lower -->
  <XMI.field>1</XMI.field> <!-- upper -->
  <XMI.field>>false</XMI.field> <!-- is_ordered -->
  <XMI.field>>false</XMI.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2152'>
  <Model.ModelElement.name>pageSpace</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XMI.field>0</XMI.field> <!-- lower -->
    <XMI.field>1</XMI.field> <!-- upper -->
    <XMI.field>>false</XMI.field> <!-- is_ordered -->
    <XMI.field>>false</XMI.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2153'>
  <Model.ModelElement.name>Program</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a192'/> <!-- Foundation.Core.Component -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  <Model.Attribute xmi.id='a2154'>
    <Model.ModelElement.name>program</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XMI.field>1</XMI.field> <!-- lower -->
      <XMI.field>1</XMI.field> <!-- upper -->
      <XMI.field>>false</XMI.field> <!-- is_ordered -->
      <XMI.field>>false</XMI.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2155'>
    <Model.ModelElement.name>returnDimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>

```

```

<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2156'>
  <Model.ModelElement.name>Model</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a192'/> <!-- Foundation.Core.Component -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2157'>
      <Model.ModelElement.name>content</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2158'>
  <Model.ModelElement.name>Variable</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2159'>
      <Model.ModelElement.name>storageType</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2160'>
  <Model.ModelElement.name>pageSpace</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2161'>
  <Model.ModelElement.name>width</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2162'>
  <Model.ModelElement.name>Formula</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2163'>
      <Model.ModelElement.name>expression</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2164'>
  <Model.ModelElement.name>ValueSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
  <Model.Attribute xmi.id='a2165'>
    <Model.ModelElement.name>isTemp</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a2166'>
    <Model.ModelElement.name>referenceDimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2167'/> <!--
        Express.ValueSetReferenceDimension.referenceDimension -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2168'>
  <Model.ModelElement.name>Relation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1129'/> <!-- Multidimensional.DimensionedObject -->
  </Model.GeneralizableElement.supertypes>

```

```

<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2169'>
    <Model.ModelElement.name>isTemp</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2170'>
    <Model.ModelElement.name>pageSpace</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a2171'>
    <Model.ModelElement.name>referenceDimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2148'> <!-- Express.Dimension -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2172'> <!--
        Express.RelationReferenceDimension.referenceDimension -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2173'>
  <Model.ModelElement.name>Worksheet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a112'> <!-- Foundation.Core.Class -->
  </Model.GeneralizableElement.supertypes>

```

```

<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2174'>
    <Model.ModelElement.name>isTemp</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a2175'>
    <Model.ModelElement.name>columnDimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2148'> <!-- Express.Dimension -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2176'> <!--
        Express.WorksheetColumnDimension.columnDimension -->
      </Model.AssociationEnd>
    </Model.Reference.referencedEnd>
  </Model.Reference>
  <Model.Reference xmi.id='a2177'>
    <Model.ModelElement.name>rowDimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2148'> <!-- Express.Dimension -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2178'> <!--
        Express.WorksheetRowDimension.rowDimension -->
      </Model.AssociationEnd>
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2179'>
  <Model.ModelElement.name>Composite</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>

```



```

<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2180'>
    <Model.ModelElement.name>searchAlgorithm</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2181'>
    <Model.ModelElement.name>pageSpace</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2182'>
  <Model.ModelElement.name>SimpleDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2183'>
      <Model.ModelElement.name>width</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

</Model.Attribute>
<Model.Attribute xmi.id='a2184'>
  <Model.ModelElement.name>isTime</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2185'>
  <Model.ModelElement.name>multiple</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2186'>
  <Model.ModelElement.name>beginningPhase</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2187'>
  <Model.ModelElement.name>endingPhase</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>

```

```

</Model.Attribute>
<Model.Attribute xmi.id='a2188'>
  <Model.ModelElement.name>searchAlgorithm</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2189'>
  <Model.ModelElement.name>pageSpace</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2190'>
  <Model.ModelElement.name>dataType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a103'> <!-- Foundation.Core.Classifier -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2191'> <!--
      Express.SimpleDimensionDataType.dataType -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2192'>
  <Model.ModelElement.name>aliasDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.TypedElement.type>

```

```

    <Model.Class xmi.idref='a2193'/> <!-- Express.AliasDimension -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2194'/> <!--
      Express.AliasDimensionBaseDimension.aliasDimension -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2193'>
  <Model.ModelElement.name>AliasDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Reference xmi.id='a2195'>
  <Model.ModelElement.name>baseDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2182'/> <!-- Express.SimpleDimension -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2196'/> <!--
      Express.AliasDimensionBaseDimension.baseDimension -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2197'>
  <Model.ModelElement.name>AggMap</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a112'/> <!-- Foundation.Core.Class -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Reference xmi.id='a2198'>
  <Model.ModelElement.name>aggMapComponent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>

```

```

    <Model.Class xmi.idref='a2199'/> <!-- Express.AggregMapComponent -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2200'/> <!--
      Express.AggregMapComponents.aggregMapComponent -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2199'>
  <Model.ModelElement.name>AggregMapComponent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2201'>
      <Model.ModelElement.name>aggregOperator</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Reference xmi.id='a2202'>
      <Model.ModelElement.name>aggregMap</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2197'/> <!-- Express.AggregMap -->
      </Model.TypedElement.type>
    </Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2203'/> <!-- Express.AggregMapComponents.aggregMap -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
  <Model.Reference xmi.id='a2204'>
    <Model.ModelElement.name>dimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
  </Model.Reference>

```

```

<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2205'/> <!--
    Express.AggMapComponentDimension.dimension -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2206'>
  <Model.ModelElement.name>relation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2168'/> <!-- Express.Relation -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2207'/> <!--
      Express.AggMapComponentRelation.relation -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2208'>
  <Model.ModelElement.name>computeClause</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2209'/> <!-- Express.PreComputeClause -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2210'/> <!-- Express.ComputeClause.computeClause
      -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2209'>
  <Model.ModelElement.name>PreComputeClause</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2211'>
      <Model.ModelElement.name>statusList</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>

```

```

<Model.StructuralFeature.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a748'/> <!-- DataTypes.Any -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2212'>
  <Model.ModelElement.name>aggMapComponent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2199'/> <!-- Express.AggMapComponent -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2213'/> <!--
      Express.ComputeClause.aggMapComponent -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Association xmi.id='a2214'>
  <Model.ModelElement.name>RelationReferenceDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2172'>
      <Model.ModelElement.name>referenceDimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2215'>
      <Model.ModelElement.name>relation</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
      </Model.AssociationEnd.multiplicity>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2168'/> <!-- Express.Relation -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2216'>
  <Model.ModelElement.name>WorksheetColumnDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2176'>
      <Model.ModelElement.name>columnDimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2217'>
      <Model.ModelElement.name>columnDimensionInWorksheet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2173'/> <!-- Express.Worksheet -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2218'>
  <Model.ModelElement.name>WorksheetRowDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2219'>
      <Model.ModelElement.name>rowDimensionInWorksheet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->

```



```

    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2173'/> <!-- Express.Worksheet -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2178'>
  <Model.ModelElement.name>rowDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2220'>
  <Model.ModelElement.name>AliasDimensionBaseDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2196'>
      <Model.ModelElement.name>baseDimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2182'/> <!-- Express.SimpleDimension -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2194'>
      <Model.ModelElement.name>aliasDimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2193'/> <!-- Express.AliasDimension -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2221'>
  <Model.ModelElement.name>ValueSetReferenceDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2167'>
      <Model.ModelElement.name>referenceDimension</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2222'>
      <Model.ModelElement.name>valueSet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2164'/> <!-- Express.ValueSet -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2223'>
  <Model.ModelElement.name>SimpleDimensionDataType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2191'>
      <Model.ModelElement.name>dataType</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a103'/> <!-- Foundation.Core.Classifier -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    </Model.TypedElement.type>
  </Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2224'>
  <Model.ModelElement.name>simpleDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2182'/> <!-- Express.SimpleDimension -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2225'>
  <Model.ModelElement.name>AggMapComponentRelation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2207'>
      <Model.ModelElement.name>relation</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2168'/> <!-- Express.Relation -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2226'>
      <Model.ModelElement.name>aggMapComponent</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2199'/> <!-- Express.AggMapComponent -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2227'>
  <Model.ModelElement.name>AggMapComponentDimension</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>

```

```

<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2205'>
    <Model.ModelElement.name>dimension</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2148'/> <!-- Express.Dimension -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2228'>
    <Model.ModelElement.name>aggMapComponent</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2199'/> <!-- Express.AggMapComponent -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2229'>
  <Model.ModelElement.name>ComputeClause</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2210'>
      <Model.ModelElement.name>computeClause</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2209'/> <!-- Express.PreComputeClause -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2213'>
      <Model.ModelElement.name>aggMapComponent</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

<Model.AssociationEnd.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2199'/> <!-- Express.AggregMapComponent -->
</Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2230'>
  <Model.ModelElement.name>AggMapComponents</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2203'>
      <Model.ModelElement.name>aggMap</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2197'/> <!-- Express.AggregMap -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2200'>
      <Model.ModelElement.name>aggMapComponent</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2199'/> <!-- Express.AggregMapComponent -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a2231'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2232'>
  <Model.ModelElement.name>DataTypes</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a728'/> <!-- DataTypes -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2233'>
    <Model.ModelElement.name>Multidimensional</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a1126'/> <!-- Multidimensional -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Tag xmi.id='a2234'>
    <Model.ModelElement.name>org.omg.xmi.namespace=CWMXExpress</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>CWMXExpress</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2147'/> <!-- Express -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2235'>
    <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2147'/> <!-- Express -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2236'>
    <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.java.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2147'/> <!-- Express -->
    </Model.Tag.elements>
  </Model.Tag>
</Model.Namespace.contents>
</Model.Package>
<Model.Package xmi.id='a2237'>
  <Model.ModelElement.name>IMSDatabase</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Package xmi.id='a2238'>
      <Model.ModelElement.name>IMSTypes</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    </Model.Package>
  </Model.Namespace.contents>
</Model.Package>

```

```

<Model.Namespace.contents>
  <Model.DataType xmi.id='a2239'>
    <Model.ModelElement.name>AccessMethodType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'>
    <Model.GeneralizableElement.isLeaf xmi.value='false'>
    <Model.GeneralizableElement.isAbstract xmi.value='false'>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'>
    <Model.DataType.typeCode>
      <XMI.CorbaTypeCode>
        <XMI.CorbaTcEnum xmi.tcName='AccessMethodType' xmi.tcId='null'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_DEDB'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_GSAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_HDAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_HIDAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_HISAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_HSAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_INDEX'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_LOGICAL'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_MSDB'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_PSINDEX'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_PHDAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_PHIDAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_SHSAM'>
          <XMI.CorbaTcEnumLabel xmi.tcName='imsam_SHISAM'>
        </XMI.CorbaTcEnum>
      </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
  </Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2240'>
  <Model.ModelElement.name>PSBLanguageType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='PSBLanguageType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imslt_ASSEM'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imslt_C'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imslt_COBOL'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imslt_PLI'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imslt_PASCAL'>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2241'>
  <Model.ModelElement.name>PCBType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='PCBType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imspt_DB'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imspt_GSAM'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imspt_TP'>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>

```

```

<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2242'>
  <Model.ModelElement.name>PositioningType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/'>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='PositioningType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsps_Single'/'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsps_Multiple'/'>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2243'>
  <Model.ModelElement.name>LTermType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/'>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='LTermType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imstp_LTERM'/'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imstp_NAME'/'>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2244'>
  <Model.ModelElement.name>RulesType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/'>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='RulesType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsrt_FIRST'/'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsrt_LAST'/'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsrt_HERE'/'>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2245'>
  <Model.ModelElement.name>ChildPointerType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/'>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='ChildPointerType' xmi.tcId='null'>

```



```

        <XMI.CorbaTcEnumLabel xmi.tcName='imscp_SNGL'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='imscp_DBLE'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='imscp_NONE'/>
    </XMI.CorbaTcEnum>
</XMI.CorbaTypeCode>
</Model.DataType.typeCode>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2246'>
    <Model.ModelElement.name>FlagsType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.DataType.typeCode>
        <XMI.CorbaTypeCode>
            <XMI.CorbaTcEnum xmi.tcName='FlagsType' xmi.tcId='null'>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsft_P'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsft_L'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsft_V'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imdf_B'/>
            </XMI.CorbaTcEnum>
        </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
    <Model.Namespace.contents>
    </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2247'>
    <Model.ModelElement.name>PointerType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.DataType.typeCode>
        <XMI.CorbaTypeCode>
            <XMI.CorbaTcEnum xmi.tcName='PointerType' xmi.tcId='null'>
                <XMI.CorbaTcEnumLabel xmi.tcName='imspn_NOTWIN'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imspn_TWIN'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imspn_HIER'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imspn_TWINBWD'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imspn_HIERBWD'/>
            </XMI.CorbaTcEnum>
        </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
    <Model.Namespace.contents>
    </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2248'>
    <Model.ModelElement.name>DeviceType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.DataType.typeCode>
        <XMI.CorbaTypeCode>
            <XMI.CorbaTcEnum xmi.tcName='DeviceType' xmi.tcId='null'>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_2305'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_2319'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_3330'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_3350'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_3375'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_3380'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_3390'/>
                <XMI.CorbaTcEnumLabel xmi.tcName='imsdt_2400'/>
            </XMI.CorbaTcEnum>
        </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
    <Model.Namespace.contents>
    </Model.Namespace.contents>
</Model.DataType>

```

```

        <XMI.CorbaTypeEnumLabel xmi.tcName='imsdt_3400'/>
        <XMI.CorbaTypeEnumLabel xmi.tcName='imsdt_TAPE'/>
    </XMI.CorbaTypeEnum>
</XMI.CorbaTypeCode>
</Model.DataType.typeCode>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2249'>
    <Model.ModelElement.name>ModelType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.DataType.typeCode>
        <XMI.CorbaTypeCode>
            <XMI.CorbaTypeEnum xmi.tcName='ModelType' xmi.tclId='null'>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsdt_1'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsdt_2'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsdt_11'/>
            </XMI.CorbaTypeEnum>
        </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
    <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2250'>
    <Model.ModelElement.name>RECFMType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.DataType.typeCode>
        <XMI.CorbaTypeCode>
            <XMI.CorbaTypeEnum xmi.tcName='RECFMType' xmi.tclId='null'>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsrf_F'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsrf_FB'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsrf_V'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsrf_VB'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsrf_U'/>
            </XMI.CorbaTypeEnum>
        </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
    <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2251'>
    <Model.ModelElement.name>AlgorithmType</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='false'/>
    <Model.GeneralizableElement.isLeaf xmi.value='false'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.DataType.typeCode>
        <XMI.CorbaTypeCode>
            <XMI.CorbaTypeEnum xmi.tcName='AlgorithmType' xmi.tclId='null'>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsat_0'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsat_1'/>
                <XMI.CorbaTypeEnumLabel xmi.tcName='imsat_2'/>
            </XMI.CorbaTypeEnum>
        </XMI.CorbaTypeCode>
    </Model.DataType.typeCode>
    <Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2252'>

```

```

<Model.ModelElement.name>LPointerType</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='false'/>
<Model.GeneralizableElement.isLeaf xmi.value='false'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.DataType.typeCode>
  <XMI.CorbaTypeCode>
    <XMI.CorbaTcEnum xmi.tcName='LPointerType' xmi.tcId='null'>
      <XMI.CorbaTcEnumLabel xmi.tcName='imslp_LTWIN'/>
      <XMI.CorbaTcEnumLabel xmi.tcName='imslp_LTWINBWD'/>
    </XMI.CorbaTcEnum>
  </XMI.CorbaTypeCode>
</Model.DataType.typeCode>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2253'>
  <Model.ModelElement.name>MSDBType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='MSDBType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsmt_NO'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsmt_TERM'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsmt_FIXED'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='imsmt_DYNAMIC'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
<Model.DataType xmi.id='a2254'>
  <Model.ModelElement.name>ParentType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.DataType.typeCode>
    <XMI.CorbaTypeCode>
      <XMI.CorbaTcEnum xmi.tcName='ParentType' xmi.tcId='null'>
        <XMI.CorbaTcEnumLabel xmi.tcName='ims_VIRTUAL'/>
        <XMI.CorbaTcEnumLabel xmi.tcName='ims_PHYSICAL'/>
      </XMI.CorbaTcEnum>
    </XMI.CorbaTypeCode>
  </Model.DataType.typeCode>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.DataType>
</Model.Namespace.contents>
</Model.Package>
<Model.Class xmi.id='a2255'>
  <Model.ModelElement.name>DBD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1099'> <!-- Record.RecordFile -->
  </Model.GeneralizableElement.supertypes>

```

```

<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2256'>
    <Model.ModelElement.name>dliAccess</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a2239'> <!-- IMSDatabase.IMSTypes.AccessMethodType -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2257'>
    <Model.ModelElement.name>isVSAM</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2258'>
    <Model.ModelElement.name>passwordFlag</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2259'>
    <Model.ModelElement.name>versionString</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>

```

```

</Model.Attribute>
<Model.Reference xmi.id='a2260'>
  <Model.ModelElement.name>dataset</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2261' /> <!-- IMSDatabase.Dataset -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2262' /> <!-- IMSDatabase.ContainsDataset.dataset -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2263'>
  <Model.ModelElement.name>segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>true</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2264' /> <!-- IMSDatabase.Segment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2265' /> <!-- IMSDatabase.ContainsSegment.segment -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2266'>
  <Model.ModelElement.name>exit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2267' /> <!-- IMSDatabase.Exit -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2268' /> <!-- IMSDatabase.Captures.exit -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2269'>
  <Model.ModelElement.name>accessMethod</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2271'/> <!--
      IMSDatabase.ExtendedByAccessMethod.accessMethod -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2272'>
  <Model.ModelElement.name>acbib</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2273'/> <!-- IMSDatabase.ACBLIB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2274'/> <!-- IMSDatabase.ContainsDBD.acbib -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2275'>
  <Model.ModelElement.name>library</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2276'/> <!-- IMSDatabase.DBDLib -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2277'/> <!-- IMSDatabase.IsInDBDLib.library -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2278'>
  <Model.ModelElement.name>pcb</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2279'/> <!-- IMSDatabase.PCB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2280'/> <!-- IMSDatabase.PcbToDbd.pcb -->
  </Model.Reference.referencedEnd>
</Model.Reference>

```

```

</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2281'>
  <Model.ModelElement.name>PSB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false' />
  <Model.GeneralizableElement.isLeaf xmi.value='false' />
  <Model.GeneralizableElement.isAbstract xmi.value='false' />
  <Model.GeneralizableElement.visibility xmi.value='public_vis' />
  <Model.Class.isSingleton xmi.value='false' />
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1099' /> <!-- Record.RecordFile -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2282'>
  <Model.ModelElement.name>compatibility</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27' /> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2283'>
  <Model.ModelElement.name>ioErrorOption</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18' /> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2284'>
  <Model.ModelElement.name>ioaSize</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.Attribute.isDerived xmi.value='false' />
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18' /> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2285'>
  <Model.ModelElement.name>language</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a2240'/> <!-- IMSDatabase.IMSTypes.PSBLanguageType -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2286'>
  <Model.ModelElement.name>lockMaximum</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2287'>
  <Model.ModelElement.name>maximumQxCalls</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2288'>
  <Model.ModelElement.name>onlineImageCopy</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2289'>
  <Model.ModelElement.name>ssaSize</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```



```

<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2290'>
  <Model.ModelElement.name>writeToOperator</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2291'>
  <Model.ModelElement.name>pcb</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2279'/> <!-- IMSDatabase.PCB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2292'/> <!-- IMSDatabase.PsbToPcb.pcb -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2293'>
  <Model.ModelElement.name>library</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2294'/> <!-- IMSDatabase.PSBLib -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2295'/> <!-- IMSDatabase.IsInPSBLib.library -->
  </Model.Reference.referencedEnd>

```

```

</Model.Reference>
<Model.Reference xmi.id='a2296'>
  <Model.ModelElement.name>acbib</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2273'> <!-- IMSDatabase.ACBLIB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2297'> <!-- IMSDatabase.ContainsPSB.acbib -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2279'>
  <Model.ModelElement.name>PCB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1099'> <!-- Record.RecordFile -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2298'>
  <Model.ModelElement.name>pcbType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2241'> <!-- IMSDatabase.IMSTypes.PCBType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2299'>
  <Model.ModelElement.name>list</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.Attribute.isDerived xmi.value='false'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>

```

```

<Model.Attribute xmi.id='a2300'>
  <Model.ModelElement.name>keyLength</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/'> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2301'>
  <Model.ModelElement.name>processingOptions</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/'> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2302'>
  <Model.ModelElement.name>positioning</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2242'/'> <!-- IMSDatabase.IMSTypes.PositioningType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2303'>
  <Model.ModelElement.name>sequentialBuffering</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>

```

```

<Model.Attribute xmi.id='a2304'>
  <Model.ModelElement.name>alternateResponse</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2305'>
  <Model.ModelElement.name>express</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2306'>
  <Model.ModelElement.name>modify</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2307'>
  <Model.ModelElement.name>sameTerminal</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/'> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>

```

```

<Model.Attribute xmi.id='a2308'>
  <Model.ModelElement.name>destinationType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2243'/'> <!-- IMSDatabase.IMSTypes.LTermType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2309'>
  <Model.ModelElement.name>ltermName</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.Attribute.isDerived xmi.value='false'/'>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/'> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2310'>
  <Model.ModelElement.name>procSeq</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2311'/'> <!-- IMSDatabase.INDEX -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2312'/'> <!-- IMSDatabase.SequencedBy.procSeq -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2313'>
  <Model.ModelElement.name>senSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/'>
  <Model.Feature.visibility xmi.value='public_vis'/'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2314'/'> <!-- IMSDatabase.SenSegment -->
  </Model.TypedElement.type>

```

```

<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2315' /> <!--
    IMSDatabase.PcbToSenSegment.senSegment -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2316'>
  <Model.ModelElement.name>psb</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2281' /> <!-- IMSDatabase.PSB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2317' /> <!-- IMSDatabase.PsbToPcb.psb -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2318'>
  <Model.ModelElement.name>dbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level' />
  <Model.Feature.visibility xmi.value='public_vis' />
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true' />
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2255' /> <!-- IMSDatabase.DBD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2319' /> <!-- IMSDatabase.PcbToDbd.dbd -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2264'>
  <Model.ModelElement.name>Segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false' />
  <Model.GeneralizableElement.isLeaf xmi.value='false' />
  <Model.GeneralizableElement.isAbstract xmi.value='false' />
  <Model.GeneralizableElement.visibility xmi.value='public_vis' />
  <Model.Class.isSingleton xmi.value='false' />
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1093' /> <!-- Record.RecordDef -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2320'>
      <Model.ModelElement.name>exitFlag</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level' />
      <Model.Feature.visibility xmi.value='public_vis' />
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2321'>
  <Model.ModelElement.name>frequency</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2322'>
  <Model.ModelElement.name>maxLength</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2323'>
  <Model.ModelElement.name>minLength</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2324'>
  <Model.ModelElement.name>rules</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```

```

<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a2244'/> <!-- IMSDatabase.IMSTypes.RulesType -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2325'>
  <Model.ModelElement.name>subsetPointers</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2326'>
  <Model.ModelElement.name>directDependent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2327'>
  <Model.ModelElement.name>pcPointer</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2245'/> <!-- IMSDatabase.IMSTypes.ChildPointerType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2328'>
  <Model.ModelElement.name>dbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```



```

<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2329'/> <!-- IMSDatabase.ContainsSegment.dbd -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2330'>
  <Model.ModelElement.name>exit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2267'/> <!-- IMSDatabase.Exit -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2331'/> <!-- IMSDatabase.CapturesExit.exit -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2332'>
  <Model.ModelElement.name>child</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2333'/> <!-- IMSDatabase.ParentChild.child -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2334'>
  <Model.ModelElement.name>parent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2335'/> <!-- IMSDatabase.ParentChild.parent -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2336'>
  <Model.ModelElement.name>logical</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>-1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>true</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2337'/> <!-- IMSDatabase.SegmentLogical -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2338'/> <!-- IMSDatabase.HasSource.logical -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2339'>
  <Model.ModelElement.name>senseg</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2340'/> <!-- IMSDatabase.SensegMapsTo.senseg -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2341'>
  <Model.ModelElement.name>SegmentComplex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2342'>
      <Model.ModelElement.name>deleteFlag</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a2246'/> <!-- IMSDatabase.IMSTypes.FlagsType -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a2343'>
      <Model.ModelElement.name>insertFlag</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a2246'/> <!-- IMSDatabase.IMSTypes.FlagsType -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2344'>
  <Model.ModelElement.name>replaceFlag</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2246'/> <!-- IMSDatabase.IMSTypes.FlagsType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2345'>
  <Model.ModelElement.name>segmPointer</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2247'/> <!-- IMSDatabase.IMSTypes.PointerType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2346'>
  <Model.ModelElement.name>dsGroup</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2347'>
  <Model.ModelElement.name>lpParent</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2348'/> <!-- IMSDatabase.LCHILD -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2349'/> <!-- IMSDatabase.IsLParent.lparent -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2350'>
  <Model.ModelElement.name>lchild</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2348'/> <!-- IMSDatabase.LCHILD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2351'/> <!-- IMSDatabase.IsLChild.lchild -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2352'>
  <Model.ModelElement.name>pairedLCHILD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2348'/> <!-- IMSDatabase.LCHILD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2353'/> <!-- IMSDatabase.IsPaired.pairedLCHILD -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2354'>
  <Model.ModelElement.name>secondaryIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>32</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>

```

```

    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
</Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2356'/> <!--
    IMSDatabase.IsIndexedBy.secondaryIndex -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2357'>
  <Model.ModelElement.name>sourcedIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
</Model.TypedElement.type>
  <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
</Model.TypedElement.type>
</Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2358'/> <!--
    IMSDatabase.HasIndexSource.sourcedIndex -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2359'>
  <Model.ModelElement.name>dataset</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
</Model.TypedElement.type>
  <Model.Class xmi.idref='a2261'/> <!-- IMSDatabase.Dataset -->
</Model.TypedElement.type>
</Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2360'/> <!-- IMSDatabase.StoresSegment.dataset -->
</Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2337'>
  <Model.ModelElement.name>SegmentLogical</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
  <Model.Attribute xmi.id='a2361'>
    <Model.ModelElement.name>keyData1</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
    </Model.StructuralFeature.multiplicity>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>

```

```

    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2362'>
  <Model.ModelElement.name>keyData2</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2363'>
  <Model.ModelElement.name>physical</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>2</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2364'/> <!-- IMSDatabase.HasSource.physical -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2365'>
  <Model.ModelElement.name>Field</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1105'/> <!-- Record.FixedOffsetField -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2366'>
      <Model.ModelElement.name>sequenceField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2367'>
  <Model.ModelElement.name>uniqueSequence</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2368'>
  <Model.ModelElement.name>length</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2369'>
  <Model.ModelElement.name>generated</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2370'>
  <Model.ModelElement.name>ddataIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2371'/> <!-- IMSDatabase.IsDuplicateData.ddataIndex
    -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2372'>
  <Model.ModelElement.name>searchIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2373'/> <!-- IMSDatabase.Searched.searchIndex -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2374'>
  <Model.ModelElement.name>subseqIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2375'/> <!--
    IMSDatabase.Subsequenced.subseqIndex -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2376'>
  <Model.ModelElement.name>senField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2377'/> <!-- IMSDatabase.SenField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2378'/> <!-- IMSDatabase.SenfldToField.senField -->
  </Model.Reference.referencedEnd>

```



```

</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2261'>
  <Model.ModelElement.name>Dataset</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2379'>
  <Model.ModelElement.name>dd1name</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2380'>
  <Model.ModelElement.name>device</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2248'/> <!-- IMSDatabase.IMSTypes.DeviceType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2381'>
  <Model.ModelElement.name>model</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2249'/> <!-- IMSDatabase.IMSTypes.ModelType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2382'>
  <Model.ModelElement.name>dd2name</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2383'>
  <Model.ModelElement.name>size1</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2384'>
  <Model.ModelElement.name>size2</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2385'>
  <Model.ModelElement.name>recordLength1</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2386'>
  <Model.ModelElement.name>recordLength2</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2387'>
  <Model.ModelElement.name>blockingFactor1</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2388'>
  <Model.ModelElement.name>blockingFactor2</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2389'>
  <Model.ModelElement.name>datasetLabel</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2390'>
  <Model.ModelElement.name>freeBlockFrequency</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2391'>
  <Model.ModelElement.name>freeSpacePercentage</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2392'>
  <Model.ModelElement.name>recordFormat</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2250'/> <!-- IMSDatabase.IMSTypes.RECFMTType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2393'>
  <Model.ModelElement.name>scanCylinders</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2394'>
  <Model.ModelElement.name>searchAlgorithm</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a2251'/> <!-- IMSDatabase.IMSTypes.AlgorithmType -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2395'>
  <Model.ModelElement.name>root</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2396'>
  <Model.ModelElement.name>rootOverflow</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2397'>
  <Model.ModelElement.name>uow</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2398'>
  <Model.ModelElement.name>uowOverflow</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2399'>
  <Model.ModelElement.name>dbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2400'/> <!-- IMSDatabase.ContainsDataset.dbd -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2401'>
  <Model.ModelElement.name>segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2402'/> <!-- IMSDatabase.StoresSegment.segment -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2314'>
  <Model.ModelElement.name>SenSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1093'/> <!-- Record.RecordDef -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2403'>

```

```

<Model.ModelElement.name>procoptSENSEG</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2404'>
  <Model.ModelElement.name>subsetPointers</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2405'>
  <Model.ModelElement.name>senField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2377'/> <!-- IMSDatabase.SenField -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2406'/> <!-- IMSDatabase.SensegToSenfld.senField -
    -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2407'>
  <Model.ModelElement.name>segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.TypedElement.type>

```

```

<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2408'/'> <!-- IMSDatabase.SensegMapsTo.segment -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2409'>
<Model.ModelElement.name>index</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/'>
<Model.Feature.visibility xmi.value='public_vis'/'>
<Model.StructuralFeature.multiplicity>
<XML.field>0</XML.field> <!-- lower -->
<XML.field>32</XML.field> <!-- upper -->
<XML.field>>false</XML.field> <!-- is_ordered -->
<XML.field>>true</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/'>
<Model.TypedElement.type>
<Model.Class xmi.idref='a2311'/'> <!-- IMSDatabase.INDEX -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2410'/'> <!-- IMSDatabase.Indices.index -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2411'>
<Model.ModelElement.name>pcb</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/'>
<Model.Feature.visibility xmi.value='public_vis'/'>
<Model.StructuralFeature.multiplicity>
<XML.field>1</XML.field> <!-- lower -->
<XML.field>1</XML.field> <!-- upper -->
<XML.field>>false</XML.field> <!-- is_ordered -->
<XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/'>
<Model.TypedElement.type>
<Model.Class xmi.idref='a2279'/'> <!-- IMSDatabase.PCB -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
<Model.AssociationEnd xmi.idref='a2412'/'> <!-- IMSDatabase.PcbToSenSegment.pcb -->
</Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2377'>
<Model.ModelElement.name>SenField</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='false'/'>
<Model.GeneralizableElement.isLeaf xmi.value='false'/'>
<Model.GeneralizableElement.isAbstract xmi.value='false'/'>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/'>
<Model.Class.isSingleton xmi.value='false'/'>
<Model.GeneralizableElement.supertypes>
<Model.Class xmi.idref='a1105'/'> <!-- Record.FixedOffsetField -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
<Model.Attribute xmi.id='a2413'>
<Model.ModelElement.name>replace</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/'>
<Model.Feature.visibility xmi.value='public_vis'/'>
<Model.StructuralFeature.multiplicity>
<XML.field>0</XML.field> <!-- lower -->
<XML.field>1</XML.field> <!-- upper -->
<XML.field>>false</XML.field> <!-- is_ordered -->
<XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/'>

```



```

<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2414'>
  <Model.ModelElement.name>field</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2415'/> <!-- IMSDatabase.SenfldToField.field -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2416'>
  <Model.ModelElement.name>senSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2417'/> <!--
      IMSDatabase.SensegToSenfld.senSegment -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2273'>
  <Model.ModelElement.name>ACBLIB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a707'/> <!-- Model_Management.Package -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Reference xmi.id='a2418'>
  <Model.ModelElement.name>dbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```

```

</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2419'/> <!-- IMSDatabase.ContainsDBD.dbd -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2420'>
  <Model.ModelElement.name>psb</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2281'/> <!-- IMSDatabase.PSB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2421'/> <!-- IMSDatabase.ContainsPSB.psb -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2270'>
  <Model.ModelElement.name>AccessMethod</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a2422'>
      <Model.ModelElement.name>dbd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a2423'/> <!--
          IMSDatabase.ExtendedByAccessMethod.dbd -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2311'>
  <Model.ModelElement.name>INDEX</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>

```

```

<Model.GeneralizableElement.isLeaf xmi.value='false'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2424'>
    <Model.ModelElement.name>dosCompatibility</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2425'>
    <Model.ModelElement.name>protect</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a2426'>
    <Model.ModelElement.name>primaryTarget</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2427'/> <!-- IMSDatabase.HIDAM -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2428'/> <!-- IMSDatabase.PrimaryIndex.primaryTarget -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
  <Model.Reference xmi.id='a2429'>
    <Model.ModelElement.name>secondaryTarget</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>

```

```

    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2430'/> <!-- IMSDatabase.Indexes.secondaryTarget --
    >
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2431'>
  <Model.ModelElement.name>sharedIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2432'/> <!-- IMSDatabase.IndexShares.sharedIndex -
    ->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2433'>
  <Model.ModelElement.name>sharingIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>16</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2434'/> <!-- IMSDatabase.IndexShares.sharingIndex -
    ->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2435'>
  <Model.ModelElement.name>senseg</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->

```

```

</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2436'/> <!-- IMSDatabase.Indices.senseg -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2437'>
  <Model.ModelElement.name>sequencedPCB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XMI.field>0</XMI.field> <!-- lower -->
    <XMI.field>-1</XMI.field> <!-- upper -->
    <XMI.field>>false</XMI.field> <!-- is_ordered -->
    <XMI.field>>true</XMI.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2279'/> <!-- IMSDatabase.PCB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2438'/> <!--
      IMSDatabase.SequencedBy.sequencedPCB -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2427'>
  <Model.ModelElement.name>HIDAM</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a2439'>
      <Model.ModelElement.name>index</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XMI.field>0</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>false</XMI.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a2440'/> <!-- IMSDatabase.PrimaryIndex.index -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2441'>
  <Model.ModelElement.name>DEDB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>

```

```

<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2442'>
    <Model.ModelElement.name>rmName</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2443'>
    <Model.ModelElement.name>stage</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a18'/> <!-- Foundation.Data_Types.Integer -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2444'>
    <Model.ModelElement.name>extendedCall</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
    </Model.TypedElement.type>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2445'>
  <Model.ModelElement.name>HDAM</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
  </Model.GeneralizableElement.supertypes>

```

```

<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2446'>
    <Model.ModelElement.name>rmName</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2447'>
    <Model.ModelElement.name>relativeBlockNumber</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a18'> <!-- Foundation.Data_Types.Integer -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2448'>
    <Model.ModelElement.name>rootAnchorPoints</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a18'> <!-- Foundation.Data_Types.Integer -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Attribute xmi.id='a2449'>
    <Model.ModelElement.name>rootMaxBytes</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'>
    <Model.Feature.visibility xmi.value='public_vis'>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'>
    <Model.Attribute.isDerived xmi.value='false'>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a18'> <!-- Foundation.Data_Types.Integer -->
    </Model.TypedElement.type>

```

```

    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2450'>
  <Model.ModelElement.name>MSDB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2451'>
      <Model.ModelElement.name>msdbField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a2452'>
      <Model.ModelElement.name>msdbType</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a2253'/> <!-- IMSDatabase.IMSTypes.MSDBtype -->
      </Model.TypedElement.type>
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2355'>
  <Model.ModelElement.name>SecondaryIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2453'>
      <Model.ModelElement.name>constant</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>

```



```

<Model.StructuralFeature.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2454'>
  <Model.ModelElement.name>exitRoutine</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2455'>
  <Model.ModelElement.name>nullValue</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2456'>
  <Model.ModelElement.name>index</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2457'/> <!-- IMSDatabase.Indexes.index -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2458'>
  <Model.ModelElement.name>segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2459'/> <!-- IMSDatabase.IsIndexedBy.segment -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2460'>
  <Model.ModelElement.name>indexSource</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2461'/> <!--
      IMSDatabase.HasIndexSource.indexSource -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2462'>
  <Model.ModelElement.name>ddataFields</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>5</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2463'/> <!-- IMSDatabase.IsDuplicateData.ddataFields
      -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2464'>
  <Model.ModelElement.name>searchFields</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>5</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>

```

```

<Model.TypedElement.type>
  <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2465'/> <!-- IMSDatabase.Searched.searchFields -->
</Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2466'>
  <Model.ModelElement.name>subseqFields</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>5</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
</Model.TypedElement.type>
<Model.Reference.referencedEnd>
  <Model.AssociationEnd xmi.idref='a2467'/> <!--
    IMSDatabase.Subsequenced.subseqFields -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2267'>
  <Model.ModelElement.name>Exit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2468'>
    <Model.ModelElement.name>key</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
  <Model.Attribute xmi.id='a2469'>
    <Model.ModelElement.name>data</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>

```

```

</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2470'>
  <Model.ModelElement.name>path</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2471'>
  <Model.ModelElement.name>log</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2472'>
  <Model.ModelElement.name>cascade</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2473'>
  <Model.ModelElement.name>cascadeKey</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>

```

```

</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2474'>
  <Model.ModelElement.name>cascadeData</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2475'>
  <Model.ModelElement.name>cascadePath</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2476'>
  <Model.ModelElement.name>dbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2477'/> <!-- IMSDatabase.Captures.dbd -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2478'>
  <Model.ModelElement.name>segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
  </Model.StructuralFeature.multiplicity>

```

```

    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2479'/> <!-- IMSDatabase.CapturesExit.segment -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2348'>
  <Model.ModelElement.name>LCHILD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
</Model.Namespace.contents>
<Model.Attribute xmi.id='a2480'>
  <Model.ModelElement.name>counter</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2481'>
  <Model.ModelElement.name>lcPointer</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2245'/> <!-- IMSDatabase.IMSTypes.ChildPointerType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2482'>
  <Model.ModelElement.name>lpParentFlag</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2483'>
  <Model.ModelElement.name>ltwin</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2252'/> <!-- IMSDatabase.IMSTypes.LPointerType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2484'>
  <Model.ModelElement.name>rules</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2244'/> <!-- IMSDatabase.IMSTypes.RulesType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2485'>
  <Model.ModelElement.name>virtualParent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a2254'/> <!-- IMSDatabase.IMSTypes.ParentType -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2486'>
  <Model.ModelElement.name>lparent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->

```

```

    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2487'/> <!-- IMSDatabase.IsLChild.lparent -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2488'>
  <Model.ModelElement.name>lchild</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2489'/> <!-- IMSDatabase.IsLParent.lchild -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2490'>
  <Model.ModelElement.name>pairedSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2491'/> <!-- IMSDatabase.IsPaired.pairedSegment -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2294'>
  <Model.ModelElement.name>PSBLib</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a707'/> <!-- Model_Management.Package -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a2492'>
      <Model.ModelElement.name>psb</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>

```



```

    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2281'/> <!-- IMSDatabase.PSB -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2493'/> <!-- IMSDatabase.IsInPSBLib.psb -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2276'>
  <Model.ModelElement.name>DBDLib</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a707'/> <!-- Model_Management.Package -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a2494'>
      <Model.ModelElement.name>dbd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a2495'/> <!-- IMSDatabase.IsInDBDLib.dbd -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>
<Model.Association xmi.id='a2496'>
  <Model.ModelElement.name>ExtendedByAccessMethod</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2423'>
      <Model.ModelElement.name>dbd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
</Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2271'>
  <Model.ModelElement.name>accessMethod</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2270'/> <!-- IMSDatabase.AccessMethod -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2497'>
  <Model.ModelElement.name>PrimaryIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
</Model.Association>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2428'>
    <Model.ModelElement.name>primaryTarget</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2427'/> <!-- IMSDatabase.HIDAM -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2440'>
    <Model.ModelElement.name>index</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2498'>
  <Model.ModelElement.name>Indexes</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='true'/>
<Model.GeneralizableElement.isLeaf xmi.value='true'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2430'>
    <Model.ModelElement.name>secondaryTarget</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2457'>
    <Model.ModelElement.name>index</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2499'>
  <Model.ModelElement.name>IsIndexedBy</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2356'>
      <Model.ModelElement.name>secondaryIndex</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>32</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2459'>
      <Model.ModelElement.name>segment</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.AssociationEnd.isNavigable xmi.value='true'/>
<Model.AssociationEnd.aggregation xmi.value='composite'/>
<Model.AssociationEnd.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
</Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2500'>
  <Model.ModelElement.name>IndexShares</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2434'>
      <Model.ModelElement.name>sharingIndex</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>16</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2432'>
      <Model.ModelElement.name>sharedIndex</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2501'>
  <Model.ModelElement.name>Searched</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>

```

```

<Model.AssociationEnd xmi.id='a2465'>
  <Model.ModelElement.name>searchFields</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>5</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2373'>
  <Model.ModelElement.name>searchIndex</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2502'>
  <Model.ModelElement.name>IsDuplicateData</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2463'>
      <Model.ModelElement.name>ddataFields</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>5</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2371'>
      <Model.ModelElement.name>ddataIndex</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
      </Model.AssociationEnd.multiplicity>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2503'>
  <Model.ModelElement.name>Subsequenced</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2375'>
      <Model.ModelElement.name>subseqIndex</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2467'>
      <Model.ModelElement.name>subseqFields</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>5</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2504'>
  <Model.ModelElement.name>IsLChild</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2351'>
      <Model.ModelElement.name>lchild</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->

```

```

    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2348'/> <!-- IMSDatabase.LCHILD -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2487'>
  <Model.ModelElement.name>lparent</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='composite'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2505'>
  <Model.ModelElement.name>HasSource</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2338'>
      <Model.ModelElement.name>logical</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2337'/> <!-- IMSDatabase.SegmentLogical -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2364'>
      <Model.ModelElement.name>physical</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>2</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2506'>
  <Model.ModelElement.name>SequencedBy</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2312'>
      <Model.ModelElement.name>procSeq</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2438'>
      <Model.ModelElement.name>sequencedPCB</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2279'/> <!-- IMSDatabase.PCB -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2507'>
  <Model.ModelElement.name>Indices</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2410'>
      <Model.ModelElement.name>index</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>32</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2311'/> <!-- IMSDatabase.INDEX -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```



```

    </Model.TypedElement.type>
  </Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2436'>
  <Model.ModelElement.name>senseg</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2508'>
  <Model.ModelElement.name>ContainsPSB</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2297'>
      <Model.ModelElement.name>acblib</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2273'/> <!-- IMSDatabase.ACBLIB -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2421'>
      <Model.ModelElement.name>psb</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2281'/> <!-- IMSDatabase.PSB -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2509'>
  <Model.ModelElement.name>ContainsDBD</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>

```

```

<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2274'>
    <Model.ModelElement.name>acblib</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2273'/> <!-- IMSDatabase.ACBLIB -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2419'>
    <Model.ModelElement.name>dbd</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2510'>
  <Model.ModelElement.name>HasIndexSource</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2461'>
      <Model.ModelElement.name>indexSource</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2358'>
    <Model.ModelElement.name>sourcedIndex</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>

```

```

<Model.AssociationEnd.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>-1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>true</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2355'/> <!-- IMSDatabase.SecondaryIndex -->
</Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2511'>
  <Model.ModelElement.name>ContainsDataset</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2400'>
      <Model.ModelElement.name>dbd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2262'>
      <Model.ModelElement.name>dataset</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2261'/> <!-- IMSDatabase.Dataset -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2512'>
  <Model.ModelElement.name>ContainsSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2265'>
      <Model.ModelElement.name>segment</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.AssociationEnd.isNavigable xmi.value='true'/>
<Model.AssociationEnd.aggregation xmi.value='none'/>
<Model.AssociationEnd.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>-1</XML.field> <!-- upper -->
  <XML.field>true</XML.field> <!-- is_ordered -->
  <XML.field>true</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
</Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2329'>
  <Model.ModelElement.name>dbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='composite'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2513'>
  <Model.ModelElement.name>PcbToDbd</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2319'>
      <Model.ModelElement.name>dbd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2280'>
      <Model.ModelElement.name>pcb</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    <Model.TypedElement.type>
    <Model.Class xmi.idref='a2279'> <!-- IMSDatabase.PCB -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2514'>
  <Model.ModelElement.name>PsbToPcb</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'>
  <Model.GeneralizableElement.isLeaf xmi.value='true'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'>
  <Model.Association.isDerived xmi.value='false'>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2292'>
      <Model.ModelElement.name>pcb</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'>
      <Model.AssociationEnd.aggregation xmi.value='none'>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2279'> <!-- IMSDatabase.PCB -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2317'>
      <Model.ModelElement.name>psb</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'>
      <Model.AssociationEnd.aggregation xmi.value='none'>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2281'> <!-- IMSDatabase.PSB -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2515'>
  <Model.ModelElement.name>PcbToSenSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'>
  <Model.GeneralizableElement.isLeaf xmi.value='true'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'>
  <Model.Association.isDerived xmi.value='false'>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2315'>
      <Model.ModelElement.name>senSegment</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'>
      <Model.AssociationEnd.aggregation xmi.value='none'>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->

```

```

</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->
</Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2412'>
  <Model.ModelElement.name>pcb</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='composite'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2279'/> <!-- IMSDatabase.PCB -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2516'>
  <Model.ModelElement.name>SensegToSenfld</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2417'>
      <Model.ModelElement.name>senSegment</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2406'>
      <Model.ModelElement.name>senField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2377'/> <!-- IMSDatabase.SenField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2517'>

```

```

<Model.ModelElement.name>SensegMapsTo</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.GeneralizableElement.isRoot xmi.value='true'/>
<Model.GeneralizableElement.isLeaf xmi.value='true'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
<Model.Association.isDerived xmi.value='false'/>
<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2408'>
    <Model.ModelElement.name>segment</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2340'>
    <Model.ModelElement.name>senseg</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'/>
    <Model.AssociationEnd.aggregation xmi.value='none'/>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2314'/> <!-- IMSDatabase.SenSegment -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2518'>
  <Model.ModelElement.name>ParentChild</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2333'>
      <Model.ModelElement.name>child</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.AssociationEnd xmi.id='a2335'>

```

```

<Model.ModelElement.name>parent</Model.ModelElement.name>
<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.AssociationEnd.isNavigable xmi.value='true'/>
<Model.AssociationEnd.aggregation xmi.value='none'/>
<Model.AssociationEnd.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
</Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2519'>
  <Model.ModelElement.name>SenFldToField</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2415'>
      <Model.ModelElement.name>field</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2378'>
      <Model.ModelElement.name>senField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2377'/> <!-- IMSDatabase.SenField -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2520'>
  <Model.ModelElement.name>Captures</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>

```



```

<Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2268'>
    <Model.ModelElement.name>exit</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'>
    <Model.AssociationEnd.aggregation xmi.value='none'>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2267'> <!-- IMSDatabase.Exit -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
  <Model.AssociationEnd xmi.id='a2477'>
    <Model.ModelElement.name>dbd</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.AssociationEnd.isNavigable xmi.value='true'>
    <Model.AssociationEnd.aggregation xmi.value='composite'>
    <Model.AssociationEnd.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a2255'> <!-- IMSDatabase.DBD -->
    </Model.TypedElement.type>
  </Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2521'>
  <Model.ModelElement.name>CapturesExit</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'>
  <Model.GeneralizableElement.isLeaf xmi.value='true'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'>
  <Model.Association.isDerived xmi.value='false'>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2331'>
      <Model.ModelElement.name>exit</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'>
      <Model.AssociationEnd.aggregation xmi.value='none'>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2267'> <!-- IMSDatabase.Exit -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2479'>
      <Model.ModelElement.name>segment</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'>
      <Model.AssociationEnd.aggregation xmi.value='composite'>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->

```

```

        <XML.field>false</XML.field> <!-- is_ordered -->
        <XML.field>false</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
        <Model.Class xmi.idref='a2264'/> <!-- IMSDatabase.Segment -->
    </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2522'>
    <Model.ModelElement.name>IsLParent</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='true'/>
    <Model.GeneralizableElement.isLeaf xmi.value='true'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
    <Model.Association.isDerived xmi.value='false'/>
    <Model.Namespace.contents>
        <Model.AssociationEnd xmi.id='a2349'>
            <Model.ModelElement.name>lpParent</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>
                <XML.field>0</XML.field> <!-- lower -->
                <XML.field>1</XML.field> <!-- upper -->
                <XML.field>false</XML.field> <!-- is_ordered -->
                <XML.field>false</XML.field> <!-- is_unique -->
            </Model.AssociationEnd.multiplicity>
            <Model.AssociationEnd.isChangeable xmi.value='true'/>
            <Model.TypedElement.type>
                <Model.Class xmi.idref='a2348'/> <!-- IMSDatabase.LCHILD -->
            </Model.TypedElement.type>
        </Model.AssociationEnd>
        <Model.AssociationEnd xmi.id='a2489'>
            <Model.ModelElement.name>lchild</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>
                <XML.field>1</XML.field> <!-- lower -->
                <XML.field>1</XML.field> <!-- upper -->
                <XML.field>false</XML.field> <!-- is_ordered -->
                <XML.field>false</XML.field> <!-- is_unique -->
            </Model.AssociationEnd.multiplicity>
            <Model.AssociationEnd.isChangeable xmi.value='true'/>
            <Model.TypedElement.type>
                <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
            </Model.TypedElement.type>
        </Model.AssociationEnd>
    </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2523'>
    <Model.ModelElement.name>IsPaired</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='true'/>
    <Model.GeneralizableElement.isLeaf xmi.value='true'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
    <Model.Association.isDerived xmi.value='false'/>
    <Model.Namespace.contents>
        <Model.AssociationEnd xmi.id='a2353'>
            <Model.ModelElement.name>pairedLCHILD</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>

```

```

    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2348'/> <!-- IMSDatabase.LCHILD -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2491'>
  <Model.ModelElement.name>pairedSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341'/> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2524'>
  <Model.ModelElement.name>IsInDBDLib</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2495'>
      <Model.ModelElement.name>dbd</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2255'/> <!-- IMSDatabase.DBD -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2277'>
      <Model.ModelElement.name>library</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2276'/> <!-- IMSDatabase.DBDLib -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2525'>
  <Model.ModelElement.name>IsInPSBLib</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2295'>
      <Model.ModelElement.name>library</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2294'/> <!-- IMSDatabase.PSBLib -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2493'>
      <Model.ModelElement.name>psb</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2281'/> <!-- IMSDatabase.PSB -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2526'>
  <Model.ModelElement.name>StoresSegment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='protected_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2360'>
      <Model.ModelElement.name>dataset</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>

```

```

    <Model.Class xmi.idref='a2261' /> <!-- IMSDatabase.Dataset -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2402'>
  <Model.ModelElement.name>segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true' />
  <Model.AssociationEnd.aggregation xmi.value='none' />
  <Model.AssociationEnd.multiplicity>
    <XMI.field>0</XMI.field> <!-- lower -->
    <XMI.field>-1</XMI.field> <!-- upper -->
    <XMI.field>>false</XMI.field> <!-- is_ordered -->
    <XMI.field>>true</XMI.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true' />
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2341' /> <!-- IMSDatabase.SegmentComplex -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a2527'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis' />
  <Model.Import.isClustered xmi.value='false' />
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0' /> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2528'>
  <Model.ModelElement.name>Record</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis' />
  <Model.Import.isClustered xmi.value='false' />
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a1088' /> <!-- Record -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2529'>
  <Model.ModelElement.name>Model_Management</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis' />
  <Model.Import.isClustered xmi.value='false' />
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a706' /> <!-- Model_Management -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Tag xmi.id='a2530'>
  <Model.ModelElement.name>org.omg.xmi.namespace=CWMXIMS</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
  <Model.Tag.values>
    <XMI.any xmi.type='string'>CWMXIMS</XMI.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a2237' /> <!-- IMSDatabase -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a2531'>
  <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.resource</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XMI.any xmi.type='string'>org.omg.cwm.x.resource</XMI.any>
  </Model.Tag.values>

```

```

    <Model.Tag.elements>
      <Model.Package xmi.idref='a2237'/> <!-- IMSDatabase -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2532'>
    <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.resource</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.java.cwm.x.resource</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2237'/> <!-- IMSDatabase -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2533'>
    <Model.ModelElement.name>org.omg.mof.idl_substitute_name=ImsField</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>ImsField</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2534'>
    <Model.ModelElement.name>org.omg.mof.java_substitute_name=ImsField</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_substitute_name</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>ImsField</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Class xmi.idref='a2365'/> <!-- IMSDatabase.Field -->
    </Model.Tag.elements>
  </Model.Tag>
</Model.Namespace.contents>
</Model.Package>
<Model.Package xmi.id='a2535'>
  <Model.ModelElement.name>InformationSet</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a2536'>
      <Model.ModelElement.name>InformationSet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1387'/> <!-- Olap.Schema -->
      </Model.GeneralizableElement.supertypes>
    </Model.Namespace.contents>
    <Model.Attribute xmi.id='a2537'>
      <Model.ModelElement.name>version</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>

```

```

    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2538'>
  <Model.ModelElement.name>rule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2539'/> <!-- InformationSet.Rule -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2540'/> <!--
      InformationSet.InformationSetReferencesRule.rule -->
  </Model.Reference.referencedEnd>
</Model.Reference>
<Model.Reference xmi.id='a2541'>
  <Model.ModelElement.name>infoSetAdmin</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2542'/> <!-- InformationSet.InfoSetAdministration -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2543'/> <!--
      InformationSet.InformationSetReferencesInfoSetAdministration.infoSetA
      dmin -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2544'>
  <Model.ModelElement.name>Segment</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1324'/> <!-- Olap.Cube -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2545'>
      <Model.ModelElement.name>regionSequence</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.Feature.scope xmi.value='instance_level'/>
<Model.Feature.visibility xmi.value='public_vis'/>
<Model.StructuralFeature.multiplicity>
  <XML.field>1</XML.field> <!-- lower -->
  <XML.field>1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>false</XML.field> <!-- is_unique -->
</Model.StructuralFeature.multiplicity>
<Model.StructuralFeature.isChangeable xmi.value='true'/>
<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a25'/> <!-- Foundation.Data_Types.ProcedureExpression -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Reference xmi.id='a2546'>
  <Model.ModelElement.name>rule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2539'/> <!-- InformationSet.Rule -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a2547'/> <!--
      InformationSet.SegmentReferencesRule.rule -->
  </Model.Reference.referencedEnd>
</Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2548'>
  <Model.ModelElement.name>SegmentRegion</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1330'/> <!-- Olap.CubeRegion -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Reference xmi.id='a2549'>
      <Model.ModelElement.name>rule</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2539'/> <!-- InformationSet.Rule -->
      </Model.TypedElement.type>
      <Model.Reference.referencedEnd>
        <Model.AssociationEnd xmi.idref='a2550'/> <!--
          InformationSet.SegmentRegionReferencesRule.rule -->
      </Model.Reference.referencedEnd>
    </Model.Reference>
  </Model.Namespace.contents>
</Model.Class>

```



```

</Model.Reference>
<Model.Reference xmi.id='a2551'>
  <Model.ModelElement.name>renderedObject</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'>
  <Model.Feature.visibility xmi.value='public_vis'>
  <Model.StructuralFeature.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>true</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a1613'> <!-- InformationVisualization.RenderedObject -->
  </Model.TypedElement.type>
  <Model.Reference.referencedEnd>
    <Model.AssociationEnd xmi.idref='a1638'> <!--
      InformationVisualization.RenderedObjectsReferenceModelElement.renderedObject -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2539'>
  <Model.ModelElement.name>Rule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'>
  <Model.GeneralizableElement.isLeaf xmi.value='false'>
  <Model.GeneralizableElement.isAbstract xmi.value='false'>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'>
  <Model.Class.isSingleton xmi.value='false'>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2552'>
      <Model.ModelElement.name>ruleExpression</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'>
      <Model.Feature.visibility xmi.value='public_vis'>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'>
      <Model.Attribute.isDerived xmi.value='false'>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a821'> <!-- Expressions.ExpressionNode -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a2553'>
      <Model.ModelElement.name>type</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'>
      <Model.Feature.visibility xmi.value='public_vis'>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'>
      <Model.Attribute.isDerived xmi.value='false'>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'> <!-- Foundation.Data_Types.String -->

```

```

    </Model.TypedElement.type>
  </Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2542'>
  <Model.ModelElement.name>InfoSetAdministration</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53'/> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2554'>
      <Model.ModelElement.name>priority</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Attribute xmi.id='a2555'>
      <Model.ModelElement.name>acknowledgement</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.Attribute.isDerived xmi.value='false'/>
      <Model.TypedElement.type>
        <Model.DataType xmi.idref='a27'/> <!-- Foundation.Data_Types.Boolean -->
      </Model.TypedElement.type>
    </Model.Attribute>
    <Model.Reference xmi.id='a2556'>
      <Model.ModelElement.name>date</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level'/>
      <Model.Feature.visibility xmi.value='public_vis'/>
      <Model.StructuralFeature.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2557'/> <!-- InformationSet.InfoSetDate -->
      </Model.TypedElement.type>
    </Model.Reference>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2558'/> <!--
        InformationSet.InfoSetAdministrationReferencesInfoSetDates.date -->
    </Model.Reference.referencedEnd>
  </Model.Namespace.contents>
</Model.Class>

```

```

    </Model.Reference.referencedEnd>
  </Model.Reference>
  <Model.Reference xmi.id='a2559'>
    <Model.ModelElement.name>responsibleParty</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level' />
    <Model.Feature.visibility xmi.value='public_vis' />
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true' />
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a851' /> <!-- BusinessInformation.ResponsibleParty -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a909' /> <!--
        BusinessInformation.ModelElementResponsibility.responsibleParty -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
  <Model.Reference xmi.id='a2560'>
    <Model.ModelElement.name>document</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level' />
    <Model.Feature.visibility xmi.value='public_vis' />
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true' />
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a893' /> <!-- BusinessInformation.Document -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a905' /> <!--
        BusinessInformation.DocumentDescribes.document -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2557'>
  <Model.ModelElement.name>InfoSetDate</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false' />
  <Model.GeneralizableElement.isLeaf xmi.value='false' />
  <Model.GeneralizableElement.isAbstract xmi.value='false' />
  <Model.GeneralizableElement.visibility xmi.value='public_vis' />
  <Model.Class.isSingleton xmi.value='false' />
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a53' /> <!-- Foundation.Core.ModelElement -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
    <Model.Attribute xmi.id='a2561'>
      <Model.ModelElement.name>type</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.Feature.scope xmi.value='instance_level' />
      <Model.Feature.visibility xmi.value='public_vis' />
      <Model.StructuralFeature.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.StructuralFeature.multiplicity>
      <Model.StructuralFeature.isChangeable xmi.value='true' />
    </Model.Attribute>
  </Model.Namespace.contents>
</Model.Class>

```

```

<Model.Attribute.isDerived xmi.value='false'/>
<Model.TypedElement.type>
  <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
</Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2562'>
  <Model.ModelElement.name>format</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
<Model.Attribute xmi.id='a2563'>
  <Model.ModelElement.name>dateTime</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Feature.scope xmi.value='instance_level'/>
  <Model.Feature.visibility xmi.value='public_vis'/>
  <Model.StructuralFeature.multiplicity>
    <XML.field>1</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>false</XML.field> <!-- is_ordered -->
    <XML.field>false</XML.field> <!-- is_unique -->
  </Model.StructuralFeature.multiplicity>
  <Model.StructuralFeature.isChangeable xmi.value='true'/>
  <Model.Attribute.isDerived xmi.value='false'/>
  <Model.TypedElement.type>
    <Model.DataType xmi.idref='a10'/> <!-- Foundation.Data_Types.String -->
  </Model.TypedElement.type>
</Model.Attribute>
</Model.Namespace.contents>
</Model.Class>
<Model.Association xmi.id='a2564'>
  <Model.ModelElement.name>InformationSetReferencesRule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2565'>
      <Model.ModelElement.name>informationSet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>false</XML.field> <!-- is_ordered -->
        <XML.field>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2536'/> <!-- InformationSet.InformationSet -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
  <Model.AssociationEnd xmi.id='a2540'>
    <Model.ModelElement.name>rule</Model.ModelElement.name>

```

```

<Model.ModelElement.annotation></Model.ModelElement.annotation>
<Model.AssociationEnd.isNavigable xmi.value='true'/>
<Model.AssociationEnd.aggregation xmi.value='none'/>
<Model.AssociationEnd.multiplicity>
  <XML.field>0</XML.field> <!-- lower -->
  <XML.field>-1</XML.field> <!-- upper -->
  <XML.field>>false</XML.field> <!-- is_ordered -->
  <XML.field>>true</XML.field> <!-- is_unique -->
</Model.AssociationEnd.multiplicity>
<Model.AssociationEnd.isChangeable xmi.value='true'/>
<Model.TypedElement.type>
  <Model.Class xmi.idref='a2539'/> <!-- InformationSet.Rule -->
</Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2566'>
  <Model.ModelElement.name>SegmentReferencesRule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2567'>
      <Model.ModelElement.name>segment</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2544'/> <!-- InformationSet.Segment -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2547'>
      <Model.ModelElement.name>rule</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>true</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2539'/> <!-- InformationSet.Rule -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2568'>
  <Model.ModelElement.name>SegmentRegionReferencesRule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>

```

```

<Model.AssociationEnd xmi.id='a2569'>
  <Model.ModelElement.name>segmentRegion</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>false</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2548'/> <!-- InformationSet.SegmentRegion -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
<Model.AssociationEnd xmi.id='a2550'>
  <Model.ModelElement.name>rule</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.AssociationEnd.isNavigable xmi.value='true'/>
  <Model.AssociationEnd.aggregation xmi.value='none'/>
  <Model.AssociationEnd.multiplicity>
    <XML.field>0</XML.field> <!-- lower -->
    <XML.field>-1</XML.field> <!-- upper -->
    <XML.field>>false</XML.field> <!-- is_ordered -->
    <XML.field>>true</XML.field> <!-- is_unique -->
  </Model.AssociationEnd.multiplicity>
  <Model.AssociationEnd.isChangeable xmi.value='true'/>
  <Model.TypedElement.type>
    <Model.Class xmi.idref='a2539'/> <!-- InformationSet.Rule -->
  </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2570'>
  <Model.ModelElement.name>InformationSetReferencesInfoSetAdministration</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2571'>
      <Model.ModelElement.name>informationSet</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='composite'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>1</XML.field> <!-- lower -->
        <XML.field>1</XML.field> <!-- upper -->
        <XML.field>>false</XML.field> <!-- is_ordered -->
        <XML.field>>false</XML.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2536'/> <!-- InformationSet.InformationSet -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2543'>
      <Model.ModelElement.name>infoSetAdmin</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XML.field>0</XML.field> <!-- lower -->
        <XML.field>-1</XML.field> <!-- upper -->
      </Model.AssociationEnd.multiplicity>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>

```

```

        <XML.field>false</XML.field> <!-- is_ordered -->
        <XML.field>true</XML.field> <!-- is_unique -->
    </Model.AssociationEnd.multiplicity>
    <Model.AssociationEnd.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
        <Model.Class xmi.idref='a2542'/> <!-- InformationSet.InfoSetAdministration -->
    </Model.TypedElement.type>
</Model.AssociationEnd>
</Model.Namespace.contents>
</Model.Association>
<Model.Association xmi.id='a2572'>
    <Model.ModelElement.name>InfoSetAdministrationReferencesInfoSetDates</
        Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value='true'/>
    <Model.GeneralizableElement.isLeaf xmi.value='true'/>
    <Model.GeneralizableElement.isAbstract xmi.value='false'/>
    <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
    <Model.Association.isDerived xmi.value='false'/>
    <Model.Namespace.contents>
        <Model.AssociationEnd xmi.id='a2558'>
            <Model.ModelElement.name>date</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>
                <XML.field>0</XML.field> <!-- lower -->
                <XML.field>-1</XML.field> <!-- upper -->
                <XML.field>false</XML.field> <!-- is_ordered -->
                <XML.field>true</XML.field> <!-- is_unique -->
            </Model.AssociationEnd.multiplicity>
            <Model.AssociationEnd.isChangeable xmi.value='true'/>
            <Model.TypedElement.type>
                <Model.Class xmi.idref='a2557'/> <!-- InformationSet.InfoSetDate -->
            </Model.TypedElement.type>
        </Model.AssociationEnd>
        <Model.AssociationEnd xmi.id='a2573'>
            <Model.ModelElement.name>infoSetAdmin</Model.ModelElement.name>
            <Model.ModelElement.annotation></Model.ModelElement.annotation>
            <Model.AssociationEnd.isNavigable xmi.value='true'/>
            <Model.AssociationEnd.aggregation xmi.value='none'/>
            <Model.AssociationEnd.multiplicity>
                <XML.field>1</XML.field> <!-- lower -->
                <XML.field>-1</XML.field> <!-- upper -->
                <XML.field>false</XML.field> <!-- is_ordered -->
                <XML.field>true</XML.field> <!-- is_unique -->
            </Model.AssociationEnd.multiplicity>
            <Model.AssociationEnd.isChangeable xmi.value='true'/>
            <Model.TypedElement.type>
                <Model.Class xmi.idref='a2542'/> <!-- InformationSet.InfoSetAdministration -->
            </Model.TypedElement.type>
        </Model.AssociationEnd>
    </Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a2574'>
    <Model.ModelElement.name>Foundation</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
        <Model.Package xmi.idref='a0'/> <!-- Foundation -->
    </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2575'>
    <Model.ModelElement.name>InformationVisualization</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>

```

```

    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a1612'/> <!-- InformationVisualization -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2576'>
    <Model.ModelElement.name>Expressions</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a820'/> <!-- Expressions -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2577'>
    <Model.ModelElement.name>BusinessInformation</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a850'/> <!-- BusinessInformation -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Import xmi.id='a2578'>
    <Model.ModelElement.name>Olap</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Import.visibility xmi.value='public_vis'/>
    <Model.Import.isClustered xmi.value='false'/>
    <Model.Import.importedNamespace>
      <Model.Package xmi.idref='a1323'/> <!-- Olap -->
    </Model.Import.importedNamespace>
  </Model.Import>
  <Model.Tag xmi.id='a2579'>
    <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.analysis</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.cwm.x.analysis</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2535'/> <!-- InformationSet -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2580'>
    <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.analysis</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>org.omg.java.cwm.x.analysis</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2535'/> <!-- InformationSet -->
    </Model.Tag.elements>
  </Model.Tag>
  <Model.Tag xmi.id='a2581'>
    <Model.ModelElement.name>org.omg.xmi.namespace=CWMXIS</
      Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
    <Model.Tag.values>
      <XML.any xmi.type='string'>CWMXIS</XML.any>
    </Model.Tag.values>
    <Model.Tag.elements>
      <Model.Package xmi.idref='a2535'/> <!-- InformationSet -->
    </Model.Tag.elements>
  </Model.Tag>
</Model.Namespace.contents>

```



```

</Model.Package>
<Model.Package xmi.id='a2582'>
  <Model.ModelElement.name>InformationReporting</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Namespace.contents>
    <Model.Class xmi.id='a2583'>
      <Model.ModelElement.name>Report</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1613'/> <!-- InformationVisualization.RenderedObject -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
        </Model.Namespace.contents>
      </Model.Class>
    <Model.Class xmi.id='a2584'>
      <Model.ModelElement.name>ReportAttribute</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a128'/> <!-- Foundation.Core.Attribute -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
        </Model.Namespace.contents>
      </Model.Class>
    <Model.Class xmi.id='a2585'>
      <Model.ModelElement.name>ReportExecution</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1252'/> <!-- Transformation.TransformationMap -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
        </Model.Namespace.contents>
      </Model.Class>
    <Model.Class xmi.id='a2586'>
      <Model.ModelElement.name>ReportField</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.GeneralizableElement.isRoot xmi.value='false'/>
      <Model.GeneralizableElement.isLeaf xmi.value='false'/>
      <Model.GeneralizableElement.isAbstract xmi.value='false'/>
      <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
      <Model.Class.isSingleton xmi.value='false'/>
      <Model.GeneralizableElement.supertypes>
        <Model.Class xmi.idref='a1613'/> <!-- InformationVisualization.RenderedObject -->
      </Model.GeneralizableElement.supertypes>
      <Model.Namespace.contents>
        </Model.Namespace.contents>
      </Model.Class>
    <Model.Class xmi.id='a2587'>
      <Model.ModelElement.name>ReportGroup</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>

```

```

<Model.GeneralizableElement.isRoot xmi.value='false'/>
<Model.GeneralizableElement.isLeaf xmi.value='false'/>
<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.Class.isSingleton xmi.value='false'/>
<Model.GeneralizableElement.supertypes>
  <Model.Class xmi.idref='a1613'/> <!-- InformationVisualization.RenderedObject -->
</Model.GeneralizableElement.supertypes>
<Model.Namespace.contents>
  <Model.Attribute xmi.id='a2588'>
    <Model.ModelElement.name>type</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>1</XML.field> <!-- lower -->
      <XML.field>1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>false</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.Attribute.isDerived xmi.value='false'/>
    <Model.TypedElement.type>
      <Model.DataType xmi.idref='a2589'/> <!-- InformationReporting.ReportGroupType -->
    </Model.TypedElement.type>
  </Model.Attribute>
  <Model.Reference xmi.id='a2590'>
    <Model.ModelElement.name>reportQuery</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.Feature.scope xmi.value='instance_level'/>
    <Model.Feature.visibility xmi.value='public_vis'/>
    <Model.StructuralFeature.multiplicity>
      <XML.field>0</XML.field> <!-- lower -->
      <XML.field>-1</XML.field> <!-- upper -->
      <XML.field>>false</XML.field> <!-- is_ordered -->
      <XML.field>>true</XML.field> <!-- is_unique -->
    </Model.StructuralFeature.multiplicity>
    <Model.StructuralFeature.isChangeable xmi.value='true'/>
    <Model.TypedElement.type>
      <Model.Class xmi.idref='a729'/> <!-- DataTypes.QueryExpression -->
    </Model.TypedElement.type>
    <Model.Reference.referencedEnd>
      <Model.AssociationEnd xmi.idref='a2591'/> <!--
        InformationReporting.ReportGroupReferenceQueryExpressions.reportQ
        uery -->
    </Model.Reference.referencedEnd>
  </Model.Reference>
</Model.Namespace.contents>
</Model.Class>
<Model.Class xmi.id='a2592'>
  <Model.ModelElement.name>ReportPackage</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Class.isSingleton xmi.value='false'/>
  <Model.GeneralizableElement.supertypes>
    <Model.Class xmi.idref='a1628'/> <!-- InformationVisualization.RenderedObjectSet -->
  </Model.GeneralizableElement.supertypes>
  <Model.Namespace.contents>
  </Model.Namespace.contents>
</Model.Class>
<Model.DataType xmi.id='a2589'>
  <Model.ModelElement.name>ReportGroupType</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='false'/>
  <Model.GeneralizableElement.isLeaf xmi.value='false'/>

```

```

<Model.GeneralizableElement.isAbstract xmi.value='false'/>
<Model.GeneralizableElement.visibility xmi.value='public_vis'/>
<Model.DataType.typeCode>
  <XMI.CorbaTypeCode>
    <XMI.CorbaTcEnum xmi.tcName='ReportGroupType' xmi.tcId='null'>
      <XMI.CorbaTcEnumLabel xmi.tcName='header'/>
      <XMI.CorbaTcEnumLabel xmi.tcName='footer'/>
      <XMI.CorbaTcEnumLabel xmi.tcName='detail'/>
      <XMI.CorbaTcEnumLabel xmi.tcName='other'/>
    </XMI.CorbaTcEnum>
  </XMI.CorbaTypeCode>
</Model.DataType.typeCode>
<Model.Namespace.contents>
</Model.Namespace.contents>
</Model.DataType>
<Model.Association xmi.id='a2593'>
  <Model.ModelElement.name>ReportGroupReferenceQueryExpressions</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.GeneralizableElement.isRoot xmi.value='true'/>
  <Model.GeneralizableElement.isLeaf xmi.value='true'/>
  <Model.GeneralizableElement.isAbstract xmi.value='false'/>
  <Model.GeneralizableElement.visibility xmi.value='public_vis'/>
  <Model.Association.isDerived xmi.value='false'/>
  <Model.Namespace.contents>
    <Model.AssociationEnd xmi.id='a2591'>
      <Model.ModelElement.name>reportQuery</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XMI.field>0</XMI.field> <!-- lower -->
        <XMI.field>-1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>true</XMI.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a729'/> <!-- DataTypes.QueryExpression -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
    <Model.AssociationEnd xmi.id='a2594'>
      <Model.ModelElement.name>reportGroup</Model.ModelElement.name>
      <Model.ModelElement.annotation></Model.ModelElement.annotation>
      <Model.AssociationEnd.isNavigable xmi.value='true'/>
      <Model.AssociationEnd.aggregation xmi.value='none'/>
      <Model.AssociationEnd.multiplicity>
        <XMI.field>0</XMI.field> <!-- lower -->
        <XMI.field>1</XMI.field> <!-- upper -->
        <XMI.field>>false</XMI.field> <!-- is_ordered -->
        <XMI.field>>false</XMI.field> <!-- is_unique -->
      </Model.AssociationEnd.multiplicity>
      <Model.AssociationEnd.isChangeable xmi.value='true'/>
      <Model.TypedElement.type>
        <Model.Class xmi.idref='a2587'/> <!-- InformationReporting.ReportGroup -->
      </Model.TypedElement.type>
    </Model.AssociationEnd>
  </Model.Namespace.contents>
</Model.Association>
<Model.Import xmi.id='a2595'>
  <Model.ModelElement.name>DataTypes</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a728'/> <!-- DataTypes -->
  </Model.Import.importedNamespace>
</Model.Import>

```

```

<Model.Import xmi.id='a2596'>
  <Model.ModelElement.name>InformationVisualization</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a1612'/> <!-- InformationVisualization -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2597'>
  <Model.ModelElement.name>Foundation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a0'/> <!-- Foundation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Import xmi.id='a2598'>
  <Model.ModelElement.name>Transformation</Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Import.visibility xmi.value='public_vis'/>
  <Model.Import.isClustered xmi.value='false'/>
  <Model.Import.importedNamespace>
    <Model.Package xmi.idref='a1213'/> <!-- Transformation -->
  </Model.Import.importedNamespace>
</Model.Import>
<Model.Tag xmi.id='a2599'>
  <Model.ModelElement.name>org.omg.mof.idl_prefix=org.omg.cwm.x.analysis</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.idl_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>org.omg.cwm.x.analysis</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a2582'/> <!-- InformationReporting -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a2600'>
  <Model.ModelElement.name>org.omg.xmi.namespace=CWMXIR</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.xmi.namespace</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>CWMXIR</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a2582'/> <!-- InformationReporting -->
  </Model.Tag.elements>
</Model.Tag>
<Model.Tag xmi.id='a2601'>
  <Model.ModelElement.name>org.omg.mof.java_prefix=org.omg.java.cwm.x.analysis</
    Model.ModelElement.name>
  <Model.ModelElement.annotation></Model.ModelElement.annotation>
  <Model.Tag.tagId>org.omg.mof.java_prefix</Model.Tag.tagId>
  <Model.Tag.values>
    <XML.any xmi.type='string'>org.omg.java.cwm.x.analysis</XML.any>
  </Model.Tag.values>
  <Model.Tag.elements>
    <Model.Package xmi.idref='a2582'/> <!-- InformationReporting -->
  </Model.Tag.elements>
</Model.Tag>
</Model.Namespace.contents>
</Model.Package>

```

2 CWM Interfaces as CORBA IDL

This chapter contains CORBA IDL modules for the CWMX packages. The IDL modules are generated from the CWMX metamodels defined in chapter 1 based on the IDL generation rules of the MOF 1.3 Specification.

The CWMX IDL modules use (by #include) IDL modules for CWM metamodels defined in volume 2 and for UML metamodels defined in the UML 1.3 Specification.

2.1 *ER.idl*

```
#ifndef ER_IDL
#define ER_IDL
#pragma prefix "org.omg.cwm.x.foundation"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Expressions.idl"
#include "ModelManagement.idl"
#include "KeysIndexes.idl"

module ER
{
    interface EntityClass;
    interface Entity;
    typedef sequence<Entity> EntitySet;
    interface NonuniqueKeyClass;
    interface NonuniqueKey;
    typedef sequence<NonuniqueKey> NonuniqueKeySet;
    interface CandidateKeyClass;
    interface CandidateKey;
    typedef sequence<CandidateKey> CandidateKeySet;
    interface ForeignKeyClass;
    interface ForeignKey;
    typedef sequence<ForeignKey> ForeignKeySet;
    interface DomainClass;
    interface Domain;
    typedef sequence<Domain> DomainSet;
    interface ErAttributeClass;
    interface ErAttribute;
    typedef sequence<ErAttribute> ErAttributeSet;
    interface ErRelationshipClass;
    interface ErRelationship;
    typedef sequence<ErRelationship> ErRelationshipSet;
    interface RelationshipEndClass;
    interface RelationshipEnd;
    typedef sequence<RelationshipEnd> RelationshipEndSet;
    interface ModelLibraryClass;
    interface ModelLibrary;
    typedef sequence<ModelLibrary> ModelLibrarySet;
    interface ErModelClass;
    interface ErModel;
    typedef sequence<ErModel> ErModelSet;
    interface SubjectAreaClass;
    interface SubjectArea;
    typedef sequence<SubjectArea> SubjectAreaSet;
    interface PrimaryKeyClass;
    interface PrimaryKey;
    typedef sequence<PrimaryKey> PrimaryKeySet;
    interface ERPackage;

    interface EntityClass : Foundation::Core::ClassClass
    {
        readonly attribute EntitySet all_of_type_entity;
        readonly attribute EntitySet all_of_class_entity;
        Entity create_entity (
            in Foundation::DataTypes::Name name,
            in Foundation::DataTypes::VisibilityKind visibility,
```

```

        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active)
        raises (Reflective::MofError);
};

interface Entity : EntityClass, Foundation::Core::Class
{
}; // end of interface Entity

interface NonuniqueKeyClass : KeysIndexes::IndexClass
{
    readonly attribute NonuniqueKeySet all_of_type_nonunique_key;
    readonly attribute NonuniqueKeySet all_of_class_nonunique_key;
    NonuniqueKey create_nonunique_key (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,
        in boolean is_partitioning)
        raises (Reflective::MofError);
};

interface NonuniqueKey : NonuniqueKeyClass, KeysIndexes::Index
{
}; // end of interface NonuniqueKey

interface CandidateKeyClass : KeysIndexes::UniqueKeyClass
{
    readonly attribute CandidateKeySet all_of_type_candidate_key;
    readonly attribute CandidateKeySet all_of_class_candidate_key;
    CandidateKey create_candidate_key (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);
};

interface CandidateKey : CandidateKeyClass, KeysIndexes::UniqueKey
{
}; // end of interface CandidateKey

interface ForeignKeyClass : KeysIndexes::KeyRelationshipClass
{
    readonly attribute ForeignKeySet all_of_type_foreign_key;
    readonly attribute ForeignKeySet all_of_class_foreign_key;
    ForeignKey create_foreign_key (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);
};

interface ForeignKey : ForeignKeyClass, KeysIndexes::KeyRelationship
{
    RelationshipEnd relationship_end ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_relationship_end (in RelationshipEnd new_value)
};

```

```

        raises (Reflective::MofError);
    void unset_relationship_end ()
        raises (Reflective::MofError);
}; // end of interface ForeignKey

interface DomainClass : Foundation::Core::ClassifierClass
{
    readonly attribute DomainSet all_of_type_domain;
    readonly attribute DomainSet all_of_class_domain;
    Domain create_domain (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in Expressions::ExpressionNode validation_rule,
        in Expressions::ExpressionNode default_value)
        raises (Reflective::MofError);
};

interface Domain : DomainClass, Foundation::Core::Classifier
{
    Expressions::ExpressionNode validation_rule ()
        raises (Reflective::MofError);
    void set_validation_rule (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    Expressions::ExpressionNode default_value ()
        raises (Reflective::MofError);
    void set_default_value (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    Foundation::Core::Classifier base_type ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_base_type (in Foundation::Core::Classifier new_value)
        raises (Reflective::MofError);
    void unset_base_type ()
        raises (Reflective::MofError);
}; // end of interface Domain

interface ErAttributeClass : Foundation::Core::UmlAttributeClass
{
    readonly attribute ErAttributeSet all_of_type_er_attribute;
    readonly attribute ErAttributeSet all_of_class_er_attribute;
    ErAttribute create_er_attribute (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface ErAttribute : ErAttributeClass, Foundation::Core::UmlAttribute
{
}; // end of interface ErAttribute

interface ErRelationshipClass : Foundation::Core::UmlAssociationClassClass
{

```



```

readonly attribute ErRelationshipSet all_of_type_er_relationship;
readonly attribute ErRelationshipSet all_of_class_er_relationship;
ErRelationship create_er_relationship (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract,
    in boolean is_active)
    raises (Reflective::MofError);
};

interface ErRelationship : ErRelationshipClass, Foundation::Core::UmlAssociationClass
{
}; // end of interface ErRelationship

interface RelationshipEndClass : Foundation::Core::AssociationEndClass
{
    readonly attribute RelationshipEndSet all_of_type_relationship_end;
    readonly attribute RelationshipEndSet all_of_class_relationship_end;
    RelationshipEnd create_relationship_end (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_navigable,
        in Foundation::DataTypes::OrderingKind ordering,
        in Foundation::DataTypes::AggregationKind aggregation,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Expressions::ExpressionNode delete,
        in Expressions::ExpressionNode update,
        in Expressions::ExpressionNode insert)
        raises (Reflective::MofError);
};

interface RelationshipEnd : RelationshipEndClass, Foundation::Core::AssociationEnd
{
    Expressions::ExpressionNode delete ()
        raises (Reflective::MofError);
    void set_delete (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    Expressions::ExpressionNode update ()
        raises (Reflective::MofError);
    void set_update (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    Expressions::ExpressionNode insert ()
        raises (Reflective::MofError);
    void set_insert (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    ForeignKey foreign_key ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_foreign_key (in ForeignKey new_value)
        raises (Reflective::MofError);
    void unset_foreign_key ()
        raises (Reflective::MofError);
}; // end of interface RelationshipEnd

interface ModelLibraryClass : ModelManagement::SubsystemClass
{

```

```

readonly attribute ModelLibrarySet all_of_type_model_library;
readonly attribute ModelLibrarySet all_of_class_model_library;
ModelLibrary create_model_library (
  in Foundation::DataTypes::Name name,
  in Foundation::DataTypes::VisibilityKind visibility,
  in boolean is_specification,
  in boolean is_root,
  in boolean is_leaf,
  in boolean is_abstract,
  in boolean is_instantiable)
  raises (Reflective::MofError);
};

interface ModelLibrary : ModelLibraryClass, ModelManagement::Subsystem
{
}; // end of interface ModelLibrary

interface ErModelClass : ModelManagement::ModelClass
{
  readonly attribute ErModelSet all_of_type_er_model;
  readonly attribute ErModelSet all_of_class_er_model;
  ErModel create_er_model (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract)
    raises (Reflective::MofError);
};

interface ErModel : ErModelClass, ModelManagement::Model
{
}; // end of interface ErModel

interface SubjectAreaClass : ModelManagement::PackageClass
{
  readonly attribute SubjectAreaSet all_of_type_subject_area;
  readonly attribute SubjectAreaSet all_of_class_subject_area;
  SubjectArea create_subject_area (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract)
    raises (Reflective::MofError);
};

interface SubjectArea : SubjectAreaClass, ModelManagement::Package
{
}; // end of interface SubjectArea

interface PrimaryKeyClass : CandidateKeyClass
{
  readonly attribute PrimaryKeySet all_of_type_primary_key;
  readonly attribute PrimaryKeySet all_of_class_primary_key;
  PrimaryKey create_primary_key (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification)

```

```

        raises (Reflective::MofError);
    };

interface PrimaryKey : PrimaryKeyClass, CandidateKey
{
}; // end of interface PrimaryKey

struct DomainBaseTypeLink
{
    Foundation::Core::Classifier base_type;
    ER::Domain domain;
};
typedef sequence<DomainBaseTypeLink> DomainBaseTypeLinkSet;

interface DomainBaseType : Reflective::RefAssociation
{
    DomainBaseTypeLinkSet all_domain_base_type_links()
        raises (Reflective::MofError);
    boolean exists (
        in Foundation::Core::Classifier base_type,
        in ER::Domain domain)
        raises (Reflective::MofError);
    Foundation::Core::Classifier base_type (in ER::Domain domain)
        raises (Reflective::MofError);
    DomainSet domain (in Foundation::Core::Classifier base_type)
        raises (Reflective::MofError);
    void add (
        in Foundation::Core::Classifier base_type,
        in ER::Domain domain)
        raises (Reflective::MofError);
    void modify_base_type (
        in Foundation::Core::Classifier base_type,
        in ER::Domain domain,
        in Foundation::Core::Classifier new_base_type)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_domain (
        in Foundation::Core::Classifier base_type,
        in ER::Domain domain,
        in ER::Domain new_domain)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Foundation::Core::Classifier base_type,
        in ER::Domain domain)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface DomainBaseType

interface ERPackageFactory
{
    ERPackage create_er_package ()
        raises (Reflective::MofError);
};

interface ERPackage : Reflective::RefPackage
{
    readonly attribute EntityClass entity_ref;
    readonly attribute NonuniqueKeyClass nonunique_key_ref;
    readonly attribute CandidateKeyClass candidate_key_ref;
    readonly attribute ForeignKeyClass foreign_key_ref;
    readonly attribute DomainClass domain_ref;
    readonly attribute ErAttributeClass er_attribute_ref;
    readonly attribute ErRelationshipClass er_relationship_ref;
};

```

```
    readonly attribute RelationshipEndClass relationship_end_ref;  
    readonly attribute ModelLibraryClass model_library_ref;  
    readonly attribute ErModelClass er_model_ref;  
    readonly attribute SubjectAreaClass subject_area_ref;  
    readonly attribute PrimaryKeyClass primary_key_ref;  
    readonly attribute DomainBaseType domain_base_type_ref;  
};  
};  
  
#endif
```

2.2 CORBAIDLDataTypes.idl

```
#ifndef CORBAIDLDATA_TYPES_IDL
#define CORBAIDLDATA_TYPES_IDL
#pragma prefix "org.omg.cwm.x.resource"
#include "Reflective.idl"
#include "Foundation.idl"
#include "DataTypes.idl"

module CORBAIDLDataTypes
{
    interface IDLTypeClass;
    interface IDLType;
    typedef sequence<IDLType> IDLTypeSet;
    interface ArrayTypeClass;
    interface ArrayType;
    typedef sequence<ArrayType> ArrayTypeSet;
    interface ExceptionTypeClass;
    interface ExceptionType;
    typedef sequence<ExceptionType> ExceptionTypeSet;
    interface SequenceTypeClass;
    interface SequenceType;
    typedef sequence<SequenceType> SequenceTypeSet;
    interface WstringTypeClass;
    interface WstringType;
    typedef sequence<WstringType> WstringTypeSet;
    interface StringTypeClass;
    interface StringType;
    typedef sequence<StringType> StringTypeSet;
    interface FixedTypeClass;
    interface FixedType;
    typedef sequence<FixedType> FixedTypeSet;
    interface AliasClass;
    interface Alias;
    typedef sequence<Alias> AliasSet;
    interface StructTypeClass;
    interface StructType;
    typedef sequence<StructType> StructTypeSet;
    interface UnionTypeClass;
    interface UnionType;
    typedef sequence<UnionType> UnionTypeSet;
    interface EnumTypeClass;
    interface EnumType;
    typedef sequence<EnumType> EnumTypeSet;
    interface CORBAIDLDataTypesPackage;

    interface IDLTypeClass : Foundation::Core::DataTypeClass
    {
        readonly attribute IDLTypeSet all_of_type_idltype;
        readonly attribute IDLTypeSet all_of_class_idltype;
        IDLType create_idltype (
            in Foundation::DataTypes::Name name,
            in Foundation::DataTypes::VisibilityKind visibility,
            in boolean is_specification,
            in boolean is_root,
            in boolean is_leaf,
            in boolean is_abstract,
            in CORBA::TypeCode type_code)
    }
}
```

```

        raises (Reflective::MofError);
};

interface IDLType : IDLTypeClass, Foundation::Core::DataType
{
    CORBA::TypeCode type_code ()
        raises (Reflective::MofError);
    void set_type_code (in CORBA::TypeCode new_value)
        raises (Reflective::MofError);
}; // end of interface IDLType

interface ArrayTypeClass : IDLTypeClass
{
    readonly attribute ArrayTypeSet all_of_type_array_type;
    readonly attribute ArrayTypeSet all_of_class_array_type;
    ArrayType create_array_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code,
        in Foundation::DataTypes::Integer length)
        raises (Reflective::MofError);
};

interface ArrayType : ArrayTypeClass, IDLType
{
    Foundation::DataTypes::Integer length ()
        raises (Reflective::MofError);
    void set_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    IDLType element_type ()
        raises (Reflective::MofError);
    void set_element_type (in IDLType new_value)
        raises (Reflective::MofError);
}; // end of interface ArrayType

interface ExceptionTypeClass : IDLTypeClass
{
    readonly attribute ExceptionTypeSet all_of_type_exception_type;
    readonly attribute ExceptionTypeSet all_of_class_exception_type;
    ExceptionType create_exception_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code)
        raises (Reflective::MofError);
};

interface ExceptionType : ExceptionTypeClass, IDLType
{
}; // end of interface ExceptionType

interface SequenceTypeClass : IDLTypeClass
{
    readonly attribute SequenceTypeSet all_of_type_sequence_type;

```

```

readonly attribute SequenceTypeSet all_of_class_sequence_type;
SequenceType create_sequence_type (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract,
    in CORBA::TypeCode type_code,
    in Foundation::DataTypes::Integer length)
    raises (Reflective::MofError);
};

```

```

interface SequenceType : SequenceTypeClass, IDLType
{
    Foundation::DataTypes::Integer length ()
        raises (Reflective::MofError);
    void set_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    IDLType element_type ()
        raises (Reflective::MofError);
    void set_element_type (in IDLType new_value)
        raises (Reflective::MofError);
}; // end of interface SequenceType

```

```

interface WstringTypeClass : IDLTypeClass
{
    readonly attribute WstringTypeSet all_of_type_wstring_type;
    readonly attribute WstringTypeSet all_of_class_wstring_type;
    WstringType create_wstring_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code,
        in Foundation::DataTypes::Integer length)
        raises (Reflective::MofError);
};

```

```

interface WstringType : WstringTypeClass, IDLType
{
    Foundation::DataTypes::Integer length ()
        raises (Reflective::MofError);
    void set_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
}; // end of interface WstringType

```

```

interface StringTypeClass : IDLTypeClass
{
    readonly attribute StringTypeSet all_of_type_string_type;
    readonly attribute StringTypeSet all_of_class_string_type;
    StringType create_string_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code,

```

```

        in Foundation::DataTypes::Integer length)
        raises (Reflective::MofError);
};

interface StringType : StringTypeClass, IDLType
{
    Foundation::DataTypes::Integer length ()
        raises (Reflective::MofError);
    void set_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
}; // end of interface StringType

interface FixedTypeClass : IDLTypeClass
{
    readonly attribute FixedTypeSet all_of_type_fixed_type;
    readonly attribute FixedTypeSet all_of_class_fixed_type;
    FixedType create_fixed_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code,
        in Foundation::DataTypes::Integer digits,
        in Foundation::DataTypes::Integer scale)
        raises (Reflective::MofError);
};

interface FixedType : FixedTypeClass, IDLType
{
    Foundation::DataTypes::Integer digits ()
        raises (Reflective::MofError);
    void set_digits (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer scale ()
        raises (Reflective::MofError);
    void set_scale (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
}; // end of interface FixedType

interface AliasClass : IDLTypeClass, DataTypes::TypeAliasClass
{
    readonly attribute AliasSet all_of_type_alias;
    readonly attribute AliasSet all_of_class_alias;
    Alias create_alias (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code)
        raises (Reflective::MofError);
};

interface Alias : AliasClass, IDLType, DataTypes::TypeAlias
{
}; // end of interface Alias

interface StructTypeClass : IDLTypeClass

```



```

{
    readonly attribute StructTypeSet all_of_type_struct_type;
    readonly attribute StructTypeSet all_of_class_struct_type;
    StructType create_struct_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code)
        raises (Reflective::MofError);
};

interface StructType : StructTypeClass, IDLType
{
}; // end of interface StructType

interface UnionTypeClass : DataTypes::CwmUnionClass, IDLTypeClass
{
    readonly attribute UnionTypeSet all_of_type_union_type;
    readonly attribute UnionTypeSet all_of_class_union_type;
    UnionType create_union_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in CORBA::TypeCode type_code)
        raises (Reflective::MofError);
};

interface UnionType : UnionTypeClass, DataTypes::CwmUnion, IDLType
{
}; // end of interface UnionType

interface EnumTypeClass : DataTypes::EnumerationClass, IDLTypeClass
{
    readonly attribute EnumTypeSet all_of_type_enum_type;
    readonly attribute EnumTypeSet all_of_class_enum_type;
    EnumType create_enum_type (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_ordered,
        in CORBA::TypeCode type_code)
        raises (Reflective::MofError);
};

interface EnumType : EnumTypeClass, DataTypes::Enumeration, IDLType
{
}; // end of interface EnumType

interface CORBAIDLDataTypesPackageFactory
{
    CORBAIDLDataTypesPackage create_corbaidldata_types_package ()
        raises (Reflective::MofError);
};

```

```

};

interface CORBAIDLDataTypesPackage : Reflective::RefPackage
{
    readonly attribute IDLTypeClass idltype_ref;
    readonly attribute ArrayTypeClass array_type_ref;
    readonly attribute ExceptionTypeClass exception_type_ref;
    readonly attribute SequenceTypeClass sequence_type_ref;
    readonly attribute WstringTypeClass wstring_type_ref;
    readonly attribute StringTypeClass string_type_ref;
    readonly attribute FixedTypeClass fixed_type_ref;
    readonly attribute AliasClass alias_ref;
    readonly attribute StructTypeClass struct_type_ref;
    readonly attribute UnionTypeClass union_type_ref;
    readonly attribute EnumTypeClass enum_type_ref;
};
};

#endif

```

2.3 *COBOLData.idl*

```
#ifndef COBOLDATA_IDL
#define COBOLDATA_IDL
#pragma prefix "org.omg.cwmx.resource"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Record.idl"
#include "KeysIndexes.idl"

module COBOLData
{
    typedef sequence<string> StringBag;
    typedef sequence<Record::RecordDef> RecordDefUList;
    interface COBOLFieldClass;
    interface COBOLField;
    typedef sequence<COBOLField> COBOLFieldSet;
    interface UsageClass;
    interface Usage;
    typedef sequence<Usage> UsageSet;
    interface COBOLFDClass;
    interface COBOLFD;
    typedef sequence<COBOLFD> COBOLFDSet;
    typedef sequence<COBOLFD> COBOLFDUList;
    interface RenamesClass;
    interface Renames;
    typedef sequence<Renames> RenamesSet;
    interface SectionClass;
    interface Section;
    typedef sequence<Section> SectionSet;
    interface COBOLFDIndexClass;
    interface COBOLFDIndex;
    typedef sequence<COBOLFDIndex> COBOLFDIndexSet;
    interface LinageInfoClass;
    interface LinageInfo;
    typedef sequence<LinageInfo> LinageInfoSet;
    typedef sequence<LinageInfo> LinageInfoUList;
    interface ReportWriterSectionClass;
    interface ReportWriterSection;
    typedef sequence<ReportWriterSection> ReportWriterSectionSet;
    interface WorkingStorageSectionClass;
    interface WorkingStorageSection;
    typedef sequence<WorkingStorageSection> WorkingStorageSectionSet;
    interface FileSectionClass;
    interface FileSection;
    typedef sequence<FileSection> FileSectionSet;
    interface LinkageSectionClass;
    interface LinkageSection;
    typedef sequence<LinkageSection> LinkageSectionSet;
    interface OccursKeyClass;
    interface OccursKey;
    typedef sequence<OccursKey> OccursKeySet;
    typedef sequence<OccursKey> OccursKeyUList;
    interface COBOLDataPackage;
    enum SignKindType {sk_unsigned, sk_leadingSign, sk_trailingSign, sk_leadingSepSign,
                      sk_trailingSepSign};
    enum FileOrganization {fo_unspecified, fo_indexed, fo_relative, fo_sequential};
    enum AccessType {at_unspecified, at_dynamic, at_random, at_sequential};
}
```

```

enum BlockKind {bk_records, bk_characters};
enum LabelKind {lk_unspecified, lk_standard, lk_omitted};
enum LinageInfoType {li_linage, li_linageFooting, li_linageTop, li_linageBottom};

```

```

interface COBOLFieldClass : Record::FieldClass
{
    readonly attribute COBOLFieldSet all_of_type_cobolfield;
    readonly attribute COBOLFieldSet all_of_class_cobolfield;
    COBOLField create_cobolfield (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in Foundation::DataTypes::Integer level,
        in SignKindType sign_kind,
        in boolean is_filler,
        in boolean is_justified_right,
        in boolean is_blank_when_zero,
        in boolean is_synchronized,
        in string picture,
        in Foundation::DataTypes::Integer occurs_lower,
        in Foundation::DataTypes::Integer occurs_upper,
        in StringBag index_name,
        in boolean is_external,
        in boolean is_global)
        raises (Reflective::MofError);
};

```

```

interface COBOLField : COBOLFieldClass, Record::Field
{
    Foundation::DataTypes::Integer level ()
        raises (Reflective::MofError);
    void set_level (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    SignKindType sign_kind ()
        raises (Reflective::MofError);
    void set_sign_kind (in SignKindType new_value)
        raises (Reflective::MofError);
    boolean is_filler ()
        raises (Reflective::MofError);
    void set_is_filler (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_justified_right ()
        raises (Reflective::MofError);
    void set_is_justified_right (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_blank_when_zero ()
        raises (Reflective::MofError);
    void set_is_blank_when_zero (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_synchronized ()
        raises (Reflective::MofError);
    void set_is_synchronized (in boolean new_value)
        raises (Reflective::MofError);
    string picture ()
        raises (Reflective::MofError);
    void set_picture (in string new_value)

```

```

    raises (Reflective::MofError);
Foundation::DataTypes::Integer occurs_lower ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_occurs_lower (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_occurs_lower ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer occurs_upper ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_occurs_upper (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_occurs_upper ()
    raises (Reflective::MofError);
StringBag index_name ()
    raises (Reflective::MofError);
void set_index_name (in StringBag new_value)
    raises (Reflective::MofError);
void unset_index_name ()
    raises (Reflective::MofError);
void add_index_name (in string new_element)
    raises (Reflective::MofError);
void modify_index_name (
    in string old_element,
    in string new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_index_name (in string old_element)
    raises (Reflective::NotFound, Reflective::MofError);
boolean is_external ()
    raises (Reflective::MofError);
void set_is_external (in boolean new_value)
    raises (Reflective::MofError);
boolean is_global ()
    raises (Reflective::MofError);
void set_is_global (in boolean new_value)
    raises (Reflective::MofError);
COBOLField depending_on_field ()
    raises (Reflective::NotSet, Reflective::MofError);
void setDepending_on_field (in COBOLField new_value)
    raises (Reflective::MofError);
void unsetDepending_on_field ()
    raises (Reflective::MofError);
COBOLFieldSet occurring_field ()
    raises (Reflective::MofError);
void setOccurring_field (in COBOLFieldSet new_value)
    raises (Reflective::MofError);
void unsetOccurring_field ()
    raises (Reflective::MofError);
void addOccurring_field (in COBOLField new_element)
    raises (Reflective::MofError);
void modifyOccurring_field (
    in COBOLField old_element,
    in COBOLField new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void removeOccurring_field (in COBOLField old_element)
    raises (Reflective::NotFound, Reflective::MofError);
OccursKeyUList occurs_key_info ()
    raises (Reflective::MofError);
void setOccurs_key_info (in OccursKeyUList new_value)
    raises (Reflective::MofError);
void unsetOccurs_key_info ()
    raises (Reflective::MofError);

```

```

void add_occurs_key_info (in OccursKey new_element)
    raises (Reflective::MofError);
void add_occurs_key_info_before (
    in OccursKey new_element,
    in OccursKey before_element)
    raises (Reflective::NotFound, Reflective::MofError);
void modify_occurs_key_info (
    in OccursKey old_element,
    in OccursKey new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_occurs_key_info (in OccursKey old_element)
    raises (Reflective::NotFound, Reflective::MofError);
OccursKeySet occurs_key_field_info ()
    raises (Reflective::MofError);
void set_occurs_key_field_info (in OccursKeySet new_value)
    raises (Reflective::MofError);
void unset_occurs_key_field_info ()
    raises (Reflective::MofError);
void add_occurs_key_field_info (in OccursKey new_element)
    raises (Reflective::MofError);
void modify_occurs_key_field_info (
    in OccursKey old_element,
    in OccursKey new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_occurs_key_field_info (in OccursKey old_element)
    raises (Reflective::NotFound, Reflective::MofError);
COBOLFieldSet redefined_by_field ()
    raises (Reflective::MofError);
void set_redefined_by_field (in COBOLFieldSet new_value)
    raises (Reflective::MofError);
void unset_redefined_by_field ()
    raises (Reflective::MofError);
void add_redefined_by_field (in COBOLField new_element)
    raises (Reflective::MofError);
void modify_redefined_by_field (
    in COBOLField old_element,
    in COBOLField new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_redefined_by_field (in COBOLField old_element)
    raises (Reflective::NotFound, Reflective::MofError);
COBOLField redefined_field ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_redefined_field (in COBOLField new_value)
    raises (Reflective::MofError);
void unset_redefined_field ()
    raises (Reflective::MofError);
COBOLFDSet padded_fd ()
    raises (Reflective::MofError);
void set_padded_fd (in COBOLFDSet new_value)
    raises (Reflective::MofError);
void unset_padded_fd ()
    raises (Reflective::MofError);
void add_padded_fd (in COBOLFD new_element)
    raises (Reflective::MofError);
void modify_padded_fd (
    in COBOLFD old_element,
    in COBOLFD new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_padded_fd (in COBOLFD old_element)
    raises (Reflective::NotFound, Reflective::MofError);
LinageInfoSet lineage_info ()

```

```

    raises (Reflective::MofError);
void set_linage_info (in LinageInfoSet new_value)
    raises (Reflective::MofError);
void unset_linage_info ()
    raises (Reflective::MofError);
void add_linage_info (in LinageInfo new_element)
    raises (Reflective::MofError);
void modify_linage_info (
    in LinageInfo old_element,
    in LinageInfo new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_linage_info (in LinageInfo old_element)
    raises (Reflective::NotFound, Reflective::MofError);
COBOLFDSet depending_fd ()
    raises (Reflective::MofError);
void set_dependent_fd (in COBOLFDSet new_value)
    raises (Reflective::MofError);
void unset_dependent_fd ()
    raises (Reflective::MofError);
void add_dependent_fd (in COBOLFD new_element)
    raises (Reflective::MofError);
void modify_dependent_fd (
    in COBOLFD old_element,
    in COBOLFD new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_dependent_fd (in COBOLFD old_element)
    raises (Reflective::NotFound, Reflective::MofError);
COBOLFDSet status_fd ()
    raises (Reflective::MofError);
void set_status_fd (in COBOLFDSet new_value)
    raises (Reflective::MofError);
void unset_status_fd ()
    raises (Reflective::MofError);
void add_status_fd (in COBOLFD new_element)
    raises (Reflective::MofError);
void modify_status_fd (
    in COBOLFD old_element,
    in COBOLFD new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_status_fd (in COBOLFD old_element)
    raises (Reflective::NotFound, Reflective::MofError);
RenamesSet first_renames ()
    raises (Reflective::MofError);
void set_first_renames (in RenamesSet new_value)
    raises (Reflective::MofError);
void unset_first_renames ()
    raises (Reflective::MofError);
void add_first_renames (in Renames new_element)
    raises (Reflective::MofError);
void modify_first_renames (
    in Renames old_element,
    in Renames new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_first_renames (in Renames old_element)
    raises (Reflective::NotFound, Reflective::MofError);
RenamesSet thru_renames ()
    raises (Reflective::MofError);
void set_thru_renames (in RenamesSet new_value)
    raises (Reflective::MofError);
void unset_thru_renames ()
    raises (Reflective::MofError);

```

```

void add_thru_renames (in Renames new_element)
    raises (Reflective::MofError);
void modify_thru_renames (
    in Renames old_element,
    in Renames new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_thru_renames (in Renames old_element)
    raises (Reflective::NotFound, Reflective::MofError);
COBOLFDSet relative_fd ()
    raises (Reflective::MofError);
void set_relative_fd (in COBOLFDSet new_value)
    raises (Reflective::MofError);
void unset_relative_fd ()
    raises (Reflective::MofError);
void add_relative_fd (in COBOLFD new_element)
    raises (Reflective::MofError);
void modify_relative_fd (
    in COBOLFD old_element,
    in COBOLFD new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_relative_fd (in COBOLFD old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface COBOLField

interface UsageClass : Foundation::Core::DataTypeClass
{
    readonly attribute UsageSet all_of_type_usage;
    readonly attribute UsageSet all_of_class_usage;
    Usage create_usage (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface Usage : UsageClass, Foundation::Core::DataType
{
}; // end of interface Usage

interface COBOLFDClass : Record::RecordFileClass, Foundation::Core::ClassClass
{
    readonly attribute COBOLFDSet all_of_type_cobolfd;
    readonly attribute COBOLFDSet all_of_class_cobolfd;
    COBOLFD create_cobolfd (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_self_describing,
        in Foundation::DataTypes::Integer skip_records,
        in boolean is_active,
        in FileOrganization organization,
        in AccessType access_mode,
        in boolean is_optional,
        in Foundation::DataTypes::Integer reserve_areas,
        in string assign_to,

```



```

        in string code_set_lit,
        in BlockKind block_size_unit,
        in Foundation::DataTypes::Integer min_blocks,
        in Foundation::DataTypes::Integer max_blocks,
        in Foundation::DataTypes::Integer min_records,
        in Foundation::DataTypes::Integer max_records,
        in LabelKind label_kind,
        in boolean is_external,
        in boolean is_global,
        in string pad_literal)
        raises (Reflective::MofError);
};

```

```

interface COBOLFD : COBOLFDClass, ::Record::RecordFile, Foundation::Core::Class
{
    FileOrganization organization ()
        raises (Reflective::MofError);
    void set_organization (in FileOrganization new_value)
        raises (Reflective::MofError);
    AccessType access_mode ()
        raises (Reflective::MofError);
    void set_access_mode (in AccessType new_value)
        raises (Reflective::MofError);
    boolean is_optional ()
        raises (Reflective::MofError);
    void set_is_optional (in boolean new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer reserve_areas ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_reserve_areas (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_reserve_areas ()
        raises (Reflective::MofError);
    string assign_to ()
        raises (Reflective::MofError);
    void set_assign_to (in string new_value)
        raises (Reflective::MofError);
    string code_set_lit ()
        raises (Reflective::MofError);
    void set_code_set_lit (in string new_value)
        raises (Reflective::MofError);
    BlockKind block_size_unit ()
        raises (Reflective::MofError);
    void set_block_size_unit (in BlockKind new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer min_blocks ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_min_blocks (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_min_blocks ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer max_blocks ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_max_blocks (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_max_blocks ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer min_records ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_min_records (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
}

```

```

void unset_min_records ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer max_records ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_max_records (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_max_records ()
    raises (Reflective::MofError);
LabelKind label_kind ()
    raises (Reflective::MofError);
void set_label_kind (in LabelKind new_value)
    raises (Reflective::MofError);
boolean is_external ()
    raises (Reflective::MofError);
void set_is_external (in boolean new_value)
    raises (Reflective::MofError);
boolean is_global ()
    raises (Reflective::MofError);
void set_is_global (in boolean new_value)
    raises (Reflective::MofError);
string pad_literal ()
    raises (Reflective::MofError);
void set_pad_literal (in string new_value)
    raises (Reflective::MofError);
FileSection file_section ()
    raises (Reflective::MofError);
void set_file_section (in FileSection new_value)
    raises (Reflective::MofError);
LinageInfoUList linage_info ()
    raises (Reflective::MofError);
void set_linage_info (in LinageInfoUList new_value)
    raises (Reflective::MofError);
void unset_linage_info ()
    raises (Reflective::MofError);
void add_linage_info (in LinageInfo new_element)
    raises (Reflective::MofError);
void add_linage_info_before (
    in LinageInfo new_element,
    in LinageInfo before_element)
    raises (Reflective::NotFound, Reflective::MofError);
void modify_linage_info (
    in LinageInfo old_element,
    in LinageInfo new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_linage_info (in LinageInfo old_element)
    raises (Reflective::NotFound, Reflective::MofError);
COBOLField status_id ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_status_id (in COBOLField new_value)
    raises (Reflective::MofError);
void unset_status_id ()
    raises (Reflective::MofError);
COBOLField depends_on ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_depends_on (in COBOLField new_value)
    raises (Reflective::MofError);
void unset_depends_on ()
    raises (Reflective::MofError);
COBOLField pad_field ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_pad_field (in COBOLField new_value)

```

```

        raises (Reflective::MofError);
void unset_pad_field ()
    raises (Reflective::MofError);
COBOLField relative_field ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_relative_field (in COBOLField new_value)
    raises (Reflective::MofError);
void unset_relative_field ()
    raises (Reflective::MofError);
}; // end of interface COBOLFD

interface RenamesClass : Foundation::Core::FeatureClass
{
    readonly attribute RenamesSet all_of_type_renames;
    readonly attribute RenamesSet all_of_class_renames;
    Renames create_renames (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope)
        raises (Reflective::MofError);
};

interface Renames : RenamesClass, Foundation::Core::Feature
{
    COBOLField first_field ()
        raises (Reflective::MofError);
    void set_first_field (in COBOLField new_value)
        raises (Reflective::MofError);
    COBOLField thru_field ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_thru_field (in COBOLField new_value)
        raises (Reflective::MofError);
    void unset_thru_field ()
        raises (Reflective::MofError);
}; // end of interface Renames

interface SectionClass : Foundation::Core::ClassifierClass
{
    readonly attribute SectionSet all_of_type_section;
    readonly attribute SectionSet all_of_class_section;
    Section create_section (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface Section : SectionClass, Foundation::Core::Classifier
{
    RecordDefUList record ()
        raises (Reflective::MofError);
    void set_record (in RecordDefUList new_value)
        raises (Reflective::MofError);
    void unset_record ()
        raises (Reflective::MofError);
    void add_record (in ::Record::RecordDef new_element)
        raises (Reflective::MofError);
};

```

```

void add_record_before (
    in ::Record::RecordDef new_element,
    in ::Record::RecordDef before_element)
    raises (Reflective::NotFound, Reflective::MofError);
void modify_record (
    in ::Record::RecordDef old_element,
    in ::Record::RecordDef new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_record (in ::Record::RecordDef old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface Section

interface COBOLFDIndexClass : KeysIndexes::IndexClass
{
    readonly attribute COBOLFDIndexSet all_of_type_cobolfdindex;
    readonly attribute COBOLFDIndexSet all_of_class_cobolfdindex;
    COBOLFDIndex create_cobolfdindex (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,
        in boolean is_partitioning,
        in boolean is_alternate)
        raises (Reflective::MofError);
};

interface COBOLFDIndex : COBOLFDIndexClass, KeysIndexes::Index
{
    boolean is_alternate ()
        raises (Reflective::MofError);
    void set_is_alternate (in boolean new_value)
        raises (Reflective::MofError);
}; // end of interface COBOLFDIndex

interface LinageInfoClass : Foundation::Core::ModelElementClass
{
    readonly attribute LinageInfoSet all_of_type_linage_info;
    readonly attribute LinageInfoSet all_of_class_linage_info;
    LinageInfo create_linage_info (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::Integer value,
        in LinageInfoType type)
        raises (Reflective::MofError);
};

interface LinageInfo : LinageInfoClass, Foundation::Core::ModelElement
{
    Foundation::DataTypes::Integer value ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_value (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_value ()
        raises (Reflective::MofError);
    LinageInfoType type ()
        raises (Reflective::MofError);
    void set_type (in LinageInfoType new_value)
        raises (Reflective::MofError);
    COBOLFD cobol_fd ()

```

```

        raises (Reflective::MofError);
void set_cobol_fd (in COBOLFD new_value)
    raises (Reflective::MofError);
COBOLField cobol_field ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_cobol_field (in COBOLField new_value)
    raises (Reflective::MofError);
void unset_cobol_field ()
    raises (Reflective::MofError);
}; // end of interface LinageInfo

interface ReportWriterSectionClass : SectionClass
{
    readonly attribute ReportWriterSectionSet all_of_type_report_writer_section;
    readonly attribute ReportWriterSectionSet all_of_class_report_writer_section;
    ReportWriterSection create_report_writer_section (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface ReportWriterSection : ReportWriterSectionClass, Section
{
}; // end of interface ReportWriterSection

interface WorkingStorageSectionClass : SectionClass
{
    readonly attribute WorkingStorageSectionSet all_of_type_working_storage_section;
    readonly attribute WorkingStorageSectionSet all_of_class_working_storage_section;
    WorkingStorageSection create_working_storage_section (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface WorkingStorageSection : WorkingStorageSectionClass, Section
{
}; // end of interface WorkingStorageSection

interface FileSectionClass : SectionClass
{
    readonly attribute FileSectionSet all_of_type_file_section;
    readonly attribute FileSectionSet all_of_class_file_section;
    FileSection create_file_section (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

```

```

interface FileSection : FileSectionClass, Section
{
    COBOLFDList cobol_fd ()
        raises (Reflective::MofError);
    void set_cobol_fd (in COBOLFDList new_value)
        raises (Reflective::MofError);
    void unset_cobol_fd ()
        raises (Reflective::MofError);
    void add_cobol_fd (in COBOLFD new_element)
        raises (Reflective::MofError);
    void add_cobol_fd_before (
        in COBOLFD new_element,
        in COBOLFD before_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_cobol_fd (
        in COBOLFD old_element,
        in COBOLFD new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_cobol_fd (in COBOLFD old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface FileSection

interface LinkageSectionClass : SectionClass
{
    readonly attribute LinkageSectionSet all_of_type_linkage_section;
    readonly attribute LinkageSectionSet all_of_class_linkage_section;
    LinkageSection create_linkage_section (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface LinkageSection : LinkageSectionClass, Section
{
}; // end of interface LinkageSection

interface OccursKeyClass : Foundation::Core::ModelElementClass
{
    readonly attribute OccursKeySet all_of_type_occurs_key;
    readonly attribute OccursKeySet all_of_class_occurs_key;
    OccursKey create_occurs_key (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_ascending)
        raises (Reflective::MofError);
};

interface OccursKey : OccursKeyClass, Foundation::Core::ModelElement
{
    boolean is_ascending ()
        raises (Reflective::MofError);
    void set_is_ascending (in boolean new_value)
        raises (Reflective::MofError);
    COBOLField occurs_key_field ()
        raises (Reflective::MofError);
    void set_occurs_key_field (in COBOLField new_value)

```

```

        raises (Reflective::MofError);
        COBOLField occurs_key_of ()
        raises (Reflective::MofError);
        void set_occurs_key_of (in COBOLField new_value)
        raises (Reflective::MofError);
}; // end of interface OccursKey

struct SectionRecordLink
{
    ::Record::RecordDef record;
    COBOLData::Section section;
};
typedef sequence<SectionRecordLink> SectionRecordLinkSet;

interface SectionRecord : Reflective::RefAssociation
{
    SectionRecordLinkSet all_section_record_links()
        raises (Reflective::MofError);
    boolean exists (
        in ::Record::RecordDef record,
        in COBOLData::Section section)
        raises (Reflective::MofError);
    RecordDefUList record (in COBOLData::Section section)
        raises (Reflective::MofError);
    SectionSet section (in ::Record::RecordDef record)
        raises (Reflective::MofError);
    void add (
        in ::Record::RecordDef record,
        in COBOLData::Section section)
        raises (Reflective::MofError);
    void add_before_record (
        in ::Record::RecordDef record,
        in COBOLData::Section section,
        in ::Record::RecordDef before)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_record (
        in ::Record::RecordDef record,
        in COBOLData::Section section,
        in ::Record::RecordDef new_record)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_section (
        in ::Record::RecordDef record,
        in COBOLData::Section section,
        in COBOLData::Section new_section)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in ::Record::RecordDef record,
        in COBOLData::Section section)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SectionRecord

interface COBOLDataPackageFactory
{
    COBOLDataPackage create_coboldata_package ()
        raises (Reflective::MofError);
};

interface COBOLDataPackage : Reflective::RefPackage
{
    readonly attribute COBOLFieldClass cobolfield_ref;
    readonly attribute UsageClass usage_ref;
};

```

```
    readonly attribute COBOLFDClass cobolfd_ref;  
    readonly attribute RenamesClass renames_ref;  
    readonly attribute SectionClass section_ref;  
    readonly attribute COBOLFDIndexClass cobolfdindex_ref;  
    readonly attribute LinageInfoClass lineage_info_ref;  
    readonly attribute ReportWriterSectionClass report_writer_section_ref;  
    readonly attribute WorkingStorageSectionClass working_storage_section_ref;  
    readonly attribute FileSectionClass file_section_ref;  
    readonly attribute LinkageSectionClass linkage_section_ref;  
    readonly attribute OccursKeyClass occurs_key_ref;  
    readonly attribute SectionRecord section_record_ref;  
};  
};  
  
#endif
```


2.4 DMSII.idl

```
#ifndef DMSII_IDL
#define DMSII_IDL
#pragma prefix "org.omg.cwm.x.resource"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Record.idl"
#include "Expressions.idl"
#include "BusinessInformation.idl"
#include "ModelManagement.idl"
#include "KeysIndexes.idl"

module DMSII
{
    interface DatabaseClass;
    interface Database;
    typedef sequence<Database> DatabaseSet;
    interface RemapClass;
    interface Remap;
    typedef sequence<Remap> RemapSet;
    interface DataSetClass;
    interface DataSet;
    typedef sequence<DataSet> DataSetSet;
    interface DataltemClass;
    interface Dataltem;
    typedef sequence<Dataltem> DataltemSet;
    typedef sequence<Dataltem> DataltemUList;
    interface VariableFormatPartClass;
    interface VariableFormatPart;
    typedef sequence<VariableFormatPart> VariableFormatPartSet;
    interface SetStructureClass;
    interface SetStructure;
    typedef sequence<SetStructure> SetStructureSet;
    interface SetClass;
    interface Set;
    typedef sequence<Set> SetSet;
    interface AccessClass;
    interface Access;
    typedef sequence<Access> AccessSet;
    interface SubsetClass;
    interface Subset;
    typedef sequence<Subset> SubsetSet;
    interface AutomaticSubsetClass;
    interface AutomaticSubset;
    typedef sequence<AutomaticSubset> AutomaticSubsetSet;
    interface KeyItemClass;
    interface KeyItem;
    typedef sequence<KeyItem> KeyItemSet;
    interface RemapItemClass;
    interface RemapItem;
    typedef sequence<RemapItem> RemapItemSet;
    interface FieldBitClass;
    interface FieldBit;
    typedef sequence<FieldBit> FieldBitSet;
    interface RemarkClass;
    interface Remark;
    typedef sequence<Remark> RemarkSet;
}
```

```

interface PhysicalDatabaseClass;
interface PhysicalDatabase;
typedef sequence<PhysicalDatabase> PhysicalDatabaseSet;
interface PhysicalDataSetClass;
interface PhysicalDataSet;
typedef sequence<PhysicalDataSet> PhysicalDataSetSet;
interface DASDLCommentClass;
interface DASDLComment;
typedef sequence<DASDLComment> DASDLCommentSet;
interface PhysicalSetClass;
interface PhysicalSet;
typedef sequence<PhysicalSet> PhysicalSetSet;
interface PhysicalDataSetOverrideClass;
interface PhysicalDataSetOverride;
typedef sequence<PhysicalDataSetOverride> PhysicalDataSetOverrideSet;
interface PhysicalSetOverrideClass;
interface PhysicalSetOverride;
typedef sequence<PhysicalSetOverride> PhysicalSetOverrideSet;
interface PhysicalAccessOverrideClass;
interface PhysicalAccessOverride;
typedef sequence<PhysicalAccessOverride> PhysicalAccessOverrideSet;
interface DASDLPropertyClass;
interface DASDLProperty;
typedef sequence<DASDLProperty> DASDLPropertySet;
typedef sequence<DASDLProperty> DASDLPropertyUList;
interface DMSIIPackage;

```

```

interface DatabaseClass : Foundation::Core::StructuralFeatureClass,
                        ModelManagement::PackageClass

```

```

{
    readonly attribute DatabaseSet all_of_type_database;
    readonly attribute DatabaseSet all_of_class_database;
    Database create_database (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_logical,
        in string guard_file,
        in string source)
        raises (Reflective::MofError);
};

```

```

interface Database : DatabaseClass, Foundation::Core::StructuralFeature,
                    ModelManagement::Package

```

```

{
    boolean is_logical ()
        raises (Reflective::MofError);
    void set_is_logical (in boolean new_value)
        raises (Reflective::MofError);
    string guard_file ()
        raises (Reflective::MofError);
    void set_guard_file (in string new_value)
        raises (Reflective::MofError);
    string source ()

```

```

        raises (Reflective::MofError);
    void set_source (in string new_value)
        raises (Reflective::MofError);
}; // end of interface Database

interface RemapClass : Record::RecordDefClass, Foundation::Core::StructuralFeatureClass
{
    readonly attribute RemapSet all_of_type_remap;
    readonly attribute RemapSet all_of_class_remap;
    Remap create_remap (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in string field_delimiter,
        in boolean is_fixed_width,
        in string record_delimiter,
        in string text_delimiter,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in boolean is_required_all,
        in boolean is_read_only_all,
        in boolean is_giving_exception,
        in Foundation::DataTypes::Expression select_condition)
        raises (Reflective::MofError);
};

interface Remap : RemapClass, Record::RecordDef, Foundation::Core::StructuralFeature
{
    boolean is_required_all ()
        raises (Reflective::MofError);
    void set_is_required_all (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_read_only_all ()
        raises (Reflective::MofError);
    void set_is_read_only_all (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_giving_exception ()
        raises (Reflective::MofError);
    void set_is_giving_exception (in boolean new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Expression select_condition ()
        raises (Reflective::MofError);
    void set_select_condition (in Foundation::DataTypes::Expression new_value)
        raises (Reflective::MofError);
    Foundation::Core::StructuralFeature structure ()
        raises (Reflective::MofError);
    void set_structure (in Foundation::Core::StructuralFeature new_value)
        raises (Reflective::MofError);
}; // end of interface Remap

interface DataSetClass : Foundation::Core::StructuralFeatureClass, Record::RecordDefClass
{
    readonly attribute DataSetSet all_of_type_data_set;
    readonly attribute DataSetSet all_of_class_data_set;
    DataSet create_data_set (

```

```

    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in Foundation::DataTypes::ScopeKind owner_scope,
    in Foundation::DataTypes::Multiplicity multiplicity,
    in Foundation::DataTypes::ChangeableKind changeability,
    in Foundation::DataTypes::ScopeKind target_scope,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract,
    in boolean is_active,
    in string field_delimiter,
    in boolean is_fixed_width,
    in string record_delimiter,
    in string text_delimiter,
    in boolean is_global,
    in string organization,
    in string reorganize,
    in boolean is_required_all)
    raises (Reflective::MofError);
};

interface DataSet : DataSetClass, Foundation::Core::StructuralFeature, Record::RecordDef
{
    boolean is_global ()
        raises (Reflective::MofError);
    void set_is_global (in boolean new_value)
        raises (Reflective::MofError);
    string organization ()
        raises (Reflective::MofError);
    void set_organization (in string new_value)
        raises (Reflective::MofError);
    string reorganize ()
        raises (Reflective::MofError);
    void set_reorganize (in string new_value)
        raises (Reflective::MofError);
    boolean is_required_all ()
        raises (Reflective::MofError);
    void set_is_required_all (in boolean new_value)
        raises (Reflective::MofError);
    Set partitioning_set ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_partitioning_set (in Set new_value)
        raises (Reflective::MofError);
    void unset_partitioning_set ()
        raises (Reflective::MofError);
}; // end of interface DataSet

interface DataItemClass : Record::FieldClass
{
    readonly attribute DataItemSet all_of_type_data_item;
    readonly attribute DataItemSet all_of_class_data_item;
    DataItem create_data_item (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,

```

```

    in Expressions::ExpressionNode null_value,
    in boolean is_required,
    in Foundation::DataTypes::Integer size,
    in Foundation::DataTypes::Integer scale_factor,
    in boolean is_signed,
    in Foundation::DataTypes::Integer occurs,
    in boolean is_virtual,
    in Expressions::ExpressionNode virtual_expression,
    in boolean is_kanji,
    in string ccs_version,
    in boolean is_gemcos_literal,
    in boolean is_gemcos_data,
    in boolean is_gemcos_ssn,
    in boolean is_gemcos_dbsn,
    in boolean is_coms_program,
    in boolean is_coms_id,
    in boolean is_coms_locator,
    in boolean is_coms_outp_q)
    raises (Reflective::MofError);
};

```

```

interface DataItem : DataItemClass, Record::Field
{
    Expressions::ExpressionNode null_value ()
        raises (Reflective::MofError);
    void set_null_value (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    boolean is_required ()
        raises (Reflective::MofError);
    void set_is_required (in boolean new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer size ()
        raises (Reflective::MofError);
    void set_size (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer scale_factor ()
        raises (Reflective::MofError);
    void set_scale_factor (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    boolean is_signed ()
        raises (Reflective::MofError);
    void set_is_signed (in boolean new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer occurs ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_occurs (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_occurs ()
        raises (Reflective::MofError);
    boolean is_virtual ()
        raises (Reflective::MofError);
    void set_is_virtual (in boolean new_value)
        raises (Reflective::MofError);
    Expressions::ExpressionNode virtual_expression ()
        raises (Reflective::MofError);
    void set_virtual_expression (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    boolean is_kanji ()
        raises (Reflective::MofError);
    void set_is_kanji (in boolean new_value)
        raises (Reflective::MofError);
}

```

```

string ccs_version ()
    raises (Reflective::MofError);
void set_ccs_version (in string new_value)
    raises (Reflective::MofError);
boolean is_gemcos_literal ()
    raises (Reflective::MofError);
void set_is_gemcos_literal (in boolean new_value)
    raises (Reflective::MofError);
boolean is_gemcos_data ()
    raises (Reflective::MofError);
void set_is_gemcos_data (in boolean new_value)
    raises (Reflective::MofError);
boolean is_gemcos_ssn ()
    raises (Reflective::MofError);
void set_is_gemcos_ssn (in boolean new_value)
    raises (Reflective::MofError);
boolean is_gemcos_dbsn ()
    raises (Reflective::MofError);
void set_is_gemcos_dbsn (in boolean new_value)
    raises (Reflective::MofError);
boolean is_coms_program ()
    raises (Reflective::MofError);
void set_is_coms_program (in boolean new_value)
    raises (Reflective::MofError);
boolean is_coms_id ()
    raises (Reflective::MofError);
void set_is_coms_id (in boolean new_value)
    raises (Reflective::MofError);
boolean is_coms_locator ()
    raises (Reflective::MofError);
void set_is_coms_locator (in boolean new_value)
    raises (Reflective::MofError);
boolean is_coms_outp_q ()
    raises (Reflective::MofError);
void set_is_coms_outp_q (in boolean new_value)
    raises (Reflective::MofError);
FieldBitSet field_bit ()
    raises (Reflective::MofError);
void set_field_bit (in FieldBitSet new_value)
    raises (Reflective::MofError);
void unset_field_bit ()
    raises (Reflective::MofError);
void add_field_bit (in FieldBit new_element)
    raises (Reflective::MofError);
void modify_field_bit (
    in FieldBit old_element,
    in FieldBit new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_field_bit (in FieldBit old_element)
    raises (Reflective::NotFound, Reflective::MofError);
DataItem occurs_data_item ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_occurs_data_item (in DataItem new_value)
    raises (Reflective::MofError);
void unset_occurs_data_item ()
    raises (Reflective::MofError);
DataItemSet occurring_data_item ()
    raises (Reflective::MofError);
void set_occurring_data_item (in DataItemSet new_value)
    raises (Reflective::MofError);
void unset_occurring_data_item ()

```

```

    raises (Reflective::MofError);
void add_occurring_data_item (in DataItem new_element)
    raises (Reflective::MofError);
void modify_occurring_data_item (
    in DataItem old_element,
    in DataItem new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_occurring_data_item (in DataItem old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SetSet key_data_set ()
    raises (Reflective::MofError);
void set_key_data_set (in SetSet new_value)
    raises (Reflective::MofError);
void unset_key_data_set ()
    raises (Reflective::MofError);
void add_key_data_set (in Set new_element)
    raises (Reflective::MofError);
void modify_key_data_set (
    in Set old_element,
    in Set new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_key_data_set (in Set old_element)
    raises (Reflective::NotFound, Reflective::MofError);
Foundation::Core::StructuralFeature structure ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_structure (in Foundation::Core::StructuralFeature new_value)
    raises (Reflective::MofError);
void unset_structure ()
    raises (Reflective::MofError);
RemapItemSet remap_item ()
    raises (Reflective::MofError);
void set_remap_item (in RemapItemSet new_value)
    raises (Reflective::MofError);
void unset_remap_item ()
    raises (Reflective::MofError);
void add_remap_item (in RemapItem new_element)
    raises (Reflective::MofError);
void modify_remap_item (
    in RemapItem old_element,
    in RemapItem new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_remap_item (in RemapItem old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface DataItem

interface VariableFormatPartClass : Foundation::Core::StructuralFeatureClass,
                                   Record::RecordDefClass
{
    readonly attribute VariableFormatPartSet all_of_type_variable_format_part;
    readonly attribute VariableFormatPartSet all_of_class_variable_format_part;
    VariableFormatPart create_variable_format_part (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,

```

```

        in boolean is_active,
        in string field_delimiter,
        in boolean is_fixed_width,
        in string record_delimiter,
        in string text_delimiter,
        in Foundation::DataTypes::Integer vf_label,
        in Foundation::DataTypes::BooleanExpression select_condition)
        raises (Reflective::MofError);
};

interface VariableFormatPart : VariableFormatPartClass, Foundation::Core::StructuralFeature,
                               Record::RecordDef
{
    Foundation::DataTypes::Integer vf_label ()
        raises (Reflective::MofError);
    void set_vf_label (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::BooleanExpression select_condition ()
        raises (Reflective::MofError);
    void set_select_condition (in Foundation::DataTypes::BooleanExpression new_value)
        raises (Reflective::MofError);
}; // end of interface VariableFormatPart

interface SetStructureClass : KeysIndexes::IndexClass, Foundation::Core::StructuralFeatureClass
{
    readonly attribute SetStructureSet all_of_type_set_structure;
    readonly attribute SetStructureSet all_of_class_set_structure;
    SetStructure create_set_structure (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,
        in boolean is_partitioning,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in string duplicates)
        raises (Reflective::MofError);
};

interface SetStructure : SetStructureClass, KeysIndexes::Index,
                          Foundation::Core::StructuralFeature
{
    string duplicates ()
        raises (Reflective::MofError);
    void set_duplicates (in string new_value)
        raises (Reflective::MofError);
}; // end of interface SetStructure

interface SetClass : SetStructureClass
{
    readonly attribute SetSet all_of_type_set;
    readonly attribute SetSet all_of_class_set;
    Set create_set (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,

```



```

        in boolean is_partitioning,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in string duplicates,
        in string dmsii_set_type,
        in string reorganize)
        raises (Reflective::MofError);
};

```

```

interface Set : SetClass, SetStructure
{
    string dmsii_set_type ()
        raises (Reflective::MofError);
    void set_dmsii_set_type (in string new_value)
        raises (Reflective::MofError);
    string reorganize ()
        raises (Reflective::MofError);
    void set_reorganize (in string new_value)
        raises (Reflective::MofError);
    Set partitioning_set ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_partitioning_set (in Set new_value)
        raises (Reflective::MofError);
    void unset_partitioning_set ()
        raises (Reflective::MofError);
    SetSet partitioned_set ()
        raises (Reflective::MofError);
    void set_partitioned_set (in SetSet new_value)
        raises (Reflective::MofError);
    void unset_partitioned_set ()
        raises (Reflective::MofError);
    void add_partitioned_set (in Set new_element)
        raises (Reflective::MofError);
    void modify_partitioned_set (
        in Set old_element,
        in Set new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_partitioned_set (in Set old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    DataltemUList key_data_item ()
        raises (Reflective::MofError);
    void set_key_data_item (in DataltemUList new_value)
        raises (Reflective::MofError);
    void unset_key_data_item ()
        raises (Reflective::MofError);
    void add_key_data_item (in Dataltem new_element)
        raises (Reflective::MofError);
    void add_key_data_item_before (
        in Dataltem new_element,
        in Dataltem before_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_key_data_item (
        in Dataltem old_element,
        in Dataltem new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_key_data_item (in Dataltem old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    DataSet partitioned_data_set ()
        raises (Reflective::NotSet, Reflective::MofError);
}

```

```

void set_partitioned_data_set (in DataSet new_value)
    raises (Reflective::MofError);
void unset_partitioned_data_set ()
    raises (Reflective::MofError);
}; // end of interface Set

interface AccessClass : SetStructureClass
{
    readonly attribute AccessSet all_of_type_access;
    readonly attribute AccessSet all_of_class_access;
    Access create_access (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,
        in boolean is_partitioning,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in string duplicates)
        raises (Reflective::MofError);
};

interface Access : AccessClass, SetStructure
{
}; // end of interface Access

interface SubsetClass : SetClass
{
    readonly attribute SubsetSet all_of_type_subset;
    readonly attribute SubsetSet all_of_class_subset;
    Subset create_subset (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,
        in boolean is_partitioning,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in string duplicates,
        in string dmsii_set_type,
        in string reorganize)
        raises (Reflective::MofError);
};

interface Subset : SubsetClass, Set
{
}; // end of interface Subset

interface AutomaticSubsetClass : SubsetClass
{
    readonly attribute AutomaticSubsetSet all_of_type_automatic_subset;
    readonly attribute AutomaticSubsetSet all_of_class_automatic_subset;
    AutomaticSubset create_automatic_subset (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,

```

```

        in boolean is_specification,
        in boolean is_unique,
        in boolean is_sorted,
        in boolean is_partitioning,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in string duplicates,
        in string dmsii_set_type,
        in string reorganize,
        in Foundation::DataTypes::BooleanExpression condition)
        raises (Reflective::MofError);
};

interface AutomaticSubset : AutomaticSubsetClass, Subset
{
    Foundation::DataTypes::BooleanExpression condition ()
        raises (Reflective::MofError);
    void set_condition (in Foundation::DataTypes::BooleanExpression new_value)
        raises (Reflective::MofError);
}; // end of interface AutomaticSubset

interface KeyItemClass : KeysIndexes::IndexedFeatureClass
{
    readonly attribute KeyItemSet all_of_type_key_item;
    readonly attribute KeyItemSet all_of_class_key_item;
    KeyItem create_key_item (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_ascending,
        in string collation)
        raises (Reflective::MofError);
};

interface KeyItem : KeyItemClass, KeysIndexes::IndexedFeature
{
    string collation ()
        raises (Reflective::MofError);
    void set_collation (in string new_value)
        raises (Reflective::MofError);
}; // end of interface KeyItem

interface RemapItemClass : Record::FieldClass
{
    readonly attribute RemapItemSet all_of_type_remap_item;
    readonly attribute RemapItemSet all_of_class_remap_item;
    RemapItem create_remap_item (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in Foundation::DataTypes::Integer occurs,
        in boolean is_required,
        in boolean is_hidden,
        in boolean is_read_only,

```

```

        in boolean is_giving_exception,
        in boolean is_virtual,
        in Expressions::ExpressionNode virtual_expression)
        raises (Reflective::MofError);
};

interface RemapItem : RemapItemClass, Record::Field
{
    Foundation::DataTypes::Integer occurs ()
        raises (Reflective::MofError);
    void set_occurs (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    boolean is_required ()
        raises (Reflective::MofError);
    void set_is_required (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_hidden ()
        raises (Reflective::MofError);
    void set_is_hidden (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_read_only ()
        raises (Reflective::MofError);
    void set_is_read_only (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_giving_exception ()
        raises (Reflective::MofError);
    void set_is_giving_exception (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_virtual ()
        raises (Reflective::MofError);
    void set_is_virtual (in boolean new_value)
        raises (Reflective::MofError);
    Expressions::ExpressionNode virtual_expression ()
        raises (Reflective::MofError);
    void set_virtual_expression (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    DataltemSet data_item ()
        raises (Reflective::MofError);
    void set_data_item (in DataltemSet new_value)
        raises (Reflective::MofError);
    void unset_data_item ()
        raises (Reflective::MofError);
    void add_data_item (in Dataltem new_element)
        raises (Reflective::MofError);
    void modify_data_item (
        in Dataltem old_element,
        in Dataltem new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_data_item (in Dataltem old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface RemapItem

interface FieldBitClass : Foundation::Core::ModelElementClass
{
    readonly attribute FieldBitSet all_of_type_field_bit;
    readonly attribute FieldBitSet all_of_class_field_bit;
    FieldBit create_field_bit (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);

```

```

};

interface FieldBit : FieldBitClass, Foundation::Core::ModelElement
{
    Dataltem data_item ()
        raises (Reflective::MofError);
    void set_data_item (in Dataltem new_value)
        raises (Reflective::MofError);
}; // end of interface FieldBit

interface RemarkClass : Foundation::Core::StructuralFeatureClass
{
    readonly attribute RemarkSet all_of_type_remark;
    readonly attribute RemarkSet all_of_class_remark;
    Remark create_remark (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in string text)
        raises (Reflective::MofError);
};

interface Remark : RemarkClass, Foundation::Core::StructuralFeature
{
    string text ()
        raises (Reflective::MofError);
    void set_text (in string new_value)
        raises (Reflective::MofError);
}; // end of interface Remark

interface PhysicalDatabaseClass : ModelManagement::PackageClass
{
    readonly attribute PhysicalDatabaseSet all_of_type_physical_database;
    readonly attribute PhysicalDatabaseSet all_of_class_physical_database;
    PhysicalDatabase create_physical_database (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface PhysicalDatabase : PhysicalDatabaseClass, ModelManagement::Package
{
}; // end of interface PhysicalDatabase

interface PhysicalDataSetClass : Foundation::Core::ModelElementClass
{
    readonly attribute PhysicalDataSetSet all_of_type_physical_data_set;
    readonly attribute PhysicalDataSetSet all_of_class_physical_data_set;
    PhysicalDataSet create_physical_data_set (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);
};

```

```

};

interface PhysicalDataSet : PhysicalDataSetClass, Foundation::Core::ModelElement
{
}; // end of interface PhysicalDataSet

interface DASDLCommentClass : BusinessInformation::DescriptionClass
{
    readonly attribute DASDLCommentSet all_of_type_dasdlcomment;
    readonly attribute DASDLCommentSet all_of_class_dasdlcomment;
    DASDLComment create_dasdlcomment (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string body,
        in Foundation::DataTypes::Name language)
        raises (Reflective::MofError);
};

interface DASDLComment : DASDLCommentClass, BusinessInformation::Description
{
}; // end of interface DASDLComment

interface PhysicalSetClass : Foundation::Core::ModelElementClass
{
    readonly attribute PhysicalSetSet all_of_type_physical_set;
    readonly attribute PhysicalSetSet all_of_class_physical_set;
    PhysicalSet create_physical_set (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);
};

interface PhysicalSet : PhysicalSetClass, Foundation::Core::ModelElement
{
}; // end of interface PhysicalSet

interface PhysicalDataSetOverrideClass : Foundation::Core::FeatureClass
{
    readonly attribute PhysicalDataSetOverrideSet all_of_type_physical_data_set_override;
    readonly attribute PhysicalDataSetOverrideSet all_of_class_physical_data_set_override;
    PhysicalDataSetOverride create_physical_data_set_override (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope)
        raises (Reflective::MofError);
};

interface PhysicalDataSetOverride : PhysicalDataSetOverrideClass, Foundation::Core::Feature
{
}; // end of interface PhysicalDataSetOverride

interface PhysicalSetOverrideClass : Foundation::Core::FeatureClass
{
    readonly attribute PhysicalSetOverrideSet all_of_type_physical_set_override;
    readonly attribute PhysicalSetOverrideSet all_of_class_physical_set_override;
    PhysicalSetOverride create_physical_set_override (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,

```

```

        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope)
        raises (Reflective::MofError);
};

interface PhysicalSetOverride : PhysicalSetOverrideClass, Foundation::Core::Feature
{
}; // end of interface PhysicalSetOverride

interface PhysicalAccessOverrideClass : Foundation::Core::FeatureClass
{
    readonly attribute PhysicalAccessOverrideSet all_of_type_physical_access_override;
    readonly attribute PhysicalAccessOverrideSet all_of_class_physical_access_override;
    PhysicalAccessOverride create_physical_access_override (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope)
        raises (Reflective::MofError);
};

interface PhysicalAccessOverride : PhysicalAccessOverrideClass, Foundation::Core::Feature
{
}; // end of interface PhysicalAccessOverride

interface DASDLPropertyClass : Foundation::Core::ModelElementClass
{
    readonly attribute DASDLPropertySet all_of_type_dasdlproperty;
    readonly attribute DASDLPropertySet all_of_class_dasdlproperty;
    DASDLProperty create_dasdlproperty (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string text)
        raises (Reflective::MofError);
};

interface DASDLProperty : DASDLPropertyClass, Foundation::Core::ModelElement
{
    string text ()
        raises (Reflective::MofError);
    void set_text (in string new_value)
        raises (Reflective::MofError);
    Foundation::Core::ModelElement owner ()
        raises (Reflective::MofError);
    void set_owner (in Foundation::Core::ModelElement new_value)
        raises (Reflective::MofError);
}; // end of interface DASDLProperty

struct DataItemStructureLink
{
    Foundation::Core::StructuralFeature structure;
    DataItem data_item;
};
typedef sequence<DataItemStructureLink> DataItemStructureLinkSet;

interface DataItemStructure : Reflective::RefAssociation
{
    DataItemStructureLinkSet all_data_item_structure_links()
        raises (Reflective::MofError);
    boolean exists (

```

```

    in Foundation::Core::StructuralFeature structure,
    in Dataltem data_item)
    raises (Reflective::MofError);
Foundation::Core::StructuralFeature structure (in Dataltem data_item)
    raises (Reflective::MofError);
DataltemSet data_item (in Foundation::Core::StructuralFeature structure)
    raises (Reflective::MofError);
void add (
    in Foundation::Core::StructuralFeature structure,
    in Dataltem data_item)
    raises (Reflective::MofError);
void modify_structure (
    in Foundation::Core::StructuralFeature structure,
    in Dataltem data_item,
    in Foundation::Core::StructuralFeature new_structure)
    raises (Reflective::NotFound, Reflective::MofError);
void modify_data_item (
    in Foundation::Core::StructuralFeature structure,
    in Dataltem data_item,
    in Dataltem new_data_item)
    raises (Reflective::NotFound, Reflective::MofError);
void remove (
    in Foundation::Core::StructuralFeature structure,
    in Dataltem data_item)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface DataltemStructure

struct DASDLPropertyOwnerLink
{
    Foundation::Core::ModelElement owner;
    DASDLProperty dasdl_property;
};
typedef sequence<DASDLPropertyOwnerLink> DASDLPropertyOwnerLinkSet;

interface DASDLPropertyOwner : Reflective::RefAssociation
{
    DASDLPropertyOwnerLinkSet all_dasdlproperty_owner_links()
        raises (Reflective::MofError);
    boolean exists (
        in Foundation::Core::ModelElement owner,
        in DASDLProperty dasdl_property)
        raises (Reflective::MofError);
    Foundation::Core::ModelElement owner (in DASDLProperty dasdl_property)
        raises (Reflective::MofError);
    DASDLPropertyUList dasdl_property (in Foundation::Core::ModelElement owner)
        raises (Reflective::MofError);
    void add (
        in Foundation::Core::ModelElement owner,
        in DASDLProperty dasdl_property)
        raises (Reflective::MofError);
    void add_before_dasdl_property (
        in Foundation::Core::ModelElement owner,
        in DASDLProperty dasdl_property,
        in DASDLProperty before)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_owner (
        in Foundation::Core::ModelElement owner,
        in DASDLProperty dasdl_property,
        in Foundation::Core::ModelElement new_owner)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_dasdl_property (

```



```

        in Foundation::Core::ModelElement owner,
        in DASDLPProperty dasdl_property,
        in DASDLPProperty new_dasdl_property)
        raises (Reflective::NotFound, Reflective::MofError);
void remove (
    in Foundation::Core::ModelElement owner,
    in DASDLPProperty dasdl_property)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface DASDLPPropertyOwner

struct RemappedStructureLink
{
    Foundation::Core::StructuralFeature structure;
    DMSII::Remap remap;
};
typedef sequence<RemappedStructureLink> RemappedStructureLinkSet;

interface RemappedStructure : Reflective::RefAssociation
{
    RemappedStructureLinkSet all_remapped_structure_links()
        raises (Reflective::MofError);
    boolean exists (
        in Foundation::Core::StructuralFeature structure,
        in DMSII::Remap remap)
        raises (Reflective::MofError);
    Foundation::Core::StructuralFeature structure (in DMSII::Remap remap)
        raises (Reflective::MofError);
    RemapSet remap (in Foundation::Core::StructuralFeature structure)
        raises (Reflective::MofError);
    void add (
        in Foundation::Core::StructuralFeature structure,
        in DMSII::Remap remap)
        raises (Reflective::MofError);
    void modify_structure (
        in Foundation::Core::StructuralFeature structure,
        in DMSII::Remap remap,
        in Foundation::Core::StructuralFeature new_structure)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_remap (
        in Foundation::Core::StructuralFeature structure,
        in DMSII::Remap remap,
        in DMSII::Remap new_remap)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Foundation::Core::StructuralFeature structure,
        in DMSII::Remap remap)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface RemappedStructure

interface DMSIIPackageFactory
{
    DMSIIPackage create_dmsii_package ()
        raises (Reflective::MofError);
};

interface DMSIIPackage : Reflective::RefPackage
{
    readonly attribute DatabaseClass database_ref;
    readonly attribute RemapClass remap_ref;
    readonly attribute DataSetClass data_set_ref;
    readonly attribute DataItemClass data_item_ref;

```

```

    readonly attribute VariableFormatPartClass variable_format_part_ref;
    readonly attribute SetStructureClass set_structure_ref;
    readonly attribute SetClass set_ref;
    readonly attribute AccessClass access_ref;
    readonly attribute SubsetClass subset_ref;
    readonly attribute AutomaticSubsetClass automatic_subset_ref;
    readonly attribute KeyItemClass key_item_ref;
    readonly attribute RemapItemClass remap_item_ref;
    readonly attribute FieldBitClass field_bit_ref;
    readonly attribute RemarkClass remark_ref;
    readonly attribute PhysicalDatabaseClass physical_database_ref;
    readonly attribute PhysicalDataSetClass physical_data_set_ref;
    readonly attribute DASDLCommentClass dasdlcomment_ref;
    readonly attribute PhysicalSetClass physical_set_ref;
    readonly attribute PhysicalDataSetOverrideClass physical_data_set_override_ref;
    readonly attribute PhysicalSetOverrideClass physical_set_override_ref;
    readonly attribute PhysicalAccessOverrideClass physical_access_override_ref;
    readonly attribute DASDLPropertyClass dasdlproperty_ref;
    readonly attribute DataItemStructure data_item_structure_ref;
    readonly attribute DASDLPropertyOwner dasdlproperty_owner_ref;
    readonly attribute RemappedStructure remapped_structure_ref;
};
};
#endif

```

2.5 Essbase.idl

```
#ifndef ESSBASE_IDL
#define ESSBASE_IDL
#pragma prefix "org.omg.cwm.x.resource"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Multidimensional.idl"
#include "Expressions.idl"
#include "Olap.idl"
#include "ModelManagement.idl"
#include "SoftwareDeployment.idl"

module Essbase
{
    interface AliasClass;
    interface Alias;
    typedef sequence<Alias> AliasSet;
    interface ApplicationClass;
    interface Application;
    typedef sequence<Application> ApplicationSet;
    interface CommentClass;
    interface Comment;
    typedef sequence<Comment> CommentSet;
    interface ConsolidationClass;
    interface Consolidation;
    typedef sequence<Consolidation> ConsolidationSet;
    interface CurrencyConversionClass;
    interface CurrencyConversion;
    typedef sequence<CurrencyConversion> CurrencyConversionSet;
    interface DataStorageClass;
    interface DataStorage;
    typedef sequence<DataStorage> DataStorageSet;
    interface DatabaseClass;
    interface Database;
    typedef sequence<Database> DatabaseSet;
    interface DimensionClass;
    interface Dimension;
    typedef sequence<Dimension> DimensionSet;
    interface FormulaClass;
    interface Formula;
    typedef sequence<Formula> FormulaSet;
    interface GenerationClass;
    interface Generation;
    typedef sequence<Generation> GenerationSet;
    interface ImmediateParentClass;
    interface ImmediateParent;
    typedef sequence<ImmediateParent> ImmediateParentSet;
    interface LevelClass;
    interface Level;
    typedef sequence<Level> LevelSet;
    interface PartitionClass;
    interface Partition;
    typedef sequence<Partition> PartitionSet;
    interface LinkedPartitionClass;
    interface LinkedPartition;
    typedef sequence<LinkedPartition> LinkedPartitionSet;
    interface MemberNameClass;
```

```

interface MemberName;
typedef sequence<MemberName> MemberNameSet;
interface OLAPServerClass;
interface OLAPServer;
typedef sequence<OLAPServer> OLAPServerSet;
interface OutlineClass;
interface Outline;
typedef sequence<Outline> OutlineSet;
interface ReplicatedPartitionClass;
interface ReplicatedPartition;
typedef sequence<ReplicatedPartition> ReplicatedPartitionSet;
interface TimeBalanceClass;
interface TimeBalance;
typedef sequence<TimeBalance> TimeBalanceSet;
interface TransparentPartitionClass;
interface TransparentPartition;
typedef sequence<TransparentPartition> TransparentPartitionSet;
interface TwoPassCalculationClass;
interface TwoPassCalculation;
typedef sequence<TwoPassCalculation> TwoPassCalculationSet;
interface UDAClass;
interface UDA;
typedef sequence<UDA> UDASet;
interface VarianceReportingClass;
interface VarianceReporting;
typedef sequence<VarianceReporting> VarianceReportingSet;
interface EssbasePackage;
enum DimensionType {ess_none, ess_accounts, ess_time, ess_country, ess_currencyPartition,
                    ess_attribute};

interface AliasClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute AliasSet all_of_type_alias;
    readonly attribute AliasSet all_of_class_alias;
    Alias create_alias (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface Alias : AliasClass, Multidimensional::DimensionedObject
{
}; // end of interface Alias

interface ApplicationClass : ModelManagement::PackageClass
{
    readonly attribute ApplicationSet all_of_type_application;
    readonly attribute ApplicationSet all_of_class_application;
    Application create_application (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
};

```

```

        raises (Reflective::MofError);
    };

interface Application : ApplicationClass, ModelManagement::Package
{
}; // end of interface Application

interface CommentClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute Essbase::CommentSet all_of_type_comment;
    readonly attribute Essbase::CommentSet all_of_class_comment;
    Essbase::Comment create_comment (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface Comment : CommentClass, Multidimensional::DimensionedObject
{
}; // end of interface Comment

interface ConsolidationClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute ConsolidationSet all_of_type_consolidation;
    readonly attribute ConsolidationSet all_of_class_consolidation;
    Consolidation create_consolidation (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface Consolidation : ConsolidationClass, Multidimensional::DimensionedObject
{
}; // end of interface Consolidation

interface CurrencyConversionClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute CurrencyConversionSet all_of_type_currency_conversion;
    readonly attribute CurrencyConversionSet all_of_class_currency_conversion;
    CurrencyConversion create_currency_conversion (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

```

```

};

interface CurrencyConversion : CurrencyConversionClass, Multidimensional::DimensionedObject
{
}; // end of interface CurrencyConversion

interface DataStorageClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute DataStorageSet all_of_type_data_storage;
    readonly attribute DataStorageSet all_of_class_data_storage;
    DataStorage create_data_storage (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface DataStorage : DataStorageClass, Multidimensional::DimensionedObject
{
}; // end of interface DataStorage

interface DatabaseClass : Multidimensional::SchemaClass
{
    readonly attribute DatabaseSet all_of_type_database;
    readonly attribute DatabaseSet all_of_class_database;
    Database create_database (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_currency)
        raises (Reflective::MofError);
};

interface Database : DatabaseClass, Multidimensional::Schema
{
    boolean is_currency ()
        raises (Reflective::MofError);
    void set_is_currency (in boolean new_value)
        raises (Reflective::MofError);
    Essbase::Outline outline ()
        raises (Reflective::MofError);
    void set_outline (in Essbase::Outline new_value)
        raises (Reflective::MofError);
}; // end of interface Database

interface DimensionClass : Multidimensional::CwmDimensionClass
{
    readonly attribute DimensionSet all_of_type_dimension;
    readonly attribute DimensionSet all_of_class_dimension;
    Dimension create_dimension (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,

```

```

        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in DimensionType type,
        in boolean is_dense)
        raises (Reflective::MofError);
};

interface Dimension : DimensionClass, Multidimensional::CwmDimension
{
    DimensionType type ()
        raises (Reflective::MofError);
    void set_type (in DimensionType new_value)
        raises (Reflective::MofError);
    boolean is_dense ()
        raises (Reflective::MofError);
    void set_is_dense (in boolean new_value)
        raises (Reflective::MofError);
    Essbase::Outline outline ()
        raises (Reflective::MofError);
    void set_outline (in Essbase::Outline new_value)
        raises (Reflective::MofError);
}; // end of interface Dimension

interface FormulaClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute FormulaSet all_of_type_formula;
    readonly attribute FormulaSet all_of_class_formula;
    Formula create_formula (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface Formula : FormulaClass, Multidimensional::DimensionedObject
{
}; // end of interface Formula

interface GenerationClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute GenerationSet all_of_type_generation;
    readonly attribute GenerationSet all_of_class_generation;
    Generation create_generation (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

```

```

interface Generation : GenerationClass, Multidimensional::DimensionedObject
{
}; // end of interface Generation

interface ImmediateParentClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute ImmediateParentSet all_of_type_immediate_parent;
    readonly attribute ImmediateParentSet all_of_class_immediate_parent;
    ImmediateParent create_immediate_parent (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface ImmediateParent : ImmediateParentClass, Multidimensional::DimensionedObject
{
}; // end of interface ImmediateParent

interface LevelClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute LevelSet all_of_type_level;
    readonly attribute LevelSet all_of_class_level;
    Level create_level (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface Level : LevelClass, Multidimensional::DimensionedObject
{
}; // end of interface Level

interface PartitionClass : Olap::CubeRegionClass
{
    readonly attribute PartitionSet all_of_type_partition;
};

interface Partition : PartitionClass, Olap::CubeRegion
{
    boolean is_source ()
        raises (Reflective::MofError);
    void set_is_source (in boolean new_value)
        raises (Reflective::MofError);
    boolean is_shared ()
        raises (Reflective::MofError);
    void set_is_shared (in boolean new_value)
        raises (Reflective::MofError);
    Expressions::ExpressionNode formula ()
        raises (Reflective::MofError);
};

```



```

    void set_formula (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
}; // end of interface Partition

interface LinkedPartitionClass : PartitionClass
{
    readonly attribute LinkedPartitionSet all_of_type_linked_partition;
    readonly attribute LinkedPartitionSet all_of_class_linked_partition;
    LinkedPartition create_linked_partition (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in boolean is_read_only,
        in boolean is_fully_realized,
        in boolean is_source,
        in boolean is_shared,
        in Expressions::ExpressionNode formula)
        raises (Reflective::MofError);
};

interface LinkedPartition : LinkedPartitionClass, Partition
{
}; // end of interface LinkedPartition

interface MemberNameClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute MemberNameSet all_of_type_member_name;
    readonly attribute MemberNameSet all_of_class_member_name;
    MemberName create_member_name (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface MemberName : MemberNameClass, Multidimensional::DimensionedObject
{
}; // end of interface MemberName

interface OLAPServerClass : SoftwareDeployment::DataManagerClass
{
    readonly attribute OLAPServerSet all_of_type_olapserver;
    readonly attribute OLAPServerSet all_of_class_olapserver;
    OLAPServer create_olapserver (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in string pathname,
        in boolean is_case_sensitive)
};

```

```

        raises (Reflective::MofError);
};

interface OLAPServer : OLAPServerClass, SoftwareDeployment::DataManager
{
}; // end of interface OLAPServer

interface OutlineClass : Foundation::Core::NamespaceClass
{
    readonly attribute OutlineSet all_of_type_outline;
    readonly attribute OutlineSet all_of_class_outline;
    Outline create_outline (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string alias_table_name)
        raises (Reflective::MofError);
};

interface Outline : OutlineClass, Foundation::Core::Namespace
{
    string alias_table_name ()
        raises (Reflective::MofError);
    void set_alias_table_name (in string new_value)
        raises (Reflective::MofError);
    Essbase::Database database ()
        raises (Reflective::MofError);
    void set_database (in Essbase::Database new_value)
        raises (Reflective::MofError);
    DimensionSet dimension ()
        raises (Reflective::MofError);
    void set_dimension (in DimensionSet new_value)
        raises (Reflective::MofError);
    void unset_dimension ()
        raises (Reflective::MofError);
    void add_dimension (in Essbase::Dimension new_element)
        raises (Reflective::MofError);
    void modify_dimension (
        in Essbase::Dimension old_element,
        in Essbase::Dimension new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_dimension (in Essbase::Dimension old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface Outline

interface ReplicatedPartitionClass : PartitionClass
{
    readonly attribute ReplicatedPartitionSet all_of_type_replicated_partition;
    readonly attribute ReplicatedPartitionSet all_of_class_replicated_partition;
    ReplicatedPartition create_replicated_partition (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in boolean is_read_only,
        in boolean is_fully_realized,
        in boolean is_source,
        in boolean is_shared,

```

```

        in Expressions::ExpressionNode formula)
        raises (Reflective::MofError);
};

interface ReplicatedPartition : ReplicatedPartitionClass, Partition
{
}; // end of interface ReplicatedPartition

interface TimeBalanceClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute TimeBalanceSet all_of_type_time_balance;
    readonly attribute TimeBalanceSet all_of_class_time_balance;
    TimeBalance create_time_balance (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface TimeBalance : TimeBalanceClass, Multidimensional::DimensionedObject
{
}; // end of interface TimeBalance

interface TransparentPartitionClass : PartitionClass
{
    readonly attribute TransparentPartitionSet all_of_type_transparent_partition;
    readonly attribute TransparentPartitionSet all_of_class_transparent_partition;
    TransparentPartition create_transparent_partition (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in boolean is_read_only,
        in boolean is_fully_realized,
        in boolean is_source,
        in boolean is_shared,
        in Expressions::ExpressionNode formula)
        raises (Reflective::MofError);
};

interface TransparentPartition : TransparentPartitionClass, Partition
{
}; // end of interface TransparentPartition

interface TwoPassCalculationClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute TwoPassCalculationSet all_of_type_two_pass_calculation;
    readonly attribute TwoPassCalculationSet all_of_class_two_pass_calculation;
    TwoPassCalculation create_two_pass_calculation (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,

```

```

        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface TwoPassCalculation : TwoPassCalculationClass, Multidimensional::DimensionedObject
{
}; // end of interface TwoPassCalculation

interface UDAClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute UDASet all_of_type_uda;
    readonly attribute UDASet all_of_class_uda;
    UDA create_uda (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface UDA : UDAClass, Multidimensional::DimensionedObject
{
}; // end of interface UDA

interface VarianceReportingClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute VarianceReportingSet all_of_type_variance_reporting;
    readonly attribute VarianceReportingSet all_of_class_variance_reporting;
    VarianceReporting create_variance_reporting (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value)
        raises (Reflective::MofError);
};

interface VarianceReporting : VarianceReportingClass, Multidimensional::DimensionedObject
{
}; // end of interface VarianceReporting

interface EssbasePackageFactory
{
    EssbasePackage create_essbase_package ()
        raises (Reflective::MofError);
};

interface EssbasePackage : Reflective::RefPackage
{
    readonly attribute AliasClass alias_ref;
    readonly attribute ApplicationClass application_ref;
};

```

```

    readonly attribute CommentClass comment_ref;
    readonly attribute ConsolidationClass consolidation_ref;
    readonly attribute CurrencyConversionClass currency_conversion_ref;
    readonly attribute DataStorageClass data_storage_ref;
    readonly attribute DatabaseClass database_ref;
    readonly attribute DimensionClass dimension_ref;
    readonly attribute FormulaClass formula_ref;
    readonly attribute GenerationClass generation_ref;
    readonly attribute ImmediateParentClass immediate_parent_ref;
    readonly attribute LevelClass level_ref;
    readonly attribute PartitionClass partition_ref;
    readonly attribute LinkedPartitionClass linked_partition_ref;
    readonly attribute MemberNameClass member_name_ref;
    readonly attribute OLAPServerClass olapserver_ref;
    readonly attribute OutlineClass outline_ref;
    readonly attribute ReplicatedPartitionClass replicated_partition_ref;
    readonly attribute TimeBalanceClass time_balance_ref;
    readonly attribute TransparentPartitionClass transparent_partition_ref;
    readonly attribute TwoPassCalculationClass two_pass_calculation_ref;
    readonly attribute UDAClass uda_ref;
    readonly attribute VarianceReportingClass variance_reporting_ref;
};
};
#endif

```

2.6 *Express.idl*

```
#ifndef EXPRESS_IDL
#define EXPRESS_IDL
#pragma prefix "org.omg.cwm.x.resource"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Multidimensional.idl"

module Express
{
    interface DimensionClass;
    interface Dimension;
    typedef sequence<Dimension> DimensionSet;
    interface DatabaseClass;
    interface Database;
    typedef sequence<Database> DatabaseSet;
    interface ConjointClass;
    interface Conjoint;
    typedef sequence<Conjoint> ConjointSet;
    interface ProgramClass;
    interface Program;
    typedef sequence<Program> ProgramSet;
    interface ModelClass;
    interface Model;
    typedef sequence<Model> ModelSet;
    interface VariableClass;
    interface Variable;
    typedef sequence<Variable> VariableSet;
    interface FormulaClass;
    interface Formula;
    typedef sequence<Formula> FormulaSet;
    interface ValueSetClass;
    interface ValueSet;
    typedef sequence<ValueSet> ValueSetSet;
    interface RelationClass;
    interface Relation;
    typedef sequence<Relation> RelationSet;
    interface WorksheetClass;
    interface Worksheet;
    typedef sequence<Worksheet> WorksheetSet;
    interface CompositeClass;
    interface Composite;
    typedef sequence<Composite> CompositeSet;
    interface SimpleDimensionClass;
    interface SimpleDimension;
    typedef sequence<SimpleDimension> SimpleDimensionSet;
    interface AliasDimensionClass;
    interface AliasDimension;
    typedef sequence<AliasDimension> AliasDimensionSet;
    interface AggMapClass;
    interface AggMap;
    typedef sequence<AggMap> AggMapSet;
    interface AggMapComponentClass;
    interface AggMapComponent;
    typedef sequence<AggMapComponent> AggMapComponentSet;
    interface PreComputeClauseClass;
    interface PreComputeClause;
}
```

```

typedef sequence<PreComputeClause> PreComputeClauseSet;
interface ExpressPackage;

interface DimensionClass : Multidimensional::CwmDimensionClass
{
    readonly attribute DimensionSet all_of_type_dimension;
};

interface Dimension : DimensionClass, Multidimensional::CwmDimension
{
}; // end of interface Dimension

interface DatabaseClass : Multidimensional::SchemaClass
{
    readonly attribute DatabaseSet all_of_type_database;
    readonly attribute DatabaseSet all_of_class_database;
    Database create_database (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface Database : DatabaseClass, Multidimensional::Schema
{
}; // end of interface Database

interface ConjointClass : DimensionClass
{
    readonly attribute ConjointSet all_of_type_conjoint;
    readonly attribute ConjointSet all_of_class_conjoint;
    Conjoint create_conjoint (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in string search_algorithm,
        in string page_space)
        raises (Reflective::MofError);
};

interface Conjoint : ConjointClass, Dimension
{
    string search_algorithm ()
        raises (Reflective::MofError);
    void set_search_algorithm (in string new_value)
        raises (Reflective::MofError);
    string page_space ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_page_space (in string new_value)
        raises (Reflective::MofError);
    void unset_page_space ()
        raises (Reflective::MofError);
}; // end of interface Conjoint

```

```

interface ProgramClass : Foundation::Core::ComponentClass
{
    readonly attribute Express::ProgramSet all_of_type_program;
    readonly attribute Express::ProgramSet all_of_class_program;
    Express::Program create_program (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in string program,
        in string return_dimension)
        raises (Reflective::MofError);
};

interface Program : ProgramClass, Foundation::Core::Component
{
    string program ()
        raises (Reflective::MofError);
    void set_program (in string new_value)
        raises (Reflective::MofError);
    string return_dimension ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_return_dimension (in string new_value)
        raises (Reflective::MofError);
    void unset_return_dimension ()
        raises (Reflective::MofError);
}; // end of interface Program

interface ModelClass : Foundation::Core::ComponentClass
{
    readonly attribute ModelSet all_of_type_model;
    readonly attribute ModelSet all_of_class_model;
    Model create_model (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in string content)
        raises (Reflective::MofError);
};

interface Model : ModelClass, Foundation::Core::Component
{
    string content ()
        raises (Reflective::MofError);
    void set_content (in string new_value)
        raises (Reflective::MofError);
}; // end of interface Model

interface VariableClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute VariableSet all_of_type_variable;
    readonly attribute VariableSet all_of_class_variable;
    Variable create_variable (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,

```



```

        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in string storage_type,
        in string page_space,
        in Foundation::DataTypes::Integer width)
        raises (Reflective::MofError);
};

interface Variable : VariableClass, Multidimensional::DimensionedObject
{
    string storage_type ()
        raises (Reflective::MofError);
    void set_storage_type (in string new_value)
        raises (Reflective::MofError);
    string page_space ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_page_space (in string new_value)
        raises (Reflective::MofError);
    void unset_page_space ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer width ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_width (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_width ()
        raises (Reflective::MofError);
}; // end of interface Variable

interface FormulaClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute FormulaSet all_of_type_formula;
    readonly attribute FormulaSet all_of_class_formula;
    Formula create_formula (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in string expression)
        raises (Reflective::MofError);
};

interface Formula : FormulaClass, Multidimensional::DimensionedObject
{
    string expression ()
        raises (Reflective::MofError);
    void set_expression (in string new_value)
        raises (Reflective::MofError);
}; // end of interface Formula

interface ValueSetClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute ValueSetSet all_of_type_value_set;
    readonly attribute ValueSetSet all_of_class_value_set;
    ValueSet create_value_set (

```

```

        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in boolean is_temp)
        raises (Reflective::MofError);
};

interface ValueSet : ValueSetClass, Multidimensional::DimensionedObject
{
    boolean is_temp ()
        raises (Reflective::MofError);
    void set_is_temp (in boolean new_value)
        raises (Reflective::MofError);
    Express::Dimension reference_dimension ()
        raises (Reflective::MofError);
    void set_reference_dimension (in Express::Dimension new_value)
        raises (Reflective::MofError);
}; // end of interface ValueSet

interface RelationClass : Multidimensional::DimensionedObjectClass
{
    readonly attribute RelationSet all_of_type_relation;
    readonly attribute RelationSet all_of_class_relation;
    Relation create_relation (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in boolean is_temp,
        in string page_space)
        raises (Reflective::MofError);
};

interface Relation : RelationClass, Multidimensional::DimensionedObject
{
    boolean is_temp ()
        raises (Reflective::MofError);
    void set_is_temp (in boolean new_value)
        raises (Reflective::MofError);
    string page_space ()
        raises (Reflective::MofError);
    void set_page_space (in string new_value)
        raises (Reflective::MofError);
    Express::Dimension reference_dimension ()
        raises (Reflective::MofError);
    void set_reference_dimension (in Express::Dimension new_value)
        raises (Reflective::MofError);
}; // end of interface Relation

interface WorksheetClass : Foundation::Core::ClassClass
{
    readonly attribute WorksheetSet all_of_type_worksheet;

```

```

readonly attribute WorksheetSet all_of_class_worksheet;
Worksheet create_worksheet (
  in Foundation::DataTypes::Name name,
  in Foundation::DataTypes::VisibilityKind visibility,
  in boolean is_specification,
  in boolean is_root,
  in boolean is_leaf,
  in boolean is_abstract,
  in boolean is_active,
  in boolean is_temp)
  raises (Reflective::MofError);
};

interface Worksheet : WorksheetClass, Foundation::Core::Class
{
  boolean is_temp ()
    raises (Reflective::MofError);
  void set_is_temp (in boolean new_value)
    raises (Reflective::MofError);
  Dimension column_dimension ()
    raises (Reflective::NotSet, Reflective::MofError);
  void set_column_dimension (in Dimension new_value)
    raises (Reflective::MofError);
  void unset_column_dimension ()
    raises (Reflective::MofError);
  Dimension row_dimension ()
    raises (Reflective::NotSet, Reflective::MofError);
  void set_row_dimension (in Dimension new_value)
    raises (Reflective::MofError);
  void unset_row_dimension ()
    raises (Reflective::MofError);
}; // end of interface Worksheet

interface CompositeClass : DimensionClass
{
  readonly attribute Express::CompositeSet all_of_type_composite;
  readonly attribute Express::CompositeSet all_of_class_composite;
  Express::Composite create_composite (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract,
    in boolean is_active,
    in string search_algorithm,
    in string page_space)
    raises (Reflective::MofError);
};

interface Composite : CompositeClass, Dimension
{
  string search_algorithm ()
    raises (Reflective::MofError);
  void set_search_algorithm (in string new_value)
    raises (Reflective::MofError);
  string page_space ()
    raises (Reflective::NotSet, Reflective::MofError);
  void set_page_space (in string new_value)
    raises (Reflective::MofError);
  void unset_page_space ()

```

```

        raises (Reflective::MofError);
}; // end of interface Composite

interface SimpleDimensionClass : DimensionClass
{
    readonly attribute SimpleDimensionSet all_of_type_simple_dimension;
    readonly attribute SimpleDimensionSet all_of_class_simple_dimension;
    SimpleDimension create_simple_dimension (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in Foundation::DataTypes::Integer width,
        in boolean is_time,
        in Foundation::DataTypes::Integer multiple,
        in string beginning_phase,
        in string ending_phase,
        in string search_algorithm,
        in string page_space)
        raises (Reflective::MofError);
};

interface SimpleDimension : SimpleDimensionClass, Dimension
{
    Foundation::DataTypes::Integer width ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_width (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_width ()
        raises (Reflective::MofError);
    boolean is_time ()
        raises (Reflective::MofError);
    void set_is_time (in boolean new_value)
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer multiple ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_multiple (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_multiple ()
        raises (Reflective::MofError);
    string beginning_phase ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_beginning_phase (in string new_value)
        raises (Reflective::MofError);
    void unset_beginning_phase ()
        raises (Reflective::MofError);
    string ending_phase ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_ending_phase (in string new_value)
        raises (Reflective::MofError);
    void unset_ending_phase ()
        raises (Reflective::MofError);
    string search_algorithm ()
        raises (Reflective::MofError);
    void set_search_algorithm (in string new_value)
        raises (Reflective::MofError);
    string page_space ()
        raises (Reflective::NotSet, Reflective::MofError);
};

```

```

void set_page_space (in string new_value)
    raises (Reflective::MofError);
void unset_page_space ()
    raises (Reflective::MofError);
Foundation::Core::Classifier data_type ()
    raises (Reflective::MofError);
void set_data_type (in Foundation::Core::Classifier new_value)
    raises (Reflective::MofError);
AliasDimensionSet alias_dimension ()
    raises (Reflective::MofError);
void set_alias_dimension (in AliasDimensionSet new_value)
    raises (Reflective::MofError);
void unset_alias_dimension ()
    raises (Reflective::MofError);
void add_alias_dimension (in AliasDimension new_element)
    raises (Reflective::MofError);
void modify_alias_dimension (
    in AliasDimension old_element,
    in AliasDimension new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_alias_dimension (in AliasDimension old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SimpleDimension

interface AliasDimensionClass : DimensionClass
{
    readonly attribute AliasDimensionSet all_of_type_alias_dimension;
    readonly attribute AliasDimensionSet all_of_class_alias_dimension;
    AliasDimension create_alias_dimension (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active)
        raises (Reflective::MofError);
};

interface AliasDimension : AliasDimensionClass, Dimension
{
    SimpleDimension base_dimension ()
        raises (Reflective::MofError);
    void set_base_dimension (in SimpleDimension new_value)
        raises (Reflective::MofError);
}; // end of interface AliasDimension

interface AggMapClass : Foundation::Core::ClassClass
{
    readonly attribute AggMapSet all_of_type_agg_map;
    readonly attribute AggMapSet all_of_class_agg_map;
    AggMap create_agg_map (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active)
        raises (Reflective::MofError);
};

```

```

interface AggMap : AggMapClass, Foundation::Core::Class
{
    AggMapComponentSet agg_map_component ()
        raises (Reflective::MofError);
    void set_agg_map_component (in AggMapComponentSet new_value)
        raises (Reflective::MofError);
    void unset_agg_map_component ()
        raises (Reflective::MofError);
    void add_agg_map_component (in AggMapComponent new_element)
        raises (Reflective::MofError);
    void modify_agg_map_component (
        in AggMapComponent old_element,
        in AggMapComponent new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_agg_map_component (in AggMapComponent old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface AggMap

interface AggMapComponentClass : Foundation::Core::ModelElementClass
{
    readonly attribute AggMapComponentSet all_of_type_agg_map_component;
    readonly attribute AggMapComponentSet all_of_class_agg_map_component;
    AggMapComponent create_agg_map_component (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string agg_operator)
        raises (Reflective::MofError);
};

interface AggMapComponent : AggMapComponentClass, Foundation::Core::ModelElement
{
    string agg_operator ()
        raises (Reflective::MofError);
    void set_agg_operator (in string new_value)
        raises (Reflective::MofError);
    AggMap agg_map ()
        raises (Reflective::MofError);
    void set_agg_map (in AggMap new_value)
        raises (Reflective::MofError);
    Express::Dimension dimension ()
        raises (Reflective::MofError);
    void set_dimension (in Express::Dimension new_value)
        raises (Reflective::MofError);
    Express::Relation relation ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_relation (in Express::Relation new_value)
        raises (Reflective::MofError);
    void unset_relation ()
        raises (Reflective::MofError);
    PreComputeClause compute_clause ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_compute_clause (in PreComputeClause new_value)
        raises (Reflective::MofError);
    void unset_compute_clause ()
        raises (Reflective::MofError);
}; // end of interface AggMapComponent

interface PreComputeClauseClass : Foundation::Core::ModelElementClass
{

```

```

readonly attribute PreComputeClauseSet all_of_type_pre_compute_clause;
readonly attribute PreComputeClauseSet all_of_class_pre_compute_clause;
PreComputeClause create_pre_compute_clause (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in any status_list)
    raises (Reflective::MofError);
};

interface PreComputeClause : PreComputeClauseClass, Foundation::Core::ModelElement
{
    any status_list ()
        raises (Reflective::MofError);
    void set_status_list (in any new_value)
        raises (Reflective::MofError);
    AggMapComponent agg_map_component ()
        raises (Reflective::MofError);
    void set_agg_map_component (in AggMapComponent new_value)
        raises (Reflective::MofError);
}; // end of interface PreComputeClause

struct RelationReferenceDimensionLink
{
    Dimension reference_dimension;
    Express::Relation relation;
};
typedef sequence<RelationReferenceDimensionLink> RelationReferenceDimensionLinkSet;

interface RelationReferenceDimension : Reflective::RefAssociation
{
    RelationReferenceDimensionLinkSet all_relation_reference_dimension_links()
        raises (Reflective::MofError);
    boolean exists (
        in Dimension reference_dimension,
        in Express::Relation relation)
        raises (Reflective::MofError);
    Dimension reference_dimension (in Express::Relation relation)
        raises (Reflective::MofError);
    RelationSet relation (in Dimension reference_dimension)
        raises (Reflective::MofError);
    void add (
        in Dimension reference_dimension,
        in Express::Relation relation)
        raises (Reflective::MofError);
    void modify_reference_dimension (
        in Dimension reference_dimension,
        in Express::Relation relation,
        in Dimension new_reference_dimension)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_relation (
        in Dimension reference_dimension,
        in Express::Relation relation,
        in Express::Relation new_relation)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Dimension reference_dimension,
        in Express::Relation relation)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface RelationReferenceDimension

```

```

struct WorksheetColumnDimensionLink
{
    Dimension column_dimension;
    Worksheet column_dimension_in_worksheet;
};
typedef sequence<WorksheetColumnDimensionLink> WorksheetColumnDimensionLinkSet;

interface WorksheetColumnDimension : Reflective::RefAssociation
{
    WorksheetColumnDimensionLinkSet all_worksheet_column_dimension_links()
        raises (Reflective::MofError);
    boolean exists (
        in Dimension column_dimension,
        in Worksheet column_dimension_in_worksheet)
        raises (Reflective::MofError);
    Dimension column_dimension (in Worksheet column_dimension_in_worksheet)
        raises (Reflective::MofError);
    WorksheetSet column_dimension_in_worksheet (in Dimension column_dimension)
        raises (Reflective::MofError);
    void add (
        in Dimension column_dimension,
        in Worksheet column_dimension_in_worksheet)
        raises (Reflective::MofError);
    void modify_column_dimension (
        in Dimension column_dimension,
        in Worksheet column_dimension_in_worksheet,
        in Dimension new_column_dimension)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_column_dimension_in_worksheet (
        in Dimension column_dimension,
        in Worksheet column_dimension_in_worksheet,
        in Worksheet new_column_dimension_in_worksheet)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Dimension column_dimension,
        in Worksheet column_dimension_in_worksheet)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface WorksheetColumnDimension

struct WorksheetRowDimensionLink
{
    Worksheet row_dimension_in_worksheet;
    Dimension row_dimension;
};
typedef sequence<WorksheetRowDimensionLink> WorksheetRowDimensionLinkSet;

interface WorksheetRowDimension : Reflective::RefAssociation
{
    WorksheetRowDimensionLinkSet all_worksheet_row_dimension_links()
        raises (Reflective::MofError);
    boolean exists (
        in Worksheet row_dimension_in_worksheet,
        in Dimension row_dimension)
        raises (Reflective::MofError);
    WorksheetSet row_dimension_in_worksheet (in Dimension row_dimension)
        raises (Reflective::MofError);
    Dimension row_dimension (in Worksheet row_dimension_in_worksheet)
        raises (Reflective::MofError);
    void add (
        in Worksheet row_dimension_in_worksheet,
        in Dimension row_dimension)

```



```

        raises (Reflective::MofError);
void modify_row_dimension_in_worksheet (
    in Worksheet row_dimension_in_worksheet,
    in Dimension row_dimension,
    in Worksheet new_row_dimension_in_worksheet)
    raises (Reflective::NotFound, Reflective::MofError);
void modify_row_dimension (
    in Worksheet row_dimension_in_worksheet,
    in Dimension row_dimension,
    in Dimension new_row_dimension)
    raises (Reflective::NotFound, Reflective::MofError);
void remove (
    in Worksheet row_dimension_in_worksheet,
    in Dimension row_dimension)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface WorksheetRowDimension

struct ValueSetReferenceDimensionLink
{
    Dimension reference_dimension;
    ValueSet value_set;
};
typedef sequence<ValueSetReferenceDimensionLink> ValueSetReferenceDimensionLinkSet;

interface ValueSetReferenceDimension : Reflective::RefAssociation
{
    ValueSetReferenceDimensionLinkSet all_value_set_reference_dimension_links()
        raises (Reflective::MofError);
    boolean exists (
        in Dimension reference_dimension,
        in ValueSet value_set)
        raises (Reflective::MofError);
    Dimension reference_dimension (in ValueSet value_set)
        raises (Reflective::MofError);
    ValueSetSet value_set (in Dimension reference_dimension)
        raises (Reflective::MofError);
    void add (
        in Dimension reference_dimension,
        in ValueSet value_set)
        raises (Reflective::MofError);
    void modify_reference_dimension (
        in Dimension reference_dimension,
        in ValueSet value_set,
        in Dimension new_reference_dimension)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_value_set (
        in Dimension reference_dimension,
        in ValueSet value_set,
        in ValueSet new_value_set)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Dimension reference_dimension,
        in ValueSet value_set)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface ValueSetReferenceDimension

struct SimpleDimensionDataTypeLink
{
    Foundation::Core::Classifier data_type;
    SimpleDimension simple_dimension;
};

```

```

typedef sequence<SimpleDimensionDataTypeLink> SimpleDimensionDataTypeLinkSet;

interface SimpleDimensionDataType : Reflective::RefAssociation
{
    SimpleDimensionDataTypeLinkSet all_simple_dimension_data_type_links()
        raises (Reflective::MofError);
    boolean exists (
        in Foundation::Core::Classifier data_type,
        in SimpleDimension simple_dimension)
        raises (Reflective::MofError);
    Foundation::Core::Classifier data_type (in SimpleDimension simple_dimension)
        raises (Reflective::MofError);
    SimpleDimensionSet simple_dimension (in Foundation::Core::Classifier data_type)
        raises (Reflective::MofError);
    void add (
        in Foundation::Core::Classifier data_type,
        in SimpleDimension simple_dimension)
        raises (Reflective::MofError);
    void modify_data_type (
        in Foundation::Core::Classifier data_type,
        in SimpleDimension simple_dimension,
        in Foundation::Core::Classifier new_data_type)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_simple_dimension (
        in Foundation::Core::Classifier data_type,
        in SimpleDimension simple_dimension,
        in SimpleDimension new_simple_dimension)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Foundation::Core::Classifier data_type,
        in SimpleDimension simple_dimension)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SimpleDimensionDataType

struct AggMapComponentRelationLink
{
    Express::Relation relation;
    AggMapComponent agg_map_component;
};
typedef sequence<AggMapComponentRelationLink> AggMapComponentRelationLinkSet;

interface AggMapComponentRelation : Reflective::RefAssociation
{
    AggMapComponentRelationLinkSet all_agg_map_component_relation_links()
        raises (Reflective::MofError);
    boolean exists (
        in Express::Relation relation,
        in AggMapComponent agg_map_component)
        raises (Reflective::MofError);
    Express::Relation relation (in AggMapComponent agg_map_component)
        raises (Reflective::MofError);
    AggMapComponentSet agg_map_component (in Express::Relation relation)
        raises (Reflective::MofError);
    void add (
        in Express::Relation relation,
        in AggMapComponent agg_map_component)
        raises (Reflective::MofError);
    void modify_relation (
        in Express::Relation relation,
        in AggMapComponent agg_map_component,
        in Express::Relation new_relation)

```

```

        raises (Reflective::NotFound, Reflective::MofError);
void modify_agg_map_component (
    in Express::Relation relation,
    in AggMapComponent agg_map_component,
    in AggMapComponent new_agg_map_component)
    raises (Reflective::NotFound, Reflective::MofError);
void remove (
    in Express::Relation relation,
    in AggMapComponent agg_map_component)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface AggMapComponentRelation

struct AggMapComponentDimensionLink
{
    Express::Dimension dimension;
    AggMapComponent agg_map_component;
};
typedef sequence<AggMapComponentDimensionLink> AggMapComponentDimensionLinkSet;

interface AggMapComponentDimension : Reflective::RefAssociation
{
    AggMapComponentDimensionLinkSet all_agg_map_component_dimension_links()
        raises (Reflective::MofError);
    boolean exists (
        in Express::Dimension dimension,
        in AggMapComponent agg_map_component)
        raises (Reflective::MofError);
    Express::Dimension dimension (in AggMapComponent agg_map_component)
        raises (Reflective::MofError);
    AggMapComponentSet agg_map_component (in Express::Dimension dimension)
        raises (Reflective::MofError);
    void add (
        in Express::Dimension dimension,
        in AggMapComponent agg_map_component)
        raises (Reflective::MofError);
    void modify_dimension (
        in Express::Dimension dimension,
        in AggMapComponent agg_map_component,
        in Express::Dimension new_dimension)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_agg_map_component (
        in Express::Dimension dimension,
        in AggMapComponent agg_map_component,
        in AggMapComponent new_agg_map_component)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in Express::Dimension dimension,
        in AggMapComponent agg_map_component)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface AggMapComponentDimension

interface ExpressPackageFactory
{
    ExpressPackage create_express_package ()
        raises (Reflective::MofError);
};

interface ExpressPackage : Reflective::RefPackage
{
    readonly attribute DimensionClass dimension_ref;
    readonly attribute DatabaseClass database_ref;
};

```

```

    readonly attribute ConjointClass conjoint_ref;
    readonly attribute ProgramClass program_ref;
    readonly attribute ModelClass model_ref;
    readonly attribute VariableClass variable_ref;
    readonly attribute FormulaClass formula_ref;
    readonly attribute ValueSetClass value_set_ref;
    readonly attribute RelationClass relation_ref;
    readonly attribute WorksheetClass worksheet_ref;
    readonly attribute CompositeClass composite_ref;
    readonly attribute SimpleDimensionClass simple_dimension_ref;
    readonly attribute AliasDimensionClass alias_dimension_ref;
    readonly attribute AggMapClass agg_map_ref;
    readonly attribute AggMapComponentClass agg_map_component_ref;
    readonly attribute PreComputeClauseClass pre_compute_clause_ref;
    readonly attribute RelationReferenceDimension relation_reference_dimension_ref;
    readonly attribute WorksheetColumnDimension worksheet_column_dimension_ref;
    readonly attribute WorksheetRowDimension worksheet_row_dimension_ref;
    readonly attribute ValueSetReferenceDimension value_set_reference_dimension_ref;
    readonly attribute SimpleDimensionDataType simple_dimension_data_type_ref;
    readonly attribute AggMapComponentRelation agg_map_component_relation_ref;
    readonly attribute AggMapComponentDimension agg_map_component_dimension_ref;
};
};

#endif

```

2.7 *IMSDatabase.idl*

```
#ifndef IMSDATABASE_IDL
#define IMSDATABASE_IDL
#pragma prefix "org.omg.cwmx.resource"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Record.idl"
#include "ModelManagement.idl"

module IMSDatabase
{
    interface DBDClass;
    interface DBD;
    typedef sequence<DBD> DBDSet;
    interface PSBClass;
    interface PSB;
    typedef sequence<PSB> PSBSet;
    interface PCBClass;
    interface PCB;
    typedef sequence<PCB> PCBSet;
    interface SegmentClass;
    interface Segment;
    typedef sequence<Segment> SegmentSet;
    typedef sequence<Segment> SegmentUList;
    interface SegmentComplexClass;
    interface SegmentComplex;
    typedef sequence<SegmentComplex> SegmentComplexSet;
    interface SegmentLogicalClass;
    interface SegmentLogical;
    typedef sequence<SegmentLogical> SegmentLogicalSet;
    interface ImsFieldClass;
    interface ImsField;
    typedef sequence<ImsField> ImsFieldSet;
    interface DatasetClass;
    interface Dataset;
    typedef sequence<Dataset> DatasetSet;
    interface SenSegmentClass;
    interface SenSegment;
    typedef sequence<SenSegment> SenSegmentSet;
    interface SenFieldClass;
    interface SenField;
    typedef sequence<SenField> SenFieldSet;
    interface ACBLIBClass;
    interface ACBLIB;
    typedef sequence<ACBLIB> ACBLIBSet;
    interface AccessMethodClass;
    interface AccessMethod;
    typedef sequence<AccessMethod> AccessMethodSet;
    interface INDEXClass;
    interface INDEX;
    typedef sequence<INDEX> INDEXSet;
    interface HIDAMClass;
    interface HIDAM;
    typedef sequence<HIDAM> HIDAMSet;
    interface DEDBClass;
    interface DEDB;
    typedef sequence<DEDB> DEDBSet;
```

```

interface HDAMClass;
interface HDAM;
typedef sequence<HDAM> HDAMSet;
interface MSDBClass;
interface MSDB;
typedef sequence<MSDB> MSDBSet;
interface SecondaryIndexClass;
interface SecondaryIndex;
typedef sequence<SecondaryIndex> SecondaryIndexSet;
interface ExitClass;
interface Exit;
typedef sequence<Exit> ExitSet;
interface LCHILDClass;
interface LCHILD;
typedef sequence<LCHILD> LCHILDSet;
interface PSBLibClass;
interface PSBLib;
typedef sequence<PSBLib> PSBLibSet;
interface DBDLibClass;
interface DBDLib;
typedef sequence<DBDLib> DBDLibSet;
interface IMSDatabasePackage;

module IMSTypes
{
    interface IMSTypesPackage;
    enum AccessMethodType {imsam_DEDB, imsam_GSAM, imsam_HDAM, imsam_HIDAM,
                           imsam_HISAM, imsam_HSAM, imsam_INDEX,
                           imsam_LOGICAL, imsam_MSDB, imsam_PSINDEX,
                           imsam_PHDAM, imsam_PHIDAM, imsam_SHSAM,
                           imsam_SHISAM};
    enum PSBLanguageType {imslt_ASSEM, imslt_C, imslt_COBOL, imslt_PLI, imslt_PASCAL};
    enum PCBType {imspt_DB, imspt_GSAM, imspt_TP};
    enum PositioningType {imsp_Single, imsp_Multiple};
    enum LTermType {imstp_LTERM, imstp_NAME};
    enum RulesType {imsrt_FIRST, imsrt_LAST, imsrt_HERE};
    enum ChildPointerType {imscp_SINGL, imscp_DBLE, imscp_NONE};
    enum FlagsType {imsft_P, imsft_L, imsft_V, imdft_B};
    enum PointerType {imspn_NOTWIN, imspn_TWIN, imspn_HIER, imspn_TWINBWD,
                     imspn_HIERBWD};
    enum DeviceType {imsdt_2305, imsd_2319, imsd_3330, imsd_3350, imsd_3375, imsd_3380,
                     imsd_3390, imsd_2400, imsd_3400, imsd_TAPE};
    enum ModelType {imsdt_1, imsd_2, imsd_11};
    enum RECFMType {imsrf_F, imsr_F, imsr_V, imsr_VB, imsr_U};
    enum AlgorithmType {imsat_0, imsat_1, imsat_2};
    enum LPointerType {imslp_LTWIN, imslp_LTWINBWD};
    enum MSDBType {imsmt_NO, imsmt_TERM, imsmt_FIXED, imsmt_DYNAMIC};
    enum ParentType {ims_VIRTUAL, ims_PHYSICAL};

    interface IMSTypesPackage : Reflective::RefPackage
    {
    };
}; // end of module IMSTypes

interface DBDClass : ::Record::RecordFileClass
{
    readonly attribute DBDSet all_of_type_dbd;
    readonly attribute DBDSet all_of_class_dbd;
    DBD create_dbd (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,

```

```

        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_self_describing,
        in Foundation::DataTypes::Integer skip_records,
        in IMSTypes::AccessMethodType dli_access,
        in boolean is_vsam,
        in boolean password_flag,
        in string version_string)
        raises (Reflective::MofError);
};

interface DBD : DBDClass, ::Record::RecordFile
{
    IMSTypes::AccessMethodType dli_access ()
        raises (Reflective::MofError);
    void set_dli_access (in IMSTypes::AccessMethodType new_value)
        raises (Reflective::MofError);
    boolean is_vsam ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_is_vsam (in boolean new_value)
        raises (Reflective::MofError);
    void unset_is_vsam ()
        raises (Reflective::MofError);
    boolean password_flag ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_password_flag (in boolean new_value)
        raises (Reflective::MofError);
    void unset_password_flag ()
        raises (Reflective::MofError);
    string version_string ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_version_string (in string new_value)
        raises (Reflective::MofError);
    void unset_version_string ()
        raises (Reflective::MofError);
    DatasetSet dataset ()
        raises (Reflective::MofError);
    void set_dataset (in DatasetSet new_value)
        raises (Reflective::MofError);
    void unset_dataset ()
        raises (Reflective::MofError);
    void add_dataset (in IMSDatabase::Dataset new_element)
        raises (Reflective::MofError);
    void modify_dataset (
        in IMSDatabase::Dataset old_element,
        in IMSDatabase::Dataset new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_dataset (in IMSDatabase::Dataset old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    SegmentUList segment ()
        raises (Reflective::MofError);
    void set_segment (in SegmentUList new_value)
        raises (Reflective::MofError);
    void unset_segment ()
        raises (Reflective::MofError);
    void add_segment (in IMSDatabase::Segment new_element)
        raises (Reflective::MofError);
    void add_segment_before (
        in IMSDatabase::Segment new_element,

```

```

    in IMSDatabase::Segment before_element)
    raises (Reflective::NotFound, Reflective::MofError);
void modify_segment (
    in IMSDatabase::Segment old_element,
    in IMSDatabase::Segment new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_segment (in IMSDatabase::Segment old_element)
    raises (Reflective::NotFound, Reflective::MofError);
ExitSet exit ()
    raises (Reflective::MofError);
void set_exit (in ExitSet new_value)
    raises (Reflective::MofError);
void unset_exit ()
    raises (Reflective::MofError);
void add_exit (in IMSDatabase::Exit new_element)
    raises (Reflective::MofError);
void modify_exit (
    in IMSDatabase::Exit old_element,
    in IMSDatabase::Exit new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_exit (in IMSDatabase::Exit old_element)
    raises (Reflective::NotFound, Reflective::MofError);
AccessMethod access_method ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_access_method (in AccessMethod new_value)
    raises (Reflective::MofError);
void unset_access_method ()
    raises (Reflective::MofError);
ACBLIBSet acblib ()
    raises (Reflective::MofError);
void set_acblib (in ACBLIBSet new_value)
    raises (Reflective::MofError);
void unset_acblib ()
    raises (Reflective::MofError);
void add_acblib (in IMSDatabase::ACBLIB new_element)
    raises (Reflective::MofError);
void modify_acblib (
    in IMSDatabase::ACBLIB old_element,
    in IMSDatabase::ACBLIB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_acblib (in IMSDatabase::ACBLIB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
DBDLIBSet library ()
    raises (Reflective::MofError);
void set_library (in DBDLIBSet new_value)
    raises (Reflective::MofError);
void unset_library ()
    raises (Reflective::MofError);
void add_library (in DBDLIB new_element)
    raises (Reflective::MofError);
void modify_library (
    in DBDLIB old_element,
    in DBDLIB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_library (in DBDLIB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
PCBSet pcb ()
    raises (Reflective::MofError);
void set_pcb (in PCBSet new_value)
    raises (Reflective::MofError);
void unset_pcb ()

```



```

        raises (Reflective::MofError);
void add_pcb (in IMSDatabase::PCB new_element)
    raises (Reflective::MofError);
void modify_pcb (
    in IMSDatabase::PCB old_element,
    in IMSDatabase::PCB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_pcb (in IMSDatabase::PCB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface DBD

interface PSBClass : ::Record::RecordFileClass
{
    readonly attribute PSBSet all_of_type_psb;
    readonly attribute PSBSet all_of_class_psb;
    PSB create_psb (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_self_describing,
        in Foundation::DataTypes::Integer skip_records,
        in boolean compatibility,
        in Foundation::DataTypes::Integer io_error_option,
        in Foundation::DataTypes::Integer ioa_size,
        in IMSTypes::PSBLanguageType language,
        in Foundation::DataTypes::Integer lock_maximum,
        in Foundation::DataTypes::Integer maximum_qx_calls,
        in boolean online_image_copy,
        in Foundation::DataTypes::Integer ssa_size,
        in boolean write_to_operator)
        raises (Reflective::MofError);
};

interface PSB : PSBClass, ::Record::RecordFile
{
    boolean compatibility ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_compatibility (in boolean new_value)
        raises (Reflective::MofError);
    void unset_compatibility ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer io_error_option ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_io_error_option (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_io_error_option ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer ioa_size ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_ioa_size (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_ioa_size ()
        raises (Reflective::MofError);
    IMSTypes::PSBLanguageType language ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_language (in IMSTypes::PSBLanguageType new_value)
        raises (Reflective::MofError);
    void unset_language ()

```

```

    raises (Reflective::MofError);
Foundation::DataTypes::Integer lock_maximum ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_lock_maximum (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_lock_maximum ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer maximum_qx_calls ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_maximum_qx_calls (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_maximum_qx_calls ()
    raises (Reflective::MofError);
boolean online_image_copy ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_online_image_copy (in boolean new_value)
    raises (Reflective::MofError);
void unset_online_image_copy ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer ssa_size ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_ssa_size (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_ssa_size ()
    raises (Reflective::MofError);
boolean write_to_operator ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_write_to_operator (in boolean new_value)
    raises (Reflective::MofError);
void unset_write_to_operator ()
    raises (Reflective::MofError);
PCBSet pcb ()
    raises (Reflective::MofError);
void set_pcb (in PCBSet new_value)
    raises (Reflective::MofError);
void unset_pcb ()
    raises (Reflective::MofError);
void add_pcb (in IMSDatabase::PCB new_element)
    raises (Reflective::MofError);
void modify_pcb (
    in IMSDatabase::PCB old_element,
    in IMSDatabase::PCB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_pcb (in IMSDatabase::PCB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
PSBLibSet library ()
    raises (Reflective::MofError);
void set_library (in PSBLibSet new_value)
    raises (Reflective::MofError);
void unset_library ()
    raises (Reflective::MofError);
void add_library (in PSBLib new_element)
    raises (Reflective::MofError);
void modify_library (
    in PSBLib old_element,
    in PSBLib new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_library (in PSBLib old_element)
    raises (Reflective::NotFound, Reflective::MofError);
ACBLIBSet acbllib ()
    raises (Reflective::MofError);

```

```

void set_acblib (in ACBLIBSet new_value)
    raises (Reflective::MofError);
void unset_acblib ()
    raises (Reflective::MofError);
void add_acblib (in IMSDatabase::ACBLIB new_element)
    raises (Reflective::MofError);
void modify_acblib (
    in IMSDatabase::ACBLIB old_element,
    in IMSDatabase::ACBLIB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_acblib (in IMSDatabase::ACBLIB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface PSB

```

```

interface PCBClass : ::Record::RecordFileClass
{
    readonly attribute PCBSet all_of_type_pcb;
    readonly attribute PCBSet all_of_class_pcb;
    PCB create_pcb (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_self_describing,
        in Foundation::DataTypes::Integer skip_records,
        in IMSTypes::PCBType pcb_type,
        in boolean list,
        in Foundation::DataTypes::Integer key_length,
        in string processing_options,
        in IMSTypes::PositioningType positioning,
        in boolean sequential_buffering,
        in boolean alternate_response,
        in boolean express,
        in boolean modify,
        in boolean same_terminal,
        in IMSTypes::LTermType destination_type,
        in string lterm_name)
        raises (Reflective::MofError);
};

```

```

interface PCB : PCBClass, ::Record::RecordFile
{
    IMSTypes::PCBType pcb_type ()
        raises (Reflective::MofError);
    void set_pcb_type (in IMSTypes::PCBType new_value)
        raises (Reflective::MofError);
    boolean list ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_list (in boolean new_value)
        raises (Reflective::MofError);
    void unset_list ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer key_length ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_key_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_key_length ()
        raises (Reflective::MofError);
    string processing_options ()

```

```

    raises (Reflective::NotSet, Reflective::MofError);
void set_processing_options (in string new_value)
    raises (Reflective::MofError);
void unset_processing_options ()
    raises (Reflective::MofError);
IMSTypes::PositioningType positioning ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_positioning (in IMSTypes::PositioningType new_value)
    raises (Reflective::MofError);
void unset_positioning ()
    raises (Reflective::MofError);
boolean sequential_buffering ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_sequential_buffering (in boolean new_value)
    raises (Reflective::MofError);
void unset_sequential_buffering ()
    raises (Reflective::MofError);
boolean alternate_response ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_alternate_response (in boolean new_value)
    raises (Reflective::MofError);
void unset_alternate_response ()
    raises (Reflective::MofError);
boolean express ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_express (in boolean new_value)
    raises (Reflective::MofError);
void unset_express ()
    raises (Reflective::MofError);
boolean modify ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_modify (in boolean new_value)
    raises (Reflective::MofError);
void unset_modify ()
    raises (Reflective::MofError);
boolean same_terminal ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_same_terminal (in boolean new_value)
    raises (Reflective::MofError);
void unset_same_terminal ()
    raises (Reflective::MofError);
IMSTypes::LTermType destination_type ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_destination_type (in IMSTypes::LTermType new_value)
    raises (Reflective::MofError);
void unset_destination_type ()
    raises (Reflective::MofError);
string lterm_name ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_lterm_name (in string new_value)
    raises (Reflective::MofError);
void unset_lterm_name ()
    raises (Reflective::MofError);
INDEX proc_seq ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_proc_seq (in INDEX new_value)
    raises (Reflective::MofError);
void unset_proc_seq ()
    raises (Reflective::MofError);
SenSegmentSet sen_segment ()
    raises (Reflective::MofError);

```

```

void set_sen_segment (in SenSegmentSet new_value)
    raises (Reflective::MofError);
void unset_sen_segment ()
    raises (Reflective::MofError);
void add_sen_segment (in SenSegment new_element)
    raises (Reflective::MofError);
void modify_sen_segment (
    in SenSegment old_element,
    in SenSegment new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_sen_segment (in SenSegment old_element)
    raises (Reflective::NotFound, Reflective::MofError);
PSBSet psb ()
    raises (Reflective::MofError);
void set_psb (in PSBSet new_value)
    raises (Reflective::MofError);
void unset_psb ()
    raises (Reflective::MofError);
void add_psb (in IMSDatabase::PSB new_element)
    raises (Reflective::MofError);
void modify_psb (
    in IMSDatabase::PSB old_element,
    in IMSDatabase::PSB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_psb (in IMSDatabase::PSB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
IMSDatabase::DBD dbd ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_dbd (in IMSDatabase::DBD new_value)
    raises (Reflective::MofError);
void unset_dbd ()
    raises (Reflective::MofError);
}; // end of interface PCB

```

```

interface SegmentClass : Record::RecordDefClass
{
    readonly attribute SegmentSet all_of_type_segment;
    readonly attribute SegmentSet all_of_class_segment;
    Segment create_segment (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in string field_delimiter,
        in boolean is_fixed_width,
        in string record_delimiter,
        in string text_delimiter,
        in boolean exit_flag,
        in string frequency,
        in Foundation::DataTypes::Integer maximum_length,
        in Foundation::DataTypes::Integer minimum_length,
        in IMSTypes::RulesType rules,
        in string subset_pointers,
        in boolean direct_dependent,
        in IMSTypes::ChildPointerType pc_pointer)
        raises (Reflective::MofError);
};

```

```

interface Segment : SegmentClass, Record::RecordDef
{
    boolean exit_flag ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_exit_flag (in boolean new_value)
        raises (Reflective::MofError);
    void unset_exit_flag ()
        raises (Reflective::MofError);
    string frequency ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_frequency (in string new_value)
        raises (Reflective::MofError);
    void unset_frequency ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer maximum_length ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_maximum_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_maximum_length ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer minimum_length ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_minimum_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_minimum_length ()
        raises (Reflective::MofError);
    IMSTypes::RulesType rules ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_rules (in IMSTypes::RulesType new_value)
        raises (Reflective::MofError);
    void unset_rules ()
        raises (Reflective::MofError);
    string subset_pointers ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_subset_pointers (in string new_value)
        raises (Reflective::MofError);
    void unset_subset_pointers ()
        raises (Reflective::MofError);
    boolean direct_dependent ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_direct_dependent (in boolean new_value)
        raises (Reflective::MofError);
    void unset_direct_dependent ()
        raises (Reflective::MofError);
    IMSTypes::ChildPointerType pc_pointer ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_pc_pointer (in IMSTypes::ChildPointerType new_value)
        raises (Reflective::MofError);
    void unset_pc_pointer ()
        raises (Reflective::MofError);
    IMSDatabase::DBD dbd ()
        raises (Reflective::MofError);
    void set_dbd (in IMSDatabase::DBD new_value)
        raises (Reflective::MofError);
    ExitSet exit ()
        raises (Reflective::MofError);
    void set_exit (in ExitSet new_value)
        raises (Reflective::MofError);
    void unset_exit ()
        raises (Reflective::MofError);
    void add_exit (in IMSDatabase::Exit new_element)

```

```

        raises (Reflective::MofError);
void modify_exit (
    in IMSDatabase::Exit old_element,
    in IMSDatabase::Exit new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_exit (in IMSDatabase::Exit old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SegmentSet child ()
    raises (Reflective::MofError);
void set_child (in SegmentSet new_value)
    raises (Reflective::MofError);
void unset_child ()
    raises (Reflective::MofError);
void add_child (in Segment new_element)
    raises (Reflective::MofError);
void modify_child (
    in Segment old_element,
    in Segment new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_child (in Segment old_element)
    raises (Reflective::NotFound, Reflective::MofError);
Segment parent ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_parent (in Segment new_value)
    raises (Reflective::MofError);
void unset_parent ()
    raises (Reflective::MofError);
SegmentLogicalSet logical ()
    raises (Reflective::MofError);
void set_logical (in SegmentLogicalSet new_value)
    raises (Reflective::MofError);
void unset_logical ()
    raises (Reflective::MofError);
void add_logical (in SegmentLogical new_element)
    raises (Reflective::MofError);
void modify_logical (
    in SegmentLogical old_element,
    in SegmentLogical new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_logical (in SegmentLogical old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SenSegmentSet senseg ()
    raises (Reflective::MofError);
void set_senseg (in SenSegmentSet new_value)
    raises (Reflective::MofError);
void unset_senseg ()
    raises (Reflective::MofError);
void add_senseg (in SenSegment new_element)
    raises (Reflective::MofError);
void modify_senseg (
    in SenSegment old_element,
    in SenSegment new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_senseg (in SenSegment old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface Segment

interface SegmentComplexClass : SegmentClass
{
    readonly attribute SegmentComplexSet all_of_type_segment_complex;
    readonly attribute SegmentComplexSet all_of_class_segment_complex;

```

```

SegmentComplex create_segment_complex (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract,
    in boolean is_active,
    in string field_delimiter,
    in boolean is_fixed_width,
    in string record_delimiter,
    in string text_delimiter,
    in boolean exit_flag,
    in string frequency,
    in Foundation::DataTypes::Integer maximum_length,
    in Foundation::DataTypes::Integer minimum_length,
    in IMSTypes::RulesType rules,
    in string subset_pointers,
    in boolean direct_dependent,
    in IMSTypes::ChildPointerType pc_pointer,
    in IMSTypes::FlagsType delete_flag,
    in IMSTypes::FlagsType insert_flag,
    in IMSTypes::FlagsType replace_flag,
    in IMSTypes::PointerType segm_pointer,
    in string ds_group)
    raises (Reflective::MofError);
};

interface SegmentComplex : SegmentComplexClass, Segment
{
    IMSTypes::FlagsType delete_flag ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_delete_flag (in IMSTypes::FlagsType new_value)
        raises (Reflective::MofError);
    void unset_delete_flag ()
        raises (Reflective::MofError);
    IMSTypes::FlagsType insert_flag ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_insert_flag (in IMSTypes::FlagsType new_value)
        raises (Reflective::MofError);
    void unset_insert_flag ()
        raises (Reflective::MofError);
    IMSTypes::FlagsType replace_flag ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_replace_flag (in IMSTypes::FlagsType new_value)
        raises (Reflective::MofError);
    void unset_replace_flag ()
        raises (Reflective::MofError);
    IMSTypes::PointerType segm_pointer ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_segm_pointer (in IMSTypes::PointerType new_value)
        raises (Reflective::MofError);
    void unset_segm_pointer ()
        raises (Reflective::MofError);
    string ds_group ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_ds_group (in string new_value)
        raises (Reflective::MofError);
    void unset_ds_group ()
        raises (Reflective::MofError);
    IMSDatabase::LCHILD lparent ()

```



```

    raises (Reflective::NotSet, Reflective::MofError);
void set_lparent (in IMSDatabase::LCHILD new_value)
    raises (Reflective::MofError);
void unset_lparent ()
    raises (Reflective::MofError);
LCHILDSet lchild ()
    raises (Reflective::MofError);
void set_lchild (in LCHILDSet new_value)
    raises (Reflective::MofError);
void unset_lchild ()
    raises (Reflective::MofError);
void add_lchild (in IMSDatabase::LCHILD new_element)
    raises (Reflective::MofError);
void modify_lchild (
    in IMSDatabase::LCHILD old_element,
    in IMSDatabase::LCHILD new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_lchild (in IMSDatabase::LCHILD old_element)
    raises (Reflective::NotFound, Reflective::MofError);
IMSDatabase::LCHILD paired_lchild ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_paired_lchild (in IMSDatabase::LCHILD new_value)
    raises (Reflective::MofError);
void unset_paired_lchild ()
    raises (Reflective::MofError);
SecondaryIndexSet secondary_index ()
    raises (Reflective::MofError);
void set_secondary_index (in SecondaryIndexSet new_value)
    raises (Reflective::MofError);
void unset_secondary_index ()
    raises (Reflective::MofError);
void add_secondary_index (in SecondaryIndex new_element)
    raises (Reflective::MofError);
void modify_secondary_index (
    in SecondaryIndex old_element,
    in SecondaryIndex new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_secondary_index (in SecondaryIndex old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SecondaryIndexSet sourced_index ()
    raises (Reflective::MofError);
void set_sourced_index (in SecondaryIndexSet new_value)
    raises (Reflective::MofError);
void unset_sourced_index ()
    raises (Reflective::MofError);
void add_sourced_index (in SecondaryIndex new_element)
    raises (Reflective::MofError);
void modify_sourced_index (
    in SecondaryIndex old_element,
    in SecondaryIndex new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_sourced_index (in SecondaryIndex old_element)
    raises (Reflective::NotFound, Reflective::MofError);
IMSDatabase::Dataset dataset ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_dataset (in IMSDatabase::Dataset new_value)
    raises (Reflective::MofError);
void unset_dataset ()
    raises (Reflective::MofError);
}; // end of interface SegmentComplex

```

```

interface SegmentLogicalClass : SegmentClass
{
    readonly attribute SegmentLogicalSet all_of_type_segment_logical;
    readonly attribute SegmentLogicalSet all_of_class_segment_logical;
    SegmentLogical create_segment_logical (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in string field_delimiter,
        in boolean is_fixed_width,
        in string record_delimiter,
        in string text_delimiter,
        in boolean exit_flag,
        in string frequency,
        in Foundation::DataTypes::Integer maximum_length,
        in Foundation::DataTypes::Integer minimum_length,
        in IMSTypes::RulesType rules,
        in string subset_pointers,
        in boolean direct_dependent,
        in IMSTypes::ChildPointerType pc_pointer,
        in boolean key_data1,
        in boolean key_data2)
        raises (Reflective::MofError);
};

interface SegmentLogical : SegmentLogicalClass, Segment
{
    boolean key_data1 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_key_data1 (in boolean new_value)
        raises (Reflective::MofError);
    void unset_key_data1 ()
        raises (Reflective::MofError);
    boolean key_data2 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_key_data2 (in boolean new_value)
        raises (Reflective::MofError);
    void unset_key_data2 ()
        raises (Reflective::MofError);
    SegmentSet physical ()
        raises (Reflective::MofError);
    void set_physical (in SegmentSet new_value)
        raises (Reflective::MofError);
    void unset_physical ()
        raises (Reflective::MofError);
    void add_physical (in Segment new_element)
        raises (Reflective::MofError);
    void modify_physical (
        in Segment old_element,
        in Segment new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_physical (in Segment old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SegmentLogical

interface ImsFieldClass : Record::FixedOffsetFieldClass
{

```

```

readonly attribute ImsFieldSet all_of_type_ims_field;
readonly attribute ImsFieldSet all_of_class_ims_field;
ImsField create_ims_field (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in Foundation::DataTypes::ScopeKind owner_scope,
    in Foundation::DataTypes::Multiplicity multiplicity,
    in Foundation::DataTypes::ChangeableKind changeability,
    in Foundation::DataTypes::ScopeKind target_scope,
    in Foundation::DataTypes::Expression initial_value,
    in Foundation::DataTypes::Integer offset,
    in Foundation::DataTypes::Integer offset_unit_bits,
    in boolean sequence_field,
    in boolean unique_sequence,
    in Foundation::DataTypes::Integer length,
    in boolean generated)
    raises (Reflective::MofError);
};

interface ImsField : ImsFieldClass, Record::FixedOffsetField
{
    boolean sequence_field ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_sequence_field (in boolean new_value)
        raises (Reflective::MofError);
    void unset_sequence_field ()
        raises (Reflective::MofError);
    boolean unique_sequence ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_unique_sequence (in boolean new_value)
        raises (Reflective::MofError);
    void unset_unique_sequence ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer length ()
        raises (Reflective::MofError);
    void set_length (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    boolean generated ()
        raises (Reflective::MofError);
    void set_generated (in boolean new_value)
        raises (Reflective::MofError);
    SecondaryIndexSet ddata_index ()
        raises (Reflective::MofError);
    void set_ddata_index (in SecondaryIndexSet new_value)
        raises (Reflective::MofError);
    void unset_ddata_index ()
        raises (Reflective::MofError);
    void add_ddata_index (in SecondaryIndex new_element)
        raises (Reflective::MofError);
    void modify_ddata_index (
        in SecondaryIndex old_element,
        in SecondaryIndex new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_ddata_index (in SecondaryIndex old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    SecondaryIndexSet search_index ()
        raises (Reflective::MofError);
    void set_search_index (in SecondaryIndexSet new_value)
        raises (Reflective::MofError);
    void unset_search_index ()

```

```

        raises (Reflective::MofError);
void add_search_index (in SecondaryIndex new_element)
    raises (Reflective::MofError);
void modify_search_index (
    in SecondaryIndex old_element,
    in SecondaryIndex new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_search_index (in SecondaryIndex old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SecondaryIndexSet subseq_index ()
    raises (Reflective::MofError);
void set_subseq_index (in SecondaryIndexSet new_value)
    raises (Reflective::MofError);
void unset_subseq_index ()
    raises (Reflective::MofError);
void add_subseq_index (in SecondaryIndex new_element)
    raises (Reflective::MofError);
void modify_subseq_index (
    in SecondaryIndex old_element,
    in SecondaryIndex new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_subseq_index (in SecondaryIndex old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SenFieldSet sen_field ()
    raises (Reflective::MofError);
void set_sen_field (in SenFieldSet new_value)
    raises (Reflective::MofError);
void unset_sen_field ()
    raises (Reflective::MofError);
void add_sen_field (in SenField new_element)
    raises (Reflective::MofError);
void modify_sen_field (
    in SenField old_element,
    in SenField new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_sen_field (in SenField old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface ImsField

```

```

interface DatasetClass : Foundation::Core::ModelElementClass
{
    readonly attribute DatasetSet all_of_type_dataset;
    readonly attribute DatasetSet all_of_class_dataset;
    Dataset create_dataset (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string dd1name,
        in IMSTypes::DeviceType device,
        in IMSTypes::ModelType model,
        in string dd2name,
        in Foundation::DataTypes::Integer size1,
        in Foundation::DataTypes::Integer size2,
        in Foundation::DataTypes::Integer record_length1,
        in Foundation::DataTypes::Integer record_length2,
        in Foundation::DataTypes::Integer blocking_factor1,
        in Foundation::DataTypes::Integer blocking_factor2,
        in string dataset_label,
        in Foundation::DataTypes::Integer free_block_frequency,
        in Foundation::DataTypes::Integer free_space_percentage,
        in IMSTypes::RECFMType record_format,

```

```

        in Foundation::DataTypes::Integer scan_cylinders,
        in IMSTypes::AlgorithmType search_algorithm,
        in Foundation::DataTypes::Integer root,
        in Foundation::DataTypes::Integer root_overflow,
        in Foundation::DataTypes::Integer uow,
        in Foundation::DataTypes::Integer uow_overflow)
        raises (Reflective::MofError);
};

```

```

interface Dataset : DatasetClass, Foundation::Core::ModelElement
{
    string dd1name ()
        raises (Reflective::MofError);
    void set_dd1name (in string new_value)
        raises (Reflective::MofError);
    IMSTypes::DeviceType device ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_device (in IMSTypes::DeviceType new_value)
        raises (Reflective::MofError);
    void unset_device ()
        raises (Reflective::MofError);
    IMSTypes::ModelType model ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_model (in IMSTypes::ModelType new_value)
        raises (Reflective::MofError);
    void unset_model ()
        raises (Reflective::MofError);
    string dd2name ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_dd2name (in string new_value)
        raises (Reflective::MofError);
    void unset_dd2name ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer size1 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_size1 (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_size1 ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer size2 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_size2 (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_size2 ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer record_length1 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_record_length1 (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_record_length1 ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer record_length2 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_record_length2 (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_record_length2 ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer blocking_factor1 ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_blocking_factor1 (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
}

```

```

void unset_blocking_factor1 ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer blocking_factor2 ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_blocking_factor2 (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_blocking_factor2 ()
    raises (Reflective::MofError);
string dataset_label ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_dataset_label (in string new_value)
    raises (Reflective::MofError);
void unset_dataset_label ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer free_block_frequency ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_free_block_frequency (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_free_block_frequency ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer free_space_percentage ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_free_space_percentage (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_free_space_percentage ()
    raises (Reflective::MofError);
IMSTypes::RECFMType record_format ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_record_format (in IMSTypes::RECFMType new_value)
    raises (Reflective::MofError);
void unset_record_format ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer scan_cylinders ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_scan_cylinders (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_scan_cylinders ()
    raises (Reflective::MofError);
IMSTypes::AlgorithmType search_algorithm ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_search_algorithm (in IMSTypes::AlgorithmType new_value)
    raises (Reflective::MofError);
void unset_search_algorithm ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer root ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_root (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_root ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer root_overflow ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_root_overflow (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_root_overflow ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer uow ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_uow (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_uow ()

```

```

        raises (Reflective::MofError);
Foundation::DataTypes::Integer uow_overflow ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_uow_overflow (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_uow_overflow ()
    raises (Reflective::MofError);
IMSDatabase::DBD dbd ()
    raises (Reflective::MofError);
void set_dbd (in IMSDatabase::DBD new_value)
    raises (Reflective::MofError);
SegmentComplexSet segment ()
    raises (Reflective::MofError);
void set_segment (in SegmentComplexSet new_value)
    raises (Reflective::MofError);
void unset_segment ()
    raises (Reflective::MofError);
void add_segment (in SegmentComplex new_element)
    raises (Reflective::MofError);
void modify_segment (
    in SegmentComplex old_element,
    in SegmentComplex new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_segment (in SegmentComplex old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface Dataset

```

```

interface SenSegmentClass : Record::RecordDefClass
{
    readonly attribute SenSegmentSet all_of_type_sen_segment;
    readonly attribute SenSegmentSet all_of_class_sen_segment;
    SenSegment create_sen_segment (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in string field_delimiter,
        in boolean is_fixed_width,
        in string record_delimiter,
        in string text_delimiter,
        in string procopt_senseg,
        in string subset_pointers)
        raises (Reflective::MofError);
};

```

```

interface SenSegment : SenSegmentClass, Record::RecordDef
{
    string procopt_senseg ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_procopt_senseg (in string new_value)
        raises (Reflective::MofError);
    void unset_procopt_senseg ()
        raises (Reflective::MofError);
    string subset_pointers ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_subset_pointers (in string new_value)
        raises (Reflective::MofError);
    void unset_subset_pointers ()

```

```

        raises (Reflective::MofError);
SenFieldSet sen_field ()
    raises (Reflective::MofError);
void set_sen_field (in SenFieldSet new_value)
    raises (Reflective::MofError);
void unset_sen_field ()
    raises (Reflective::MofError);
void add_sen_field (in SenField new_element)
    raises (Reflective::MofError);
void modify_sen_field (
    in SenField old_element,
    in SenField new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_sen_field (in SenField old_element)
    raises (Reflective::NotFound, Reflective::MofError);
IMSDatabase::Segment segment ()
    raises (Reflective::MofError);
void set_segment (in IMSDatabase::Segment new_value)
    raises (Reflective::MofError);
INDEXSet index ()
    raises (Reflective::MofError);
void set_index (in INDEXSet new_value)
    raises (Reflective::MofError);
void unset_index ()
    raises (Reflective::MofError);
void add_index (in IMSDatabase::INDEX new_element)
    raises (Reflective::MofError);
void modify_index (
    in IMSDatabase::INDEX old_element,
    in IMSDatabase::INDEX new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_index (in IMSDatabase::INDEX old_element)
    raises (Reflective::NotFound, Reflective::MofError);
IMSDatabase::PCB pcb ()
    raises (Reflective::MofError);
void set_pcb (in IMSDatabase::PCB new_value)
    raises (Reflective::MofError);
}; // end of interface SenSegment

```

```

interface SenFieldClass : Record::FixedOffsetFieldClass
{
    readonly attribute SenFieldSet all_of_type_sen_field;
    readonly attribute SenFieldSet all_of_class_sen_field;
    SenField create_sen_field (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in Foundation::DataTypes::ScopeKind owner_scope,
        in Foundation::DataTypes::Multiplicity multiplicity,
        in Foundation::DataTypes::ChangeableKind changeability,
        in Foundation::DataTypes::ScopeKind target_scope,
        in Foundation::DataTypes::Expression initial_value,
        in Foundation::DataTypes::Integer offset,
        in Foundation::DataTypes::Integer offset_unit_bits,
        in boolean replace)
        raises (Reflective::MofError);
};

```

```

interface SenField : SenFieldClass, Record::FixedOffsetField
{
    boolean replace ()

```



```

        raises (Reflective::NotSet, Reflective::MofError);
void set_replace (in boolean new_value)
    raises (Reflective::MofError);
void unset_replace ()
    raises (Reflective::MofError);
ImsField field ()
    raises (Reflective::MofError);
void set_field (in ImsField new_value)
    raises (Reflective::MofError);
SenSegment sen_segment ()
    raises (Reflective::MofError);
void set_sen_segment (in SenSegment new_value)
    raises (Reflective::MofError);
}; // end of interface SenField

interface ACBLIBClass : ModelManagement::PackageClass
{
    readonly attribute ACBLIBSet all_of_type_acblib;
    readonly attribute ACBLIBSet all_of_class_acblib;
    ACBLIB create_acblib (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface ACBLIB : ACBLIBClass, ModelManagement::Package
{
    DBDSet dbd ()
        raises (Reflective::MofError);
void set_dbd (in DBDSet new_value)
    raises (Reflective::MofError);
void unset_dbd ()
    raises (Reflective::MofError);
void add_dbd (in IMSDatabase::DBD new_element)
    raises (Reflective::MofError);
void modify_dbd (
    in IMSDatabase::DBD old_element,
    in IMSDatabase::DBD new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_dbd (in IMSDatabase::DBD old_element)
    raises (Reflective::NotFound, Reflective::MofError);
PSBSet psb ()
    raises (Reflective::MofError);
void set_psb (in PSBSet new_value)
    raises (Reflective::MofError);
void unset_psb ()
    raises (Reflective::MofError);
void add_psb (in IMSDatabase::PSB new_element)
    raises (Reflective::MofError);
void modify_psb (
    in IMSDatabase::PSB old_element,
    in IMSDatabase::PSB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_psb (in IMSDatabase::PSB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface ACBLIB

```

```

interface AccessMethodClass : Foundation::Core::ModelElementClass
{
    readonly attribute AccessMethodSet all_of_type_access_method;
    readonly attribute AccessMethodSet all_of_class_access_method;
    AccessMethod create_access_method (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);
};

interface AccessMethod : AccessMethodClass, Foundation::Core::ModelElement
{
    IMSDatabase::DBD dbd ()
        raises (Reflective::MofError);
    void set_dbd (in IMSDatabase::DBD new_value)
        raises (Reflective::MofError);
}; // end of interface AccessMethod

interface INDEXClass : AccessMethodClass
{
    readonly attribute INDEXSet all_of_type_index;
    readonly attribute INDEXSet all_of_class_index;
    INDEX create_index (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean dos_compatibility,
        in boolean protect)
        raises (Reflective::MofError);
};

interface INDEX : INDEXClass, AccessMethod
{
    boolean dos_compatibility ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_dos_compatibility (in boolean new_value)
        raises (Reflective::MofError);
    void unset_dos_compatibility ()
        raises (Reflective::MofError);
    boolean protect ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_protect (in boolean new_value)
        raises (Reflective::MofError);
    void unset_protect ()
        raises (Reflective::MofError);
    HIDAM primary_target ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_primary_target (in HIDAM new_value)
        raises (Reflective::MofError);
    void unset_primary_target ()
        raises (Reflective::MofError);
    SecondaryIndex secondary_target ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_secondary_target (in SecondaryIndex new_value)
        raises (Reflective::MofError);
    void unset_secondary_target ()
        raises (Reflective::MofError);
    INDEX shared_index ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_shared_index (in INDEX new_value)

```

```

        raises (Reflective::MofError);
void unset_shared_index ()
    raises (Reflective::MofError);
INDEXSet sharing_index ()
    raises (Reflective::MofError);
void set_sharing_index (in INDEXSet new_value)
    raises (Reflective::MofError);
void unset_sharing_index ()
    raises (Reflective::MofError);
void add_sharing_index (in INDEX new_element)
    raises (Reflective::MofError);
void modify_sharing_index (
    in INDEX old_element,
    in INDEX new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_sharing_index (in INDEX old_element)
    raises (Reflective::NotFound, Reflective::MofError);
SenSegmentSet senseg ()
    raises (Reflective::MofError);
void set_senseg (in SenSegmentSet new_value)
    raises (Reflective::MofError);
void unset_senseg ()
    raises (Reflective::MofError);
void add_senseg (in SenSegment new_element)
    raises (Reflective::MofError);
void modify_senseg (
    in SenSegment old_element,
    in SenSegment new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_senseg (in SenSegment old_element)
    raises (Reflective::NotFound, Reflective::MofError);
PCBSet sequenced_pcb ()
    raises (Reflective::MofError);
void set_sequenced_pcb (in PCBSet new_value)
    raises (Reflective::MofError);
void unset_sequenced_pcb ()
    raises (Reflective::MofError);
void add_sequenced_pcb (in PCB new_element)
    raises (Reflective::MofError);
void modify_sequenced_pcb (
    in PCB old_element,
    in PCB new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_sequenced_pcb (in PCB old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface INDEX

```

```

interface HIDAMClass : AccessMethodClass
{
    readonly attribute HIDAMSet all_of_type_hidam;
    readonly attribute HIDAMSet all_of_class_hidam;
    HIDAM create_hidam (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification)
        raises (Reflective::MofError);
};

```

```

interface HIDAM : HIDAMClass, AccessMethod
{
    IMSDatabase::INDEX index ()

```

```

        raises (Reflective::NotSet, Reflective::MofError);
void set_index (in IMSDatabase::INDEX new_value)
    raises (Reflective::MofError);
void unset_index ()
    raises (Reflective::MofError);
}; // end of interface HIDAM

interface DEDBClass : AccessMethodClass
{
    readonly attribute DEDBSet all_of_type_dedb;
    readonly attribute DEDBSet all_of_class_dedb;
    DEDB create_dedb (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string rm_name,
        in Foundation::DataTypes::Integer stage,
        in boolean extended_call)
        raises (Reflective::MofError);
};

interface DEDB : DEDBClass, AccessMethod
{
    string rm_name ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_rm_name (in string new_value)
        raises (Reflective::MofError);
    void unset_rm_name ()
        raises (Reflective::MofError);
    Foundation::DataTypes::Integer stage ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_stage (in Foundation::DataTypes::Integer new_value)
        raises (Reflective::MofError);
    void unset_stage ()
        raises (Reflective::MofError);
    boolean extended_call ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_extended_call (in boolean new_value)
        raises (Reflective::MofError);
    void unset_extended_call ()
        raises (Reflective::MofError);
}; // end of interface DEDB

interface HDAMClass : AccessMethodClass
{
    readonly attribute HDAMSet all_of_type_hdam;
    readonly attribute HDAMSet all_of_class_hdam;
    HDAM create_hdam (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string rm_name,
        in Foundation::DataTypes::Integer relative_block_number,
        in Foundation::DataTypes::Integer root_anchor_points,
        in Foundation::DataTypes::Integer root_max_bytes)
        raises (Reflective::MofError);
};

interface HDAM : HDAMClass, AccessMethod
{
    string rm_name ()

```

```

        raises (Reflective::NotSet, Reflective::MofError);
void set_rm_name (in string new_value)
    raises (Reflective::MofError);
void unset_rm_name ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer relative_block_number ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_relative_block_number (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_relative_block_number ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer root_anchor_points ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_root_anchor_points (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_root_anchor_points ()
    raises (Reflective::MofError);
Foundation::DataTypes::Integer root_max_bytes ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_root_max_bytes (in Foundation::DataTypes::Integer new_value)
    raises (Reflective::MofError);
void unset_root_max_bytes ()
    raises (Reflective::MofError);
}; // end of interface HDAM

```

```

interface MSDBClass : AccessMethodClass
{
    readonly attribute MSDBSet all_of_type_msdb;
    readonly attribute MSDBSet all_of_class_msdb;
    MSDB create_msdb (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string msdb_field,
        in IMSTypes::MSDBtype msdb_type)
        raises (Reflective::MofError);
};

```

```

interface MSDB : MSDBClass, AccessMethod
{
    string msdb_field ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_msdb_field (in string new_value)
        raises (Reflective::MofError);
    void unset_msdb_field ()
        raises (Reflective::MofError);
    IMSTypes::MSDBtype msdb_type ()
        raises (Reflective::MofError);
    void set_msdb_type (in IMSTypes::MSDBtype new_value)
        raises (Reflective::MofError);
}; // end of interface MSDB

```

```

interface SecondaryIndexClass : Foundation::Core::ModelElementClass
{
    readonly attribute SecondaryIndexSet all_of_type_secondary_index;
    readonly attribute SecondaryIndexSet all_of_class_secondary_index;
    SecondaryIndex create_secondary_index (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string constant,

```

```

        in string exit_routine,
        in string null_value)
        raises (Reflective::MofError);
};

interface SecondaryIndex : SecondaryIndexClass, Foundation::Core::ModelElement
{
    string constant ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_constant (in string new_value)
        raises (Reflective::MofError);
    void unset_constant ()
        raises (Reflective::MofError);
    string exit_routine ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_exit_routine (in string new_value)
        raises (Reflective::MofError);
    void unset_exit_routine ()
        raises (Reflective::MofError);
    string null_value ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_null_value (in string new_value)
        raises (Reflective::MofError);
    void unset_null_value ()
        raises (Reflective::MofError);
    IMSDatabase::INDEX index ()
        raises (Reflective::MofError);
    void set_index (in IMSDatabase::INDEX new_value)
        raises (Reflective::MofError);
    SegmentComplex segment ()
        raises (Reflective::MofError);
    void set_segment (in SegmentComplex new_value)
        raises (Reflective::MofError);
    SegmentComplex index_source ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_index_source (in SegmentComplex new_value)
        raises (Reflective::MofError);
    void unset_index_source ()
        raises (Reflective::MofError);
    ImsFieldSet ddata_fields ()
        raises (Reflective::MofError);
    void set_ddata_fields (in ImsFieldSet new_value)
        raises (Reflective::MofError);
    void unset_ddata_fields ()
        raises (Reflective::MofError);
    void add_ddata_fields (in ImsField new_element)
        raises (Reflective::MofError);
    void modify_ddata_fields (
        in ImsField old_element,
        in ImsField new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_ddata_fields (in ImsField old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    ImsFieldSet search_fields ()
        raises (Reflective::MofError);
    void set_search_fields (in ImsFieldSet new_value)
        raises (Reflective::MofError);
    void unset_search_fields ()
        raises (Reflective::MofError);
    void add_search_fields (in ImsField new_element)
        raises (Reflective::MofError);

```

```

void modify_search_fields (
    in ImsField old_element,
    in ImsField new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_search_fields (in ImsField old_element)
    raises (Reflective::NotFound, Reflective::MofError);
ImsFieldSet subseq_fields ()
    raises (Reflective::MofError);
void set_subseq_fields (in ImsFieldSet new_value)
    raises (Reflective::MofError);
void unset_subseq_fields ()
    raises (Reflective::MofError);
void add_subseq_fields (in ImsField new_element)
    raises (Reflective::MofError);
void modify_subseq_fields (
    in ImsField old_element,
    in ImsField new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_subseq_fields (in ImsField old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SecondaryIndex

interface ExitClass : Foundation::Core::ModelElementClass
{
    readonly attribute ExitSet all_of_type_exit;
    readonly attribute ExitSet all_of_class_exit;
    Exit create_exit (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean key,
        in boolean data,
        in boolean path,
        in boolean log,
        in boolean cascade,
        in boolean cascade_key,
        in boolean cascade_data,
        in boolean cascade_path)
        raises (Reflective::MofError);
};

interface Exit : ExitClass, Foundation::Core::ModelElement
{
    boolean key ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_key (in boolean new_value)
        raises (Reflective::MofError);
    void unset_key ()
        raises (Reflective::MofError);
    boolean data ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_data (in boolean new_value)
        raises (Reflective::MofError);
    void unset_data ()
        raises (Reflective::MofError);
    boolean path ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_path (in boolean new_value)
        raises (Reflective::MofError);
    void unset_path ()
        raises (Reflective::MofError);
};

```

```

boolean log ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_log (in boolean new_value)
    raises (Reflective::MofError);
void unset_log ()
    raises (Reflective::MofError);
boolean cascade ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_cascade (in boolean new_value)
    raises (Reflective::MofError);
void unset_cascade ()
    raises (Reflective::MofError);
boolean cascade_key ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_cascade_key (in boolean new_value)
    raises (Reflective::MofError);
void unset_cascade_key ()
    raises (Reflective::MofError);
boolean cascade_data ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_cascade_data (in boolean new_value)
    raises (Reflective::MofError);
void unset_cascade_data ()
    raises (Reflective::MofError);
boolean cascade_path ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_cascade_path (in boolean new_value)
    raises (Reflective::MofError);
void unset_cascade_path ()
    raises (Reflective::MofError);
IMSDatabase::DBD dbd ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_dbd (in IMSDatabase::DBD new_value)
    raises (Reflective::MofError);
void unset_dbd ()
    raises (Reflective::MofError);
IMSDatabase::Segment segment ()
    raises (Reflective::NotSet, Reflective::MofError);
void set_segment (in IMSDatabase::Segment new_value)
    raises (Reflective::MofError);
void unset_segment ()
    raises (Reflective::MofError);
}; // end of interface Exit

interface LCHILDClass : Foundation::Core::ModelElementClass
{
    readonly attribute IMSDatabase::LCHILDSet all_of_type_lchild;
    readonly attribute IMSDatabase::LCHILDSet all_of_class_lchild;
    IMSDatabase::LCHILD create_lchild (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean counter,
        in IMSTypes::ChildPointerType lc_pointer,
        in boolean lparent_flag,
        in IMSTypes::LPointerType ltwin,
        in IMSTypes::RulesType rules,
        in IMSTypes::ParentType virtual_parent)
        raises (Reflective::MofError);
};

```



```

interface LCHILD : LCHILDClass, Foundation::Core::ModelElement
{
    boolean counter ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_counter (in boolean new_value)
        raises (Reflective::MofError);
    void unset_counter ()
        raises (Reflective::MofError);
    IMSTypes::ChildPointerType lc_pointer ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_lc_pointer (in IMSTypes::ChildPointerType new_value)
        raises (Reflective::MofError);
    void unset_lc_pointer ()
        raises (Reflective::MofError);
    boolean lparent_flag ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_lparent_flag (in boolean new_value)
        raises (Reflective::MofError);
    void unset_lparent_flag ()
        raises (Reflective::MofError);
    IMSTypes::LPointerType ltwin ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_ltwin (in IMSTypes::LPointerType new_value)
        raises (Reflective::MofError);
    void unset_ltwin ()
        raises (Reflective::MofError);
    IMSTypes::RulesType rules ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_rules (in IMSTypes::RulesType new_value)
        raises (Reflective::MofError);
    void unset_rules ()
        raises (Reflective::MofError);
    IMSTypes::ParentType virtual_parent ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_virtual_parent (in IMSTypes::ParentType new_value)
        raises (Reflective::MofError);
    void unset_virtual_parent ()
        raises (Reflective::MofError);
    SegmentComplex lparent ()
        raises (Reflective::MofError);
    void set_lparent (in SegmentComplex new_value)
        raises (Reflective::MofError);
    SegmentComplex lchild ()
        raises (Reflective::MofError);
    void set_lchild (in SegmentComplex new_value)
        raises (Reflective::MofError);
    SegmentComplex paired_segment ()
        raises (Reflective::NotSet, Reflective::MofError);
    void set_paired_segment (in SegmentComplex new_value)
        raises (Reflective::MofError);
    void unset_paired_segment ()
        raises (Reflective::MofError);
}; // end of interface LCHILD

interface PSBLibClass : ModelManagement::PackageClass
{
    readonly attribute PSBLibSet all_of_type_psbllib;
    readonly attribute PSBLibSet all_of_class_psbllib;
    PSBLib create_psbllib (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,

```

```

        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface PSBLib : PSBLibClass, ModelManagement::Package
{
    PSBSet psb ()
        raises (Reflective::MofError);
    void set_psb (in PSBSet new_value)
        raises (Reflective::MofError);
    void unset_psb ()
        raises (Reflective::MofError);
    void add_psb (in IMSDatabase::PSB new_element)
        raises (Reflective::MofError);
    void modify_psb (
        in IMSDatabase::PSB old_element,
        in IMSDatabase::PSB new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_psb (in IMSDatabase::PSB old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface PSBLib

interface DBDLibClass : ModelManagement::PackageClass
{
    readonly attribute DBDLibSet all_of_type_dbdlib;
    readonly attribute DBDLibSet all_of_class_dbdlib;
    DBDLib create_dbdlib (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface DBDLib : DBDLibClass, ModelManagement::Package
{
    DBDSet dbd ()
        raises (Reflective::MofError);
    void set_dbd (in DBDSet new_value)
        raises (Reflective::MofError);
    void unset_dbd ()
        raises (Reflective::MofError);
    void add_dbd (in IMSDatabase::DBD new_element)
        raises (Reflective::MofError);
    void modify_dbd (
        in IMSDatabase::DBD old_element,
        in IMSDatabase::DBD new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_dbd (in IMSDatabase::DBD old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface DBDLib

interface IMSDatabasePackageFactory
{
    IMSDatabasePackage create_imsdatabase_package ()
        raises (Reflective::MofError);
};

```

```

};

interface IMSDatabasePackage : Reflective::RefPackage
{
    readonly attribute IMSTypes::IMSTypesPackage imstypes_ref;
    readonly attribute DBDClass dbd_ref;
    readonly attribute PSBClass psb_ref;
    readonly attribute PCBClass pcb_ref;
    readonly attribute SegmentClass segment_ref;
    readonly attribute SegmentComplexClass segment_complex_ref;
    readonly attribute SegmentLogicalClass segment_logical_ref;
    readonly attribute ImsFieldClass ims_field_ref;
    readonly attribute DatasetClass dataset_ref;
    readonly attribute SenSegmentClass sen_segment_ref;
    readonly attribute SenFieldClass sen_field_ref;
    readonly attribute ACBLIBClass acblib_ref;
    readonly attribute AccessMethodClass access_method_ref;
    readonly attribute INDEXClass index_ref;
    readonly attribute HIDAMClass hidam_ref;
    readonly attribute DEDBClass dedb_ref;
    readonly attribute HDAMClass hdam_ref;
    readonly attribute MSDBClass msdb_ref;
    readonly attribute SecondaryIndexClass secondary_index_ref;
    readonly attribute ExitClass exit_ref;
    readonly attribute LCHILDClass lchild_ref;
    readonly attribute PSBLIBClass psblib_ref;
    readonly attribute DBDLIBClass dbdlib_ref;
};
};

#endif

```

2.8 *InformationSet.idl*

```
#ifndef INFORMATION_SET_IDL
#define INFORMATION_SET_IDL
#pragma prefix "org.omg.cwm.x.analysis"
#include "Reflective.idl"
#include "Foundation.idl"
#include "BusinessInformation.idl"
#include "Expressions.idl"
#include "Olap.idl"
#include "InformationVisualization.idl"

module InformationSet
{
    typedef sequence<BusinessInformation::Document> DocumentSet;
    typedef sequence<BusinessInformation::ResponsibleParty> ResponsiblePartySet;
    typedef sequence<InformationVisualization::RenderedObject> RenderedObjectSet;
    interface InformationSetClass;
    interface InformationSet;
    typedef sequence<InformationSet> InformationSetSet;
    interface SegmentClass;
    interface Segment;
    typedef sequence<Segment> SegmentSet;
    interface SegmentRegionClass;
    interface SegmentRegion;
    typedef sequence<SegmentRegion> SegmentRegionSet;
    interface RuleClass;
    interface Rule;
    typedef sequence<Rule> RuleSet;
    interface InfoSetAdministrationClass;
    interface InfoSetAdministration;
    typedef sequence<InfoSetAdministration> InfoSetAdministrationSet;
    interface InfoSetDateClass;
    interface InfoSetDate;
    typedef sequence<InfoSetDate> InfoSetDateSet;
    interface InformationSetPackage;

    interface InformationSetClass : Olap::SchemaClass
    {
        readonly attribute InformationSetSet all_of_type_information_set;
        readonly attribute InformationSetSet all_of_class_information_set;
        InformationSet create_information_set (
            in Foundation::DataTypes::Name name,
            in Foundation::DataTypes::VisibilityKind visibility,
            in boolean is_specification,
            in boolean is_root,
            in boolean is_leaf,
            in boolean is_abstract,
            in string version)
            raises (Reflective::MofError);
    };

    interface InformationSet : InformationSetClass, Olap::Schema
    {
        string version ()
            raises (Reflective::MofError);
        void set_version (in string new_value)
            raises (Reflective::MofError);
    };
}
```

```

RuleSet rule ()
    raises (Reflective::MofError);
void set_rule (in RuleSet new_value)
    raises (Reflective::MofError);
void unset_rule ()
    raises (Reflective::MofError);
void add_rule (in ::InformationSet::Rule new_element)
    raises (Reflective::MofError);
void modify_rule (
    in ::InformationSet::Rule old_element,
    in ::InformationSet::Rule new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_rule (in ::InformationSet::Rule old_element)
    raises (Reflective::NotFound, Reflective::MofError);
InfoSetAdministrationSet info_set_admin ()
    raises (Reflective::MofError);
void set_info_set_admin (in InfoSetAdministrationSet new_value)
    raises (Reflective::MofError);
void unset_info_set_admin ()
    raises (Reflective::MofError);
void add_info_set_admin (in InfoSetAdministration new_element)
    raises (Reflective::MofError);
void modify_info_set_admin (
    in InfoSetAdministration old_element,
    in InfoSetAdministration new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_info_set_admin (in InfoSetAdministration old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface InformationSet

interface SegmentClass : Olap::CubeClass
{
    readonly attribute SegmentSet all_of_type_segment;
    readonly attribute SegmentSet all_of_class_segment;
    Segment create_segment (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in boolean is_virtual,
        in Foundation::DataTypes::ProcedureExpression region_sequence)
        raises (Reflective::MofError);
};

interface Segment : SegmentClass, Olap::Cube
{
    Foundation::DataTypes::ProcedureExpression region_sequence ()
        raises (Reflective::MofError);
    void set_region_sequence (in Foundation::DataTypes::ProcedureExpression new_value)
        raises (Reflective::MofError);
    RuleSet rule ()
        raises (Reflective::MofError);
    void set_rule (in RuleSet new_value)
        raises (Reflective::MofError);
    void unset_rule ()
        raises (Reflective::MofError);
    void add_rule (in ::InformationSet::Rule new_element)
        raises (Reflective::MofError);
}

```

```

void modify_rule (
    in ::InformationSet::Rule old_element,
    in ::InformationSet::Rule new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_rule (in ::InformationSet::Rule old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface Segment

interface SegmentRegionClass : Olap::CubeRegionClass
{
    readonly attribute SegmentRegionSet all_of_type_segment_region;
    readonly attribute SegmentRegionSet all_of_class_segment_region;
    SegmentRegion create_segment_region (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in boolean is_active,
        in boolean is_read_only,
        in boolean is_fully_realized)
        raises (Reflective::MofError);
};

interface SegmentRegion : SegmentRegionClass, Olap::CubeRegion
{
    RuleSet rule ()
        raises (Reflective::MofError);
    void set_rule (in RuleSet new_value)
        raises (Reflective::MofError);
    void unset_rule ()
        raises (Reflective::MofError);
    void add_rule (in ::InformationSet::Rule new_element)
        raises (Reflective::MofError);
    void modify_rule (
        in ::InformationSet::Rule old_element,
        in ::InformationSet::Rule new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_rule (in ::InformationSet::Rule old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    RenderedObjectSet rendered_object ()
        raises (Reflective::MofError);
    void set_rendered_object (in RenderedObjectSet new_value)
        raises (Reflective::MofError);
    void unset_rendered_object ()
        raises (Reflective::MofError);
    void add_rendered_object (in InformationVisualization::RenderedObject new_element)
        raises (Reflective::MofError);
    void modify_rendered_object (
        in InformationVisualization::RenderedObject old_element,
        in InformationVisualization::RenderedObject new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_rendered_object (in InformationVisualization::RenderedObject old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SegmentRegion

interface RuleClass : Foundation::Core::ModelElementClass
{
    readonly attribute RuleSet all_of_type_rule;
    readonly attribute RuleSet all_of_class_rule;

```

```

Rule create_rule (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in Expressions::ExpressionNode rule_expression,
    in string type)
    raises (Reflective::MofError);
};

interface Rule : RuleClass, Foundation::Core::ModelElement
{
    Expressions::ExpressionNode rule_expression ()
        raises (Reflective::MofError);
    void set_rule_expression (in Expressions::ExpressionNode new_value)
        raises (Reflective::MofError);
    string type ()
        raises (Reflective::MofError);
    void set_type (in string new_value)
        raises (Reflective::MofError);
}; // end of interface Rule

interface InfoSetAdministrationClass : Foundation::Core::ModelElementClass
{
    readonly attribute InfoSetAdministrationSet all_of_type_info_set_administration;
    readonly attribute InfoSetAdministrationSet all_of_class_info_set_administration;
    InfoSetAdministration create_info_set_administration (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string priority,
        in boolean acknowledgement)
        raises (Reflective::MofError);
};

interface InfoSetAdministration : InfoSetAdministrationClass, Foundation::Core::ModelElement
{
    string priority ()
        raises (Reflective::MofError);
    void set_priority (in string new_value)
        raises (Reflective::MofError);
    boolean acknowledgement ()
        raises (Reflective::MofError);
    void set_acknowledgement (in boolean new_value)
        raises (Reflective::MofError);
    InfoSetDateSet date ()
        raises (Reflective::MofError);
    void set_date (in InfoSetDateSet new_value)
        raises (Reflective::MofError);
    void unset_date ()
        raises (Reflective::MofError);
    void add_date (in InfoSetDate new_element)
        raises (Reflective::MofError);
    void modify_date (
        in InfoSetDate old_element,
        in InfoSetDate new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_date (in InfoSetDate old_element)
        raises (Reflective::NotFound, Reflective::MofError);
    ResponsiblePartySet responsible_party ()
        raises (Reflective::MofError);
    void set_responsible_party (in ResponsiblePartySet new_value)

```

```

        raises (Reflective::MofError);
void unset_responsible_party ()
    raises (Reflective::MofError);
void add_responsible_party (in BusinessInformation::ResponsibleParty new_element)
    raises (Reflective::MofError);
void modify_responsible_party (
    in BusinessInformation::ResponsibleParty old_element,
    in BusinessInformation::ResponsibleParty new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_responsible_party (in BusinessInformation::ResponsibleParty old_element)
    raises (Reflective::NotFound, Reflective::MofError);
DocumentSet document ()
    raises (Reflective::MofError);
void set_document (in DocumentSet new_value)
    raises (Reflective::MofError);
void unset_document ()
    raises (Reflective::MofError);
void add_document (in BusinessInformation::Document new_element)
    raises (Reflective::MofError);
void modify_document (
    in BusinessInformation::Document old_element,
    in BusinessInformation::Document new_element)
    raises (Reflective::NotFound, Reflective::MofError);
void remove_document (in BusinessInformation::Document old_element)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface InfoSetAdministration

```

```

interface InfoSetDateClass : Foundation::Core::ModelElementClass
{
    readonly attribute InfoSetDateSet all_of_type_info_set_date;
    readonly attribute InfoSetDateSet all_of_class_info_set_date;
    InfoSetDate create_info_set_date (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in string type,
        in string format,
        in string date_time)
        raises (Reflective::MofError);
};

```

```

interface InfoSetDate : InfoSetDateClass, Foundation::Core::ModelElement
{
    string type ()
        raises (Reflective::MofError);
    void set_type (in string new_value)
        raises (Reflective::MofError);
    string format ()
        raises (Reflective::MofError);
    void set_format (in string new_value)
        raises (Reflective::MofError);
    string date_time ()
        raises (Reflective::MofError);
    void set_date_time (in string new_value)
        raises (Reflective::MofError);
}; // end of interface InfoSetDate

```

```

struct InformationSetReferencesRuleLink
{
    InformationSet information_set;
    ::InformationSet::Rule rule;
};

```



```

};
typedef sequence<InformationSetReferencesRuleLink> InformationSetReferencesRuleLinkSet;

interface InformationSetReferencesRule : Reflective::RefAssociation
{
    InformationSetReferencesRuleLinkSet all_information_set_references_rule_links()
        raises (Reflective::MofError);
    boolean exists (
        in InformationSet information_set,
        in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    InformationSet information_set (in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    RuleSet rule (in InformationSet information_set)
        raises (Reflective::MofError);
    void add (
        in InformationSet information_set,
        in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    void modify_information_set (
        in InformationSet information_set,
        in ::InformationSet::Rule rule,
        in InformationSet new_information_set)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_rule (
        in InformationSet information_set,
        in ::InformationSet::Rule rule,
        in ::InformationSet::Rule new_rule)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in InformationSet information_set,
        in ::InformationSet::Rule rule)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface InformationSetReferencesRule

struct SegmentReferencesRuleLink
{
    ::InformationSet::Segment segment;
    ::InformationSet::Rule rule;
};
typedef sequence<SegmentReferencesRuleLink> SegmentReferencesRuleLinkSet;

interface SegmentReferencesRule : Reflective::RefAssociation
{
    SegmentReferencesRuleLinkSet all_segment_references_rule_links()
        raises (Reflective::MofError);
    boolean exists (
        in ::InformationSet::Segment segment,
        in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    ::InformationSet::Segment segment (in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    RuleSet rule (in ::InformationSet::Segment segment)
        raises (Reflective::MofError);
    void add (
        in ::InformationSet::Segment segment,
        in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    void modify_segment (
        in ::InformationSet::Segment segment,
        in ::InformationSet::Rule rule,

```

```

        in ::InformationSet::Segment new_segment)
        raises (Reflective::NotFound, Reflective::MofError);
void modify_rule (
    in ::InformationSet::Segment segment,
    in ::InformationSet::Rule rule,
    in ::InformationSet::Rule new_rule)
    raises (Reflective::NotFound, Reflective::MofError);
void remove (
    in ::InformationSet::Segment segment,
    in ::InformationSet::Rule rule)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SegmentReferencesRule

struct SegmentRegionReferencesRuleLink
{
    SegmentRegion segment_region;
    ::InformationSet::Rule rule;
};
typedef sequence<SegmentRegionReferencesRuleLink>
    SegmentRegionReferencesRuleLinkSet;

interface SegmentRegionReferencesRule : Reflective::RefAssociation
{
    SegmentRegionReferencesRuleLinkSet all_segment_region_references_rule_links()
        raises (Reflective::MofError);
    boolean exists (
        in SegmentRegion segment_region,
        in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    SegmentRegion segment_region (in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    RuleSet rule (in SegmentRegion segment_region)
        raises (Reflective::MofError);
    void add (
        in SegmentRegion segment_region,
        in ::InformationSet::Rule rule)
        raises (Reflective::MofError);
    void modify_segment_region (
        in SegmentRegion segment_region,
        in ::InformationSet::Rule rule,
        in SegmentRegion new_segment_region)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_rule (
        in SegmentRegion segment_region,
        in ::InformationSet::Rule rule,
        in ::InformationSet::Rule new_rule)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in SegmentRegion segment_region,
        in ::InformationSet::Rule rule)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface SegmentRegionReferencesRule

struct InformationSetReferencesInfoSetAdministrationLink
{
    InformationSet information_set;
    InfoSetAdministration info_set_admin;
};
typedef sequence<InformationSetReferencesInfoSetAdministrationLink>
    InformationSetReferencesInfoSetAdministrationLinkSet;

```

```

interface InformationSetReferencesInfoSetAdministration : Reflective::RefAssociation
{
    InformationSetReferencesInfoSetAdministrationLinkSet
        all_information_set_references_info_set_administration_links()
        raises (Reflective::MofError);
    boolean exists (
        in InformationSet information_set,
        in InfoSetAdministration info_set_admin)
        raises (Reflective::MofError);
    InformationSet information_set (in InfoSetAdministration info_set_admin)
        raises (Reflective::MofError);
    InfoSetAdministrationSet info_set_admin (in InformationSet information_set)
        raises (Reflective::MofError);
    void add (
        in InformationSet information_set,
        in InfoSetAdministration info_set_admin)
        raises (Reflective::MofError);
    void modify_information_set (
        in InformationSet information_set,
        in InfoSetAdministration info_set_admin,
        in InformationSet new_information_set)
        raises (Reflective::NotFound, Reflective::MofError);
    void modify_info_set_admin (
        in InformationSet information_set,
        in InfoSetAdministration info_set_admin,
        in InfoSetAdministration new_info_set_admin)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in InformationSet information_set,
        in InfoSetAdministration info_set_admin)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface InformationSetReferencesInfoSetAdministration

struct InfoSetAdministrationReferencesInfoSetDatesLink
{
    InfoSetDate date;
    InfoSetAdministration info_set_admin;
};
typedef sequence<InfoSetAdministrationReferencesInfoSetDatesLink>
    InfoSetAdministrationReferencesInfoSetDatesLinkSet;

interface InfoSetAdministrationReferencesInfoSetDates : Reflective::RefAssociation
{
    InfoSetAdministrationReferencesInfoSetDatesLinkSet
        all_info_set_administration_references_info_set_dates_links()
        raises (Reflective::MofError);
    boolean exists (
        in InfoSetDate date,
        in InfoSetAdministration info_set_admin)
        raises (Reflective::MofError);
    InfoSetDateSet date (in InfoSetAdministration info_set_admin)
        raises (Reflective::MofError);
    InfoSetAdministrationSet info_set_admin (in InfoSetDate date)
        raises (Reflective::MofError);
    void add (
        in InfoSetDate date,
        in InfoSetAdministration info_set_admin)
        raises (Reflective::MofError);
    void modify_date (
        in InfoSetDate date,
        in InfoSetAdministration info_set_admin,

```

```

        in InfoSetDate new_date)
        raises (Reflective::NotFound, Reflective::MofError);
void modify_info_set_admin (
    in InfoSetDate date,
    in InfoSetAdministration info_set_admin,
    in InfoSetAdministration new_info_set_admin)
    raises (Reflective::NotFound, Reflective::MofError);
void remove (
    in InfoSetDate date,
    in InfoSetAdministration info_set_admin)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface InfoSetAdministrationReferencesInfoSetDates

interface InformationSetPackageFactory
{
    InformationSetPackage create_information_set_package ()
        raises (Reflective::MofError);
};

interface InformationSetPackage : Reflective::RefPackage
{
    readonly attribute InformationSetClass information_set_ref;
    readonly attribute SegmentClass segment_ref;
    readonly attribute SegmentRegionClass segment_region_ref;
    readonly attribute RuleClass rule_ref;
    readonly attribute InfoSetAdministrationClass info_set_administration_ref;
    readonly attribute InfoSetDateClass info_set_date_ref;
    readonly attribute InformationSetReferencesRule information_set_references_rule_ref;
    readonly attribute SegmentReferencesRule segment_references_rule_ref;
    readonly attribute SegmentRegionReferencesRule segment_region_references_rule_ref;
    readonly attribute InformationSetReferencesInfoSetAdministration
        information_set_references_info_set_administration_ref;
    readonly attribute InfoSetAdministrationReferencesInfoSetDates
        info_set_administration_references_info_set_dates_ref;
};
};

#endif

```

2.9 *InformationReporting.idl*

```
#ifndef INFORMATION_REPORTING_IDL
#define INFORMATION_REPORTING_IDL
#pragma prefix "org.omg.cwmx.analysis"
#include "Reflective.idl"
#include "Foundation.idl"
#include "Expressions.idl"
#include "Transformation.idl"
#include "InformationVisualization.idl"
#include "DataTypes.idl"

module InformationReporting
{
    typedef sequence<DataTypes::QueryExpression> QueryExpressionSet;
    interface ReportClass;
    interface Report;
    typedef sequence<Report> ReportSet;
    interface ReportAttributeClass;
    interface ReportAttribute;
    typedef sequence<ReportAttribute> ReportAttributeSet;
    interface ReportExecutionClass;
    interface ReportExecution;
    typedef sequence<ReportExecution> ReportExecutionSet;
    interface ReportFieldClass;
    interface ReportField;
    typedef sequence<ReportField> ReportFieldSet;
    interface ReportGroupClass;
    interface ReportGroup;
    typedef sequence<ReportGroup> ReportGroupSet;
    interface ReportPackageClass;
    interface ReportPackage;
    typedef sequence<ReportPackage> ReportPackageSet;
    interface InformationReportingPackage;
    enum ReportGroupType {header, footer, detail, other};

    interface ReportClass : InformationVisualization::RenderedObjectClass
    {
        readonly attribute ReportSet all_of_type_report;
        readonly attribute ReportSet all_of_class_report;
        Report create_report (
            in Foundation::DataTypes::Name name,
            in Foundation::DataTypes::VisibilityKind visibility,
            in boolean is_specification,
            in boolean is_root,
            in boolean is_leaf,
            in boolean is_abstract,
            in Expressions::ExpressionNode formula)
            raises (Reflective::MofError);
    };

    interface Report : ReportClass, InformationVisualization::RenderedObject
    {
    }; // end of interface Report

    interface ReportAttributeClass : Foundation::Core::UmlAttributeClass
    {
        readonly attribute ReportAttributeSet all_of_type_report_attribute;
    };
};
```

```

readonly attribute ReportAttributeSet all_of_class_report_attribute;
ReportAttribute create_report_attribute (
  in Foundation::DataTypes::Name name,
  in Foundation::DataTypes::VisibilityKind visibility,
  in boolean is_specification,
  in Foundation::DataTypes::ScopeKind owner_scope,
  in Foundation::DataTypes::Multiplicity multiplicity,
  in Foundation::DataTypes::ChangeableKind changeability,
  in Foundation::DataTypes::ScopeKind target_scope,
  in Foundation::DataTypes::Expression initial_value)
  raises (Reflective::MofError);
};

interface ReportAttribute : ReportAttributeClass, Foundation::Core::UmlAttribute
{
}; // end of interface ReportAttribute

interface ReportExecutionClass : Transformation::TransformationMapClass
{
  readonly attribute ReportExecutionSet all_of_type_report_execution;
  readonly attribute ReportExecutionSet all_of_class_report_execution;
  ReportExecution create_report_execution (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in Foundation::DataTypes::ProcedureExpression function,
    in string function_description,
    in boolean is_primary)
    raises (Reflective::MofError);
};

interface ReportExecution : ReportExecutionClass, Transformation::TransformationMap
{
}; // end of interface ReportExecution

interface ReportFieldClass : InformationVisualization::RenderedObjectClass
{
  readonly attribute ReportFieldSet all_of_type_report_field;
  readonly attribute ReportFieldSet all_of_class_report_field;
  ReportField create_report_field (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,
    in boolean is_specification,
    in boolean is_root,
    in boolean is_leaf,
    in boolean is_abstract,
    in Expressions::ExpressionNode formula)
    raises (Reflective::MofError);
};

interface ReportField : ReportFieldClass, InformationVisualization::RenderedObject
{
}; // end of interface ReportField

interface ReportGroupClass : InformationVisualization::RenderedObjectClass
{
  readonly attribute ReportGroupSet all_of_type_report_group;
  readonly attribute ReportGroupSet all_of_class_report_group;
  ReportGroup create_report_group (
    in Foundation::DataTypes::Name name,
    in Foundation::DataTypes::VisibilityKind visibility,

```

```

        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract,
        in Expressions::ExpressionNode formula,
        in ReportGroupType type)
        raises (Reflective::MofError);
};

interface ReportGroup : ReportGroupClass, InformationVisualization::RenderedObject
{
    ReportGroupType type ()
        raises (Reflective::MofError);
    void set_type (in ReportGroupType new_value)
        raises (Reflective::MofError);
    QueryExpressionSet report_query ()
        raises (Reflective::MofError);
    void set_report_query (in QueryExpressionSet new_value)
        raises (Reflective::MofError);
    void unset_report_query ()
        raises (Reflective::MofError);
    void add_report_query (in DataTypes::QueryExpression new_element)
        raises (Reflective::MofError);
    void modify_report_query (
        in DataTypes::QueryExpression old_element,
        in DataTypes::QueryExpression new_element)
        raises (Reflective::NotFound, Reflective::MofError);
    void remove_report_query (in DataTypes::QueryExpression old_element)
        raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface ReportGroup

interface ReportPackageClass : InformationVisualization::CwmRenderedObjectSetClass
{
    readonly attribute ReportPackageSet all_of_type_report_package;
    readonly attribute ReportPackageSet all_of_class_report_package;
    ReportPackage create_report_package (
        in Foundation::DataTypes::Name name,
        in Foundation::DataTypes::VisibilityKind visibility,
        in boolean is_specification,
        in boolean is_root,
        in boolean is_leaf,
        in boolean is_abstract)
        raises (Reflective::MofError);
};

interface ReportPackage : ReportPackageClass,
                        InformationVisualization::CwmRenderedObjectSet
{
}; // end of interface ReportPackage

struct ReportGroupReferenceQueryExpressionsLink
{
    DataTypes::QueryExpression report_query;
    ReportGroup report_group;
};
typedef sequence<ReportGroupReferenceQueryExpressionsLink>
                        ReportGroupReferenceQueryExpressionsLinkSet;

interface ReportGroupReferenceQueryExpressions : Reflective::RefAssociation
{
    ReportGroupReferenceQueryExpressionsLinkSet

```

```

        all_report_group_reference_query_expressions_links()
    raises (Reflective::MofError);
    boolean exists (
        in DataTypes::QueryExpression report_query,
        in ReportGroup report_group)
    raises (Reflective::MofError);
    QueryExpressionSet report_query (in ReportGroup report_group)
    raises (Reflective::MofError);
    ReportGroup report_group (in DataTypes::QueryExpression report_query)
    raises (Reflective::MofError);
    void add (
        in DataTypes::QueryExpression report_query,
        in ReportGroup report_group)
    raises (Reflective::MofError);
    void modify_report_query (
        in DataTypes::QueryExpression report_query,
        in ReportGroup report_group,
        in DataTypes::QueryExpression new_report_query)
    raises (Reflective::NotFound, Reflective::MofError);
    void modify_report_group (
        in DataTypes::QueryExpression report_query,
        in ReportGroup report_group,
        in ReportGroup new_report_group)
    raises (Reflective::NotFound, Reflective::MofError);
    void remove (
        in DataTypes::QueryExpression report_query,
        in ReportGroup report_group)
    raises (Reflective::NotFound, Reflective::MofError);
}; // end of interface ReportGroupReferenceQueryExpressions

interface InformationReportingPackageFactory
{
    InformationReportingPackage create_information_reporting_package ()
    raises (Reflective::MofError);
};

interface InformationReportingPackage : Reflective::RefPackage
{
    readonly attribute ReportClass report_ref;
    readonly attribute ReportAttributeClass report_attribute_ref;
    readonly attribute ReportExecutionClass report_execution_ref;
    readonly attribute ReportFieldClass report_field_ref;
    readonly attribute ReportGroupClass report_group_ref;
    readonly attribute ReportPackageClass report_package_ref;
    readonly attribute ReportGroupReferenceQueryExpressions
        report_group_reference_query_expressions_ref;
};
};

#endif

```


3 CWMX Document Types Definitions

This chapter contains an XML document type definition (DTD) for each CWMX metamodel. The document types are generated using Rule Set 3 of the XMI 1.1 Specification.

Each metamodel has a separate DTD which builds on the DTD of each of its imported metamodels. A comment at the front of each DTD lists the DTD of each imported metamodel.

Section 3.1 contains a DTD for all of CWMX. It includes the CWM DTD (from volume 2, section 3.1) and each CWMX metamodel's DTD (from sections 3.2 through 3.10).

3.1 CWMX.dtd

```
<!ENTITY % CWMDTD SYSTEM "CWM.dtd">
%CWMDTD;
<!ENTITY % ERDTD SYSTEM "ER.dtd">
%ERDTD;
<!ENTITY % CORBAIDLDataTypesDTD SYSTEM "CORBAIDLDataTypes.dtd">
%CORBAIDLDataTypesDTD;
<!ENTITY % COBOLDDataDTD SYSTEM "COBOLDData.dtd">
%COBOLDDataDTD;
<!ENTITY % DMSIIDTD SYSTEM "DMSII.dtd">
%DMSIIDTD;
<!ENTITY % EssbaseDTD SYSTEM "Essbase.dtd">
%EssbaseDTD;
<!ENTITY % ExpressDTD SYSTEM "Express.dtd">
%ExpressDTD;
<!ENTITY % IMSDatabaseDTD SYSTEM "IMSDatabase.dtd">
%IMSDatabaseDTD;
<!ENTITY % InformationSetDTD SYSTEM "InformationSet.dtd">
%InformationSetDTD;
<!ENTITY % InformationReportingDTD SYSTEM "InformationReporting.dtd">
%InformationReportingDTD;
```

3.2 *ER.dtd*

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Expressions.dtd -->
<!-- Foundation.dtd -->
<!-- KeysIndexes.dtd -->
<!-- Model_Management.dtd -->

<ATTLIST XMI xmlns:CWMXER CDATA #IMPLIED >

<!-- PACKAGE: CWMXER:ER -->

<!-- CLASS: CWMXER:Entity -->
<ENTITY % CWMXER:EntityProperties '(%UML:ClassProperties;)' >
<ENTITY % CWMXER:EntityAssociations '(%UML:ClassAssociations;)' >
<ENTITY % CWMXER:EntityCompositions '(%UML:ClassCompositions;)' >
<ENTITY % CWMXER:EntityAttPropsList '%UML:ClassAttPropsList;' >
<ENTITY % CWMXER:EntityAttRefsList '%UML:ClassAttRefsList;' >
<ELEMENT CWMXER:Entity ( %CWMXER:EntityProperties;
    , (XML.extension* , %CWMXER:EntityAssociations; )
    , %CWMXER:EntityCompositions; )? >
<ATTLIST CWMXER:Entity %CWMXER:EntityAttPropsList; %CWMXER:EntityAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXER:NonuniqueKey -->
<ENTITY % CWMXER:NonuniqueKeyProperties '(%CWM:IndexProperties;)' >
<ENTITY % CWMXER:NonuniqueKeyAssociations '(%CWM:IndexAssociations;)' >
<ENTITY % CWMXER:NonuniqueKeyCompositions '(%CWM:IndexCompositions;)' >
<ENTITY % CWMXER:NonuniqueKeyAttPropsList '%CWM:IndexAttPropsList;' >
<ENTITY % CWMXER:NonuniqueKeyAttRefsList '%CWM:IndexAttRefsList;' >
<ELEMENT CWMXER:NonuniqueKey ( %CWMXER:NonuniqueKeyProperties;
    , (XML.extension* , %CWMXER:NonuniqueKeyAssociations; )
    , %CWMXER:NonuniqueKeyCompositions; )? >
<ATTLIST CWMXER:NonuniqueKey %CWMXER:NonuniqueKeyAttPropsList;
    %CWMXER:NonuniqueKeyAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXER:CandidateKey -->
<ENTITY % CWMXER:CandidateKeyProperties '(%CWM:UniqueKeyProperties;)' >
<ENTITY % CWMXER:CandidateKeyAssociations '(%CWM:UniqueKeyAssociations;)' >
<ENTITY % CWMXER:CandidateKeyCompositions '(%CWM:UniqueKeyCompositions;)' >
<ENTITY % CWMXER:CandidateKeyAttPropsList '%CWM:UniqueKeyAttPropsList;' >
```

```

<!ENTITY % CWMXER:CandidateKeyAttRefsList '%CWM:UniqueKeyAttRefsList;' >

<!ELEMENT CWMXER:CandidateKey ( %CWMXER:CandidateKeyProperties;
    ,(XML.extension* , %CWMXER:CandidateKeyAssociations; )
    , %CWMXER:CandidateKeyCompositions; )?>

<!ATTLIST CWMXER:CandidateKey %CWMXER:CandidateKeyAttPropsList;
    %CWMXER:CandidateKeyAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXER:ForeignKey -->

<!ELEMENT CWMXER:ForeignKey.relationshipEnd (CWMXER:RelationshipEnd)?>

<!ENTITY % CWMXER:ForeignKeyProperties '(%CWM:KeyRelationshipProperties;)' >

<!ENTITY % CWMXER:ForeignKeyAssociations '(%CWM:KeyRelationshipAssociations;
    ,CWMXER:ForeignKey.relationshipEnd?)' >

<!ENTITY % CWMXER:ForeignKeyCompositions '(%CWM:KeyRelationshipCompositions;)' >

<!ENTITY % CWMXER:ForeignKeyAttPropsList '%CWM:KeyRelationshipAttPropsList;' >

<!ENTITY % CWMXER:ForeignKeyAttRefsList '%CWM:KeyRelationshipAttRefsList;
    relationshipEnd IDREFS #IMPLIED' >

<!ELEMENT CWMXER:ForeignKey ( %CWMXER:ForeignKeyProperties;
    ,(XML.extension* , %CWMXER:ForeignKeyAssociations; )
    , %CWMXER:ForeignKeyCompositions; )?>

<!ATTLIST CWMXER:ForeignKey %CWMXER:ForeignKeyAttPropsList;
    %CWMXER:ForeignKeyAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXER:Domain -->

<!ELEMENT CWMXER:Domain.validationRule (CWM:ExpressionNode
    |CWM:ConstantNode
    |CWM:ElementNode
    |CWM:FeatureNode)>

<!ELEMENT CWMXER:Domain.default (CWM:ExpressionNode
    |CWM:ConstantNode
    |CWM:ElementNode
    |CWM:FeatureNode)>

<!ELEMENT CWMXER:Domain.baseType (UML:Classifier)?>

<!ENTITY % CWMXER:DomainProperties '(%UML:ClassifierProperties;
    ,(CWMXER:Domain.validationRule)?
    ,(CWMXER:Domain.default)?)' >

<!ENTITY % CWMXER:DomainAssociations '(%UML:ClassifierAssociations;
    ,CWMXER:Domain.baseType?)' >

<!ENTITY % CWMXER:DomainCompositions '(%UML:ClassifierCompositions;)' >

<!ENTITY % CWMXER:DomainAttPropsList '%UML:ClassifierAttPropsList;
    validationRule CDATA #IMPLIED
    default CDATA #IMPLIED' >

<!ENTITY % CWMXER:DomainAttRefsList '%UML:ClassifierAttRefsList;
    baseType IDREFS #IMPLIED' >

<!ELEMENT CWMXER:Domain ( %CWMXER:DomainProperties;
    ,(XML.extension* , %CWMXER:DomainAssociations; )
    , %CWMXER:DomainCompositions; )?>

```

```
<!ATTLIST CWMXER:Domain %CWMXER:DomainAttPropsList; %CWMXER:DomainAttRefsList;  
%XML.element.att; %XML.link.att; >
```

<!ENTITY % CWMXER:AttributeProperties '(%UML:AttributeProperties;)' >

```
<!ENTITY % CWMXER:AttributeCompositions '(%UML:AttributeCompositions;)' >
```

<!ENTITY % CWMXER:AttributeAttRefsList '%UML:AttributeAttRefsList;' >

```
<!ATTLIST CWMXER:Attribute %CWMXER:AttributeAttPropsList; %CWMXER:AttributeAttRefsList;  
    %XML.element.att; %XML.link.att; >
```

<!ENTITY % CWMXER:RelationshipProperties '(%UML:AssociationClassProperties;)' >

```
<!ENTITY % CWMXER:RelationshipCompositions '(%UML:AssociationClassCompositions;)' >
```

```
<!ENTITY % CWMXER:RelationshipAttRefsList '%UML:AssociationClassAttRefsList;' >
```

```
<!ATTLIST CWMXER:Relationship %CWMXER:RelationshipAttPropsList;  
                                %CWMXER:RelationshipAttRefsList; %XML.element.att; %XML.link.att; >
```

```
<|ELEMENT CWMXER:RelationshipEnd.delete (CWM:ExpressionNode
|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>
```

```
<!ELEMENT CWMXER:RelationshipEnd.insert (CWM:ExpressionNode
|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>
```

```
<!ENTITY % CWMXER:RelationshipEndProperties '(%UML:AssociationEndProperties;
,(CWMXER:RelationshipEnd.delete)?
,(CWMXER:RelationshipEnd.update)?
,(CWMXER:RelationshipEnd.insert)?)' >
```

```

<!ENTITY % CWMXER:RelationshipEndAssociations '(%UML:AssociationEndAssociations;
,CWMXER:RelationshipEnd.foreignKey?)' >

<!ENTITY % CWMXER:RelationshipEndCompositions '(%UML:AssociationEndCompositions;)' >

<!ENTITY % CWMXER:RelationshipEndAttPropsList '%UML:AssociationEndAttPropsList;
delete CDATA #IMPLIED
update CDATA #IMPLIED
insert CDATA #IMPLIED' >

<!ENTITY % CWMXER:RelationshipEndAttRefsList '%UML:AssociationEndAttRefsList;
foreignKey IDREFS #IMPLIED' >

<!ELEMENT CWMXER:RelationshipEnd ( %CWMXER:RelationshipEndProperties;
,(XML.extension* , %CWMXER:RelationshipEndAssociations; )
, %CWMXER:RelationshipEndCompositions; )?>

<!ATTLIST CWMXER:RelationshipEnd %CWMXER:RelationshipEndAttPropsList;
%CWMXER:RelationshipEndAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXER:ModelLibrary -->

<!ENTITY % CWMXER:ModelLibraryProperties '(%UML:SubsystemProperties;)' >

<!ENTITY % CWMXER:ModelLibraryAssociations '(%UML:SubsystemAssociations;)' >

<!ENTITY % CWMXER:ModelLibraryCompositions '(%UML:SubsystemCompositions;)' >

<!ENTITY % CWMXER:ModelLibraryAttPropsList '%UML:SubsystemAttPropsList;' >

<!ENTITY % CWMXER:ModelLibraryAttRefsList '%UML:SubsystemAttRefsList;' >

<!ELEMENT CWMXER:ModelLibrary ( %CWMXER:ModelLibraryProperties;
,(XML.extension* , %CWMXER:ModelLibraryAssociations; )
, %CWMXER:ModelLibraryCompositions; )?>

<!ATTLIST CWMXER:ModelLibrary %CWMXER:ModelLibraryAttPropsList;
%CWMXER:ModelLibraryAttRefsList; %XML.element.att; %XML.link.att;
>

<!-- CLASS: CWMXER:Model -->

<!ENTITY % CWMXER:ModelProperties '(%UML:ModelProperties;)' >

<!ENTITY % CWMXER:ModelAssociations '(%UML:ModelAssociations;)' >

<!ENTITY % CWMXER:ModelCompositions '(%UML:ModelCompositions;)' >

<!ENTITY % CWMXER:ModelAttPropsList '%UML:ModelAttPropsList;' >

<!ENTITY % CWMXER:ModelAttRefsList '%UML:ModelAttRefsList;' >

<!ELEMENT CWMXER:Model ( %CWMXER:ModelProperties;
,(XML.extension* , %CWMXER:ModelAssociations; )
, %CWMXER:ModelCompositions; )?>

<!ATTLIST CWMXER:Model %CWMXER:ModelAttPropsList; %CWMXER:ModelAttRefsList;
%XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXER:SubjectArea -->

<!ENTITY % CWMXER:SubjectAreaProperties '(%UML:PackageProperties;)' >

```

```

<!ENTITY % CWMXER:SubjectAreaAssociations '(%UML:PackageAssociations;)' >
<!ENTITY % CWMXER:SubjectAreaCompositions '(%UML:PackageCompositions;)' >
<!ENTITY % CWMXER:SubjectAreaAttPropsList '%UML:PackageAttPropsList;' >
<!ENTITY % CWMXER:SubjectAreaAttRefsList '%UML:PackageAttRefsList;' >
<!ELEMENT CWMXER:SubjectArea ( %CWMXER:SubjectAreaProperties;
    ,(XML.extension* , %CWMXER:SubjectAreaAssociations; )
    , %CWMXER:SubjectAreaCompositions; )?>
<ATTLIST CWMXER:SubjectArea %CWMXER:SubjectAreaAttPropsList;
    %CWMXER:SubjectAreaAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXER:PrimaryKey -->
<!ENTITY % CWMXER:PrimaryKeyProperties '(%CWMXER:CandidateKeyProperties;)' >
<!ENTITY % CWMXER:PrimaryKeyAssociations '(%CWMXER:CandidateKeyAssociations;)' >
<!ENTITY % CWMXER:PrimaryKeyCompositions '(%CWMXER:CandidateKeyCompositions;)' >
<!ENTITY % CWMXER:PrimaryKeyAttPropsList '%CWMXER:CandidateKeyAttPropsList;' >
<!ENTITY % CWMXER:PrimaryKeyAttRefsList '%CWMXER:CandidateKeyAttRefsList;' >
<!ELEMENT CWMXER:PrimaryKey ( %CWMXER:PrimaryKeyProperties;
    ,(XML.extension* , %CWMXER:PrimaryKeyAssociations; )
    , %CWMXER:PrimaryKeyCompositions; )?>
<ATTLIST CWMXER:PrimaryKey %CWMXER:PrimaryKeyAttPropsList;
    %CWMXER:PrimaryKeyAttRefsList; %XML.element.att; %XML.link.att; >

```

3.3 CORBAIDLDataTypes.dtd

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- DataTypes.dtd -->

<!ATTLIST XMI xmlns:CWMXIDL CDATA #IMPLIED >

<!-- PACKAGE: CWMXIDL:CORBAIDLDataTypes -->

<!-- CLASS: CWMXIDL:IDLType -->
<!ELEMENT CWMXIDL:IDLType.typeCode (XMI:CorbaTypeCode)>
<!ENTITY % CWMXIDL:IDLTypeProperties '(%UML:DataTypeProperties;
,(CWMXIDL:IDLType.typeCode)?)' >
<!ENTITY % CWMXIDL:IDLTypeAssociations '(%UML:DataTypeAssociations;)' >
<!ENTITY % CWMXIDL:IDLTypeCompositions '(%UML:DataTypeCompositions;)' >
<!ENTITY % CWMXIDL:IDLTypeAttPropsList '%UML:DataTypeAttPropsList;' >
<!ENTITY % CWMXIDL:IDLTypeAttRefsList '%UML:DataTypeAttRefsList;' >
<!ELEMENT CWMXIDL:IDLType ( %CWMXIDL:IDLTypeProperties;
,(XML.extension* , %CWMXIDL:IDLTypeAssociations; )
, %CWMXIDL:IDLTypeCompositions; )?>
<!ATTLIST CWMXIDL:IDLType %CWMXIDL:IDLTypeAttPropsList;
%CWMXIDL:IDLTypeAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:ArrayType -->
<!ELEMENT CWMXIDL:ArrayType.length (#PCDATA|XMI.reference)*>
<!ELEMENT CWMXIDL:ArrayType.elementType (CWMXIDL:IDLType)?>
<!ENTITY % CWMXIDL:ArrayTypeProperties '(%CWMXIDL:IDLTypeProperties;
,(CWMXIDL:ArrayType.length)?)' >
<!ENTITY % CWMXIDL:ArrayTypeAssociations '(%CWMXIDL:IDLTypeAssociations;
,CWMXIDL:ArrayType.elementType)?' >
<!ENTITY % CWMXIDL:ArrayTypeCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:ArrayTypeAttPropsList '%CWMXIDL:IDLTypeAttPropsList;
length CDATA #IMPLIED' >
<!ENTITY % CWMXIDL:ArrayTypeAttRefsList '%CWMXIDL:IDLTypeAttRefsList;
elementType IDREFS #IMPLIED' >
<!ELEMENT CWMXIDL:ArrayType ( %CWMXIDL:ArrayTypeProperties;
,(XML.extension* , %CWMXIDL:ArrayTypeAssociations; )
, %CWMXIDL:ArrayTypeCompositions; )?>
<!ATTLIST CWMXIDL:ArrayType %CWMXIDL:ArrayTypeAttPropsList;
%CWMXIDL:ArrayTypeAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:ExceptionType -->
```



```

<!ENTITY % CWMXIDL:ExceptionTypeProperties '(%CWMXIDL:IDLTypeProperties;)' >
<!ENTITY % CWMXIDL:ExceptionTypeAssociations '(%CWMXIDL:IDLTypeAssociations;)' >
<!ENTITY % CWMXIDL:ExceptionTypeCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:ExceptionTypeAttPropsList '%CWMXIDL:IDLTypeAttPropsList;' >
<!ENTITY % CWMXIDL:ExceptionTypeAttRefsList '%CWMXIDL:IDLTypeAttRefsList;' >
<!ELEMENT CWMXIDL:ExceptionType ( %CWMXIDL:ExceptionTypeProperties;
    ,(XML.extension* , %CWMXIDL:ExceptionTypeAssociations; )
    , %CWMXIDL:ExceptionTypeCompositions; )?>
<!ATTLIST CWMXIDL:ExceptionType %CWMXIDL:ExceptionTypeAttPropsList;
    %CWMXIDL:ExceptionTypeAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXIDL:SequenceType -->
<!ELEMENT CWMXIDL:SequenceType.length (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIDL:SequenceType.elementType (CWMXIDL:IDLType)?>
<!ENTITY % CWMXIDL:SequenceTypeProperties '(%CWMXIDL:IDLTypeProperties;
    ,(CWMXIDL:SequenceType.length)?)' >
<!ENTITY % CWMXIDL:SequenceTypeAssociations '(%CWMXIDL:IDLTypeAssociations;
    ,CWMXIDL:SequenceType.elementType?)' >
<!ENTITY % CWMXIDL:SequenceTypeCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:SequenceTypeAttPropsList '%CWMXIDL:IDLTypeAttPropsList;
    length CDATA #IMPLIED' >
<!ENTITY % CWMXIDL:SequenceTypeAttRefsList '%CWMXIDL:IDLTypeAttRefsList;
    elementType IDREFS #IMPLIED' >
<!ELEMENT CWMXIDL:SequenceType ( %CWMXIDL:SequenceTypeProperties;
    ,(XML.extension* , %CWMXIDL:SequenceTypeAssociations; )
    , %CWMXIDL:SequenceTypeCompositions; )?>
<!ATTLIST CWMXIDL:SequenceType %CWMXIDL:SequenceTypeAttPropsList;
    %CWMXIDL:SequenceTypeAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXIDL:WstringType -->
<!ELEMENT CWMXIDL:WstringType.length (#PCDATA|XML.reference)*>
<!ENTITY % CWMXIDL:WstringTypeProperties '(%CWMXIDL:IDLTypeProperties;
    ,(CWMXIDL:WstringType.length)?)' >
<!ENTITY % CWMXIDL:WstringTypeAssociations '(%CWMXIDL:IDLTypeAssociations;)' >
<!ENTITY % CWMXIDL:WstringTypeCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:WstringTypeAttPropsList '%CWMXIDL:IDLTypeAttPropsList;
    length CDATA #IMPLIED' >
<!ENTITY % CWMXIDL:WstringTypeAttRefsList '%CWMXIDL:IDLTypeAttRefsList;' >
<!ELEMENT CWMXIDL:WstringType ( %CWMXIDL:WstringTypeProperties;
    ,(XML.extension* , %CWMXIDL:WstringTypeAssociations; )
    , %CWMXIDL:WstringTypeCompositions; )?>

```

```

<!ATTLIST CWMXIDL:WstringType %CWMXIDL:WstringTypeAttPropsList;
                %CWMXIDL:WstringTypeAttRefsList; %XML.element.att; %XML.link.att;
                >

<!-- CLASS: CWMXIDL:StringType -->
<!ELEMENT CWMXIDL:StringType.length (#PCDATA|XML.reference)*>
<!ENTITY % CWMXIDL:StringTypeProperties '(%CWMXIDL:IDLTypeProperties;
    ,(CWMXIDL:StringType.length)?)' >
<!ENTITY % CWMXIDL:StringTypeAssociations '(%CWMXIDL:IDLTypeAssociations;)' >
<!ENTITY % CWMXIDL:StringTypeCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:StringTypeAttPropsList '%CWMXIDL:IDLTypeAttPropsList;
    length CDATA #IMPLIED' >
<!ENTITY % CWMXIDL:StringTypeAttRefsList '%CWMXIDL:IDLTypeAttRefsList;' >
<!ELEMENT CWMXIDL:StringType ( %CWMXIDL:StringTypeProperties;
    ,(XML.extension* , %CWMXIDL:StringTypeAssociations; )
    , %CWMXIDL:StringTypeCompositions; )?>
<!ATTLIST CWMXIDL:StringType %CWMXIDL:StringTypeAttPropsList;
                %CWMXIDL:StringTypeAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:FixedType -->
<!ELEMENT CWMXIDL:FixedType.digits (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIDL:FixedType.scale (#PCDATA|XML.reference)*>
<!ENTITY % CWMXIDL:FixedTypeProperties '(%CWMXIDL:IDLTypeProperties;
    ,(CWMXIDL:FixedType.digits)?
    ,(CWMXIDL:FixedType.scale)?)' >
<!ENTITY % CWMXIDL:FixedTypeAssociations '(%CWMXIDL:IDLTypeAssociations;)' >
<!ENTITY % CWMXIDL:FixedTypeCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:FixedTypeAttPropsList '%CWMXIDL:IDLTypeAttPropsList;
    digits CDATA #IMPLIED
    scale CDATA #IMPLIED' >
<!ENTITY % CWMXIDL:FixedTypeAttRefsList '%CWMXIDL:IDLTypeAttRefsList;' >
<!ELEMENT CWMXIDL:FixedType ( %CWMXIDL:FixedTypeProperties;
    ,(XML.extension* , %CWMXIDL:FixedTypeAssociations; )
    , %CWMXIDL:FixedTypeCompositions; )?>
<!ATTLIST CWMXIDL:FixedType %CWMXIDL:FixedTypeAttPropsList;
                %CWMXIDL:FixedTypeAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:Alias -->
<!ENTITY % CWMXIDL:AliasProperties '(%CWMXIDL:IDLTypeProperties;)' >
<!ENTITY % CWMXIDL:AliasAssociations '(%CWMXIDL:IDLTypeAssociations;
    ,CWM:TypeAlias.type?)' >
<!ENTITY % CWMXIDL:AliasCompositions '(%CWMXIDL:IDLTypeCompositions;)' >
<!ENTITY % CWMXIDL:AliasAttPropsList '%CWMXIDL:IDLTypeAttPropsList;' >

```

```

<!ENTITY % CWMXIDL:AliasAttRefsList '%CWMXIDL:IDTypeAttRefsList;
    type IDREFS #IMPLIED' >

<!ELEMENT CWMXIDL:Alias ( %CWMXIDL:AliasProperties;
    ,(XML.extension* , %CWMXIDL:AliasAssociations; )
    , %CWMXIDL:AliasCompositions; )?>

<!ATTLIST CWMXIDL:Alias %CWMXIDL:AliasAttPropsList; %CWMXIDL:AliasAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:StructType -->

<!ENTITY % CWMXIDL:StructTypeProperties '(%CWMXIDL:IDTypeProperties;)' >
<!ENTITY % CWMXIDL:StructTypeAssociations '(%CWMXIDL:IDTypeAssociations;)' >
<!ENTITY % CWMXIDL:StructTypeCompositions '(%CWMXIDL:IDTypeCompositions;)' >
<!ENTITY % CWMXIDL:StructTypeAttPropsList '%CWMXIDL:IDTypeAttPropsList;' >
<!ENTITY % CWMXIDL:StructTypeAttRefsList '%CWMXIDL:IDTypeAttRefsList;' >
<!ELEMENT CWMXIDL:StructType ( %CWMXIDL:StructTypeProperties;
    ,(XML.extension* , %CWMXIDL:StructTypeAssociations; )
    , %CWMXIDL:StructTypeCompositions; )?>
<!ATTLIST CWMXIDL:StructType %CWMXIDL:StructTypeAttPropsList;
    %CWMXIDL:StructTypeAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:UnionType -->

<!ENTITY % CWMXIDL:UnionTypeProperties '(%CWM:UnionProperties;
    ,(CWMXIDL:IDType.typeCode?))' >
<!ENTITY % CWMXIDL:UnionTypeAssociations '(%CWM:UnionAssociations;)' >
<!ENTITY % CWMXIDL:UnionTypeCompositions '(%CWM:UnionCompositions;)' >
<!ENTITY % CWMXIDL:UnionTypeAttPropsList '%CWM:UnionAttPropsList;' >
<!ENTITY % CWMXIDL:UnionTypeAttRefsList '%CWM:UnionAttRefsList;' >
<!ELEMENT CWMXIDL:UnionType ( %CWMXIDL:UnionTypeProperties;
    ,(XML.extension* , %CWMXIDL:UnionTypeAssociations; )
    , %CWMXIDL:UnionTypeCompositions; )?>
<!ATTLIST CWMXIDL:UnionType %CWMXIDL:UnionTypeAttPropsList;
    %CWMXIDL:UnionTypeAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIDL:EnumType -->

<!ENTITY % CWMXIDL:EnumTypeProperties '(%CWM:EnumerationProperties;
    ,(CWMXIDL:IDType.typeCode?))' >
<!ENTITY % CWMXIDL:EnumTypeAssociations '(%CWM:EnumerationAssociations;)' >
<!ENTITY % CWMXIDL:EnumTypeCompositions '(%CWM:EnumerationCompositions;)' >
<!ENTITY % CWMXIDL:EnumTypeAttPropsList '%CWM:EnumerationAttPropsList;' >
<!ENTITY % CWMXIDL:EnumTypeAttRefsList '%CWM:EnumerationAttRefsList;' >
<!ELEMENT CWMXIDL:EnumType ( %CWMXIDL:EnumTypeProperties;
    ,(XML.extension* , %CWMXIDL:EnumTypeAssociations; )
    , %CWMXIDL:EnumTypeCompositions; )?>

```

```
<!ATTLIST CWMXIDL:EnumType %CWMXIDL:EnumTypeAttPropsList;  
          %CWMXIDL:EnumTypeAttRefsList; %XML.element.att; %XML.link.att; >
```

3.4 *COBOLData.dtd*

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- Record.dtd -->
<!-- KeysIndexes.dtd -->

<!ATTLIST XMI xmlns:CWMXCOBOL CDATA #IMPLIED >

<!-- PACKAGE: CWMXCOBOL:COBOLData -->

<!ENTITY % CWMXCOBOL:SignKindType
    ' xmi.value ( unsigned| leadingSign| trailingSign| leadingSepSign| trailingSepSign )
    #REQUIRED'>

<!ENTITY % CWMXCOBOL:FileOrganization
    ' xmi.value ( unspecified| indexed| relative| sequential ) #REQUIRED'>

<!ENTITY % CWMXCOBOL:AccessType
    ' xmi.value ( unspecified| dynamic| random| sequential ) #REQUIRED'>

<!ENTITY % CWMXCOBOL:BlockKind
    ' xmi.value ( records| characters ) #REQUIRED'>

<!ENTITY % CWMXCOBOL:LabelKind
    ' xmi.value ( unspecified| standard| omitted ) #REQUIRED'>

<!ENTITY % CWMXCOBOL:LinageInfoType
    ' xmi.value ( lineage| lineageFooting| lineageTop| lineageBottom ) #REQUIRED'>

<!-- ***** CWMXCOBOL:LineageInfoField ***** -->
<!ELEMENT CWMXCOBOL:COBOLFD.linageInfo ( CWMXCOBOL:LinageInfo)* >

<!-- ***** CWMXCOBOL:FileSectionFD ***** -->
<!ELEMENT CWMXCOBOL:FileSection.cobolFD ( CWMXCOBOL:COBOLFD)* >

<!-- ***** CWMXCOBOL:OccursKeyField ***** -->
<!ELEMENT CWMXCOBOL:COBOLField.occursKeyInfo ( CWMXCOBOL:OccursKey)* >

<!-- CLASS: CWMXCOBOL:COBOLField -->
<!ELEMENT CWMXCOBOL:COBOLField.level (#PCDATA|XMI.reference)*>
<!ELEMENT CWMXCOBOL:COBOLField.signKind EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.signKind %CWMXCOBOL:SignKindType;>
<!ELEMENT CWMXCOBOL:COBOLField.isFiller EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.isFiller
    xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXCOBOL:COBOLField.isJustifiedRight EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.isJustifiedRight
    xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXCOBOL:COBOLField.isBlankWhenZero EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.isBlankWhenZero
    xmi.value ( true | false ) #REQUIRED>
```

```

<!ELEMENT CWMXCOBOL:COBOLField.isSynchronized EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.isSynchronized
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXCOBOL:COBOLField.picture (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLField.occursLower (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLField.occursUpper (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLField.indexName (#PCDATA|XML.reference)*>

<!ELEMENT CWMXCOBOL:COBOLField.isExternal EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.isExternal
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXCOBOL:COBOLField.isGlobal EMPTY>
<!ATTLIST CWMXCOBOL:COBOLField.isGlobal
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXCOBOL:COBOLField DependingOnField (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:COBOLField OccurringField (CWMXCOBOL:COBOLField)*>
<!ELEMENT CWMXCOBOL:COBOLField OccursKeyFieldInfo (CWMXCOBOL:OccursKey)*>
<!ELEMENT CWMXCOBOL:COBOLField RedefinedByField (CWMXCOBOL:COBOLField)*>
<!ELEMENT CWMXCOBOL:COBOLField RedefinedField (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:COBOLField PaddedFD (CWMXCOBOL:COBOLFD)*>
<!ELEMENT CWMXCOBOL:COBOLField LinageInfo (CWMXCOBOL:LinageInfo)*>
<!ELEMENT CWMXCOBOL:COBOLField DependingFD (CWMXCOBOL:COBOLFD)*>
<!ELEMENT CWMXCOBOL:COBOLField StatusFD (CWMXCOBOL:COBOLFD)*>
<!ELEMENT CWMXCOBOL:COBOLField FirstRenames (CWMXCOBOL:Renames)*>
<!ELEMENT CWMXCOBOL:COBOLField ThruRenames (CWMXCOBOL:Renames)*>
<!ELEMENT CWMXCOBOL:COBOLField RelativeFD (CWMXCOBOL:COBOLFD)*>

<!ENTITY % CWMXCOBOL:COBOLFieldProperties '(%CWMREC:FieldProperties;
    ,(CWMXCOBOL:COBOLField.level)?
    ,(CWMXCOBOL:COBOLField.signKind)?
    ,(CWMXCOBOL:COBOLField.isFiller)?
    ,(CWMXCOBOL:COBOLField.isJustifiedRight)?
    ,(CWMXCOBOL:COBOLField.isBlankWhenZero)?
    ,(CWMXCOBOL:COBOLField.isSynchronized)?
    ,(CWMXCOBOL:COBOLField.picture)?
    ,(CWMXCOBOL:COBOLField.occursLower)?
    ,(CWMXCOBOL:COBOLField.occursUpper)?
    ,(CWMXCOBOL:COBOLField.indexName)*
    ,(CWMXCOBOL:COBOLField.isExternal)?
    ,(CWMXCOBOL:COBOLField.isGlobal)?)' >

<!ENTITY % CWMXCOBOL:COBOLFieldAssociations '(%CWMREC:FieldAssociations;
    ,CWMXCOBOL:COBOLField DependingOnField?
    ,CWMXCOBOL:COBOLField OccurringField*
    ,CWMXCOBOL:COBOLField OccursKeyFieldInfo*
    ,CWMXCOBOL:COBOLField RedefinedByField*
    ,CWMXCOBOL:COBOLField RedefinedField?
    ,CWMXCOBOL:COBOLField PaddedFD*
    ,CWMXCOBOL:COBOLField LinageInfo*
    ,CWMXCOBOL:COBOLField DependingFD*

```

```

,CWMXCOBOL:COBOLField.statusFD*
,CWMXCOBOL:COBOLField.firstRenames*
,CWMXCOBOL:COBOLField.thruRenames*
,CWMXCOBOL:COBOLField.relativeFD*)' >

<!ENTITY % CWMXCOBOL:COBOLFieldCompositions '(%CWMREC:FieldCompositions;
,CWMXCOBOL:COBOLField.occursKeyInfo*)' >

<!ENTITY % CWMXCOBOL:COBOLFieldAttPropsList '%CWMREC:FieldAttPropsList;
level CDATA #IMPLIED
signKind ( unsigned| leadingSign| trailingSign| leadingSepSign| trailingSepSign ) #IMPLIED
isFiller (true|false) #IMPLIED
isJustifiedRight (true|false) #IMPLIED
isBlankWhenZero (true|false) #IMPLIED
isSynchronized (true|false) #IMPLIED
picture CDATA #IMPLIED
occursLower CDATA #IMPLIED
occursUpper CDATA #IMPLIED
isExternal (true|false) #IMPLIED
isGlobal (true|false) #IMPLIED' >

<!ENTITY % CWMXCOBOL:COBOLFieldAttRefsList '%CWMREC:FieldAttRefsList;
dependingOnField IDREFS #IMPLIED
redefinedField IDREFS #IMPLIED' >

<!ELEMENT CWMXCOBOL:COBOLField ( %CWMXCOBOL:COBOLFieldProperties;
,(XML.extension* , %CWMXCOBOL:COBOLFieldAssociations; )
,%CWMXCOBOL:COBOLFieldCompositions; )?>

<!ATTLIST CWMXCOBOL:COBOLField %CWMXCOBOL:COBOLFieldAttPropsList;
%CWMXCOBOL:COBOLFieldAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXCOBOL:Usage -->

<!ENTITY % CWMXCOBOL:UsageProperties '(%UML:DataTypeProperties;)' >

<!ENTITY % CWMXCOBOL:UsageAssociations '(%UML:DataTypeAssociations;)' >

<!ENTITY % CWMXCOBOL:UsageCompositions '(%UML:DataTypeCompositions;)' >

<!ENTITY % CWMXCOBOL:UsageAttPropsList '%UML:DataTypeAttPropsList;' >

<!ENTITY % CWMXCOBOL:UsageAttRefsList '%UML:DataTypeAttRefsList;' >

<!ELEMENT CWMXCOBOL:Usage ( %CWMXCOBOL:UsageProperties;
,(XML.extension* , %CWMXCOBOL:UsageAssociations; )
,%CWMXCOBOL:UsageCompositions; )?>

<!ATTLIST CWMXCOBOL:Usage %CWMXCOBOL:UsageAttPropsList;
%CWMXCOBOL:UsageAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXCOBOL:COBOLFD -->

<!ELEMENT CWMXCOBOL:COBOLFD.organization EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.organization %CWMXCOBOL:FileOrganization;>

<!ELEMENT CWMXCOBOL:COBOLFD.accessMode EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.accessMode %CWMXCOBOL:AccessType;>

<!ELEMENT CWMXCOBOL:COBOLFD.isOptional EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.isOptional
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXCOBOL:COBOLFD.reserveAreas (#PCDATA|XML.reference)*>

```

```

<!ELEMENT CWMXCOBOL:COBOLFD.assignTo (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.codeSetLit (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.blockSizeUnit EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.blockSizeUnit %CWMXCOBOL:BlockKind;>
<!ELEMENT CWMXCOBOL:COBOLFD.minBlocks (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.maxBlocks (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.minRecords (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.maxRecords (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.labelKind EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.labelKind %CWMXCOBOL:LabelKind;>
<!ELEMENT CWMXCOBOL:COBOLFD.isExternal EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.isExternal
  xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXCOBOL:COBOLFD.isGlobal EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFD.isGlobal
  xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXCOBOL:COBOLFD.padLiteral (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:COBOLFD.fileSection (CWMXCOBOL:FileSection)?>
<!ELEMENT CWMXCOBOL:COBOLFD.statusID (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:COBOLFD.dependsOn (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:COBOLFD.padField (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:COBOLFD.relativeField (CWMXCOBOL:COBOLField)?>
<!ENTITY % CWMXCOBOL:COBOLFDProperties '(%CWMREC:RecordFileProperties;
  ,UML:Class.isActive)?
  ,(CWMXCOBOL:COBOLFD.organization)?
  ,(CWMXCOBOL:COBOLFD.accessMode)?
  ,(CWMXCOBOL:COBOLFD.isOptional)?
  ,(CWMXCOBOL:COBOLFD.reserveAreas)?
  ,(CWMXCOBOL:COBOLFD.assignTo)?
  ,(CWMXCOBOL:COBOLFD.codeSetLit)?
  ,(CWMXCOBOL:COBOLFD.blockSizeUnit)?
  ,(CWMXCOBOL:COBOLFD.minBlocks)?
  ,(CWMXCOBOL:COBOLFD.maxBlocks)?
  ,(CWMXCOBOL:COBOLFD.minRecords)?
  ,(CWMXCOBOL:COBOLFD.maxRecords)?
  ,(CWMXCOBOL:COBOLFD.labelKind)?
  ,(CWMXCOBOL:COBOLFD.isExternal)?
  ,(CWMXCOBOL:COBOLFD.isGlobal)?
  ,(CWMXCOBOL:COBOLFD.padLiteral)?)' >
<!ENTITY % CWMXCOBOL:COBOLFDAssociations '(%CWMREC:RecordFileAssociations;
  ,UML:Classifier.participant*
  ,UML:Classifier.powerTypeRange*
  ,CWMXCOBOL:COBOLFD.fileSection?
  ,CWMXCOBOL:COBOLFD.statusID?
  ,CWMXCOBOL:COBOLFD.dependsOn?
  ,CWMXCOBOL:COBOLFD.padField?
  ,CWMXCOBOL:COBOLFD.relativeField?)' >
<!ENTITY % CWMXCOBOL:COBOLFDCompositions '(%CWMREC:RecordFileCompositions;
  ,UML:Classifier.feature*
  ,CWMXCOBOL:COBOLFD.linageInfo*)' >

```



```

<!ENTITY % CWMXCOBOL:COBOLFDAttPropsList '%CWMREC:RecordFileAttPropsList;
    isActive (true|false) #IMPLIED
    organization ( unspecified| indexed| relative| sequential ) #IMPLIED
    accessMode ( unspecified| dynamic| random| sequential ) #IMPLIED
    isOptional (true|false) #IMPLIED
    reserveAreas CDATA #IMPLIED
    assignTo CDATA #IMPLIED
    codeSetLit CDATA #IMPLIED
    blockSizeUnit ( records| characters ) #IMPLIED
    minBlocks CDATA #IMPLIED
    maxBlocks CDATA #IMPLIED
    minRecords CDATA #IMPLIED
    maxRecords CDATA #IMPLIED
    labelKind ( unspecified| standard| omitted ) #IMPLIED
    isExternal (true|false) #IMPLIED
    isGlobal (true|false) #IMPLIED
    padLiteral CDATA #IMPLIED' >

<!ENTITY % CWMXCOBOL:COBOLFDAttRefsList '%CWMREC:RecordFileAttRefsList;
    fileSection IDREFS #IMPLIED
    statusID IDREFS #IMPLIED
    dependsOn IDREFS #IMPLIED
    padField IDREFS #IMPLIED
    relativeField IDREFS #IMPLIED' >

<!ELEMENT CWMXCOBOL:COBOLFD ( %CWMXCOBOL:COBOLFDProperties;
    , (XML.extension* , %CWMXCOBOL:COBOLFDAssociations; )
    , %CWMXCOBOL:COBOLFDCompositions; )?>

<!ATTLIST CWMXCOBOL:COBOLFD %CWMXCOBOL:COBOLFDAttPropsList;
    %CWMXCOBOL:COBOLFDAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXCOBOL:Renames -->

<!ELEMENT CWMXCOBOL:Renames.firstField (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:Renames.thruField (CWMXCOBOL:COBOLField)?>
<!ENTITY % CWMXCOBOL:RenamesProperties '(%UML:FeatureProperties;)' >
<!ENTITY % CWMXCOBOL:RenamesAssociations '(%UML:FeatureAssociations;
    ,CWMXCOBOL:Renames.firstField?
    ,CWMXCOBOL:Renames.thruField?)' >
<!ENTITY % CWMXCOBOL:RenamesCompositions '(%UML:FeatureCompositions;)' >
<!ENTITY % CWMXCOBOL:RenamesAttPropsList '%UML:FeatureAttPropsList;' >
<!ENTITY % CWMXCOBOL:RenamesAttRefsList '%UML:FeatureAttRefsList;
    firstField IDREFS #IMPLIED
    thruField IDREFS #IMPLIED' >
<!ELEMENT CWMXCOBOL:Renames ( %CWMXCOBOL:RenamesProperties;
    , (XML.extension* , %CWMXCOBOL:RenamesAssociations; )
    , %CWMXCOBOL:RenamesCompositions; )?>
<!ATTLIST CWMXCOBOL:Renames %CWMXCOBOL:RenamesAttPropsList;
    %CWMXCOBOL:RenamesAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXCOBOL:Section -->

<!ELEMENT CWMXCOBOL:Section.record (CWMREC:RecordDef)*>

```

```

<!ENTITY % CWMXCOBOL:SectionProperties '(%UML:ClassifierProperties;)' >
<!ENTITY % CWMXCOBOL:SectionAssociations '(%UML:ClassifierAssociations;
,CWMXCOBOL:Section.record*)' >
<!ENTITY % CWMXCOBOL:SectionCompositions '(%UML:ClassifierCompositions;)' >
<!ENTITY % CWMXCOBOL:SectionAttPropsList '%UML:ClassifierAttPropsList;' >
<!ENTITY % CWMXCOBOL:SectionAttRefsList '%UML:ClassifierAttRefsList;' >
<!ELEMENT CWMXCOBOL:Section ( %CWMXCOBOL:SectionProperties;
,(XML.extension* , %CWMXCOBOL:SectionAssociations; )
, %CWMXCOBOL:SectionCompositions; )?>
<!ATTLIST CWMXCOBOL:Section %CWMXCOBOL:SectionAttPropsList;
%CWMXCOBOL:SectionAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXCOBOL:COBOLFDIndex -->
<!ELEMENT CWMXCOBOL:COBOLFDIndex.isAlternate EMPTY>
<!ATTLIST CWMXCOBOL:COBOLFDIndex.isAlternate
xmi.value ( true | false ) #REQUIRED>
<!ENTITY % CWMXCOBOL:COBOLFDIndexProperties '(%CWM:IndexProperties;
,(CWMXCOBOL:COBOLFDIndex.isAlternate?))' >
<!ENTITY % CWMXCOBOL:COBOLFDIndexAssociations '(%CWM:IndexAssociations;)' >
<!ENTITY % CWMXCOBOL:COBOLFDIndexCompositions '(%CWM:IndexCompositions;)' >
<!ENTITY % CWMXCOBOL:COBOLFDIndexAttPropsList '%CWM:IndexAttPropsList;
isAlternate (true|false) #IMPLIED ' >
<!ENTITY % CWMXCOBOL:COBOLFDIndexAttRefsList '%CWM:IndexAttRefsList;' >
<!ELEMENT CWMXCOBOL:COBOLFDIndex ( %CWMXCOBOL:COBOLFDIndexProperties;
,(XML.extension* , %CWMXCOBOL:COBOLFDIndexAssociations; )
, %CWMXCOBOL:COBOLFDIndexCompositions; )?>
<!ATTLIST CWMXCOBOL:COBOLFDIndex %CWMXCOBOL:COBOLFDIndexAttPropsList;
%CWMXCOBOL:COBOLFDIndexAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXCOBOL:LinageInfo -->
<!ELEMENT CWMXCOBOL:LinageInfo.value (#PCDATA|XML.reference)*>
<!ELEMENT CWMXCOBOL:LinageInfo.type EMPTY>
<!ATTLIST CWMXCOBOL:LinageInfo.type %CWMXCOBOL:LinageInfoType;>
<!ELEMENT CWMXCOBOL:LinageInfo.cobolFD (CWMXCOBOL:COBOLFD)?>
<!ELEMENT CWMXCOBOL:LinageInfo.cobolField (CWMXCOBOL:COBOLField)?>
<!ENTITY % CWMXCOBOL:LinageInfoProperties '(%UML:ModelElementProperties;
,(CWMXCOBOL:LinageInfo.value)?
,(CWMXCOBOL:LinageInfo.type?))' >
<!ENTITY % CWMXCOBOL:LinageInfoAssociations '(%UML:ModelElementAssociations;
,CWMXCOBOL:LinageInfo.cobolFD?
,CWMXCOBOL:LinageInfo.cobolField?)' >
<!ENTITY % CWMXCOBOL:LinageInfoCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXCOBOL:LinageInfoAttPropsList '%UML:ModelElementAttPropsList;

```

```

        value CDATA #IMPLIED
        type ( lineage| lineageFooting| lineageTop| lineageBottom ) #IMPLIED' >

<!ENTITY % CWMXCOBOL:LinageInfoAttRefsList '%UML:ModelElementAttRefsList;
        cobolFD IDREFS #IMPLIED
        cobolField IDREFS #IMPLIED' >

<!ELEMENT CWMXCOBOL:LinageInfo ( %CWMXCOBOL:LinageInfoProperties;
        ,(XML.extension* , %CWMXCOBOL:LinageInfoAssociations; )
        , %CWMXCOBOL:LinageInfoCompositions; )?>

<!ATTLIST CWMXCOBOL:LinageInfo %CWMXCOBOL:LinageInfoAttPropsList;
        %CWMXCOBOL:LinageInfoAttRefsList; %XML.element.att;
        %XML.link.att; >

<!-- CLASS: CWMXCOBOL:ReportWriterSection -->

<!ENTITY % CWMXCOBOL:ReportWriterSectionProperties '(%CWMXCOBOL:SectionProperties;)'
        >

<!ENTITY % CWMXCOBOL:ReportWriterSectionAssociations
        '(%CWMXCOBOL:SectionAssociations;)' >

<!ENTITY % CWMXCOBOL:ReportWriterSectionCompositions
        '(%CWMXCOBOL:SectionCompositions;)' >

<!ENTITY % CWMXCOBOL:ReportWriterSectionAttPropsList
        '%CWMXCOBOL:SectionAttPropsList;' >

<!ENTITY % CWMXCOBOL:ReportWriterSectionAttRefsList '%CWMXCOBOL:SectionAttRefsList;'
        >

<!ELEMENT CWMXCOBOL:ReportWriterSection (
        %CWMXCOBOL:ReportWriterSectionProperties;
        ,(XML.extension* , %CWMXCOBOL:ReportWriterSectionAssociations; )
        , %CWMXCOBOL:ReportWriterSectionCompositions; )?>

<!ATTLIST CWMXCOBOL:ReportWriterSection %CWMXCOBOL:ReportWriterSectionAttPropsList;
        %CWMXCOBOL:ReportWriterSectionAttRefsList; %XML.element.att;
        %XML.link.att; >

<!-- CLASS: CWMXCOBOL:WorkingStorageSection -->

<!ENTITY % CWMXCOBOL:WorkingStorageSectionProperties
        '(%CWMXCOBOL:SectionProperties;)' >

<!ENTITY % CWMXCOBOL:WorkingStorageSectionAssociations
        '(%CWMXCOBOL:SectionAssociations;)' >

<!ENTITY % CWMXCOBOL:WorkingStorageSectionCompositions
        '(%CWMXCOBOL:SectionCompositions;)' >

<!ENTITY % CWMXCOBOL:WorkingStorageSectionAttPropsList
        '%CWMXCOBOL:SectionAttPropsList;' >

<!ENTITY % CWMXCOBOL:WorkingStorageSectionAttRefsList
        '%CWMXCOBOL:SectionAttRefsList;' >

<!ELEMENT CWMXCOBOL:WorkingStorageSection (
        %CWMXCOBOL:WorkingStorageSectionProperties;
        ,(XML.extension* , %CWMXCOBOL:WorkingStorageSectionAssociations; )
        , %CWMXCOBOL:WorkingStorageSectionCompositions; )?>

<!ATTLIST CWMXCOBOL:WorkingStorageSection
        %CWMXCOBOL:WorkingStorageSectionAttPropsList;
        %CWMXCOBOL:WorkingStorageSectionAttRefsList; %XML.element.att;

```

```

%XML.link.att; >

<!-- CLASS: CWMXCOBOL:FileSection -->
<!ENTITY % CWMXCOBOL:FileSectionProperties '(%CWMXCOBOL:SectionProperties;)' >
<!ENTITY % CWMXCOBOL:FileSectionAssociations '(%CWMXCOBOL:SectionAssociations;)' >
<!ENTITY % CWMXCOBOL:FileSectionCompositions '(%CWMXCOBOL:SectionCompositions;
,CWMXCOBOL:FileSection.cobolFD*)' >
<!ENTITY % CWMXCOBOL:FileSectionAttPropsList '%CWMXCOBOL:SectionAttPropsList;' >
<!ENTITY % CWMXCOBOL:FileSectionAttRefsList '%CWMXCOBOL:SectionAttRefsList;' >
<!ELEMENT CWMXCOBOL:FileSection ( %CWMXCOBOL:FileSectionProperties;
,(XML.extension* , %CWMXCOBOL:FileSectionAssociations; )
,%CWMXCOBOL:FileSectionCompositions; )?>
<!ATTLIST CWMXCOBOL:FileSection %CWMXCOBOL:FileSectionAttPropsList;
%CWMXCOBOL:FileSectionAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXCOBOL:LinkageSection -->
<!ENTITY % CWMXCOBOL:LinkageSectionProperties '(%CWMXCOBOL:SectionProperties;)' >
<!ENTITY % CWMXCOBOL:LinkageSectionAssociations '(%CWMXCOBOL:SectionAssociations;)'
>
<!ENTITY % CWMXCOBOL:LinkageSectionCompositions
'(%CWMXCOBOL:SectionCompositions;)' >
<!ENTITY % CWMXCOBOL:LinkageSectionAttPropsList '%CWMXCOBOL:SectionAttPropsList;' >
<!ENTITY % CWMXCOBOL:LinkageSectionAttRefsList '%CWMXCOBOL:SectionAttRefsList;' >
<!ELEMENT CWMXCOBOL:LinkageSection ( %CWMXCOBOL:LinkageSectionProperties;
,(XML.extension* , %CWMXCOBOL:LinkageSectionAssociations; )
,%CWMXCOBOL:LinkageSectionCompositions; )?>
<!ATTLIST CWMXCOBOL:LinkageSection %CWMXCOBOL:LinkageSectionAttPropsList;
%CWMXCOBOL:LinkageSectionAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXCOBOL:OccursKey -->
<!ELEMENT CWMXCOBOL:OccursKey.isAscending EMPTY>
<!ATTLIST CWMXCOBOL:OccursKey.isAscending
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXCOBOL:OccursKey.occursKeyField (CWMXCOBOL:COBOLField)?>
<!ELEMENT CWMXCOBOL:OccursKey.occursKeyOf (CWMXCOBOL:COBOLField)?>
<!ENTITY % CWMXCOBOL:OccursKeyProperties '(%UML:ModelElementProperties;
,(CWMXCOBOL:OccursKey.isAscending)?)' >
<!ENTITY % CWMXCOBOL:OccursKeyAssociations '(%UML:ModelElementAssociations;
,CWMXCOBOL:OccursKey.occursKeyField?
,CWMXCOBOL:OccursKey.occursKeyOf?)' >
<!ENTITY % CWMXCOBOL:OccursKeyCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXCOBOL:OccursKeyAttPropsList '%UML:ModelElementAttPropsList;

```

```

isAscending (true|false) #IMPLIED ' >
<!ENTITY % CWMXCOBOL:OccursKeyAttRefsList '%UML:ModelElementAttRefsList;
occursKeyField IDREFS #IMPLIED
occursKeyOf IDREFS #IMPLIED' >

<!ELEMENT CWMXCOBOL:OccursKey ( %CWMXCOBOL:OccursKeyProperties;
,(XML.extension* , %CWMXCOBOL:OccursKeyAssociations; )
, %CWMXCOBOL:OccursKeyCompositions; )?>

<!ATTLIST CWMXCOBOL:OccursKey %CWMXCOBOL:OccursKeyAttPropsList;
%CWMXCOBOL:OccursKeyAttRefsList; %XML.element.att;
%XML.link.att; >

```

3.5 DMSII.dtd

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- Expressions.dtd -->
<!-- Model_Management.dtd -->
<!-- Record.dtd -->
<!-- KeysIndexes.dtd -->
<!-- BusinessInformation.dtd -->

<!ATTLIST XMI xmlns:CWMXDMSII CDATA #IMPLIED >

<!-- PACKAGE: CWMXDMSII:DMSII -->

<!-- ***** CWMXDMSII:FieldBits ***** -->

<!ELEMENT CWMXDMSII:DataItem.fieldBit ( CWMXDMSII:FieldBit)* >

<!-- CLASS: CWMXDMSII:Database -->

<!ELEMENT CWMXDMSII:Database.isLogical EMPTY>
<!ATTLIST CWMXDMSII:Database.isLogical
  xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Database.guardFile (#PCDATA|XML.reference)*>
<!ELEMENT CWMXDMSII:Database.source (#PCDATA|XML.reference)*>

<!ENTITY % CWMXDMSII:DatabaseProperties '(%UML:StructuralFeatureProperties;
  ,(UML:GeneralizableElement.isRoot)?
  ,(UML:GeneralizableElement.isLeaf)?
  ,(UML:GeneralizableElement.isAbstract)?
  ,(CWMXDMSII:Database.isLogical)?
  ,(CWMXDMSII:Database.guardFile)?
  ,(CWMXDMSII:Database.source)?)' >

<!ENTITY % CWMXDMSII:DatabaseAssociations '(%UML:StructuralFeatureAssociations;
  ,UML:GeneralizableElement.generalization*
  ,UML:GeneralizableElement.specialization*)' >

<!ENTITY % CWMXDMSII:DatabaseCompositions '(%UML:StructuralFeatureCompositions;
  ,UML:Package.elementImport*
  ,UML:Namespace.ownedElement*)' >

<!ENTITY % CWMXDMSII:DatabaseAttPropsList '%UML:StructuralFeatureAttPropsList;
  isRoot (true|false) #IMPLIED
  isLeaf (true|false) #IMPLIED
  isAbstract (true|false) #IMPLIED
  isLogical (true|false) #IMPLIED
  guardFile CDATA #IMPLIED
  source CDATA #IMPLIED' >

<!ENTITY % CWMXDMSII:DatabaseAttRefsList '%UML:StructuralFeatureAttRefsList;' >

<!ELEMENT CWMXDMSII:Database ( %CWMXDMSII:DatabaseProperties;
  ,(XML.extension* , %CWMXDMSII:DatabaseAssociations; )
  , %CWMXDMSII:DatabaseCompositions; )?>

<!ATTLIST CWMXDMSII:Database %CWMXDMSII:DatabaseAttPropsList;
  %CWMXDMSII:DatabaseAttRefsList; %XML.element.att; %XML.link.att;
  >
```

```

<!-- CLASS: CWMXDMSII:Remap -->

<!ELEMENT CWMXDMSII:Remap.isRequiredAll EMPTY>
<!-- ATTLIST CWMXDMSII:Remap.isRequiredAll
      xmi.value ( true | false ) #REQUIRED -->

<!ELEMENT CWMXDMSII:Remap.isReadOnlyAll EMPTY>
<!-- ATTLIST CWMXDMSII:Remap.isReadOnlyAll
      xmi.value ( true | false ) #REQUIRED -->

<!ELEMENT CWMXDMSII:Remap.isGivingException EMPTY>
<!-- ATTLIST CWMXDMSII:Remap.isGivingException
      xmi.value ( true | false ) #REQUIRED -->

<!ELEMENT CWMXDMSII:Remap.selectCondition (UML:Expression
|UML:ObjectSetExpression
|UML:TimeExpression
|UML:BooleanExpression
|UML>ActionExpression
|UML:IterationExpression
|UML:TypeExpression
|UML:ArgListsExpression
|UML:MappingExpression
|UML:ProcedureExpression)>

<!ELEMENT CWMXDMSII:Remap.structure (UML:StructuralFeature)?>

<!-- ENTITY % CWMXDMSII:RemapProperties (%CWMREC:RecordDefProperties;
      , (UML:StructuralFeature.multiplicity)?
      , (UML:StructuralFeature.changeability)?
      , (UML:StructuralFeature.targetScope)?
      , (UML:Feature.ownerScope)?
      , (CWMXDMSII:Remap.isRequiredAll)?
      , (CWMXDMSII:Remap.isReadOnlyAll)?
      , (CWMXDMSII:Remap.isGivingException)?
      , (CWMXDMSII:Remap.selectCondition)? ) -->

<!-- ENTITY % CWMXDMSII:RemapAssociations (%CWMREC:RecordDefAssociations;
      , UML:StructuralFeature.type?
      , UML:Feature.owner?
      , CWMXDMSII:Remap.structure?) -->

<!-- ENTITY % CWMXDMSII:RemapCompositions (%CWMREC:RecordDefCompositions;) -->

<!-- ENTITY % CWMXDMSII:RemapAttPropsList (%CWMREC:RecordDefAttPropsList;
      multiplicity CDATA #IMPLIED
      changeability ( changeable| frozen| addOnly ) #IMPLIED
      targetScope ( instance| classifier ) #IMPLIED
      ownerScope ( instance| classifier ) #IMPLIED
      isRequiredAll (true|false) #IMPLIED
      isReadOnlyAll (true|false) #IMPLIED
      isGivingException (true|false) #IMPLIED
      selectCondition CDATA #IMPLIED -->

<!-- ENTITY % CWMXDMSII:RemapAttRefsList (%CWMREC:RecordDefAttRefsList;
      type IDREFS #IMPLIED
      owner IDREFS #IMPLIED
      structure IDREFS #IMPLIED -->

<!-- ELEMENT CWMXDMSII:Remap ( %CWMXDMSII:RemapProperties;
      , (XML.extension* , %CWMXDMSII:RemapAssociations; )
      , %CWMXDMSII:RemapCompositions; )? -->

<!-- ATTLIST CWMXDMSII:Remap %CWMXDMSII:RemapAttPropsList;
      %CWMXDMSII:RemapAttRefsList; %XML.element.att; %XML.link.att; -->

```

```

<!-- CLASS: CWMXDMSII:DataSet -->

<!ELEMENT CWMXDMSII:DataSet.isGlobal EMPTY>
<!ATTLIST CWMXDMSII:DataSet.isGlobal
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:DataSet.organization (#PCDATA|XML.reference)*>

<!ELEMENT CWMXDMSII:DataSet.reorganize (#PCDATA|XML.reference)*>

<!ELEMENT CWMXDMSII:DataSet.isRequiredAll EMPTY>
<!ATTLIST CWMXDMSII:DataSet.isRequiredAll
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:DataSet.partitioningSet (CWMXDMSII:Set)?>

<!ENTITY % CWMXDMSII:DataSetProperties '(%UML:StructuralFeatureProperties;
    ,CWMREC:RecordDef.fieldDelimiter)?
    ,CWMREC:RecordDef.isFixedWidth)?
    ,CWMREC:RecordDef.recordDelimiter)?
    ,CWMREC:RecordDef.textDelimiter)?
    ,UML:Class.isActive)?
    ,UML:GeneralizableElement.isRoot)?
    ,UML:GeneralizableElement.isLeaf)?
    ,UML:GeneralizableElement.isAbstract)?
    ,CWMXDMSII:DataSet.isGlobal)?
    ,CWMXDMSII:DataSet.organization)?
    ,CWMXDMSII:DataSet.reorganize)?
    ,CWMXDMSII:DataSet.isRequiredAll)?' >

<!ENTITY % CWMXDMSII:DataSetAssociations '(%UML:StructuralFeatureAssociations;
    ,CWMREC:RecordDef.file*
    ,UML:Classifier.participant*
    ,UML:Classifier.powertypeRange*
    ,UML:GeneralizableElement.generalization*
    ,UML:GeneralizableElement.specialization*
    ,CWMXDMSII:DataSet.partitioningSet?)*' >

<!ENTITY % CWMXDMSII:DataSetCompositions '(%UML:StructuralFeatureCompositions;
    ,UML:Classifier.feature*
    ,UML:Namespace.ownedElement*)' >

<!ENTITY % CWMXDMSII:DataSetAttPropsList '%UML:StructuralFeatureAttPropsList;
    fieldDelimiter CDATA #IMPLIED
    isFixedWidth (true|false) #IMPLIED
    isActive (true|false) #IMPLIED
    isRoot (true|false) #IMPLIED
    isLeaf (true|false) #IMPLIED
    isAbstract (true|false) #IMPLIED
    isGlobal (true|false) #IMPLIED
    organization CDATA #IMPLIED
    reorganize CDATA #IMPLIED
    isRequiredAll (true|false) #IMPLIED' >

<!ENTITY % CWMXDMSII:DataSetAttRefsList '%UML:StructuralFeatureAttRefsList;
    partitioningSet IDREFS #IMPLIED' >

<!ELEMENT CWMXDMSII:DataSet ( %CWMXDMSII:DataSetProperties;
    , (XML:extension* , %CWMXDMSII:DataSetAssociations; )
    , %CWMXDMSII:DataSetCompositions; )?>

<!ATTLIST CWMXDMSII:DataSet %CWMXDMSII:DataSetAttPropsList;
    %CWMXDMSII:DataSetAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:DataItem -->

<!ELEMENT CWMXDMSII:DataItem.nullValue (CWM:ExpressionNode

```



```

|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>

<!ELEMENT CWMXDMSII:Dataltem.isRequired EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isRequired
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.size (#PCDATA|XMI.reference)*>

<!ELEMENT CWMXDMSII:Dataltem.scaleFactor (#PCDATA|XMI.reference)*>

<!ELEMENT CWMXDMSII:Dataltem.isSigned EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isSigned
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.occurs (#PCDATA|XMI.reference)*>

<!ELEMENT CWMXDMSII:Dataltem.isVirtual EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isVirtual
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.virtualExpression (CWM:ExpressionNode
|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>

<!ELEMENT CWMXDMSII:Dataltem.isKanji EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isKanji
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.ccsVersion (#PCDATA|XMI.reference)*>

<!ELEMENT CWMXDMSII:Dataltem.isGemcosLiteral EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isGemcosLiteral
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isGemcosData EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isGemcosData
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isGemcosSSN EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isGemcosSSN
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isGemcosDBSN EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isGemcosDBSN
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isComsProgram EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isComsProgram
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isComsID EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isComsID
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isComsLocator EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isComsLocator
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.isComsOutpQ EMPTY>
<!ATTLIST CWMXDMSII:Dataltem.isComsOutpQ
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXDMSII:Dataltem.occursDataltem (CWMXDMSII:Dataltem)?>

<!ELEMENT CWMXDMSII:Dataltem.occuringDataltem (CWMXDMSII:Dataltem)*>

```

```

<!ELEMENT CWMXDMSII:DataItem.keyDataSet (CWMXDMSII:Set)*>

<!ELEMENT CWMXDMSII:DataItem.structure (UML:StructuralFeature)?>

<!ELEMENT CWMXDMSII:DataItem.remapItem (CWMXDMSII:RemapItem)*>

<!ENTITY % CWMXDMSII:DataItemProperties '(%CWMREC:FieldProperties;
    ,CWMXDMSII:DataItem.nullValue)?
    ,CWMXDMSII:DataItem.isRequired)?
    ,CWMXDMSII:DataItem.size)?
    ,CWMXDMSII:DataItem.scaleFactor)?
    ,CWMXDMSII:DataItem.isSigned)?
    ,CWMXDMSII:DataItem.occurs)?
    ,CWMXDMSII:DataItem.isVirtual)?
    ,CWMXDMSII:DataItem.virtualExpression)?
    ,CWMXDMSII:DataItem.isKanji)?
    ,CWMXDMSII:DataItem.ccsVersion)?
    ,CWMXDMSII:DataItem.isGemcosLiteral)?
    ,CWMXDMSII:DataItem.isGemcosData)?
    ,CWMXDMSII:DataItem.isGemcosSSN)?
    ,CWMXDMSII:DataItem.isGemcosDBSN)?
    ,CWMXDMSII:DataItem.isComsProgram)?
    ,CWMXDMSII:DataItem.isComsID)?
    ,CWMXDMSII:DataItem.isComsLocator)?
    ,CWMXDMSII:DataItem.isComsOutpQ)?)' >

<!ENTITY % CWMXDMSII:DataItemAssociations '(%CWMREC:FieldAssociations;
    ,CWMXDMSII:DataItem.occursDataItem?
    ,CWMXDMSII:DataItem.occuringDataItem*
    ,CWMXDMSII:DataItem.keyDataSet*
    ,CWMXDMSII:DataItem.structure?
    ,CWMXDMSII:DataItem.remapItem*)' >

<!ENTITY % CWMXDMSII:DataItemCompositions '(%CWMREC:FieldCompositions;
    ,CWMXDMSII:DataItem.fieldBit*)' >

<!ENTITY % CWMXDMSII:DataItemAttPropsList '%CWMREC:FieldAttPropsList;
    nullValue CDATA #IMPLIED
    isRequired (true|false) #IMPLIED
    size CDATA #IMPLIED
    scaleFactor CDATA #IMPLIED
    isSigned (true|false) #IMPLIED
    occurs CDATA #IMPLIED
    isVirtual (true|false) #IMPLIED
    virtualExpression CDATA #IMPLIED
    isKanji (true|false) #IMPLIED
    ccsVersion CDATA #IMPLIED
    isGemcosLiteral (true|false) #IMPLIED
    isGemcosData (true|false) #IMPLIED
    isGemcosSSN (true|false) #IMPLIED
    isGemcosDBSN (true|false) #IMPLIED
    isComsProgram (true|false) #IMPLIED
    isComsID (true|false) #IMPLIED
    isComsLocator (true|false) #IMPLIED
    isComsOutpQ (true|false) #IMPLIED' >

<!ENTITY % CWMXDMSII:DataItemAttRefsList '%CWMREC:FieldAttRefsList;
    occursDataItem IDREFS #IMPLIED
    structure IDREFS #IMPLIED' >

<!ELEMENT CWMXDMSII:DataItem ( %CWMXDMSII:DataItemProperties;
    ,XML:extension* , %CWMXDMSII:DataItemAssociations; )
    , %CWMXDMSII:DataItemCompositions; )?>

<!ATTLIST CWMXDMSII:DataItem %CWMXDMSII:DataItemAttPropsList;
    %CWMXDMSII:DataItemAttRefsList; %XML:element.att; %XML:link.att; >

```

```

<!-- CLASS: CWMXDMSII:VariableFormatPart -->
<!ELEMENT CWMXDMSII:VariableFormatPart.vfLabel (#PCDATA|XML.reference)*>
<!ELEMENT CWMXDMSII:VariableFormatPart.selectCondition (UML:BooleanExpression)>
<!ENTITY % CWMXDMSII:VariableFormatPartProperties '(%UML:StructuralFeatureProperties;
,(CWMREC:RecordDef.fieldDelimiter)?
,(CWMREC:RecordDef.isFixedWidth)?
,(CWMREC:RecordDef.recordDelimiter)?
,(CWMREC:RecordDef.textDelimiter)?
,(UML:Class.isActive)?
,(UML:GeneralizableElement.isRoot)?
,(UML:GeneralizableElement.isLeaf)?
,(UML:GeneralizableElement.isAbstract)?
,(CWMXDMSII:VariableFormatPart.vfLabel)?
,(CWMXDMSII:VariableFormatPart.selectCondition)?)' >
<!ENTITY % CWMXDMSII:VariableFormatPartAssociations '(%UML:StructuralFeatureAssociations;
,CWMREC:RecordDef.file*
,UML:Classifier.participant*
,UML:Classifier.powerTypeRange*
,UML:GeneralizableElement.generalization*
,UML:GeneralizableElement.specialization*)' >
<!ENTITY % CWMXDMSII:VariableFormatPartCompositions
'(%UML:StructuralFeatureCompositions;
,UML:Classifier.feature*
,UML:Namespace.ownedElement*)' >
<!ENTITY % CWMXDMSII:VariableFormatPartAttPropsList '%UML:StructuralFeatureAttPropsList;
fieldDelimiter CDATA #IMPLIED
isFixedWidth (true|false) #IMPLIED
isActive (true|false) #IMPLIED
isRoot (true|false) #IMPLIED
isLeaf (true|false) #IMPLIED
isAbstract (true|false) #IMPLIED
vfLabel CDATA #IMPLIED
selectCondition CDATA #IMPLIED' >
<!ENTITY % CWMXDMSII:VariableFormatPartAttRefsList '%UML:StructuralFeatureAttRefsList;' >
<!ELEMENT CWMXDMSII:VariableFormatPart ( %CWMXDMSII:VariableFormatPartProperties;
,(XML.extension* , %CWMXDMSII:VariableFormatPartAssociations; )
,%CWMXDMSII:VariableFormatPartCompositions; )?>
<!ATTLIST CWMXDMSII:VariableFormatPart %CWMXDMSII:VariableFormatPartAttPropsList;
%CWMXDMSII:VariableFormatPartAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXDMSII:SetStructure -->
<!ELEMENT CWMXDMSII:SetStructure.duplicates (#PCDATA|XML.reference)*>
<!ENTITY % CWMXDMSII:SetStructureProperties '(%CWM:IndexProperties;
,(UML:StructuralFeature.multiplicity)?
,(UML:StructuralFeature.changeability)?
,(UML:StructuralFeature.targetScope)?
,(UML:Feature.ownerScope)?
,(CWMXDMSII:SetStructure.duplicates)?)' >
<!ENTITY % CWMXDMSII:SetStructureAssociations '(%CWM:IndexAssociations;
,UML:StructuralFeature.type?
,UML:Feature.owner?)' >
<!ENTITY % CWMXDMSII:SetStructureCompositions '(%CWM:IndexCompositions;)' >

```

```

<!ENTITY % CWMXDMSII:SetStructureAttPropsList '%CWM:IndexAttPropsList;
multiplicity CDATA #IMPLIED
changeability ( changeable| frozen| addOnly ) #IMPLIED
targetScope ( instance| classifier ) #IMPLIED
ownerScope ( instance| classifier ) #IMPLIED
duplicates CDATA #IMPLIED' >

<!ENTITY % CWMXDMSII:SetStructureAttRefsList '%CWM:IndexAttRefsList;
type IDREFS #IMPLIED
owner IDREFS #IMPLIED' >

<!ELEMENT CWMXDMSII:SetStructure ( %CWMXDMSII:SetStructureProperties;
,(XML.extension* , %CWMXDMSII:SetStructureAssociations; )
, %CWMXDMSII:SetStructureCompositions; )?>

<!ATTLIST CWMXDMSII:SetStructure %CWMXDMSII:SetStructureAttPropsList;
%CWMXDMSII:SetStructureAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXDMSII:Set -->

<!ELEMENT CWMXDMSII:Set.setType (#PCDATA|XML.reference)*>
<!ELEMENT CWMXDMSII:Set.reorganize (#PCDATA|XML.reference)*>
<!ELEMENT CWMXDMSII:Set.partitioningSet (CWMXDMSII:Set)?>
<!ELEMENT CWMXDMSII:Set.partitionedSet (CWMXDMSII:Set)*>
<!ELEMENT CWMXDMSII:Set.keyDataItem (CWMXDMSII:DataItem)*>
<!ELEMENT CWMXDMSII:Set.partitionedDataSet (CWMXDMSII:DataSet)?>

<!ENTITY % CWMXDMSII:SetProperties '(%CWMXDMSII:SetStructureProperties;
,(CWMXDMSII:Set.setType)?
,(CWMXDMSII:Set.reorganize)?)' >

<!ENTITY % CWMXDMSII:SetAssociations '(%CWMXDMSII:SetStructureAssociations;
,CWMXDMSII:Set.partitioningSet?
,CWMXDMSII:Set.partitionedSet*
,CWMXDMSII:Set.keyDataItem*
,CWMXDMSII:Set.partitionedDataSet?)' >

<!ENTITY % CWMXDMSII:SetCompositions '(%CWMXDMSII:SetStructureCompositions;)' >

<!ENTITY % CWMXDMSII:SetAttPropsList '%CWMXDMSII:SetStructureAttPropsList;
setType CDATA #IMPLIED
reorganize CDATA #IMPLIED' >

<!ENTITY % CWMXDMSII:SetAttRefsList '%CWMXDMSII:SetStructureAttRefsList;
partitioningSet IDREFS #IMPLIED
partitionedDataSet IDREFS #IMPLIED' >

<!ELEMENT CWMXDMSII:Set ( %CWMXDMSII:SetProperties;
,(XML.extension* , %CWMXDMSII:SetAssociations; )
, %CWMXDMSII:SetCompositions; )?>

<!ATTLIST CWMXDMSII:Set %CWMXDMSII:SetAttPropsList; %CWMXDMSII:SetAttRefsList;
%XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:Access -->

<!ENTITY % CWMXDMSII:AccessProperties '(%CWMXDMSII:SetStructureProperties;)' >

<!ENTITY % CWMXDMSII:AccessAssociations '(%CWMXDMSII:SetStructureAssociations;)' >

```

```

<!ENTITY % CWMXDMSII:AccessCompositions '(%CWMXDMSII:SetStructureCompositions;)' >
<!ENTITY % CWMXDMSII:AccessAttPropsList '%CWMXDMSII:SetStructureAttPropsList;' >
<!ENTITY % CWMXDMSII:AccessAttRefsList '%CWMXDMSII:SetStructureAttRefsList;' >
<!ELEMENT CWMXDMSII:Access ( %CWMXDMSII:AccessProperties;
    ,(XML.extension* , %CWMXDMSII:AccessAssociations; )
    , %CWMXDMSII:AccessCompositions; )?>
<!ATTLIST CWMXDMSII:Access %CWMXDMSII:AccessAttPropsList;
    %CWMXDMSII:AccessAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:Subset -->
<!ENTITY % CWMXDMSII:SubsetProperties '(%CWMXDMSII:SetProperties;)' >
<!ENTITY % CWMXDMSII:SubsetAssociations '(%CWMXDMSII:SetAssociations;)' >
<!ENTITY % CWMXDMSII:SubsetCompositions '(%CWMXDMSII:SetCompositions;)' >
<!ENTITY % CWMXDMSII:SubsetAttPropsList '%CWMXDMSII:SetAttPropsList;' >
<!ENTITY % CWMXDMSII:SubsetAttRefsList '%CWMXDMSII:SetAttRefsList;' >
<!ELEMENT CWMXDMSII:Subset ( %CWMXDMSII:SubsetProperties;
    ,(XML.extension* , %CWMXDMSII:SubsetAssociations; )
    , %CWMXDMSII:SubsetCompositions; )?>
<!ATTLIST CWMXDMSII:Subset %CWMXDMSII:SubsetAttPropsList;
    %CWMXDMSII:SubsetAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:AutomaticSubset -->
<!ELEMENT CWMXDMSII:AutomaticSubset.condition (UML:BooleanExpression)>
<!ENTITY % CWMXDMSII:AutomaticSubsetProperties '(%CWMXDMSII:SubsetProperties;
    ,(CWMXDMSII:AutomaticSubset.condition)?)' >
<!ENTITY % CWMXDMSII:AutomaticSubsetAssociations '(%CWMXDMSII:SubsetAssociations;)' >
<!ENTITY % CWMXDMSII:AutomaticSubsetCompositions '(%CWMXDMSII:SubsetCompositions;)'
    >
<!ENTITY % CWMXDMSII:AutomaticSubsetAttPropsList '%CWMXDMSII:SubsetAttPropsList;
    condition CDATA #IMPLIED' >
<!ENTITY % CWMXDMSII:AutomaticSubsetAttRefsList '%CWMXDMSII:SubsetAttRefsList;' >
<!ELEMENT CWMXDMSII:AutomaticSubset ( %CWMXDMSII:AutomaticSubsetProperties;
    ,(XML.extension* , %CWMXDMSII:AutomaticSubsetAssociations; )
    , %CWMXDMSII:AutomaticSubsetCompositions; )?>
<!ATTLIST CWMXDMSII:AutomaticSubset %CWMXDMSII:AutomaticSubsetAttPropsList;
    %CWMXDMSII:AutomaticSubsetAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXDMSII:KeyItem -->
<!ELEMENT CWMXDMSII:KeyItem.collation (#PCDATA|XML.reference)*>
<!ENTITY % CWMXDMSII:KeyItemProperties '(%CWM:IndexedFeatureProperties;
    ,(CWMXDMSII:KeyItem.collation)?)' >

```

```

<!ENTITY % CWMXDMSII:KeyItemAssociations '(%CWM:IndexedFeatureAssociations;)' >
<!ENTITY % CWMXDMSII:KeyItemCompositions '(%CWM:IndexedFeatureCompositions;)' >
<!ENTITY % CWMXDMSII:KeyItemAttPropsList '%CWM:IndexedFeatureAttPropsList;
collation CDATA #IMPLIED' >
<!ENTITY % CWMXDMSII:KeyItemAttRefsList '%CWM:IndexedFeatureAttRefsList;' >
<!ELEMENT CWMXDMSII:KeyItem ( %CWMXDMSII:KeyItemProperties;
,(XML.extension* , %CWMXDMSII:KeyItemAssociations; )
, %CWMXDMSII:KeyItemCompositions; )?>
<!ATTLIST CWMXDMSII:KeyItem %CWMXDMSII:KeyItemAttPropsList;
%CWMXDMSII:KeyItemAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:RemapItem -->
<!ELEMENT CWMXDMSII:RemapItem.occurs (#PCDATA|XML.reference)*>
<!ELEMENT CWMXDMSII:RemapItem.isRequired EMPTY>
<!ATTLIST CWMXDMSII:RemapItem.isRequired
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXDMSII:RemapItem.isHidden EMPTY>
<!ATTLIST CWMXDMSII:RemapItem.isHidden
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXDMSII:RemapItem.isReadOnly EMPTY>
<!ATTLIST CWMXDMSII:RemapItem.isReadOnly
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXDMSII:RemapItem.isGivingException EMPTY>
<!ATTLIST CWMXDMSII:RemapItem.isGivingException
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXDMSII:RemapItem.isVirtual EMPTY>
<!ATTLIST CWMXDMSII:RemapItem.isVirtual
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXDMSII:RemapItem.virtualExpression (CWM:ExpressionNode
|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>
<!ELEMENT CWMXDMSII:RemapItem.dataItem (CWMXDMSII:DataItem)*>
<!ENTITY % CWMXDMSII:RemapItemProperties '(%CWMREC:FieldProperties;
,(CWMXDMSII:RemapItem.occurs)?
,(CWMXDMSII:RemapItem.isRequired)?
,(CWMXDMSII:RemapItem.isHidden)?
,(CWMXDMSII:RemapItem.isReadOnly)?
,(CWMXDMSII:RemapItem.isGivingException)?
,(CWMXDMSII:RemapItem.isVirtual)?
,(CWMXDMSII:RemapItem.virtualExpression)?)' >
<!ENTITY % CWMXDMSII:RemapItemAssociations '(%CWMREC:FieldAssociations;
,CWMXDMSII:RemapItem.dataItem*)' >
<!ENTITY % CWMXDMSII:RemapItemCompositions '(%CWMREC:FieldCompositions;)' >
<!ENTITY % CWMXDMSII:RemapItemAttPropsList '%CWMREC:FieldAttPropsList;
occurs CDATA #IMPLIED
isRequired (true|false) #IMPLIED
isHidden (true|false) #IMPLIED
isReadOnly (true|false) #IMPLIED
isGivingException (true|false) #IMPLIED' >

```

```

isVirtual (true|false) #IMPLIED
virtualExpression CDATA #IMPLIED' >

<!ENTITY % CWMXDMSII:RemapItemAttRefsList '%CWMREC:FieldAttRefsList;' >

<!ELEMENT CWMXDMSII:RemapItem ( %CWMXDMSII:RemapItemProperties;
,(XML.extension* , %CWMXDMSII:RemapItemAssociations; )
, %CWMXDMSII:RemapItemCompositions; )?>

<!ATTLIST CWMXDMSII:RemapItem %CWMXDMSII:RemapItemAttPropsList;
%CWMXDMSII:RemapItemAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXDMSII:FieldBit -->

<!ELEMENT CWMXDMSII:FieldBit.dataItem (CWMXDMSII:DataItem)?>

<!ENTITY % CWMXDMSII:FieldBitProperties '(%UML:ModelElementProperties;)' >

<!ENTITY % CWMXDMSII:FieldBitAssociations '(%UML:ModelElementAssociations;
,CWMXDMSII:FieldBit.dataItem?)' >

<!ENTITY % CWMXDMSII:FieldBitCompositions '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXDMSII:FieldBitAttPropsList '%UML:ModelElementAttPropsList;' >

<!ENTITY % CWMXDMSII:FieldBitAttRefsList '%UML:ModelElementAttRefsList;
dataItem IDREFS #IMPLIED' >

<!ELEMENT CWMXDMSII:FieldBit ( %CWMXDMSII:FieldBitProperties;
,(XML.extension* , %CWMXDMSII:FieldBitAssociations; )
, %CWMXDMSII:FieldBitCompositions; )?>

<!ATTLIST CWMXDMSII:FieldBit %CWMXDMSII:FieldBitAttPropsList;
%CWMXDMSII:FieldBitAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:Remark -->

<!ELEMENT CWMXDMSII:Remark.text (#PCDATA|XML.reference)*>

<!ENTITY % CWMXDMSII:RemarkProperties '(%UML:StructuralFeatureProperties;
,CWMXDMSII:Remark.text?)' >

<!ENTITY % CWMXDMSII:RemarkAssociations '(%UML:StructuralFeatureAssociations;)' >

<!ENTITY % CWMXDMSII:RemarkCompositions '(%UML:StructuralFeatureCompositions;)' >

<!ENTITY % CWMXDMSII:RemarkAttPropsList '%UML:StructuralFeatureAttPropsList;
text CDATA #IMPLIED' >

<!ENTITY % CWMXDMSII:RemarkAttRefsList '%UML:StructuralFeatureAttRefsList;' >

<!ELEMENT CWMXDMSII:Remark ( %CWMXDMSII:RemarkProperties;
,(XML.extension* , %CWMXDMSII:RemarkAssociations; )
, %CWMXDMSII:RemarkCompositions; )?>

<!ATTLIST CWMXDMSII:Remark %CWMXDMSII:RemarkAttPropsList;
%CWMXDMSII:RemarkAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXDMSII:PhysicalDatabase -->

<!ENTITY % CWMXDMSII:PhysicalDatabaseProperties '(%UML:PackageProperties;)' >

<!ENTITY % CWMXDMSII:PhysicalDatabaseAssociations '(%UML:PackageAssociations;)' >

```

```

<!ENTITY % CWMXDMSII:PhysicalDatabaseCompositions '(%UML:PackageCompositions;)' >
<!ENTITY % CWMXDMSII:PhysicalDatabaseAttPropsList '(%UML:PackageAttPropsList;)' >
<!ENTITY % CWMXDMSII:PhysicalDatabaseAttRefsList '(%UML:PackageAttRefsList;)' >
<!ELEMENT CWMXDMSII:PhysicalDatabase ( %CWMXDMSII:PhysicalDatabaseProperties;
,(XML.extension* , %CWMXDMSII:PhysicalDatabaseAssociations; )
, %CWMXDMSII:PhysicalDatabaseCompositions; )?>
<!ATTLIST CWMXDMSII:PhysicalDatabase %CWMXDMSII:PhysicalDatabaseAttPropsList;
          %CWMXDMSII:PhysicalDatabaseAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXDMSII:PhysicalDataSet -->
<!ENTITY % CWMXDMSII:PhysicalDataSetProperties '(%UML:ModelElementProperties;)' >
<!ENTITY % CWMXDMSII:PhysicalDataSetAssociations '(%UML:ModelElementAssociations;)' >
<!ENTITY % CWMXDMSII:PhysicalDataSetCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXDMSII:PhysicalDataSetAttPropsList '(%UML:ModelElementAttPropsList;)' >
<!ENTITY % CWMXDMSII:PhysicalDataSetAttRefsList '(%UML:ModelElementAttRefsList;)' >
<!ELEMENT CWMXDMSII:PhysicalDataSet ( %CWMXDMSII:PhysicalDataSetProperties;
,(XML.extension* , %CWMXDMSII:PhysicalDataSetAssociations; )
, %CWMXDMSII:PhysicalDataSetCompositions; )?>
<!ATTLIST CWMXDMSII:PhysicalDataSet %CWMXDMSII:PhysicalDataSetAttPropsList;
          %CWMXDMSII:PhysicalDataSetAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXDMSII:DASDLComment -->
<!ENTITY % CWMXDMSII:DASDLCommentProperties '(%CWM:DescriptionProperties;)' >
<!ENTITY % CWMXDMSII:DASDLCommentAssociations '(%CWM:DescriptionAssociations;)' >
<!ENTITY % CWMXDMSII:DASDLCommentCompositions '(%CWM:DescriptionCompositions;)' >
<!ENTITY % CWMXDMSII:DASDLCommentAttPropsList '(%CWM:DescriptionAttPropsList;)' >
<!ENTITY % CWMXDMSII:DASDLCommentAttRefsList '(%CWM:DescriptionAttRefsList;)' >
<!ELEMENT CWMXDMSII:DASDLComment ( %CWMXDMSII:DASDLCommentProperties;
,(XML.extension* , %CWMXDMSII:DASDLCommentAssociations; )
, %CWMXDMSII:DASDLCommentCompositions; )?>
<!ATTLIST CWMXDMSII:DASDLComment %CWMXDMSII:DASDLCommentAttPropsList;
          %CWMXDMSII:DASDLCommentAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXDMSII:PhysicalSet -->
<!ENTITY % CWMXDMSII:PhysicalSetProperties '(%UML:ModelElementProperties;)' >
<!ENTITY % CWMXDMSII:PhysicalSetAssociations '(%UML:ModelElementAssociations;)' >
<!ENTITY % CWMXDMSII:PhysicalSetCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXDMSII:PhysicalSetAttPropsList '(%UML:ModelElementAttPropsList;)' >
<!ENTITY % CWMXDMSII:PhysicalSetAttRefsList '(%UML:ModelElementAttRefsList;)' >

```



```

<!ELEMENT CWMXDMSII:PhysicalSet ( %CWMXDMSII:PhysicalSetProperties;
    ,(XML.extension* , %CWMXDMSII:PhysicalSetAssociations; )
    , %CWMXDMSII:PhysicalSetCompositions; )?>

<!ATTLIST CWMXDMSII:PhysicalSet %CWMXDMSII:PhysicalSetAttPropsList;
    %CWMXDMSII:PhysicalSetAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXDMSII:PhysicalDataSetOverride -->

<!ENTITY % CWMXDMSII:PhysicalDataSetOverrideProperties '(%UML:FeatureProperties;)' >
<!ENTITY % CWMXDMSII:PhysicalDataSetOverrideAssociations '(%UML:FeatureAssociations;)' >
<!ENTITY % CWMXDMSII:PhysicalDataSetOverrideCompositions '(%UML:FeatureCompositions;)'
    >

<!ENTITY % CWMXDMSII:PhysicalDataSetOverrideAttPropsList '%UML:FeatureAttPropsList;' >
<!ENTITY % CWMXDMSII:PhysicalDataSetOverrideAttRefsList '%UML:FeatureAttRefsList;' >

<!ELEMENT CWMXDMSII:PhysicalDataSetOverride (
    %CWMXDMSII:PhysicalDataSetOverrideProperties;
    ,(XML.extension* , %CWMXDMSII:PhysicalDataSetOverrideAssociations; )
    , %CWMXDMSII:PhysicalDataSetOverrideCompositions; )?>

<!ATTLIST CWMXDMSII:PhysicalDataSetOverride
    %CWMXDMSII:PhysicalDataSetOverrideAttPropsList;
    %CWMXDMSII:PhysicalDataSetOverrideAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXDMSII:PhysicalSetOverride -->

<!ENTITY % CWMXDMSII:PhysicalSetOverrideProperties '(%UML:FeatureProperties;)' >
<!ENTITY % CWMXDMSII:PhysicalSetOverrideAssociations '(%UML:FeatureAssociations;)' >
<!ENTITY % CWMXDMSII:PhysicalSetOverrideCompositions '(%UML:FeatureCompositions;)' >
<!ENTITY % CWMXDMSII:PhysicalSetOverrideAttPropsList '%UML:FeatureAttPropsList;' >
<!ENTITY % CWMXDMSII:PhysicalSetOverrideAttRefsList '%UML:FeatureAttRefsList;' >

<!ELEMENT CWMXDMSII:PhysicalSetOverride ( %CWMXDMSII:PhysicalSetOverrideProperties;
    ,(XML.extension* , %CWMXDMSII:PhysicalSetOverrideAssociations; )
    , %CWMXDMSII:PhysicalSetOverrideCompositions; )?>

<!ATTLIST CWMXDMSII:PhysicalSetOverride %CWMXDMSII:PhysicalSetOverrideAttPropsList;
    %CWMXDMSII:PhysicalSetOverrideAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXDMSII:PhysicalAccessOverride -->

<!ENTITY % CWMXDMSII:PhysicalAccessOverrideProperties '(%UML:FeatureProperties;)' >
<!ENTITY % CWMXDMSII:PhysicalAccessOverrideAssociations '(%UML:FeatureAssociations;)' >
<!ENTITY % CWMXDMSII:PhysicalAccessOverrideCompositions '(%UML:FeatureCompositions;)'
    >

<!ENTITY % CWMXDMSII:PhysicalAccessOverrideAttPropsList '%UML:FeatureAttPropsList;' >
<!ENTITY % CWMXDMSII:PhysicalAccessOverrideAttRefsList '%UML:FeatureAttRefsList;' >

```

```

<!ELEMENT CWMXDMSII:PhysicalAccessOverride (
    %CWMXDMSII:PhysicalAccessOverrideProperties;
    ,(XML.extension* , %CWMXDMSII:PhysicalAccessOverrideAssociations; )
    , %CWMXDMSII:PhysicalAccessOverrideCompositions; )?>

<!ATTLIST CWMXDMSII:PhysicalAccessOverride
    %CWMXDMSII:PhysicalAccessOverrideAttPropsList;
    %CWMXDMSII:PhysicalAccessOverrideAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXDMSII:DASDLProperty -->

<!ELEMENT CWMXDMSII:DASDLProperty.text (#PCDATA|XML.reference)*>

<!ELEMENT CWMXDMSII:DASDLProperty.owner (UML:ModelElement)?>

<!ENTITY % CWMXDMSII:DASDLPropertyProperties '(%UML:ModelElementProperties;
    ,(CWMXDMSII:DASDLProperty.text)?)' >

<!ENTITY % CWMXDMSII:DASDLPropertyAssociations '(%UML:ModelElementAssociations;
    ,CWMXDMSII:DASDLProperty.owner?)' >

<!ENTITY % CWMXDMSII:DASDLPropertyCompositions '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXDMSII:DASDLPropertyAttPropsList '%UML:ModelElementAttPropsList;
    text CDATA #IMPLIED' >

<!ENTITY % CWMXDMSII:DASDLPropertyAttRefsList '%UML:ModelElementAttRefsList;
    owner IDREFS #IMPLIED' >

<!ELEMENT CWMXDMSII:DASDLProperty ( %CWMXDMSII:DASDLPropertyProperties;
    ,(XML.extension* , %CWMXDMSII:DASDLPropertyAssociations; )
    , %CWMXDMSII:DASDLPropertyCompositions; )?>

<!ATTLIST CWMXDMSII:DASDLProperty %CWMXDMSII:DASDLPropertyAttPropsList;
    %CWMXDMSII:DASDLPropertyAttRefsList; %XML.element.att;
    %XML.link.att; >

```

3.6 Essbase.dtd

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- Expressions.dtd -->
<!-- Multidimensional.dtd -->
<!-- Model_Management.dtd -->
<!-- SoftwareDeployment.dtd -->
<!-- Olap.dtd -->

<!ATTLIST XMI xmlns:CWMXESSB CDATA #IMPLIED >

<!-- PACKAGE: CWMXESSB:Essbase -->

<!ENTITY % CWMXESSB:DimensionType
    ' xmi.value ( none| accounts| time| country| currencyPartition| attribute ) #REQUIRED'>

<!-- ***** CWMXESSB:DatabaseOwnsOutline ***** -->

<!ELEMENT CWMXESSB:Database.outline ( CWMXESSB:Outline)? >

<!-- CLASS: CWMXESSB:Alias -->

<!ENTITY % CWMXESSB:AliasProperties '(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:AliasAssociations '(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:AliasCompositions '(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:AliasAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:AliasAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >

<!ELEMENT CWMXESSB:Alias ( %CWMXESSB:AliasProperties;
    ,(XMI.extension* , %CWMXESSB:AliasAssociations; )
    , %CWMXESSB:AliasCompositions; )?>

<!ATTLIST CWMXESSB:Alias %CWMXESSB:AliasAttPropsList; %CWMXESSB:AliasAttRefsList;
    %XMI.element.att; %XMI.link.att; >

<!-- CLASS: CWMXESSB:Application -->

<!ENTITY % CWMXESSB:ApplicationProperties '(%UML:PackageProperties;)' >
<!ENTITY % CWMXESSB:ApplicationAssociations '(%UML:PackageAssociations;)' >
<!ENTITY % CWMXESSB:ApplicationCompositions '(%UML:PackageCompositions;)' >
<!ENTITY % CWMXESSB:ApplicationAttPropsList '%UML:PackageAttPropsList;' >
<!ENTITY % CWMXESSB:ApplicationAttRefsList '%UML:PackageAttRefsList;' >

<!ELEMENT CWMXESSB:Application ( %CWMXESSB:ApplicationProperties;
    ,(XMI.extension* , %CWMXESSB:ApplicationAssociations; )
    , %CWMXESSB:ApplicationCompositions; )?>

<!ATTLIST CWMXESSB:Application %CWMXESSB:ApplicationAttPropsList;
    %CWMXESSB:ApplicationAttRefsList; %XMI.element.att; %XMI.link.att;
    >
```

```

<!-- CLASS: CWMXESSB:Comment -->
<!ENTITY % CWMXESSB:CommentProperties '(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:CommentAssociations '(%CWMMDB:DimensionedObjectAssociations;)'
>
<!ENTITY % CWMXESSB:CommentCompositions
'(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:CommentAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:CommentAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >
<!ELEMENT CWMXESSB:Comment ( %CWMXESSB:CommentProperties;
,(XML.extension* , %CWMXESSB:CommentAssociations; )
, %CWMXESSB:CommentCompositions; )?>
<!ATTLIST CWMXESSB:Comment %CWMXESSB:CommentAttPropsList;
%CWMXESSB:CommentAttRefsList; %XML.element.att; %XML.link.att;
>

<!-- CLASS: CWMXESSB:Consolidation -->
<!ENTITY % CWMXESSB:ConsolidationProperties '(%CWMMDB:DimensionedObjectProperties;)'
>
<!ENTITY % CWMXESSB:ConsolidationAssociations
'(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:ConsolidationCompositions
'(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:ConsolidationAttPropsList
'%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:ConsolidationAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;'
>
<!ELEMENT CWMXESSB:Consolidation ( %CWMXESSB:ConsolidationProperties;
,(XML.extension* , %CWMXESSB:ConsolidationAssociations; )
, %CWMXESSB:ConsolidationCompositions; )?>
<!ATTLIST CWMXESSB:Consolidation %CWMXESSB:ConsolidationAttPropsList;
%CWMXESSB:ConsolidationAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXESSB:CurrencyConversion -->
<!ENTITY % CWMXESSB:CurrencyConversionProperties
'(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:CurrencyConversionAssociations
'(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:CurrencyConversionCompositions
'(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:CurrencyConversionAttPropsList
'%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:CurrencyConversionAttRefsList
'%CWMMDB:DimensionedObjectAttRefsList;' >
<!ELEMENT CWMXESSB:CurrencyConversion ( %CWMXESSB:CurrencyConversionProperties;
,(XML.extension* , %CWMXESSB:CurrencyConversionAssociations; )

```

```

, %CWMXESSB:CurrencyConversionCompositions; )?>

<!ATTLIST CWMXESSB:CurrencyConversion %CWMXESSB:CurrencyConversionAttPropsList;
          %CWMXESSB:CurrencyConversionAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXESSB:DataStorage -->

<!ENTITY % CWMXESSB:DataStorageProperties '(%CWMMDB:DimensionedObjectProperties;)' >

<!ENTITY % CWMXESSB:DataStorageAssociations
          '(%CWMMDB:DimensionedObjectAssociations;)' >

<!ENTITY % CWMXESSB:DataStorageCompositions
          '(%CWMMDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXESSB:DataStorageAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;'
          >

<!ENTITY % CWMXESSB:DataStorageAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >

<!ELEMENT CWMXESSB:DataStorage ( %CWMXESSB:DataStorageProperties;
          , (XML.extension* , %CWMXESSB:DataStorageAssociations; )
          , %CWMXESSB:DataStorageCompositions; )?>

<!ATTLIST CWMXESSB:DataStorage %CWMXESSB:DataStorageAttPropsList;
          %CWMXESSB:DataStorageAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXESSB:Database -->

<!ELEMENT CWMXESSB:Database.isCurrency EMPTY>
<!ATTLIST CWMXESSB:Database.isCurrency
          xmi.value ( true | false ) #REQUIRED>

<!ENTITY % CWMXESSB:DatabaseProperties '(%CWMMDB:SchemaProperties;
          ,(CWMXESSB:Database.isCurrency)?)' >

<!ENTITY % CWMXESSB:DatabaseAssociations '(%CWMMDB:SchemaAssociations;)' >

<!ENTITY % CWMXESSB:DatabaseCompositions '(%CWMMDB:SchemaCompositions;
          ,CWMXESSB:Database.outline?)' >

<!ENTITY % CWMXESSB:DatabaseAttPropsList '%CWMMDB:SchemaAttPropsList;
          isCurrency (true|false) #IMPLIED ' >

<!ENTITY % CWMXESSB:DatabaseAttRefsList '%CWMMDB:SchemaAttRefsList;' >

<!ELEMENT CWMXESSB:Database ( %CWMXESSB:DatabaseProperties;
          , (XML.extension* , %CWMXESSB:DatabaseAssociations; )
          , %CWMXESSB:DatabaseCompositions; )?>

<!ATTLIST CWMXESSB:Database %CWMXESSB:DatabaseAttPropsList;
          %CWMXESSB:DatabaseAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXESSB:Dimension -->

<!ELEMENT CWMXESSB:Dimension.type EMPTY>
<!ATTLIST CWMXESSB:Dimension.type %CWMXESSB:DimensionType;>

<!ELEMENT CWMXESSB:Dimension.isDense EMPTY>
<!ATTLIST CWMXESSB:Dimension.isDense
          xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXESSB:Dimension.outline (CWMXESSB:Outline)?>

```

```

<!ENTITY % CWMXESSB:DimensionProperties '(%CWMMDB:DimensionProperties;
,(CWMXESSB:Dimension.type)?
,(CWMXESSB:Dimension.isDense)?)' >

<!ENTITY % CWMXESSB:DimensionAssociations '(%CWMMDB:DimensionAssociations;
,CWMXESSB:Dimension.outline?)' >

<!ENTITY % CWMXESSB:DimensionCompositions '(%CWMMDB:DimensionCompositions;)' >

<!ENTITY % CWMXESSB:DimensionAttPropsList '(%CWMMDB:DimensionAttPropsList;
type ( none| accounts| time| country| currencyPartition| attribute ) #IMPLIED
isDense (true|false) #IMPLIED ' >

<!ENTITY % CWMXESSB:DimensionAttRefsList '(%CWMMDB:DimensionAttRefsList;
outline IDREFS #IMPLIED' >

<!ELEMENT CWMXESSB:Dimension ( %CWMXESSB:DimensionProperties;
,(XML.extension* , %CWMXESSB:DimensionAssociations; )
, %CWMXESSB:DimensionCompositions; )?>

<!ATTLIST CWMXESSB:Dimension %CWMXESSB:DimensionAttPropsList;
%CWMXESSB:DimensionAttRefsList; %XML.element.att; %XML.link.att;
>

<!-- CLASS: CWMXESSB:Formula -->

<!ENTITY % CWMXESSB:FormulaProperties '(%CWMMDB:DimensionedObjectProperties;)' >

<!ENTITY % CWMXESSB:FormulaAssociations '(%CWMMDB:DimensionedObjectAssociations;)' >

<!ENTITY % CWMXESSB:FormulaCompositions '(%CWMMDB:DimensionedObjectCompositions;)'
>

<!ENTITY % CWMXESSB:FormulaAttPropsList '(%CWMMDB:DimensionedObjectAttPropsList;)' >

<!ENTITY % CWMXESSB:FormulaAttRefsList '(%CWMMDB:DimensionedObjectAttRefsList;)' >

<!ELEMENT CWMXESSB:Formula ( %CWMXESSB:FormulaProperties;
,(XML.extension* , %CWMXESSB:FormulaAssociations; )
, %CWMXESSB:FormulaCompositions; )?>

<!ATTLIST CWMXESSB:Formula %CWMXESSB:FormulaAttPropsList;
%CWMXESSB:FormulaAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXESSB:Generation -->

<!ENTITY % CWMXESSB:GenerationProperties '(%CWMMDB:DimensionedObjectProperties;)' >

<!ENTITY % CWMXESSB:GenerationAssociations
'(%CWMMDB:DimensionedObjectAssociations;)' >

<!ENTITY % CWMXESSB:GenerationCompositions
'(%CWMMDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXESSB:GenerationAttPropsList '(%CWMMDB:DimensionedObjectAttPropsList;)' >

<!ENTITY % CWMXESSB:GenerationAttRefsList '(%CWMMDB:DimensionedObjectAttRefsList;)' >

<!ELEMENT CWMXESSB:Generation ( %CWMXESSB:GenerationProperties;
,(XML.extension* , %CWMXESSB:GenerationAssociations; )
, %CWMXESSB:GenerationCompositions; )?>

<!ATTLIST CWMXESSB:Generation %CWMXESSB:GenerationAttPropsList;
%CWMXESSB:GenerationAttRefsList; %XML.element.att; %XML.link.att;
>

```

```

<!-- CLASS: CWMXESSB:ImmediateParent -->

<!ENTITY % CWMXESSB:ImmediateParentProperties
          '(%CWMMDB:DimensionedObjectProperties;)' >

<!ENTITY % CWMXESSB:ImmediateParentAssociations
          '(%CWMMDB:DimensionedObjectAssociations;)' >

<!ENTITY % CWMXESSB:ImmediateParentCompositions
          '(%CWMMDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXESSB:ImmediateParentAttPropsList
          '%CWMMDB:DimensionedObjectAttPropsList;' >

<!ENTITY % CWMXESSB:ImmediateParentAttRefsList
          '%CWMMDB:DimensionedObjectAttRefsList;' >

<!ELEMENT CWMXESSB:ImmediateParent ( %CWMXESSB:ImmediateParentProperties;
          ,(XMI.extension* , %CWMXESSB:ImmediateParentAssociations; )
          , %CWMXESSB:ImmediateParentCompositions; )?>

<!ATTLIST CWMXESSB:ImmediateParent %CWMXESSB:ImmediateParentAttPropsList;
          %CWMXESSB:ImmediateParentAttRefsList; %XMI.element.att;
          %XMI.link.att; >

<!-- CLASS: CWMXESSB:Level -->

<!ENTITY % CWMXESSB:LevelProperties '(%CWMMDB:DimensionedObjectProperties;)' >

<!ENTITY % CWMXESSB:LevelAssociations '(%CWMMDB:DimensionedObjectAssociations;)' >

<!ENTITY % CWMXESSB:LevelCompositions '(%CWMMDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXESSB:LevelAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;' >

<!ENTITY % CWMXESSB:LevelAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >

<!ELEMENT CWMXESSB:Level ( %CWMXESSB:LevelProperties;
          ,(XMI.extension* , %CWMXESSB:LevelAssociations; )
          , %CWMXESSB:LevelCompositions; )?>

<!ATTLIST CWMXESSB:Level %CWMXESSB:LevelAttPropsList; %CWMXESSB:LevelAttRefsList;
          %XMI.element.att; %XMI.link.att; >

<!-- CLASS: CWMXESSB:Partition -->

<!ELEMENT CWMXESSB:Partition.isSource EMPTY>
<!ATTLIST CWMXESSB:Partition.isSource
          xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXESSB:Partition.isShared EMPTY>
<!ATTLIST CWMXESSB:Partition.isShared
          xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXESSB:Partition.formula (CWM:ExpressionNode
|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>

<!ENTITY % CWMXESSB:PartitionProperties '(%CWMOLAP:CubeRegionProperties;
          ,(CWMXESSB:Partition.isSource)?
          ,(CWMXESSB:Partition.isShared)?
          ,(CWMXESSB:Partition.formula)?)' >

```

```

<!ENTITY % CWMXESSB:PartitionAssociations '(%CWMOLAP:CubeRegionAssociations;)' >
<!ENTITY % CWMXESSB:PartitionCompositions '(%CWMOLAP:CubeRegionCompositions;)' >
<!ENTITY % CWMXESSB:PartitionAttPropsList '%CWMOLAP:CubeRegionAttPropsList;
    isSource (true|false) #IMPLIED
    isShared (true|false) #IMPLIED
    formula CDATA #IMPLIED' >
<!ENTITY % CWMXESSB:PartitionAttRefsList '%CWMOLAP:CubeRegionAttRefsList;' >
<!ELEMENT CWMXESSB:Partition ( %CWMXESSB:PartitionProperties;
    ,(XML.extension* , %CWMXESSB:PartitionAssociations; )
    , %CWMXESSB:PartitionCompositions; )?>
<!ATTLIST CWMXESSB:Partition %CWMXESSB:PartitionAttPropsList;
    %CWMXESSB:PartitionAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXESSB:LinkedPartition -->
<!ENTITY % CWMXESSB:LinkedPartitionProperties '(%CWMXESSB:PartitionProperties;)' >
<!ENTITY % CWMXESSB:LinkedPartitionAssociations '(%CWMXESSB:PartitionAssociations;)' >
<!ENTITY % CWMXESSB:LinkedPartitionCompositions '(%CWMXESSB:PartitionCompositions;)' >
<!ENTITY % CWMXESSB:LinkedPartitionAttPropsList '%CWMXESSB:PartitionAttPropsList;' >
<!ENTITY % CWMXESSB:LinkedPartitionAttRefsList '%CWMXESSB:PartitionAttRefsList;' >
<!ELEMENT CWMXESSB:LinkedPartition ( %CWMXESSB:LinkedPartitionProperties;
    ,(XML.extension* , %CWMXESSB:LinkedPartitionAssociations; )
    , %CWMXESSB:LinkedPartitionCompositions; )?>
<!ATTLIST CWMXESSB:LinkedPartition %CWMXESSB:LinkedPartitionAttPropsList;
    %CWMXESSB:LinkedPartitionAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXESSB:MemberName -->
<!ENTITY % CWMXESSB:MemberNameProperties '(%CWMMDDB:DimensionedObjectProperties;)'
    >
<!ENTITY % CWMXESSB:MemberNameAssociations
    '%CWMMDDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:MemberNameCompositions
    '%CWMMDDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:MemberNameAttPropsList
    '%CWMMDDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:MemberNameAttRefsList '%CWMMDDB:DimensionedObjectAttRefsList;'
    >
<!ELEMENT CWMXESSB:MemberName ( %CWMXESSB:MemberNameProperties;
    ,(XML.extension* , %CWMXESSB:MemberNameAssociations; )
    , %CWMXESSB:MemberNameCompositions; )?>
<!ATTLIST CWMXESSB:MemberName %CWMXESSB:MemberNameAttPropsList;
    %CWMXESSB:MemberNameAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXESSB:OLAPServer -->

```



```

<!ENTITY % CWMXESSB:OLAPServerProperties '(%CWM:DataManagerProperties;)' >
<!ENTITY % CWMXESSB:OLAPServerAssociations '(%CWM:DataManagerAssociations;)' >
<!ENTITY % CWMXESSB:OLAPServerCompositions '(%CWM:DataManagerCompositions;)' >
<!ENTITY % CWMXESSB:OLAPServerAttPropsList '%CWM:DataManagerAttPropsList;' >
<!ENTITY % CWMXESSB:OLAPServerAttRefsList '%CWM:DataManagerAttRefsList;' >
<!ELEMENT CWMXESSB:OLAPServer ( %CWMXESSB:OLAPServerProperties;
    , (XML.extension* , %CWMXESSB:OLAPServerAssociations; )
    , %CWMXESSB:OLAPServerCompositions; )? >
<!ATTLIST CWMXESSB:OLAPServer %CWMXESSB:OLAPServerAttPropsList;
    %CWMXESSB:OLAPServerAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXESSB:Outline -->
<!ELEMENT CWMXESSB:Outline.aliasTableName (#PCDATA|XML.reference)*>
<!ELEMENT CWMXESSB:Outline.database (CWMXESSB:Database)?>
<!ELEMENT CWMXESSB:Outline.dimension (CWMXESSB:Dimension)*>
<!ENTITY % CWMXESSB:OutlineProperties '(%UML:NamespaceProperties;
    ,(CWMXESSB:Outline.aliasTableName)?)' >
<!ENTITY % CWMXESSB:OutlineAssociations '(%UML:NamespaceAssociations;
    ,CWMXESSB:Outline.database?
    ,CWMXESSB:Outline.dimension*)' >
<!ENTITY % CWMXESSB:OutlineCompositions '(%UML:NamespaceCompositions;)' >
<!ENTITY % CWMXESSB:OutlineAttPropsList '%UML:NamespaceAttPropsList;
    aliasTableName CDATA #IMPLIED' >
<!ENTITY % CWMXESSB:OutlineAttRefsList '%UML:NamespaceAttRefsList;
    database IDREFS #IMPLIED' >
<!ELEMENT CWMXESSB:Outline ( %CWMXESSB:OutlineProperties;
    , (XML.extension* , %CWMXESSB:OutlineAssociations; )
    , %CWMXESSB:OutlineCompositions; )? >
<!ATTLIST CWMXESSB:Outline %CWMXESSB:OutlineAttPropsList;
    %CWMXESSB:OutlineAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXESSB:ReplicatedPartition -->
<!ENTITY % CWMXESSB:ReplicatedPartitionProperties '(%CWMXESSB:PartitionProperties;)' >
<!ENTITY % CWMXESSB:ReplicatedPartitionAssociations '(%CWMXESSB:PartitionAssociations;)' >
<!ENTITY % CWMXESSB:ReplicatedPartitionCompositions
    '(%CWMXESSB:PartitionCompositions;)' >
<!ENTITY % CWMXESSB:ReplicatedPartitionAttPropsList '%CWMXESSB:PartitionAttPropsList;' >
<!ENTITY % CWMXESSB:ReplicatedPartitionAttRefsList '%CWMXESSB:PartitionAttRefsList;' >
<!ELEMENT CWMXESSB:ReplicatedPartition ( %CWMXESSB:ReplicatedPartitionProperties;
    , (XML.extension* , %CWMXESSB:ReplicatedPartitionAssociations; )
    , %CWMXESSB:ReplicatedPartitionCompositions; )? >

```

```

<!ATTLIST CWMXESSB:ReplicatedPartition %CWMXESSB:ReplicatedPartitionAttPropsList;
          %CWMXESSB:ReplicatedPartitionAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXESSB:TimeBalance -->
<!ENTITY % CWMXESSB:TimeBalanceProperties '(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:TimeBalanceAssociations
          '(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:TimeBalanceCompositions
          '(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:TimeBalanceAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;'
          >
<!ENTITY % CWMXESSB:TimeBalanceAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >
<!ELEMENT CWMXESSB:TimeBalance ( %CWMXESSB:TimeBalanceProperties;
          , (XML.extension* , %CWMXESSB:TimeBalanceAssociations; )
          , %CWMXESSB:TimeBalanceCompositions; )?>
<!ATTLIST CWMXESSB:TimeBalance %CWMXESSB:TimeBalanceAttPropsList;
          %CWMXESSB:TimeBalanceAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXESSB:TransparentPartition -->
<!ENTITY % CWMXESSB:TransparentPartitionProperties '(%CWMXESSB:PartitionProperties;)' >
<!ENTITY % CWMXESSB:TransparentPartitionAssociations
          '(%CWMXESSB:PartitionAssociations;)' >
<!ENTITY % CWMXESSB:TransparentPartitionCompositions
          '(%CWMXESSB:PartitionCompositions;)' >
<!ENTITY % CWMXESSB:TransparentPartitionAttPropsList '%CWMXESSB:PartitionAttPropsList;'
          >
<!ENTITY % CWMXESSB:TransparentPartitionAttRefsList '%CWMXESSB:PartitionAttRefsList;' >
<!ELEMENT CWMXESSB:TransparentPartition ( %CWMXESSB:TransparentPartitionProperties;
          , (XML.extension* , %CWMXESSB:TransparentPartitionAssociations; )
          , %CWMXESSB:TransparentPartitionCompositions; )?>
<!ATTLIST CWMXESSB:TransparentPartition %CWMXESSB:TransparentPartitionAttPropsList;
          %CWMXESSB:TransparentPartitionAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXESSB:TwoPassCalculation -->
<!ENTITY % CWMXESSB:TwoPassCalculationProperties
          '(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:TwoPassCalculationAssociations
          '(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:TwoPassCalculationCompositions
          '(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:TwoPassCalculationAttPropsList
          '%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:TwoPassCalculationAttRefsList

```

```

                                '%CWMMDB:DimensionedObjectAttRefsList;' >
<!ELEMENT CWMXESSB:TwoPassCalculation ( %CWMXESSB:TwoPassCalculationProperties;
                                ,(XML.extension* , %CWMXESSB:TwoPassCalculationAssociations; )
                                , %CWMXESSB:TwoPassCalculationCompositions; )?>
<!-- ATTLIST CWMXESSB:TwoPassCalculation %CWMXESSB:TwoPassCalculationAttPropsList;
                                %CWMXESSB:TwoPassCalculationAttRefsList; %XML.element.att;
                                %XML.link.att; >

<!-- CLASS: CWMXESSB:UDA -->
<!ENTITY % CWMXESSB:UDAProperties '(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:UDAAssociations '(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:UDACompositions '(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:UDAAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:UDAAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >
<!ELEMENT CWMXESSB:UDA ( %CWMXESSB:UDAProperties;
                                ,(XML.extension* , %CWMXESSB:UDAAssociations; )
                                , %CWMXESSB:UDACompositions; )?>
<!-- ATTLIST CWMXESSB:UDA %CWMXESSB:UDAAttPropsList; %CWMXESSB:UDAAttRefsList;
                                %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXESSB:VarianceReporting -->
<!ENTITY % CWMXESSB:VarianceReportingProperties
                                '(%CWMMDB:DimensionedObjectProperties;)' >
<!ENTITY % CWMXESSB:VarianceReportingAssociations
                                '(%CWMMDB:DimensionedObjectAssociations;)' >
<!ENTITY % CWMXESSB:VarianceReportingCompositions
                                '(%CWMMDB:DimensionedObjectCompositions;)' >
<!ENTITY % CWMXESSB:VarianceReportingAttPropsList
                                '%CWMMDB:DimensionedObjectAttPropsList;' >
<!ENTITY % CWMXESSB:VarianceReportingAttRefsList
                                '%CWMMDB:DimensionedObjectAttRefsList;' >
<!ELEMENT CWMXESSB:VarianceReporting ( %CWMXESSB:VarianceReportingProperties;
                                ,(XML.extension* , %CWMXESSB:VarianceReportingAssociations; )
                                , %CWMXESSB:VarianceReportingCompositions; )?>
<!-- ATTLIST CWMXESSB:VarianceReporting %CWMXESSB:VarianceReportingAttPropsList;
                                %CWMXESSB:VarianceReportingAttRefsList; %XML.element.att;
                                %XML.link.att; >

```

3.7 *Express.dtd*

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- DataTypes.dtd -->
<!-- Multidimensional.dtd -->

<!ATTLIST XMI xmlns:CWMXExpress CDATA #IMPLIED >

<!-- PACKAGE: CWMXExpress:Express -->

<!-- ***** CWMXExpress:ComputeClause ***** -->
<!ELEMENT CWMXExpress:AggMapComponent.computeClause (
    CWMXExpress:PreComputeClause)? >

<!-- ***** CWMXExpress:AggMapComponents ***** -->
<!ELEMENT CWMXExpress:AggMap.aggMapComponent ( CWMXExpress:AggMapComponent)* >

<!-- CLASS: CWMXExpress:Dimension -->
<!ENTITY % CWMXExpress:DimensionProperties '(%CWMMDb:DimensionProperties;)' >
<!ENTITY % CWMXExpress:DimensionAssociations '(%CWMMDb:DimensionAssociations;)' >
<!ENTITY % CWMXExpress:DimensionCompositions '(%CWMMDb:DimensionCompositions;)' >
<!ENTITY % CWMXExpress:DimensionAttPropsList '%CWMMDb:DimensionAttPropsList;' >
<!ENTITY % CWMXExpress:DimensionAttRefsList '%CWMMDb:DimensionAttRefsList;' >
<!ELEMENT CWMXExpress:Dimension ( %CWMXExpress:DimensionProperties;
    ,(XML.extension* , %CWMXExpress:DimensionAssociations; )
    , %CWMXExpress:DimensionCompositions; )?>
<!ATTLIST CWMXExpress:Dimension %CWMXExpress:DimensionAttPropsList;
    %CWMXExpress:DimensionAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXExpress:Database -->
<!ENTITY % CWMXExpress:DatabaseProperties '(%CWMMDb:SchemaProperties;)' >
<!ENTITY % CWMXExpress:DatabaseAssociations '(%CWMMDb:SchemaAssociations;)' >
<!ENTITY % CWMXExpress:DatabaseCompositions '(%CWMMDb:SchemaCompositions;)' >
<!ENTITY % CWMXExpress:DatabaseAttPropsList '%CWMMDb:SchemaAttPropsList;' >
<!ENTITY % CWMXExpress:DatabaseAttRefsList '%CWMMDb:SchemaAttRefsList;' >
<!ELEMENT CWMXExpress:Database ( %CWMXExpress:DatabaseProperties;
    ,(XML.extension* , %CWMXExpress:DatabaseAssociations; )
    , %CWMXExpress:DatabaseCompositions; )?>
<!ATTLIST CWMXExpress:Database %CWMXExpress:DatabaseAttPropsList;
    %CWMXExpress:DatabaseAttRefsList; %XML.element.att;
    %XML.link.att; >
```

```

<!-- CLASS: CWMXExpress:Conjoint -->
<!ELEMENT CWMXExpress:Conjoint.searchAlgorithm (#PCDATA|XML.reference)*>
<!ELEMENT CWMXExpress:Conjoint.pageSpace (#PCDATA|XML.reference)*>
<!ENTITY % CWMXExpress:ConjointProperties '(%CWMXExpress:DimensionProperties;
,(CWMXExpress:Conjoint.searchAlgorithm)?
,(CWMXExpress:Conjoint.pageSpace)?)' >
<!ENTITY % CWMXExpress:ConjointAssociations '(%CWMXExpress:DimensionAssociations;)' >
<!ENTITY % CWMXExpress:ConjointCompositions '(%CWMXExpress:DimensionCompositions;)' >
<!ENTITY % CWMXExpress:ConjointAttPropsList '%CWMXExpress:DimensionAttPropsList;
searchAlgorithm CDATA #IMPLIED
pageSpace CDATA #IMPLIED' >
<!ENTITY % CWMXExpress:ConjointAttRefsList '%CWMXExpress:DimensionAttRefsList;' >
<!ELEMENT CWMXExpress:Conjoint ( %CWMXExpress:ConjointProperties;
,(XML.extension* , %CWMXExpress:ConjointAssociations; )
, %CWMXExpress:ConjointCompositions; )?>
<!ATTLIST CWMXExpress:Conjoint %CWMXExpress:ConjointAttPropsList;
%CWMXExpress:ConjointAttRefsList; %XML.element.att; %XML.link.att;
>

<!-- CLASS: CWMXExpress:Program -->
<!ELEMENT CWMXExpress:Program.program (#PCDATA|XML.reference)*>
<!ELEMENT CWMXExpress:Program.returnDimension (#PCDATA|XML.reference)*>
<!ENTITY % CWMXExpress:ProgramProperties '(%UML:ComponentProperties;
,(CWMXExpress:Program.program)?
,(CWMXExpress:Program.returnDimension)?)' >
<!ENTITY % CWMXExpress:ProgramAssociations '(%UML:ComponentAssociations;)' >
<!ENTITY % CWMXExpress:ProgramCompositions '(%UML:ComponentCompositions;)' >
<!ENTITY % CWMXExpress:ProgramAttPropsList '%UML:ComponentAttPropsList;
program CDATA #IMPLIED
returnDimension CDATA #IMPLIED' >
<!ENTITY % CWMXExpress:ProgramAttRefsList '%UML:ComponentAttRefsList;' >
<!ELEMENT CWMXExpress:Program ( %CWMXExpress:ProgramProperties;
,(XML.extension* , %CWMXExpress:ProgramAssociations; )
, %CWMXExpress:ProgramCompositions; )?>
<!ATTLIST CWMXExpress:Program %CWMXExpress:ProgramAttPropsList;
%CWMXExpress:ProgramAttRefsList; %XML.element.att; %XML.link.att;
>

<!-- CLASS: CWMXExpress:Model -->
<!ELEMENT CWMXExpress:Model.content (#PCDATA|XML.reference)*>
<!ENTITY % CWMXExpress:ModelProperties '(%UML:ComponentProperties;
,(CWMXExpress:Model.content)?)' >
<!ENTITY % CWMXExpress:ModelAssociations '(%UML:ComponentAssociations;)' >

```

```

<!ENTITY % CWMXExpress:ModelCompositions '(%UML:ComponentCompositions;)' >

<!ENTITY % CWMXExpress:ModelAttPropsList '%UML:ComponentAttPropsList;
content CDATA #IMPLIED' >

<!ENTITY % CWMXExpress:ModelAttRefsList '%UML:ComponentAttRefsList;' >

<!ELEMENT CWMXExpress:Model ( %CWMXExpress:ModelProperties;
,(XML.extension* , %CWMXExpress:ModelAssociations; )
, %CWMXExpress:ModelCompositions; )?>

<!ATTLIST CWMXExpress:Model %CWMXExpress:ModelAttPropsList;
%CWMXExpress:ModelAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXExpress:Variable -->

<!ELEMENT CWMXExpress:Variable.storageType (#PCDATA|XML.reference)*>
<!ELEMENT CWMXExpress:Variable.pageSpace (#PCDATA|XML.reference)*>
<!ELEMENT CWMXExpress:Variable.width (#PCDATA|XML.reference)*>

<!ENTITY % CWMXExpress:VariableProperties '(%CWMMDB:DimensionedObjectProperties;
,(CWMXExpress:Variable.storageType)?
,(CWMXExpress:Variable.pageSpace)?
,(CWMXExpress:Variable.width)?)' >

<!ENTITY % CWMXExpress:VariableAssociations '(%CWMMDB:DimensionedObjectAssociations;)'
>

<!ENTITY % CWMXExpress:VariableCompositions
'(%CWMMDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXExpress:VariableAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;
storageType CDATA #IMPLIED
pageSpace CDATA #IMPLIED
width CDATA #IMPLIED' >

<!ENTITY % CWMXExpress:VariableAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >

<!ELEMENT CWMXExpress:Variable ( %CWMXExpress:VariableProperties;
,(XML.extension* , %CWMXExpress:VariableAssociations; )
, %CWMXExpress:VariableCompositions; )?>

<!ATTLIST CWMXExpress:Variable %CWMXExpress:VariableAttPropsList;
%CWMXExpress:VariableAttRefsList; %XML.element.att; %XML.link.att;
>

<!-- CLASS: CWMXExpress:Formula -->

<!ELEMENT CWMXExpress:Formula.expression (#PCDATA|XML.reference)*>

<!ENTITY % CWMXExpress:FormulaProperties '(%CWMMDB:DimensionedObjectProperties;
,(CWMXExpress:Formula.expression)?)' >

<!ENTITY % CWMXExpress:FormulaAssociations '(%CWMMDB:DimensionedObjectAssociations;)'
>

<!ENTITY % CWMXExpress:FormulaCompositions
'(%CWMMDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXExpress:FormulaAttPropsList '%CWMMDB:DimensionedObjectAttPropsList;
expression CDATA #IMPLIED' >

<!ENTITY % CWMXExpress:FormulaAttRefsList '%CWMMDB:DimensionedObjectAttRefsList;' >

```

```

<!ELEMENT CWMXExpress:Formula ( %CWMXExpress:FormulaProperties;
    ,(XML.extension* , %CWMXExpress:FormulaAssociations; )
    , %CWMXExpress:FormulaCompositions; )?>

<!ATTLIST CWMXExpress:Formula %CWMXExpress:FormulaAttPropsList;
    %CWMXExpress:FormulaAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXExpress:ValueSet -->

<!ELEMENT CWMXExpress:ValueSet.isTemp EMPTY>
<!ATTLIST CWMXExpress:ValueSet.isTemp
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXExpress:ValueSet.referenceDimension (CWMXExpress:Dimension)?>

<!ENTITY % CWMXExpress:ValueSetProperties '(%CWMMDDB:DimensionedObjectProperties;
    ,(CWMXExpress:ValueSet.isTemp)?)' >

<!ENTITY % CWMXExpress:ValueSetAssociations '(%CWMMDDB:DimensionedObjectAssociations;
    ,CWMXExpress:ValueSet.referenceDimension?)' >

<!ENTITY % CWMXExpress:ValueSetCompositions
    '(%CWMMDDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXExpress:ValueSetAttPropsList '%CWMMDDB:DimensionedObjectAttPropsList;
    isTemp (true|false) #IMPLIED' >

<!ENTITY % CWMXExpress:ValueSetAttRefsList '%CWMMDDB:DimensionedObjectAttRefsList;
    referenceDimension IDREFS #IMPLIED' >

<!ELEMENT CWMXExpress:ValueSet ( %CWMXExpress:ValueSetProperties;
    ,(XML.extension* , %CWMXExpress:ValueSetAssociations; )
    , %CWMXExpress:ValueSetCompositions; )?>

<!ATTLIST CWMXExpress:ValueSet %CWMXExpress:ValueSetAttPropsList;
    %CWMXExpress:ValueSetAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXExpress:Relation -->

<!ELEMENT CWMXExpress:Relation.isTemp EMPTY>
<!ATTLIST CWMXExpress:Relation.isTemp
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXExpress:Relation.pageSpace (#PCDATA|XML.reference)*>

<!ELEMENT CWMXExpress:Relation.referenceDimension (CWMXExpress:Dimension)?>

<!ENTITY % CWMXExpress:RelationProperties '(%CWMMDDB:DimensionedObjectProperties;
    ,(CWMXExpress:Relation.isTemp)?
    ,(CWMXExpress:Relation.pageSpace)?)' >

<!ENTITY % CWMXExpress:RelationAssociations '(%CWMMDDB:DimensionedObjectAssociations;
    ,CWMXExpress:Relation.referenceDimension?)' >

<!ENTITY % CWMXExpress:RelationCompositions
    '(%CWMMDDB:DimensionedObjectCompositions;)' >

<!ENTITY % CWMXExpress:RelationAttPropsList '%CWMMDDB:DimensionedObjectAttPropsList;
    isTemp (true|false) #IMPLIED
    pageSpace CDATA #IMPLIED' >

<!ENTITY % CWMXExpress:RelationAttRefsList '%CWMMDDB:DimensionedObjectAttRefsList;
    referenceDimension IDREFS #IMPLIED' >

```

```

<!ELEMENT CWMXExpress:Relation ( %CWMXExpress:RelationProperties;
    ,(XML.extension* , %CWMXExpress:RelationAssociations; )
    , %CWMXExpress:RelationCompositions; )?>

<!ATTLIST CWMXExpress:Relation %CWMXExpress:RelationAttPropsList;
    %CWMXExpress:RelationAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXExpress:Worksheet -->

<!ELEMENT CWMXExpress:Worksheet.isTemp EMPTY>
<!ATTLIST CWMXExpress:Worksheet.isTemp
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXExpress:Worksheet.columnDimension (CWMXExpress:Dimension)?>
<!ELEMENT CWMXExpress:Worksheet.rowDimension (CWMXExpress:Dimension)?>

<!ENTITY % CWMXExpress:WorksheetProperties '(%UML:ClassProperties;
    ,(CWMXExpress:Worksheet.isTemp)?)' >

<!ENTITY % CWMXExpress:WorksheetAssociations '(%UML:ClassAssociations;
    ,CWMXExpress:Worksheet.columnDimension?
    ,CWMXExpress:Worksheet.rowDimension?)' >

<!ENTITY % CWMXExpress:WorksheetCompositions '(%UML:ClassCompositions;)' >

<!ENTITY % CWMXExpress:WorksheetAttPropsList '%UML:ClassAttPropsList;
    isTemp (true|false) #IMPLIED' >

<!ENTITY % CWMXExpress:WorksheetAttRefsList '%UML:ClassAttRefsList;
    columnDimension IDREFS #IMPLIED
    rowDimension IDREFS #IMPLIED' >

<!ELEMENT CWMXExpress:Worksheet ( %CWMXExpress:WorksheetProperties;
    ,(XML.extension* , %CWMXExpress:WorksheetAssociations; )
    , %CWMXExpress:WorksheetCompositions; )?>

<!ATTLIST CWMXExpress:Worksheet %CWMXExpress:WorksheetAttPropsList;
    %CWMXExpress:WorksheetAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXExpress:Composite -->

<!ELEMENT CWMXExpress:Composite.searchAlgorithm (#PCDATA|XML.reference)*>
<!ELEMENT CWMXExpress:Composite.pageSpace (#PCDATA|XML.reference)*>

<!ENTITY % CWMXExpress:CompositeProperties '(%CWMXExpress:DimensionProperties;
    ,(CWMXExpress:Composite.searchAlgorithm)?
    ,(CWMXExpress:Composite.pageSpace)?)' >

<!ENTITY % CWMXExpress:CompositeAssociations '(%CWMXExpress:DimensionAssociations;)' >

<!ENTITY % CWMXExpress:CompositeCompositions
    '(%CWMXExpress:DimensionCompositions;)' >

<!ENTITY % CWMXExpress:CompositeAttPropsList '%CWMXExpress:DimensionAttPropsList;
    searchAlgorithm CDATA #IMPLIED
    pageSpace CDATA #IMPLIED' >

<!ENTITY % CWMXExpress:CompositeAttRefsList '%CWMXExpress:DimensionAttRefsList;' >

<!ELEMENT CWMXExpress:Composite ( %CWMXExpress:CompositeProperties;
    ,(XML.extension* , %CWMXExpress:CompositeAssociations; )
    , %CWMXExpress:CompositeCompositions; )?>

```



```

<!-- CLASS: CWMXExpress:SimpleDimension -->

<ELEMENT CWMXExpress:SimpleDimension.width (#PCDATA|XML.reference)*>

<!-- ELEMENT CWMXExpress:SimpleDimension.isTime EMPTY -->
<!-- ATTLIST CWMXExpress:SimpleDimension.isTime -->
  xmi.value ( true | false ) #REQUIRED>

<ELEMENT CWMXExpress:SimpleDimension.multiple (#PCDATA|XML.reference)*>

<ELEMENT CWMXExpress:SimpleDimension.beginningPhase (#PCDATA|XML.reference)*>

<ELEMENT CWMXExpress:SimpleDimension.endingPhase (#PCDATA|XML.reference)*>

<ELEMENT CWMXExpress:SimpleDimension.searchAlgorithm (#PCDATA|XML.reference)*>

<ELEMENT CWMXExpress:SimpleDimension.pageSpace (#PCDATA|XML.reference)*>

<!-- ELEMENT CWMXExpress:SimpleDimension.dataType (UML:Classifier)? -->

<!-- ELEMENT CWMXExpress:SimpleDimension.aliasDimension (CWMXExpress:AliasDimension)* -->

<!-- ENTITY % CWMXExpress:SimpleDimensionProperties -->
  (%CWMXExpress:DimensionProperties;
  ,CWMXExpress:SimpleDimension.width)?
  ,CWMXExpress:SimpleDimension.isTime)?
  ,CWMXExpress:SimpleDimension.multiple)?
  ,CWMXExpress:SimpleDimension.beginningPhase)?
  ,CWMXExpress:SimpleDimension.endingPhase)?
  ,CWMXExpress:SimpleDimension.searchAlgorithm)?
  ,CWMXExpress:SimpleDimension.pageSpace)?' >

<!-- ENTITY % CWMXExpress:SimpleDimensionAssociations -->
  (%CWMXExpress:DimensionAssociations;
  ,CWMXExpress:SimpleDimension.dataType?
  ,CWMXExpress:SimpleDimension.aliasDimension*)' >

<!-- ENTITY % CWMXExpress:SimpleDimensionCompositions -->
  (%CWMXExpress:DimensionCompositions;)' >

<!-- ENTITY % CWMXExpress:SimpleDimensionAttPropsList -->
  (%CWMXExpress:DimensionAttPropsList;
  width CDATA #IMPLIED
  isTime (true|false) #IMPLIED
  multiple CDATA #IMPLIED
  beginningPhase CDATA #IMPLIED
  endingPhase CDATA #IMPLIED
  searchAlgorithm CDATA #IMPLIED
  pageSpace CDATA #IMPLIED)' >

<!-- ENTITY % CWMXExpress:SimpleDimensionAttRefsList -->
  (%CWMXExpress:DimensionAttRefsList;
  dataType IDREFS #IMPLIED)' >

<!-- ELEMENT CWMXExpress:SimpleDimension ( %CWMXExpress:SimpleDimensionProperties;
  , (XML.extension* , %CWMXExpress:SimpleDimensionAssociations; )
  , %CWMXExpress:SimpleDimensionCompositions; )? -->

<!-- ATTLIST CWMXExpress:SimpleDimension -->
  (%CWMXExpress:SimpleDimensionAttPropsList;
  %CWMXExpress:SimpleDimensionAttRefsList; %XML.element.att;
  %XML.link.att; >

<!-- CLASS: CWMXExpress:AliasDimension -->

```

```

<!ELEMENT CWMXExpress:AliasDimension.baseDimension (CWMXExpress:SimpleDimension)?>
<!ENTITY % CWMXExpress:AliasDimensionProperties '(%CWMXExpress:DimensionProperties;)' >
<!ENTITY % CWMXExpress:AliasDimensionAssociations
    '(%CWMXExpress:DimensionAssociations;
    ,CWMXExpress:AliasDimension.baseDimension?)' >
<!ENTITY % CWMXExpress:AliasDimensionCompositions
    '(%CWMXExpress:DimensionCompositions;)' >
<!ENTITY % CWMXExpress:AliasDimensionAttPropsList
    '%CWMXExpress:DimensionAttPropsList;' >
<!ENTITY % CWMXExpress:AliasDimensionAttRefsList '%CWMXExpress:DimensionAttRefsList;
    baseDimension IDREFS #IMPLIED' >
<!ELEMENT CWMXExpress:AliasDimension ( %CWMXExpress:AliasDimensionProperties;
    ,(XML.extension* , %CWMXExpress:AliasDimensionAssociations; )
    , %CWMXExpress:AliasDimensionCompositions; )?>
<!ATTLIST CWMXExpress:AliasDimension %CWMXExpress:AliasDimensionAttPropsList;
    %CWMXExpress:AliasDimensionAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXExpress:AggMap -->
<!ENTITY % CWMXExpress:AggMapProperties '(%UML:ClassProperties;)' >
<!ENTITY % CWMXExpress:AggMapAssociations '(%UML:ClassAssociations;)' >
<!ENTITY % CWMXExpress:AggMapCompositions '(%UML:ClassCompositions;
    ,CWMXExpress:AggMap.aggMapComponent*)' >
<!ENTITY % CWMXExpress:AggMapAttPropsList '%UML:ClassAttPropsList;' >
<!ENTITY % CWMXExpress:AggMapAttRefsList '%UML:ClassAttRefsList;' >
<!ELEMENT CWMXExpress:AggMap ( %CWMXExpress:AggMapProperties;
    ,(XML.extension* , %CWMXExpress:AggMapAssociations; )
    , %CWMXExpress:AggMapCompositions; )?>
<!ATTLIST CWMXExpress:AggMap %CWMXExpress:AggMapAttPropsList;
    %CWMXExpress:AggMapAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXExpress:AggMapComponent -->
<!ELEMENT CWMXExpress:AggMapComponent.aggOperator (#PCDATA|XML.reference)*>
<!ELEMENT CWMXExpress:AggMapComponent.aggMap (CWMXExpress:AggMap)?>
<!ELEMENT CWMXExpress:AggMapComponent.dimension (CWMXExpress:Dimension)?>
<!ELEMENT CWMXExpress:AggMapComponent.relation (CWMXExpress:Relation)?>
<!ENTITY % CWMXExpress:AggMapComponentProperties '(%UML:ModelElementProperties;
    ,(CWMXExpress:AggMapComponent.aggOperator?))' >
<!ENTITY % CWMXExpress:AggMapComponentAssociations '(%UML:ModelElementAssociations;
    ,CWMXExpress:AggMapComponent.aggMap?
    ,CWMXExpress:AggMapComponent.dimension?
    ,CWMXExpress:AggMapComponent.relation?)' >
<!ENTITY % CWMXExpress:AggMapComponentCompositions

```

```

                                '(%UML:ModelElementCompositions;
                                ,CWMXExpress:AggMapComponent.computeClause?)' >
<!ENTITY % CWMXExpress:AggMapComponentAttPropsList '%UML:ModelElementAttPropsList;
                                aggOperator CDATA #IMPLIED' >

<!ENTITY % CWMXExpress:AggMapComponentAttRefsList '%UML:ModelElementAttRefsList;
                                aggMap IDREFS #IMPLIED
                                dimension IDREFS #IMPLIED
                                relation IDREFS #IMPLIED' >

<!ELEMENT CWMXExpress:AggMapComponent (
                                %CWMXExpress:AggMapComponentProperties;
                                ,(XML.extension* , %CWMXExpress:AggMapComponentAssociations; )
                                , %CWMXExpress:AggMapComponentCompositions; )?>

<!ATTLIST CWMXExpress:AggMapComponent %CWMXExpress:AggMapComponentAttPropsList;
                                %CWMXExpress:AggMapComponentAttRefsList; %XML.element.att;
                                %XML.link.att; >

<!-- CLASS: CWMXExpress:PreComputeClause -->

<!ELEMENT CWMXExpress:PreComputeClause.statusList (XML.any)>

<!ELEMENT CWMXExpress:PreComputeClause.aggMapComponent
                                (CWMXExpress:AggMapComponent)?>

<!ENTITY % CWMXExpress:PreComputeClauseProperties '(%UML:ModelElementProperties;
                                ,CWMXExpress:PreComputeClause.statusList?)' >

<!ENTITY % CWMXExpress:PreComputeClauseAssociations '(%UML:ModelElementAssociations;
                                ,CWMXExpress:PreComputeClause.aggMapComponent?)' >

<!ENTITY % CWMXExpress:PreComputeClauseCompositions
                                '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXExpress:PreComputeClauseAttPropsList '%UML:ModelElementAttPropsList;' >

<!ENTITY % CWMXExpress:PreComputeClauseAttRefsList '%UML:ModelElementAttRefsList;
                                aggMapComponent IDREFS #IMPLIED' >

<!ELEMENT CWMXExpress:PreComputeClause ( %CWMXExpress:PreComputeClauseProperties;
                                ,(XML.extension* , %CWMXExpress:PreComputeClauseAssociations; )
                                , %CWMXExpress:PreComputeClauseCompositions; )?>

<!ATTLIST CWMXExpress:PreComputeClause %CWMXExpress:PreComputeClauseAttPropsList;
                                %CWMXExpress:PreComputeClauseAttRefsList; %XML.element.att;
                                %XML.link.att; >

```

3.8 *IMSDatabase.dtd*

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- Record.dtd -->
<!-- Model_Management.dtd -->

<!ATTLIST XMI xmlns:CWMXIMS CDATA #IMPLIED >

<!-- PACKAGE: CWMXIMS:IMSDatabase -->

<!ENTITY % CWMXIMS:AccessMethodType
    'xmi.value ( imsam_DEDB| imsam_GSAM| imsam_HDAM| imsam_HIDAM| imsam_HISAM|
                imsam_HSAM| imsam_INDEX| imsam_LOGICAL| imsam_MSDB|
                imsam_PSINDEX| imsam_PHDAM| imsam_PHIDAM| imsam_SHSAM|
                imsam_SHISAM ) #REQUIRED'>

<!ENTITY % CWMXIMS:PSBLanguageType
    'xmi.value ( imslt_ASSEM| imslt_COBOL| imslt_PLI| imslt_PASCAL )
    #REQUIRED'>

<!ENTITY % CWMXIMS:PCBType
    'xmi.value ( imsst_DB| imsst_GSAM| imsst_TP ) #REQUIRED'>

<!ENTITY % CWMXIMS:PositioningType
    'xmi.value ( imssp_Single| imssp_Multiple ) #REQUIRED'>

<!ENTITY % CWMXIMS:LTermType
    'xmi.value ( imstp_LTERM| imstp_NAME ) #REQUIRED'>

<!ENTITY % CWMXIMS:RulesType
    'xmi.value ( imsrst_FIRST| imsrst_LAST| imsrst_HERE ) #REQUIRED'>

<!ENTITY % CWMXIMS:ChildPointerType
    'xmi.value ( imscsp_SNGL| imscsp_DBLE| imscsp_NONE ) #REQUIRED'>

<!ENTITY % CWMXIMS:FlagsType
    'xmi.value ( imsst_P| imsst_L| imsst_V| imdft_B ) #REQUIRED'>

<!ENTITY % CWMXIMS:PointerType
    'xmi.value ( imspn_NOTWIN| imspn_TWIN| imspn_HIER| imspn_TWINBWD|
                imspn_HIERBWD ) #REQUIRED'>

<!ENTITY % CWMXIMS:DeviceType
    'xmi.value ( imsdtd_2305| imsdtd_2319| imsdtd_3330| imsdtd_3350| imsdtd_3375| imsdtd_3380|
                imsdtd_3390| imsdtd_2400| imsdtd_3400| imsdtd_TAPE ) #REQUIRED'>

<!ENTITY % CWMXIMS:ModelType
    'xmi.value ( imsdtd_1| imsdtd_2| imsdtd_11 ) #REQUIRED'>

<!ENTITY % CWMXIMS:RECFMType
    'xmi.value ( imsrft_F| imsrft_FB| imsrft_V| imsrft_VB| imsrft_U ) #REQUIRED'>

<!ENTITY % CWMXIMS:AlgorithmType
    'xmi.value ( imsat_0| imsat_1| imsat_2 ) #REQUIRED'>

<!ENTITY % CWMXIMS:LPointerType
    'xmi.value ( imslp_LTWIN| imslp_LTWINBWD ) #REQUIRED'>

<!ENTITY % CWMXIMS:MSDBtype
    'xmi.value ( imsmtd_NO| imsmtd_TERM| imsmtd_FIXED| imsmtd_DYNAMIC ) #REQUIRED'>

<!ENTITY % CWMXIMS:ParentType
    'xmi.value ( imsv_VIRTUAL| imsv_PHYSICAL ) #REQUIRED'>
```

```

<!-- PACKAGE: CWMXIMS:IMSTypes -->

<!-- ***** CWMXIMS:ExtendedByAccessMethod ***** -->
<!ELEMENT CWMXIMS:DBD.accessMethod ( CWMXIMS:AccessMethod
|CWMXIMS:INDEX
|CWMXIMS:HIDAM
|CWMXIMS:DEDB
|CWMXIMS:HDAM
|CWMXIMS:MSDB)? >

<!-- ***** CWMXIMS:IsIndexedBy ***** -->
<!ELEMENT CWMXIMS:SegmentComplex.secondaryIndex ( CWMXIMS:SecondaryIndex)* >

<!-- ***** CWMXIMS:IsLChild ***** -->
<!ELEMENT CWMXIMS:SegmentComplex.lchild ( CWMXIMS:LCHILD)* >

<!-- ***** CWMXIMS:ContainsDataset ***** -->
<!ELEMENT CWMXIMS:DBD.dataset ( CWMXIMS:Dataset)* >

<!-- ***** CWMXIMS:ContainsSegment ***** -->
<!ELEMENT CWMXIMS:DBD.segment ( CWMXIMS:Segment
|CWMXIMS:SegmentComplex
|CWMXIMS:SegmentLogical)* >

<!-- ***** CWMXIMS:PcbToSenSegment ***** -->
<!ELEMENT CWMXIMS:PCB.senSegment ( CWMXIMS:SenSegment)* >

<!-- ***** CWMXIMS:SensegToSenfld ***** -->
<!ELEMENT CWMXIMS:SenSegment.senField ( CWMXIMS:SenField)* >

<!-- ***** CWMXIMS:Captures ***** -->
<!ELEMENT CWMXIMS:DBD.exit ( CWMXIMS:Exit)* >

<!-- ***** CWMXIMS:CapturesExit ***** -->
<!ELEMENT CWMXIMS:Segment.exit ( CWMXIMS:Exit)* >

<!-- CLASS: CWMXIMS:DBD -->
<!ELEMENT CWMXIMS:DBD.dliAccess EMPTY>
<ATTLIST CWMXIMS:DBD.dliAccess %CWMXIMS:AccessMethodType;>

<!ELEMENT CWMXIMS:DBD.isVSAM EMPTY>
<ATTLIST CWMXIMS:DBD.isVSAM
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:DBD.passwordFlag EMPTY>

```

```

<!ATTLIST CWMXIMS:DBD.passwordFlag
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:DBD.versionString (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:DBD.acblib (CWMXIMS:ACBLIB)*>

<!ELEMENT CWMXIMS:DBD.library (CWMXIMS:DBDLib)*>

<!ELEMENT CWMXIMS:DBD.pcb (CWMXIMS:PCB)*>

<!ENTITY % CWMXIMS:DBDProperties '(%CWMREC:RecordFileProperties;
    ,(CWMXIMS:DBD.dliAccess)?
    ,(CWMXIMS:DBD.isVSAM)?
    ,(CWMXIMS:DBD.passwordFlag)?
    ,(CWMXIMS:DBD.versionString)?)' >

<!ENTITY % CWMXIMS:DBDAssociations '(%CWMREC:RecordFileAssociations;
    ,CWMXIMS:DBD.acblib*
    ,CWMXIMS:DBD.library*
    ,CWMXIMS:DBD.pcb*)' >

<!ENTITY % CWMXIMS:DBDCompositions '(%CWMREC:RecordFileCompositions;
    ,CWMXIMS:DBD.dataset*
    ,CWMXIMS:DBD.segment*
    ,CWMXIMS:DBD.exit*
    ,CWMXIMS:DBD.accessMethod?)' >

<!ENTITY % CWMXIMS:DBDAttPropsList '%CWMREC:RecordFileAttPropsList;
    dliAccess ( imsam_DEDB| imsam_GSAM| imsam_HDAM| imsam_HIDAM| imsam_HISAM|
    imsam_HSAM| imsam_INDEX| imsam_LOGICAL| imsam_MSDB|
    imsam_PSINDEX| imsam_PHDAM| imsam_PHIDAM| imsam_SHSAM|
    imsam_SHISAM ) #IMPLIED
    isVSAM (true|false) #IMPLIED
    passwordFlag (true|false) #IMPLIED
    versionString CDATA #IMPLIED' >

<!ENTITY % CWMXIMS:DBDAttRefsList '%CWMREC:RecordFileAttRefsList;' >

<!ELEMENT CWMXIMS:DBD ( %CWMXIMS:DBDProperties;
    ,(XML.extension* , %CWMXIMS:DBDAssociations; )
    , %CWMXIMS:DBDCompositions; )?>

<!ATTLIST CWMXIMS:DBD %CWMXIMS:DBDAttPropsList; %CWMXIMS:DBDAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:PSB -->

<!ELEMENT CWMXIMS:PSB.compatibility EMPTY>
<!ATTLIST CWMXIMS:PSB.compatibility
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PSB.ioErrorOption (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PSB.ioaSize (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PSB.language EMPTY>
<!ATTLIST CWMXIMS:PSB.language %CWMXIMS:PSBLanguageType;>

<!ELEMENT CWMXIMS:PSB.lockMaximum (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PSB.maximumQxCalls (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PSB.onlinelImageCopy EMPTY>
<!ATTLIST CWMXIMS:PSB.onlinelImageCopy
    xmi.value ( true | false ) #REQUIRED>

```

```

<!ELEMENT CWMXIMS:PSB.ssaSize (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PSB.writeToOperator EMPTY>
<!ATTLIST CWMXIMS:PSB.writeToOperator
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PSB.pcb (CWMXIMS:PCB)*>

<!ELEMENT CWMXIMS:PSB.library (CWMXIMS:PSBLib)*>

<!ELEMENT CWMXIMS:PSB.acblib (CWMXIMS:ACBLIB)*>

<!ENTITY % CWMXIMS:PSBProperties '(%CWMREC:RecordFileProperties;
    ,(CWMXIMS:PSB.compatibility)?
    ,(CWMXIMS:PSB.ioErrorOption)?
    ,(CWMXIMS:PSB.ioaSize)?
    ,(CWMXIMS:PSB.language)?
    ,(CWMXIMS:PSB.lockMaximum)?
    ,(CWMXIMS:PSB.maximumQxCalls)?
    ,(CWMXIMS:PSB.onlinelImageCopy)?
    ,(CWMXIMS:PSB.ssaSize)?
    ,(CWMXIMS:PSB.writeToOperator)?)' >

<!ENTITY % CWMXIMS:PSBAssociations '(%CWMREC:RecordFileAssociations;
    ,CWMXIMS:PSB.pcb*
    ,CWMXIMS:PSB.library*
    ,CWMXIMS:PSB.acblib*)' >

<!ENTITY % CWMXIMS:PSBCompositions '(%CWMREC:RecordFileCompositions;)' >

<!ENTITY % CWMXIMS:PSBAttPropsList '%CWMREC:RecordFileAttPropsList;
    compatibility (true|false) #IMPLIED
    ioErrorOption CDATA #IMPLIED
    ioaSize CDATA #IMPLIED
    language ( imslt_ASSEM| imslt_C| imslt_COBOL| imslt_PLI| imslt_PASCAL ) #IMPLIED
    lockMaximum CDATA #IMPLIED
    maximumQxCalls CDATA #IMPLIED
    onlinelImageCopy (true|false) #IMPLIED
    ssaSize CDATA #IMPLIED
    writeToOperator (true|false) #IMPLIED ' >

<!ENTITY % CWMXIMS:PSBAttRefsList '%CWMREC:RecordFileAttRefsList;' >

<!ELEMENT CWMXIMS:PSB ( %CWMXIMS:PSBProperties;
    ,(XML.extension* , %CWMXIMS:PSBAssociations; )
    , %CWMXIMS:PSBCompositions; )?>

<!ATTLIST CWMXIMS:PSB %CWMXIMS:PSBAttPropsList; %CWMXIMS:PSBAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:PCB -->

<!ELEMENT CWMXIMS:PCB.pcbType EMPTY>
<!ATTLIST CWMXIMS:PCB.pcbType %CWMXIMS:PCBType;>

<!ELEMENT CWMXIMS:PCB.list EMPTY>
<!ATTLIST CWMXIMS:PCB.list
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PCB.keyLength (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PCB.processingOptions (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:PCB.positioning EMPTY>
<!ATTLIST CWMXIMS:PCB.positioning %CWMXIMS:PositioningType;>

<!ELEMENT CWMXIMS:PCB.sequentialBuffering EMPTY>

```

```

<!ATTLIST CWMXIMS:PCB.sequentialBuffering
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PCB.alternateResponse EMPTY>
<!ATTLIST CWMXIMS:PCB.alternateResponse
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PCB.express EMPTY>
<!ATTLIST CWMXIMS:PCB.express
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PCB.modify EMPTY>
<!ATTLIST CWMXIMS:PCB.modify
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PCB.sameTerminal EMPTY>
<!ATTLIST CWMXIMS:PCB.sameTerminal
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:PCB.destinationType EMPTY>
<!ATTLIST CWMXIMS:PCB.destinationType %CWMXIMS:LTermType;>

<!ELEMENT CWMXIMS:PCB.ltermName (#PCDATA|XMI.reference)*>

<!ELEMENT CWMXIMS:PCB.procSeq (CWMXIMS:INDEX)?>

<!ELEMENT CWMXIMS:PCB.psb (CWMXIMS:PSB)*>

<!ELEMENT CWMXIMS:PCB.dbd (CWMXIMS:DBD)?>

<!ENTITY % CWMXIMS:PCBProperties '(%CWMREC:RecordFileProperties;
    ,(CWMXIMS:PCB.pcbType)?
    ,(CWMXIMS:PCB.list)?
    ,(CWMXIMS:PCB.keyLength)?
    ,(CWMXIMS:PCB.processingOptions)?
    ,(CWMXIMS:PCB.positioning)?
    ,(CWMXIMS:PCB.sequentialBuffering)?
    ,(CWMXIMS:PCB.alternateResponse)?
    ,(CWMXIMS:PCB.express)?
    ,(CWMXIMS:PCB.modify)?
    ,(CWMXIMS:PCB.sameTerminal)?
    ,(CWMXIMS:PCB.destinationType)?
    ,(CWMXIMS:PCB.ltermName)?)' >

<!ENTITY % CWMXIMS:PCBAssociations '(%CWMREC:RecordFileAssociations;
    ,CWMXIMS:PCB.procSeq?
    ,CWMXIMS:PCB.psb*
    ,CWMXIMS:PCB.dbd?)' >

<!ENTITY % CWMXIMS:PCBCompositions '(%CWMREC:RecordFileCompositions;
    ,CWMXIMS:PCB.senSegment*)' >

<!ENTITY % CWMXIMS:PCBAttPropsList '%CWMREC:RecordFileAttPropsList;
    pcbType ( imspt_DB| imspt_GSAM| imspt_TP ) #IMPLIED
    list (true|false) #IMPLIED
    keyLength CDATA #IMPLIED
    processingOptions CDATA #IMPLIED
    positioning ( imsp_Single| imsp_Multiple ) #IMPLIED
    sequentialBuffering (true|false) #IMPLIED
    alternateResponse (true|false) #IMPLIED
    express (true|false) #IMPLIED
    modify (true|false) #IMPLIED
    sameTerminal (true|false) #IMPLIED
    destinationType ( imstp_LTERM| imstp_NAME ) #IMPLIED
    ltermName CDATA #IMPLIED' >

<!ENTITY % CWMXIMS:PCBAttRefsList '%CWMREC:RecordFileAttRefsList;
    procSeq IDREFS #IMPLIED

```



```

dbd IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:PCB ( %CWMXIMS:PCBProperties;
,(XML.extension* , %CWMXIMS:PCBAssociations; )
,%CWMXIMS:PCBCompositions; )?>

<!ATTLIST CWMXIMS:PCB %CWMXIMS:PCBAttPropsList; %CWMXIMS:PCBAttRefsList;
%XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:Segment -->

<!ELEMENT CWMXIMS:Segment.exitFlag EMPTY>
<!ATTLIST CWMXIMS:Segment.exitFlag
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Segment.frequency (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIMS:Segment.maximumLength (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIMS:Segment.minimumLength (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIMS:Segment.rules EMPTY>
<!ATTLIST CWMXIMS:Segment.rules %CWMXIMS:RulesType;>
<!ELEMENT CWMXIMS:Segment.subsetPointers (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIMS:Segment.directDependent EMPTY>
<!ATTLIST CWMXIMS:Segment.directDependent
xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXIMS:Segment.pcPointer EMPTY>
<!ATTLIST CWMXIMS:Segment.pcPointer %CWMXIMS:ChildPointerType;>
<!ELEMENT CWMXIMS:Segment.dbd (CWMXIMS:DBD)?>
<!ELEMENT CWMXIMS:Segment.child (CWMXIMS:Segment)*>
<!ELEMENT CWMXIMS:Segment.parent (CWMXIMS:Segment)?>
<!ELEMENT CWMXIMS:Segment.logical (CWMXIMS:SegmentLogical)*>
<!ELEMENT CWMXIMS:Segment.senseg (CWMXIMS:SenSegment)*>
<!ENTITY % CWMXIMS:SegmentProperties '(%CWMREC:RecordDefProperties;
,(CWMXIMS:Segment.exitFlag)?
,(CWMXIMS:Segment.frequency)?
,(CWMXIMS:Segment.maximumLength)?
,(CWMXIMS:Segment.minimumLength)?
,(CWMXIMS:Segment.rules)?
,(CWMXIMS:Segment.subsetPointers)?
,(CWMXIMS:Segment.directDependent)?
,(CWMXIMS:Segment.pcPointer)?)' >
<!ENTITY % CWMXIMS:SegmentAssociations '(%CWMREC:RecordDefAssociations;
,CWMXIMS:Segment.dbd?
,CWMXIMS:Segment.child*
,CWMXIMS:Segment.parent?
,CWMXIMS:Segment.logical*
,CWMXIMS:Segment.senseg*)' >
<!ENTITY % CWMXIMS:SegmentCompositions '(%CWMREC:RecordDefCompositions;
,CWMXIMS:Segment.exit*)' >
<!ENTITY % CWMXIMS:SegmentAttPropsList '%CWMREC:RecordDefAttPropsList;
exitFlag (true|false) #IMPLIED
frequency CDATA #IMPLIED
maximumLength CDATA #IMPLIED

```

```

        minimumLength CDATA #IMPLIED
        rules ( imsrt_FIRST| imsrt_LAST| imsrt_HERE ) #IMPLIED
        subsetPointers CDATA #IMPLIED
        directDependent (true|false) #IMPLIED
        pcPointer ( imscp_SNGL| imscp_DBLE| imscp_NONE ) #IMPLIED' >

<!ENTITY % CWMXIMS:SegmentAttRefsList '%CWMREC:RecordDefAttRefsList;
        dbd IDREFS #IMPLIED
        parent IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:Segment ( %CWMXIMS:SegmentProperties;
        ,(XML.extension* , %CWMXIMS:SegmentAssociations; )
        , %CWMXIMS:SegmentCompositions; )?>

<!ATTLIST CWMXIMS:Segment %CWMXIMS:SegmentAttPropsList;
        %CWMXIMS:SegmentAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:SegmentComplex -->

<!ELEMENT CWMXIMS:SegmentComplex.deleteFlag EMPTY>
<!ATTLIST CWMXIMS:SegmentComplex.deleteFlag %CWMXIMS:FlagsType;>

<!ELEMENT CWMXIMS:SegmentComplex.insertFlag EMPTY>
<!ATTLIST CWMXIMS:SegmentComplex.insertFlag %CWMXIMS:FlagsType;>

<!ELEMENT CWMXIMS:SegmentComplex.replaceFlag EMPTY>
<!ATTLIST CWMXIMS:SegmentComplex.replaceFlag %CWMXIMS:FlagsType;>

<!ELEMENT CWMXIMS:SegmentComplex.segmPointer EMPTY>
<!ATTLIST CWMXIMS:SegmentComplex.segmPointer %CWMXIMS:PointerType;>

<!ELEMENT CWMXIMS:SegmentComplex.dsGroup (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:SegmentComplex.lparent (CWMXIMS:LCHILD)?>

<!ELEMENT CWMXIMS:SegmentComplex.pairedLCHILD (CWMXIMS:LCHILD)?>

<!ELEMENT CWMXIMS:SegmentComplex.sourcedIndex (CWMXIMS:SecondaryIndex)*>

<!ELEMENT CWMXIMS:SegmentComplex.dataset (CWMXIMS:Dataset)?>

<!ENTITY % CWMXIMS:SegmentComplexProperties '(%CWMXIMS:SegmentProperties;
        ,(CWMXIMS:SegmentComplex.deleteFlag)?
        ,(CWMXIMS:SegmentComplex.insertFlag)?
        ,(CWMXIMS:SegmentComplex.replaceFlag)?
        ,(CWMXIMS:SegmentComplex.segmPointer)?
        ,(CWMXIMS:SegmentComplex.dsGroup)?)' >

<!ENTITY % CWMXIMS:SegmentComplexAssociations '(%CWMXIMS:SegmentAssociations;
        ,CWMXIMS:SegmentComplex.lparent?
        ,CWMXIMS:SegmentComplex.pairedLCHILD?
        ,CWMXIMS:SegmentComplex.sourcedIndex*
        ,CWMXIMS:SegmentComplex.dataset?)' >

<!ENTITY % CWMXIMS:SegmentComplexCompositions '(%CWMXIMS:SegmentCompositions;
        ,CWMXIMS:SegmentComplex.lchild*
        ,CWMXIMS:SegmentComplex.secondaryIndex*)' >

<!ENTITY % CWMXIMS:SegmentComplexAttPropsList '%CWMXIMS:SegmentAttPropsList;
        deleteFlag ( imsft_P| imsft_L| imsft_V| imdft_B ) #IMPLIED
        insertFlag ( imsft_P| imsft_L| imsft_V| imdft_B ) #IMPLIED
        replaceFlag ( imsft_P| imsft_L| imsft_V| imdft_B ) #IMPLIED
        segmPointer ( imspn_NOTWIN| imspn_TWIN| imspn_HIER| imspn_TWINBWD|
                imspn_HIERBWD ) #IMPLIED
        dsGroup CDATA #IMPLIED' >

<!ENTITY % CWMXIMS:SegmentComplexAttRefsList '%CWMXIMS:SegmentAttRefsList;

```

```

        lparent IDREFS #IMPLIED
        pairedLCHILD IDREFS #IMPLIED
        dataset IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:SegmentComplex ( %CWMXIMS:SegmentComplexProperties;
,(XML.extension* , %CWMXIMS:SegmentComplexAssociations; )
, %CWMXIMS:SegmentComplexCompositions; )?>

<!ATTLIST CWMXIMS:SegmentComplex %CWMXIMS:SegmentComplexAttPropsList;
          %CWMXIMS:SegmentComplexAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXIMS:SegmentLogical -->

<!ELEMENT CWMXIMS:SegmentLogical.keyData1 EMPTY>
<!ATTLIST CWMXIMS:SegmentLogical.keyData1
  xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:SegmentLogical.keyData2 EMPTY>
<!ATTLIST CWMXIMS:SegmentLogical.keyData2
  xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:SegmentLogical.physical (CWMXIMS:Segment)*>

<!ENTITY % CWMXIMS:SegmentLogicalProperties '(%CWMXIMS:SegmentProperties;
,(CWMXIMS:SegmentLogical.keyData1)?
,(CWMXIMS:SegmentLogical.keyData2)?)' >

<!ENTITY % CWMXIMS:SegmentLogicalAssociations '(%CWMXIMS:SegmentAssociations;
,CWMXIMS:SegmentLogical.physical*)' >

<!ENTITY % CWMXIMS:SegmentLogicalCompositions '(%CWMXIMS:SegmentCompositions;)' >

<!ENTITY % CWMXIMS:SegmentLogicalAttPropsList '%CWMXIMS:SegmentAttPropsList;
  keyData1 (true|false) #IMPLIED
  keyData2 (true|false) #IMPLIED ' >

<!ENTITY % CWMXIMS:SegmentLogicalAttRefsList '%CWMXIMS:SegmentAttRefsList;' >

<!ELEMENT CWMXIMS:SegmentLogical ( %CWMXIMS:SegmentLogicalProperties;
,(XML.extension* , %CWMXIMS:SegmentLogicalAssociations; )
, %CWMXIMS:SegmentLogicalCompositions; )?>

<!ATTLIST CWMXIMS:SegmentLogical %CWMXIMS:SegmentLogicalAttPropsList;
          %CWMXIMS:SegmentLogicalAttRefsList; %XML.element.att;
          %XML.link.att; >

<!-- CLASS: CWMXIMS:Field -->

<!ELEMENT CWMXIMS:Field.sequenceField EMPTY>
<!ATTLIST CWMXIMS:Field.sequenceField
  xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Field.uniqueSequence EMPTY>
<!ATTLIST CWMXIMS:Field.uniqueSequence
  xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Field.length (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:Field.generated EMPTY>
<!ATTLIST CWMXIMS:Field.generated
  xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Field.ddataIndex (CWMXIMS:SecondaryIndex)*>

<!ELEMENT CWMXIMS:Field.searchIndex (CWMXIMS:SecondaryIndex)*>

```

```

<!ELEMENT CWMXIMS:Field.subseqIndex (CWMXIMS:SecondaryIndex)*>
<!ELEMENT CWMXIMS:Field.senField (CWMXIMS:SenField)*>
<!ENTITY % CWMXIMS:FieldProperties '(%CWMREC:FixedOffsetFieldProperties;
,(CWMXIMS:Field.sequenceField)?
,(CWMXIMS:Field.uniqueSequence)?
,(CWMXIMS:Field.length)?
,(CWMXIMS:Field.generated)?)' >
<!ENTITY % CWMXIMS:FieldAssociations '(%CWMREC:FixedOffsetFieldAssociations;
,CWMXIMS:Field.ddataIndex*
,CWMXIMS:Field.searchIndex*
,CWMXIMS:Field.subseqIndex*
,CWMXIMS:Field.senField*)' >
<!ENTITY % CWMXIMS:FieldCompositions '(%CWMREC:FixedOffsetFieldCompositions;)' >
<!ENTITY % CWMXIMS:FieldAttPropsList '%CWMREC:FixedOffsetFieldAttPropsList;
sequenceField (true|false) #IMPLIED
uniqueSequence (true|false) #IMPLIED
length CDATA #IMPLIED
generated (true|false) #IMPLIED ' >
<!ENTITY % CWMXIMS:FieldAttRefsList '%CWMREC:FixedOffsetFieldAttRefsList;' >
<!ELEMENT CWMXIMS:Field ( %CWMXIMS:FieldProperties;
,(XML:extension* , %CWMXIMS:FieldAssociations; )
,%CWMXIMS:FieldCompositions; )?>
<!ATTLIST CWMXIMS:Field %CWMXIMS:FieldAttPropsList; %CWMXIMS:FieldAttRefsList;
%XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:Dataset -->
<!ELEMENT CWMXIMS:Dataset.dd1name (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.device EMPTY>
<!ATTLIST CWMXIMS:Dataset.device %CWMXIMS:DeviceType;>
<!ELEMENT CWMXIMS:Dataset.model EMPTY>
<!ATTLIST CWMXIMS:Dataset.model %CWMXIMS:ModelType;>
<!ELEMENT CWMXIMS:Dataset.dd2name (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.size1 (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.size2 (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.recordLength1 (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.recordLength2 (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.blockingFactor1 (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.blockingFactor2 (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.datasetLabel (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.freeBlockFrequency (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.freeSpacePercentage (#PCDATA|XML:reference)*>
<!ELEMENT CWMXIMS:Dataset.recordFormat EMPTY>
<!ATTLIST CWMXIMS:Dataset.recordFormat %CWMXIMS:RECFMType;>

```

```

<!ELEMENT CWMXIMS:Dataset.scanCylinders (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:Dataset.searchAlgorithm EMPTY>
<!ATTLIST CWMXIMS:Dataset.searchAlgorithm %CWMXIMS:AlgorithmType;>

<!ELEMENT CWMXIMS:Dataset.root (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:Dataset.rootOverflow (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:Dataset.uow (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:Dataset.uowOverflow (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:Dataset.dbd (CWMXIMS:DBD)?>

<!ELEMENT CWMXIMS:Dataset.segment (CWMXIMS:SegmentComplex)*>

<!ENTITY % CWMXIMS:DatasetProperties '(%UML:ModelElementProperties;
    ,(CWMXIMS:Dataset.dd1name)?
    ,(CWMXIMS:Dataset.device)?
    ,(CWMXIMS:Dataset.model)?
    ,(CWMXIMS:Dataset.dd2name)?
    ,(CWMXIMS:Dataset.size1)?
    ,(CWMXIMS:Dataset.size2)?
    ,(CWMXIMS:Dataset.recordLength1)?
    ,(CWMXIMS:Dataset.recordLength2)?
    ,(CWMXIMS:Dataset.blockingFactor1)?
    ,(CWMXIMS:Dataset.blockingFactor2)?
    ,(CWMXIMS:Dataset.datasetLabel)?
    ,(CWMXIMS:Dataset.freeBlockFrequency)?
    ,(CWMXIMS:Dataset.freeSpacePercentage)?
    ,(CWMXIMS:Dataset.recordFormat)?
    ,(CWMXIMS:Dataset.scanCylinders)?
    ,(CWMXIMS:Dataset.searchAlgorithm)?
    ,(CWMXIMS:Dataset.root)?
    ,(CWMXIMS:Dataset.rootOverflow)?
    ,(CWMXIMS:Dataset.uow)?
    ,(CWMXIMS:Dataset.uowOverflow)?)' >

<!ENTITY % CWMXIMS:DatasetAssociations '(%UML:ModelElementAssociations;
    ,CWMXIMS:Dataset.dbd?
    ,CWMXIMS:Dataset.segment*)' >

<!ENTITY % CWMXIMS:DatasetCompositions '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXIMS:DatasetAttPropsList '%UML:ModelElementAttPropsList;
    dd1name CDATA #IMPLIED
    device ( imsd_2305| imsd_2319| imsd_3330| imsd_3350| imsd_3375| imsd_3380|
              imsd_3390| imsd_2400| imsd_3400| imsd_TAPE ) #IMPLIED
    model ( imsd_1| imsd_2| imsd_11 ) #IMPLIED
    dd2name CDATA #IMPLIED
    size1 CDATA #IMPLIED
    size2 CDATA #IMPLIED
    recordLength1 CDATA #IMPLIED
    recordLength2 CDATA #IMPLIED
    blockingFactor1 CDATA #IMPLIED
    blockingFactor2 CDATA #IMPLIED
    datasetLabel CDATA #IMPLIED
    freeBlockFrequency CDATA #IMPLIED
    freeSpacePercentage CDATA #IMPLIED
    recordFormat ( imsr_F| imsr_FB| imsr_V| imsr_VB| imsr_U ) #IMPLIED
    scanCylinders CDATA #IMPLIED
    searchAlgorithm ( imsat_0| imsat_1| imsat_2 ) #IMPLIED
    root CDATA #IMPLIED
    rootOverflow CDATA #IMPLIED
    uow CDATA #IMPLIED
    uowOverflow CDATA #IMPLIED' >

```

```

<!ENTITY % CWMXIMS:DatasetAttRefsList '%UML:ModelElementAttRefsList;
    dbd IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:Dataset ( %CWMXIMS:DatasetProperties;
    ,(XML.extension* , %CWMXIMS:DatasetAssociations; )
    , %CWMXIMS:DatasetCompositions; )?>

<!ATTLIST CWMXIMS:Dataset %CWMXIMS:DatasetAttPropsList;
    %CWMXIMS:DatasetAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:SenSegment -->

<!ELEMENT CWMXIMS:SenSegment.procoptSENSEG (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIMS:SenSegment.subsetPointers (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIMS:SenSegment.segment (CWMXIMS:Segment)?>
<!ELEMENT CWMXIMS:SenSegment.index (CWMXIMS:INDEX)*>
<!ELEMENT CWMXIMS:SenSegment.pcb (CWMXIMS:PCB)?>

<!ENTITY % CWMXIMS:SenSegmentProperties '(%CWMREC:RecordDefProperties;
    ,(CWMXIMS:SenSegment.procoptSENSEG)?
    ,(CWMXIMS:SenSegment.subsetPointers)?)' >

<!ENTITY % CWMXIMS:SenSegmentAssociations '(%CWMREC:RecordDefAssociations;
    ,CWMXIMS:SenSegment.segment?
    ,CWMXIMS:SenSegment.index*
    ,CWMXIMS:SenSegment.pcb?)' >

<!ENTITY % CWMXIMS:SenSegmentCompositions '(%CWMREC:RecordDefCompositions;
    ,CWMXIMS:SenSegment.senField*)' >

<!ENTITY % CWMXIMS:SenSegmentAttPropsList '%CWMREC:RecordDefAttPropsList;
    procoptSENSEG CDATA #IMPLIED
    subsetPointers CDATA #IMPLIED' >

<!ENTITY % CWMXIMS:SenSegmentAttRefsList '%CWMREC:RecordDefAttRefsList;
    segment IDREFS #IMPLIED
    pcb IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:SenSegment ( %CWMXIMS:SenSegmentProperties;
    ,(XML.extension* , %CWMXIMS:SenSegmentAssociations; )
    , %CWMXIMS:SenSegmentCompositions; )?>

<!ATTLIST CWMXIMS:SenSegment %CWMXIMS:SenSegmentAttPropsList;
    %CWMXIMS:SenSegmentAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXIMS:SenField -->

<!ELEMENT CWMXIMS:SenField.replace EMPTY>
<!ATTLIST CWMXIMS:SenField.replace
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:SenField.field (CWMXIMS:Field)?>
<!ELEMENT CWMXIMS:SenField.senSegment (CWMXIMS:SenSegment)?>

<!ENTITY % CWMXIMS:SenFieldProperties '(%CWMREC:FixedOffsetFieldProperties;
    ,(CWMXIMS:SenField.replace)?)' >

<!ENTITY % CWMXIMS:SenFieldAssociations '(%CWMREC:FixedOffsetFieldAssociations;
    ,CWMXIMS:SenField.field?
    ,CWMXIMS:SenField.senSegment?)' >

```

```

<!ENTITY % CWMXIMS:SenFieldCompositions '(%CWMREC:FixedOffsetFieldCompositions;)' >
<!ENTITY % CWMXIMS:SenFieldAttPropsList '%CWMREC:FixedOffsetFieldAttPropsList;
replace (true|false) #IMPLIED' >
<!ENTITY % CWMXIMS:SenFieldAttRefsList '%CWMREC:FixedOffsetFieldAttRefsList;
field IDREFS #IMPLIED
senSegment IDREFS #IMPLIED' >
<!ELEMENT CWMXIMS:SenField ( %CWMXIMS:SenFieldProperties;
,(XML.extension* , %CWMXIMS:SenFieldAssociations; )
,%CWMXIMS:SenFieldCompositions; )?>
<!ATTLIST CWMXIMS:SenField %CWMXIMS:SenFieldAttPropsList;
%CWMXIMS:SenFieldAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:ACBLIB -->
<!ELEMENT CWMXIMS:ACBLIB.dbd (CWMXIMS:DBD)*>
<!ELEMENT CWMXIMS:ACBLIB.psb (CWMXIMS:PSB)*>
<!ENTITY % CWMXIMS:ACBLIBProperties '(%UML:PackageProperties;)' >
<!ENTITY % CWMXIMS:ACBLIBAssociations '(%UML:PackageAssociations;
,CWMXIMS:ACBLIB.dbd*
,CWMXIMS:ACBLIB.psb*)' >
<!ENTITY % CWMXIMS:ACBLIBCompositions '(%UML:PackageCompositions;)' >
<!ENTITY % CWMXIMS:ACBLIBAttPropsList '%UML:PackageAttPropsList;' >
<!ENTITY % CWMXIMS:ACBLIBAttRefsList '%UML:PackageAttRefsList;' >
<!ELEMENT CWMXIMS:ACBLIB ( %CWMXIMS:ACBLIBProperties;
,(XML.extension* , %CWMXIMS:ACBLIBAssociations; )
,%CWMXIMS:ACBLIBCompositions; )?>
<!ATTLIST CWMXIMS:ACBLIB %CWMXIMS:ACBLIBAttPropsList;
%CWMXIMS:ACBLIBAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:AccessMethod -->
<!ELEMENT CWMXIMS:AccessMethod.dbd (CWMXIMS:DBD)?>
<!ENTITY % CWMXIMS:AccessMethodProperties '(%UML:ModelElementProperties;)' >
<!ENTITY % CWMXIMS:AccessMethodAssociations '(%UML:ModelElementAssociations;
,CWMXIMS:AccessMethod.dbd?)' >
<!ENTITY % CWMXIMS:AccessMethodCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXIMS:AccessMethodAttPropsList '%UML:ModelElementAttPropsList;' >
<!ENTITY % CWMXIMS:AccessMethodAttRefsList '%UML:ModelElementAttRefsList;
dbd IDREFS #IMPLIED' >
<!ELEMENT CWMXIMS:AccessMethod ( %CWMXIMS:AccessMethodProperties;
,(XML.extension* , %CWMXIMS:AccessMethodAssociations; )
,%CWMXIMS:AccessMethodCompositions; )?>
<!ATTLIST CWMXIMS:AccessMethod %CWMXIMS:AccessMethodAttPropsList;
%CWMXIMS:AccessMethodAttRefsList; %XML.element.att;
%XML.link.att; >

```

```

<!-- CLASS: CWMXIMS:INDEX -->

<!ELEMENT CWMXIMS:INDEX.dosCompatibility EMPTY>
<!ATTLIST CWMXIMS:INDEX.dosCompatibility
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:INDEX.protect EMPTY>
<!ATTLIST CWMXIMS:INDEX.protect
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:INDEX.primaryTarget (CWMXIMS:HIDAM)?>
<!ELEMENT CWMXIMS:INDEX.secondaryTarget (CWMXIMS:SecondaryIndex)?>
<!ELEMENT CWMXIMS:INDEX.sharedIndex (CWMXIMS:INDEX)?>
<!ELEMENT CWMXIMS:INDEX.sharingIndex (CWMXIMS:INDEX)*>
<!ELEMENT CWMXIMS:INDEX.senseg (CWMXIMS:SenSegment)*>
<!ELEMENT CWMXIMS:INDEX.sequencedPCB (CWMXIMS:PCB)*>

<!ENTITY % CWMXIMS:INDEXProperties '(%CWMXIMS:AccessMethodProperties;
    ,CWMXIMS:INDEX.dosCompatibility)?
    ,CWMXIMS:INDEX.protect)?' >

<!ENTITY % CWMXIMS:INDEXAssociations '(%CWMXIMS:AccessMethodAssociations;
    ,CWMXIMS:INDEX.primaryTarget?
    ,CWMXIMS:INDEX.secondaryTarget?
    ,CWMXIMS:INDEX.sharedIndex?
    ,CWMXIMS:INDEX.sharingIndex*
    ,CWMXIMS:INDEX.senseg*
    ,CWMXIMS:INDEX.sequencedPCB*)' >

<!ENTITY % CWMXIMS:INDEXCompositions '(%CWMXIMS:AccessMethodCompositions;)' >

<!ENTITY % CWMXIMS:INDEXAttPropsList '%CWMXIMS:AccessMethodAttPropsList;
    dosCompatibility (true|false) #IMPLIED
    protect (true|false) #IMPLIED' >

<!ENTITY % CWMXIMS:INDEXAttRefsList '%CWMXIMS:AccessMethodAttRefsList;
    primaryTarget IDREFS #IMPLIED
    secondaryTarget IDREFS #IMPLIED
    sharedIndex IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:INDEX ( %CWMXIMS:INDEXProperties;
    ,(XML.extension* , %CWMXIMS:INDEXAssociations; )
    , %CWMXIMS:INDEXCompositions; )?>

<!ATTLIST CWMXIMS:INDEX %CWMXIMS:INDEXAttPropsList; %CWMXIMS:INDEXAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:HIDAM -->

<!ELEMENT CWMXIMS:HIDAM.index (CWMXIMS:INDEX)?>

<!ENTITY % CWMXIMS:HIDAMProperties '(%CWMXIMS:AccessMethodProperties;)' >

<!ENTITY % CWMXIMS:HIDAMAssociations '(%CWMXIMS:AccessMethodAssociations;
    ,CWMXIMS:HIDAM.index?)' >

<!ENTITY % CWMXIMS:HIDAMCompositions '(%CWMXIMS:AccessMethodCompositions;)' >

<!ENTITY % CWMXIMS:HIDAMAttPropsList '%CWMXIMS:AccessMethodAttPropsList;' >

<!ENTITY % CWMXIMS:HIDAMAttRefsList '%CWMXIMS:AccessMethodAttRefsList;

```



```

        index IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:HIDAM ( %CWMXIMS:HIDAMProperties;
    ,(XML.extension* , %CWMXIMS:HIDAMAssociations; )
    , %CWMXIMS:HIDAMCompositions; )?>

<!-- ATTLIST CWMXIMS:HIDAM %CWMXIMS:HIDAMAttPropsList; %CWMXIMS:HIDAMAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:DEDB -->

<!ELEMENT CWMXIMS:DEDB.rmName (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:DEDB.stage (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:DEDB.extendedCall EMPTY>
<!-- ATTLIST CWMXIMS:DEDB.extendedCall
    xmi.value ( true | false ) #REQUIRED>

<!-- ENTITY % CWMXIMS:DEDBProperties '(%CWMXIMS:AccessMethodProperties;
    ,(CWMXIMS:DEDB.rmName)?
    ,(CWMXIMS:DEDB.stage)?
    ,(CWMXIMS:DEDB.extendedCall)?)' >

<!-- ENTITY % CWMXIMS:DEDBAssociations '(%CWMXIMS:AccessMethodAssociations;)' >

<!-- ENTITY % CWMXIMS:DEDBCompositions '(%CWMXIMS:AccessMethodCompositions;)' >

<!-- ENTITY % CWMXIMS:DEDBAttPropsList '%CWMXIMS:AccessMethodAttPropsList;
    rmName CDATA #IMPLIED
    stage CDATA #IMPLIED
    extendedCall (true|false) #IMPLIED ' >

<!-- ENTITY % CWMXIMS:DEDBAttRefsList '%CWMXIMS:AccessMethodAttRefsList;' >

<!-- ELEMENT CWMXIMS:DEDB ( %CWMXIMS:DEDBProperties;
    ,(XML.extension* , %CWMXIMS:DEDBAssociations; )
    , %CWMXIMS:DEDBCompositions; )?>

<!-- ATTLIST CWMXIMS:DEDB %CWMXIMS:DEDBAttPropsList; %CWMXIMS:DEDBAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:HDAM -->

<!-- ELEMENT CWMXIMS:HDAM.rmName (#PCDATA|XML.reference)*>

<!-- ELEMENT CWMXIMS:HDAM.relativeBlockNumber (#PCDATA|XML.reference)*>

<!-- ELEMENT CWMXIMS:HDAM.rootAnchorPoints (#PCDATA|XML.reference)*>

<!-- ELEMENT CWMXIMS:HDAM.rootMaxBytes (#PCDATA|XML.reference)*>

<!-- ENTITY % CWMXIMS:HDAMProperties '(%CWMXIMS:AccessMethodProperties;
    ,(CWMXIMS:HDAM.rmName)?
    ,(CWMXIMS:HDAM.relativeBlockNumber)?
    ,(CWMXIMS:HDAM.rootAnchorPoints)?
    ,(CWMXIMS:HDAM.rootMaxBytes)?)' >

<!-- ENTITY % CWMXIMS:HDAMAssociations '(%CWMXIMS:AccessMethodAssociations;)' >

<!-- ENTITY % CWMXIMS:HDAMCompositions '(%CWMXIMS:AccessMethodCompositions;)' >

<!-- ENTITY % CWMXIMS:HDAMAttPropsList '%CWMXIMS:AccessMethodAttPropsList;
    rmName CDATA #IMPLIED
    relativeBlockNumber CDATA #IMPLIED
    rootAnchorPoints CDATA #IMPLIED

```

```

    rootMaxBytes CDATA #IMPLIED' >

<!ENTITY % CWMXIMS:HDAMAttRefsList '%CWMXIMS:AccessMethodAttRefsList;' >

<!ELEMENT CWMXIMS:HDAM ( %CWMXIMS:HDAMProperties;
    ,(XML.extension* , %CWMXIMS:HDAMAssociations; )
    , %CWMXIMS:HDAMCompositions; )?>

<!ATTLIST CWMXIMS:HDAM %CWMXIMS:HDAMAttPropsList; %CWMXIMS:HDAMAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:MSDB -->

<!ELEMENT CWMXIMS:MSDB.msdbField (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:MSDB.msdbType EMPTY>
<!ATTLIST CWMXIMS:MSDB.msdbType %CWMXIMS:MSDBtype;>

<!ENTITY % CWMXIMS:MSDBProperties '(%CWMXIMS:AccessMethodProperties;
    ,(CWMXIMS:MSDB.msdbField)?
    ,(CWMXIMS:MSDB.msdbType)?)' >

<!ENTITY % CWMXIMS:MSDBAssociations '(%CWMXIMS:AccessMethodAssociations;)' >

<!ENTITY % CWMXIMS:MSDBCompositions '(%CWMXIMS:AccessMethodCompositions;)' >

<!ENTITY % CWMXIMS:MSDBAttPropsList '%CWMXIMS:AccessMethodAttPropsList;
    msdbField CDATA #IMPLIED
    msdbType ( imsmt_NO| imsmt_TERM| imsmt_FIXED| imsmt_DYNAMIC ) #IMPLIED' >

<!ENTITY % CWMXIMS:MSDBAttRefsList '%CWMXIMS:AccessMethodAttRefsList;' >

<!ELEMENT CWMXIMS:MSDB ( %CWMXIMS:MSDBProperties;
    ,(XML.extension* , %CWMXIMS:MSDBAssociations; )
    , %CWMXIMS:MSDBCompositions; )?>

<!ATTLIST CWMXIMS:MSDB %CWMXIMS:MSDBAttPropsList; %CWMXIMS:MSDBAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:SecondaryIndex -->

<!ELEMENT CWMXIMS:SecondaryIndex.constant (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:SecondaryIndex.exitRoutine (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:SecondaryIndex.nullValue (#PCDATA|XML.reference)*>

<!ELEMENT CWMXIMS:SecondaryIndex.index (CWMXIMS:INDEX)?>

<!ELEMENT CWMXIMS:SecondaryIndex.segment (CWMXIMS:SegmentComplex)?>

<!ELEMENT CWMXIMS:SecondaryIndex.indexSource (CWMXIMS:SegmentComplex)?>

<!ELEMENT CWMXIMS:SecondaryIndex.ddataFields (CWMXIMS:Field)*>

<!ELEMENT CWMXIMS:SecondaryIndex.searchFields (CWMXIMS:Field)*>

<!ELEMENT CWMXIMS:SecondaryIndex.subseqFields (CWMXIMS:Field)*>

<!ENTITY % CWMXIMS:SecondaryIndexProperties '(%UML:ModelElementProperties;
    ,(CWMXIMS:SecondaryIndex.constant)?
    ,(CWMXIMS:SecondaryIndex.exitRoutine)?
    ,(CWMXIMS:SecondaryIndex.nullValue)?)' >

<!ENTITY % CWMXIMS:SecondaryIndexAssociations '(%UML:ModelElementAssociations;
    ,CWMXIMS:SecondaryIndex.index?

```

```

,CWMXIMS:SecondaryIndex.segment?
,CWMXIMS:SecondaryIndex.indexSource?
,CWMXIMS:SecondaryIndex.ddataFields*
,CWMXIMS:SecondaryIndex.searchFields*
,CWMXIMS:SecondaryIndex.subseqFields*)' >

<!ENTITY % CWMXIMS:SecondaryIndexCompositions '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXIMS:SecondaryIndexAttPropsList '%UML:ModelElementAttPropsList;
constant CDATA #IMPLIED
exitRoutine CDATA #IMPLIED
nullValue CDATA #IMPLIED' >

<!ENTITY % CWMXIMS:SecondaryIndexAttRefsList '%UML:ModelElementAttRefsList;
index IDREFS #IMPLIED
segment IDREFS #IMPLIED
indexSource IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:SecondaryIndex ( %CWMXIMS:SecondaryIndexProperties;
,(XML.extension* , %CWMXIMS:SecondaryIndexAssociations; )
,%CWMXIMS:SecondaryIndexCompositions; )?>

<!ATTLIST CWMXIMS:SecondaryIndex %CWMXIMS:SecondaryIndexAttPropsList;
%CWMXIMS:SecondaryIndexAttRefsList; %XML.element.att;
%XML.link.att; >

<!-- CLASS: CWMXIMS:Exit -->

<!ELEMENT CWMXIMS:Exit.key EMPTY>
<!ATTLIST CWMXIMS:Exit.key
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.data EMPTY>
<!ATTLIST CWMXIMS:Exit.data
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.path EMPTY>
<!ATTLIST CWMXIMS:Exit.path
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.log EMPTY>
<!ATTLIST CWMXIMS:Exit.log
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.cascade EMPTY>
<!ATTLIST CWMXIMS:Exit.cascade
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.cascadeKey EMPTY>
<!ATTLIST CWMXIMS:Exit.cascadeKey
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.cascadeData EMPTY>
<!ATTLIST CWMXIMS:Exit.cascadeData
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.cascadePath EMPTY>
<!ATTLIST CWMXIMS:Exit.cascadePath
xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:Exit.dbd (CWMXIMS:DBD)?>

<!ELEMENT CWMXIMS:Exit.segment (CWMXIMS:Segment)?>

<!ENTITY % CWMXIMS:ExitProperties '(%UML:ModelElementProperties;
,(CWMXIMS:Exit.key)?
,(CWMXIMS:Exit.data)?

```

```

        ,(CWMXIMS:Exit.path)?
        ,(CWMXIMS:Exit.log)?
        ,(CWMXIMS:Exit.cascade)?
        ,(CWMXIMS:Exit.cascadeKey)?
        ,(CWMXIMS:Exit.cascadeData)?
        ,(CWMXIMS:Exit.cascadePath)?)' >

<!ENTITY % CWMXIMS:ExitAssociations '(%UML:ModelElementAssociations;
    ,CWMXIMS:Exit.dbd?
    ,CWMXIMS:Exit.segment?)' >

<!ENTITY % CWMXIMS:ExitCompositions '(%UML:ModelElementCompositions;) >

<!ENTITY % CWMXIMS:ExitAttPropsList '%UML:ModelElementAttPropsList;
    key (true|false) #IMPLIED
    data (true|false) #IMPLIED
    path (true|false) #IMPLIED
    log (true|false) #IMPLIED
    cascade (true|false) #IMPLIED
    cascadeKey (true|false) #IMPLIED
    cascadeData (true|false) #IMPLIED
    cascadePath (true|false) #IMPLIED' >

<!ENTITY % CWMXIMS:ExitAttRefsList '%UML:ModelElementAttRefsList;
    dbd IDREFS #IMPLIED
    segment IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:Exit ( %CWMXIMS:ExitProperties;
    ,(XML:extension* , %CWMXIMS:ExitAssociations; )
    , %CWMXIMS:ExitCompositions; )?>

<!ATTLIST CWMXIMS:Exit %CWMXIMS:ExitAttPropsList; %CWMXIMS:ExitAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:LCHILD -->

<!ELEMENT CWMXIMS:LCHILD.counter EMPTY>
<!ATTLIST CWMXIMS:LCHILD.counter
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:LCHILD.lcPointer EMPTY>
<!ATTLIST CWMXIMS:LCHILD.lcPointer %CWMXIMS:ChildPointerType;>

<!ELEMENT CWMXIMS:LCHILD.lparentFlag EMPTY>
<!ATTLIST CWMXIMS:LCHILD.lparentFlag
    xmi.value ( true | false ) #REQUIRED>

<!ELEMENT CWMXIMS:LCHILD.ltwIn EMPTY>
<!ATTLIST CWMXIMS:LCHILD.ltwIn %CWMXIMS:LPointerType;>

<!ELEMENT CWMXIMS:LCHILD.rules_ EMPTY>
<!ATTLIST CWMXIMS:LCHILD.rules_ %CWMXIMS:RulesType;>

<!ELEMENT CWMXIMS:LCHILD.virtualParent EMPTY>
<!ATTLIST CWMXIMS:LCHILD.virtualParent %CWMXIMS:ParentType;>

<!ELEMENT CWMXIMS:LCHILD.lparent (CWMXIMS:SegmentComplex)?>

<!ELEMENT CWMXIMS:LCHILD.lchild (CWMXIMS:SegmentComplex)?>

<!ELEMENT CWMXIMS:LCHILD.pairedSegment (CWMXIMS:SegmentComplex)?>

<!ENTITY % CWMXIMS:LCHILDProperties '(%UML:ModelElementProperties;
    ,(CWMXIMS:LCHILD.counter)?
    ,(CWMXIMS:LCHILD.lcPointer)?
    ,(CWMXIMS:LCHILD.lparentFlag)?
    ,(CWMXIMS:LCHILD.ltwIn)?

```

```

,(CWMXIMS:LCHILD.rules_)?
,(CWMXIMS:LCHILD.virtualParent)?' >

<!ENTITY % CWMXIMS:LCHILDAssociations '(%UML:ModelElementAssociations;
,CWMXIMS:LCHILD.lparent?
,CWMXIMS:LCHILD.lchild?
,CWMXIMS:LCHILD.pairedSegment?)' >

<!ENTITY % CWMXIMS:LCHILDCompositions '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXIMS:LCHILDAttPropsList '%UML:ModelElementAttPropsList;
counter (true|false) #IMPLIED
lcPointer ( imscp_SINGL| imscp_DBL| imscp_NONE ) #IMPLIED
lparentFlag (true|false) #IMPLIED
ltwin ( imslp_LTWIN| imslp_LTWINBWD ) #IMPLIED
rules ( imsrst_FIRST| imsrst_LAST| imsrst_HERE ) #IMPLIED
virtualParent ( ims_VIRTUAL| ims_PHYSICAL ) #IMPLIED' >

<!ENTITY % CWMXIMS:LCHILDAttRefsList '%UML:ModelElementAttRefsList;
lparent IDREFS #IMPLIED
lchild IDREFS #IMPLIED
pairedSegment IDREFS #IMPLIED' >

<!ELEMENT CWMXIMS:LCHILD ( %CWMXIMS:LCHILDProperties;
,(XML.extension* , %CWMXIMS:LCHILDAssociations; )
, %CWMXIMS:LCHILDCompositions; )?>

<!ATTLIST CWMXIMS:LCHILD %CWMXIMS:LCHILDAttPropsList;
%CWMXIMS:LCHILDAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:PSBLib -->

<!ELEMENT CWMXIMS:PSBLib.psb (CWMXIMS:PSB)*>

<!ENTITY % CWMXIMS:PSBLibProperties '(%UML:PackageProperties;)' >

<!ENTITY % CWMXIMS:PSBLibAssociations '(%UML:PackageAssociations;
,CWMXIMS:PSBLib.psb*)' >

<!ENTITY % CWMXIMS:PSBLibCompositions '(%UML:PackageCompositions;)' >

<!ENTITY % CWMXIMS:PSBLibAttPropsList '%UML:PackageAttPropsList;' >

<!ENTITY % CWMXIMS:PSBLibAttRefsList '%UML:PackageAttRefsList;' >

<!ELEMENT CWMXIMS:PSBLib ( %CWMXIMS:PSBLibProperties;
,(XML.extension* , %CWMXIMS:PSBLibAssociations; )
, %CWMXIMS:PSBLibCompositions; )?>

<!ATTLIST CWMXIMS:PSBLib %CWMXIMS:PSBLibAttPropsList; %CWMXIMS:PSBLibAttRefsList;
%XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIMS:DBDLib -->

<!ELEMENT CWMXIMS:DBDLib.dbd (CWMXIMS:DBD)*>

<!ENTITY % CWMXIMS:DBDLibProperties '(%UML:PackageProperties;)' >

<!ENTITY % CWMXIMS:DBDLibAssociations '(%UML:PackageAssociations;
,CWMXIMS:DBDLib.dbd*)' >

<!ENTITY % CWMXIMS:DBDLibCompositions '(%UML:PackageCompositions;)' >

<!ENTITY % CWMXIMS:DBDLibAttPropsList '%UML:PackageAttPropsList;' >

<!ENTITY % CWMXIMS:DBDLibAttRefsList '%UML:PackageAttRefsList;' >

```

```
<!ELEMENT CWMXIMS:DBDLib ( %CWMXIMS:DBDLibProperties;  
    ,(XML.extension* , %CWMXIMS:DBDLibAssociations; )  
    , %CWMXIMS:DBDLibCompositions; )?>  
  
<!ATTLIST CWMXIMS:DBDLib %CWMXIMS:DBDLibAttPropsList;  
    %CWMXIMS:DBDLibAttRefsList; %XML.element.att; %XML.link.att; >
```

3.9 *InformationSet.dtd*

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- Foundation.dtd -->
<!-- InformationVisualization.dtd -->
<!-- Expressions.dtd -->
<!-- BusinessInformation.dtd -->
<!-- Olap.dtd -->

<!ATTLIST XMI xmlns:CWMXIS CDATA #IMPLIED >

<!-- PACKAGE: CWMXIS:InformationSet -->

<!-- ***** CWMXIS:InformationSetReferencesInfoSetAdministration ***** -->

<!ELEMENT CWMXIS:InformationSet.infoSetAdmin ( CWMXIS:InfoSetAdministration)* >

<!-- CLASS: CWMXIS:InformationSet -->

<!ELEMENT CWMXIS:InformationSet.version (#PCDATA|XMI.reference)*>
<!ELEMENT CWMXIS:InformationSet.rule (CWMXIS:Rule)*>
<!ENTITY % CWMXIS:InformationSetProperties '(%CWMOLAP:SchemaProperties;
,(CWMXIS:InformationSet.version)?)' >
<!ENTITY % CWMXIS:InformationSetAssociations '(%CWMOLAP:SchemaAssociations;
,CWMXIS:InformationSet.rule*)' >
<!ENTITY % CWMXIS:InformationSetCompositions '(%CWMOLAP:SchemaCompositions;
,CWMXIS:InformationSet.infoSetAdmin*)' >
<!ENTITY % CWMXIS:InformationSetAttPropsList '%CWMOLAP:SchemaAttPropsList;
version CDATA #IMPLIED' >
<!ENTITY % CWMXIS:InformationSetAttRefsList '%CWMOLAP:SchemaAttRefsList;' >
<!ELEMENT CWMXIS:InformationSet ( %CWMXIS:InformationSetProperties;
,(XMI.extension* , %CWMXIS:InformationSetAssociations; )
,%CWMXIS:InformationSetCompositions; )?>
<!ATTLIST CWMXIS:InformationSet %CWMXIS:InformationSetAttPropsList;
%CWMXIS:InformationSetAttRefsList; %XMI.element.att; %XMI.link.att;
>

<!-- CLASS: CWMXIS:Segment -->

<!ELEMENT CWMXIS:Segment.regionSequence (UML:ProcedureExpression)>
<!ELEMENT CWMXIS:Segment.rule (CWMXIS:Rule)*>
<!ENTITY % CWMXIS:SegmentProperties '(%CWMOLAP:CubeProperties;
,(CWMXIS:Segment.regionSequence)?)' >
<!ENTITY % CWMXIS:SegmentAssociations '(%CWMOLAP:CubeAssociations;
,CWMXIS:Segment.rule*)' >
<!ENTITY % CWMXIS:SegmentCompositions '(%CWMOLAP:CubeCompositions;)' >
<!ENTITY % CWMXIS:SegmentAttPropsList '%CWMOLAP:CubeAttPropsList;
regionSequence CDATA #IMPLIED' >
```

```

<!ENTITY % CWMXIS:SegmentAttRefsList '%CWMOLAP:CubeAttRefsList;' >

<!ELEMENT CWMXIS:Segment ( %CWMXIS:SegmentProperties;
    ,(XML.extension* , %CWMXIS:SegmentAssociations; )
    , %CWMXIS:SegmentCompositions; )?>

<!ATTLIST CWMXIS:Segment %CWMXIS:SegmentAttPropsList; %CWMXIS:SegmentAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIS:SegmentRegion -->

<!ELEMENT CWMXIS:SegmentRegion.rule (CWMXIS:Rule)*>

<!ELEMENT CWMXIS:SegmentRegion.renderedObject (CWMIV:RenderedObject)*>

<!ENTITY % CWMXIS:SegmentRegionProperties '(%CWMOLAP:CubeRegionProperties;)' >

<!ENTITY % CWMXIS:SegmentRegionAssociations '(%CWMOLAP:CubeRegionAssociations;
    ,CWMXIS:SegmentRegion.rule*
    ,CWMXIS:SegmentRegion.renderedObject*)' >

<!ENTITY % CWMXIS:SegmentRegionCompositions '(%CWMOLAP:CubeRegionCompositions;)' >

<!ENTITY % CWMXIS:SegmentRegionAttPropsList '%CWMOLAP:CubeRegionAttPropsList;' >

<!ENTITY % CWMXIS:SegmentRegionAttRefsList '%CWMOLAP:CubeRegionAttRefsList;' >

<!ELEMENT CWMXIS:SegmentRegion ( %CWMXIS:SegmentRegionProperties;
    ,(XML.extension* , %CWMXIS:SegmentRegionAssociations; )
    , %CWMXIS:SegmentRegionCompositions; )?>

<!ATTLIST CWMXIS:SegmentRegion %CWMXIS:SegmentRegionAttPropsList;
    %CWMXIS:SegmentRegionAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXIS:Rule -->

<!ELEMENT CWMXIS:Rule.ruleExpression (CWM:ExpressionNode
|CWM:ConstantNode
|CWM:ElementNode
|CWM:FeatureNode)>

<!ELEMENT CWMXIS:Rule.type (#PCDATA|XML.reference)*>

<!ENTITY % CWMXIS:RuleProperties '(%UML:ModelElementProperties;
    ,(CWMXIS:Rule.ruleExpression)?
    ,(CWMXIS:Rule.type)?)' >

<!ENTITY % CWMXIS:RuleAssociations '(%UML:ModelElementAssociations;)' >

<!ENTITY % CWMXIS:RuleCompositions '(%UML:ModelElementCompositions;)' >

<!ENTITY % CWMXIS:RuleAttPropsList '%UML:ModelElementAttPropsList;
    ruleExpression CDATA #IMPLIED
    type CDATA #IMPLIED' >

<!ENTITY % CWMXIS:RuleAttRefsList '%UML:ModelElementAttRefsList;' >

<!ELEMENT CWMXIS:Rule ( %CWMXIS:RuleProperties;
    ,(XML.extension* , %CWMXIS:RuleAssociations; )
    , %CWMXIS:RuleCompositions; )?>

<!ATTLIST CWMXIS:Rule %CWMXIS:RuleAttPropsList; %CWMXIS:RuleAttRefsList;
    %XML.element.att; %XML.link.att; >

```



```

<!-- CLASS: CWMXIS:InfoSetAdministration -->
<!ELEMENT CWMXIS:InfoSetAdministration.priority (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIS:InfoSetAdministration.acknowledgement EMPTY>
<!ATTLIST CWMXIS:InfoSetAdministration.acknowledgement
    xmi.value ( true | false ) #REQUIRED>
<!ELEMENT CWMXIS:InfoSetAdministration.date (CWMXIS:InfoSetDate)*>
<!ELEMENT CWMXIS:InfoSetAdministration.responsibleParty (CWM:ResponsibleParty)*>
<!ELEMENT CWMXIS:InfoSetAdministration.document (CWM:Document)*>
<!ENTITY % CWMXIS:InfoSetAdministrationProperties '(%UML:ModelElementProperties;
    ,(CWMXIS:InfoSetAdministration.priority)?
    ,(CWMXIS:InfoSetAdministration.acknowledgement)?)' >
<!ENTITY % CWMXIS:InfoSetAdministrationAssociations '(%UML:ModelElementAssociations;
    ,CWMXIS:InfoSetAdministration.date*
    ,CWMXIS:InfoSetAdministration.responsibleParty*
    ,CWMXIS:InfoSetAdministration.document*)' >
<!ENTITY % CWMXIS:InfoSetAdministrationCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXIS:InfoSetAdministrationAttPropsList '%UML:ModelElementAttPropsList;
    priority CDATA #IMPLIED
    acknowledgement (true|false) #IMPLIED' >
<!ENTITY % CWMXIS:InfoSetAdministrationAttRefsList '%UML:ModelElementAttRefsList;' >
<!ELEMENT CWMXIS:InfoSetAdministration ( %CWMXIS:InfoSetAdministrationProperties;
    ,(XML.extension* , %CWMXIS:InfoSetAdministrationAssociations; )
    , %CWMXIS:InfoSetAdministrationCompositions; )?>
<!ATTLIST CWMXIS:InfoSetAdministration %CWMXIS:InfoSetAdministrationAttPropsList;
    %CWMXIS:InfoSetAdministrationAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXIS:InfoSetDate -->
<!ELEMENT CWMXIS:InfoSetDate.type (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIS:InfoSetDate.format (#PCDATA|XML.reference)*>
<!ELEMENT CWMXIS:InfoSetDate.dateTime (#PCDATA|XML.reference)*>
<!ENTITY % CWMXIS:InfoSetDateProperties '(%UML:ModelElementProperties;
    ,(CWMXIS:InfoSetDate.type)?
    ,(CWMXIS:InfoSetDate.format)?
    ,(CWMXIS:InfoSetDate.dateTime)?)' >
<!ENTITY % CWMXIS:InfoSetDateAssociations '(%UML:ModelElementAssociations;)' >
<!ENTITY % CWMXIS:InfoSetDateCompositions '(%UML:ModelElementCompositions;)' >
<!ENTITY % CWMXIS:InfoSetDateAttPropsList '%UML:ModelElementAttPropsList;
    type CDATA #IMPLIED
    format CDATA #IMPLIED
    dateTime CDATA #IMPLIED' >
<!ENTITY % CWMXIS:InfoSetDateAttRefsList '%UML:ModelElementAttRefsList;' >
<!ELEMENT CWMXIS:InfoSetDate ( %CWMXIS:InfoSetDateProperties;
    ,(XML.extension* , %CWMXIS:InfoSetDateAssociations; )
    , %CWMXIS:InfoSetDateCompositions; )?>

```

```
<!ATTLIST CWMXIS:InfoSetDate %CWMXIS:InfoSetDateAttPropsList;  
    %CWMXIS:InfoSetDateAttRefsList; %XML.element.att; %XML.link.att; >
```

3.10 *InformationReporting.dtd*

```
<!-- The following must precede this DTD: -->
<!-- XmiFixed.dtd -->
<!-- DataTypes.dtd -->
<!-- InformationVisualization.dtd -->
<!-- Foundation.dtd -->
<!-- Transformation.dtd -->

<!ATTLIST XMI xmlns:CWMXIR CDATA #IMPLIED >

<!-- PACKAGE: CWMXIR:InformationReporting -->

<!ENTITY % CWMXIR:ReportGroupType
    ' xmi.value ( header| footer| detail| other ) #REQUIRED'>

<!-- CLASS: CWMXIR:Report -->

<!ENTITY % CWMXIR:ReportProperties '(%CWMIV:RenderedObjectProperties;)' >
<!ENTITY % CWMXIR:ReportAssociations '(%CWMIV:RenderedObjectAssociations;)' >
<!ENTITY % CWMXIR:ReportCompositions '(%CWMIV:RenderedObjectCompositions;)' >
<!ENTITY % CWMXIR:ReportAttPropsList '%CWMIV:RenderedObjectAttPropsList;' >
<!ENTITY % CWMXIR:ReportAttRefsList '%CWMIV:RenderedObjectAttRefsList;' >

<!ELEMENT CWMXIR:Report ( %CWMXIR:ReportProperties;
    , (XML.extension* , %CWMXIR:ReportAssociations; )
    , %CWMXIR:ReportCompositions; )?>

<!ATTLIST CWMXIR:Report %CWMXIR:ReportAttPropsList; %CWMXIR:ReportAttRefsList;
    %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIR:ReportAttribute -->

<!ENTITY % CWMXIR:ReportAttributeProperties '(%UML:AttributeProperties;)' >
<!ENTITY % CWMXIR:ReportAttributeAssociations '(%UML:AttributeAssociations;)' >
<!ENTITY % CWMXIR:ReportAttributeCompositions '(%UML:AttributeCompositions;)' >
<!ENTITY % CWMXIR:ReportAttributeAttPropsList '%UML:AttributeAttPropsList;' >
<!ENTITY % CWMXIR:ReportAttributeAttRefsList '%UML:AttributeAttRefsList;' >

<!ELEMENT CWMXIR:ReportAttribute ( %CWMXIR:ReportAttributeProperties;
    , (XML.extension* , %CWMXIR:ReportAttributeAssociations; )
    , %CWMXIR:ReportAttributeCompositions; )?>

<!ATTLIST CWMXIR:ReportAttribute %CWMXIR:ReportAttributeAttPropsList;
    %CWMXIR:ReportAttributeAttRefsList; %XML.element.att; %XML.link.att;
    >

<!-- CLASS: CWMXIR:ReportExecution -->

<!ENTITY % CWMXIR:ReportExecutionProperties '(%CWMTFM:TransformationMapProperties;)' >
<!ENTITY % CWMXIR:ReportExecutionAssociations
    '(%CWMTFM:TransformationMapAssociations;)' >
```

```

<!ENTITY % CWMXIR:ReportExecutionCompositions
    '(%CWMTFM:TransformationMapCompositions;)' >

<!ENTITY % CWMXIR:ReportExecutionAttPropsList '(%CWMTFM:TransformationMapAttPropsList;
    >

<!ENTITY % CWMXIR:ReportExecutionAttRefsList '(%CWMTFM:TransformationMapAttRefsList;' >

<!ELEMENT CWMXIR:ReportExecution ( %CWMXIR:ReportExecutionProperties;
    ,(XML.extension* , %CWMXIR:ReportExecutionAssociations; )
    , %CWMXIR:ReportExecutionCompositions; )?>

<!ATTLIST CWMXIR:ReportExecution %CWMXIR:ReportExecutionAttPropsList;
    %CWMXIR:ReportExecutionAttRefsList; %XML.element.att;
    %XML.link.att; >

<!-- CLASS: CWMXIR:ReportField -->

<!ENTITY % CWMXIR:ReportFieldProperties '(%CWMIV:RenderedObjectProperties;)' >

<!ENTITY % CWMXIR:ReportFieldAssociations '(%CWMIV:RenderedObjectAssociations;)' >

<!ENTITY % CWMXIR:ReportFieldCompositions '(%CWMIV:RenderedObjectCompositions;)' >

<!ENTITY % CWMXIR:ReportFieldAttPropsList '(%CWMIV:RenderedObjectAttPropsList;' >

<!ENTITY % CWMXIR:ReportFieldAttRefsList '(%CWMIV:RenderedObjectAttRefsList;' >

<!ELEMENT CWMXIR:ReportField ( %CWMXIR:ReportFieldProperties;
    ,(XML.extension* , %CWMXIR:ReportFieldAssociations; )
    , %CWMXIR:ReportFieldCompositions; )?>

<!ATTLIST CWMXIR:ReportField %CWMXIR:ReportFieldAttPropsList;
    %CWMXIR:ReportFieldAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIR:ReportGroup -->

<!ELEMENT CWMXIR:ReportGroup.type EMPTY>
<!ATTLIST CWMXIR:ReportGroup.type %CWMXIR:ReportGroupType;>

<!ELEMENT CWMXIR:ReportGroup.reportQuery (CWM:QueryExpression)*>

<!ENTITY % CWMXIR:ReportGroupProperties '(%CWMIV:RenderedObjectProperties;
    ,(CWMXIR:ReportGroup.type)?)' >

<!ENTITY % CWMXIR:ReportGroupAssociations '(%CWMIV:RenderedObjectAssociations;
    ,CWMXIR:ReportGroup.reportQuery*)' >

<!ENTITY % CWMXIR:ReportGroupCompositions '(%CWMIV:RenderedObjectCompositions;)' >

<!ENTITY % CWMXIR:ReportGroupAttPropsList '(%CWMIV:RenderedObjectAttPropsList;
    type ( header| footer| detail| other ) #IMPLIED' >

<!ENTITY % CWMXIR:ReportGroupAttRefsList '(%CWMIV:RenderedObjectAttRefsList;' >

<!ELEMENT CWMXIR:ReportGroup ( %CWMXIR:ReportGroupProperties;
    ,(XML.extension* , %CWMXIR:ReportGroupAssociations; )
    , %CWMXIR:ReportGroupCompositions; )?>

<!ATTLIST CWMXIR:ReportGroup %CWMXIR:ReportGroupAttPropsList;
    %CWMXIR:ReportGroupAttRefsList; %XML.element.att; %XML.link.att; >

<!-- CLASS: CWMXIR:ReportPackage -->

<!ENTITY % CWMXIR:ReportPackageProperties '(%CWMIV:RenderedObjectSetProperties;)' >

```

```

<!ENTITY % CWMXIR:ReportPackageAssociations '(%CWMIV:RenderedObjectSetAssociations;)'
>

<!ENTITY % CWMXIR:ReportPackageCompositions
'(%CWMIV:RenderedObjectSetCompositions;)' >

<!ENTITY % CWMXIR:ReportPackageAttPropsList '%CWMIV:RenderedObjectSetAttPropsList;' >

<!ENTITY % CWMXIR:ReportPackageAttRefsList '%CWMIV:RenderedObjectSetAttRefsList;' >

<!ELEMENT CWMXIR:ReportPackage ( %CWMXIR:ReportPackageProperties;
,(XML.extension* , %CWMXIR:ReportPackageAssociations; )
, %CWMXIR:ReportPackageCompositions; )?>

<!ATTLIST CWMXIR:ReportPackage %CWMXIR:ReportPackageAttPropsList;
%CWMXIR:ReportPackageAttRefsList; %XML.element.att;
%XML.link.att; >

```