

MagicDraw file format

File format in MagicDraw 3.6-17.0

MagicDraw (or other modeling tool developed by us) file was a plain xmi file with extensions used to save diagrams information and project structure information (a mount table). The mdzip file was a compressed xmi file. The code engineering information was saved in the different file with extension "mdr".

File format in MagicDraw 17.0.1 and up

The inner project structure as well as the file format has been changed in version 17.0.1. An XMI version is upgraded to 2.4 with corresponding changes required by this standard.

A project is a collection of xmi, text, and binary resources. A native file format contains all resources saved as separate files. All these files are collected to one zip file.

Zip Based File Format (mdzip, xml.zip)

A zip file (xml.zip, mdzip) contains at least the following entries:

- Meta information about entries records.properties.
- Project identifier starts with PROJECT-, for example, PROJECT-f41072fa-d1aa-4e90-b7d9-fdd8862ed8af.
- Project structure information:

com.nomagic.ci.metamodel.project.

- Shared and not shared uml model information:

com.nomagic.magicdraw.uml_model.shared_model,

com.nomagic.magicdraw.uml_model.model

- Proxies information backup of other external resources. Starts with proxy.
- Project options ends with

com.nomagic.magicdraw.core.project.options.personalprojectoptions and

com.nomagic.magicdraw.core.project.options.commonprojectoptions.

- Diagrams and other "binary" resources, entry names do not contain any meaningful information, just resource id, like 80a7058a-91b1-41fb-9a3a-cd7a8c6be52d.

Plain Text File Format (xml, mdxml)

Export to a non zip file formats like .xml or .mdxml is a combination of all zip file entries into the one plain xmi file. In this file, a uml model is saved as it was in earlier versions, while other information is saved as xmi:Extension.

Changes Related to Diagrams

Until version 17.0.1, diagrams were saved as an xmi extension. All diagrams were saved under one extension <mdOwnedDiagrams>. Both a diagram model and diagram symbols were saved in this extension. The <mdOwnedDiagrams> extension was saved outside a uml:Model element.

Version 17.0.1 saves the diagram element as a separate element extension named <modelExtension>. This extension is stored in a place where the diagram element is serialized (inside a diagram owner). If one owner has more than one diagram, they may be written in the one extension. In this extension, a diagram element is saved as a normal model element, and all its attributes and references are saved according xmi rules. A diagram element contains DiagramRepresentationObject, which contains information about a diagram type, required features, and so on. DiagramRepresentationObject contains DiagramContentsDescriptor, which describes used elements in a diagram and contains BinaryObject. BinaryObject has the attribute "streamContentID" which identifies a zip entry for diagram symbols. All information is stored in <mdOwnedViews>. The following schema presents finding diagram symbols entry id:

Diagram > DiagramRepresentationObject > DiagramContentsDescriptor > BinaryObject > BinaryObject.streamContentID

Genmodel for DiagramRepresentationObject and DiagramContentsDescriptor

```
<?xml version="1.0" encoding="UTF-8"?>
<genmodel:GenModel xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI" xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
xmlns:genmodel="http://www.eclipse.org/emf/2002/GenModel" modelDirectory="
/com.nomagic.magicdraw.foundation/src"
modelPluginID="com.nomagic.magicdraw.foundation" modelName="Diagram
updateClasspath="false"
rootExtendsInterface="org.eclipse.emf.cdo.CDOObject"
rootExtendsClass="org.eclipse.emf.internal.cdo.CDOObjectImpl"
reflectiveDelegation="true" importerID="org.eclipse.emf.importer.ecore"
featureDelegation="Reflective"
containmentProxies="true" complianceLevel="5.0" copyrightFields="false"
usedGenPackages=".../com.nomagic.ci.metamodel.project/model/binary.
genmodel#//binary"
classNamePattern="">
  <foreignModel>diagram.ecore</foreignModel>
  <modelPluginVariables>CDO=org.eclipse.emf.cdo<
/modelPluginVariables>
  <genPackages prefix="Diagram" basePackage="com.nomagic.magicdraw.
foundation"
  disposableProviderFactory="true"
 .ecorePackage="diagram.ecore#/">
    <genClasses image="false".ecoreClass="diagram.ecore#//
AbstractDiagramRepresentationObject">
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/ID"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/type"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/umlType"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/diagramProperties"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/initialFrameSizeSet"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/requiredFeature"/>
      <genFeatures property="None" children="true"
createChild="true"
     .ecoreFeature="ecore:EReference diagram.ecore#//
AbstractDiagramRepresentationObject
/diagramContents"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//AbstractDiagramRepresentationObject
/diagramStyleID"/></genClasses>
    <genClasses.ecoreClass="diagram.ecore#
//DiagramContentsDescriptor">
      <genFeatures property="None" notify="false"
createChild="false"
     .ecoreFeature="ecore:EReference diagram.ecore#
//DiagramContentsDescriptor/representation"/>
      <genFeatures createChild="false".ecoreFeature="
ecore:EAttribute
      diagram.ecore#//DiagramContentsDescriptor
/exporterName"/>
```

```
ecore:EAttribute          <genFeatures createChild="false" ecoreFeature="
    diagram.ecore#//DiagramContentsDescriptor
/exporterVersion"/>
    <genFeatures createChild="false" ecoreFeature="
ecore:EAttribute diagram.ecore#//DiagramContentsDescriptor/usedElements"/>
    <genFeatures property="None" children="true"
createChild="true" ecoreFeature="ecore:EReference diagram.ecore#
//DiagramContentsDescriptor/ binaryObject"/>
    <genFeatures createChild="false" ecoreFeature="
ecore:EAttribute diagram.ecore#//DiagramContentsDescriptor/contentHash"/>
    </genClasses>
    </genPackages>
</genmodel:GenModel>
```