



MMBase2UML User's Guide

A UML Model generator for MMBase applications

Januari 12, 2004

Author: Kors van Beem

Revisions

Revision History
Revision v1.2

October 8, 2003

Kors van Beem

First public release

Table of Contents

1. What is MMBase2UML?	4
2. Quick start	5
3. Using MMBase2UML	6
3.1. Mapping of MMBase onto UML	6
3.2. MMBase2UML using the script	6
3.3. MMBase2UML using the ANT build file	6
3.4. Running MMBase2UML	7
3.5. Using the model	8
A. References	10

1. What is MMBase2UML?

MMBase2UML is a reversed engineering tool to analyse an existing MMBase application and create a representative UML model with. The application can read either one application, several applications or a remote cloud. For every MMBase application a XMI file is created. The created XMI file can be read by a modelling tool like Poseidon. MMBase2UML creates UML Class diagrams based on the UML 1.4 definition.

2. Quick start

After installation of UML2MMBase has been completed, the following steps can be followed for a quick start:

1. Install java 1.4. Tested with JDK1.4.1_02.
2. run the "mmbase2uml.bat" script and pass the directory name of an mmbase application. For example:

```
mmbase2uml.bat /tomcat/webapps/mmbase/WEB-INF/config/applications/APPLICATION. Make sure there is no backup data or testdata in the application direcytory.
```
3. Start Poseidon and open the newly created XMI file. This file can be found in the directory where MMBase2UML was run and has the name `name_of_application.xmi`. (E.g. `APPLICATION.xmi`).
4. Drag the classes on the left side of the screen in Poseidon into the empty diagram. The relations between the objects will appear automatically.

3. Using MMBase2UML

3.1. Mapping of MMBase onto UML

The UML model is build of information from the MMBase application. The mapping of the information from the application to the UML model elements is the same as in UML2MMBase. For more information about this mapping is therefore referenced to the UML2MMBase documentation.

3.2. MMBase2UML using the script

MMBase2UML is provided with a simple script that can be used to start the application: `mmbase2uml.bat`. The script is used to generate a UML model of one MMBase application or to process all application in a MMBase configuration. The script expect one parameter as argument which is the path to an MMBase application or application configuration directory. The generated UML model(s) will be stored in the current working directory in XML format.

3.3. MMBase2UML using the ANT build file

MMBase2UML is run with one or two parameters. The standard ant targets pass two parameters. The first one points to one or more mmbase application. The second one specifies the output path where the XML file should be written to. The following ant targets exist:

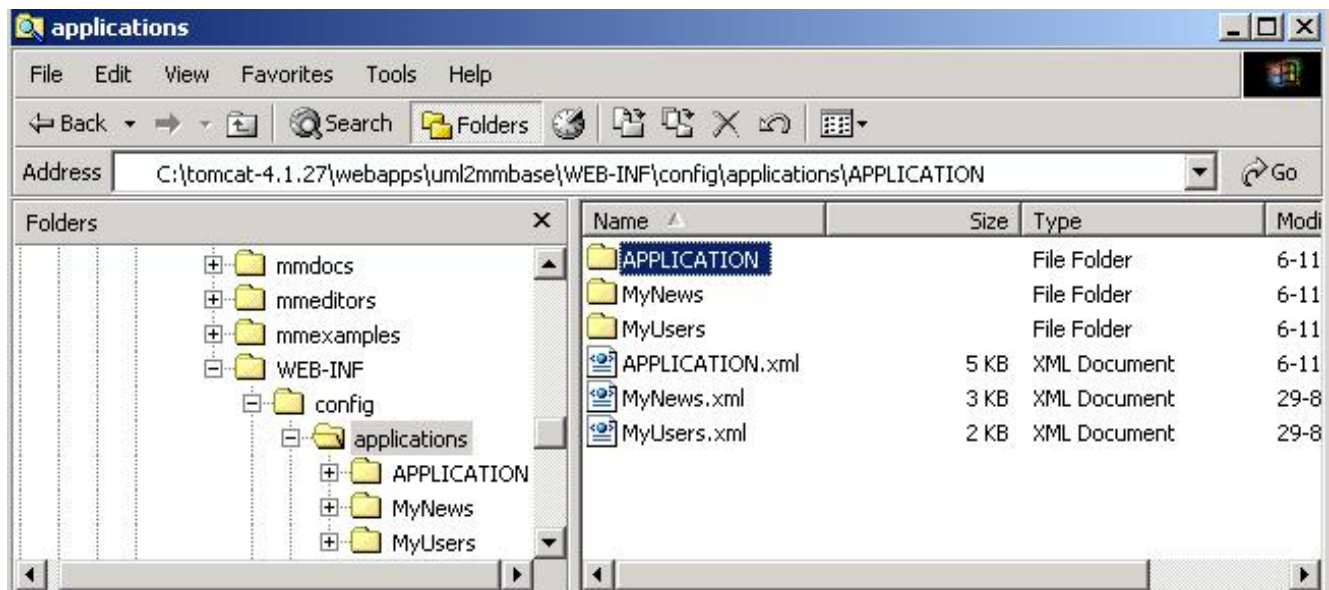
Table 1. The ant targets and the passed properties

ant target	first parameter	second parameter
<code>mmbase2uml.singleapp</code>	<code>single.app.dir</code>	<code>mmbase2uml.output.dir</code>
<code>mmbase2uml.multiapp</code>	<code>multi.app.dir</code>	<code>mmbase2uml.output.dir</code>
<code>mmbase2uml.remote</code>	<code>remote.app.dir</code>	<code>mmbase2uml.output.dir</code>

The `single.app.dir` property in the buildfile may point to a single application. The directory where the application resides should contain a directory that contains the builders and an XML file with the application definition. For example:

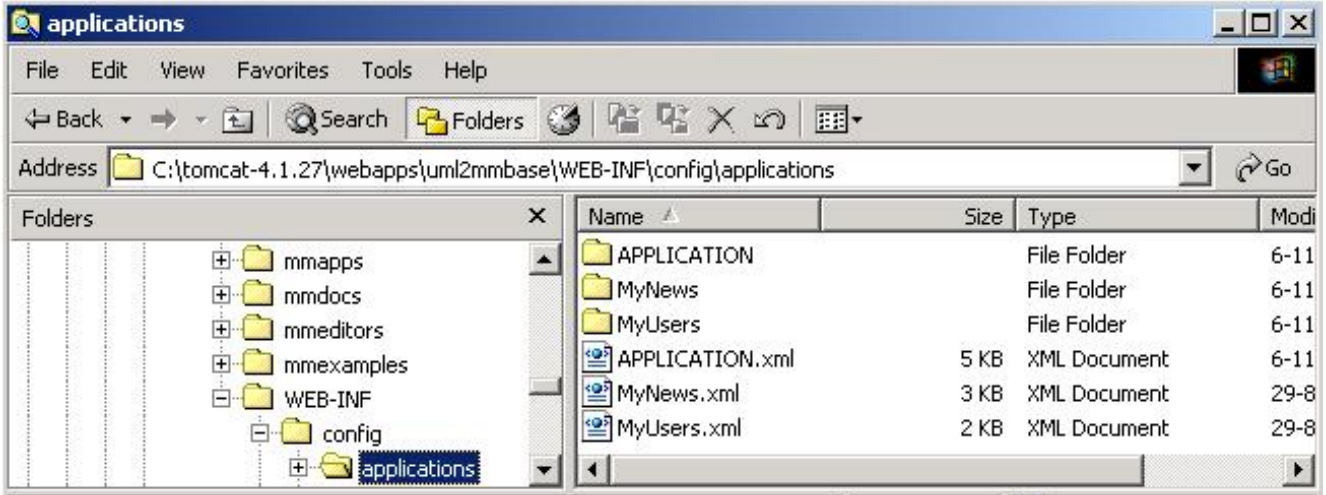
```
/tomcat/webapps/mmbase/WEB-INF/config/applications/APPLICATION
```

Figure 1. A single application reference



The `multi.app.dir` in the buildfile may point to a directory that contains more than one applications. The directory should contain one ore more combinations of application definitions and directories containing the builders of this definition. For example:

```
/tomcat/webapps/mmbase/WEB-INF/config/applications :
```

Figure 2. Multiple applications references

The `remote.app.url` property in the buildfile specifies an remote application. This url should look like `rmi://host:port/context`.

The `mmbase2uml.output.dir` property in the buildfile specifies the directory where the XML files should be written to.

3.4. Running MMBase2UML

When the parameters are set properly the application can be run using one of the three ant targets. This will result in something like this:

Figure 3. Running MMBase2UML

```
C:\> CMD.EXE
Buildfile: build.xml
check.mmbase.compiled:
compile-mmbase:
mmbase2uml.multiapp:
  [java] Parsing multiple offline applications
  [java] Found 3 application(s).
  [java] PARSING APPLICATION: APPLICATION
  [java] Creating UmlPackage.
  [java] Writing the model to a file.
  [java] PARSING APPLICATION: MyNews
  [java] Writing the model to a file.
  [java] PARSING APPLICATION: MyUsers
  [java] Writing the model to a file.

BUILD SUCCESSFUL
Total time: 17 seconds
C:\projects\uml2mmbase>dir d:*.xmi
Volume in drive D is disk2
Volume Serial Number is 2C23-BB8F

Directory of D:\

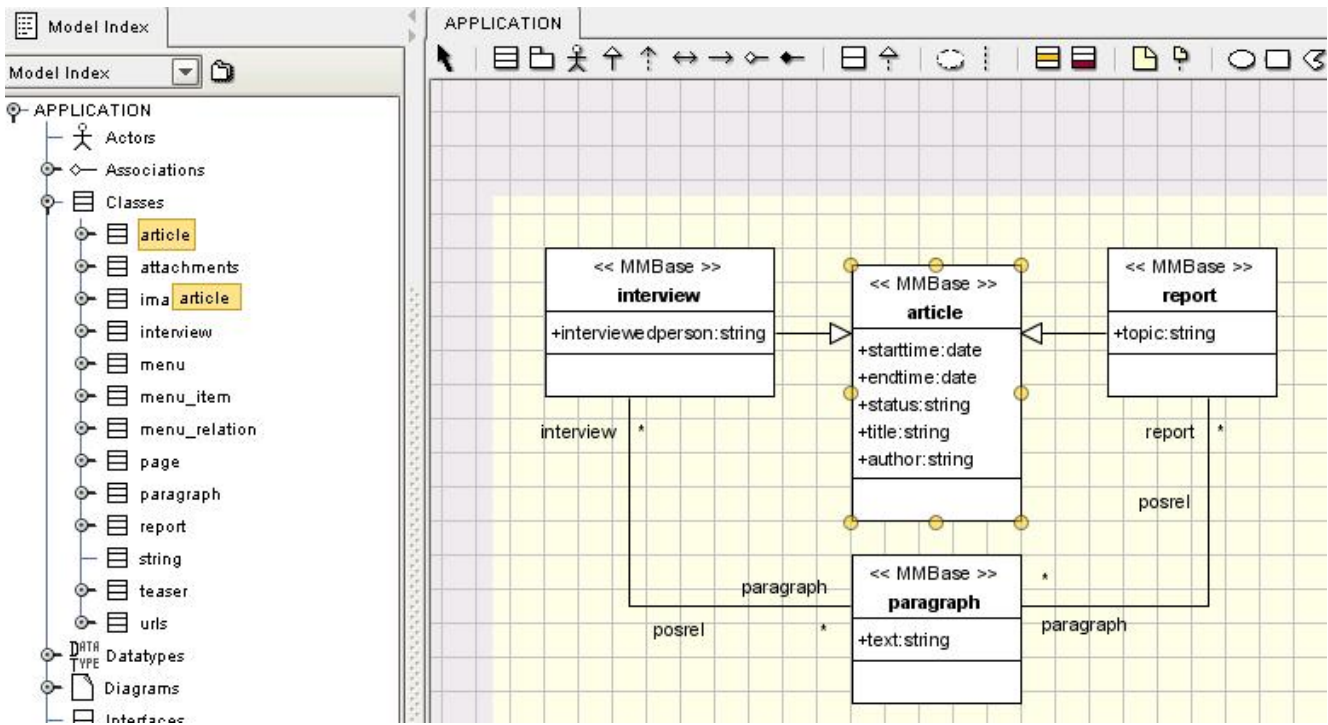
11-11-2003  15:09                47.759 APPLICATION.xmi
11-11-2003  15:09                34.782 MyNews.xmi
11-11-2003  15:09                30.159 MyUsers.xmi
             3 File(s)              112.700 bytes
             0 Dir(s)  40.902.565.888 bytes free

C:\projects\uml2mmbase>
```

3.5. Using the model

After the XML file is created the model can be opened in an UML modelling tool like Poseidon. Initially the Diagram in Poseidon will be empty. To make the classes and the associations visible, the classes can be dragged from the left side of the screen on to the diagram. The generalizations and associations will appear automatically:

Figure 4. Visualization of the UML model



References

- MMConfig: <http://mmapps.sourceforge.net/mmbaseconfig>
[<http://carlit.mine.nu/~keesj/share/org/mmbase/modules/config/package-summary.html>]
- AndroMDA: <http://www.andromda.org>
- Jakarta Ant: <http://ant.apache.org>
- Velocity: <http://jakarta.apache.org/velocity/index.html>
- PoseidonUML: <http://www.gentleware.com/products/index.php3>
- MagicDraw: <http://www.magicdraw.com> [<http://www.magicdraw.com/>]
- Together Control Center: <http://www.togethersoft.com>
- UML1.4 specification: <http://www.omg.org/docs/formal/01-09-67.pdf>
- MOF specification: <http://www.omg.org/docs/formal/00-04-03.pdf>
- JMI Implementations: <http://java.sun.com/products/jmi/implementations.html>
- JMI specification: <http://java.sun.com/products/jmi/download.html>