
HighQSoft



UML 2 ODS

Version 1.5

Manual

[Mark Quinsland](#) 10.09.2007

Contents

1	Introduction	1
2	Intended Audience	3
3	Installation	5
4	Use Cases	7
4.1	ODS to UML	7
4.2	UML to ODS	8
4.3	ODSdiff	8
4.4	Deltatool	9
5	Parameters	11
6	Glossary	13

Chapter 1

Introduction

The development of an ods-model is a quite complex process. To help in this it is required to bridge the gap between ODS and the standard-development- and modeling-tools in the UML-world. The foundation of the whole idea is the XMI-file. It is a well-defined standard for interchange of information in the UML-world. It is commonly used to transport work between UML-tools from different tool-suppliers. So its a kind of least-common-denominator in the UML-world.

Because of the use of XML, any UML-tool can be used that support XML version 1.4. To verify this package, we used e.g. ARGO-UML (free) and ...tbd..

The UML 2 ODS Tool provides a simple way to have an applicationmodel converted from an ODS format into and UML XMI file and vice versa. It builds a bridge between the two worlds of UML and ODS.

The tool can be used to do development-from-scratch and to enhance existing models. This allows to begin to use it in any stage of an project: new or already in use.



Chapter 2

Intended Audience

This toolset is addressing the needs of model-developers. It shall be used by people having experience in this area and a good knowledge about UML. This tool does not reduce the complexity of the overall process. It helps to do the job with tool-support. It does NOT replace the required knowledge and skills. It does not help to learn ODS and UML.

The user of UML2ODS has to know about UML, ODS and ATHOS. Other prerequisite is knowledge about XML-atf and CORBA. The user has to have knowledge of ATHOS-services. This concepts will not be explained in this document. See the ATHOS-documentation.

Chapter 3

Installation

The installation of the tool is quite simple. On the distribution CD, you will find an autoplay feature. If you do not wish to use the autoplay you can either call the **setup.exe** in the root directory of the CD or call the `install.jar` directly from the CDROM dir if your system is configured to open JAR files with the java runtime environment. Once the installer has started, you will be welcomed and receive a short information about the tool. Please choose the packages you wish to install afterwards. The files will be copied to the location specified. After completion, you will have the following structure:

```
[install dir]
|
--- bin
|   |
|   --- win32    << includes all necessary binaries
|
--- etc          << includes all configuration files, schema files etc.
|
--- docs         << Documentation
|
--- examples     << running examples of the tool to demonstrate the usage
```



Chapter 4

Use Cases

The toolset has different functions which are described in detail in the following sections. All parts of the toolset are packed into one java-program. For each of the functions, there is a batch- or shell-procedure that encapsulates the call to the java-program. These procedure have to be seen as samples. There will be many cases where these procedures are not sufficient or where the tool is encapsulated in an bigger application. And each of the functions can be used in different modes with different parameters. So the supplied procedures have to be seen as samples, but not as complete solution.

4.1 ODS to UML

This mode is typically used when an existing model shall be represented with UML-tools or when enhancements shall be done. First step is to access the model in its current physical implementation. This can be

- ods-datamodel is in an xml-atf-file
- ods-datamodel is in an database

The difference is just subtle details in the access-method. (athos/corba)

In directory <installation-path>/uml2ods/examples is a procedure ODS2UML_Gen_UML. It is a template how a call for this functionality has to look like. The following is a copy of the internal doc of the procedure. It explains in short all possible parameters. This operation can work in 2 modes:

- ods-source is a atf
- ods-source is a corba-service

The procedure is made so that it supports only the atf-mode. If corba-mode is required,

- the command has to be invoked manually
- the procedure has to be modified

```
rem This will run the HQ ODS-UML converter to produce an xmi file
rem for use with UML modeling tools.
```

```

rem
rem usage:  genuml
rem usage:  -service is the name of an athos-service. see athos-documentation
rem usage:  -uml is the fully-qualified path name to the xmi file to be generated
rem          or to be used as input.
rem usage:  -atf the fully-qualified path name to the atf or atfx file to be generated
rem          or to be used as input.
rem in case the source is not an ATF, the following optional parameters are required:
rem usage:  -port the Corba name service port optional
rem usage:  -host the Corba name service host optional
rem usage:  -pwd the password of a user authorized to make a connection to the ODS server
rem usage:  -user the name of a user authorized to make a connection to the ODS server
rem usage:  -type the type of connection to make to the server  values are Athos and CORBA. Default is Athos

```

4.2 UML to ODS

This mode is used to recreate a ods-copy of what was done in UML. All details for the parameters (services) is like in ods2uml.

In directory <installation-path>/uml2ods/examples is a proc UML2ODS_Gen. It is a template how a call for this functionality has to look like.

```

rem This will run the HQ ODS-UML converter to produce an atf or atfx file
rem usage:  genATF

rem usage:  -service is the servicename of the ODS service. This should be an atf or atfx service
rem usage:  -uml is the fully-qualified path name to the xmi file to be be used as input
rem usage:  -atf the fully-qualified path name to the atf or atfx file to be generated

```

4.3 ODSdiff

In directory <installation-path>/uml2ods/examples is a proc odsdiff. It is a template how a call for this functionality has to look like.

This will run a utility to produce an html file documenting the differences between 2 models. You must supply the connection information for each of the models. For convenience, they are called model A and model B. Other than the connection info, the only parameter is for where the output file should be sent.

You must ensure that the ODS2UML.jar and the other Athos jars are in the classpath.

```

usage: ODSdiff

usage:  -service service name for comparison model A
usage:  -atf name of atf file for comparison model A - if the service is an atf file service
usage:  -type service type ('athos', 'corba', 'ATF') name for comparison model A
usage:  -pwd authentication parm for comparison model A
usage:  -user authentication parm for comparison model A

usage:  -serviceB service name for comparison model B
usage:  -atfB name of atf file for comparison model B - if the service is an atf file service
usage:  -typeB service type ('athos', 'corba', 'ATF') name for comparison model B
usage:  -pwdB authentication parm for comparison model B
usage:  -userB authentication parm for comparison model B

usage:  -port Optional port where the corba name service can be found
usage:  -host Optional host where the corba name service can be found
usage:  -html The name of the output html file. Subdirectories must be valid

```

4.4 Deltatool

The purpose of the whole toolset is development of ods-datamodels. One case is that a model has to be extended. With dos2uml, uml2ods, odsdiff we can work in closed loop and find the difference between an old and an enhanced model. To have another name for it (and because we want to do complex things), we call the result DELTA. Next what we need, is a tool that applies this delta to the database. Remember: steps done was

- bring an existing model into uml
- modify it
- recreate the asam-representation
- make a DIFF on these 2 ods-datamodels
- have a DELTA

This DELTA must now be applied to the database. To enable this, the DELTa is expressed as a series of sql-commands that

- modify tables
- create tables
- delete tables
- handle constraints on relations

The DELTA is not automatically applied. It is generated and it is up to an very experienced database-professional to check it, correct it and then to apply. Her we are in the normal world of database-work: changing an existing database is an VERY challenging task. And this remains true even we use the ASAM-ODS-method.

Chapter 5

Parameters

Chapter 6

Glossary

Index

odsdiff, [8](#)