Enhydra DODS Quick start

Table of Contents

1. Introduction ....
2. System requirements ....
3. DODS source building ....
4. Starting DODS generator ....
    DODS generator wizard 4
5. Running DODS in ant ....
6. Running DODS in projects ....

Chapter 1. Introduction

This document describes how to build, get up and run quickly with DODS, specifically:

- How to build DODS
- How to start dods generator wizard
- How to start dods in projects

Chapter 2. System requirements

The minimum recommended system configuration is:

- 128 MB RAM
- 200 MHz processor or better
- 120 MB disk space (see note below)
- Java Development Kit (JDK) 1.4
- JDBC-compliant SQL database for database access

Note: The disk space specified above reflects the minimum system requirements to install and run the DODS example applications. A minimum installation consists of:

- JDK 1.4(approximately 50 MB)
- InstantDB (approximately 2 MB)

Chapter 3. DODS source building

- If Enhydra5.1 is installed, DODS is included within, so nothing more needs to be done.
- If DODS will be used independent (without Enhydra), it is necessary to do the following things:

  Edit file build.properties in the <DODS_SOURCE> directory. (<DODS_SOURCE> is location of DODS source).

  In build.properties file, you must specify the following variable:

  `jdk.dir - the path to your installation of the Java Development Kit`

  For example:

  `jdk.dir=C:/j2sdk1.4.0`

  Please note that Unix stile slashes (/) must always be used instead of Dos stile backslashes (\).

  Beside this variable, you can specify the variable:

  `dods.inst.dir`

  This is location of the directory in which DODS will be built.The default directory is "output".

  If you want to change location of any of external jars (ant.jar, optional.jar, log4j.jar, avalon-framework-cvs-20020315.jar, batik.jar, fop.jar, xerces.jar, xalan.jar, util.jar) or DocBook directory, you can specify the new location in this file. The procedure is the same as for directory for building Enhydra.
- If you don't have ant.bat file in your system PATH, you have to put it there.
- Start Command Promt and go to <DODS_SOURCE> directory.

  DODS building is completely Ant based. You can give one of the following options to the compile command:

  - compile buildAll - builds and configures DODS with javadoc and docbook documentation
  - compile buildNoDoc - builds and configures DODS without documentation (much faster, used for testing)
  - compile copyBuild - copies already built DODS
  - compile clean - removes the <DODS_HOME> (in order to start a new compilation from scratch)

  where <DODS_HOME> is directory in which DODS is built.
- After DODS building, you MUST add <DODS_HOME>\bin directory to the begining of the system PATH.

Chapter 4. Starting DODS generator

If you are running on Windows, to start DODS generator, you should use batch files in

- <DODS_HOME>/bin directory, for independent DODS, or
- <ENHYDRA_HOME>/bin directory, for DODS within Enhydra 5.1

where <ENHYDRA_HOME> is home (root) directory of Enhydra 5.1

At the command window, change directory to the /bin directory.

For example,

- for independent DODS:

  ```
  cd /usr/local/<DODS_HOME>/bin
  ```
  if you installed DODS in /usr/local/<DODS_HOME> directory

- for DODS within Enhydra 5.1:

  ```
  cd /usr/local/<ENHYDRA_HOME>/bin
  ```
  if you installed Enhydra 5.1 in /usr/local/<ENHYDRA_HOME> directory

The dods generator start batch file is located in this bin directory. So, to start the dods generator, enter:

```
dods
```

If you are running Linux, you can use start scripts in the same directory. So to start the dods generator on Linux, enter:

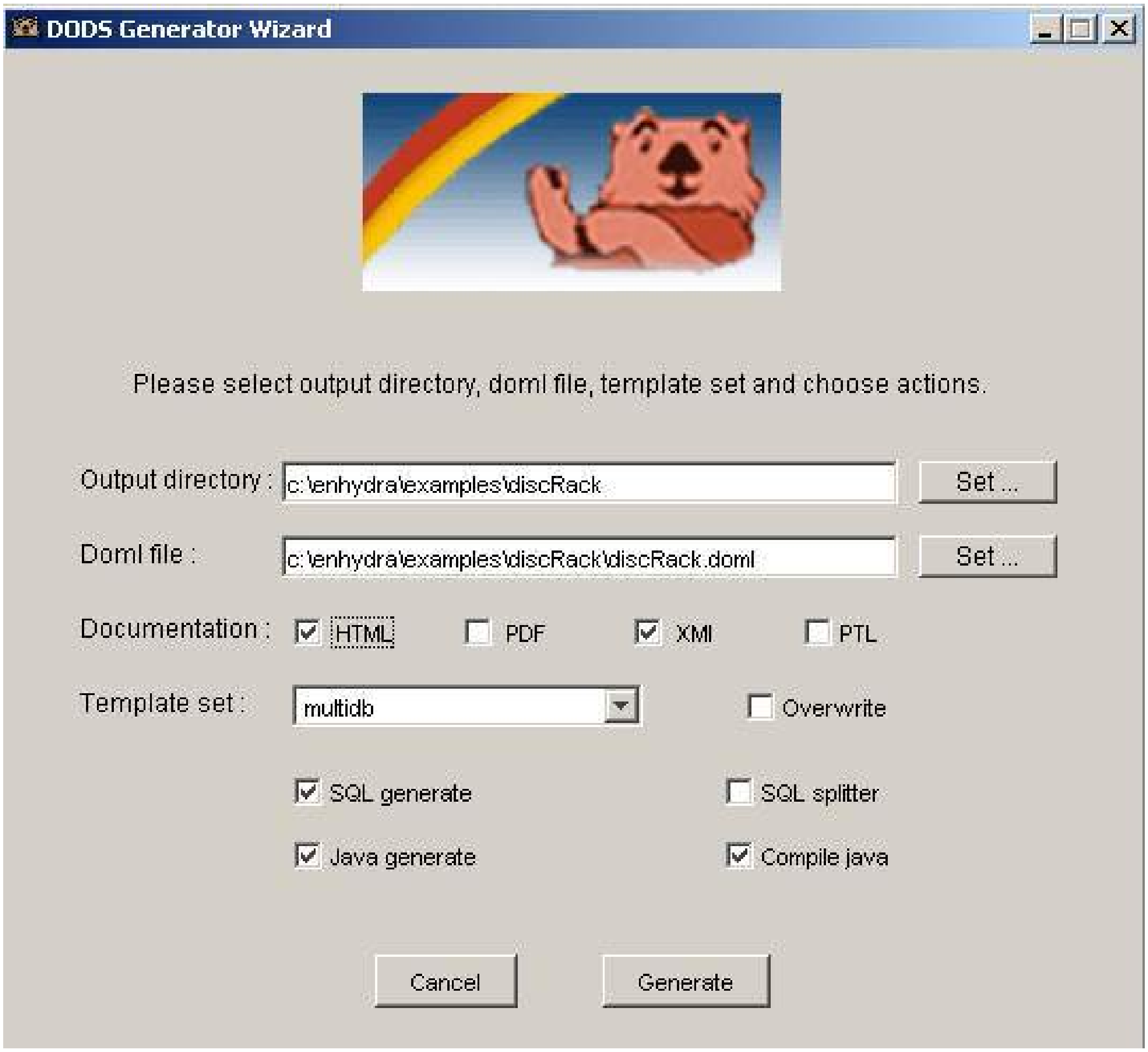
```
./dods
```

- Note:

  <DODS_HOME>/bin directory should be added in system path. Then, DODS can be started from any directory (by typing dods).

You will see dods generator wizard displayed in the output window.

DODS generator wizard



Please select output directory, doml file, template set and choose actions.

Output directory : c:\enhydra\examples\discRack

Doml file : c:\enhydra\examples\discRack\discRack.doml

Documentation : ☑ HTML ☐ PDF ☑ XMI ☐ PTL

Template set : multidb ☐ Overwrite

☑ SQL generate ☐ SQL splitter

☑ Java generate ☑ Compile java

Figure 1: DODS Generator Wizard

- You must choose project root and doml file.
- There are four options on the Generator Wizard:

  - SQL generate
  - SQL Splitter
  - Java generate
  - Compile Java

  At least one of the Generate fields must be checked.

- There is one combo box on the Generator Wizard. It contains template sets:
    - standard
    - multidb
    - webdocwf
    - multidb_webdocwf
    - <user_defined_templates>
- There is a possibility on the Generator Wizard for generating four types of documentation:
    - HTML
    - PDF
    - XMI
    - PTL
- On the Generator Wizard, there is also a check box:

  `overwrite`
  for code generating (java and sql), no matter if the code already existed.

You can manually start dods generator without using wizard by typing dods with additional parameters:

Command line:

```
        dods  [-?/help] [-a action] [-t templateset] [-b/-database]  [-f/force]
              [-h/html] [-p/pdf] [-x/xmi] [-r/ptl] domlfile outputdir
```

where:

- outputdir is full path to output directory that will be used.
- domlfile is full path to .doml file for generating code.

options:

- [-? -help] shows help.
- [-a action] - ant task parameter for code generation:
    - dods:build_all - to create all sql files and java classes (default).
    - dods:sql - to create only sql files.
    - dods:java - to create only java files and to compile them.

  - dods:javaNoCompile - to create only java files and not to compile them.
  - dods:noCompile - to create SQL files and java files and not to compile them.
  - dods:build_all_split - to create all sql files and java classes and to compile it. SQL files will be divided into separate files using SQLSplitter .
  - dods:sqlsplit - to create only sql files and separate in different files using SQLSplitter.
  - dods:noCompileSplit - to create SQL files and separate sql commands using SQLSplitter and java files and not to compile them.
- [-t templateset] - template set for generating java and sql code:
  - standard - generate standard java code (default).
  - multidb - generate java code with multi database support.
  - webdocwf - generate java code with WebDocWF support.
  - multidb_webdocwf - generate java code with multi database and WebDocWF support.
  - <user defined> - any user defined template set.
- [-b/-database] - sets database vendor for generating sql
- [-f/-force] - with this switch, code will be always generated, without it, only changes will be regenerated.
- [-h/-html] - generates DODS html documentation from .doml file.
- [-p/-pdf] - generates DODS pdf documentation from .doml file.
- [-x/-xmi] - generates DODS xmi documentation from .doml file.
- [-r/-ptl] - generates DODS ptl (Rational Rose) documentation from .doml file.

Chapter 5. Running DODS in ant

Invokes DODS to generate a set of java classes from a doml file. The files will only be regenerated/compiled if the date on the doml file is newer than at least one of the generated files.

This taskdef extends Ant's <javac> task; refer to documentation for parameters that affect compilation.

Typically made visible to an Ant build file with the following declaration:

```
<taskdef name="dods" classname="org.enhydra.ant.taskdefs.Dods"/>
```

Parameters:

doml - The doml input file describing data object mapping. Required = Yes

outputDir - Target for generated classes, expressed as a directory path. Required = Yes

force - Forces DODS always to regenerate source files. Possible values: ("true", "false"(default)). Required = No

action - Name of Ant task from generate.xml. Required = No

templateSet - Template template set for generating java code. Required = No

database - sets database vendor for generating sql. Required = No.

html - Indicates DODS to generate html documentation from .doml file. Possible values: ("true", "false"(default)). Required = No

pdf - Indicates DODS to generate pdf documentation from .doml file. Possible values: ("true", "false"(default)). Required = No

xmi - Indicates DODS to generate xmi documentation from .doml file. Possible values: ("true", "false"(default)). Required = No

ptl - Indicates DODS to generate ptl (Rational Rose) documentation from .doml file. Possible values: ("true", "false"(default)). Required = No

action parameters:

- without parameters - to create all sql files and java classes and to compile it.
- dods:build_all - to create all sql files and java classes.
- dods:sql - to create only sql files.
- dods:java -to create only java files and to compile them.
- dods:javaNoCompile -To create only java files and not to compile them.
- dods:noCompile -To create SQL files and java files and not to compile them.
- dods:build_all_split - to create all sql files and java classes and to compile it. SQL files will be divided into separate files using SQLSplitter
- dods:sqlsplit - to create only sql files and separate in different files using SQLSplitter.
- dods:noCompileSplit - To create SQL files and separate sql commands using SQLSplitter and java files and not

to compile them.

templateset parameters:

- standard - generate standard java code.
- multidb - generate java code with multi database support.
- webdocwf - generate java code with WebDocWF support.
- multidb_webdocwf - generate java code with multi database and WebDocWF support.
- <user defined> - any user defined template set.

Example:

```
<dods doml="${basedir}/discRack.doml"
         outputDir="${basedir}/src"
                 templateset="multidb"/>
```

Chapter 6. Running DODS in projects

In the project root directory, there are three files: build_dods.xml, build_java.xml and build_sql.xml that are used for generating sql and java files.

- build_dods.xml is used to create all sql files and java classes and to compile them.
- build_sql.xml is used to create only sql files.
- build_java.xml is used to create only java files and to compile them.

Sql and java files will be generated into folder which is defined by package in doml file.

Classes will be created into <project_root>/classes directory.