

**OD.doc**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> OD.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 30, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>OD.doc</b>	<b>1</b>
1.1	OD -- The Oberon-A module definition utility . . . . .	1
1.2	What is OD? . . . . .	1
1.3	A technical description of OD . . . . .	2
1.4	Copyright and distribution . . . . .	2
1.5	What do I need to run OD? . . . . .	2
1.6	Running OD from the CLI . . . . .	3
1.7	Running OD from the Workbench . . . . .	3
1.8	Running OD from the FPE utility . . . . .	3
1.9	Contacting the author . . . . .	4
1.10	Reporting bugs and suggestions . . . . .	4
1.11	Who did what and why . . . . .	4
1.12	Release History . . . . .	4

# Chapter 1

## OD.doc

### 1.1 OD -- The Oberon-A module definition utility

\$RCSfile: OD.doc \$

Description: Documentation for the Oberon-A module definition utility.

Created by: fjc (Frank Copeland)

\$Revision: 1.1 \$

\$Author: fjc \$

\$Date: 1994/08/08 21:38:40 \$

Description	What is OD?
Distribution	Copyright and distribution
Requirements	What do I need to run OD?

Running OD...

From the CLI

From the Workbench

From FPE

The Author	Contacting the author
Bugs & Suggestions	Reporting bugs and suggestions
Acknowledgements	Who did what and why
Changes	Changes since the last release
To Do	Bugs to fix and improvements to make
Release history	The history of OD

### 1.2 What is OD?

OD is the Oberon-A module definition utility. Its purpose is to create a summary of the objects exported by a module, to act as a reference for programmers. It is similar in most ways to the Oberon System's 'browser' utility.

The definition file created by OD closely resembles an Oberon-2 module containing only declarations and procedure headings. It is produced directly from a module's symbol file, and contains only those

declarations exported by the module and visible to its clients. It is structured roughly as follows:

```
DEFINITION <module>;

CONST
  <name> = <value>;
  ...

TYPE
  <name> = <type>;
  ...

VAR
  <name> : <type>;
  ...

<procedure heading>
...

END <module>.
```

Within each division, identifiers are listed alphabetically. It is not possible to reproduce the structure of the original module.

Type-bound procedures and library call procedures are shown as part of the declaration of the associated record type.

---

Implementation    A technical description of OD

## 1.3 A technical description of OD

THIS SPACE INTENTIONALLY LEFT BLANK

## 1.4 Copyright and distribution

OD is part of Oberon-A and is:

Copyright © 1994, Frank Copeland

See Oberon-A.doc for its conditions of use and distribution.

## 1.5 What do I need to run OD?

OD requires Release 2.04 (V37) of the Amiga operating system, or a later version.

---

## 1.6 Running OD from the CLI

Format: OD [FROM] <file | pattern> [TO <directory>] [VERBOSE]  
Template: FROM/A/M, TO/K, VERBOSE/S  
Purpose: Generates definition files from symbol files.  
Path: OBERON-A:C/OD

Specification: OD reads the symbol file specified in the FROM parameter and generates a definition file which is output in the directory specified in the TO parameter, or in the current directory if there is none. You can specify several symbol files to be processed by giving multiple FROM arguments, or by using AmigaDOS pattern matching.

OD uses the standard AmigaDOS pattern matching routines, so the FROM arguments must fully specify the symbol file names, including the ".Sym" extension. If a TO parameter is given, it must be the name of an existing directory.

The VERBOSE switch causes OD to output additional information in the definition file. This includes the offsets of variables and record fields, and the sizes of types.

## 1.7 Running OD from the Workbench

OD cannot be run from the Workbench at present.

## 1.8 Running OD from the FPE utility

A tool button in the FPE window can be configured to run OD (see FPE.doc). In the button editor, set the Command field to the full path name of the OD program. Set the Arguments field to "Code/!M.Sym", or wherever else the module's symbol file may be found. If you wish the definition file to go somewhere else than the current directory, add the destination to the Arguments field. Specify a console window as the Console field. Put at least 10000 in the stack field.

For example:

```
Command="DH1:Oberon-A/OD"  
Arguments="Code/!M.Sym TO OBERON-A:Defs"  
Console="CON:0/11/540/189/OD'ing !M.../CLOSE/WAIT"  
Stack=10000
```

To create a definition file:

1. select the module in the Module gadget.
2. click on the tool button OD is bound to.
3. sit back and relax for a bit.

## 1.9 Contacting the author

OD was written by Frank Copeland.

All bug reports, suggestions and comments can be directed to:

Email : fjc@wossname.apana.org.au

Mailing list : oberon-a@wossname.apana.org.au

Snail Mail :

Frank J Copeland  
PO BOX 236  
RESERVOIR VIC 3073  
AUSTRALIA

Remember the J. It saves a lot of confusion at my end :-).

## 1.10 Reporting bugs and suggestions

You are encouraged to report any and all bugs you find, as well as any comments or suggestions for improvements you may have.

Before reporting a suspected bug, check the file ToDo.doc to see if it has already been noted. If it is a new insect, clearly describe its behaviour including the actions necessary to make it repeatable. Indicate in your report which version of OD you are using. Include an example of a definition file or symbol file that demonstrates the bug.

## 1.11 Who did what and why

THIS SPACE INTENTIONALLY LEFT BLANK

## 1.12 Release History

THIS SPACE INTENTIONALLY LEFT BLANK

---