

**inmos**

**occam2 toolset  
TDS  
support  
guide**

**INMOS Limited**

72 TDS 200 00

Copyright INMOS Limited 1989

This document may not be copied, in whole or in part, without prior written consent of INMOS.

 , **Inmos** , IMS and OCCAM are trademarks of the INMOS Group of Companies.

INMOS is a member of the SGS-THOMSON Microelectronics Group of Companies.

INMOS document number: 72 TDS 200 00

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>OCCAM language versions</b>	<b>1</b>
<b>3</b>	<b>Text file representation</b>	<b>1</b>
<b>4</b>	<b>Libraries</b>	<b>2</b>
	4.1 The library system	2
	4.2 The TDS libraries	3
<b>5</b>	<b>Other program changes that may be needed</b>	<b>4</b>
	5.1 Termination	4
	5.2 Screen control codes	5
<b>6</b>	<b>The TDS support tools disk</b>	<b>5</b>
	6.1 Flattener	5
	6.2 Lister	6
	6.3 Directive convertor	7
	6.4 The TDS import utilities	8
	6.5 The streamio conversion library	8
	Example program	12



## 1 Introduction

This document describes the main differences between the OCCAM 2 toolset and the TDS (IMS D700D). It then goes on to describe some tools designed to ease the task of porting code from the TDS to the toolset (and vice versa).

## 2 OCCAM language versions

The toolset checker `icheck` accepts the full OCCAM 2 language.

The toolset compiler `occam` accepts the full language except for in-line `VALOF` constructs.

The TDS product compiler in IMS D700D accepts the same language as the toolset compiler.

Earlier TDS compilers have a few minor incompatibilities (e.g string continuation characters, channels without protocols).

The configuration language accepted by the toolset configurer `iconf` allows OCCAM source to be used within `PROCESSOR` statements, except that libraries cannot be referenced.

## 3 Text file representation

Text files in the TDS are stored in a special format (OPS format or TDS format) that can only be read by tools which understand it. TDS format files must be converted to HOST text files before use by the toolset. Suitable tools include the `WRITE HOST` utility and *Selective lister* tool within the TDS, and the `iflat` and `list` tools supplied on the toolset TDS conversion disk. These provide alternative ways of converting TDS format text files into ordinary host text files suitable for your favourite editor.

The tools differ principally in their handling of fold crease lines, and especially any embedded filed folds. If any file may in the future be reimported into the TDS, it is important to ensure that both top and bottom creases are converted into language comment lines, which can be used (by a tool such as the TDS *Unlister*) to recreate the folded structure.

## 4 Libraries

### 4.1 The library system

In the IMS D700D libraries (both of constants and of procedures) are referenced by **#USE** lines in which the library file may be explicitly referenced by a file name in quotes, or indirectly referenced by a "library logical name" with the form of an OCCAM identifier. The library logical name system allows an arbitrary translation to be set up outside the source text itself.

In the toolset, libraries may be text libraries (usually constants and protocol definitions, but not restricted to these) or binary code libraries containing compiled procedures and functions. Text libraries are referenced by **#INCLUDE** directives, code libraries by **#USE** directives. Both directives include a single file name in quotes.

All toolset tools which need to look for library text or code use a common routine for converting the string in the directive to a file name. This process may include using a "search path" defining a sequence of directories to be searched if the string does not explicitly include a directory name. The search path system gives the user the ability to reference different versions of libraries by changing the search path, but does not provide the full freedom of indirection which the logical name system provides.

These different approaches to the location of library files mean that it will be necessary to make some changes to TDS program source to make its directives appropriate for the toolset. Some tools to aid in this activity are included in the toolset TDS conversion disk.

The tool **iflat** does some translation of **#USE** directives. The tool **idirect** can also be used to make systematic changes to **#USE** directives.

## 4.2 The TDS libraries

TDS programs to be converted may include library references to constants and procedures that are declared in standard toolset libraries shipped with the toolset software. They may also include library references to libraries which are principally designed to interface with the TDS run time system.

Input and output procedures in the IMS D700D TDS libraries `userio` and `interf` all use channels with protocol `CHAN OF INT` or `CHAN OF ANY`. The actual detailed protocols used on these channels depend on whether they are going to the keyboard, screen, or filing systems. The toolset versions of these libraries use OCCAM protocols to implement the key and screen stream protocol of the TDS. Programs using key stream and screen stream protocols can be easily converted for use by the toolset by means of interface procedures such as `so.keystream.from.keyboard`, `so.keystream.from.file`, `so.scrstream.to.stdout`, and `so.scrstream.to.file` in the toolset library `streamio.lib`. To minimise changes to an existing program a version of the library, `streamco.lib` is supplied with the TDS support tools. This library does not enforce the use of OCCAM protocols, thereby minimising changes to existing code. Routines in this library may be called in parallel with the rest of the user's code, leaving it otherwise unchanged except for the inclusion of TDS library procedures recompiled with the toolset compiler from the TDS library source in `\TDS2\IOLIBS\SRC`.

TDS programs which access the folded filing system cannot be trivially converted for use by the toolset as there is no fold management system in the toolset. It will therefore be necessary to redesign such programs to read and/or write ordinary host files using toolset library procedures.

The following table shows which TDS library references can be trivially converted for use by the toolset.

<b>TDS</b>	<b>Toolset</b>
<b>reinit</b>	<b>xlink.lib</b>
<b>blockcrc</b>	<b>crc.lib</b>
<b>userhdr</b>	<i>some in userhdr.inc</i>
<b>filerhdr</b>	<i>TDS folded filer interface - obsolete</i>
<b>krnlhdr</b>	<i>TDS interface - obsolete</i>
<b>uservals</b>	<i>some in userhdr.inc</i>
<b>afhdr</b>	<i>some in afhostio.inc</i>
<b>strings</b>	<b>string.lib</b>
<b>ioconv</b>	<b>convert.lib</b>
<b>extrio</b>	<b>convert.lib</b>
<b>slice</b>	<i>obsolete</i>
<b>ufiler</b>	<i>low level TDS fold handling</i>
<b>userio</b>	<b>streamio.lib</b>
<b>msdos</b>	<i>low level TDS access to host files</i>
<b>interf</b>	<i>recompile as needed from TDS source</i>
<b>afio</b>	<i>rewrite to use afhostio or recompile from TDS source</i>
<b>afiler</b>	<i>rewrite to use afhostio or recompile from TDS source</i>
<b>afinterf</b>	<i>rewrite to use hostio equivalents</i>
<b>t4board</b>	<i>UART driver in process.lib</i>
<b>t2board</b>	<i>UART driver in process.lib</i>
<b>mathhdr</b>	<i>not needed</i>
<b>mathvals</b>	<b>mathvals.inc</b>
<b>snglmath</b>	<b>snglmath.lib</b>
<b>dblmath</b>	<b>dblmath.lib</b>
<b>t4utils</b>	<b>tbmaths.lib</b>
<b>t4math</b>	<b>tbmaths.lib</b>

## **5 Other program changes that may be needed**

This section indicates other areas which may need attention when converting a program from the TDS to the toolset.

### **5.1 Termination**

A TDS **EXE** terminates leaving the user in the TDS editing environment from which the **EXE** was called. In the toolset it will probably be necessary to add a call to the library procedure **so.exit** to terminate the server.



## 5.2 Screen control codes

A TDS **EXE** using the full range of screen stream protocol for screen cursor control, etc, uses translations for these commands which are in the TDS server **SERVER14**. A toolset program will have to incorporate a procedure such as **scrstream.to.ANSI** to generate control sequences as sequences of ASCII characters in order to send to the standard output stream of the **ISERVER**.

## 6 The TDS support tools disk

This disk contains a number of tools and TDS utilities to help porting programs between the IMS D700D Transputer Development System and the IMS D705B OCCAM 2 toolset and vice versa.

This disk contains:

<b>readme</b>	Text file containing brief description of disk contents.
<b>iflat.exe</b>	TDS OCCAM to toolset flattener tool.
<b>list.exe</b>	Generic TDS to flat file format tool.
<b>clist.set</b>	C comment format file for <b>list</b> .
<b>occam.set</b>	OCCAM comment format file for <b>list</b> .
<b>null.set</b>	Null comment format file for <b>list</b> .
<b>lister.set</b>	Default set file for <b>list</b> (set to <b>null.set</b> ).
<b>idirect.b4h</b>	Directive convertor tool.
<b>idirect.exe</b>	Directive convertor driver program.
<b>import.top</b>	Top level TDS file with toolset import utilities.
<b>import.cut</b>	The TDS import utilities.
<b>streamco.lib</b>	Streamio conversion library.
<b>streamco.liu</b>	Streamio conversion library usage file.

### 6.1 Flattener

The flattener **iflat** runs from DOS on a TDS format text file containing an OCCAM program and converts it to one or more normal DOS format files, making any necessary conversions as required. The output files are given the filename suffix **.occ**.

Usage: **iflat** [*TDSfile*] [*directory*] {*options*}

The *directory* parameter may be used to direct all the output files to a different directory.

Options (each preceded by a /, and terminated by a space), are:

- I information
- R convert every occurrence of **CHAN** to **CHAN OF ANY**
- L disable **#USE** list output and library substitutions
- S reference SCs using **#SC** instead of **#USE**
- E output bottom crease lines in the form **--}}}**

Running **iflat** with no parameters will give brief help information.

Every line beginning **#USE** is converted to the form **#USE "name.lib"**.

Other conversions made by **iflat** include:

- filed folds are converted to individual files referenced by **#INCLUDE** directives.
- **#SC** is converted to **#USE**
- top fold creases are converted to **OCCAM** comments
- bottom fold creases are optionally converted to **OCCAM** comments

## 6.2 Lister

The tool **list** runs from DOS and does a similar job to **iflat** but does no source conversions, creates only one file and the comments produced for fold creases are determined by a control file. A number of control files are supplied: **occam.set** converts top crease comments to **OCCAM** format; **clist.set** converts top crease comments to **C** format; **null.set** removes all crease comments. The default name for the control file is **lister.set**, which is the same as **null.set**. If this control file is used you need not specify a control file on the command line. The driver file is read from the standard input (so you could type the driving instructions from the keyboard if you wished).

Usage: **list TDSfile outputfile < controlfile**

### 6.3 Directive convertor

The `idirect` tool runs from DOS on a transputer board with a T414 or T800 transputer. The `iserver` must be available. It changes the names of libraries referenced in flat (toolset) format files, as directed by a control file.

Usage: `idirect filename {options}`

Options (each preceded by a /), are:

F filename specify control file (*compulsory option*)  
O filename specify output file (default is to overwrite input file)  
I information

Running `idirect` with no parameters provides brief help information.

The control file may contain lines of the following formats:

**#USE string1 directive string2**

Replaces `#USE string1` by `directive string2`. `#USE string1` must be separated by spaces. `directive` is one of `#USE` or `#INCLUDE`, and `string2` is any text. For example:

```
#USE userhdr #INCLUDE "userhdr.inc"
```

This replaces the `#USE` line for a library with the logical name `userhdr` by the `#INCLUDE` directive for `userhdr.inc`.

**INSERT directive**

Adds given directive to top of all files.

**DELETE directive**

Removes any occurrences of the given directive from all files.

For the `INSERT` command the directive must be one of `#USE`, `#INCLUDE`, or `#COMMENT`. For the `DELETE` command the directive must be `#USE`.

If the same string is given as the substitute string for more than one library it will only be inserted once.

## 6.4 The TDS Import utilities

The TDS import utility set (**IMPORT.CUT**) is a set of utilities for running inside the TDS (IMS D700D). It provides two utilities:

- Import SC
- Export SC

The Import SC utility reads in a separately compiled unit in the toolset format and converts it to a TDS (sourceless) SC. Point the cursor at an empty fold and run this utility. The utility will display a parameter fold asking for the name of a file containing a compiled (and linked) toolset code SC.

The Export SC utility converts a TDS SC to a toolset format code file. Point cursor at a valid SC fold and run this utility.

The file **IMPORT.TOP** contains a brief description of the two utilities.

SCs to be imported or exported in this way should be self-contained and make no external library references. In the toolset such an SC may be made by linking the object code from any SC with the object code of any SCs and libraries it calls.

## 6.5 The streamio conversion library

This library **streamco.lib** provides a simplified interface to the **streamio** library. Thus TDS programs which use the **interf** and **userio** libraries can be used with the toolset with the minimum of change. The key stream protocol is a **CHAN OF INT** and the screen stream protocol is a **CHAN OF ANY**. The names of the routines match the TDS versions wherever possible. Where a routine only has a counterpart in the toolset the name is the same as the toolset version, but without the **ks.** or **ss.** prefix. The function of the routines is as described in the TDS and toolset user manuals. The routines are listed below. The directory containing this library should be on the path defined by **ISEARCH**.

```
PROC scrstream.to.array (CHAN OF ANY scrn,  
                        [ ]BYTE buffer)
```

```
PROC scrstream.from.array (VAL [ ]BYTE buffer,  
                           CHAN OF ANY scrn)
```

```
PROC scrstream.to.file (CHAN OF ANY scrn,  
                       CHAN OF SP fs, ts,  
                       VAL [ ]BYTE filename,  
                       INT result)
```

```
PROC scrstream.fan.out (CHAN OF ANY scrn, screen.out1,  
                        CHAN OF ANY screen.out2)  
  
PROC scrstream.sink (CHAN OF ANY echo)  
  
PROC scrstream.copy (CHAN OF ANY scrn, scrn.out)  
  
PROC keystream.from.file (CHAN OF SP fs, ts,  
                          CHAN OF INT kbd,  
                          VAL []BYTE filename,  
                          INT result)  
  
PROC keystream.sink (CHAN OF INT keys)  
  
PROC keystream.to.screen (CHAN OF INT keyboard,  
                          CHAN OF ANY screen)  
  
PROC scrstream.to.ANSI (CHAN OF ANY scrn,  
                       CHAN OF SP fs, ts)  
  
PROC scrstream.to.TVI920 (CHAN OF ANY scrn,  
                          CHAN OF SP fs, ts)  
  
PROC read.char (CHAN OF INT source, INT char)  
  
PROC read.text.line (CHAN OF INT source,  
                    INT len, []BYTE line, INT char)  
  
PROC write.char (CHAN OF ANY sink, VAL BYTE char)  
  
PROC write.len.string (CHAN OF ANY sink,  
                      VAL INT len, VAL []BYTE str)  
  
PROC write.full.string (CHAN OF ANY sink,  
                       VAL []BYTE str)  
  
PROC write.text.line (CHAN OF ANY sink,  
                    VAL []BYTE str)  
  
PROC newline (CHAN OF ANY sink)  
  
PROC write.endstream (CHAN OF ANY sink)  
  
PROC read.int (CHAN OF INT source, INT number, char)
```

PROC read.hex.int (CHAN OF INT source,  
INT number, char)

PROC read.int64 (CHAN OF INT source, INT64 number,  
INT char)

PROC read.hex.int64 (CHAN OF INT source, INT64 number,  
INT char)

PROC read.real32 (CHAN OF INT source, REAL32 number,  
INT char)

PROC read.real64 (CHAN OF INT source, REAL64 number,  
INT char)

PROC read.echo.char (CHAN OF INT source,  
CHAN OF ANY echo, INT char)

PROC read.echo.hex.int (CHAN OF INT source,  
CHAN OF ANY echo,  
INT number, char)

PROC read.echo.int (CHAN OF INT source,  
CHAN OF ANY echo,  
INT number, char)

PROC read.echo.text.line (CHAN OF INT source,  
CHAN OF ANY echo,  
INT len, [ ]BYTE line,  
INT char)

PROC read.echo.int64 (CHAN OF INT source,  
CHAN OF ANY echo,  
INT64 number, INT char)

PROC read.echo.hex.int64 (CHAN OF INT source,  
CHAN OF ANY echo,  
INT64 number, INT char)

PROC read.echo.real32 (CHAN OF INT source,  
CHAN OF ANY echo,  
REAL32 number, INT char)

PROC read.echo.real64 (CHAN OF INT source,  
CHAN OF ANY echo,  
REAL64 number, INT char)

PROC write.int (CHAN OF ANY sink, VAL INT number,  
field)

PROC write.hex.int (CHAN OF ANY sink, VAL INT number,  
field)

PROC write.int64 (CHAN OF ANY sink,  
VAL INT64 number, VAL INT field)

PROC write.hex.int64 (CHAN OF ANY sink,  
VAL INT64 number, VAL INT field)

PROC write.real32 (CHAN OF ANY sink,  
VAL REAL32 number,  
VAL INT Ip, Dp)

PROC write.real64 (CHAN OF ANY sink,  
VAL REAL64 number,  
VAL INT Ip, Dp)

PROC goto.xy (CHAN OF ANY sink, VAL INT x, y)

PROC clear.eol (CHAN OF ANY sink)

PROC clear.eos (CHAN OF ANY sink)

PROC beep (CHAN OF ANY sink)

PROC up (CHAN OF ANY sink)

PROC down (CHAN OF ANY sink)

PROC left (CHAN OF ANY sink)

PROC right (CHAN OF ANY sink)

PROC insert.char (CHAN OF ANY sink, VAL BYTE ch)

PROC delete.chr (CHAN OF ANY sink)

PROC delete.chl (CHAN OF ANY sink)

PROC ins.line (CHAN OF ANY sink)

PROC del.line (CHAN OF ANY sink)

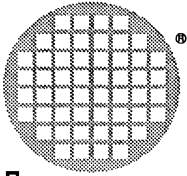
## Example program

```

#include "hostio.inc"
#include "streamio.inc"
PROC tdsprog (CHAN OF SP from.link, to.link,
             [ ]INT freespace)
  #USE "hostio.lib"
  #USE "streamio.lib"
  #USE "streamco.lib"
  SEQ
    [2]CHAN OF SP fs, ts:
    CHAN OF BOOL mstopit, kstop:
    BYTE kres, sres:
    PAR
      so.multiplexor (from.link, to.link, ts, fs, mstopit)
    SEQ
      CHAN OF ANY screen:
      CHAN OF ANY keyboard:
        -- NB must be CHAN OF ANY
    PAR
      so.keystream.from.kbd (fs[0], ts[0],
                           keyboard, kstop, 1500)
      so.scrstream.to.stdout (fs[1], ts[1], screen, sres)
    SEQ
      ... library refs for TDS program
      ... TDS screen/keyboard program - Unchanged
      kstop ! TRUE
      ks.keystream.sink (keyboard)
      write.endstream (screen)
      mstopit ! TRUE
    so.exit (from.link, to.link, sps.success)
  :

```





**inmos**<sup>®</sup>

**INMOS Limited**

1000 Aztec West  
Almondsbury  
Bristol BS12 4SQ  
U.K.  
Telephone (0454) 616616  
TLX 444723

**INMOS SARL**

Immeuble Monaco  
7 rue Le Corbusier  
SILIC 219  
94518 Rungis Cedex  
France  
Telephone (1) 46.87.22.01  
TLX 201222

**INMOS GmbH**

Danziger Strasse 2  
8057 Eching  
West Germany  
Telephone (089) 319 10 28  
TLX 522645

**INMOS Corporation**

P.O. Box 16000  
Colorado Springs  
Colorado 80935  
U.S.A.  
Telephone (719) 630 4000  
TLX (Easy Link) 62944936

**INMOS Japan K.K.**

4th Floor No 1 Kowa Bldg  
11-41 Akasaka 1-chome  
Minato-ku  
Tokyo 107  
Japan  
Telephone 03-505-2840  
TLX J29507 TEI JPN

---

●, **inmos**, **IMS** and **occam** are trademarks of the INMOS Group of Companies.