

# The xsldbg Handbook

Keith Isdale



# The xsldbg Handbook

# Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
<b>2</b>	<b>Using xsldb</b>	<b>9</b>
2.1	xsldb command arguments . . . . .	9
2.2	Legend of terms . . . . .	9
2.3	Overview of available commands . . . . .	10
<b>3</b>	<b>Command Reference</b>	<b>11</b>
3.1	Addparam . . . . .	11
3.2	Addwatch . . . . .	11
3.3	Base . . . . .	11
3.4	Break . . . . .	12
3.5	Bye . . . . .	12
3.6	Cat . . . . .	13
3.7	Cd . . . . .	13
3.8	Chdir . . . . .	13
3.9	Continue . . . . .	13
3.10	Data . . . . .	14
3.11	Delete . . . . .	14
3.12	Delparam . . . . .	14
3.13	Delwatch . . . . .	15
3.14	Dir . . . . .	15
3.15	Disable . . . . .	15
3.16	Du . . . . .	15
3.17	Dump . . . . .	16
3.18	Enable . . . . .	16
3.19	Entities . . . . .	16
3.20	Exit . . . . .	16
3.21	Frame . . . . .	17
3.22	Free . . . . .	17
3.23	Globals . . . . .	17
3.24	Help . . . . .	17
3.25	Load . . . . .	18
3.26	Locals . . . . .	18

## The xsldb Handbook

3.27 Ls . . . . .	18
3.28 Next . . . . .	18
3.29 Options . . . . .	19
3.30 Output . . . . .	19
3.31 Public . . . . .	19
3.32 Pwd . . . . .	19
3.33 Quit . . . . .	20
3.34 Readconfig . . . . .	20
3.35 Run . . . . .	20
3.36 Save . . . . .	20
3.37 Search . . . . .	20
3.38 Set . . . . .	21
3.39 Setoption . . . . .	21
3.40 Shell . . . . .	23
3.41 Showbreak . . . . .	23
3.42 Showparam . . . . .	24
3.43 Showwatch . . . . .	24
3.44 Source . . . . .	24
3.45 Step . . . . .	24
3.46 Stepdown . . . . .	25
3.47 Stepup . . . . .	25
3.48 Stylesheets . . . . .	25
3.49 System . . . . .	26
3.50 Templates . . . . .	26
3.51 Trace . . . . .	26
3.52 Tty . . . . .	26
3.53 Validate . . . . .	27
3.54 Walk . . . . .	27
3.55 Where . . . . .	27
3.56 Write . . . . .	28
3.57 Writeconfig . . . . .	28
<b>4 Credits and License</b>	<b>29</b>
4.1 Thanks to . . . . .	29
<b>A Installation</b>	<b>30</b>
A.1 How to obtain xsldb . . . . .	30
A.2 Requirements . . . . .	30
A.3 Compilation and Installation . . . . .	30
A.4 Configuration . . . . .	30

# List of Tables

3.1	Addparam usage	11
3.2	Addwatch usage	11
3.3	Base usage	11
3.5	Break usage	12
3.6	Bye usage	13
3.7	Cat usage	13
3.8	Cd usage	13
3.9	Chdir usage	13
3.10	Continue usage	14
3.11	Data usage	14
3.12	Delete usage	14
3.13	Delparam usage	14
3.14	Delwatch usage	15
3.15	Dir usage	15
3.16	Disable usage	15
3.17	Du usage	15
3.18	Dump usage	16
3.19	Enable usage	16
3.20	Entities usage	16
3.21	Exit usage	16
3.22	Frame usage	17
3.23	Free usage	17
3.24	Globals usage	17
3.25	Help usage	17
3.26	Load usage	18
3.27	Locals usage	18
3.28	Ls usage	18
3.29	Next usage	18
3.30	Options usage	19
3.31	Output usage	19

## The xsldb Handbook

3.32 Public usage . . . . .	19
3.33 Pwd usage . . . . .	19
3.34 Quit usage . . . . .	20
3.35 Readconfig usage . . . . .	20
3.36 Run usage . . . . .	20
3.37 Save usage . . . . .	20
3.38 Search usage . . . . .	21
3.39 Set usage . . . . .	21
3.40 Setoption usage . . . . .	23
3.41 Shell usage . . . . .	23
3.43 Showbreak usage . . . . .	24
3.44 Showparam usage . . . . .	24
3.45 Showwatch usage . . . . .	24
3.46 Source usage . . . . .	24
3.47 Step usage . . . . .	25
3.48 Stepdown usage . . . . .	25
3.49 Stepup usage . . . . .	25
3.50 Stylesheets usage . . . . .	25
3.51 System usage . . . . .	26
3.52 Templates usage . . . . .	26
3.53 Trace usage . . . . .	26
3.54 Tty usage . . . . .	27
3.55 Validate usage . . . . .	27
3.56 Walk usage . . . . .	27
3.57 Where usage . . . . .	27
3.58 Write usage . . . . .	28
3.59 Writeconfig usage . . . . .	28

### **Abstract**

xsldbg is a tool intended to help understand stylesheets. What makes it different to other stylesheet debuggers is the ability to search for items of interest and trace stylesheet execution.

# Chapter 1

## Introduction

xsldbg is a text based tool to debug stylesheets (the eXtensible Stylesheet Language) and has commands similar to the Unix/Linux debugger gdb. It has three major modes of execution of stylesheets.

Run the whole stylesheet
Step to next xsl instruction
Continue until next break point is found, or stylesheet has restarted



## Chapter 2

# Using xslDBG

### 2.1 xslDBG command arguments

On systems with readline library available you can use the back/forward keys to navigate the history of entered commands. On all systems the last entered command can be repeated by just pressing the <ENTER> key.

If your operating system supports it file names will be expanded.

Several commands take more than one argument. Quotes may be used to lead to complex expressions being treated as one argument. eg break "\*" | "@" would allow you to see a breakpoint on the template with the name "\*" | "@"

### 2.2 Legend of terms

The following table describes the terms used in the subsequent command guide

TEMPLATE_NAME	: A valid template name contains only ASCII character codes 0x00 to 0x7F. And can be a fully qualified name ie "xsl:templateName".
FILENAME	: A valid file name local to the system of the user. It can have a "~" prefix on *nix and CYGWIN platforms. Or environment variables under RISC OS
URI	: A Uniform Resource Identifiers as defined by RFC 2396
MODE_NAME	The mode of template which can be fully qualified name ie "xsl:modeName".
QNAME	: A fully qualified name ie "xsl:localPart"
LINENO	: A valid line number in associated <FILENAME>
NUMBER_OF_FRAMES	: A valid line number frames to change position by
BREAKPOINT_ID	: A valid break point number
WATCH_ID	: A valid watch expression number as indicated by showwatch command
SPEED	: speed to walk through code at, between 0 to 9
(Comment)	: a comment about command meaning or usage
{ opt1   opt2   opt2 .. etc }	: Choose one of the opt's
XPATH	: a xpath selection of node(s)
PARAM_ID	: a valid parameter number as indicated by showparam command
PARAM_NAME	: a valid parameter name as indicated by showparam command
PATH	: A path to change working directory to. On some operating systems a "~" prefix will be replaced by your home directory path

## The xsldbg Handbook

TEXT : Free form text ( <i>no restrictions</i> )
COMMAND : A valid command for the xsldbg
QNAME : A valid variable/parameter name
SOURCE : The stylesheet being/to be executed. See <FILENAME> and <URI>
DATA : The xml data(document) being/to be processed by the stylesheet. See <FILENAME> and <URI>
DEVICE_PATH : Is a valid terminal on the operating system
TTY_LEVEL : Is a valid level of input/output to use

### 2.3 Overview of available commands

Help related : <a href="#">help</a>
Running related : { <a href="#">bye</a>   <a href="#">exit</a>   <a href="#">quit</a> }, <a href="#">step</a> , <a href="#">stepup</a> , <a href="#">stepdown</a> , <a href="#">next</a> , <a href="#">continue</a> , <a href="#">run</a> , <a href="#">trace</a> , <a href="#">setoption</a> , <a href="#">options</a>
Libxslt parameter/option related : <a href="#">addparam</a> , <a href="#">delparam</a> , <a href="#">showparam</a> , <a href="#">output</a> , <a href="#">setoption</a> , <a href="#">optionsreadconfigwriteconfig</a>
Template related : <a href="#">templates</a> , <a href="#">where</a> , <a href="#">frame</a>
Break point related : <a href="#">break</a> , <a href="#">showbreak</a> , <a href="#">delete</a> , <a href="#">enable</a>
Expression viewing(xpath) : <a href="#">cat</a>
Node viewing : <a href="#">ls</a> , <a href="#">dir</a> , <a href="#">du</a> , <a href="#">cat</a> , <a href="#">pwd</a>
Variable viewing : <a href="#">globals</a> , <a href="#">locals</a> , <a href="#">cat</a> , <a href="#">addwatch</a>
Variable setting: <a href="#">set</a>
Node selection : <a href="#">source</a> , <a href="#">data</a> , <a href="#">cd</a>
Searching : <a href="#">search</a>
Operating system related : <a href="#">chdir</a> , <a href="#">shell</a> , <a href="#">tty</a>
File related : <a href="#">output</a> , <a href="#">entities</a> , <a href="#">system</a> , <a href="#">public</a>
Disabled file commands: <a href="#">validate</a> , <a href="#">load</a> , <a href="#">save</a> , <a href="#">write</a> , <a href="#">free</a>

## Chapter 3

# Command Reference

### 3.1 Addparam

Add a libxslt parameter; equivalent to providing `--param <QNAME>:<XPATH>` via command line.

<code>addparam &lt;QNAME&gt; &lt;XPATH&gt;</code> <i>(The &lt;XPATH&gt; must not contain any spaces nor double quotation marks.)</i>
<code>addparam &lt;QNAME&gt; "&lt;XPATH&gt;"</code> <i>(Must not contain double quotation marks in &lt;XPATH&gt;)</i>

Table 3.1: Addparam usage

### 3.2 Addwatch

Add an expression to be watched. See `showwatch` for display watch values

Shortcut name: `watch`

<code>addwatch &lt;XPATH&gt;</code>
-------------------------------------

Table 3.2: Addwatch usage

### 3.3 Base

Print the base for this node

<code>base</code>
-------------------

Table 3.3: Base usage

### 3.4 Break

Break at a template, at a location in a stylesheet or xml file loaded by xsldbg, or at the current node. New for xsldbg 3.1.4: When in gdb compatibility mode orphaned breakpoints can be set at a specific file and line number and be resolved to an active later on.

Shortcut name: b

xsldbg will try to guess the complete URL given a

file name without a path specified.
a file name in the same directory as the 'top' stylesheet loaded
a file name relative to the current working directory of xsldbg

ie if you have loaded a stylesheet file of `../en/xsldoc.xml` you can do this

**break -l xsldoc.xml 26**

This command will match a partial or complete QNAME template and or mode name provided. e.g. 'template' will match any QNAME with a local part of 'template'

Any name spaces in the provided QNAME will be expanded as specified by the names spaces defined in the XSL SOURCE file. e.g. 'xsl:test1' will be expanded to 'http://www.w3.org/199/XSL/Transform:test1'

A requested breakpoint may need to be resolved to its associated URL and line number. This is done automatically after the first template has been seen by xsldbg. Breakpoints are re-validated shortly after the start of each run.

Automatic breakpoint validation is used when gdb mode is enabled - the default behaviour of xsldbg

<code>break -l &lt;FILENAME&gt; &lt;LINENO&gt;</code> (To set breakpoint at specified file, line number)
<code>break -l &lt;URI&gt; &lt;LINENO&gt;</code> (To set breakpoint at specified URI, line number)
<code>break &lt;TEMPLATE_NAME&gt;</code> (To break at named or matched template.)
<code>break &lt;TEMPLATE_NAME&gt; &lt;MODE_NAME&gt;</code> (To break at named template with given mode.)>
<code>break "" &lt;MODE_NAME&gt;</code> (To break at any template that has a given mode name)
<code>break *</code> (To break at any template found.)
<code>break \*</code> (To break at the * template. Other name that include * will not be treated specially.)
<code>break</code> (To break point at current node. Yes that includes xml data nodes!)

Table 3.5: Break usage

### 3.5 Bye

Exit processing stylesheet as soon as possible.

bye
-----

Table 3.6: Bye usage

### 3.6 Cat

Print the result of a xpath expression on relative current node.

Usage: cat <XPATH> <i>(To view a variable or parameter)</i>
Usage: cat \$<QNAME>

Table 3.7: Cat usage

### 3.7 Cd

Change to the path specified by a xpath.

<< = preceding-sibling::node()
>> = following-sibling::node()
<- = ancestor::node()
-> = decendant::node()

Table 3.8: Cd usage

### 3.8 Chdir

Change the working directory

chdir <PATH> <i>(A relative or absolute path for operating system)</i>
--

Table 3.9: Chdir usage

### 3.9 Continue

Continue running stylesheet, stopping at any break points found.

Shortcut name: c

continue
----------

Table 3.10: Continue usage

### 3.10 Data

Switch to displaying the current node in xml data. Or change xml data used

data <i>(Switch to the current document node.)</i>
data <DATA> <i>(To change to a new xml data file. A leading ~ is replaced by the \$HOME environment variable value. You will need to use the run command to process it)</i>

Table 3.11: Data usage

### 3.11 Delete

Delete a template breakpoint

Shortcut name: d

delete <i>(To delete breakpoint at current node)</i>
delete <BREAKPOINT_ID> <i>(To delete breakpoint at specified break point number)</i>
delete -l <FILENAME> <LINENO> <i>(Delete at specified file, line number)</i>
delete -l <URI> <LINENO> <i>(Delete at specified URI, line number)</i>
delete <TEMPLATENAME> <i>(To delete break point at named template.)</i>
delete * <i>(To delete all break points.)</i>

Table 3.12: Delete usage

### 3.12 Delparam

Delete a libxslt parameter

delparam <i>(Delete all parameters present)</i>
delparam <PARAM_ID>
delparam <PARAM_NAME>

Table 3.13: Delparam usage

### 3.13 Delwatch

Delete a watch expression or remove all watch expressions as displayed by “showwatch.” command

delwatch <WATCHID> <i>(Delete a watch expression with given ID)</i>
delwatch * <i>(Delete all watch expressions)</i>

Table 3.14: Delwatch usage

### 3.14 Dir

Print list of nodes in a similarly way to the dir shell command.

dir
-----

Table 3.15: Dir usage

### 3.15 Disable

Disable a breakpoint

disable <i>(To disable breakpoint at current node)</i>
disable <BREAKPOINT_ID> <i>(To disable breakpoint at specified break point number)</i>
disable -l <FILENAME> <LINENO> <i>(Disable breakpoint at specified file, line number)</i>
disable -l <URI> <LINENO> <i>(Disable breakpoint at specified URI, line number)</i>

Table 3.16: Disable usage

### 3.16 Du

Print a summary of child nodes in a tree format.

du
----

Table 3.17: Du usage

### 3.17 Dump

Dump the details of this node

dump
------

Table 3.18: Dump usage

### 3.18 Enable

Enable or disable a breakpoint (Toggle enable/disable/)

Shortcut name: e

enable	<i>(To enable/disable breakpoint at current node)</i>
enable <BREAKPOINT_ID>	<i>(To enable/disable breakpoint at specified break point number)</i>
enable -l <FILENAME> <LINENO>	<i>(Enable/disable breakpoint at specified file, line number)</i>
enable -l <URI> <LINENO>	<i>(Enable/disable breakpoint at specified URL, line number)</i>

Table 3.19: Enable usage

### 3.19 Entities

Print list of external General Parsed entities used data file (document)

Shortcut name: ent

entities
----------

Table 3.20: Entities usage

### 3.20 Exit

Exit processing stylesheet as soon as possible.

exit
------

Table 3.21: Exit usage



### 3.21 Frame

Print the stack frame at a given depth

Shortcut name: f

frame <FRAME_DEPTH> <i>(Depth is a number from 0 to the current depth of call stack)</i>
--

Table 3.22: Frame usage

### 3.22 Free

Free stylesheet and data (Disabled see run)

free
------

Table 3.23: Free usage

### 3.23 Globals

Print a list of global stylesheet variables or parameters. Print the value of a global variable

globals <i>(Print list of all globally available variables)</i>
globals -f <i>(Print list of all globally available variables and their values)</i>
globals <QNAME> <i>(Print the value of variable specified)</i>

Table 3.24: Globals usage

### 3.24 Help

Display help on command or overview

Shortcut name: h

help <i>(Show overview of product)</i>
help <COMMAND> <i>(Show help about a command)</i>

Table 3.25: Help usage

### 3.25 Load

Load the xsldbg's options and user preferences from disk

load
------

Table 3.26: Load usage

### 3.26 Locals

Print a list of local stylesheet variables or parameters. Print the value of a local variable

locals <i>(Print list of all locally available variables)</i>
locals -f <i>(Print list of all locally available variables and their values)</i>
locals <QNAME> <i>(Print the value of variable specified)</i>

Table 3.27: Locals usage

### 3.27 Ls

List nodes in a brief format

ls
----

Table 3.28: Ls usage

### 3.28 Next

Skip over an xsl:call-template or xsl:apply-templates. This command has the same effect of entering the commands "step" and then "up"

Shortcut name: n

next <i>(proceed to next sibling instruction)</i>
---

Table 3.29: Next usage

### 3.29 Options

Print the values for xsldbg's option

options
---------

Table 3.30: Options usage

### 3.30 Output

Specify a local, writable file to be used for output of results

Shortcut name: o

<code>output &lt;FILENAME&gt;</code> <i>(A local writable file name. Which can have a ~ prefix on *nix and CYGWIN platforms. Or environment variables under RISC OS)</i>
<code>output &lt;URI&gt;</code> <i>(The &lt;URI&gt; must only use the "file://" protocol. This is then converted to a file name suitable for the operating system)</i>
<code>output -</code> <i>( Send to standard output. Must only be used when using xsldbg's command line prompt )</i>

Table 3.31: Output usage

### 3.31 Public

Print the value that a public ID maps via the current catalog

Shortcut name: pub

public "<PublicID>"
---------------------

Table 3.32: Public usage

### 3.32 Pwd

Print the current working directory.

pwd
-----

Table 3.33: Pwd usage

### 3.33 Quit

Exit processing stylesheet as soon as possible.

Shortcut name: q

quit
------

Table 3.34: Quit usage

### 3.34 Readconfig

Read xsldbg's configuration from settings stored on disk.

readconfig
------------

Table 3.35: Readconfig usage

### 3.35 Run

Restart the stylesheet.

Shortcut name: r

run
-----

Table 3.36: Run usage

### 3.36 Save

Save the xsldbg's options and user preferences to disk

save
------

Table 3.37: Save usage

### 3.37 Search

Search a dataBase of all information gathered from stylesheets loaded

All output files are stored in, value of the "searchresultspath" option if set, or the same directory as the provided stylesheet. searchresults.xml is normally transformed by search.xsl, but will be transformed using searchhtml.xsl if the "preferhtml" option is set.

When the search command is issued a xml file (searchresults.xml) will be created. You can then process this file with your own stylesheet to present data in other ways. If "preferhtml" option is not set then searchresult.txt is printed to display.

Depending on the amount of data collected it might take a while to complete this command.

search <XPATH> (See what xpath can be used see search.dtd. The default <XPATH> is '/ /search/*' )
search -sort <XPATH> (Tell search.xsl to sort the result before outputing it)

Table 3.38: Search usage

### 3.38 Set

Set the value of a variable

set <VARIABLE_NAME> <XPATH>
-----------------------------

Table 3.39: Set usage

### 3.39 Setoption

Set an option for execution of stylesheet

You will need to use run command to active changes

Example usage : setoption net 1
Example usage : setoption encoding "Some encoding value"
setoption <OPTION_NAME> (<INTEGER_VALUE>   <STRING_VALUE>)
Where <INTEGER_VALUE> is optional for options that can be treated as integer values
Where <OPTION_NAME> can be either

## The xsldbg Handbook

- `debug` (If `<INTEGER_VALUE>` is true dump the tree of the result instead)
- `catalogs` (If `<INTEGER_VALUE>` is true use the catalogs from `$SGML_CATALOG_FILES` or `SGML$CatalogFiles` for risc operating system)
- `html` (If `<INTEGER_VALUE>` is true the input document is an HTML file)
- `docbook` (If `<INTEGER_VALUE>` is true and docbook is still supported by libxml the input document is SGML docbook)
- `xinclude` (If `<INTEGER_VALUE>` is true do XInclude processing on document input)
- `preferhtml` (If `<INTEGER_VALUE>` is true prefer html output for search results. See search command)
- `autoencode` (If `<INTEGER_VALUE>` is true try to use the encoding from the stylesheet)
- `utf8input` (If `<INTEGER_VALUE>` is true all input from user is in UTF-8. This is normally used when xsldbg is running as a thread)
- `gdb` (Run in gdb compatability mode)
  - For a value of 1 this means
    - \* Print lots more messages. Increase the frequency of printing “Breakpoint at ...”
    - \* At most `GDB_LINES_TO_PRINT` lines will be printed when evaluating expressions, followed by a “...”. See options.h to change this value, the default is three lines of text
    - \* Both local and globals will be printed when the “locals” command is issued
    - \* When printing expressions with `cat/print`. The evaluated value will be prefixed by “= ” `<EXPRESSION>`
  - For a value of 2 this means
    - \* Print messages needed by KDbg as well as the output state above (when value is 1)
- `net` (If `<INTEGER_VALUE>` is true fetch DTDs or entities over network)
- `valid` (If `<INTEGER_VALUE>` is true perform the DTD loading phase)
- `repeat` (If `<INTEGER_VALUE>` is greater than zero(0) then run the transformation `<INTEGER_VALUE>` times)
- `profile` (If `<INTEGER_VALUE>` is true dump profiling information)
- `timing` (If `<INTEGER_VALUE>` is true display the time used)
- `out` (If `<INTEGER_VALUE>` is true dump the result)

Where value is true if the word “true” is supplied or the value is NOT equal to zero

Where value is false if the word “false” is supplied or value IS equal to zero

Where value is by default true, when no value is supplied, eg setoption net

Where value is by default false if the option name is preceded by “no”, e.g. setoption nonet

`stdout` Print all error messages to stdout. Normally error messages go to stderr.

`setoption <OPTION_NAME> “<STRING_VALUE>”` (Must not contain double quotation marks in `<STRING_VALUE>`)

`setoption <OPTION_NAME> <STRING_VALUE>` (Must not contain any spaces, nor double quotation marks in `<STRING_VALUE>`)

Where <OPTION_NAME> can be either
<ul style="list-style-type: none"> <li>• data (Data file's URI)</li> <li>• source (Source file's URI)</li> <li>• output (Output file's SystemID )</li> <li>• docspath (Path to use when looking for documentation)</li> <li>• catalognames (The names of the catalogs to use when the catalogs option is set. Value will be lost if set before setting catalogs option)</li> <li>• encoding (What encoding to use for standard output)</li> <li>• searchresultspath What path is to be used when storing the results of searching. If this is not set then xsldbg will use the path of the stylesheet</li> <li>• cwd (The directory changed into by the chdir command)</li> <li>• comment (A freeform comment about the debugger current session)</li> </ul>

Table 3.40: Setoption usage

### 3.40 Shell

Execute shell command

shell <TEXT> (<TEXT> is the text to be passed to operating system for execution)
--

Table 3.41: Shell usage

### 3.41 Showbreak

To display list of template break points.

Shortcut name: show

If a mode exists on a template breakpoint then it will be appended to the end of template name for breakpoint. An example of the output is:

Breakpoint 3 enabled for template :"" in file test1.xml : line 105
Breakpoint 2 enabled for template :"" testMode" in file test1.xml : line 109
Breakpoint 1 enabled for template :"" http://www.w3.org/1999/XSL/Transform:testMode" in file test1.xml : line 113
Total of 3 breakpoints present

showbreak
-----------

Table 3.43: Showbreak usage

### 3.42 Showparam

Print the libxslt parameters present

showparam
-----------

Table 3.44: Showparam usage

### 3.43 Showwatch

Show the current expression being watched

Shortcut name: watches

showwatch	<i>(Show the currently selected watches and their values)</i>
showwatch 1	<i>(Enable the automatic printing of watch expressions. This is used by default.)</i>
showwatch 0	<i>(Disable the automatic printing of watch expressions.)</i>

Table 3.45: Showwatch usage

### 3.44 Source

Switch to displaying the current node in stylesheet. Or change stylesheet used

source	<i>(Switch to the current node in stylesheet.)</i>
source <SOURCE>	<i>(To change to a new source file. A leading ~ is replaced by the \$HOME environment variable value. You will need to use the run command to execute it)</i>

Table 3.46: Source usage

### 3.45 Step

Step until next stylesheet instruction.

Shortcut name: s



step
------

Table 3.47: Step usage

### 3.46 Stepdown

Step down to a newer 'call frame'.

Shortcut name: down

stepdown <i>(step down one frame)</i>
stepdown <NUMBER_OF_FRAMES> <i>(step down specified number of frames)</i>

Table 3.48: Stepdown usage

### 3.47 Stepup

Step up to a older 'call frame'.

Shortcut name: up

This is not an accurate command, xsldbg will stop as close as it can.

stepup <i>(step up one frame)</i>
stepup <NUMBER_OF_FRAMES> <i>(step up specified number of frames)</i>

Table 3.49: Stepup usage

### 3.48 Stylesheets

Print out a list of stylesheets loaded

Shortcut name: style

stylesheets
-------------

Table 3.50: Stylesheets usage

### 3.49 System

Print the value that a system file maps via the current catalog

Shortcut name: sys

system "<SystemID>"
---------------------

Table 3.51: System usage

### 3.50 Templates

Print a list of available templates. Search for a template

Shortcut name: t

templates
templates <TEMPLATE> <i>(Print details of template named &lt;TEMPLATE&gt; if it can be found)</i>

Table 3.52: Templates usage

### 3.51 Trace

Trace one execution of the stylesheet printing the file and lines of intermediate steps

trace
-------

Table 3.53: Trace usage

### 3.52 Tty

Open a terminal. Set the level of tty redirection.

tty <DEVICE_PATH> <i>(Where &lt;DEVICE_PATH&gt; is a valid terminal on the operating system. Just tries to open the terminal)</i>
---

`tty <TTY_LEVEL>` (Set the level of tty redirection, where `<TTY_LEVEL>` is a valid level of input/output to use) Where level is

- 0 = Default input/output
- 1 = Terminal output of results of transformation, tracing and walking (Default state when tty device has been opened. Not fully implemented yet.)
- 2 = Full redirection to terminal (Not implemented yet.)
- All other integer values are assumed to mean level 0

Table 3.54: Tty usage

### 3.53 Validate

Validate the output file generated by stylesheet (Disabled)

`validate`

Table 3.55: Validate usage

### 3.54 Walk

Walk through code using a range of speeds

`walk <SPEED>` (Use `Ctrl-c` to stop execution, `<SPEED>` is a value between 0 and 9. Where 0 means stop, 1 is very fast, 9 is very slow)

Table 3.56: Walk usage

### 3.55 Where

Print a trace of templates calls (frame stack) and print the working directory.

Shortcut name: w

`where`

Table 3.57: Where usage

### 3.56 Write

To be completed

write
-------

Table 3.58: Write usage

### 3.57 Writeconfig

Write xsldb's configuration from settings stored on disk.

writeconfig
-------------

Table 3.59: Writeconfig usage

## Chapter 4

# Credits and License

xsldbg

Program copyright 2004 Keith Isdale [keith.kdwebdev.org](http://keith.kdwebdev.org)

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).

### 4.1 Thanks to

The writers the libxml and libxsl.

[Johannes Sixt](#) for helping with adding xsldbg support to KDbg

## Appendix A

# Installation

### A.1 How to obtain xsldbg

See the kxsldbg component of the extragear/sdk module in your nearest KDE SVN.

xsldbg is part of the KDE project <http://www.kde.org/>.

xsldbg can be found on the [download site](#) of the KDE project.

### A.2 Requirements

In order to successfully use xsldbg, you need KDE libxslt, libexslt and libxml installed which are available on a typical KDE installation.

### A.3 Compilation and Installation

xsldbg is usually compiled as part of the kxsldbg component in the extragear/sdk module

### A.4 Configuration

xsldbg is configured using arguments passed via the command line and its [setoption](#) command