X/Open Window Management Verification Suite

VSW User Guide
VSW Release 4.1.1

July 1992

The Open Group

Any comments relating to the material in this document may be sent by email to the Open Group at:

`ogtesting@opengroup.org`

1. Foreword

1.1 VSW Documentation

The X/Open Window Management Verification Suite, known as VSW, enables you to build and execute test programs which assess your system for conformance to the standards for X/Open Window Management established in the Common Application Environment Issue 4 (1991).

VSW consists of tests for the requirements in chapters 3 to 12 of the document
X/Open CAE Specification (1991)
X/Open Window Management: Xlib - C Language Binding

These tests have been derived from two sources as follows:

1. The MIT X test suite release 1.2. This has been produced by UniSoft under contract to MIT. This is a comprehensive Xlib and X Protocol test suite, including tests for chapters 2 to 10 of the MIT X11R4 Xlib specifications (reprinted as chapters 3 to 11 of the "X/Open CAE Specification (1991) - X/Open Window Management: Xlib - C Language Binding").

   These files contain MIT/UniSoft copyright notices.

2. The X/Open Window Management Additional Tests.

   These are tests for chapter 12 of the "X/Open CAE Specification (1991) - X/Open Window Management: Xlib - C Language Binding".

   These files contain X/Open copyright notices.

   They have been developed by UniSoft under contract to X/Open.

This document is designed to be used in conjunction with the documentation for the MIT X test suite release 1.2. There are a number of cross references to the corresponding chapters of the User Guide for the X test suite (release 1.2).

This document is in 2 parts. Part 1 is the VSW User Guide, which describes how to use the User Guide for the X test suite (release 1.2) to build and execute VSW. Part 2 gives additional reference information in a series of appendices, including details of VSW support services.

1.1.1 Part 1: VSW User Guide

The contents of the VSW User Guide are as follows:

1. The VSW User Guide contains cross references to the User Guide for the X test suite (release 1.2) to describe the steps to perform when using VSW, and the order in which to perform them.
2. The VSW User Guide describes differences between VSW and the MIT X test suite.
3. Where the MIT X test suite contains parameters which must be set to specific values when using VSW, the required settings are documented.
4. The VSW User Guide contains more detailed documentation required by an inexperienced test suite user, or a user completely unfamiliar with the X Window System.

1.1.2 Part 2: VSW Appendices

The appendices describe the support services for VSW, details on how to apply for waivers and address conformance issues.

X/Open Window Management Verification Suite

Part 1: User Guide
VSW Release 4.1.1

July 1992

X/Open Company Limited

2. VSW User Guide

2.1 Introduction

The steps involved in using VSW are described in this section. You should work through these steps in order.

The details of how to perform each step are almost the same as for the MIT X test suite. For this reason, there are a number of cross references to the User Guide for the X test suite (release 1.2). Where there are cross references, refer to the User Guide for the X test suite (release 1.2), and work through all the sections indicated. Then return to this VSW User Guide for the next step.

Note that VSW tests both the X Window System library and header files on the platform used to build VSW, as well as the X Window System display. Therefore, should you find it necessary to modify or reinstall the X Window System library or header files, you should return to section 2.5 of this VSW User Guide (Building the VSW libraries and utilities), and work through that stage and all the following stages.

2.2 Installing VSW

Before you install VSW you must first check there is file space available. When you are ready to load the VSW distribution, you must check the tape format, read the tape and check the contents loaded.

2.2.1 File space requirements

The target file system must have enough free space available to read the data on the VSW distribution tape. The total size of VSW after it has been read from tape is 12Mb.

The total size of VSW after it has been executed is dependent on the machine on which you are executing VSW. The disc usage is much greater on a RISC system than a CISC system. The approximate disc usage is as follows:

Build mode	**CISC system**	**RISC system**
`space-saving executables`	55Mb	88Mb
`normal executables`	165Mb	275Mb

You should build VSW in space-saving mode (refer to section 2.6 of this VSW User Guide).

2.2.2 Tape format

VSX is distributed either on 1600 BPI PE ½ inch magnetic tape, or on QIC II-compatible ¼ inch cartridge tape. The data will have been written using either the `cpio` or the `tar` utility. Details of the tape format, the utility used to write the data, its distribution number for X/Open and the command to use for extraction are written on the tape label.

Read the tape label to establish the tape format used, and thus the command required in order to extract the data.

2.2.3 Reading the tape

When you read the tape, the contents are installed in a hierarchy which starts from the current working directory. Work through the following steps:

1. Change to the directory in which you wish to install the distribution. You can install VSW to use a previously installed version of the TET (version 1.10 or newer). In this case, change to the **TET_ROOT** directory. For example, if you wish to use the version of TET previously installed from a VSX distribution (4.2.3 or newer), change to the directory in which the VSX installation tape was previously read. Refer to your VSX User and Installation Guide.

2. Set the environment variable TET_ROOT to the full path name of the directory in which you wish to install the distribution.

   NOTE - the TET_ROOT environment variable must be set for all the remaining stages when using VSW. For this reason, you are advised to set it in your `.profile`.

3. Read the tape in the **TET_ROOT** directory, to ensure that the locations for the files are correct. Read the distribution tape using the correct command:

   To extract all files except the TET (i.e. you already have the TET installed and built).

   For `tar` format tapes, use the appropriate command to extract all files under the directory `xtest` from a POSIX tar tape; e.g.:

   `tar xovf` *tape–device–name* `xtest`

   For `cpio` format tapes, use the appropriate command to extract all files under the directory `xtest` from a POSIX `cpio` tape with 5120 byte blocks; e.g.:

   `cpio -icdBv "xtest/*" <` *tape–device–name*

   To extract the entire distribution including the TET

   For `tar` format tapes, use the appropriate command to extract all files from a POSIX `tar` tape; e.g.:

   `tar xovf` *tape–device–name*

   For `cpio` format tapes, use the appropriate command to extract all files from a POSIX `cpio` tape with 5120 byte blocks; e.g.:

   `cpio -icdBv <` *tape–device–name*

2.2.4 Checking the contents

When you have finished reading the entire tape, you should check that the contents have been read correctly. Work through the following steps:

1. Check that the following main directories are in the current directory:

Directory Name	**Summary of contents**
`port`	A portability library used by TET
`tet_tmp_dir`	A temporary directory used by TET
`tet`	The Test Environment Toolkit (TET)
`xtest`	The tests comprising VSW

   If you did not extract the TET, only the directory `xtest` will be added to the contents of the current directory.

2. Check that the following file is in the `xtest` directory:

   `VSWrel4.1.1` This is the last file written to the tape. This file tells you the VSW release number and that all the contents of the tape have been read.

   If not, check there were no read errors while the tape was being read and that there is space available on the file system.

More details of the contents of the files in the MIT X test suite are given in appendix A of the User Guide for the X test suite (release 1.2).

2.3 Configuring VSW

Refer to the User Guide for the X test suite (release 1.2), and work through all the steps described in the section entitled "Configuring the X test suite".

In that section you will find a number of references to a later section of the User Guide for the X test suite (release 1.2) entitled "The portability library". This is an optional library supplied with the MIT X test suite which is not needed on X/Open CAE4 compliant systems. You should not use the portability library, and you should ignore that section of the User Guide for the X test suite (release 1.2). At this point, you can delete the directory `$TET_ROOT/port` (if you extracted the entire distribution from the installation tape).

Where configuration variables are described below, use the settings specified below, not those specified in the User Guide for the X test suite (release 1.2).

2.3.1 Configuration Parameters defined by the TET

TET_BUILD_TOOL
: If you wish to use the report writer `wrpt` to format build journals, you should set this parameter to `wbuild` in the file `tetbuild.cfg`.

  Refer to a later section of this VSW User Guide for more details of `wrpt`.

TET_CLEAN_TOOL
: If you wish to use the report writer `wrpt` to format clean journals, you should set this parameter to `wclean` in the file `tetclean.cfg`.

TET_OUTPUT_CAPTURE
: If you wish to use the report writer `wrpt` to format build or clean journals, you should set this parameter to `False` in the files `tetbuild.cfg` and `tetclean.cfg`.

2.3.2 Configuration for the TET

TETBASE
: VSW is supplied with a copy of TET (version 1.10). If you wish to use a previously installed version of the TET, set `TETBASE` to be the directory containing the TET. On a system with VSX4 installed, this is the **TET_ROOT** directory.

PORTLIB
: This configuration variable must be set to be empty.

```
PORTLIB=
```

PORTINC
: This configuration variable must be set to be empty.

```
PORTINC=
```

2.3.3 C Compiler Directives

DEFINES
: As VSW is to be run on CAE4 compliant platforms, _XOPEN_SOURCE must be added to this configuration variable in the `tetbuild.cfg` file.

XP_DEFINES
: As VSW is to be run on CAE4 compliant platforms, _XOPEN_SOURCE must be added to this configuration variable in the `tetbuild.cfg` file.

XP_OPEN_DIS
: This must be set so that the XOpenDisplay function in your X library is used to make connections to the X server when executing the X Protocol tests. This is done by using the following setting:

```
XP_OPEN_DIS=XlibOpaque.c
```

2.4 Building the TET

If you wish to use a previously installed version of the TET, ignore the section entitled "Building the TET" in the User Guide for the X test suite (release 1.2). Instead, ensure that your path includes the previously installed **tcc** utility, and the utilities supplied with VSW, by adding `$TET_ROOT/bin` and `$TET_ROOT/xtest/bin` to the search path for your shell, and include them in the PATH environment variable set in your `.profile`.

If you wish to use the version of the TET supplied with VSW, work through the section entitled "Building the TET" in the User Guide for the X test suite (release 1.2). Ignore the section entitled "The portability library". This should not be required on CAE4 compliant platforms.

2.5 Building the VSW libraries and utilities

Refer to the User Guide for the X test suite (release 1.2), and work through all the steps described in the section entitled "Building the X test suite libraries and utilities".

In addition you should build the following utilities which are exclusively for use by VSW.

2.5.1 Building the report writer `wrpt`

A report writer named `wrpt` has been developed specifically for use with VSW.

For more details, refer to the section of this VSW User Guide entitled "Report writer".

Build `wrpt` and install in the xtest bin directory as follows:

```
cd $TET_ROOT/xtest/xopen_src/bin/wrpt
pmake install
```

2.5.2 Building the TET build/clean tools

If you wish to use the report writer `wrpt` to format build or clean journals, you must use a TET build tool to produce the TET journal file in the required format.

Build and clean tools named `wbuild` and `wclean` are provided specifically for use with VSW.

These are shell scripts which invoke programs named `wbuild.exec` or `wclean.exec`, which in turn invoke the standard build and clean tools used by the MIT X test suite which are named `pmake` and `pclean`.

If you wish to use the report writer `wrpt` to format build or clean journals, build and install the tools as follows:

```
cd $TET_ROOT/xtest/xopen_src/bin/wmake
pmake install
```

You should also set the configuration parameters used by the TET as described in the section of this VSW User Guide entitled "Configuring VSW".

If you do not wish to use the report writer `wrpt` to format build or clean journals, you do not need to build the TET build/clean tools.

2.6 Building the tests

Refer to the User Guide for the X test suite (release 1.2), and work through all the steps described in the section entitled "Building the tests in space-saving format using the TET".

The X/Open Window Management Additional Tests, which are not part of the MIT X test suite, are found in the directory `$TET_ROOT/xtest/tset/XOPEN`. The files `xtest/link_scen` and `xtest/tet_scen` provided with VSW are replacements for the corresponding files in the MIT X test suite, and include the X/Open Window Management Additional Tests. This means that the **tcc** commands to build and execute the test suite will also build and execute the X/Open Window Management Additional Tests.

2.7 Executing VSW

Refer to the User Guide for the X test suite (release 1.2), and work through the following steps described in the section entitled "Executing the X test suite". These steps are as follows.

2.7.1 Setting up your X server

In the section entitled "Setting up your X server", please read the notes on "Formal verification testing".

2.7.2 Execute configuration parameters

Using the section entitled "Execute configuration parameters", set all the execution configuration parameters to reflect the configuration of the X server to be tested and the underlying operating system on which Xlib is implemented.

For each of the following configuration variables, use the settings specified below, not those specified in the User Guide for the X test suite (release 1.2).

XT_DISPLAY
: Note that when the variable XT_SCREEN_COUNT is greater than one, the complete test suite must be run separately for each screen of the display. You should do this by setting XT_DISPLAY to specify each screen of the display and running VSW to produce a separate report for each screen in turn. For each run, XT_ALT_SCREEN should specify a screen other than the one specified in XT_DISPLAY.

XT_POSIX_SYSTEM
: This should be set to `Yes`.

XT_DEBUG_OVERRIDE_REDIRECT
: This should be set to `No`.

XT_DEBUG_PAUSE_AFTER
: This should be set to `No`.

XT_DEBUG_PIXMAP_ONLY
: This should be set to `No`.

XT_DEBUG_WINDOW_ONLY
This should be set to `No`.

XT_DEBUG_DEFAULT_DEPTHS
This should be set to `No`.

XT_DEBUG_NO_PIXCHECK
This should be set to `No`.

XT_DEBUG_BYTE_SEX
This should be set to `NATIVE`.

XT_DEBUG_VISUAL_CHECK
This should be set to `0`.

You are also advised to use the following parameter setting to reduce the journal file size when executing VSW.

XT_OPTION_NO_CHECK
This may be set to `Yes`.

The following parameters are unique to VSW, and are used by `wrpt` when producing conformance reports. They must be set.

VSW_SYS
This must be set to a short description of the environment in which the test suite is being run. It should be complete enough to allow the results to be obtained independently.

VSW_OPER
This must be set to the test suite operator.

VSW_ORG
This must be set to the name of the organisation running the test suite.

2.7.3 Executing tests in space-saving format using the TET

Using the section entitled "Executing tests in space-saving format using the TET", execute the tests which you previously built in space-saving format.

Should you wish to execute modified scenarios, refer to the section entitled "Executing modified scenarios using the TET".

2.8 Report writer

2.8.1 Introduction

You can use the VSW reporting program to format reports from the results of the building and execution stages. You can generate reports from a complete journal file or from the results for the part you want to use. In addition, you can generate summary reports which summarise the results for testsets in a given section, area or testset. When you use the reporting program, you can use other options to control the length and width of the text on the page.

When you want to compare the results in several journal files, you can use the comparative reporting program, explained at the end of this chapter.

2.8.2 The reporting programs

The VSW reporting program, `wrpt`, formats the results in the VSW journal files generated by the building and execution stages. When you use `wrpt`, the environment variable **PATH** must be correctly set so that commands can be executed. The reporting program and its subsidiary programs are located in the directory `$TET_ROOT/xtest/bin`. This directory should be included in your **PATH**.

As with the VSX report writer `vrpt`, reports cannot be generated with `wrpt` from combined "build-execute" or "build-execute-clean" journals. You should complete the build stage for VSW and examine the output contained in the journal file before continuing with the execute stage.

2.8.3 Reporting program usage summary

`wrpt [-l`*level*`] [-r`*coverage*`] [-f`*file*`] [-v] [-H] [-p] [-P] [-L`*len*`] [-W`*wid*`] [-t`*lines*`] [`*file* `...]`

2.8.4 Report writer options

Reporting on the entire journal
To generate a report on an entire journal file, use the command:

`wrpt` *journal–file*

Reporting on a section or area (OPTIONAL)
There is only one section within VSW reported by `wrpt` which is called `tset`.

The names of the areas† within this section are as follows:

```
CH02  CH03  CH04  CH05  CH06
CH07  CH08  CH09  CH10  XPROTO
XOPEN
```

The first ten areas contain tests for chapters 2 to 10 of the MIT X11R4 Xlib specifications (reprinted as chapters 3 to 11 of the "X/Open CAE Specification (1991) - X/Open Window Management: Xlib - C Language Binding").

The `XOPEN` area contains the X/Open Window Management Additional Tests. These are tests for chapter 12 of the "X/Open CAE Specification (1991) - X/Open Window Management: Xlib - C Language Binding").

By default, the reporting program generates a report from the complete journal file. To produce a report from results for part of the test suite, use the `-r` option of `wrpt`, followed by the name of the section or area you want to use. For example, to report on the `tset/XOPEN` testset results in *journal–file*, use the command:

`wrpt -r tset/XOPEN` *journal–file*

To report on the `XCloseDisplay` (clsdsply) testset within `CH02` , use the command:

`wrpt -r tset/CH02/clsdsply` *journal–file*

Note that the Conformance Summary produced as part of the cover pages on validation test reports always contains the complete results for the journal file(s) being processed. Only the body of the report is affected by the `-r` option.

Reporting on individual testsets
You can also use the `-r` option for `wrpt` to report on a range of testsets. The `-P` option is useful here to stop the cover pages being produced. To report on the results from all the testsets between `XCloseDisplay` (clsdsply) and `XDefaultScreen` (dfltscrn) use the command‡:

---

† The areas within VSW are sometimes described as "sections" in the User Guide for the X test suite (release 1.2).

‡ The long line in this example has been folded at the \ character for formatting purposes. The command can be typed all on one line, in which case the \ character must be omitted.

```
wrpt -r tset/CH02/clsdsply:tset/CH02/dfltscrn \
      -P journal–file
```

Summary reports
To generate a report which summarizes the testset results by section or area, use the `-l` option. The area summary report, which is useful as a management summary, is given in tabular format. For example, to generate a summary report at section level, use the command:

```
wrpt -l sect journal–file
```

For area level reports, use the command:

```
wrpt -l area journal–file
```

Varying the report text format
You can use the `-L` page length and `-W` page width options to format the text in reports according to your paper size. When you reduce the page width, long output lines are automatically wrapped onto the next line of the report. Note that the Conformance Summary produced as part of the cover pages contains a wide table which does not get wrapped, so if you are using a page width of less than the default 80 characters, you will probably want to disable the cover pages by using the `-P` option. For example, to format the text using a page length of 50 lines and width of 64 characters, use the command:

```
wrpt -L50 -W64 -P journal–file
```

Additional report writer options
The `wrpt` user manual, in `xopen_man/wrpt.m`, gives full details of the additional options you can use with the `wrpt` reporting program.

2.8.5 Comparative reporting

You can use an alternative reporting program to compare the results in a number of different journal files. The reporting program `wrptm` enables you to compare the results from tests on a range of machines, or from a series of execution runs on the same machine with different software releases.

The comparative reporting program handles results from up to five journal files on the default page width of 80 columns (more on wider pages). The successes and failures are printed in tables, without any extra information about the reasons for tests failing. Use the standard reporting program `wrpt` to generate reports with the details of test failures.

Options
You can use the `-W` and `-L` options, for page width and page length, with `wrptm`.

Give the command `wrptm` with the options you want to use, on journal files generated from the results of running the `tcc`. Use the command:

```
bin/wrptm
```

from the `xtest` directory if your **PATH** does not include `bin`.

2.8.6 Sample report output

Conformance summary information

```
           X/OPEN Window Management Verification Suite Release 4.1.1
Test-Set Summary                                          Test-Set Summary


CONFORMANCE Summary

This is to certify that this system when tested for
conformance to the X/Open Window Management Specification
achieved the results below.

X/Open Window Management subset:
             |    TOTALS   Succeeded   Warnings Unresolved Unsupported NotInUse|
Section      | Expect Actual       Failed      FIP    Uninitiated  Untested    |
_____________|_____________________________________________________________|
             |                                                             |
tset         |  3927   3903   2902    88    0    0     5     0   101  542  265 |
TOTAL        |  3927   3903   2902    88    0    0     5     0   101  542  265 |
_____________|_____________________________________________________________|


Number of amendments _____________

                                                      ____________________
                                                      Signature/Date
```

```
Test Agency: UniSoft Ltd.                          System Tested: oursys
Test Date:   Jul 10, 1992                                  Page 4 of 474
```

Test results

```
           X/OPEN Window Management Verification Suite Release 4.1.1
Test-Set Summary                                              Test-Set Summary



Test-Set Name: /tset/CH02/allplns/allplns
-----------------------------------------

Test-Set Results:
----------------

Test-Set Started: 15:54:52

Test-Set Ended:   15:55:05

Test-Set Results Summary:
------------------------

   2  Tests Executed
   2  Tests Succeeded


Test-Set Name: /tset/CH02/blckpxl/blckpxl
-----------------------------------------

Test-Set Results:
----------------

Test-Set Started: 15:55:09

Test Results:

**************************************************************************
/tset/CH02/blckpxl/blckpxl 1Failed

Test Description:
A call to XBlackPixel returns the black pixel value for the
default colourmap of the screen screen_number.

Test Strategy:
Obtain the value of the black pixel using XBlackPixel.
Verify that the value is that given in parameter XT_BLACK_PIXEL.

Test Information:
XBlackPixel() returns incorrect value for black pixel
Expected value 0x0; Observed value 0x1
**************************************************************************

Test-Set Ended:   15:55:27

Test-Set Results Summary:
------------------------

   1  Tests Executed
   1  Tests Failed

Test Agency: UniSoft Ltd.                              System Tested: oursys
Test Date:   Jul 10, 1992                                     Page 6 of 474
```

Summary information

```
          X/OPEN Window Management Verification Suite Release 4.1.1
Test-Set Summary                                          Test-Set Summary


Section Name: tset
-----------------

Section Started: 15:54:52


Section Ended:   20:32:32

Section Results Summary:
-----------------------

11Areas Containing   616  Test-Sets Completed

3903  Tests Executed
2902  Tests Succeeded
  88  Tests Failed
   5  Tests Unresolved
 101  Tests Unsupported
 542  Tests Untested
 265  Tests Not In Use
```

```
          X/OPEN Window Management Verification Suite Release 4.1.1
Test-Set Summary                                          Test-Set Summary

Test Parameters:
TET_VERSION          =1.10
TET_OUTPUT_CAPTURE   =False
TET_RESCODES_FILE    =tet_code
TET_EXEC_IN_PLACE    =False
TET_SAVE_FILES       =Err*.err,*.sav
XT_DISPLAY           =rhea:0.0
XT_ALT_SCREEN        =UNSUPPORTED
XT_FONTPATH          =
/usr/lib/X11/fonts/xtest/,/usr/lib/X11/fonts/misc/
XT_SPEEDFACTOR       =5
XT_RESET_DELAY       =1
XT_VISUAL_CLASSES    =StaticGray(1,8) GrayScale(1,8) StaticColor(8)
PseudoColor(8) TrueColor(8) DirectColor(8)
XT_FONTCURSOR_GOOD   =0
XT_FONTCURSOR_BAD    =9999
XT_FONTPATH_GOOD     =
/usr/lib/X11/fonts/100dpi/,/usr/lib/X11/fonts/75dpi/
XT_FONTPATH_BAD      =/no-such-path-name
XT_BAD_FONT_NAME     =non-existent-font-name
XT_GOOD_COLORNAME    =red
XT_BAD_COLORNAME     =nosuchcolor
XT_DISPLAYMOTIONBUFFERSIZE =0
XT_SCREEN_COUNT      =1
XT_PIXMAP_DEPTHS     =1 8
XT_BLACK_PIXEL       =0
XT_WHITE_PIXEL       =1
XT_HEIGHT_MM         =224
XT_WIDTH_MM          =288
XT_PROTOCOL_VERSION  =11
XT_PROTOCOL_REVISION =0
XT_VENDOR_RELEASE    =1
XT_DOES_SAVE_UNDERS  =Yes
XT_DOES_BACKING_STORE =2
XT_POSIX_SYSTEM      =Yes
XT_DECNET            =No
XT_TCP               =Yes
XT_HOSTNAME          =rhea
XT_LOCAL             =Yes
VSW_SYS              =oursys
VSW_OPER             =S.J. Boutell
VSW_ORG              =UniSoft Ltd.
XT_SAVE_SERVER_IMAGE =No
XT_OPTION_NO_CHECK   =No
XT_OPTION_NO_TRACE   =No
XT_DEBUG             =0
XT_DEBUG_OVERRIDE_REDIRECT =No
XT_DEBUG_PAUSE_AFTER =No
XT_DEBUG_PIXMAP_ONLY =No
XT_DEBUG_WINDOW_ONLY =No
XT_DEBUG_DEFAULT_DEPTHS =No
XT_DEBUG_NO_PIXCHECK =No
XT_DEBUG_BYTE_SEX    =NATIVE

Test Agency: UniSoft Ltd.                        System Tested: oursys
Test Date:   Jul 10, 1992                              Page 473 of 474
```

```
          X/OPEN Window Management Verification Suite Release 4.1.1
Test-Set Summary                                          Test-Set Summary


XT_DEBUG_VISUAL_CHECK =0
XT_FONTDIR           =/tree/Xtest/vsw/xtest/fonts/


Report Command Line:wrpt journal
```

2.8.7 Troubleshooting

The report writer `wrpt` uses the `awk` utility to format reports (as with VSX).

Dependent on the reliability of `awk` on your system, you may encounter the following problem when you run `wrpt`.

1. `wrpt` gives the error message "received SIGPIPE".

   If the `wrpt` output was not being piped to another process, e.g. a pager, which exited before reading all the output, then this may be due to `awk` becoming terminating prematurely. You should try using the **-t** option to `wrpt` to truncate test failure information to a manageable number of lines. If `awk` still terminates prematurely, try replacing `awk` with `nawk` or `gawk` if these are available on your system.

CONTENTS