Trusted Solaris 2.5.1 Man Pages: 5TSOL Macros

Sun Microsystems Federal, Inc.
A Sun Microsystems, Inc. Business
901 San Antonio Road, MS USJC01-201
Palo Alto, CA 94303
U.S.A.

Preface

The Trusted Solaris operating environment is based on the SunOS operating system and other components of the Solaris operating environment and also bundles security-enhanced versions of the Common Desktop Environment (CDE), X windows, and Solstice AdminSuite tools. Therefore, the Trusted Solaris Reference Manual includes man pages not only for the operating system but also for the other products included in the Trusted Solaris product as well. In the Trusted Solaris Reference Manual, as in other UNIX reference manuals, each collection of information on a particular topic is called a man page, even though a man *page* may actually consist of *many pages* of text.

A man page is intended to answer concisely the question "What does it do"? Man pages are not intended to be tutorials. Depending what you are trying to do, refer to the Trusted Solaris user, developer, or administrator manuals for when and why to use a command or other features described in the man pages.

ACCESSING MAN PAGES

The man pages that make up the reference manual may be accessed in three ways.

Note: The following discussion of man page viewing options uses the term **package**, which is a unit of software typically delivered on a CD. Whoever installs a system usually decides whether or not all the packages are also installed. Installing the documentation packages is optional, because they are not required for operations. As a result, not everyone has access to every package. The packages that contain man pages in the Trusted Solaris operating environment are: SUNWman, plus SUNWaudmo , SUNWdtma , SUNWdtmad ,

SUNWkcsrt , SUNWkcspg , SUNWmfman , SUNWmfrun , SUNWolman , SUNWrtvcu , SUNWsadmm , SUNWtltkm , SUNWxwacx , SUNWxwman , SUNWxwplt , and SUNWxwpmn.

The first means of accessing the man pages is by using the **man**(1) command to view the man pages online. An account can use the **man** command when the man page package that contains the desired man page is available on the local system or mounted from a remote server, if, in addition, a terminal emulator (such as **dtterm**(1)) and the **man**(1) command are in one of the account's execution profiles. (For more about Trusted Solaris execution profiles and user accounts, see the Trusted Solaris user and administrator documentation.) To view a man page, enter the **man** command followed by the name of the man page. For example, to view the **ls**(1) man page that describes the command used to list directory's contents, a user enters the command: .

The second way to read man pages is by looking them up in the printed Trusted Solaris Reference Manual, which is in the Trusted Solaris documentation set, part number: TS2DS-251-9999.

The third means of reading the man pages is by viewing them in AnswerBook format. When the Trusted Solaris AnswerBook package, SUNWtab, is available on the local system or mounted from a remote server, anyone with the **answerbook**(1) command and a terminal emulator in an execution profile can display any of the man pages in the Trusted Solaris reference manual. The Trusted Solaris AnswerBook CD is packaged with the Trusted Solaris software CD. After the AnswerBook tool is launched, clicking the AnswerBook Navigator Search button brings up a dialog box where the name of a man page or terms contained in a man page can be entered to locate a specific man page.

For access to all available man pages for the operating system and for the bundled CDE, X windows, and Solstice AdminSuite products, the following man directories should be set in the MANPATH environment variable: **/usr/man**, **/usr/openwin/man**, and **/usr/dt/man**. For more about the format and contents of the man pages, see also the information in the **Intro** man pages for each section.

Trusted Solaris man pages are identified with a TSOL suffix in the section name. The TSOL suffix is used for man pages that are either added or modified from the base Solaris or bundled products.

- Section 1BTSOL describes printer commands adapted for Trusted Solaris from the Berkeley Software Distribution (BSD) print subsystem, which are used chiefly for printing administration.

  Note: Use of the equivalent System V print commands is recommended (such as **lp**(1TSOL)instead of **lpr**(1BTSOL)) because although the BSD

commands are included for compatibility, they will be removed in future releases. Also, the BSD print management commands are not useful for managing print jobs on remote printers.

• Section 1MTSOL describes Trusted Solaris system maintenance and administration commands.

• Section 1TSOL describes modified user commands from the base SunOS operating system, and new Trusted Solaris user commands.

• Section 2TSOL describes Trusted Solaris new or modified system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value.

• 3*TSOL subsections describe functions found in various Trusted Solaris libraries, other than those functions that directly invoke UNIX system primitives, which are described in Section 2TSOL.

Subsections include: 3CTSOL, 3NTSOL, 3RTSOL, 3TSOL, and 3X11TSOL.

• Section 4TSOL outlines the formats of various files. The C structure declarations for the file formats are given where applicable.

• Section 5TSOL contains documentation for Trusted Solaris macros.

• 7*TSOL subsections describe various special files that refer to specific hardware peripherals and device drivers.

Subsections include: 7DTSOL, 7MTSOL, and 7TSOL.

• 9*TSOL subsections provide reference information for writing device drivers in the kernel operating system environment.

Trusted Solaris subsections are: 9FTSOL and 9TSOL.

Following is a generic list of headings on each man page. The man pages of each manual section include only the headings they need. For example, if there are no bugs to report, there is no BUGS section. See the **Intro** pages for more information and detail about each section, and **man**(1) for more information about man pages in general.

NAME

This section gives the names of the commands or functions documented, followed by a brief description of what they do.

SYNOPSIS

This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full pathname is shown. Literal characters (commands and options) are in **bold** font and variables (arguments, parameters and substitution characters) are in *italic* font. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.

The following special characters are used in this section:

[ ] The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument *must* be specified.

... Ellipses. Several values may be provided for the previous argument, or the previous argument can be specified multiple times, for example, '*filename* ...'.

| Separator. Only one of the arguments separated by this character can be specified at time.

{ } Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.

PROTOCOL

This section occurs only in subsection 3RTSOL to indicate the protocol description file. The protocol specification pathname is always listed in **bold** font.

AVAILABILITY

This section briefly states any limitations on the availability of the command. These limitations could be hardware or software specific.

A specification of a class of hardware platform, such as **x86** or **SPARC**, denotes that the command or interface is applicable for the hardware platform specified.

In Section 1TSOL and Section 1MTSOL, **AVAILABILITY** indicates which package contains the command being described on the manual page. In order to use the command, the specified package must have been installed with the operating system. If the package was not installed, the security administrator can use **pkgadd**(1) or **swmtool**(1) to install the missing package.

MT-LEVEL

This section lists the **MT-LEVEL** of the library functions described in the Section 3 manual pages. The **MT-LEVEL** defines the libraries' ability to support threads. See **Intro**(3TSOL) for more information.

DESCRIPTION

This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, functions and such, are described under USAGE.

IOCTL

This section appears on pages in Section 7TSOL only. Only the device class which supplies appropriate parameters to the **ioctl**(2) system call is called **ioctl** and generates its own heading. **ioctl** calls for a specific device are listed alphabetically (on the man page for that specific device). **ioctl** calls are used for a particular class of devices all of which have an **io** ending, such as **mtio**(7).

OPTIONS

This lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option and where appropriate default values are supplied.

OPERANDS

This section lists the command operands and describes how they affect the actions of the command.

OUTPUT

This section describes the output - standard output, standard error, or output files - generated by the command.

RETURN VALUES

If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a

function can return only constant values, such as 0 or –1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared as **void** do not return values, so they are not discussed in RETURN VALUES.

ERRORS

On failure, most functions place an error code in the global variable **errno** indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.

USAGE

This section is provided as a *guidance* on use. This section lists special rules, features and commands that require in-depth explanations. The subsections listed below are used to explain built-in functionality:

Commands
Modifiers
Variables
Expressions
Input Grammar

EXAMPLES

This section provides examples of how to use a command or function. Wherever possible a complete example including command line entry and machine response is shown. When an example is given for a command entered by a normal user, the prompt is shown as

example%

If the user must be in an administrative role, the example uses either the profile shell prompt for the secadmin or admin roles:

$

or the root role prompt: **#**

Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS and USAGE sections.

ENVIRONMENT

This section lists any environment variables that the command or function affects, followed by a brief description of the effect.

EXIT STATUS

This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion and values other than zero for various error conditions.

FILES

This section lists all filenames referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.

SEE ALSO

This section lists references to other man pages, in-house documentation, and outside publications.

DIAGNOSTICS

This section lists diagnostic messages with a brief explanation of the condition causing the error. Messages appear in **bold** font with the exception of variables, which are in *italic* font.

WARNINGS

This section lists warnings about special conditions which could seriously affect your working conditions — this is not a list of diagnostics.

NOTES

This section lists additional information that does not belong anywhere else on the page. It takes the form of an *aside* to the user, covering points of special interest. Critical information is never covered here.

BUGS

This section describes known bugs and wherever possible suggests workarounds.

SUMMARY OF TRUSTED SOLARIS CHANGES

On base man pages that have Trusted Solaris modifications, this section summarizes the changes described thoughout the man page in a single easy-to-find place.

NAME

Intro, intro – introduction to miscellany

DESCRIPTION

Section 5TSOL contains man pages for Trusted Solaris macros.

NOTE

Trusted Solaris terms used on the man pages are defined in the **DEFINITIONS** section of the **Intro**(1TSOL) and **Intro**(2TSOL) man pages and explained further in the *Trusted Solaris User's Guide* , the *Trusted Solaris Developer's Guide* and the *Trusted Solaris Administrator's Procedures* manuals.

The **answerbook**(1) and printed versions of the *Trusted Solaris Reference Manual* includes only the Trusted Solaris man pages, while the online man pages that are viewable with the **man**(1) command include all the base man pages along with the Trusted Solaris man pages.

Name	**Description**
PRIV_ASSERT(5TSOL)	See **priv_macros**(5TSOL)
PRIV_CLEAR(5TSOL)	See **priv_macros**(5TSOL)
PRIV_EMPTY(5TSOL)	See **priv_macros**(5TSOL)
PRIV_EQUAL(5TSOL)	See **priv_macros**(5TSOL)
PRIV_FILL(5TSOL)	See **priv_macros**(5TSOL)
PRIV_INTERSECT(5TSOL)	See **priv_macros**(5TSOL)
PRIV_INVERSE(5TSOL)	See **priv_macros**(5TSOL)
PRIV_ISASSERT(5TSOL)	See **priv_macros**(5TSOL)
PRIV_ISEMPTY(5TSOL)	See **priv_macros**(5TSOL)
PRIV_ISFULL(5TSOL)	See **priv_macros**(5TSOL)
PRIV_ISSUBSET(5TSOL)	See **priv_macros**(5TSOL)
PRIV_TEST(5TSOL)	See **priv_macros**(5TSOL)
PRIV_UNION(5TSOL)	See **priv_macros**(5TSOL)
PRIV_XOR(5TSOL)	See **priv_macros**(5TSOL)
priv_macros(5TSOL)	test, assign, clear, or store a privilege or privilege set

NAME

priv_macros – Test, assign, clear, or store a privilege or privilege set

SYNOPSIS

```
#include <tsol/priv.h>

PRIV_ASSERT (priv_set, priv_id)
PRIV_ISASSERT (priv_set, priv_id)
PRIV_EQUAL (set_a, set_b)
PRIV_EMPTY (priv_set)
PRIV_FILL (priv_set)
PRIV_ISEMPTY (priv_set)
PRIV_ISFULL (priv_set)
PRIV_CLEAR (priv_set, priv_id)
PRIV_INTERSECT (set_a, set_b)
PRIV_INVERSE (priv_set)
PRIV_ISSUBSET (set_a, set_b)
PRIV_UNION (set_a, set_b)
PRIV_TEST (priv_id, errno)
PRIV_XOR (set_a, set_b)
        priv_set_t *priv_set, *set_a, *set_b;
        priv_t priv_id;
```

DESCRIPTION

PRIV_ASSERT (*priv_set , priv_id***)** asserts the *priv_id* privilege in the *priv_set*.

PRIV_ISASSERT (*priv_set , priv_id***)** is nonzero if the *priv_id* privilege in *priv_set* is asserted; if not, the value is zero.

PRIV_EQUAL (*set_a , set_b***)** is true if *set_a* and *set_b* are identical.

PRIV_EMPTY (*priv_set***)** initializes a *priv_set* to the null set.

PRIV_FILL (*priv_set***)** fills *priv_set*.

PRIV_ISEMPTY (*priv_set***)** is nonzero if *priv_set* is a null set; if not, the value is zero.

PRIV_ISFULL (*priv_set***)** is nonzero if *priv_set* is a full set; if not, the value is zero.

PRIV_CLEAR (*priv_set , priv_id***)** clears the *priv_id* in *priv_set*.

PRIV_INTERSECT (*set_a, set_b***)** stores the intersection of *set_a* and *set_b* in *set_b.*

PRIV_INVERSE (*priv_set***)** stores the inverse of *priv_set* in *priv_set*.

PRIV_ISSUBSET (*set_a, set_b***)** is nonzero if all privileges asserted in *set_a* are also asserted in *set_b* (that is, if *set_a* is a subset of *set_b*).

PRIV_UNION (*set_a, set_b***)** stores the union of *set_a* and *set_b* in *set_b.*

PRIV_TEST (*priv_id, errno***)** tests if *priv_id* is asserted in the effective set, and sets **errno** if not.

PRIV_XOR (*set_a, set_b***)** stores the **EXCLUSIVE OR** of *set_a* and *set_b* in *set_b.*

ERRORS

The behavior of these macros is undefined if *priv_id* is less than one or greater than the constant **MAX_PRIV**.

SEE ALSO

getppriv(2TSOL) **setppriv**(2TSOL)

Index

M
man(1), 5TSOL-7