# Oracle Solaris 11.4 Tunable Parameters Reference Manual

E61034-05
March 2024

Oracle Solaris 11.4 Tunable Parameters Reference Manual,

E61034-05



Primary Author: Cathleen Reiher

# Contents


## Using This Documentation

Product Documentation Library xiii
Feedback xiii

## 1 Overview of Oracle Solaris System Tuning

Tuning an Oracle Solaris System 1-1
- /etc/system.d/ Directory Files 1-2
- kmdb Utility 1-3
- mdb Command 1-3
- Tuning Format of Tunable Parameters Descriptions 1-4

Special Oracle Solaris tune and var Structures 1-5
Viewing Oracle Solaris System Configuration Information 1-6
- sysdef Command 1-6
- kstat Utility 1-6
- kstat2 Utility 1-6
- Oracle Solaris Observability Tools 1-7

## 2 Oracle Solaris Kernel Tunable Parameters

General Kernel and Memory Parameters 2-1
- default_stksize Parameter 2-2
- logevent_max_q_sz Parameter 2-3
- lwp_default_stksize Parameter 2-3
- noexec_user_stack Parameter 2-4
- physmem Parameter 2-5

fsflush and Related Parameters 2-6
- autoup Parameter 2-6
- doiflush Parameter 2-7
- dopageflush Parameter 2-8
- fsflush Parameter 2-8
- tune_t_fsflushr Parameter 2-9

Process-Sizing Parameters 2-10

max_nprocs Parameter 2-10
maxuprc Parameter 2-11
maxusers Parameter 2-11
ngroups_max Parameter 2-12
pidmax Parameter 2-13
reserved_procs Parameter 2-14
Paging-Related Parameters 2-14
desfree Parameter 2-15
fastscan Parameter 2-17
handspreadpages Parameter 2-17
lotsfree Parameter 2-18
maxpgio Parameter 2-19
min_percent_cpu Parameter 2-20
minfree Parameter 2-20
pageout_reserve Parameter 2-21
pages_before_pager Parameter 2-22
pages_pp_maximum Parameter 2-23
slowscan Parameter 2-23
throttlefree Parameter 2-24
tune_t_minarmem Parameter 2-25
Kernel Memory Allocator 2-25
kmem_flags Flag 2-26
kmem_stackinfo Variable 2-27
General Driver Parameters 2-28
SPARC: dax_stats_flags Flag 2-28
ddi_msix_alloc_limit Parameter 2-28
moddebug Parameter 2-29
Network Driver Parameters 2-30
IP Protocol Parameters in the Kernel 2-30
ip_squeue_worker_wait Parameter 2-30
ip_squeue_fanout Parameter 2-31
ipcl_conn_hash_size Parameter 2-31
igb Parameters 2-32
intr_force Parameter 2-32
mr_enable Parameter 2-33
ixgbe Parameters 2-33
intr_throttling Parameter 2-33
rx_copy_threshold Parameter 2-34
rx_limit_per_intr Parameter 2-34
rx_queue_number Parameter 2-35
rx_ring_size Parameter 2-35

tx_copy_threshold Parameter 2-36
tx_queue_number Parameter 2-36
tx_ring_size Parameter 2-37
General I/O Parameters 2-37
maxphys Parameter 2-37
rlim_fd_cur Parameter 2-38
rlim_fd_max Parameter 2-39
rlim_fd_sys Parameter 2-40
General File System Parameters 2-40
dnlc_dir_enable Parameter 2-40
dnlc_dir_max_size Parameter 2-41
dnlc_dir_min_size Parameter 2-42
dnlc_dircache_percent Parameter 2-42
ncsize Parameter 2-43
TMPFS Parameters 2-44
tmpfs:tmpfs_maxkmem Parameter 2-44
tmpfs:tmpfs_minfree Parameter 2-44
Pseudo Terminals 2-45
pt_cnt Parameter 2-46
pt_max_pty Parameter 2-47
pt_pctofmem Parameter 2-47
STREAMS Parameters 2-48
nstrpush Parameter 2-48
strmsgsz Parameter 2-48
strctlsz Parameter 2-49
System V Message Queues 2-49
System V Semaphores 2-50
Timer Behavior 2-50
hires_tick Parameter 2-50
timer_max Parameter 2-51
SPARC: Platform Specific Parameters 2-51
default_tsb_size Parameter 2-51
enable_tsb_rss_sizing Parameter 2-52
tsb_alloc_hiwater_factor Parameter 2-52
tsb_rss_factor Parameter 2-53
Locality Group Parameters 2-54
lgrp_mem_pset_aware Parameter 2-54
lpg_alloc_prefer Parameter 2-55

## 3 Oracle Solaris ZFS Tunable Parameters


Tuning ZFS Considerations 3-1
ZFS Memory Management Parameters 3-1
- user_reserve_hint_pct ZFS Parameter 3-2
- zfs_arc_min Parameter 3-3
- zfs_arc_max Parameter 3-3
- zfs_arc_max_percent Parameter 3-4

ZFS File-Level Prefetch 3-4
- zfs_prefetch_disable Parameter 3-5

ZFS Device I/O Queue Depth 3-6
- zfs_vdev_max_pending Parameter 3-6

Tuning ZFS When Using Flash Storage 3-7
- Adding Flash Devices as ZFS Log or Cache Devices 3-7
- Ensuring Proper Cache Flush Behavior for Flash and NVRAM Storage Devices 3-8

Tuning ZFS for Database Products 3-10


## 4 NFS Tunable Parameters


Tuning the NFS Environment 4-1
NFS Module Parameters 4-1
- nfs:nfs_allow_preepoch_time Parameter 4-1
- nfs:nfs_async_clusters Parameter 4-2
- nfs:nfs3_async_clusters Parameter 4-3
- nfs:nfs4_async_clusters Parameter 4-4
- nfs:nfs_async_timeout Parameter 4-5
- nfs:nfs3_bsize Parameter 4-6
- nfs:nfs4_bsize Parameter 4-6
- nfs:nfs_cots_timeo Parameter 4-7
- nfs:nfs3_cots_timeo Parameter 4-8
- nfs:nfs4_cots_timeo Parameter 4-8
- nfs:nfs_disable_rddir_cache Parameter 4-9
- nfs:nfs_do_symlink_cache Parameter 4-10
- nfs:nfs3_do_symlink_cache Parameter 4-11
- nfs:nfs4_do_symlink_cache Parameter 4-11
- nfs:nfs_dynamic Parameter 4-12
- nfs:nfs3_dynamic Parameter 4-12
- nfs:nfs3_jukebox_delay Parameter 4-13
- nfs:nfs_lookup_neg_cache Parameter 4-14
- nfs:nfs3_lookup_neg_cache Parameter 4-14
- nfs:nfs4_lookup_neg_cache Parameter 4-15
- nfs:nfs_max_threads Parameter 4-16

nfs:nfs3_max_threads Parameter 4-17
nfs:nfs4_max_threads Parameter 4-17
nfs:nfs3_max_transfer_size Parameter 4-18
nfs:nfs4_max_transfer_size Parameter 4-19
nfs:nfs3_max_transfer_size_clts Parameter 4-20
nfs:nfs3_max_transfer_size_cots Parameter 4-20
nfs:nacache Parameter 4-21
nfs:nfs_nra Parameter 4-22
nfs:nfs3_nra Parameter 4-23
nfs:nfs4_nra Parameter 4-23
nfs:nrnode Parameter 4-24
nfs:nfs3_pathconf_disable_cache Parameter 4-25
nfs:nfs_shrinkreaddir Parameter 4-26
nfs:nfs3_shrinkreaddir Parameter 4-26
nfs:nfs_write_error_interval Parameter 4-27
nfs:nfs_write_error_to_cons_only Parameter 4-28
NFS-Related SMF Configuration Parameters 4-28
server_authz_cache_refresh Parameter 4-29
netgroup_refresh Parameter 4-29
nfssrv Module Parameters 4-29
nfssrv:rfs_write_async Parameter 4-29
rpcmod Module Parameters 4-30
rpcmod:clnt_max_conns Parameter 4-30
rpcmod:clnt_idle_timeout Parameter 4-31
rpcmod:svc_idle_timeout Parameter 4-31

## 5 Internet Protocol Suite Tunable Parameters

Overview of Tuning IP Suite Parameters 5-1
IP Suite Parameter Validation 5-1
Internet Request for Comments 5-1
IP Tunable Parameters 5-2
_addrs_per_if Parameter 5-2
_forwarding_src_routed Parameter (IPv4 or IPv6) 5-2
_icmp_err_interval and _icmp_err_burst Parameters 5-3
_policy_mask Parameter 5-3
_respond_to_echo_broadcast (IP) and _respond_to_echo_multicast Parameters (IPv4 or IPv6) 5-4
hoplimit Parameter (IPv6) 5-4
hostmodel Parameter (IPv4 or IPv6) 5-5
recv-multicast-scaling Parameter 5-5
send-redirects Parameter (IPv4 or IPv6) 5-6

ttl Parameter (IPv4) 5-6
IP Tunable Parameters Related to Duplicate Address Detection 5-7
_arp_defend_interval/_ndp_defend_interval Parameter 5-7
_arp_defend_period/_ndp_defend_period Parameter 5-7
_arp_defend_rate/_ndp_defend_rate Parameter 5-8
_arp_fastprobe_count Parameter 5-8
_arp_fastprobe_interval Parameter 5-9
_arp_probe_count Parameter 5-9
_arp_probe_interval Parameter 5-10
_defend_interval Parameter 5-10
_dup_recovery Parameter 5-10
_max_defend Parameter 5-11
_max_temp_defend Parameter 5-11
arp-publish-count/ndp-unsolicit-count Parameter 5-12
arp-publish-interval/ndp-unsolicit-interval Parameter 5-12
IP Tunable Parameters With Additional Cautions 5-12
_icmp_return_data_bytes Parameter (IPv4 or IPv6) 5-13
_pathmtu_interval Parameter 5-13
TCP Tunable Parameters 5-13
_conn_req_max_q Parameter 5-14
_conn_req_max_q0 Parameter 5-14
_conn_req_min Parameter 5-15
_deferred_ack_interval Parameter 5-15
_deferred_acks_max Parameter 5-16
_ipv4_ttl Parameter 5-16
_ipv6_hoplimit Parameter 5-17
_local_dack_interval Parameter 5-17
_local_dacks_max Parameter 5-17
_local_slow_start_initial Parameter 5-18
_rev_src_routes Parameter 5-18
_rst_sent_rate Parameter 5-19
_rst_sent_rate_enabled Parameter 5-19
_slow_start_after_idle Parameter 5-20
_slow_start_initial Parameter 5-20
_time_wait_interval Parameter 5-20
_tstamp_always Parameter 5-21
_wscale_always Parameter 5-21
cwnd-max Parameter 5-22
ecn Parameter 5-22
largest-anon-port Parameter 5-23
max-buf Parameter 5-24

recv-buf Parameter 5-24
sack Parameter 5-24
send-buf Parameter 5-25
smallest-anon-port Parameter 5-25
tcp_cwnd_normal Parameter 5-26
TCP Parameters With Additional Cautions 5-26
_ip_abort_interval Parameter 5-27
_keepalive_interval Parameter 5-27
_recv_hiwat_minmss Parameter 5-28
_rexmit_interval_extra Parameter 5-28
_rexmit_interval_initial Parameter 5-29
_rexmit_interval_max Parameter 5-29
_rexmit_interval_min Parameter 5-30
_tstamp_if_wscale Parameter 5-30
UDP Tunable Parameters 5-30
_ipv4_ttl Parameter 5-31
_ipv6_hoplimit Parameter 5-31
largest-anon-port Parameter 5-31
max-buf Parameter 5-32
recv-buf Parameter 5-32
send-buf Parameter 5-33
smallest-anon-port Parameter 5-33
SCTP Tunable Parameters 5-34
_addip_enabled Parameter 5-34
_cookie_life Parameter 5-34
_deferred_ack_interval Parameter 5-34
_heartbeat_interval Parameter 5-35
_ignore_path_mtu Parameter 5-35
_initial_mtu Parameter 5-36
_initial_out_streams Parameter 5-36
_initial_ssthresh Parameter 5-36
_ipv4_ttl Parameter 5-37
_ipv6_hoplimit Parameter 5-37
_maxburst Parameter 5-38
_max_in_streams Parameter 5-38
_max_init_retr Parameter 5-38
_new_secret_interval Parameter 5-39
_pa_max_retr Parameter 5-39
_pp_max_retr Parameter 5-40
_prsctp_enabled Parameter 5-40
_rto_initial Parameter 5-40

_rto_max Parameter 5-41
_rto_min Parameter 5-41
_shutack_wait_bound Parameter 5-42
_xmit_lowat Parameter 5-42
cwnd-max Parameter 5-42
largest_anon_port Parameter 5-43
max-buf Parameter 5-43
recv-buf Parameter 5-44
send-buf Parameter 5-44
smallest-anon-port Parameter 5-44
ICMP Tunable Parameters 5-45
_ipv4_ttl Parameter 5-45
_ipv6_hoplimit Parameter 5-45
Per-Route Metrics 5-46

## 6 System Facility Parameters

System Default Parameters 6-1
autofs Property 6-1
cron Facility 6-1
devfsadm File 6-1
fs File 6-2
ftp Facility 6-2
inetinit Facility 6-2
init Service 6-2
ipsec Facility 6-2
kbd Configuration Properties 6-3
keyserv Facility 6-3
login Facility 6-3
mpathd Facility 6-3
nfs Properties 6-3
nfslogd Log File 6-4
nss Facility 6-4
passwd Facility 6-4
su Facility 6-4
syslog Facility 6-4
tar Facility 6-4
telnetd Facility (Deprecated) 6-5
utmpd Daemon 6-5

A System Check Script

Confirming Flush Behavior on the System A-1

Index

# Using This Documentation

- **Overview** – Provides information about Oracle Solaris tunable parameters and how to configure these if the default values are insufficient for a specific customer setup.
- **Audience** – System administrators who might need to change kernel tunable parameters in certain situations.
- **Required knowledge** – Oracle Solaris or UNIX system administration experience and general file system administration experience.

## Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E37838-01.

## Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

1

# Overview of Oracle Solaris System Tuning

This section provides overview information about the format of the tuning information in this manual. This section also describes the different ways to tune an Oracle Solaris system.

- Tuning an Oracle Solaris System
- Special Oracle Solaris tune and var Structures
- Viewing Oracle Solaris System Configuration Information

## Tuning an Oracle Solaris System

As an operating system, Oracle Solaris adjusts easily to system load and thus requires minimal tuning. However, in certain cases, tuning might be necessary. This book provides details about the officially supported tuning options available for Oracle Solaris.

The Oracle Solaris kernel consists of a core portion, which is always loaded, and a number of loadable modules that are loaded as these modules are being referenced. Many kernel parameters listed in this manual are core parameters. However, a few parameters belong to loadable modules.

> **Note:**
>
> Tuning system parameters is the least effective method to use to improve performance. Rather, improving and tuning the application, as well as adding more physical memory and balancing disk I/O patterns, are better options.

The tunable parameters described in this book can change from one Oracle Solaris release to the next. Publication of these tunable parameters does not preclude changes to the tunable parameters and their descriptions without notice.

The following table describes the different ways tunable parameters can be applied.

| Apply Tunable Parameters in These Ways | For More Information |
|---|---|
| Set the parameter in a configuration file in the `/etc/system.d` directory. | /etc/system.d/ Directory Files |
| Use the kernel debugger (`kmdb`). | kmdb Utility |
| Use the modular debugger (`mdb`). | mdb Command |
| Use the `ipadm` command to set TCP/IP parameters. | Internet Protocol Suite Tunable Parameters |
| Modify the `/etc/default` files. | System Facility Parameters |

# /etc/system.d/ Directory Files

The /etc/system file provides a static mechanism for adjusting the values of kernel parameters. Values specified in this file are read at boot time and then applied. No changes specified in the file take effect unless the system is rebooted.

However, to tune parameters, do **NOT** edit the /etc/system file. Instead, use files in the /etc/system.d directory. This method enables you to tune system parameters without directly manipulating the /etc/system file. The method consists of the following steps:

1. Create an empty file in the /etc/system.d directory.
2. Provide the file with a company specific name and indications of its contents.
3. In the file itself, add the customized setting for the tunable you are configuring.

   As a first step, add only those tunables that are required by in-house or third-party applications. After baseline testing has been established, evaluate system performance to determine if additional tunable settings are required.

Files in the /etc/system.d directory are read at boot time and their contents are added to the /etc/system file. At the end of the boot process, the new configurations are applied to the system.

One pass is made to set all the values before the configuration parameters are calculated.

**Example 1-1 Setting a ZFS Parameter for a Specific System**

The following entry sets the ZFS ARC maximum (zfs_arc_max) to 30 GB.

```
set zfs:zfs_arc_max = 0x780000000
```

Suppose that the name of your company is Widget, Inc. You would store this entry in the /etc/system.d/widget:zfs file. To apply the new zfs_arc_max setting, reboot the machine.

You can recover from an incorrect value by using one of the following approaches:

- Resetting the Parameter in the /etc/system.d/*file*

  Remove the defective parameter setting from your configuration file in the /etc/system.d directory. At boot time, the /etc/system file is updated with the previous configurations which are then reapplied to the system.

- Using a Cloned Boot Environment

  Before you introduce system parameter changes, clone the boot environment first.

  ```
  # beadm create BE-clonename
  ```

  Then, if your current BE becomes unusable after applying changes to /etc/system, reboot the system. From the x86 GRUB menu or SPARC boot menu, select the BE clone. After booting completes, you can optionally activate the BE clone to become the default BE to be used in subsequent system boots.

- Using File Copies

  Make a copy of the `/etc/system` file before updating it with new parameters from configuration files in the `/etc/system.d` directory so that you can easily recover from incorrect value. For example:

  ```
  # cp
                /etc/system /etc/system.good
  ```

  If a value specified in the configuration file in `/etc/system.d` causes the system to become unbootable, you can recover with the following command:

  ```
  ok boot -a
  ```

  This command causes the system to ask for the name of various files used in the boot process. Press the Return key to accept the default values until the name of the `/etc/system` file is requested. When the `Name of system file [/etc/system]:` prompt appears, type the name of the good `/etc/system` file or `/dev/null`:

  ```
  Name of system file [/etc/system]: /etc/system.good
  ```

  If `/dev/null` is specified, this path causes the system to attempt to read from `/dev/null` for its configuration information. Because this file is empty, the system uses the default values. After the system is booted, the `/etc/system` file can be corrected.

  For more information about system recovery, see *Troubleshooting System Administration Issues in Oracle Solaris 11.4*.

## `kmdb` Utility

The `kmdb` utility is an interactive kernel debugger with the same general syntax as the modular debugger (`mdb`). With this debugger, you can set breakpoints. When a breakpoint is reached, you can examine data or step through the execution of kernel code.

> **Caution:**
>
> Because the `kmdb` utility stops the kernel that is currently running, only use it if you are an expert user such as a kernel developer. Do not use the `kmdb` utility on production systems.

`kmdb` can be loaded and unloaded on demand. You do not have to reboot the system to perform interactive kernel debugging, as was the case with the `kadb` utility.

For more information, see the `kmdb`(1) man page.

## `mdb` Command

The modular debugger, `mdb`, is the recommended post-mortem debugger for the kernel. This debugger also includes a number of desirable usability features such as command-line editing, command history, built-in output pager, syntax checking, and command pipelining. A

programming API is available that enables a compilation of modules to perform desired tasks within the context of the debugger.

For more information, see the *Oracle Solaris Modular Debugger Guide* and the `mdb`(1) man page.

**Example 1-2 Using `mdb` to Display Information**

Display a high-level view of a system's memory usage.

Note that the following command uses the `--kernel` option in place of the obsolescent `-k` option:

```
# mdb --kernel --unsafe-io-access
Loading modules: [ unix genunix specfs dtrace mac cpu.generic
cpu_ms.AuthenticAMD.15 uppc pcplusmp scsi_vhci zfs mpt sd ip
hook neti arp usba sockfs kssl qlc fctl stmf stmf_sbd md lofs
random idm fcp crypto cpc smbsrv nfs fcip sppp ufs logindmux
ptm nsmb scu mpt_sas pmcs emlxs ]
> ::memstat
Page Summary                Pages                MB  %Tot
------------     ----------------  ----------------  ----
Kernel                     160876               628   16%
ZFS File Data              303401              1185   30%
Anon                        25335                98    2%
Exec and libs                1459                 5    0%
Page cache                   5083                19    1%
Free (cachelist)             6616                25    1%
Free (freelist)            510870              1995   50%

Total                     1013640              3959
Physical                  1013639              3959
> $q
```

When using either the `kmdb` or `mdb` debugger, the module name prefix is not required. After a module is loaded, its symbols form a common name space with the core kernel symbols and any other previously loaded module symbols.

# Tuning Format of Tunable Parameters Descriptions

This section describes the format for tuning Oracle Solaris parameters.

***Parameter***
The exact name of the tunable.
Some parameters use the naming convention *module*:*parameter* to indicate that the parameter belongs to a loadable module. For example, `tmpfs:tmpfs_maxkmem` means that `tmpfs_maxkmem` is a parameter of the `tmpfs` module.

**Description**
Briefly describes what the parameter does or controls.

**Data Type**
Indicates the signed or unsigned short integer or long integer. A long integer is twice the width in bits as an integer. For example, an unsigned integer = 32 bits, an unsigned long integer = 64 bits.

**Units**
(Optional) Describes the unit type.

**Default**
Indicates the value that the system uses by default.

**Range**
Specifies the possible range allowed by system validation or the bounds of the data type.

- MAXINT – A shorthand description for the maximum value of a signed integer (2,147,483,647)
- MAXUINT – A shorthand description for the maximum value of an unsigned integer (4,294,967,295)

**Dynamic?**
Indicates whether the parameter can by configured on a running system with the `mdb` or `kmdb` debugger (`Yes`), or only during boot time initialization (`No`).

**Validation**
Checks that the system applies to the value of the variable either as specified in the `/etc/system` file or the default value, as well as when the validation is applied.

**Implicit**
(Optional) Provides unstated constraints that might exist on the parameter, especially in relation to other parameters.

**When to Change**
Explains why someone might want to change this value. Includes error messages or return codes.

**Zone Configuration**
Identifies whether the parameter can be set in a exclusive-IP zone or must be set in the global zone. None of the parameters can be set in shared-IP zones.

**Commitment Level**
Identifies the stability of the interface. Many of the parameters in this manual are still evolving and are classified as unstable. For more information, see the `attributes`(7) man page.

# Special Oracle Solaris `tune` and `var` Structures

Oracle Solaris tunable parameters come in a variety of forms. The `tune` structure defined in the`/usr/include/sys/tuneable.h` file is the runtime representation of `tune_t_fsflushr`, `tune_t_minarmem`, and `tune_t_flkrec`. After the kernel is initialized, all references to these variables are found in the appropriate field of the `tune` structure.

The proper way to set parameters for this structure at boot time is to initialize the special parameter that corresponds to the desired field name. The system initialization process then loads these values into the `tune` structure.

A second structure into which various tunable parameters are placed is the `var` structure named `v`. You can find the definition of a `var` structure in the `/usr/include/sys/var.h` file. The runtime representation of variables such as `autoup` and `bufhwm` is stored here.

> **Caution:**
>
> Do not change either the `tune` or `v` structure on a running system. Changing any field in these structures on a running system might cause asystem panic.

# Viewing Oracle Solaris System Configuration Information

Several tools are available to examine system configuration information. Some tools require superuser privilege. Other tools can be run by a non-privileged user. Every structure and data item can be examined with the kernel debugger by using `mdb` on a running system or by booting under `kmdb`.

For more information, see the `mdb`(1) or `kmdb`(1) man page.

## `sysdef` Command

The `sysdef` command provides the values of memory and process resource limits, and portions of the `tune` and `v` structures. For example, the `sysdef` "Tunable Parameters" section from a SPARC T3-4 system with 500 GB of memory is as follows:

```
2206203904      maximum memory allowed in buffer cache (bufhwm)
65546           maximum number of processes (v.v_proc)
99              maximum global priority in sys class (MAXCLSYSPRI)
65541           maximum processes per user id (v.v_maxup)
30              auto update time limit in seconds (NAUTOUP)
25              page stealing low water mark (GPGSLO)
1               fsflush run rate (FSFLUSHR)
25              minimum resident memory for avoiding deadlock (MINARMEM)
25              minimum swapable memory for avoiding deadlock (MINASMEM)
```

For more information, see the `sysdef`(8) man page.

## `kstat` Utility

`kstat`s are data structures maintained by various kernel subsystems and drivers. They provide a mechanism for exporting data from the kernel to user programs without requiring that the program read kernel memory or have superuser privilege. For more information, see the `kstat`(8) and the `kstat`(3KSTAT) man pages.

## `kstat2` Utility

The `kstat2` utility shows values of kernel statistics (`kstats`). If you do not know the full name of the statistic you want, you can use a shell glob pattern or Perl Compatible Regular expression (PCRE). With options, you can show multiple statistic values over specified increments of time, and show the timestamp for each value, for example. For more information and examples, see the kstat2(8) man page. See also Oracle Solaris Observability Tools and Troubleshooting IPsec and IKE Semantic Errors in *Securing the Network in Oracle Solaris 11.4*.

# Oracle Solaris Observability Tools

The Oracle Solaris System Web Interface shows `kstats` and many other statistics in a graphical form. The Observability Tools also shows relationships among statistics and shows values over time to help you diagnose problems with the system. For more information, see *Using Oracle Solaris 11.4 StatsStore and System Web Interface*.

2

# Oracle Solaris Kernel Tunable Parameters

This chapter describes most of the Oracle Solaris kernel tunable parameters.


- General Kernel and Memory Parameters
- fsflush and Related Parameters
- Process-Sizing Parameters
- Paging-Related Parameters
- Kernel Memory Allocator
- General Driver Parameters
- Network Driver Parameters
- General I/O Parameters
- General File System Parameters
- TMPFS Parameters
- Pseudo Terminals
- STREAMS Parameters
- System V Message Queues
- System V Semaphores
- Timer Behavior
- SPARC: Platform Specific Parameters
- Locality Group Parameters


For other types of tunable parameters, refer to the following:

- Oracle Solaris ZFS tunables parameters – Oracle Solaris ZFS Tunable Parameters
- NFS tunable parameters – NFS Tunable Parameters
- Internet Protocol Suite tunable parameters – Internet Protocol Suite Tunable Parameters
- System facility tunable parameters – System Facility Parameters

## General Kernel and Memory Parameters

This section describes general kernel parameters that are related to physical memory and stack configuration. For ZFS-related memory parameters, see Oracle Solaris ZFS Tunable Parameters.

# default_stksize Parameter

**Description**
Specifies the default stack size of all threads. No thread can be created with a stack size smaller than default_stksize. If default_stksize is set, it overrides lwp_default_stksize. See also lwp_default_stksize Parameter.

**Data Type**
Integer

**Default**

- 4 x PAGESIZE on SPARC systems with sun4v processors
- 5 x PAGESIZE on x64 systems

**Range**
Minimum is the default values:

- 4 x PAGESIZE on SPARC systems with sun4v processors
- 5 x PAGESIZE on x64 systems

Maximum is 32 times the default value.

**Units**
Bytes in multiples of the value returned by the getpagesize parameter. For more information, see the getpagesize(3C) man page.

**Dynamic?**
Yes. Affects threads created after the variable is changed.

**Validation**
Must be greater than or equal to 8192 and less than or equal to 262,144 (256 x 1024). Also must be a multiple of the system page size. If these conditions are not met, the following message is displayed:

```
Illegal stack size, Using N
```

The value of *N* is the default value of default_stksize.

**When to Change**
When the system panics because it has run out of stack space. The best solution for this problem is to determine why the system is running out of space and then make a correction.
Increasing the default stack size means that almost every kernel thread will have a larger stack, resulting in increased kernel memory consumption for no good reason. Generally, that space will be unused. The increased consumption means other resources that are competing for the same pool of memory will have the amount of space available to them reduced, possibly decreasing the system's ability to perform work. Among the side effects is a reduction in the number of threads that the kernel can create. This solution should be treated as no more than an interim workaround until the root cause is remedied.

**Commitment Level**
Unstable

## logevent_max_q_sz Parameter

**Description**
Maximum number of system events allowed to be queued and waiting for delivery to the `syseventd` daemon. Once the size of the system event queue reaches this limit, no other system events are allowed on the queue.

**Data Type**
Integer

**Default**
5000

**Range**
0 to MAXINT

**Units**
System events

**Dynamic?**
Yes

**Validation**
The system event framework checks this value every time a system event is generated by `ddi_log_sysevent` and `sysevent_post_event`.
For more information, see the `ddi_log_sysevent`(9F) and `sysevent_post_event`(3SYSEVENT) man pages.

**When to Change**
When error log messages indicate that a system event failed to be logged, generated, or posted.

**Commitment Level**
Unstable

## lwp_default_stksize Parameter

**Description**
Specifies the default value of the stack size to be used when a kernel thread is created, and when the calling routine does not provide an explicit size to be used. Any stack size that you specify is increased by a one-page redzone.

**Data Type**
Integer

**Default**

- Default SPARC stack size is 3 pages (3 x 8,192 = 24,576) + 8 KB redzone
- Default x64 stack size is 5 pages (5 x 4,096 = 20,480) + 4 KB redzone

**Range**
Minimum is the default values:

- 3 x `PAGESIZE` on SPARC systems
- 5 x `PAGESIZE` on x64 systems

Maximum is 32 times the default value.

**Units**
Bytes in multiples of the value returned by the `getpagesize` parameter. For more information, see the `getpagesize`(3C) man page.

**Dynamic?**
Yes. Affects threads created after the variable is changed.

**Validation**
Must be greater than or equal to 8192 and less than or equal to 262,144 (256 x 1024). Also must be a multiple of the system page size. If these conditions are not met, the following message is displayed:

```
Illegal stack size, Using N
```

The value of *N* is the default value of `lwp_default_stksize`.

**When to Change**
When the system panics because it has run out of stack space. The best solution for this problem is to determine why the system is running out of space and then make a correction.
Increasing the default stack size means that almost every kernel thread will have a larger stack, resulting in increased kernel memory consumption for no good reason. Generally, that space will be unused. The increased consumption means other resources that are competing for the same pool of memory will have the amount of space available to them reduced, possibly decreasing the system's ability to perform work. Among the side effects is a reduction in the number of threads that the kernel can create. This solution should be treated as no more than an interim workaround until the root cause is remedied.

**Commitment Level**
Unstable

## `noexec_user_stack` Parameter

> **Note:**
>
> Although `noexec_user_stack` is still operational, this parameter is deprecated in this Oracle Solaris release. Use the `nxheap` and `nxstack` security extensions instead. You can control and configure Oracle Solaris extensions at the system level and at the process level with the `sxadm` command. For procedures and examples that show the use of `nxheap` and `nxstack`, see Protecting the Process Heap and Executable Stacks From Compromise in *Securing Systems and Attached Devices in Oracle Solaris 11.4*. For more information about the `sxadm` command, see the `sxadm`(8) man page. For guidelines to secure and harden Oracle Solaris, see *Oracle Solaris 11.4 Security and Hardening Guidelines*.

**Description**
Enables the stack to be marked as nonexecutable, which helps make buffer-overflow attacks more difficult.
An Oracle Solaris system running a 64-bit kernel makes the stacks of all 64-bit applications nonexecutable by default. Setting this parameter is necessary to make the stacks of all 32-bit applications nonexecutable by default if they weren't linked with the nxstack security extensions flag. This parameter, together with `noexec_user_stack_log`, can be set in a file in the `/etc/system.d` directory. See Protecting the Process Heap and Executable Stacks From Compromise in *Securing Systems and Attached Devices in Oracle Solaris 11.4*.

**Data Type**
Signed integer

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Toggle (on/off)

**Dynamic?**
Yes. Does not affect currently running processes, only processes created after the value is set.

**Validation**
None

**When to Change**
Should be enabled at all times unless applications are deliberately placing executable code on the stack without using `mprotect` to make the stack executable. For more information, see the `mprotect`(2) man page.

**Commitment Level**
Unstable

## `physmem` Parameter

**Description**
Modifies the system's configuration of the number of physical pages of memory after the Oracle Solaris OS and firmware are accounted for.

**Data Type**
Unsigned long

**Default**
Number of usable pages of physical memory available on the system, not counting the memory where the core kernel and data are stored

**Range**
1 to amount of physical memory on system

**Units**
Pages

**Dynamic?**
No

**Validation**
None

**When to Change**
Whenever you want to test the effect of running the system with less physical memory. Because this parameter does *not* take into account the memory used by the core kernel and data, as well as various other data structures allocated early in the startup process, the value of `physmem` should be less than the actual number of pages that represent the smaller amount of memory.

**Commitment Level**
Unstable

## `fsflush` and Related Parameters

This section describes `fsflush` and related tunables.

### `autoup` Parameter

**Description**
Along with `tune_t_flushr`, `autoup` controls the amount of memory examined for dirty pages in each invocation and frequency of file system synchronizing operations. The value of `autoup` is also used to control whether a buffer is written out from the free list. Buffers marked with the `B_DELWRI` flag (which identifies file content pages that have changed) are written out whenever the buffer has been on the list for longer than *autoup* seconds. Increasing the value of `autoup` keeps the buffers in memory for a longer time.

**Data Type**
Signed integer

**Default**
30

**Range**
1 to MAXINT

**Units**
Seconds

**Dynamic?**
No

**Validation**
If `autoup` is less than or equal to zero, it is reset to 30 and a warning message is displayed. This check is done only at boot time.

**Implicit**
`autoup` should be an integer multiple of `tune_t_fsflushr`. At a minimum, `autoup` should be at least 6 times the value of `tune_t_fsflushr`. If not, excessive amounts of memory are scanned each time `fsflush` is invoked.

The total system pages multiplied by `tune_t_fsflushr` should be greater than or equal to `autoup` to cause memory to be checked if `dopageflush` is non-zero.

**When to Change**
Here are several potential situations for changing `autoup`, `tune_t_fsflushr`, or both:

- Systems with large amounts of memory – In this case, increasing `autoup` reduces the amount of memory scanned in each invocation of `fsflush`.
- Systems with minimal memory demand – Increasing both `autoup` and `tune_t_fsflushr` reduces the number of scans made. `autoup` should be increased also to maintain the current ratio of `autoup` / `tune_t_fsflushr`.
- Systems with large numbers of transient files (for example, mail servers or software build systems) – If large numbers of files are created and then deleted, `fsflush` might unnecessarily write data pages for those files to disk.

**Commitment Level**
Unstable

## `doiflush` Parameter

**Description**
Controls whether file system metadata syncs will be executed during `fsflush` invocations. This synchronization is done every *N*th invocation of `fsflush` where *N* = `(autoup / tune_t_fsflushr)`. Because this algorithm is integer division, if `tune_t_fsflushr` is greater than `autoup`, a synchronization is done on every invocation of `fsflush` because the code checks to see if its iteration counter is greater than or equal to *N*. Note that *N* is computed once on invocation of `fsflush`. Later changes to `tune_t_fsflushr` or `autoup` have no effect on the frequency of synchronization operations.

**Data Type**
Signed integer

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Toggle (on/off)

**Dynamic?**
Yes

**Validation**
None

**When to Change**
When files are frequently modified over a period of time and the load caused by the flushing perturbs system behavior.
Files whose existence, and therefore consistency of state, does not matter if the system reboots are better kept in a TMPFS file system (for example, `/tmp`). Inode traffic can be

reduced on systems by using the `mount` -noatime option. This option eliminates inode updates when the file is accessed.
For a system engaged in realtime processing, you might want to disable this option and use explicit application file synchronizing to achieve consistency.

**Commitment Level**
Unstable

## `dopageflush` Parameter

**Description**
Controls whether memory is examined for modified pages during `fsflush` invocations. In each invocation of `fsflush`, the number of physical memory pages in the system is determined. This number might have changed because of a dynamic reconfiguration operation. Each invocation scans by using this algorithm: total number of pages x `tune_t_fsflushr` / `autoup` pages

**Data Type**
Signed integer

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Toggle (on/off)

**Dynamic?**
Yes

**Validation**
None

**When to Change**
If the system page scanner rarely runs, which is indicated by a value of 0 in the `sr` column of `vmstat` output.

**Commitment Level**
Unstable

## `fsflush` Parameter

The system daemon, `fsflush`, runs periodically to do three main tasks:

1. On every invocation, `fsflush` flushes dirty file system pages over a certain age to disk.
2. On every invocation, `fsflush` examines a portion of memory and causes modified pages to be written to their backing store. Pages are written if they are modified and if they do not meet one of the following conditions:
   - Pages are kernel page
   - Pages are free

- Pages are locked
- Pages are associated with a swap device
- Pages are currently involved in an I/O operation

The net effect is to flush pages from files that are mapped with `mmap` with write permission and that have actually been changed.

Pages are flushed to backing store but left attached to the process using them. This will simplify page reclamation when the system runs low on memory by avoiding delay for writing the page to backing store before claiming it, if the page has not been modified since the flush.

3. `fsflush` writes file system metadata to disk. This write is done every *n*th invocation, where *n* is computed from various configuration variables. See tune_t_fsflushr Parameter and autoup Parameter for details.

The following features are configurable:

- Frequency of invocation (`tune_t_fsflushr`)
- Whether memory scanning is executed (`dopageflush`)
- Whether file system data flushing occurs (`doiflush`)
- The frequency with which file system data flushing occurs (`autoup`)

For most systems, memory scanning and file system metadata synchronizing are the dominant activities for `fsflush`. Depending on system usage, memory scanning can be of little use or consume too much CPU time.

## `tune_t_fsflushr` Parameter

**Description**
Specifies the number of seconds between `fsflush` invocations

**Data Type**
Signed integer

**Default**
1

**Range**
1 to MAXINT

**Units**
Seconds

**Dynamic?**
No

**Validation**
If the value is less than or equal to zero, the value is reset to 1 and a warning message is displayed. This check is done only at boot time.

**When to Change**
See the `autoup` parameter.

**Commitment Level**
Unstable

# Process-Sizing Parameters

Several parameters (or variables) are used to control the number of processes that are available on the system and the number of processes that an individual user can create. The foundation parameter is `maxusers`. This parameter drives the values assigned to `max_nprocs` and `maxuprc`.

## `max_nprocs` Parameter

**Description**
Specifies the maximum number of processes that can be created on a system. Includes system processes and user processes. Any value specified in `/etc/system.d`*`file`* is used in the computation of `maxuprc`.
This value is also used in determining the size of several other system data structures. Other data structures where this parameter plays a role are as follows:

- Determining the size of the directory name lookup cache (if `ncsize` is not specified)
- Verifying that the amount of memory used by configured system V semaphores does not exceed system limits
- Configuring Hardware Address Translation resources for x86 platforms

**Data Type**
Signed integer

**Default**
10 + (16 x `maxusers`) if `maxusers` is set in `/etc/system.d/`*`file`*
The larger of 30,000 or 10 + (128 x number of CPUs), if `maxusers` is not set in `/etc/system.d/`*`file`*

**Range**
26 to value of `maxpid`

**Dynamic?**
No

**Validation**
Yes. If the value exceeds `maxpid`, it is set to `maxpid` .

**When to Change**
Changing this parameter is one of the steps necessary to enable support for more than 30,000 processes on a system.

**Commitment Level**
Unstable

## maxuprc Parameter

**Description**
Specifies the maximum number of processes that can be created on a system by any one user.

**Data Type**
Signed integer

**Default**
max_nprocs - reserved_procs

**Range**
1 to max_nprocs - reserved_procs

**Units**
Processes

**Dynamic?**
No

**Validation**
Yes. This value is compared to max_nprocs - reserved_procs and set to the smaller of the two values.

**When to Change**
When you want to specify a hard limit for the number of processes a user can create that is less than the default value of however many processes the system can create. Attempting to exceed this limit generates the following warning messages on the console or in the messages file:

```
out of per-user processes for uid N
```

**Commitment Level**
Unstable

## maxusers Parameter

**Description**
Originally, maxusers defined the number of logged in users the system could support. When a kernel was generated, various tables were sized based on this setting. Current Oracle Solaris releases do much of its sizing based on the amount of memory on the system. Thus, much of the past use of maxusers has changed. The following list identifies subsystems that are still derived from maxusers:

- The maximum number of processes on the system
- The number of quota structures held in the system
- The size of the directory name look-up cache (DNLC)

**Data Type**
Signed integer

**Default**
Lesser of the amount of memory in MB or 2048, and the greater of that value and nCPUs x 8

**Range**
1 to the greater of 2048 or nCPUs x 8, based on the size of physical memory, if not set in `/etc/system.d/file`
1 to the greater of 4096 or the nCPUs x 8, if set in `/etc/system.d/file`

**Units**
Users

**Dynamic?**
No. After computation of dependent parameters is done, `maxusers` is never referenced again.

**Validation**
If the value is greater than the maximum allowed, it is reset to the maximum. A message to that effect is displayed.

**When to Change**
When the default number of user processes derived by the system is too low. This situation is evident when the following message displays on the system console:

```
out of processes
```

You might also change this parameter when the default number of processes is too high, as in these situations:

- Database servers that have a lot of memory and relatively few running processes can save system memory when the default value of `maxusers` is reduced.
- If file servers have a lot of memory and few running processes, you might reduce this value. However, you should explicitly set the size of the DNLC. See ncsize Parameter.

**Commitment Level**
Unstable

## `ngroups_max` Parameter

**Description**
Specifies the maximum number of supplemental groups per process.

**Data Type**
Signed integer

**Units**
Groups

**Dynamic?**
No

**Validation**
Yes. If `ngroups_max` is set to an invalid value, it is automatically reset to the closest legal value. For example, if it is set to less than zero, it is reset to 0. If it is set to greater than 1024, it is reset to 1024.

**When to Change**
Review the following considerations if you are using NFS `AUTH_SYS` authentication and you want to increase the default `ngroups_max` value:

1. If `ngroups_max` is set to 16 or if the NFS client's `AUTH_SYS` credential that is provided has 15 or fewer groups, the client's group information is used.
2. If `ngroups_max` is set to greater than 16 **and** the NFS client's `AUTH_SYS` credential from the name server contains exactly 16 groups, the maximum allowed, the NFS server consults the name server and matches the client's UID to a user name. Then, the name server computes a list of groups to which the user belongs.

**Commitment Level**
Unstable

## `pidmax` Parameter

**Description**
Specifies the value of the largest possible process ID.
`pidmax` sets the value for the `maxpid` variable. Once `maxpid` is set, `pidmax` is ignored. `maxpid` is used elsewhere in the kernel to determine the maximum process ID and for validation checking.
Any attempts to set `maxpid` by adding an entry to a file in the `/etc/system.d` directory have no effect.

**Data Type**
Signed integer

**Default**
30,000

**Range**
5 to 999,999

**Units**
Processes

**Dynamic?**
No. Used only at boot time to set the value of `pidmax`.

**Validation**
Yes. Value is compared to the value of `reserved_procs` and 999,999. If less than `reserved_procs` or greater than 999,999, the value is set to 999,999.

**Implicit**
`max_nprocs` range checking ensures that `max_nprocs` is always less than or equal to this value.

**When to Change**
Required to enable support for more than 30,000 processes on a system. See also max_nprocs Parameter.

**Commitment Level**
Unstable

### `reserved_procs` Parameter

**Description**
Specifies the number of system process slots to be reserved in the process table for processes with a UID of root (0). For example, `fsflush` has a UID of root (0).

**Data Type**
Signed integer

**Default**
5

**Range**
5 to MAXINT

**Units**
Processes

**Dynamic?**
No. Not used after the initial parameter computation.

**Validation**
Any `/etc/system.d/`*`file`* setting is honored.

**Commitment Level**
Unstable

**When to Change**
Consider increasing to 10 + the normal number of UID 0 (root) processes on system. This setting provides some cushion should it be necessary to obtain a root shell when the system is otherwise unable to create user-level processes.

## Paging-Related Parameters

The Oracle Solaris OS uses a demand paged virtual memory system. As the system runs, pages are brought into memory as needed. When memory becomes occupied above a certain threshold and demand for memory continues, paging begins. Paging goes through several levels that are controlled by certain parameters.

The general paging algorithm is as follows:

- A memory deficit is noticed. The page scanner thread runs and begins to walk through memory. A two-step algorithm is employed:
  1. A page is marked as unused.
  2. If still unused after a time interval, the page is viewed as a subject for reclaim.

  If the page has been modified, a request is made to the pageout thread to schedule the page for I/O. Also, the page scanner continues looking at memory.

Pageout causes the page to be written to the page's backing store and placed on the free list. When the page scanner scans memory, no distinction is made as to the origin of the page. The page might have come from a data file, or it might represent a page from an executable's text, data, or stack.

- As memory pressure on the system increases, the algorithm becomes more aggressive in the pages it will consider as candidates for reclamation and in how frequently the paging algorithm runs. (For more information, see fastscan Parameter and slowscan Parameter.) As available memory falls between the range `lotsfree` and `minfree`, the system linearly increases the amount of memory scanned in each invocation of the pageout thread from the value specified by `slowscan` to the value specified by `fastscan`. The system uses the `desfree` parameter to control a number of decisions about resource usage and behavior.

The system initially constrains itself to use no more than 4 percent of one CPU for `pageout` operations. As memory pressure increases, the amount of CPU time consumed in support of `pageout` operations linearly increases until a maximum of 80 percent of one CPU is consumed. The algorithm looks through some amount of memory between `slowscan` and `fastscan`, then stops when one of the following occurs:

- Enough pages have been found to satisfy the memory shortfall.
- The planned number of pages have been looked at.
- Too much time has elapsed.

If a memory shortfall is still present when pageout finishes its scan, another scan is scheduled for 1/4 second in the future.

The configuration mechanism of the paging subsystem was changed. Instead of depending on a set of predefined values for `fastscan`, `slowscan`, and `handspreadpages`, the system determines the appropriate settings for these parameters at boot time. Setting any of these parameters in files in the `/etc/system.d` directory can cause the system to use less than optimal values.

> **Caution:**
>
> Remove all tuning of the VM system from files in the `/etc/system.d` directory. Run with the default settings and determine if it is necessary to adjust any of these parameters. Do not set either `cachefree` or `priority_paging`.

Dynamic reconfiguration (DR) for CPU and memory is supported. A system in a DR operation that involves the addition or deletion of memory recalculates values for the relevant parameters, unless the parameter has been explicitly set in `/etc/system.d/`*`file`* . In that case, the value specified in `/etc/system.d/`*`file`* is used, unless a constraint on the value of the variable has been violated. In this case, the value is reset.

## `desfree` Parameter

**Description**
Specifies the preferred amount of memory to be free at all times on the system.

**Data Type**
Unsigned integer

**Default**
`lotsfree` / 2

**Range**
The minimum value is 256 KB or 1/128th of physical memory, whichever is greater, expressed as pages using the page size returned by `getpagesize`.
The maximum value is the number of physical memory pages. The maximum value should be no more than 15 percent of physical memory. The system does not enforce this range other than that described in the Validation section.

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in `/etc/system.d/`*`file`* or calculated from the new physical memory value.

**Validation**
If `desfree` is greater than `lotsfree`, `desfree` is set to `lotsfree` / 2. No message is displayed.

**Implicit**
The relationship of `lotsfree` being greater than `desfree`, which is greater than `minfree`, should be maintained at all times.

**Side Effects**
Several side effects can arise from increasing the value of this parameter. When the new value nears or exceeds the amount of available memory on the system, the following can occur:

- Asynchronous I/O requests are not processed, unless available memory exceeds `desfree`. Increasing the value of `desfree` can result in rejection of requests that otherwise would succeed.
- NFS asynchronous writes are executed as synchronous writes.
- The swapper is awakened earlier, and the behavior of the swapper is biased towards more aggressive actions.
- The system might not preload (prefault) as many executable pages as possible into the system. This side effect results in applications potentially running slower than they otherwise would.

**When to Change**
For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 256 KB, expressed as pages using the page size returned by `getpagesize`.

**Commitment Level**
Unstable

## fastscan Parameter

**Description**
Defines the maximum number of pages per second that the system looks at when memory pressure is highest.

**Data Type**
Signed integer

**Default**
The fastscan default value is set in one of the following ways:

- The fastscan value set in /etc/system.d/*file* is used.
- The maxfastscan value set in /etc/system.d/*file* is used.
- If neither fastscan nor maxfastscan is set in /etc/system.d/*file* , fastscan is set to 64 MB when the system is booted. Then, after the system is booted for a few minutes, the fastscan value is set to the number of pages that the scanner can scan in one second using 10% of a CPU.

In all three cases, if the derived value is more than half the memory in the system, the fastscan value is capped at the value of half the memory in the system.

**Range**
64 MB to half the system's physical memory

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided by /etc/system.d/*file* or calculated from the new physical memory value.

**Validation**
The maximum value is the lesser of 64 MB and 1/2 of physical memory.

**When to Change**
When more aggressive scanning of memory is preferred during periods of memory shortfall, especially when the system is subject to periods of intense memory demand or when performing heavy file I/O.

**Commitment Level**
Unstable

## handspreadpages Parameter

**Description**
The Oracle Solaris OS uses a two-handed clock algorithm to look for pages that are candidates for reclaiming when memory is low. The first hand of the clock walks through memory marking pages as unused. The second hand walks through memory some distance after the first hand, checking to see if the page is still marked as unused. If so, the page is subject to being reclaimed. The distance between the first hand and the second hand is handspreadpages.

**Data Type**
Unsigned long

**Default**
`fastscan`

**Range**
1 to maximum number of physical memory pages on the system.

**Units**
Pages

**Dynamic?**
Yes. This parameter requires that the kernel `reset_hands` parameter also be set to a non-zero value. Once the new value of `handspreadpages` has been recognized, `reset_hands` is set to zero.

**Validation**
The value is set to the lesser of either the amount of physical memory and the `handspreadpages` *value*.

**When to Change**
When you want to increase the amount of time that pages are potentially resident before being reclaimed. Increasing this value increases the separation between the hands, and therefore, the amount of time before a page can be reclaimed.

**Commitment Level**
Unstable

## `lotsfree` Parameter

**Description**
Serves as the initial trigger for system paging to begin. When this threshold is crossed, the page scanner wakes up to begin looking for memory pages to reclaim.

**Data Type**
Unsigned long

**Default**
The greater of 1/64th of physical memory or 512 KB

**Range**
The minimum value is 512 KB or 1/64th of physical memory, whichever is greater, expressed as pages using the page size returned by `getpagesize`. For more information, see the `getpagesize`(3C) man page.
The maximum value is the number of physical memory pages. The maximum value should be no more than 30 percent of physical memory. The system does not enforce this range, other than that described in the Validation section.

**Units**
Pages

**Dynamic?**
Yes, but dynamic changes are lost if a memory-based DR operation occurs.

**Validation**
If `lotsfree` is greater than the amount of physical memory, the value is reset to the default.

**Implicit**
The relationship of `lotsfree` being greater than `desfree`, which is greater than `minfree`, should be maintained at all times.

**When to Change**
When demand for pages is subject to sudden sharp spikes, the memory algorithm might be unable to keep up with demand. One workaround is to start reclaiming memory at an earlier time. This solution gives the paging system some additional margin.
A rule of thumb is to set this parameter to 2 times what the system needs to allocate in a few seconds. This parameter is workload dependent. A DBMS server can probably work fine with the default settings. However, you might need to adjust this parameter for a system doing heavy file system I/O.
For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 512 KB, expressed as pages using the page size returned by `getpagesize`.

**Commitment Level**
Unstable

## `maxpgio` Parameter

**Description**
Defines the maximum number of page I/O requests that can be queued by the paging system. This number is divided by 4 to get the actual maximum number used by the paging system. This parameter is used to throttle the number of requests as well as to control process swapping.

**Data Type**
Signed integer

**Default**
400

**Range**
1 to a variable maximum that depends on the system architecture, but mainly by the I/O subsystem, such as the number of controllers, disks, and disk swap size

**Units**
I/0s

**Dynamic?**
No

**Validation**
None

**Implicit**
The maximum number of I/O requests from the pager is limited by the size of a list of request buffers, which is currently sized at 256.

**When to Change**
Increase this parameter to page out memory faster. A larger value might help to recover faster from memory pressure if more than one swap device is configured or if the swap device is a striped device. Note that the existing I/O subsystem should be able to handle the additional I/O load. Also, increased swap I/O could degrade application I/O performance if the swap partition and application files are on the same disk.

**Commitment Level**
Unstable

## `min_percent_cpu` Parameter

**Description**
Defines the minimum percentage of CPU that `pageout` can consume. This parameter is used as the starting point for determining the maximum amount of time that can be consumed by the page scanner.

**Data Type**
Signed integer

**Default**
4

**Range**
1 to 80

**Units**
Percentage

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Increasing this value on systems with multiple CPUs and lots of memory, which are subject to intense periods of memory demand, enables the pager to spend more time attempting to find memory.

**Commitment Level**
Unstable

## `minfree` Parameter

**Description**
Specifies the minimum acceptable memory level. When memory drops below this number, the system biases allocations toward allocations necessary to successfully complete pageout operations or to swap processes completely out of memory. Either allocation denies or blocks other allocation requests.

**Data Type**
Unsigned integer

**Default**
`desfree` / 2

**Range**
The minimum value is 128 KB or 1/256th of physical memory, whichever is greater, expressed as pages using the page size returned by `getpagesize`.
The maximum value is the number of physical memory pages. The maximum value should be no more than 7.5 percent of physical memory. The system does not enforce this range other than that described in the Validation section.

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in `/etc/system.d/`*`file`* or calculated from the new physical memory value.

**Validation**
If `minfree` is greater than `desfree`, `minfree` is set to `desfree` / 2. No message is displayed.

**Implicit**
The relationship of `lotsfree` being greater than `desfree`, which is greater than `minfree`, should be maintained at all times.

**When to Change**
The default value is generally adequate. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 128 KB, expressed as pages using the page size returned by `getpagesize`.

**Commitment Level**
Unstable

## `pageout_reserve` Parameter

**Description**
Specifies the number of pages reserved for the exclusive use of the pageout or scheduler threads. When available memory is less than this value, nonblocking allocations are denied for any processes other than pageout or the scheduler. Pageout needs to have a small pool of memory for its use so it can allocate the data structures necessary to do the I/O for writing a page to its backing store.

**Data Type**
Unsigned integer

**Default**
`throttlefree` / 2

**Range**
The minimum value is 64 KB or 1/512th of physical memory, whichever is greater, expressed as pages using the page size returned by `getpagesize(3C)`.
The maximum is the number of physical memory pages. The maximum value should be no more than 2 percent of physical memory. The system does not enforce this range, other than that described in the Validation section.

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in `/etc/system.d/`*`file`* or calculated from the new physical memory value.

**Validation**
If `pageout_reserve` is greater than `throttlefree` / 2, `pageout_reserve` is set to `throttlefree` / 2. No message is displayed.

**Implicit**
The relationship of `lotsfree` being greater than `desfree`, which is greater than `minfree`, should be maintained at all times.

**When to Change**
The default value is generally adequate. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 64 KB, expressed as pages using the page size returned by `getpagesize`.

**Commitment Level**
Unstable

## `pages_before_pager` Parameter

**Description**
Defines part of a system threshold that immediately frees pages after an I/O completes instead of storing the pages for possible reuse. The threshold is `lotsfree + pages_before_pager`. The NFS environment also uses this threshold to curtail its asynchronous activities as memory pressure mounts.

**Data Type**
Signed integer

**Default**
200

**Range**
1 to amount of physical memory

**Units**
Pages

**Dynamic?**
No

**Validation**
None

**When to Change**
You might change this parameter when the majority of I/O is done for pages that are truly read or written once and never referenced again. Setting this variable to a larger amount of memory keeps adding pages to the free list.

You might also change this parameter when the system is subject to bursts of severe memory pressure. A larger value here helps maintain a larger cushion against the pressure.

**Commitment Level**
Unstable

## pages_pp_maximum Parameter

**Description**
Defines the number of pages that must be unlocked. If a request to lock pages would force available memory below this value, that request is refused.

**Data Type**
Unsigned long

**Default**
The greater of (tune_t_minarmem + 100 and [4% of memory available at boot time + 4 MB])

**Range**
Minimum value enforced by the system is tune_t_minarmem + 100. The system does not enforce a maximum value.

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in /etc/system.d/*file* or was calculated from the new physical memory value.

**Validation**
If the value specified in a file in the /etc/system.d directory or the calculated default is less than tune_t_minarmem + 100, the value is reset to tune_t_minarmem + 100.
No message is displayed if the value from a /etc/system.d/*file* file is increased.
Validation is done only at boot time and during dynamic reconfiguration operations that involve adding or deleting memory.

**When to Change**
When memory-locking requests fail or when attaching to a shared memory segment with the SHARE_MMU flag fails, yet the amount of memory available seems to be sufficient.
Excessively large values can cause memory locking requests (mlock, mlockall, and memcntl) to fail unnecessarily. For more information, see the mlock(3C), mlockall(3C), and memcntl(2) man pages.

**Commitment Level**
Unstable

## slowscan Parameter

**Description**
Defines the minimum number of pages per second that the system looks at when attempting to reclaim memory.

**Data Type**
Signed integer

**Default**
The smaller of 1/20th of physical memory in pages and 100.

**Range**
1 to `fastscan` / 2

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in `/etc/system.d/`*`file`* or calculated from the new physical memory value.

**Validation**
If `slowscan` is larger than `fastscan` / 2, `slowscan` is reset to `fastscan` / 2. No message is displayed.

**When to Change**
When more aggressive scanning of memory is preferred during periods of memory shortfall, especially when the system is subject to periods of intense memory demand.

**Commitment Level**
Unstable

## `throttlefree` Parameter

**Description**
Specifies the memory level at which blocking memory allocation requests are put to sleep, even if the memory is sufficient to satisfy the request.

**Data Type**
Unsigned integer

**Default**
`minfree`

**Range**
The minimum value is 128 KB or 1/256th of physical memory, whichever is greater, expressed as pages using the page size returned by `getpagesize`.
The maximum value is the number of physical memory pages. The maximum value should be no more than 4 percent of physical memory. The system does not enforce this range other than that described in the Validation section.

**Units**
Pages

**Dynamic?**
Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in `/etc/system.d/`*`file`* or calculated from the new physical memory value.

**Validation**
If `throttlefree` is greater than `desfree`, `throttlefree` is set to `minfree`. No message is displayed.

**Implicit**
The relationship of `lotsfree` is greater than `desfree`, which is greater than `minfree`, should be maintained at all times.

**When to Change**
The default value is generally adequate. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 128 KB, expressed as pages using the page size returned by `getpagesize`. For more information, see the `getpagesize`(3C) man page.

**Commitment Level**
Unstable

### `tune_t_minarmem` Parameter

**Description**
Defines the minimum available resident (not swappable) memory to maintain necessary to avoid deadlock. Used to reserve a portion of memory for use by the core of the OS. Pages restricted in this way are not seen when the OS determines the maximum amount of memory available.

**Data Type**
Signed integer

**Default**
25

**Range**
1 to physical memory

**Units**
Pages

**Dynamic?**
No

**Validation**
None. Large values result in wasted physical memory.

**When to Change**
The default value is generally adequate. Consider increasing the default value if the system locks up and debugging information indicates that no memory was available.

**Commitment Level**
Unstable

## Kernel Memory Allocator

The Oracle Solaris kernel memory allocator distributes chunks of memory for use by clients inside the kernel. The allocator creates a number of caches of varying size for use by its clients. Clients can also request the allocator to create a cache for use by that client (for

example, to allocate structures of a particular size). Statistics about each cache that the allocator manages can be seen by using the `kstat -c kmem_cache` command.

Occasionally, systems might panic because of memory corruption. The kernel memory allocator supports a debugging interface (a set of flags), that performs various integrity checks on the buffers. The kernel memory allocator also collects information on the allocators. The integrity checks provide the opportunity to detect errors closer to where they actually occurred. The collected information provides additional data for support people when they try to ascertain the reason for the panic.

Use of the flags incurs additional overhead and memory usage during system operations. The flags should only be used when a memory corruption problem is suspected.

## `kmem_flags` Flag

**Description**
The Oracle Solaris kernel memory allocator has various debugging and test options. Five supported flag settings are described here.

| Flag | Setting | Description |
|---|---|---|
| `AUDIT` | `0x1` | The allocator maintains a log that contains recent history of its activity. The number of items logged depends on whether `CONTENTS` is also set. The log is a fixed size. When space is exhausted, earlier records are reclaimed. |
| `TEST` | `0x2` | The allocator writes a pattern into freed memory and checks that the pattern is unchanged when the buffer is next allocated. If some portion of the buffer is changed, then the memory was probably used by a client that had previously allocated and freed the buffer. If an overwrite is identified, the system panics. |
| `REDZONE` | `0x4` | The allocator provides extra memory at the end of the requested buffer and inserts a special pattern into that memory. When the buffer is freed, the pattern is checked to see if data was written past the end of the buffer. If an overwrite is identified, the kernel panics. |
| `CONTENTS` | `0x8` | The allocator logs up to 256 bytes of buffer contents when the buffer is freed. This flag requires that `AUDIT` also be set.<br>The numeric value of these flags can be logically added together and set by a file in the `/etc/system.d` directory. |
| `LITE` | `0x100` | Does minimal integrity checking when a buffer is allocated and freed. When enabled, the allocator checks that the redzone has not been written into, that a freed buffer is not being freed again, and that the buffer being freed is the size that was allocated. Do not combine this flag with any other flags. |

**Data Type**
Signed integer

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 - 15 or 256 (0x100)

**Dynamic?**
Yes. Changes made during runtime only affect new kernel memory caches. After system initialization, the creation of new caches is rare.

**Validation**
None

**When to Change**
When memory corruption is suspected

**Commitment Level**
Unstable

## `kmem_stackinfo` Variable

**Description**
If the `kmem_stackinfo` variable is enabled in an `/etc/system.d/`*`file`* at kernel thread creation time, the kernel thread stack is filled with a specific pattern instead of filled with zeros. During kernel thread execution, this kernel thread stack pattern is progressively overwritten. A simple count from the stack top until the pattern is not found gives a high watermark value, which is the maximum kernel stack space used by a kernel thread. This mechanism allows the following features:

- Compute the percentage of kernel thread stack really used (a high watermark) for current kernel threads in the system
- When a kernel thread ends, the system logs the last kernel threads that have used the most of their kernel thread stacks before dying to a small circular memory buffer

**Data Type**
Unsigned integer

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**Validation**
None

**When to Change**
When you want to monitor kernel thread stack usage. Keep in mind that when `kmem_stackinfo` is enabled, the performance of creating and deleting kthreads is decreased. See *Oracle Solaris Modular Debugger Guide*.

**Zone Configuration**
This parameter must be set in the global zone.

**Commitment Level**
Unstable

# General Driver Parameters

This section describes other drivers that apply to the kernel.

## SPARC: `dax_stats_flags` Flag

**Description**
This parameter controls what `dax kstats` are collected by the dax driver. This parameter can be set through the `/etc/system.d` directory. See /etc/system.d/ Directory Files for more information. Alternatively, you can also use the `mdb --kernel -w --unsafe-write-access` command. To read the parameter's current value, use the `mdb --kernel --unsafe-io-access dax_stats_flags/D`. For information about the `mdb --kernel -w --unsafe-write-access` and `mdb --kernel --unsafe-io-access` commands, see kmdb Debugger Entry in *Oracle Solaris Modular Debugger Guide* and the `mdb`(1) man page. Note that you should use the `--kernel` option instead of the obsolescent `-k` option.

**Data Type**
Unsigned Integer

**Default**
1

**Range**
Possible Values

- 1 – `dax` unit statistics collection only, excluding queue and CPU statistics.
- 2 – `dax` queue and `dax` CPU statistics collection. This setting is not recommended because collecting queue and CPU statistics can have an impact on performance.
- 3 – `dax` unit, queue and CPU statistics collection

**Dynamice?**
No

**Validation**
None

**When to Change**
When `dax` queue and or `dax` cpu statistics are needed.

**Commitment Level**
Unstable

## `ddi_msix_alloc_limit` Parameter

**Description**
x86 only: This parameter controls the number of Extended Message Signaled Interrupts (MSI-X) that a device instance can allocate. Due to an existing system limitation, the default value is 2. You can increase the number of MSI-X interrupts that a device instance can allocate by increasing the value of this parameter. This

parameter can be set either by editing an `/etc/system.d/file` or by setting it with `mdb` before the device driver attach occurs.

**Data Type**
Signed integer

**Default**
SPARC based systems: 8
x86 based systems: 2 If the system supports x2APIC, the apix module can increase the default value to 8.

**Range**
2-8

**Dynamic?**
Yes

**Validation**
None

**When to Change**
To increase the number of MSI-X interrupts that a device instance can allocate. However, if you increase the number of MSI-X interrupts that a device instance can allocate, adequate interrupts might not be available to satisfy all allocation requests. If this happens, some devices might stop functioning or the system might fail to boot. Reduce the value or remove the parameter in this case.

**Commitment Level**
Unstable

## `moddebug` Parameter

**Description**
When this parameter is enabled, messages about various steps in the module loading process are displayed.

**Data Type**
Signed integer

**Default**
0 (messages off)

**Range**
Here are the most useful values:

- 0x80000000 – Prints `[un] loading...` message. For every module loaded, messages such as the following appear on the console and in the `/var/adm/messages` file:

  ```
  Apr 20 17:18:04 neo genunix: [ID 943528 kern.notice] load 'sched/TS_DPTBL' id 15
  loaded @ 0x7be1b2f8/0x19c8380 size 176/2096
  Apr 20 17:18:04 neo genunix: [ID 131579 kern.notice] installing TS_DPTBL,
  module id 15.
  ```

- 0x40000000 – Prints detailed error messages. For every module loaded, messages such as the following appear on the console and in the `/var/adm/messages` file:

```
Apr 20 18:30:00 neo unix: Errno = 2
Apr 20 18:30:00 neo unix: kobj_open: vn_open of /platform/sun4v/kernel/exec/
sparcv9/intpexec fails
Apr 20 18:30:00 neo unix: Errno = 2
Apr 20 18:30:00 neo unix: kobj_open: '/kernel/exec/sparcv9/intpexec'
Apr 20 18:30:00 neo unix:  vp = 60015777600
Apr 20 18:30:00 neo unix: kobj_close: 0x60015777600
Apr 20 18:30:00 neo unix: kobj_open: vn_open of /platform/SUNW,Sun-Fire-
T200/kernel/exec/sparcv9
/intpexec fails,
Apr 20 18:30:00 neo unix: Errno = 2
Apr 20 18:30:00 neo unix: kobj_open: vn_open of /platform/sun4v/kernel/exec/
sparcv9/intpexec fails
```

- 0x20000000 - Prints even more detailed messages. This value doesn't print any additional information beyond what the `0x40000000` flag does during system boot. However, this value does print additional information about releasing the module when the module is unloaded.

These values can be added together to set the final value.

**Dynamic?**
Yes

**Validation**
None

**When to Change**
When a module is either not loading as expected, or the system seems to hang while loading modules. Note that when `0x40000000` is set, system boot is slowed down considerably by the number of messages written to the console.

**Commitment Level**
Unstable

# Network Driver Parameters

This section describes network parameters that affect the kernel.

## IP Protocol Parameters in the Kernel

The following IP parameters can be set only in a file in the `/etc/system.d` directory. After the file is modified, reboot the system.

For example, the following entry sets the `ipcl_conn_hash_size` parameter:

```
set ip:ipcl_conn_hash_size=value
```

### `ip_squeue_worker_wait` Parameter

**Description**
Governs the maximum delay in waking up a worker thread to process TCP/IP packets that are enqueued on an squeue. An *squeue* is a serialization queue that is used by the TCP/IP kernel code to process TCP/IP packets.

**Default**
10 milliseconds

**Range**
0 – 50 milliseconds

**Dynamic?**
Yes

**When to Change**
Consider tuning this parameter if latency is an issue, and network traffic is light. For example, if the stem serves mostly interactive network traffic.
The default value usually works best on a network file server, a web server, or any stem that has substantial network traffic.

**Zone Configuration**
This parameter can only be set in the global zone.

**Commitment Level**
Unstable

## `ip_squeue_fanout` Parameter

**Description**
Determines the mode of associating TCP/IP connections with squeues.
A value of 0 associates a new TCP/IP connection with the CPU that creates the connection.
A value of 1 associates the connection with multiple squeues that belong to different CPUs.

**Default**
1

**Range**
0 or 1

**Dynamic?**
Yes

**When to Change**
Consider setting this parameter to 1 to spread the load across all CPUs in certain situations. For example, when the number of CPUs exceed the number of NICs, and one CPU is not capable of handling the network load of a single NIC, change this parameter to 1.

**Zone Configuration**
This parameter can only be set in the global zone.

**Commitment Level**
Unstable

## `ipcl_conn_hash_size` Parameter

**Description**
Controls the size of the connection hash table used by IP. The default value of 0 means that the system automatically sizes an appropriate value for this parameter at boot time, depending on the available memory.

**Data Type**
Unsigned integer

**Default**
0

**Range**
0 to 82,500

**Dynamic?**
No. The parameter can only be changed at boot time.

**When to Change**
If the system consistently has tens of thousands of TCP connections, the value can be increased accordingly. Increasing the hash table size means that more memory is wired down, thereby reducing available memory to user applications.

**Commitment Level**
Unstable

## `igb` Parameters

### `intr_force` Parameter

**Description**
This parameter is used to force an interrupt type, such as MSI, MSI-X, or legacy, that is used by the `igb` network driver. This parameter can be set by editing the `/etc/driver/drv/igb.conf` file before the `igb` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
0 (do not force an interrupt type)

**Range**
0 (do not force an interrupt type)
1 (force MSI-X interrupt type)
2 (force MSI interrupt type)
3 (force legacy interrupt type)

**Dynamic?**
No

**Validation**
None

**When to Change**
To force an interrupt type that is used by the `igb` network driver.

**Commitment Level**
Unstable

## mr_enable Parameter

**Description**
This parameter enables or disables multiple receive and transmit queues that are used by the igb network driver. This parameter can be set by editing the /etc/driver/drv/igb.conf file before the igb driver attach occurs.

**Data Type**
Boolean

**Default**
0 (disable multiple queues)

**Range**
0 (disable multiple queues) or 1 (enable multiple queues)

**Dynamic?**
No

**Validation**
None

**When to Change**
To enable or disable multiple receive and transmit queues that are used by the igb network driver.

**Commitment Level**
Unstable

# ixgbe Parameters

## intr_throttling Parameter

**Description**
This parameter controls the interrupt throttling rate of the ixgbe network driver. You can increase the rate of interrupt by decreasing the value of this parameter. This parameter can be set by editing the /etc/driver/drv/ixgbe.conf file before the ixgbe driver attach occurs.

**Data Type**
Unsigned integer

**Default**
200

**Range**
0 to 65535

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the interrupt throttling rate that is used by the `ixgbe` network driver.

**Commitment Level**
Unstable

## `rx_copy_threshold` Parameter

**Description**
This parameter controls the receive buffer copy threshold that is used by the `ixgbe` network driver. You can increase the receive buffer copy threshold by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
128

**Range**
0 to 9126

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the receive buffer copy threshold that is used by the `ixgbe` network driver.

**Commitment Level**
Unstable

## `rx_limit_per_intr` Parameter

**Description**
This parameter controls the maximum number of receive queue buffer descriptors per interrupt that are used by the `ixgbe` network driver. You can increase the number of receive queue buffer descriptors by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
256

**Range**
16 to 4096

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the number of receive queue buffer descriptors that are handled per interrupt by the `ixgbe` network driver.

**Commitment Level**
Unstable

## `rx_queue_number` Parameter

**Description**
This parameter controls the number of receive queues that are used by the `ixgbe` network driver. You can increase the number of receive queues by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
8

**Range**
1 to 64

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the number of receive queues that are used by the `ixgbe` network driver.

**Commitment Level**
Unstable

## `rx_ring_size` Parameter

**Description**
This parameter controls the receive queue size that is used by the `ixgbe` network driver. You can increase the receive queue size by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
1024

**Range**
64 to 4096

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the receive queue size that is used by the `ixgbe` network driver.

**Commitment Level**
Unstable

## `tx_copy_threshold` Parameter

**Description**
This parameter controls the transmit buffer copy threshold that is used by the `ixgbe` network driver. You can increase the transmit buffer copy threshold by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
512

**Range**
0 to 9126

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the transmit buffer copy threshold that is used by the `ixgbe` network driver.

**Commitment Level**
Unstable

## `tx_queue_number` Parameter

**Description**
This parameter controls the number of transmit queues that are used by the `ixgbe` network driver. You can increase the number of transmit queues by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
8

**Range**
1 to 32

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the number of transmit queues that are used by the `ixgbe` network driver.

**Commitment Level**
Unstable

### `tx_ring_size` Parameter

**Description**
This parameter controls the transmit queue size that is used by the `ixgbe` network driver. You can increase the transmit queue size by increasing the value of this parameter. This parameter can be set by editing the `/etc/driver/drv/ixgbe.conf` file before the `ixgbe` driver attach occurs.

**Data Type**
Unsigned integer

**Default**
1024

**Range**
64 to 4096

**Dynamic?**
No

**Validation**
None

**When to Change**
To change the transmit queue size that is used by the `ixgbe` network driver.

**Commitment Level**
Unstable

## General I/O Parameters

This section describes parameters in function of input and output processes in the kernel.

### `maxphys` Parameter

**Description**
Defines the maximum size of physical I/O requests. If a driver encounters a request larger than this size, the driver breaks the request into `maxphys` sized chunks. File systems can and do impose their own limit.

**Data Type**
Signed integer

**Default**
131,072 (sun4v) or 57,344 (x86). The `sd` driver uses the value of 1,048,576 if the drive supports wide transfers. The `ssd` driver uses 1,048,576 by default.

**Range**
stem-specific page size to `MAXINT`

**Units**
Bytes

**Dynamic?**
Yes, but many file systems load this value into a per-mount point data structure when the file system is mounted. A number of drivers load the value at the time a device is attached to a driver-specific data structure.

**Validation**
None

**When to Change**
When doing I/O to and from raw devices in large chunks. Note that a DBMS doing OLTP operations issues large numbers of small I/Os. Changing `maxphys` does not result in any performance improvement in that case.

**Commitment Level**
Unstable

## `rlim_fd_cur` Parameter

**Description**
Defines the "soft" limit on file descriptors that a single process can have open. A process might adjust its file descriptor limit to any value up to the "hard" limit defined by `rlim_fd_max` by using the `setrlimit` call or by issuing the `limit` command in whatever shell it is running. You do not require superuser privilege to adjust the limit to any value less than or equal to the hard limit.

**Data Type**
Signed integer

**Default**
256 through Oracle Solaris 11.4 SRU 26
4095 starting with Oracle Solaris 11.4 SRU 27

**Range**
128 to `MAXINT`

**Units**
File descriptors

**Dynamic?**
No

**Validation**
Compared to `rlim_fd_max`. If `rlim_fd_cur` is greater than `rlim_fd_max`, `rlim_fd_cur` is reset to `rlim_fd_max`.

**When to Change**
When the default number of open files for a process is not enough. Increasing this value means only that it might not be necessary for a program to use `setrlimit` to increase the maximum number of file descriptors available to it.

**Commitment Level**
Unstable

## `rlim_fd_max` Parameter

**Description**
Specifies the "hard" limit on file descriptors that a single process might have open. Overriding this limit requires superuser privilege.

**Data Type**
Signed integer

**Default**
65536 through Oracle Solaris 11.4 SRU 26
65535 starting with Oracle Solaris 11.4 SRU 27

**Range**
128 to `MAXINT`

**Units**
File descriptors

**Dynamic?**
No

**Validation**
Compared to `rlim_fd_sys`. If `rlim_fd_max` is greater than `rlim_fd_sys`, `rlim_fd_max` is reset to `rlim_fd_sys`

**When to Change**
When the maximum number of open files for a process is not enough. Other limitations in system facilities can mean that a larger number of file descriptors is not as useful as it might be. For example:
`select` is by default limited to 1024 descriptors per `fd_set` in 32-bit applications. For more information, see the `select`(3C) man page. A 32-bit application code can be recompiled with a larger `fd_set` size (less than or equal to 65,536). A 64-bit application uses an `fd_set` size of 65,536, which cannot be changed.
An alternative to changing this on a system wide basis is to use the `plimit` command. If a parent process has its limits changed by `plimit`, all children inherit the increased limit. This alternative is useful for daemons such as `inetd`.

**Commitment Level**
Unstable

### rlim_fd_sys Parameter

**Description**
Specifies the maximum limit to which a process can raise its hard limit on file descriptors. This parameter specifies the system maximum value for the process.max-file-descriptor resource control.
You cannot override this limit. You can only change this limit if your system runs at least Oracle Solaris 11.4 SRU 27.

**Data Type**
Unsigned integer

**Default**
MAXINT through Oracle Solaris 11.4 SRU 26
Starting with Oracle Solaris 11.4 SRU 27, calculated based on physmem to be approximately 65K per gigabyte of memory. The value is rounded up to the nearest value of (2^*N*)-1.

**Range**
128 to MAXINT

**Dynamic?**
No

**Validation**
Compared to rlim_fd_max. If rlim_fd_sys is less than rlim_fd_max, rlim_fd_sys is reset to rlim_fd_max

**When to Change**
When the maximum hard limit of open files for a process is insufficient or when a system restricts the maximum hard limit for privileged processes

**Commitment Level**
Unstable

## General File System Parameters

This section describes parameters that relate to file systems.

### dnlc_dir_enable Parameter

**Description**
Enables large directory caching

> **Note:**
> This parameter has no effect on NFS or ZFS file systems.

**Data Type**
Unsigned integer

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes, but do not change this tunable dynamically. You can enable this parameter if it was originally disabled. Or, you can disable this parameter if it was originally enabled. However, enabling, disabling, and then enabling this parameter might lead to stale directory caches.

**Validation**
No

**When to Change**
Directory caching has no known problems. However, if problems occur, then set `dnlc_dir_enable` to 0 to disable caching.

**Commitment Level**
Unstable

## `dnlc_dir_max_size` Parameter

**Description**
Specifies the maximum number of entries cached for one directory.

> **Note:**
>
> This parameter has no effect on NFS or ZFS file systems.

**Data Type**
Unsigned integer

**Default**
MAXUINT (no maximum)

**Range**
0 to MAXUINT

**Dynamic?**
Yes, this parameter can be changed at any time.

**Validation**
None

**When to Change**
If performance problems occur with large directories, then decrease `dnlc_dir_max_size`.

**Commitment Level**
Unstable

## dnlc_dir_min_size Parameter

**Description**
Specifies the minimum number of entries cached for one directory.

> **Note:**
>
> This parameter has no effect on NFS or ZFS file systems.

**Data Type**
Unsigned integer

**Default**
40

**Range**
0 to MAXUINT (no maximum)

**Units**
Entries

**Dynamic?**
Yes, this parameter can be changed at any time.

**Validation**
None

**When to Change**
If performance problems occur with caching small directories, then increase `dnlc_dir_min_size`. Note that individual file systems might have their own range limits for caching directories.

**Commitment Level**
Unstable

## dnlc_dircache_percent Parameter

**Description**
Calculates the maximum percentage of physical memory that the DNLC directory cache can consume.

**Data Type**
Integer

**Default**
100

**Range**
0 to 100

**Units**
Percentage

**Dynamic?**
No

**Validation**
At boot time, the value range is checked and default value is enforced.

**When to Change**
When the system experiences a memory shortage and high kernel memory consumption, consider lowering this value. If performance issues are seen with the default value, consider increasing the value.

> **Note:**
>
> The DNLC is used by UFS and ZFS file systems and NFS clients. Setting this tunable might be considered for better performance when there are memory shortages and high kernel memory consumption or when a memory is needed by the ARC or other kernel caches.

**Commitment Level**
Unstable

## `ncsize` Parameter

**Description**
Defines the number of entries in the directory name look-up cache (DNLC). This parameter is used by UFS, NFS, and ZFS to cache elements of path names that have been resolved. The DNLC also caches negative look-up information, which means it caches a name not found in the cache.

**Data Type**
Signed integer

**Default**
(4 x (`v.v_proc` + `maxusers`) + 320) + (4 x (`v.v_proc` + `maxusers`) + 320) / 100

**Range**
0 to `MAXINT`

**Units**
DNLC entries

**Dynamic?**
No

**Validation**
None. Larger values cause the time it takes to unmount a file system to increase as the cache must be flushed of entries for that file system during the unmount process.

**When to Change**
You can use the `kstat -n dnlcstats` command to view the number of hits and misses in the DNLC. A high ratio of misses to hits might indicate that the DNLC is too small. Increasing the value of `ncsize` can help improve performance.

Excessive values of `ncsize` have an immediate impact on the system memory usage. Memory that is used directly by the DNLC increases in proportion to `ncsize`. In addition, caching more entries in the DNLC might require filesystem context relating to those entries to remain in memory when it would otherwise be freed.

**Commitment Level**
Unstable

# TMPFS Parameters

This section describes parameters that affect temporary file storage facility.

## `tmpfs:tmpfs_maxkmem` Parameter

**Description**
Defines the maximum amount of kernel memory that TMPFS can use for its data structures (tmpnodes and directory entries).

**Data Type**
Unsigned long

**Default**
One page or 4 percent of physical memory, whichever is greater.

**Range**
Number of bytes in one page (8192 for sun4v systems, 4096 for all other systems) to 25 percent of the available kernel memory at the time TMPFS was first used.

**Units**
Bytes

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Increase if the following message is displayed on the console or written in the messages file:

```
tmp_memalloc: tmpfs over memory limit
```

The current amount of memory used by TMPFS for its data structures is held in the `tmp_kmemspace` field. This field can be examined with a kernel debugger.

**Commitment Level**
Unstable

## `tmpfs:tmpfs_minfree` Parameter

**Description**
Defines the minimum amount of swap space that TMPFS leaves for the rest of the system.

**Data Type**
Signed long

**Default**
256

**Range**
0 to maximum swap space size

**Units**
Pages

**Dynamic?**
Yes

**Validation**
None

**When to Change**
To maintain a reasonable amount of swap space on systems with large amounts of TMPFS usage, you can increase this number. The limit has been reached when the console or messages file displays the following message:

```
fs-name: File system full, swap space limit exceeded
```

**Commitment Level**
Unstable

# Pseudo Terminals

Pseudo terminals (`ptys`) are used for two purposes in Oracle Solaris software:

- Supporting remote logins
- Providing the interface through which the X Window system creates command interpreter windows

The default number of pseudo terminals is sufficient for a desktop workstation. So, tuning focuses on the number of `ptys` available for remote logins.

The default number of `ptys` is now based on the amount of memory on the system. This default should be changed only to restrict or increase the number of users who can log in to the system.

Three related variables are used in the configuration process:

- `pt_cnt` – Default maximum number of `ptys`.
- `pt_pctofmem` – Percentage of kernel memory that can be dedicated to pseudo terminal support structures. A value of zero means that no remote users can log in to the system.
- `pt_max_pty` – Hard maximum for number of `ptys`.

`pt_cnt` has a default value of zero, which tells the system to limit logins based on the amount of memory specified in `pct_pctofmem`, unless `pt_max_pty` is set. If `pt_cnt` is non-zero, `ptys` are allocated until this limit is reached. When that threshold is crossed, the system looks at `pt_max_pty`. If `pt_max_pty` has a non-zero value, it is compared to `pt_cnt`. The pseudo terminal allocation is allowed if `pt_cnt` is less than `pt_max_pty`. If `pt_max_pty` is zero, `pt_cnt`

is compared to the number of ptys supported based on pt_pctofmem. If pt_cnt is less than this value, the pseudo terminal allocation is allowed. Note that the limit based on pt_pctofmem only comes into play if both pt_cnt and ptms_ptymax have default values of zero.

To put a hard limit on ptys that is different than the maximum derived from pt_pctofmem, set pt_cnt and ptms_ptymax in /etc/system.d*file* to the preferred number of ptys. The setting of ptms_pctofmem is not relevant in this case.

To dedicate a different percentage of system memory to pseudo terminal support and let the operating system manage the explicit limits, do the following:

- Do not set pt_cnt or ptms_ptymax in /etc/system.d/*file* .
- Set pt_pctofmem in /etc/system.d/*file* to the preferred percentage. For example, set pt_pctofmem=10 for a 10 percent setting.

Note that the memory is not actually allocated until it is used in support of a pseudo terminal. Once memory is allocated, it remains allocated.

## pt_cnt Parameter

**Description**
The number of available /dev/pts entries is dynamic up to a limit determined by the amount of physical memory available on the system. pt_cnt is one of three variables that determines the minimum number of logins that the system can accommodate. The default maximum number of /dev/pts devices the system can support is determined at boot time by computing the number of pseudo terminal structures that can fit in a percentage of system memory (see pt_pctofmem). If pt_cnt is zero, the system allocates up to that maximum. If pt_cnt is non-zero, the system allocates to the greater of pt_cnt and the default maximum.

**Data Type**
Unsigned integer

**Default**
0

**Range**
0 to maxpid

**Units**
Logins/windows

**Dynamic?**
No

**Validation**
None

**When to Change**
When you want to explicitly control the number of users who can remotely log in to the system.

**Commitment Level**
Unstable

## pt_max_pty Parameter

**Description**
Defines the maximum number of ptys the system offers

**Data Type**
Unsigned integer

**Default**
0 (Uses system-defined maximum)

**Range**
0 to MAXUINT

**Units**
Logins/windows

**Dynamic?**
Yes

**Validation**
None

**Implicit**
Should be greater than or equal to pt_cnt. Value is not checked until the number of ptys allocated exceeds the value of pt_cnt.

**When to Change**
When you want to place an absolute ceiling on the number of logins supported, even if the system could handle more based on its current configuration values.

**Commitment Level**
Unstable

## pt_pctofmem Parameter

**Description**
Specifies the maximum percentage of physical memory that can be consumed by data structures to support /dev/pts entries. A system consumes 176 bytes per /dev/pts entry.

**Data Type**
Unsigned integer

**Default**
5

**Range**
0 to 100

**Units**
Percentage

**Dynamic?**
No

**Validation**
None

**When to Change**
When you want to either restrict or increase the number of users who can log in to the system. A value of zero means that no remote users can log in to the system.

**Commitment Level**
Unstable

# STREAMS Parameters

This section describes STREAMS-related parameters.

## `nstrpush` Parameter

**Description**
Specifies the number of modules that can be inserted into (pushed onto) a STREAM.

**Data Type**
Signed integer

**Default**
9

**Range**
9 to 16

**Units**
Modules

**Dynamic?**
Yes

**Validation**
None

**When to Change**
At the direction of your software vendor. No messages are displayed when a STREAM exceeds its permitted push count. A value of `EINVAL` is returned to the program that attempted the push.

**Commitment Level**
Unstable

## `strmsgsz` Parameter

**Description**
Specifies the maximum number of bytes that a single system call can pass to a STREAM to be placed in the data part of a message. Any `write` exceeding this size is broken into multiple messages. For more information, see the `write`(2) man page.

**Data Type**
Signed integer

**Default**
65,536

**Range**
0 to 262,144

**Units**
Bytes

**Dynamic?**
Yes

**Validation**
None

**When to Change**
When `putmsg` calls return `ERANGE`. For more information, see the `putmsg`(2) man page.

**Commitment Level**
Unstable

## `strctlsz` Parameter

**Description**
Specifies the maximum number of bytes that a single system call can pass to a STREAM to be placed in the control part of a message

**Data Type**
Signed integer

**Default**
1024

**Range**
0 to MAXINT

**Units**
Bytes

**Dynamic?**
Yes

**Validation**
None

**When to Change**
At the direction of your software vendor. `putmsg`calls return `ERANGE` if they attempt to exceed this limit.

**Commitment Level**
Unstable

# System V Message Queues

System V message queues provide a message-passing interface that enables the exchange of messages by queues created in the kernel. Interfaces are provided in the Oracle Solaris

environment to enqueue and dequeue messages. Messages can have a type associated with them. Enqueueing places messages at the end of a queue. Dequeuing removes the first message of a specific type from the queue or the first message if no type is specified.

For detailed information about tuning these system resources, see Chapter 6, About Resource Controls in *Administering Resource Management in Oracle Solaris 11.4*.

# System V Semaphores

System V semaphores provide counting semaphores in the Oracle Solaris OS. A *semaphore* is a counter used to provide access to a shared data object for multiple processes. In addition to the standard set and release operations for semaphores, System V semaphores can have values that are incremented and decremented as needed (for example, to represent the number of resources available). System V semaphores also provide the ability to do operations on a group of semaphores simultaneously as well as to have the system undo the last operation by a process if the process dies.

# Timer Behavior

This section describes parameters that determine timer behavior.

## `hires_tick` Parameter

**Description**
When set, this parameter causes the Oracle Solaris OS to use a system clock rate of 1000 instead of the default value of 100.

**Data Type**
Signed integer

**Default**
0

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
No. Causes new system timing variable to be set at boot time. Not referenced after boot.

**Validation**
None

**When to Change**
When you want timeouts with a resolution of less than 10 milliseconds, and greater than or equal to 1 millisecond.

**Commitment Level**
Unstable

## timer_max Parameter

**Description**
Specifies the number of POSIX™ timers available.

**Data Type**
Signed integer

**Default**
1000

**Range**
0 to MAXINT

**Dynamic?**
No. Increasing the value can cause a system crash.

**Validation**
None

**When to Change**
When the default number of timers offered by the system is inadequate. Applications receive an EAGAIN error when executing timer_create system calls.

**Commitment Level**
Unstable

# SPARC: Platform Specific Parameters

The following parameters apply to sun4v platforms.

## default_tsb_size Parameter

**Description**
Selects size of the initial translation storage buffers (TSBs) allocated to all processes.

**Data Type**
Integer

**Default**
Default is 0 (8 KB), which corresponds to 512 entries

**Range**
Possible values are:

| Value | Description |
|---|---|
| 0 | 8 KB |
| 1 | 16 KB |
| 3 | 32 KB |
| 4 | 128 KB |
| 5 | 256 KB |

| Value | Description |
|---|---|
| 6 | 512 KB |
| 7 | 1 MB |

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Generally, you do not need to change this value. However, doing so might provide some advantages if the majority of processes on the system have a larger than average working set, or if resident set size (RSS) sizing is disabled.

**Commitment Level**
Unstable

## `enable_tsb_rss_sizing` Parameter

**Description**
Enables a resident set size (RSS) based TSB sizing heuristic.

**Data Type**
Boolean

**Default**
1 (TSBs can be resized)

**Range**
0 (TSBs remain at `tsb_default_size`) or 1 (TSBs can be resized)
If set to 0, then `tsb_rss_factor` is ignored.

**Dynamic?**
Yes

**Validation**
Yes

**When to Change**
Can be set to 0 to prevent growth of the TSBs. Under most circumstances, this parameter should be left at the default setting.

**Commitment Level**
Unstable

## `tsb_alloc_hiwater_factor` Parameter

**Description**
Initializes `tsb_alloc_hiwater` to impose an upper limit on the amount of physical memory that can be allocated for translation storage buffers (TSBs) as follows: `tsb_alloc_hiwater` = physical memory (bytes) / `tsb_alloc_hiwater_factor`

When the memory that is allocated to TSBs is equal to the value of `tsb_alloc_hiwater`, the TSB memory allocation algorithm attempts to reclaim TSB memory as pages are unmapped. Exercise caution when using this factor to increase the value of `tsb_alloc_hiwater`. To prevent system hangs, the resulting high water value must be considerably lower than the value of `swapfs_minfree` and `segspt_minfree`.

**Data Type**
Integer

**Default**
32

**Range**
1 to MAXINIT
Note that a factor of 1 makes all physical memory available for allocation to TSBs, which could cause the system to hang. A factor that is too high will not leave memory available for allocation to TSBs, decreasing system performance.

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Change the value of this parameter if the system has many processes that attach to very large shared memory segments. Under most circumstances, tuning of this variable is not necessary.

**Commitment Level**
Unstable

## `tsb_rss_factor` Parameter

**Description**
Controls the RSS to TSB span ratio of the RSS sizing heuristic. This factor divided by 512 yields the percentage of the TSB span which must be resident in memory before the TSB is considered as a candidate for resizing.

**Data Type**
Integer

**Default**
384, resulting in a value of 75%. Thus, when the TSB is 3/4 full, its size will be increased. Note that some virtual addresses typically map to the same slot in the TSB. Therefore, conflicts can occur before the TSB is at 100% full.

**Range**
0 to 512

**Dynamic?**
Yes

**Validation**
None

**When to Change**
If the system is experiencing an excessive number of traps due to TSB misses, for example, due to virtual address conflicts in the TSB, you might consider decreasing this value toward 0.
For example, changing `tsb_rss_factor` to 256 (effectively, 50%) instead of 384 (effectively, 75%) might help eliminate virtual address conflicts in the TSB in some cases, but will use more kernel memory, particularly on a heavily loaded system. TSB activity can be monitored with the `trapstat` -T command.

**Commitment Level**
Unstable

# Locality Group Parameters

This section provides generic memory tunables, which apply to any SPARC or x86 system that uses a Non-Uniform Memory Architecture (NUMA).

## `lgrp_mem_pset_aware` Parameter

**Description**
If a process is running within a user processor set, this variable determines whether *randomly* placed memory for the process is selected from among all the lgroups in the system or only from those lgroups that are spanned by the processors in the processor set.
For more information about creating processor sets, see the `psrset`(8) man page.

**Data Type**
Boolean

**Default**
0, the Oracle Solaris OS selects memory from all the lgroups in the system

**Range**

- 0, the Oracle Solaris OS selects memory from all the lgroups in the system (default)
- 1, try selecting memory only from those lgroups that are spanned by the processors in the processor set. If the first attempt fails, memory can be allocated in any lgroup.

**Dynamic?**
No

**Validation**
None

**When to Change**
Setting this value to a value of one (1) might lead to more reproducible performance when processor sets are used to isolate applications from one another.

**Commitment Level**
Uncommitted

# `lpg_alloc_prefer` Parameter

**Description**
Controls a heuristic for allocation of large memory pages when the requested page size is not immediately available in the local memory group, but could be satisfied from a remote memory group.
By default, the Oracle Solaris OS allocates a remote large page if local free memory is fragmented, but remote free memory is not. Setting this parameter to 1 indicates that additional effort should be spent attempting to allocate larger memory pages locally, potentially moving smaller pages around to coalesce larger pages in the local memory group.

**Data Type**
Boolean

**Default**
0 (Prefer remote allocation if local free memory is fragmented and remote free memory is not)

**Range**
0 (Prefer remote allocation if local free memory is fragmented and remote free memory is not)
1 (Prefer local allocation whenever possible, even if local free memory is fragmented and remote free memory is not)

**Dynamic?**
No

**Validation**
None

**When to Change**
This parameter might be set to 1 if long-running programs on the system tend to allocate memory that is accessed by a single program, or if memory that is accessed by a group of programs is known to be running in the same locality group (lgroup). In these circumstances, the extra cost of page coalesce operations can be amortized over the long run of the programs.
This parameter might be left at the default value (0) if multiple programs tend to share memory across different locality groups, or if pages tend to be used for short periods of time. In these circumstances, quick allocation of the requested size tends to be more important than allocation in a particular location.
TLB miss activity might be observed by using the `trapstat` -T command.

**Commitment Level**
Uncommitted

3

# Oracle Solaris ZFS Tunable Parameters

This chapter describes ZFS tunable parameters that might need consideration, depending on your system and application requirements. In addition, tunable recommendations for using ZFS with database products are provided.

- Tuning ZFS Considerations
- ZFS Memory Management Parameters
- ZFS File-Level Prefetch
- ZFS Device I/O Queue Depth
- Tuning ZFS When Using Flash Storage
- Tuning ZFS for Database Products

For other types of tunable parameters, refer to the following:

- Oracle Solaris kernel tunable parameters – Oracle Solaris Kernel Tunable Parameters
- NFS tunable parameters – NFS Tunable Parameters
- Internet Protocol Suite tunable parameters – Internet Protocol Suite Tunable Parameters
- System facility tunable parameters – System Facility Parameters

## Tuning ZFS Considerations

Review the following considerations before tuning ZFS:

- Default values are generally the best value. If a better value exists, it should be the default. While alternative values might help a given workload, it could quite possibly degrade some other aspects of performance. Occasionally, catastrophically so.
- The ZFS best practices should be followed before ZFS tuning is applied. These practices are a set of recommendations that have been shown to work in different environments and are expected to keep working in the foreseeable future. So, before turning to tuning, make sure you've read and understood the best practices. For more information, see Chapter 12, Recommended Oracle Solaris ZFS Practices in *Managing ZFS File Systems in Oracle Solaris 11.4*.
- Unless noted otherwise, the tunable parameters are global and impact ZFS behavior across the system.

## ZFS Memory Management Parameters

This section describes parameters related to ZFS memory management.

# `user_reserve_hint_pct` ZFS Parameter

**Description**
Informs the system about how much memory is reserved for application use, and therefore limits how much memory can be used by the ZFS ARC cache as the cache increases over time.
By means of this parameter, administrators can maintain a large reserve of available free memory for future application demands. The `user_reserve_hint_pct` parameter is intended to be used in place of the zfs_arc_max Parameter parameter to restrict the growth of the ZFS ARC cache.

> **Note:**
>
> Review Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.x*, in My Oracle Support (MOS) for guidance in tuning this parameter.

**Data Type**
Unsigned Integer (64-bit)

**Default**
0
If a dedicated system is used to run a set of applications with a known memory footprint, set the parameter to the value of that footprint, such as the sum of the SGA of Oracle database.
To assign a value to the parameter, run the script that is provided in Document 1663862.1 in My Oracle Support (MOS). To make the tuning persistent across reboots, refer to script output for instructions about using -p option.

**Range**
0-99

**Units**
Percent

**Dynamic**
Yes
You can adjust the setting of this parameter dynamically on a running system.

**When to Change**
For upward adjustments, increase the value if the initial value is determined to be insufficient over time for application requirements, or if application demand increases on the system. Perform this adjustment only within a scheduled system maintenance window. After you have changed the value, reboot the system.
For downward adjustments, decrease the value if allowed by application requirements. Make sure to use decrease the value only by small amounts, no greater than 5% at a time.

**Commitment Level**
Unstable

## zfs_arc_min Parameter

**Description**
Determines the minimum size of the ZFS Adaptive Replacement Cache (ARC). See also zfs_arc_max Parameter.

**Data Type**
Unsigned Integer (64-bit)

**Default**
1% of total memory (physmem)

**Range**
Default value to zfs_arc_max

**Units**
Bytes

**Dynamic?**
No

**Validation**
Yes, the range is validated.

**When to Change**
When a system's workload demand for memory fluctuates, the ZFS ARC caches data at a period of weak demand and then shrinks at a period of strong demand. However, ZFS does not shrink below the value of zfs_arc_min. Generally, you do not need to change the default value.

**Commitment Level**
Unstable

## zfs_arc_max Parameter

**Description**
Determines the maximum size of the ZFS Adaptive Replacement Cache (ARC). However, see user_reserve_hint_pct ZFS Parameter.
See also zfs_arc_min Parameter.

**Data Type**
Unsigned Integer (64-bit)

**Default**
75% of memory on systems with less than 4 GB of memory
physmem minus 1 GB on systems with greater than 4 GB of memory

**Range**
Default value of zfs_arc_min to physmem

**Units**
Bytes

**Dynamic?**
No

**Validation**
Yes, the range is validated.

**When to Change**
If a future memory requirement is significantly large and well defined, you might consider reducing the value of this parameter to cap the ARC so that it does not compete with the memory requirement. For example, if you know that a future workload requires 20% of memory, it makes sense to cap the ARC such that it does not consume more than the remaining 80% of memory.

**Commitment Level**
Unstable

### `zfs_arc_max_percent` Parameter

**Description**
Determines the maximum size of the ZFS Adaptive Replacement Cache (ARC) as a percentage of total memory. See also user_reserve_hint_pct ZFS Parameter.

**Data Type**
Integer

**Default**
90% of total memory. ZFS only grows to the point where ZFS sees memory pressure from applications and from other kernel demands.

**Range**
Percentage from 0 to 100

**Units**
Percentage

**Dynamic?**
Yes

**Validation**
Yes, the range is validated.

**When to Change**
Setting `zfs_arc_max_percent` to 70% can help reduce disruptive events such as a large ARC reduction under memory pressure along with multi-second periods of no I/Os. Note that this smaller ARC might lead to additional cache misses and more IOPS requests to the storage devices. However, this result might be an acceptable price in exchange for the extra performance stability that a smaller ARC provides.

**Commitment Level**
Unstable

## ZFS File-Level Prefetch

This section describes the parameter that regulates the behavior of the prefetching mechanism.

## zfs_prefetch_disable Parameter

**Description**
This parameter determines a file-level prefetching mechanism called zfetch. This mechanism looks at the patterns of reads to files and anticipates on some reads, thereby reducing application wait times. The current behavior suffers from two drawbacks:

- Sequential read patterns made of small reads very often hit in the cache. In this case, the current behavior consumes a significant amount of CPU time trying to find the next I/O to issue, whereas performance is governed more by the CPU availability.
- The zfetch code has been observed to limit scalability of some loads. CPU profiling can be done by using the lockstat -I command or er_kernel. For detailed descriptions, see the following sources:
  - lockstat(8)
  - Oracle Developer Studio documentation (https://docs.oracle.com/en/operating-systems/studio.html)
  - Oracle Developer Studio Documentation (https://www.oracle.com/application-development/technologies/developerstudio-documentation.html)

  Although never recommended, you can disable prefetching by setting zfs_prefetch_disable in a file the /etc/system.d directory. For instructions, see /etc/system.d/ Directory Files.

  Device-level prefetching is disabled when zfs_vdev_cache_size is disabled. This means that tuning vdev cache shift is no longer necessary if zfs_vdev_cache_size is disabled.

**Data Type**
Boolean

**Default**
0 (enabled)

**Range**
0 (enabled) or 1 (disabled)

**Dynamic?**
Yes

**Validation**
No

**When to Change**
If the results of er_kernel show significant time in zfetch_* functions, or if lock profiling with lockstat shows contention around zfetch locks, and if the affected workload is critical, then disabling file level prefetching should be considered. Other workloads that are running concurrently can be degraded.

**Commitment Level**
Unstable

# ZFS Device I/O Queue Depth

This section describes the parameter pertaining to concurrent I/O processes for ZFS.

## `zfs_vdev_max_pending` Parameter

**Description**
This parameter controls the maximum number of concurrent I/Os pending to each device.

**Data Type**
Integer

**Default**
10

**Range**
0 to MAXINT

**Dynamic?**
Yes

**Validation**
No

**When to Change**
In a storage array where LUNs are made of a large number of disk drives, the ZFS queue can become a limiting factor on read IOPS. This behavior is one of the underlying reasoning for the best practice of presenting as many LUNS as there are backing spindles to the ZFS storage pool. That is, if you create LUNS from a 10 disk-wide array level raid-group, then using 5 to 10 LUNs to build a storage pool allows ZFS to manage enough of an I/O queue without the need to set this specific tunable. However, when no separate intent log is in use and the pool is made of JBOD disks, using a small `zfs_vdev_max_pending` value, such as 10, can improve the synchronous write latency as those are competing for the disk resource. Using separate intent log devices can alleviate the need to tune this parameter for loads that are synchronously write intensive since those synchronous writes are not competing with a deep queue of non-synchronous writes.
Tuning this parameter is not expected to be effective for NVRAM-based storage arrays in the case where volumes are made of small number of spindles. However, when ZFS is presented with a volume made of a large (greater than 10) number of spindles, then this parameter can limit the read throughput obtained on the volume. The reason is that with a maximum of 10 or 35 queued I/Os per LUN, this can translate into less than 1 I/O per storage spindle, which is not enough for individual disks to deliver their IOPS. This issue would appear in `iostat actv` queue output approaching the value of `zfs_vdev_max_pending`.
Device drivers may also limit the number of outstanding I/Os per LUN. If you are using LUNs on storage arrays that can handle large numbers of concurrent IOPS, then the device driver constraints can limit concurrency. Consult the configuration for the drivers your system uses. For example, the limit for the QLogic FCl HBA (qlc) driver is described as the execution-throttle parameter in `/kernel/drv/qlc.conf`.

**Commitment Level**
Unstable

# Tuning ZFS When Using Flash Storage

The following information applies to Flash SSDs, F20 PCIe Accelerator Card, F40 PCIe Accelerator Card, F5100 Flash Storage Array, and F80 PCIe Accelerator Card.

Review the following general comments when using ZFS with Flash storage:

- Consider using LUNs or low latency disks that are managed by a controller with persistent memory, if available, for the ZIL (ZFS intent log). This option can be considerably more cost effective than using flash for low latency commits. The size of the log devices must only be large enough to hold 10 seconds of maximum write throughput. Examples would include a storage array based LUN, or a disk connected to an HBA with a battery protected write cache.

  If no such device is available, segment a separate pool of flash devices for use as log devices in a ZFS storage pool.
- The F40, F20, and F80 Flash Accelerator cards contain and export 4 independent flash modules to the OS. The F5100 contains up to 80 independent flash modules. Each flash module appear to the operating system as a single device. SSDs are viewed as a single device by the OS. Flash devices may be used as ZFS log devices to reduce commit latency, particularly if used in an NFS server. For example, a single flash module of a flash device used as a ZFS log device can reduce latency of single lightly threaded operations by 10x. More flash devices can be striped together to achieve higher throughput for large amounts of synchronous operations.
- Log devices should be mirrored for reliability. For maximum protection, the mirrors should be created on separate flash devices. In the case of F20, F40, and F80 PCIe accelerator cards, maximum protection is achieved by ensuring that mirrors reside on different physical PCIe cards. Maximum protection with the F5100 storage array is obtained by placing mirrors on separate F5100 devices.
- Flash devices that are not used as log devices may be used as second level cache devices. This serves to both offload IOPS from primary disk storage and also to improve read latency for commonly used data.

## Adding Flash Devices as ZFS Log or Cache Devices

Review the following recommendations when adding flash devices as ZFS log or cache devices.

- A ZFS log or cache device can be added to an existing ZFS storage pool by using the `zpool add` command. Be very careful with `zpool add` commands. Mistakenly adding a log device as a normal pool device is a mistake that will require you to destroy and restore the pool from scratch. Individual log devices themselves can be removed from a pool.
- Familiarize yourself with the `zpool add` command before attempting this operation on active storage. You can use the `zpool add -n` option to preview the configuration without creating the configuration. For example, the following incorrect `zpool add` preview syntax attempts to add a device as a log device:

  ```
  # zpool add -n tank c4t1d0
  vdev verification failed: use -f to override the following errors:
  ```

```
mismatched replication level: pool uses mirror and new vdev is disk
Unable to build pool from specified devices: invalid vdev configuration
```

This is the correct `zpool add` preview syntax for adding a log device to an existing pool:

```
# zpool add -n tank log c4t1d0
would update 'tank' to the following configuration:
tank
mirror
c4t0d0
c5t0d0
logs
c4t1d0
```

If multiple devices are specified, they are striped together. For more information, see the examples below or the `zpool`(8) man page.

A flash device, `c4t1d0`, can be added as a ZFS log device:

```
# zpool add pool log c4t1d0
```

If 2 flash devices are available, you can add mirrored log devices:

```
# zpool add pool log mirror c4t1d0 c4t2d0
```

Available flash devices can be added as a cache device for reads.

```
# zpool add pool cache c4t3d0
```

You can't mirror cache devices, they will be striped together.

```
# zpool add pool cache c4t3d0 c4t4d0
```

# Ensuring Proper Cache Flush Behavior for Flash and NVRAM Storage Devices

ZFS is designed to work with storage devices that manage a disk-level cache. ZFS commonly asks the storage device to ensure that data is safely placed on stable storage by requesting a cache flush. For JBOD storage, this works as designed and without problems. For many NVRAM-based storage arrays, a performance problem might occur if the array takes the cache flush request and actually does something with it, rather than ignoring it. Some storage arrays flush their large caches despite the fact that the NVRAM protection makes those caches as good as stable storage.

ZFS issues infrequent flushes (every 5 second or so) after the uberblock updates. The flushing infrequency is fairly inconsequential so no tuning is warranted here. ZFS also issues a flush every time an application requests a synchronous write (`O_DSYNC`, `fsync`, NFS commit, and so on). The completion of this type of flush is waited upon by the application and impacts performance. Greatly so, in fact. From a performance standpoint, this neutralizes the benefits of having an NVRAM-based storage.

Cache flush tuning was recently shown to help flash device performance when used as log devices. When all LUNs exposed to ZFS come from NVRAM-protected storage

array and procedures ensure that no unprotected LUNs will be added in the future, ZFS can be tuned to not issue the flush requests by setting `zfs_nocacheflush`. If some LUNs exposed to ZFS are not protected by NVRAM, then this tuning can lead to data loss, application level corruption, or even pool corruption. In some NVRAM-protected storage arrays, the cache flush command is a no-op, so tuning in this situation makes no performance difference.

A recent OS change is that the flush request semantic has been qualified to instruct storage devices to ignore the requests if they have the proper protection. This change requires a fix to our disk drivers and for the NVRAM device to support the updated semantics. If the NVRAM device does not recognize this improvement, use these instructions to tell the Oracle Solaris OS not to send any synchronize cache commands to the array. If you use these instructions, make sure all targeted LUNS are indeed protected by NVRAM.

Occasionally, flash and NVRAM devices do not properly advertise to the OS that they are non-volatile devices, and that caches do not need to be flushed. Cache flushing is an expensive operation. Unnecessary flushes can drastically impede performance in some cases.

Review the following `zfs_nocacheflush` syntax restrictions before applying the tuning entries below:

- The tuning syntax below can be included in `sd.conf` but there must be only a single `sd-config-list` entry per vendor/product.
- If multiple devices entries are desired, multiple pairs of vendor IDs and `sd` tuning strings can be specified on the same line by using the following syntax:

```
#              "012345670123456789012345","tuning    ",
sd-config-list="|-VID1-||-----PID1-----|","param1:val1, param2:val2",
               "|-VIDN-||-----PIDN-----|","param1:val1, param3:val3";
```

  Make sure the vendor ID (VID) string is padded to 8 characters and the Product ID (PID) string is padded to 16 characters as described in the preceding example.

> **Caution:**
>
> All cache sync commands are ignored by the device. Use at your own risk.

1. Use the `format` utility to run the `inquiry` subcommand on a LUN from the storage array. For example:

```
# format
.
.
.
Specify disk (enter its number): x
format> inquiry
Vendor:   ATA
Product:  Marvell
Revision: XXXX
format>
```

2. Select one of the following based on your architecture:
   - For all devices, copy the file `/kernel/drv/sd.conf` to the `/etc/driver/drv/sd.conf` file.

- For F40 flash devices, add the following entry to `/kernel/drv/sd.conf`. In the entry below, ensure that `ATA` is padded to 8 characters, and `3E128-TS2-550B01` contains 16 characters. Total string length is 24.

  ```
  sd-config-list="ATA     3E128-TS2-550B01","disksort:false, cache-nonvolatile:true, physical-block-size:4096";
  ```

- For F80 flash devices, add the following entry to `/kernel/drv/sd.conf`. Ensure that `ATA` is padded to 8 characters, and `3E128-TS2-550B01` contains 16 characters. Total string length is 24.

  ```
  sd-config-list="ATA     2E256-TU2-510B00","disksort:false, cache-nonvolatile:true, physical-block-size:4096";
  ```

- For F20 and F5100 flash devices, choose one of the following based on your architecture. In the entries below, `ATA` is padded to 8 characters, and `MARVELL SD88SA02` contains 16 characters. The total string length is 24.

- Add the following entry to `/etc/driver/drv/sd.conf`

  ```
  sd-config-list="ATA     MARVELL SD88SA02","throttle-max:32, disksort:false, cache-nonvolatile:true";
  ```

3. Carefully add whitespace to make the vendor ID (VID) 8 characters long (here "`ATA `") and Product ID (PID) 16 characters long (here `MARVELL`) in the `sd-config-list` entry as illustrated.
4. Reboot the system.

   You can tune `zfs_nocacheflush` back to it's default value (0) with no adverse effect on performance.
5. Confirm that the flush behavior is correct.

   Use the script provided in System Check Script for verification.

# Tuning ZFS for Database Products

Review the following considerations when using ZFS with a database product.

- ZFS checksums every block stored on disk. This alleviates the need for the database layer to checksum data an additional time. If checksums are computed by ZFS instead of at the database layer, any discrepancy can be caught and fixed before the data is returned to the application.
- Keep pool space under 90% utilization to maintain pool performance.

For a comprehensive information about best practices when using ZFS Storage in an Exadata environment, log in to your My Oracle Support account (https://support.oracle.com) and refer to the following documents:

- Doc ID 2087231.1
- Doc ID 1354980.1

To tune ZFS for an Oracle Database, see Configuring Oracle Solaris ZFS for an Oracle Database (https://www.oracle.com/technetwork/server-storage/solaris/config-solaris-zfs-wp-167894.pdf).

Review the following considerations when using ZFS with MySQL.

- **ZFS recordsize**

Match the ZFS `recordsize` property to the storage engine block size for better OLTP performance.

- **InnoDB**

  With a known application memory footprint, such as for a database application, you might cap the ARC size so that the application will not need to reclaim its necessary memory from the ZFS cache.

- Create a separate pool for the logs.
- Set a different path for data and log in the `my.cnf` file.
- Set the ZFS `recordsize` property to 16K for the InnoDB data files, and use the default `recordsize` value for InnoDB logs, prior to creating data files.

# 4
# NFS Tunable Parameters

This section describes the NFS tunable parameters.

- Tuning the NFS Environment
- NFS Module Parameters
- NFS-Related SMF Configuration Parameters
- rpcmod Module Parameters

For other types of tunable parameters, refer to the following:

- Oracle Solaris kernel tunable parameters – Oracle Solaris Kernel Tunable Parameters
- Oracle Solaris ZFS tunable parameters – Oracle Solaris ZFS Tunable Parameters
- Internet Protocol Suite tunable parameters – Internet Protocol Suite Tunable Parameters
- System facility tunable parameters – System Facility Parameters

## Tuning the NFS Environment

You can define NFS parameters in files in the `/etc/system.d` directory, which are read during the boot process. Each parameter includes the name of its associated kernel module. For more information, see Tuning an Oracle Solaris System.

> **Caution:**
>
> The names of the parameters, the modules that they reside in, and the default values can change between releases. Check the documentation for the version of the active SunOS release before making changes or applying values from previous releases.

## NFS Module Parameters

This section describes parameters related to the NFS kernel module.

### `nfs:nfs_allow_preepoch_time` Parameter

**Description**
Controls whether files with incorrect or *negative* time stamps should be made visible on the NFS client.
Historically, neither the NFS client nor the NFS server would do any range checking on the file times being returned. The over-the-wire timestamp values are unsigned and 32-bits long. So, all values have been legal.

The timestamp values on the 64-bit Oracle Solaris kernel are signed and 64-bits long. It is impossible to determine whether a time field represents a full 32-bit time or a negative time, that is, a time prior to January 1, 1970.
It is impossible to determine whether to sign extend a time value when converting from 32 bits to 64 bits. The time value should be sign extended if the time value is truly a negative number. However, the time value should not be sign extended if it does truly represent a full 32-bit time value. This problem is resolved by simply disallowing full 32-bit time values.

**Data Type**
Integer (32-bit)

**Default**
0 (32-bit time stamps disabled)

**Range**
0 (32-bit time stamps disabled) or 1 (32-bit time stamps enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Even during normal operation, it is possible for the timestamp values on some files to be set very far in the future or very far in the past. If access to these files is preferred using NFS mounted file systems, set this parameter to 1 to allow the timestamp values to be passed through unchecked.

**Commitment Level**
Unstable

## `nfs:nfs_async_clusters` Parameter

**Description**
Controls the mix of asynchronous requests that are generated by the NFS version 2 client. The four types of asynchronous requests are read-ahead, putpage, pageio, and readdir-ahead. The client attempts to round-robin between these different request types to attempt to be fair and not starve one request type in favor of another. However, the functionality in some NFS version 2 servers such as write gathering depends upon certain behaviors of existing NFS Version 2 clients. In particular, this functionality depends upon the client sending out multiple WRITE requests at about the same time. If one request is taken out of the queue at a time, the client would be defeating this server functionality designed to enhance performance for the client. Thus, use this parameter to control the number of requests of each request type that are sent out before changing types.

**Data Type**
Unsigned integer (32-bit)

**Default**
1

**Range**
0 to $2^{32}$ - 1

**Units**
Asynchronous requests

**Dynamic?**
Yes, but the cluster setting for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the value of this parameter to 0 causes all of the queued requests of a particular request type to be processed before moving on to the next type. This effectively disables the fairness portion of the algorithm.

**When to Change**
To increase the number of each type of asynchronous request that is generated before switching to the next type. Doing so might help with NFS server functionality that depends upon clusters of requests coming from the NFS client.

**Commitment Level**
Unstable

## `nfs:nfs3_async_clusters` Parameter

**Description**
Controls the mix of asynchronous requests that are generated by the NFS version 3 client. The five types of asynchronous requests are read-ahead, putpage, pageio, readdir-ahead, and commit. The client attempts to round-robin between these different request types to attempt to be fair and not starve one request type in favor of another.
However, the functionality in some NFS version 3 servers such as write gathering depends upon certain behaviors of existing NFS version 3 clients. In particular, this functionality depends upon the client sending out multiple WRITE requests at about the same time. If one request is taken out of the queue at a time, the client would be defeating this server functionality designed to enhance performance for the client.
Thus, use this parameter to control the number of requests of each request type that are sent out before changing types.

**Data Type**
Unsigned integer (32-bit)

**Default**
1

**Range**
0 to $2^{32}$ - 1

**Units**
Asynchronous requests

**Dynamic?**
Yes, but the cluster setting for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the value of this parameter to 0 causes all of the queued requests of a particular request type to be processed before moving on to the next type. This value effectively disables the fairness portion of the algorithm.

**When to Change**
To increase the number of each type of asynchronous operation that is generated before switching to the next type. Doing so might help with NFS server functionality that depends upon clusters of operations coming from the NFS client.

**Commitment Level**
Unstable

## `nfs:nfs4_async_clusters` Parameter

**Description**
Controls the mix of asynchronous requests that are generated by the NFS version 4 client. The six types of asynchronous requests are read-ahead, putpage, pageio, readdir-ahead, commit, and inactive. The client attempts to round-robin between these different request types to attempt to be fair and not starve one request type in favor of another.
However, the functionality in some NFS version 4 servers such as write gathering depends upon certain behaviors of existing NFS version 4 clients. In particular, this functionality depends upon the client sending out multiple WRITE requests at about the same time. If one request is taken out of the queue at a time, the client would be defeating this server functionality designed to enhance performance for the client. Thus, use this parameter to control the number of requests of each request type that are sent out before changing types.

**Data Type**
Unsigned integer (32-bit)

**Default**
1

**Range**
0 to $2^{32}$ - 1

**Units**
Asynchronous requests

**Dynamic?**
Yes, but the cluster setting for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the value of this parameter to 0 causes all of the queued requests of a particular request type to be processed before moving on to the next type. This effectively disables the fairness portion of the algorithm.

**When to Change**
To increase the number of each type of asynchronous request that is generated before switching to the next type. Doing so might help with NFS server functionality that depends upon clusters of requests coming from the NFS client.

**Commitment Level**
Unstable

## `nfs:nfs_async_timeout` Parameter

**Description**
Controls the duration of time that threads, which execute asynchronous I/O requests, sleep with nothing to do before exiting. When there are no more requests to execute, each thread goes to sleep. If no new requests come in before this timer expires, the thread wakes up and exits. If a request does arrive, a thread is woken up to execute requests until there are none again. Then, the thread goes back to sleep waiting for another request to arrive, or for the timer to expire.

**Data Type**
Integer (32-bit)

**Default**
6000 (1 minute expressed as 60 sec * 100Hz)

**Range**
0 to $2^{31}$ - 1

**Units**
Hz. (Typically, the clock runs at 100Hz.)

**Dynamic?**
Yes

**Validation**
None. However, setting this parameter to a non positive value causes these threads exit as soon as there are no requests in the queue for them to process.

**When to Change**
If the behavior of applications in the system is known precisely and the rate of asynchronous I/O requests can be predicted, it might be possible to tune this parameter to optimize performance slightly in either of the following ways:

- By making the threads expire more quickly, thus freeing up kernel resources more quickly
- By making the threads expire more slowly, thus avoiding thread create and destroy overhead

**Commitment Level**
Unstable

## nfs:nfs3_bsize Parameter

**Description**
Controls the logical block size used by the NFS version 3 client. This block size represents the amount of data that the client attempts to read from or write to the NFS server when it needs to do an I/O.

**Data Type**
Unsigned integer (32-bit)

**Default**
32,768 (32 KB)

**Range**
0 to $2^{32}$ - 1

**Units**
Bytes

**Dynamic?**
Yes, but the block size for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. Setting this parameter too low or too high might cause the system to malfunction. Do not set this parameter to anything less than `PAGESIZE` for the specific platform. Do not set this parameter too high because it might cause the system to hang while waiting for memory allocations to be granted.

**When to Change**
Examine the value of this parameter when attempting to change the maximum data transfer size. Change this parameter in conjunction with `nfs:nfs3_max_transfer_size` parameter. If larger transfers are preferred, increase both parameters. If smaller transfers are preferred, then just reducing this parameter should suffice.

**Commitment Level**
Unstable

## nfs:nfs4_bsize Parameter

**Description**
Controls the logical block size used by the NFS version 4 client. This block size represents the amount of data that the client attempts to read from or write to the NFS server when it needs to do an I/O.

**Data Type**
Unsigned integer (32-bit)

**Default**
32,768 (32 KB)

**Range**
0 to $2^{32}$ - 1

**Units**
Bytes

**Dynamic?**
Yes, but the block size for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. Setting this parameter too low or too high might cause the system to malfunction. Do not set this parameter to anything less than `PAGESIZE` for the specific platform. Do not set this parameter too high because it might cause the system to hang while waiting for memory allocations to be granted.

**When to Change**
Examine the value of this parameter when attempting to change the maximum data transfer size. Change this parameter in conjunction with `nfs:nfs4_max_transfer_size` parameter. If larger transfers are preferred, increase both parameters. If smaller transfers are preferred, then just reducing this parameter should suffice.

**Commitment Level**
Unstable

## `nfs:nfs_cots_timeo` Parameter

**Description**
Controls the default RPC timeout for NFS version 2 mounted file systems using connection-oriented transports such as TCP for the transport protocol.

**Data Type**
Unsigned integer (32-bit)

**Default**
600 (60 seconds)

**Range**
0 to $2^{32}$ - 1

**Units**
10th of seconds

**Dynamic?**
Yes, but the RPC timeout for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
TCP does a good job ensuring requests and responses are delivered appropriately. However, if the round-trip times are very large in a particularly slow network, the NFS version 2 client might time out prematurely.

Increase this parameter to prevent the client from timing out incorrectly. The range of values is very large, so increasing this value too much might result in situations where a retransmission is not detected for long periods of time.

**Commitment Level**
Unstable

## nfs:nfs3_cots_timeo Parameter

**Description**
Controls the default RPC timeout for NFS version 3 mounted file systems using connection-oriented transports such as TCP for the transport protocol.

**Data Type**
Unsigned integer (32-bit)

**Default**
600 (60 seconds)

**Range**
0 to $2^{32}$ - 1

**Units**
10th of seconds

**Dynamic?**
Yes, but the RPC timeout for a file system is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
TCP does a good job ensuring requests and responses are delivered appropriately. However, if the round-trip times are very large in a particularly slow network, the NFS version 3 client might time out prematurely.
Increase this parameter to prevent the client from timing out incorrectly. The range of values is very large, so increasing this value too much might result in situations where a retransmission is not detected for long periods of time.

**Commitment Level**
Unstable

## nfs:nfs4_cots_timeo Parameter

**Description**
Controls the default RPC timeout for NFS version 4 mounted file systems using connection-oriented transports such as TCP for the transport protocol.
The NFS Version 4 protocol specification disallows retransmission over the same TCP connection. Thus, this parameter primarily controls how quickly the NFS client responds to certain events, such as detecting a forced unmount operation or detecting how quickly the NFS server fails over to a new server.

**Data Type**
Unsigned integer (32-bit)

**Default**
600 (60 seconds)

**Range**
0 to $2^{32}$ - 1

**Units**
10th of seconds

**Dynamic?**
Yes, but this parameter is set when the file system is mounted. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
TCP does a good job ensuring requests and responses are delivered appropriately. However, if the round-trip times are very large in a particularly slow network, the NFS version 4 client might time out prematurely.
Increase this parameter to prevent the client from timing out incorrectly. The range of values is very large, so increasing this value too much might result in situations where a retransmission is not detected for long periods of time.

**Commitment Level**
Unstable

## `nfs:nfs_disable_rddir_cache` Parameter

**Description**
Controls the use of a cache to hold responses from `READDIR` and `READDIRPLUS` requests. This cache avoids over-the-wire calls to the NFS server to retrieve directory information.

**Data Type**
Integer (32-bit)

**Default**
0 (caching enabled)

**Range**
0 (caching enabled) or 1 (caching disabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Examine the value of this parameter if interoperability problems develop due to a NFS server that does not update the modification time on a directory when a file or directory is created in it or removed from it. The symptoms are that new names do not appear in directory listings after they have been added to the directory or that old names do not disappear after they have been removed from the directory.
This parameter controls the caching for NFS version 2, 3, and 4 mounted file systems. This parameter applies to all NFS mounted file systems, so caching cannot be disabled or enabled on a per file system basis.
If you disable this parameter, you should also disable the following parameters to prevent bad entries in the DNLC negative cache:

- nfs:nfs_lookup_neg_cache Parameter
- nfs:nfs3_lookup_neg_cache Parameter
- nfs:nfs4_lookup_neg_cache Parameter

**Commitment Level**
Unstable

## `nfs:nfs_do_symlink_cache` Parameter

**Description**
Controls whether the contents of symbolic link files are cached for NFS version 2 mounted file systems.

**Data Type**
Integer (32-bit)

**Default**
1 (caching enabled)

**Range**
0 (caching disabled) or 1 (caching enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
If a NFS server changes the contents of a symbolic link file without updating the modification timestamp on the file or if the granularity of the timestamp is too large, then changes to the contents of the symbolic link file might not be visible on the NFS client for extended periods. In this case, use this parameter to disable the caching of symbolic link contents. Doing so makes the changes immediately visible to applications running on the client.

**Commitment Level**
Unstable

## nfs:nfs3_do_symlink_cache Parameter

**Description**
Controls whether the contents of symbolic link files are cached for NFS version 3 mounted file systems.

**Data Type**
Integer (32-bit)

**Default**
1 (caching enabled)

**Range**
0 (caching disabled) or 1 (caching enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
If a NFS server changes the contents of a symbolic link file without updating the modification timestamp on the file or if the granularity of the timestamp is too large, then changes to the contents of the symbolic link file might not be visible on the NFS client for extended periods. In this case, use this parameter to disable the caching of symbolic link contents. Doing so makes the changes immediately visible to applications running on the client.

**Commitment Level**
Unstable

## nfs:nfs4_do_symlink_cache Parameter

**Description**
Controls whether the contents of symbolic link files are cached for NFS version 4 mounted file systems.

**Data Type**
Integer (32-bit)

**Default**
1 (caching enabled)

**Range**
0 (caching disabled) or 1 (caching enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
If a NFS server changes the contents of a symbolic link file without updating the modification timestamp on the file or if the granularity of the timestamp is too large, then changes to the contents of the symbolic link file might not be visible on the NFS client for extended periods. In this case, use this parameter to disable the caching of symbolic link contents. Doing so makes the changes immediately visible to applications running on the client.

**Commitment Level**
Unstable

## `nfs:nfs_dynamic` Parameter

**Description**
Controls whether a feature known as *dynamic retransmission* is enabled for NFS version 2 mounted file systems using connectionless transports such as UDP. This feature attempts to reduce retransmissions by monitoring NFS server response times and then adjusting RPC timeouts and read- and write- transfer sizes.

**Data Type**
Integer (32-bit)

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
Do not change this parameter.

**Commitment Level**
Unstable

## `nfs:nfs3_dynamic` Parameter

**Description**
Controls whether a feature known as *dynamic retransmission* is enabled for NFS version 3 mounted file systems using connectionless transports such as UDP. This feature attempts to reduce retransmissions by monitoring NFS server response times and then adjusting RPC timeouts and read- and write- transfer sizes.

**Data Type**
Integer (32-bit)

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Boolean values

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
Do not change this parameter.

**Commitment Level**
Unstable

## `nfs:nfs3_jukebox_delay` Parameter

**Description**
Controls the duration of time that the NFS version 3 client waits to transmit a new request after receiving the `NFS3ERR_JUKEBOX` error from a previous request. The `NFS3ERR_JUKEBOX` error is generally returned from the NFS server when the file is temporarily unavailable for some reason. This error is generally associated with hierarchical storage, and CD or tape jukeboxes.

**Data Type**
Long integer (64-bit)

**Default**
1000 (10 seconds expressed as 10 sec * 100Hz)

**Range**
0 to $2^{63}$ - 1 on 64-bit platforms

**Units**
Hz. (Typically, the clock runs at 100Hz.)

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Examine the value of this parameter and perhaps adjust it to match the behaviors exhibited by the NFS server. Increase this value if the delays in making the file available are long in order to reduce network overhead due to repeated retransmissions. Decrease this value to reduce the delay in discovering that the file has become available.

**Commitment Level**
Unstable

## nfs:nfs_lookup_neg_cache Parameter

**Description**
Controls whether a negative name cache is used for NFS version 2 mounted file systems. This negative name cache records file names that were looked up, but not found. The cache is used to avoid over-the-network look-up requests made for file names that are already known to not exist.

**Data Type**
Integer (32-bit)

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
For the cache to perform correctly, negative entries must be strictly verified before they are used. This consistency mechanism is relaxed slightly for read-only mounted file systems. It is assumed that the file system on the NFS server is not changing or is changing very slowly, and that it is okay for such changes to propagate slowly to the NFS client. The consistency mechanism becomes the normal attribute cache mechanism in this case.
If file systems are mounted read-only on the NFS client, but are expected to change on the NFS server and these changes need to be seen immediately by the client, use this parameter to disable the negative cache.
If you disable the `nfs:nfs_disable_rddir_cache` parameter, you should probably also disable this parameter. For more information, see nfs:nfs_disable_rddir_cache Parameter.

**Commitment Level**
Unstable

## nfs:nfs3_lookup_neg_cache Parameter

**Description**
Controls whether a negative name cache is used for NFS version 3 read-only mounted file systems. This negative name cache records file names that were looked up, but were not found. The cache is used to avoid over-the-network look-up requests made for file names that are already known to not exist.

**Data Type**
Integer (32-bit)

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
For the cache to perform correctly, negative entries must be strictly verified before they are used. This consistency mechanism is relaxed slightly for read-only mounted file systems. It is assumed that the file system on the NFS server is not changing or is changing very slowly, and that it is okay for such changes to propagate slowly to the NFS client. The consistency mechanism becomes the normal attribute cache mechanism in this case.
If file systems are mounted read-only on the client, but are expected to change on the NFS server and these changes need to be seen immediately by the NFS client, use this parameter to disable the negative cache.
If you disable the nfs:nfs_disable_rddir_cache Parameter parameter, you should probably also disable this parameter.

**Commitment Level**
Unstable

## `nfs:nfs4_lookup_neg_cache` Parameter

**Description**
Controls whether a negative name cache is used for NFS version 4 mounted file systems. This negative name cache records file names that were looked up, but were not found. The cache is used to avoid over-the-network look-up requests made for file names that are already known to not exist.

**Data Type**
Integer (32-bit)

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
For the cache to perform correctly, negative entries must be strictly verified before they are used. This consistency mechanism is relaxed slightly for read-only mounted file systems. It is assumed that the file system on the NFS server is not changing or is changing very slowly, and that it is okay for such changes to propagate slowly to the NFS client. The consistency mechanism becomes the normal attribute cache mechanism in this case.
If file systems are mounted read-only on the NFS client, but are expected to change on the NFS server and these changes need to be seen immediately by the client, use this parameter to disable the negative cache.
If you disable the nfs:nfs_disable_rddir_cache Parameter parameter, you should probably also disable this parameter.

**Commitment Level**
Unstable

## `nfs:nfs_max_threads` Parameter

**Description**
Controls the number of kernel threads that perform asynchronous I/O for each file system for the NFS version 2 client. Because NFS is based on RPC and RPC is inherently synchronous, separate execution contexts are required to perform NFS operations that are asynchronous from the calling thread.
The operations that can be executed asynchronously are read for read-ahead, readdir for readdir read-ahead, write for putpage and pageio operations, commit, and inactive for cleanup operations that the NFS client performs when it stops using a file.

**Data Type**
Unsigned short

**Default**
8

**Range**
0 to $2^{16}$ - 1

**Units**
Threads

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
To increase or reduce the number of simultaneous I/O operations that are outstanding at any given time. For example, for a very low bandwidth network, you might want to decrease this value so that the NFS client does not overload the network. Alternately, if the network is very high bandwidth, and the NFS client and NFS server have sufficient resources, you might want to increase this value. Doing so can more

effectively utilize the available network bandwidth, and the client and server resources.

**Commitment Level**
Unstable

## `nfs:nfs3_max_threads` Parameter

**Description**
Controls the number of kernel threads that perform asynchronous I/O for each file system for the NFS version 3 client. Because NFS is based on RPC and RPC is inherently synchronous, separate execution contexts are required to perform NFS operations that are asynchronous from the calling thread.
The operations that can be executed asynchronously are read for read-ahead, readdir for readdir read-ahead, write for putpage and pageio requests, and commit.

**Data Type**
Unsigned short

**Default**
8

**Range**
0 to $2^{16}$ - 1

**Units**
Threads

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
To increase or reduce the number of simultaneous I/O operations that are outstanding at any given time. For example, for a very low bandwidth network, you might want to decrease this value so that the NFS client does not overload the network. Alternately, if the network is very high bandwidth, and the NFS client and NFS server have sufficient resources, you might want to increase this value. Doing so can more effectively utilize the available network bandwidth, and the client and server resources.

**Commitment Level**
Unstable

## `nfs:nfs4_max_threads` Parameter

**Description**
Controls the number of kernel threads that perform asynchronous I/O for each file system for the NFS version 4 client. Because NFS is based on RPC and RPC is inherently synchronous, separate execution contexts are required to perform NFS operations that are asynchronous from the calling thread.

The operations that can be executed asynchronously are read for read-ahead, write-behind, directory read-ahead, and cleanup operations that the client performs when it stops using a file.

**Data Type**
Unsigned short

**Default**
8

**Range**
0 to $2^{16}$ - 1

**Units**
Threads

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None

**When to Change**
To increase or reduce the number of simultaneous I/O operations that are outstanding at any given time. For example, for a very low bandwidth network, you might want to decrease this value so that the NFS client does not overload the network. Alternately, if the network is very high bandwidth, and the NFS client and NFS server have sufficient resources, you might want to increase this value. Doing so can more effectively utilize the available network bandwidth, and the client and server resources.

**Commitment Level**
Unstable

## `nfs:nfs3_max_transfer_size` Parameter

**Description**
Controls the maximum size of the data portion of an NFS version 3 `READ`, `WRITE`, `READDIR`, or `READDIRPLUS` request. This parameter controls both the maximum size of the request that the NFS server returns as well as the maximum size of the request that the NFS client generates.

**Data Type**
Unsigned integer (32-bit)

**Default**
4,194,304 (4 MB)

**Range**
0 to $2^{31}$ - 1

**Units**
Bytes

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the maximum transfer size on the NFS server to 0 is likely to cause NFS clients to malfunction or just decide not to attempt to talk to the server.
There is also a limit on the maximum transfer size when using NFS over the UDP transport. UDP has a hard limit of 64 KB per datagram. This 64 KB must include the RPC header as well as other NFS information, in addition to the data portion of the request. Setting the limit too high might result in errors from UDP and communication problems between the NFS client and the NFS server.

**When to Change**
To tune the size of data transmitted over the network. In general, the nfs:nfs3_bsize Parameter parameter should also be updated to reflect changes in this parameter.
For example, when you attempt to increase the transfer size beyond 32 KB, update `nfs:nfs3_bsize` to reflect the increased value. Otherwise, no change in the over-the-wire request size is observed.
If you want to use a smaller transfer size than the default transfer size, use the `mount` command's -wsize -rsize option on a per-file system basis.

**Commitment Level**
Unstable

## `nfs:nfs4_max_transfer_size` Parameter

**Description**
Controls the maximum size of the data portion of an NFS version 4 `READ`, `WRITE`, `READDIR`, or `READDIRPLUS` request. This parameter controls both the maximum size of the request that the NFS server returns as well as the maximum size of the request that the NFS client generates.

**Data Type**
Unsigned integer (32-bit)

**Default**
32, 768 (32 KB)

**Range**
0 to $2^{32}$ - 1

**Units**
Bytes

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the maximum transfer size on the NFS server to 0 is likely to causeFS clients to malfunction or just decide not to attempt to talk to the server.
There is also a limit on the maximum transfer size when using NFS over the UDP transport. For more information about the maximum for UDP, see nfs:nfs3_max_transfer_size Parameter.

**When to Change**
To tune the size of data transmitted over the network. In general, the nfs:nfs4_bsize Parameter parameter should also be updated to reflect changes in this parameter. For example, when you attempt to increase the transfer size beyond 32 KB, update `nfs:nfs4_bsize` to reflect the increased value. Otherwise, no change in the over-the-wire request size is observed.
If you want to use a smaller transfer size than the default transfer size, use the `mount` command's -wsize -rsize option on a per-file system basis.

**Commitment Level**
Unstable

## `nfs:nfs3_max_transfer_size_clts` Parameter

**Description**
Controls the maximum size of the data portion of an NFS version 3 `READ`, `WRITE`, `READDIR`, or `READDIRPLUS` request over UDP. This parameter controls both the maximum size of the request that the NFS server returns as well as the maximum size of the request that the NFS client generates.

**Data Type**
Unsigned integer (32-bit)

**Default**
32, 768 (32 KB)

**Range**
0 to $2^{32}$ - 1

**Units**
Bytes

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the maximum transfer size on the NFS server to 0 is likely to cause NFS clients to malfunction or just decide not to attempt to talk to the server.

**When to Change**
Do not change this parameter.

**Commitment Level**
Unstable

## `nfs:nfs3_max_transfer_size_cots` Parameter

**Description**
Controls the maximum size of the data portion of an NFS version 3 `READ`, `WRITE`, `READDIR`, or `READDIRPLUS` request over TCP. This parameter controls both the maximum size of the request that the NFS server returns as well as the maximum size of the request that the NFS client generates.

**Data Type**
Unsigned integer (32-bit)

**Default**
1,048,576 bytes (1MB)

**Range**
0 to $2^{32}$ - 1

**Units**
Bytes

**Dynamic?**
Yes, but this parameter is set per file system at mount time. To affect a particular file system, unmount and mount the file system after changing this parameter.

**Validation**
None. However, setting the maximum transfer size on the NFS server to 0 is likely to cause NFS clients to malfunction or just decide not to attempt to talk to the server.

**When to Change**
Do not change this parameter unless transfer sizes larger than 1 MB are preferred.

**Commitment Level**
Unstable

## `nfs:nacache` Parameter

**Description**
Tunes the number of hash queues that access the file access cache on the NFS client. The file access cache stores file access rights that users have with respect to files that they are trying to access. The cache itself is dynamically allocated. However, the hash queues used to index into the cache are statically allocated. The algorithm assumes that there is one access cache entry per active file and four of these access cache entries per hash bucket. Thus, by default, the value of this parameter is set to the value of the nfs:nrnode Parameter parameter.

**Data Type**
Integer (32-bit)

**Default**
The default setting of this parameter is 0. This value means that the value of `nacache` should be set to the value of the `nrnode` parameter.

**Range**
1 to $2^{31}$ - 1

**Units**
Access cache entries

**Dynamic?**
No. This value can only be changed by adding or changing the parameter in an `/etc/system.d/`*`file`* , and then rebooting system.

**Validation**
None. However, setting this parameter to a negative value will probably cause the system to try to allocate a very large set of hash queues. While trying to do so, the system is likely to hang.

**When to Change**
Examine the value of this parameter if the basic assumption of one access cache entry per file would be violated. This violation could occur for systems in a timesharing mode where multiple users are accessing the same file at about the same time. In this case, it might be helpful to increase the expected size of the access cache so that the hashed access to the cache stays efficient.

**Commitment Level**
Unstable

## `nfs:nfs_nra` Parameter

**Description**
Controls the number of read-ahead operations that are queued by the NFS version 2 client when sequential access to a file is discovered. These read-ahead operations increase concurrency and read throughput. Each read-ahead request is generally for one logical block of file data.

**Data Type**
Integer (32-bit)

**Default**
4

**Range**
0 to $2^{31}$ - 1

**Units**
Logical blocks.

**Dynamic?**
Yes

**Validation**
None

**When to Change**
To increase or reduce the number of read-ahead requests that are outstanding for a specific file at any given time. For example, for a very low bandwidth network or on a low memory client, you might want to decrease this value so that the NFS client does not overload the network or the system memory. Alternately, if the network is very high bandwidth, and the NFS client and NFS server have sufficient resources, you might want to increase this value. Doing so can more effectively utilize the available network bandwidth, and the client and server resources.

**Commitment Level**
Unstable

## nfs:nfs3_nra Parameter

**Description**
Controls the number of read-ahead operations that are queued by the NFS version 3 client when sequential access to a file is discovered. These read-ahead operations increase concurrency and read throughput. Each read-ahead request is generally for one logical block of file data.

**Data Type**
Integer (32-bit)

**Default**
4

**Range**
0 to $2^{31}$ - 1

**Units**
Logical blocks. (See nfs:nfs3_bsize Parameter.)

**Dynamic?**
Yes

**Validation**
None

**When to Change**
To increase or reduce the number of read-ahead requests that are outstanding for a specific file at any given time. For example, for a very low bandwidth network or on a low memory client, you might want to decrease this value so that the NFS client does not overload the network or the system memory. Alternately, if the network is very high bandwidth and the NFS client and NFS server have sufficient resources, you might want to increase this value. Doing so can more effectively utilize the available network bandwidth, and the client and server resources.

**Commitment Level**
Unstable

## nfs:nfs4_nra Parameter

**Description**
Controls the number of read-ahead operations that are queued by the NFS version 4 client when sequential access to a file is discovered. These read-ahead operations increase concurrency and read throughput. Each read-ahead request is generally for one logical block of file data.

**Data Type**
Integer (32-bit)

**Default**
4

**Range**
0 to $2^{31}$ - 1

**Units**
Logical blocks. (See nfs:nfs3_bsize Parameter.)

**Dynamic?**
Yes

**Validation**
None

**When to Change**
To increase or reduce the number of read-ahead requests that are outstanding for a specific file at any given time. For example, for a very low bandwidth network or on a low memory client, you might want to decrease this value so that the NFS client does not overload the network or the system memory. Alternately, if the network is very high bandwidth and the NFS client and NFS server have sufficient resources, you might want to increase this value. Doing so can more effectively utilize the available network bandwidth, and the client and server resources.

**Commitment Level**
Unstable

## `nfs:nrnode` Parameter

**Description**
Controls the size of the `rnode` cache on the NFS client.
The `rnode`, used by NFS version 2, 3, and 4 clients, is the central data structure that describes a file on the NFS client. The `rnode` contains the file handle that identifies the file on the NFS server. The `rnode` also contains pointers to various caches used by the NFS client to avoid network calls to the server. Each `rnode` has a one-to-one association with a `vnode`. The `vnode` caches file data.
The NFS client attempts to maintain a minimum number of `rnode`s to attempt to avoid destroying cached data and metadata. When an `rnode` is reused or freed, the cached data and metadata must be destroyed.

**Data Type**
Integer (32-bit)

**Default**
The default setting of this parameter is 0, which means that the value `nrnode` should be set to the value of the `ncsize` parameter. Actually, any non positive value of `nrnode` results in `nrnode` being set to the value of `ncsize`.

**Range**
1 to $2^{31}$ - 1

**Units**
`rnode`s

**Dynamic?**
No. This value can only be changed by adding or changing the parameter in an `/etc/system.d/`*`file`* , and then rebooting the system.

**Validation**
The system enforces a maximum value such that the `rnode` cache can only consume 25 percent of available memory.

**When to Change**
Because `rnode`s are created and destroyed dynamically, the system tends to settle upon a *nrnode*-size cache, automatically adjusting the size of the cache as memory pressure on the system increases or as more files are simultaneously accessed. However, in certain situations, you could set the value of `nrnode` if the mix of files being accessed can be predicted in advance. For example, if the NFS client is accessing a few very large files, you could set the value of `nrnode` to a small number so that system memory can cache file data instead of `rnode`s. Alternately, if the client is accessing many small files, you could increase the value of `nrnode` to optimize for storing file metadata to reduce the number of network calls for metadata.
Although it is not recommended, the `rnode` cache can be effectively disabled by setting the value of `nrnode` to 1. This value instructs the client to only cache 1 `rnode`, which means that it is reused frequently.

**Commitment Level**
Unstable

## `nfs:nfs3_pathconf_disable_cache` Parameter

**Description**
Controls the caching of `pathconf` information for NFS Version 3 mounted file systems.

**Data Type**
Integer (32-bit)

**Default**
0 (caching enabled)

**Range**
0 (caching enabled) or 1 (caching disabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
The `pathconf` information is cached on a per file basis. However, if the NFS server can change the information for a specific file dynamically, use this parameter to disable caching. There is no mechanism for the NFS client to validate its cache entry.

**Commitment Level**
Unstable

## nfs:nfs_shrinkreaddir Parameter

**Description**
Some older NFS servers might incorrectly handle NFS version 2 READDIR requests for more than 1024 bytes of directory information. This problem is due to a bug in the server implementation. However, this parameter contains a workaround in the NFS version 2 client.
When this parameter is enabled, the client does not generate READDIR request for larger than 1024 bytes of directory information. If this parameter is disabled, then the over-the-wire size is set to the lesser of either the size passed in by using the getdents system call or by using NFS_MAXDATA, which is 8192 bytes. For more information, see getdents(2).

**Data Type**
Integer (32-bit)

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Examine the value of this parameter if an older NFS version 2 only server is used and interoperability problems occur when the server tries to read directories. Enabling this parameter might cause a slight decrease in performance for applications that read directories.

**Commitment Level**
Unstable

## nfs:nfs3_shrinkreaddir Parameter

**Description**
Some older NFS servers might incorrectly handle NFS version 3 READDIR requests for more than 1024 bytes of directory information. This problem is due to a bug in the server implementation. However, this parameter contains a workaround in the NFS version 3 client.
When this parameter is enabled, the client does not generate READDIR request for larger than 1024 bytes of directory information. If this parameter is disabled, then the over-the-wire size is set to the minimum of either the size passed in by using the getdents system call or by using MAXBSIZE, which is 8192 bytes. For more information, see the getdents(2) man page.

**Data Type**
Integer (32-bit)

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Examine the value of this parameter if an older NFS version 3 only server is used and interoperability problems occur when the server tries to read directories. Enabling this parameter might cause a slight decrease in performance for applications that read directories.

**Commitment Level**
Unstable

## `nfs:nfs_write_error_interval` Parameter

**Description**
Controls the time duration in between logging `ENOSPC` and `EDQUOT` write errors received by the NFS client. This parameter affects NFS version 2, 3, and 4 clients.

**Data Type**
Long integer (64-bit)

**Default**
5 seconds

**Range**
0 to $2^{63}$ - 1

**Units**
Seconds

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Increase or decrease the value of this parameter in response to the volume of messages being logged by the NFS client. Typically, you might want to increase the value of this parameter to decrease the number of `out of space` messages being printed when a full file system on a NFS server is being actively used.

**Commitment Level**
Unstable

### `nfs:nfs_write_error_to_cons_only` Parameter

**Description**
Controls whether NFS write errors are logged to the system console and `syslog` or to the system console only. This parameter affects messages for NFS version 2, 3, and 4 clients.

**Data Type**
Integer (32-bit)

**Default**
0 (system console and `syslog`)

**Range**
0 (system console and `syslog`) or 1 (system console)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Examine the value of this parameter to avoid filling up the file system containing the messages logged by the `syslogd` daemon. When this parameter is enabled, messages are printed on the system console only and are not copied to the `syslog` messages file.

**Commitment Level**
Unstable

## NFS-Related SMF Configuration Parameters

In Oracle Solaris 11.2, the `network/nfs/server` service includes the `nfs-props` property group, which provides configurable parameters to control the refresh of the NFS authentication cache and to control the `mountd netgroup` cache.

- server_authz_cache_refresh Parameter
- netgroup_refresh Parameter

You can use `sharectl` command to get and set these properties.

```
# sharectl get -p server_authz_cache_refresh nfs
server_authz_cache_refresh=600
$ sharectl set -p server_authz_cache_refresh=1 nfs
```

You can also get and set these properties by using SMF commands but you will need to refresh the `network/nfs/server` service.

```
# svccfg -s nfs/server:default setprop nfs-props/server_authz_cache_refresh=1
# svcprop -p nfs-props/server_authz_cache_refresh svc:/network/nfs/
server:default
1
# svcadm restart nfs/server:default
```

### server_authz_cache_refresh Parameter

This parameter controls the refresh of the NFS authentication cache. The default value of the integer property is 600, the minimum is 0, and the max is INT32_MAX. A value of zero ('0') means no expiration.

### netgroup_refresh Parameter

This parameter controls the mountd netgroup cache. The default value of the integer property is 600, the minimum is 0, and the max is INT32_MAX. A value of zero ('0') means no expiration.

## nfssrv Module Parameters

This section describes NFS parameter for the nfssrv module.

### nfssrv:rfs_write_async Parameter

**Description**
Controls the behavior of the NFS version 2 server when it processes WRITE requests. The NFS version 2 protocol mandates that all modified data and metadata associated with the WRITE request reside on stable storage before the server can respond to the client. NFS version 2 WRITE requests are limited to 8192 bytes of data. Thus, each WRITE request might cause multiple small writes to the storage subsystem. This can cause a performance problem.
One method to accelerate NFS version 2 WRITE requests is to take advantage of a client behavior. Clients tend to send WRITE requests in batches. The server can take advantage of this behavior by clustering together the different WRITE requests into a single request to the underlying file system. Thus, the data to be written to the storage subsystem can be written in fewer, larger requests. This method can significantly increase the throughput for WRITE requests.

**Data Type**
Integer (32-bit)

**Default**
1 (clustering enabled)

**Range**
0 (clustering disabled) or 1 (clustering enabled)

**Units**
Boolean values

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Some very small NFS clients, particularly PC clients, might not batch `WRITE` requests. Thus, the behavior required from the clients might not exist. In addition, the clustering in the NFS version 2 server might just add overhead and slow down performance instead of increasing it.

**Commitment Level**
Unstable

## `rpcmod` Module Parameters

This section describes NFS parameters for the `rpcmod` module.

### `rpcmod:clnt_max_conns` Parameter

**Description**
Controls the number of TCP connections that the NFS client uses when communicating with each NFS server. The kernel RPC is constructed so that it can multiplex RPCs over a single connection. However, multiple connections can be used, if preferred.

**Data Type**
Integer (32-bit)

**Default**
1

**Range**
1 to $2^{31}$ - 1

**Units**
Connections

**Dynamic?**
Yes

**Validation**
None

**When to Change**
In general, one connection is sufficient to achieve full network bandwidth. However, if TCP cannot utilize the bandwidth offered by the network in a single stream, then multiple connections might increase the throughput between the NFS client and the NFS server.
Increasing the number of connections doesn't come without consequences. Increasing the number of connections also increases kernel resource usage needed to keep track of each connection.

**Commitment Level**
Unstable

## rpcmod:clnt_idle_timeout Parameter

**Description**
Controls the duration of time on the NFS client that a connection between the NFS client and NFS server is allowed to remain idle before being closed.

**Data Type**
Long integer (64-bit)

**Default**
300,000 milliseconds (5 minutes)

**Range**
0 to $2^{63}$ - 1

**Units**
Milliseconds

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Use this parameter to change the time that idle connections are allowed to exist on the NFS client before being closed. You might want to close connections at a faster rate to avoid consuming system resources.

**Commitment Level**
Unstable

## rpcmod:svc_idle_timeout Parameter

**Description**
Controls the duration of time on the NFS server that a connection between the NFS client and NFS server is allowed to remain idle before being closed.

**Data Type**
Long integer (64-bit)

**Default**
360,000 milliseconds (6 minutes)

**Range**
0 to $2^{63}$ - 1

**Units**
Milliseconds

**Dynamic?**
Yes

**Validation**
None

**When to Change**
Use this parameter to change the time that idle connections are allowed to exist on the NFS server before being closed. You might want to close connections at a faster rate to avoid consuming system resources.

**Commitment Level**
Unstable

5

# Internet Protocol Suite Tunable Parameters

This chapter describes various Internet Protocol (IP) suite properties.

- IP Tunable Parameters
- TCP Tunable Parameters
- UDP Tunable Parameters
- SCTP Tunable Parameters
- ICMP Tunable Parameters
- Per-Route Metrics

For other types of tunable parameters, refer to the following information:

- Oracle Solaris kernel tunable parameters – Oracle Solaris Kernel Tunable Parameters
- Oracle Solaris ZFS tunable parameters – Oracle Solaris ZFS Tunable Parameters
- NFS tunable parameters – NFS Tunable Parameters
- System facility tunable parameters – System Facility Parameters

## Overview of Tuning IP Suite Parameters

You can set all of the tuning parameters that are described in this chapter by using the following `ipadm` command syntax:

```
$ ipadm set-prop -p parameter ip|ipv4|ipv6|tcp|udp|sctp
```

For example, you would set the `extra-priv-ports` tunable parameter as follows:

```
$ ipadm set-prop -p extra-priv-ports=1047 tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   extra-priv-ports      rw   1047         1047         2049,4045    1-65535
```

For more information, see the `ipadm`(8) man page.

### IP Suite Parameter Validation

All of the parameters that are described are checked to verify that they fall in the parameter range. The parameter's range is provided with the description for each parameter.

### Internet Request for Comments

Internet protocol and standard specifications are described in Internet Request for Comments (RFC) documents at http://www.ietf.org/rfc.html.

To review RFC topics, type the RFC number or an Internet-draft file name in the Internet Engineering Task Force (IETF) Repository Retrieval search field.

# IP Tunable Parameters

This section describes parameters pertaining to the IP protocol.

## _addrs_per_if Parameter

**Description**
Defines the maximum number of logical IP interfaces associated with a real interface. Each logical interface in the Oracle Solaris kernel maps to a single IP address.

**Default**
256

**Range**
1 to 8,192

**Dynamic?**
Yes

**When to Change**
Do not change the value. If more logical interfaces are required, you might consider increasing the value. However, recognize that this change might have a negative impact on IP's performance.

**Commitment Level**
Unstable

## _forwarding_src_routed Parameter (IPv4 or IPv6)

**Description**
Controls whether IPv4 or IPv6 forwards packets with source IPv4 routing options or IPv6 routing headers.

**Default**
Off

**Range**
Off or On

**Dynamic?**
Yes

**When to Change**
Keep this parameter disabled to prevent denial of service attacks.

**Commitment Level**
Unstable

## _icmp_err_interval and _icmp_err_burst Parameters

**Description**
Controls the rate of IP in generating ICMP error messages. IP generates only up to _icmp_err_burst IP error messages in any _icmp_err_interval.
The _icmp_err_interval parameter protects IP from denial of service attacks. Setting this parameter to 0 disables rate limiting. It does not disable the generation of error messages.

**Default**
100 milliseconds for _icmp_err_interval
10 error messages for _icmp_err_burst

**Range**
0 – 99,999 milliseconds for _icmp_err_interval
1 – 99,999 error messages for _icmp_err_burst

**Dynamic?**
Yes

**When to Change**
If you need a higher error message generation rate for diagnostic purposes.

**Commitment Level**
Unstable

## _policy_mask Parameter

**Description**
Enables or disables IPQoS processing in any of the following callout positions: forward outbound, forward inbound, local outbound, and local inbound. This parameter is a bitmask as follows:

| Not Used | Not Used | Not Used | Not Used | Forward Outbound | Forward Inbound | Local Outbound | Local Inbound |
|---|---|---|---|---|---|---|---|
| X | X | X | X | 0 | 0 | 0 | 0 |

A 1 in any of the position masks or disables IPQoS processing in that particular callout position. For example, a value of 0x01 disables IPQoS processing for all the local inbound packets.

**Default**
The default value is 0, meaning that IPQoS processing is enabled in all the callout positions.

**Range**
0 (0x00) to 15 (0x0F). A value of 15 indicates that IPQoS processing is disabled in all the callout positions.

**Dynamic?**
Yes

**When to Change**
If you want to enable or disable IPQoS processing in any of the callout positions.

**Commitment Level**
Unstable

## `_respond_to_echo_broadcast` (IP) and `_respond_to_echo_multicast` Parameters (IPv4 or IPv6)

**Description**
Controls whether IP responds to a broadcast ICMPv4 echo request or a multicast ICMPv4 or ICMPv6 echo request.

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
If you do not want this behavior for security reasons, disable it.

**Commitment Level**
Unstable

## `hoplimit` Parameter (IPv6)

**Description**
Sets the value of the hop limit in the IPv6 header for the outbound ICMPv6 error and reply packets. The hop limit defines the maximum number of routers a packet can pass through on the path to the destination. It is primarily used to clear messages from the network when a misconfiguration would otherwise case messages to endlessly loop through the same set of routers.
The `hoplimit` set on outbound ICMP requests and on UDP, TCP, and SCTP messages is not controlled by this property. It is instead controlled by the `_ipv6_hoplimit` property for each respective protocol.

**Default**
255

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Generally, you do not need to change this value.

**Commitment Level**
Stable

## `hostmodel` Parameter (IPv4 or IPv6)

**Description**
Controls send and receive behavior for IPv4 or IPv6 packets on a multi-homed system.

**Default**
`weak`

**Range**
`weak`, `strong`, or `src-priority`

- `weak`
  - Outgoing packets - The source address of the packet going out need not match the address configured on the outgoing interface.
  - Incoming packets - The destination address of the incoming packet need not match the address configured on the incoming interface.
- `strong`
  - Outgoing packets - The source address of the packet going out must match the address configured on the outgoing interface.
  - Incoming packets - The destination address of the incoming packet must match the address configured on the incoming interface.
- `src-priority`
  - Outgoing packets - If multiple routes for the IP destination in the packet are available, the system prefers routes where the IP source address in the packet is configured on the outgoing interface.

    If no such route is available, the system falls back to selecting the *best* route, as with the weak ES case.
  - Incoming packets - The destination address of the incoming packet must be configured on any one of the host's interface.

**Dynamic?**
Yes

**When to Change**
If a system has interfaces that cross strict networking domains (for example, a firewall or a VPN node), set this parameter to strong.

**Commitment Level**
Stable

## `recv-multicast-scaling` Parameter

**Description**
Defines system-wide default value to enable receiving multicast packets through a more scalable data path that uses extra worker threads to deliver multicast packets. Multicast applications are usually streaming data type, where packets are small but the packet rate can be very high. The normal data path cannot handle very high packet rate without drops. This problem is especially acute when there are multiple receivers for the same multicast

group. However, for the general case, there might be a tradeoff between latency and scalability.

**Default**
`0` (off)

**Range**
`0` (off) or `1` (on)

**Dynamic?**
Yes

**When to Change**
Change from the default setting in cases of high volume of UDP multicast traffic.

**Commitment Level**
Stable

## `send-redirects` Parameter (IPv4 or IPv6)

**Description**
Controls whether IPv4 or IPv6 sends out ICMPv4 or ICMPv6 redirect messages.

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
If you do not want this behavior for security reasons, disable it.

**Commitment Level**
Stable

## `ttl` Parameter (IPv4)

**Description**
Controls the time to live (TTL) value in the IPv4 header for the outbound IPv4 ICMP error and reply packets. The TTL defines the maximum number of routers a packet can pass through on the path to the destination. It is primarily used to clear messages from the network when a misconfiguration would otherwise cause messages to endlessly loop through the same set of routers.
The `ttl` set on outbound ICMP requests and on UDP, TCP, and SCTP messages is not controlled by this property. It is instead controlled by the _ipv4_ttl Parameter property for each respective protocol.

**Default**
255

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Generally, you do not need to change this value.

**Commitment Level**
Stable

## IP Tunable Parameters Related to Duplicate Address Detection

The following parameters can be configured to perform duplicate address detection (DAD) in the network.

### `_arp_defend_interval`/`_ndp_defend_interval` Parameter

**Description**
Interval in which the system continues to broadcast announcements for a specific address using IPv4 ARP and IPv6 NDP, respectively, to detect duplicate addresses in the network after the initial duplicate address detection process completes successfully.

**Default**
300,000 milliseconds

**Range**
0-360,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

### `_arp_defend_period`/`_ndp_defend_period` Parameter

**Description**
Time period within which unrequested address-defense ARP or NDP messages are generated on any one physical network interface. These parameters work together with _arp_defend_rate/_ndp_defend_rate Parameter.
These parameters does not apply to normal ARP or NDP resolution or to address defense due to detected conflicts. Rather, the parameters are implemented only on unbidden conflict detection traffic.

**Default**
3,600 seconds

**Range**
0-3,600

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _arp_defend_rate/_ndp_defend_rate Parameter

**Description**
Number of unrequested address-defense ARP or NDP messages that can be generated in an hour period on any one physical network interface. The time period can be revised by configuring _arp_defend_period/_ndp_defend_period Parameter. The _arp_defend_rate/_ndp_defend_rate work together with the _arp_defend_period/_ndp_defend_period to prevent a system with a large number of configured IP addresses from flooding the network with ARP traffic.
By default, the system will continuously broadcast an ARP announcement or NDP advertisement every five minutes for each address that already passed duplicate address detection. However, the total number of such ARP announcements or NDP advertisements from an interface is further limited to 100 messages per hour regardless of the number of configured addresses.
These parameters does not apply to normal ARP or NDP resolution nor to address defense due to detected conflicts. Rather, the parameters are implemented only on unbidden conflict detection traffic.

**Default**
100 messages/hour

**Range**
0-20,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _arp_fastprobe_count Parameter

**Description**
In a transmit-pause sequence, the number of probes that are transmitted to detect duplicate addresses before pausing. The length of time is defined in _arp_fastprobe_interval Parameter. The parameter is used for faster probing for duplicate addresses.

**Default**
3 packets

**Range**
0-20

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _arp_fastprobe_interval Parameter

**Description**
Similar function to _arp_probe_interval Parameter, which is the time between the sending of a set number of probes to detect duplicate addresses. To accelerate the process in bringing up an IP interface, and if the underlying driver can properly report link up or link down events, the system uses this parameter as the interval between sending out probes. This parameter works together with _arp_fastprobe_count Parameter.

**Default**
150 milliseconds

**Range**
10-20,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _arp_probe_count Parameter

**Description**
In a transmit-pause sequence, the number of probes that are transmitted to detect duplicate addresses before pausing. The length of the pause is determined by _arp_probe_interval Parameter. After the pause time expires, probing resumes.

**Default**
3 packets

**Range**
0-20

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _arp_probe_interval Parameter

**Description**
Time between the sending of a set number of probes to detect duplicate addresses. The number of probes that is sent after each interval is defined in _arp_probe_count Parameter.

**Default**
1,500 milliseconds

**Range**
10-20,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _defend_interval Parameter

**Description**
Length of time a system defends its local address when it is detected to be in conflict with another system's IP address. The number of attempts to defend the address within this period is defined in _max_defend Parameter.

**Default**
30 seconds

**Range**
0-999,999

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## _dup_recovery Parameter

**Description**
Time between the transmission of probes after the system marks a non-temporary address down because it conflicts with the same address in a remote system. The local system sends out probes periodically to test whether the conflict persists. If the probe receives no reply, the conflict is considered cleared and the address is marked up again.

**Default**
300,000 milliseconds

**Range**
0-360,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## `_max_defend` Parameter

**Description**
The number of times an IP address is defended if the address conflicts with another system's IP address. Defense of the address occurs within the time specified in _defend_interval Parameter.

**Default**
3 counts

**Range**
0-1,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

## `_max_temp_defend` Parameter

**Description**
Number of times a system defends a temporary local address or a DHCP controlled address when that address is in conflict with another system's IP address. When the value of `_max_temp_defend` is passed, the system gives up the address.

**Default**
1 count

**Range**
0-1,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Unstable

### `arp-publish-count/ndp-unsolicit-count` Parameter

**Description**
Number of unsolicited IPv4 ARP announcements or IPv6 NDP advertisements transmitted over an interface to update the address cache of network peers. The announcements are sent after a local IP address has been successfully brought up and are transmitted at intervals controlled by the arp-publish-interval/ndp-unsolicit-interval Parameter parameters.

**Default**
3 packets

**Range**
1-20

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Stable

### `arp-publish-interval/ndp-unsolicit-interval` Parameter

**Description**
Time between successive unsolicited IPv4 ARP announcements or IPv6 NDP advertisements that are sent after a local IP address is successfully brought up. The announcements are sent to update the address cache of network peers.

**Default**
2,000 milliseconds

**Range**
1,000-20,000

**Dynamic?**
Yes

**When to Change**
Never

**Commitment Level**
Stable

## IP Tunable Parameters With Additional Cautions

Changing the following parameters is not recommended.

### _icmp_return_data_bytes Parameter (IPv4 or IPv6)

**Description**
When IPv4 or IPv6 sends an ICMPv4 or ICMPv6 error message, it includes the IP header of the packet that caused the error message. This parameter controls how many extra bytes of the packet beyond the IPv4 or IPv6 header are included in the ICMPv4 or ICMPv6 error message.

**Default**
64 for IPv4
1,280 for IPv6

**Range**
8-65,536 for IPv4
8-1,280 for IPv6

**Dynamic?**
Yes

**When to Change**
Do not change the value. Including more information in an ICMP error message might help in diagnosing network problems. If this feature is needed, increase the value.

**Commitment Level**
Unstable

### _pathmtu_interval Parameter

**Description**
Specifies the interval in milliseconds at which IP flushes the path maximum transfer unit (PMTU) discovery information, and tries to rediscover PMTU.
Refer to RFC 1191 on PMTU discovery.

**Default**
1,200 milliseconds (20 minutes)

**Range**
2-999,999,999

**Dynamic?**
Yes

**When to Change**
Do not change this value.

**Commitment Level**
Unstable

## TCP Tunable Parameters

This section describes parameters specific to the TCP transport protocol.

## _conn_req_max_q Parameter

**Description**
Specifies the default maximum number of pending TCP connections for a TCP listener waiting to be accepted by `accept`. See also _conn_req_max_q0 Parameter.

**Default**
128

**Range**
1 to 4,294,967,295

**Dynamic?**
Yes

**When to Change**
For applications such as web servers that might receive several connection requests, the default value might be increased to match the incoming rate.
Do not increase the parameter to a very large value. The pending TCP connections can consume excessive memory. Also, if an application cannot handle that many connection requests fast enough because the number of pending TCP connections is too large, new incoming requests might be denied.
Note that increasing `_conn_req_max_q` does not mean that applications can have that many pending TCP connections. Applications can use `listen` to change the maximum number of pending TCP connections for each socket. This parameter is the maximum an application can use `listen` to set the number to. Thus, even if this parameter is set to a very large value, the actual maximum number for a socket might be much less than `_conn_req_max_q`, depending on the value used in `listen`.

**Commitment Level**
Unstable

## _conn_req_max_q0 Parameter

**Description**
Specifies the default maximum number of incomplete (three-way handshake not yet finished) pending TCP connections for a TCP listener.
For more information about TCP three-way handshake, refer to RFC 793. See also _conn_req_max_q Parameter.

**Default**
1,024

**Range**
0 to 4,294,967,295

**Dynamic?**
Yes

**When to Change**
For applications such as web servers that might receive excessive connection requests, you can increase the default value to match the incoming rate.

The following explains the relationship between `_conn_req_max_q0` and the maximum number of pending connections for each socket.
When a connection request is received, TCP first checks if the number of pending TCP connections (three-way handshake is done) waiting to be accepted exceeds the maximum (*N*) for the listener. If the connections are excessive, the request is denied. If the number of connections is allowable, then TCP checks if the number of incomplete pending TCP connections exceeds the sum of *N* and `_conn_req_max_q0`. If it does not, the request is accepted. Otherwise, the oldest incomplete pending TCP request is dropped.

**Commitment Level**
Unstable

## `_conn_req_min` Parameter

**Description**
Specifies the default minimum value for the maximum number of pending TCP connection requests for a listener waiting to be accepted. This is the lowest maximum value of `listen` that an application can use.

**Default**
1

**Range**
1 to 1,024

**Dynamic?**
Yes

**When to Change**
This parameter can be a solution for applications that use `listen` to set the maximum number of pending TCP connections to a value too low. Increase the value to match the incoming connection request rate.

**Commitment Level**
Unstable

## `_deferred_ack_interval` Parameter

**Description**
Specifies the time-out value for the TCP-delayed acknowledgment (ACK) timer for stems that are not directly connected.
Refer to RFC 1122, 4.2.3.2.

**Default**
100 milliseconds

**Range**
1 millisecond to 60,000 milliseconds

**Dynamic?**
Yes

**When to Change**
Do not increase this value to more than 500 milliseconds.

Increase the value under the following circumstances:

- Slow network links (less than 57.6 Kbps) with greater than 512 bytes maximum segment size (MSS)
- The interval for receiving more than one TCP segment is short

**Commitment Level**
Unstable

## _deferred_acks_max Parameter

**Description**
Specifies the maximum number of TCP segments received from remote destinations (not the same subnet) before an acknowledgment (ACK) is generated. TCP segments are measured in units of maximum segment size (MSS) for individual connections. If set to 0 or 1, no ACKs are delayed, assuming all segments are 1 MSS long. The actual number is dynamically calculated for each connection. The value is the default maximum.

**Default**
2

**Range**
0 to 16

**Dynamic?**
Yes

**When to Change**
This parameter should not be changed in normal circumstances.

**Commitment Level**
Unstable

## _ipv4_ttl Parameter

**Description**
Controls the time to live (TTL) value in the IPv4 header for outbound TCP messages sent over IPv4. For more information, see the description for ttl Parameter (IPv4).

**Default**
64 bytes

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## _ipv6_hoplimit Parameter

**Description**
Sets the value of the hop limit in the IPv6 header for the outbound TCP messages sent over IPv6. For more information, see the description for hoplimit Parameter (IPv6).

**Default**
60

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## _local_dack_interval Parameter

**Description**
Specifies the time-out value for TCP-delayed acknowledgment (ACK) timer for stems that are directly connected.
Refer to RFC 1122, 4.2.3.2.

**Default**
50 milliseconds

**Range**
10 milliseconds to 500 milliseconds

**Dynamic?**
Yes

**When to Change**
Do not increase this value to more than 500 milliseconds.
Increase the value under the following circumstances:

- Slow network links (less than 57.6 Kbps) with greater than 512 bytes maximum segment size (MSS)
- The interval for receiving more than one TCP segment is short

**Commitment Level**
Unstable

## _local_dacks_max Parameter

**Description**
Specifies the maximum number of TCP segments received from peers on the same subnet before an acknowledgment (ACK) is generated. TCP segments are measured in units of

maximum segment size (MSS) for individual connections. If set to 0 or 1, it means no ACKs are delayed, assuming all segments are 1 MSS long. The actual number is dynamically calculated for each connection. The value is the default maximum.

**Default**
8

**Range**
0 to 16

**Dynamic?**
Yes

**When to Change**
Do not change the value. In some circumstances, when the network traffic becomes very bursty because of the delayed ACK effect, decrease the value. Do not decrease this value below 2.

**Commitment Level**
Unstable

## `_local_slow_start_initial` Parameter

**Description**
Defines the initial congestion window size in the maximum segment size (MSS) of a TCP connection between systems on the same subnet.

**Default**
10

**Range**
1 to 16,384

**Dynamic?**
Yes

**When to Change**
Consider increasing this parameter value if applications would benefit from a larger initial window.

**Commitment Level**
Unstable

## `_rev_src_routes` Parameter

**Description**
If set to 0, TCP does not reverse the IP source routing option for incoming connections for security reasons. If set to 1, TCP does the normal reverse source routing.

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
If IP source routing is needed for diagnostic purposes, enable it.

**Commitment Level**
Unstable

## `_rst_sent_rate` Parameter

**Description**
Sets the maximum number of RST segments that TCP can send out per second.

**Default**
40

**Range**
0 to 4,294,967,295

**Dynamic?**
Yes

**When to Change**
In a TCP environment, there might be a legitimate reason to generate more RSTs than the default value allows. In this case, increase the default value of this parameter.

**Commitment Level**
Unstable

## `_rst_sent_rate_enabled` Parameter

**Description**
If this parameter is set to 1, the maximum rate of sending a RST segment is controlled by the `ipadm` parameter, `_rst_sent_rate`. If this parameter is set to 0, no rate control when sending a RST segment is available.

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
This tunable helps defend against denial of service attacks on TCP by limiting the rate by which a RST segment is sent out. The only time this rate control should be disabled is when strict conformance to RFC 793 is required.

**Commitment Level**
Unstable

## _slow_start_after_idle Parameter

**Description**
The congestion window size in the maximum segment size (MSS) of a TCP connection after it has been idled (no segment received) for a period of one retransmission timeout (RTO).
Refer to RFC 2414 on how the initial congestion window size is calculated.

**Default**
4

**Range**
1 to 16,384

**Dynamic?**
Yes

**When to Change**
For more information, see _slow_start_initial Parameter.

**Commitment Level**
Unstable

## _slow_start_initial Parameter

**Description**
Defines the maximum initial congestion window size in the maximum segment size (MSS) of a TCP connection.
Refer to RFC 2414 on how the initial congestion window size is calculated.

**Default**
10

**Range**
1 to 10

**Dynamic?**
Yes

**When to Change**
Do not change the value.
If the initial cwnd size causes network congestion under special circumstances, decrease the value.

**Commitment Level**
Unstable

## _time_wait_interval Parameter

**Description**
Specifies the time in milliseconds that a TCP connection stays in TIME-WAIT state.
For more information, refer to RFC 1122, 4.2.2.13.

**Default**
60,000 (60 seconds)

**Range**
1 second to 600,000 milliseconds

**Dynamic?**
Yes

**When to Change**
This parameter does not need to be changed in normal circumstances. If the normal usage of a system results in thousands and thousands of TCP connections waiting in TIME-WAIT state, the parameter value may be decreased. The value should not be lower than 10 seconds.

**Commitment Level**
Unstable

## _tstamp_always Parameter

**Description**
If set to 1, TCP always sends a SYN segment with the timestamp option. If set to 2, timestamps are completely disabled, regardless of whether the TCP connection was opened actively or passively. Note that if TCP receives a SYN segment with the timestamp option, TCP responds with a SYN segment with the timestamp option even if the parameter is set to 0.

**Default**
0 (disabled)

**Range**
0 (disabled), 1 (enabled), or 2 (disabled regardless of how TCP connection was opened)

**Dynamic?**
Yes

**When to Change**
If getting an accurate measurement of round-trip time (RTT) and TCP sequence number wraparound is a problem, enable this parameter.
Refer to RFC 1323 for more reasons to enable this option.

**Commitment Level**
Unstable

## _wscale_always Parameter

**Description**
When this parameter is enabled, which is the default setting, TCP always sends a SYN segment with the window scale option, even if the window scale option value is 0. Note that if TCP receives a SYN segment with the window scale option, even if the parameter is disabled, TCP responds with a SYN segment with the window scale option. In addition, the option value is set according to the receive window size.
Refer to RFC 1323 for the window scale option.

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
If there is an interoperability problem with an old TCP stack that does not support the window scale option, disable this parameter.

**Commitment Level**
Unstable

## `cwnd-max` Parameter

**Description**
Defines the maximum value of the TCP congestion window in bytes.
For more information about the TCP congestion window, refer to RFC 1122 and RFC 2581.

**Default**
1,048,576

**Range**
128 to 1,073,741,824

**Dynamic?**
Yes

**When to Change**
This parameter does not need to be changed in normal circumstances. If the system needs to communicate with peers far away (round trip time in the order of hundreds of milliseconds) using very fast network (in the order of Gbps), increase the default value to match the bandwidth-delay product to those peers. Note that max-buf Parameter should also be increased at the same time.

**Commitment Level**
Stable

## `ecn` Parameter

**Description**
Controls Explicit Congestion Notification (ECN) support.
If this parameter is set to `never` TCP does not negotiate with a peer that supports the ECN mechanism.
If this parameter is set to `passive`when initiating a connection, TCP does not tell a peer that it supports ECN mechanism.
However, TCP tells a peer that it supports ECN mechanism when accepting a new incoming connection request if the peer indicates that it supports ECN mechanism in the SYN segment.

If this parameter is set to `active`, in addition to negotiating with a peer on the ECN mechanism when accepting connections, TCP indicates in the outgoing SYN segment that it supports the ECN mechanism when TCP makes active outgoing connections.
Refer to RFC 3168 for information about ECN.

**Default**
`Active`

**Range**
`never`, `passive`, or `active`

**Dynamic?**
Yes

**When to Change**
ECN can help TCP better handle congestion control. However, there might be existing TCP implementations, firewalls, NATs, and other non-conforming network devices that are confused by this mechanism. These devices do not comply to the IETF standard. It is suggested that these devices be replaced. In situations where replacing non-conforming devices is not feasible, this parameter value can be set to `passive` or `never`.

**Commitment Level**
Stable

## `largest-anon-port` Parameter

**Description**
This parameter controls the largest port number TCP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol but not a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

**Unit**
Port number

**Default**
65,535

**Range**
32,768 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
Stable

## max-buf Parameter

**Description**
Defines the maximum send and receive buffer size in bytes. This parameter controls how large the send and receive buffers are set to by an application that uses setsockopt.

**Default**
1,048,576

**Range**
128,000 to 1,073,741,824

**Dynamic?**
Yes

**When to Change**
If TCP connections are being made in a high-speed network environment, increase the value to match the network link speed. The cwnd-max Parameter parameter should probably be increased at the same time.

**Commitment Level**
Stable

## recv-buf Parameter

**Description**
Defines the default receive window size in bytes. Refer to Per-Route Metrics for a discussion of setting a different value on a per-route basis. See also max-buf Parameter and _recv_hiwat_minmss Parameter.

**Default**
256,000

**Range**
2,048 to the current value of max-buf

**Dynamic?**
Yes

**When to Change**
An application can use setsockopt (SO_RCVBUF) to change the individual connection's receive buffer. See the setsockopt(3C) man page.

**Commitment Level**
Stable

## sack Parameter

**Description**
If set to active, TCP always sends a SYN segment with the selective acknowledgment (SACK) permitted option. If TCP receives a SYN segment with a SACK-permitted option and this parameter is set to passive TCP responds with a

SACK-permitted option. If the parameter is set to `never` TCP does not send a SACK-permitted option, regardless of whether the incoming segment contains the SACK permitted option.
Refer to RFC 2018 for information about the SACK option.

**Default**
`active`

**Range**
`never`, `passive`, or `active`

**Dynamic?**
Yes

**When to Change**
SACK processing can improve TCP retransmission performance so it should be actively enabled. Sometimes, the other side can be confused with the SACK option actively enabled. If this confusion occurs, set the value to `passive` so that SACK processing is enabled only when incoming connections allow SACK processing.

**Commitment Level**
Stable

## `send-buf` Parameter

**Description**
Defines the default send window size in bytes. Refer to Per-Route Metrics for a discussion of setting a different value on a per-route basis. See also max-buf Parameter.

**Default**
49,152

**Range**
4,096 to the current value of max-buf Parameter

**Dynamic?**
Yes

**When to Change**
An application can use `setsockopt (SO_SNDBUF)` to change the individual connection's send buffer. See the `setsockopt`(3C) man page.

**Commitment Level**
Stable

## `smallest-anon-port` Parameter

**Description**
This parameter controls the smallest port number TCP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol but not a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

**Unit**
Port number

**Default**
32,768

**Range**
1,024 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
Stable

### `tcp_cwnd_normal` Parameter

**Description**
One of three variables for the congestion window burst throttle, along side `tcp_cwnd_infinite` and `tcp_cwnd_ss` that together manage packet transfers in cases of congestion.
To prevent performance degradation from transfer congestion, change the parameter's value in a file in the `/etc/system.d` directory as follows:

```
# echo "set ip:tcp_cwnd_normal=0xFF" >> /etc/system.d/site:filename
# reboot
```

where *site:filename* refers to the file that contains the new parameter setting (`0xFF`). The new setting will be read from `/etc/system.d/file` into the `/etc/system` file during the reboot. The naming convention *site:filename* enables you to identify the file and the change that you implemented on the parameter. For more information about using files in `/etc/system.d`, see /etc/system.d/ Directory Files.
For more information about the congestion window, refer to RFC 2581 and RFC 3390.

**Default**
16

**Range**
1-65535

**Dynamic?**
Yes

**When to Change**
See Description

**Commitment Level**
Unstable

## TCP Parameters With Additional Cautions

Changing the following parameters is not recommended.

## _ip_abort_interval Parameter

**Description**
Specifies the default total retransmission timeout value for a TCP connection. For a given TCP connection, if TCP has been retransmitting for _ip_abort_interval period of time and it has not received any acknowledgment from the other endpoint during this period, TCP closes this connection.
For TCP retransmission timeout (RTO) calculation, refer to RFC 1122, 4.2.3. See also _rexmit_interval_max Parameter.

**Default**
5 minutes

**Range**
500 milliseconds to 1193 hours

**Dynamic?**
Yes

**When to Change**
Do not change this value. See _rexmit_interval_max for exceptions.

**Commitment Level**
Unstable

## _keepalive_interval Parameter

**Description**
This ipadm parameter sets a probe interval that is first sent out after a TCP connection is idle on a system-wide basis.
Oracle Solaris supports the TCP keep-alive mechanism as described in RFC 1122. This mechanism is enabled by setting the SO_KEEPALIVE socket option on a TCP socket.
If SO_KEEPALIVE is enabled for a socket, the first keep-alive probe is sent out after a TCP connection is idle for two hours, the default value of the tcp_keepalive_interval parameter. If the peer does not respond to the probe after eight minutes, the TCP connection is aborted. For more information, refer to _rexmit_interval_initial Parameter.
You can also use the TCP_KEEPALIVE_THRESHOLD socket option on individual applications to override the default interval so that each application can have its own interval on each socket. The option value is an unsigned integer in milliseconds. Also see the tcp(4P) man page.

**Default**
2 hours

**Range**
10 seconds to 10 days

**Units**
Unsigned integer (milliseconds)

**Dynamic?**
Yes

**When to Change**
Do not change the value. Lowering it may cause unnecessary network traffic and might also increase the chance of premature termination of the connection because of a transient network problem.

**Commitment Level**
Unstable

## `_recv_hiwat_minmss` Parameter

**Description**
Controls the default minimum receive window size. The minimum is `_recv_hiwat_minmss` times the size of maximum segment size (MSS) of a connection.

**Default**
8

**Range**
1 to 65,536

**Dynamic?**
Yes

**When to Change**
Do not change the value. If changing it is necessary, do not change the value lower than 4.

**Commitment Level**
Unstable

## `_rexmit_interval_extra` Parameter

**Description**
Specifies a constant added to the calculated retransmission time out value (RTO).

**Default**
0 milliseconds

**Range**
0 to 7,200,000 milliseconds

**Dynamic?**
Yes

**When to Change**
Do not change the value.
When the RTO calculation fails to obtain a good value for a connection, you can change this value to avoid unnecessary retransmissions.

**Commitment Level**
Unstable

## _rexmit_interval_initial Parameter

**Description**
Specifies the default initial retransmission timeout (RTO) value for a TCP connection. Refer to Per-Route Metrics for a discussion of setting a different value on a per-route basis.

**Default**
1,000 milliseconds

**Range**
1 millisecond to 20,000 milliseconds

**Dynamic?**
Yes

**When to Change**
Do not change this value. Lowering the value can result in unnecessary retransmissions. The TCP_RTO_INITIAL socket option can be used to change the initial retransmission timeout on a per-socket basis.

**Commitment Level**
Unstable

## _rexmit_interval_max Parameter

**Description**
Defines the default maximum retransmission timeout value (RTO). The calculated RTO for all TCP connections cannot exceed this value. See also _ip_abort_interval Parameter.

**Default**
60,000 milliseconds

**Range**
1 millisecond to 7,200,000 milliseconds

**Dynamic?**
Yes

**When to Change**
Do not change the value in a normal network environment.
If, in some special circumstances, the round-trip time (RTT) for a connection is about 10 seconds, you can increase this value. If you change this value, you should also change the _ip_abort_interval parameter. Change the value of _ip_abort_interval to at least four times the value of _rexmit_interval_max. The TCP_RTO_MAX socket option can be used to change the initial retransmission timeout on a per-socket basis.

**Commitment Level**
Unstable

### _rexmit_interval_min Parameter

**Description**
Specifies the default minimum retransmission time out (RTO) value. The calculated RTO for all TCP connections cannot be lower than this value. See also _rexmit_interval_max Parameter.

**Default**
200 milliseconds

**Range**
1 millisecond to 7,200,000 milliseconds

**Dynamic?**
Yes

**When to Change**
Do not change the value in a normal network environment.
TCP's RTO calculation should cope with most RTT fluctuations. If, in some very special circumstances, the round-trip time (RTT) for a connection is about 10 seconds, increase this value. If you change this value, you should change the `_rexmit_interval_max` parameter. Change the value of `_rexmit_interval_max` to at least eight times the value of `_rexmit_interval_min`. The TCP_RTO_MIN socket option can be used to change the initial retransmission timeout on a per-socket basis.

**Commitment Level**
Unstable

### _tstamp_if_wscale Parameter

**Description**
If this parameter is set to 1, and the window scale option is enabled for a connection, TCP also enables the `timestamp` option for that connection.

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
Do not change this value. In general, when TCP is used in high-speed network, protection against sequence number wraparound is essential. Thus, you need the `timestamp` option.

**Commitment Level**
Unstable

## UDP Tunable Parameters

This section describes parameters specific to the UDP protocol.

## _ipv4_ttl Parameter

**Description**
Controls the time to live (TTL) value in the IPv4 header for outbound UDP messages sent over IPv4. For more information, see the description for ttl Parameter (IPv4).

**Default**
64 bytes

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## _ipv6_hoplimit Parameter

**Description**
Sets the value of the hop limit in the IPv6 header for the outbound UDP messages sent over IPv6. For more information, see the description for hoplimit Parameter (IPv6).

**Default**
60

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## largest-anon-port Parameter

**Description**
This parameter controls the largest port number UDP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol but not a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

**Unit**
Port number

**Default**
65,535

**Range**
32,768 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
Stable

## `max-buf` Parameter

**Description**
Defines the maximum send and receive buffer size for a UDP socket. It controls how large the send and receive buffers might be set to by an application that uses `setsockopt`.

**Default**
2,097,152

**Range**
65,536 to 1,073,741,824

**Dynamic?**
Yes

**When to Change**
Increase the value of this parameter to match the network link speed if associations are being made in a high-speed network environment.

**Commitment Level**
Stable

## `recv-buf` Parameter

**Description**
Defines the default receive buffer size for a UDP socket. For more information, see max-buf Parameter.

**Default**
57,344 bytes

**Range**
128 to the current value of `max-buf`

**Dynamic?**
Yes

**When to Change**
An application can use `setsockopt (SO_RCVBUF)` to change the individual connection's receive buffer. See the `setsockopt`(3C) man page.

**Commitment Level**
able

## `send-buf` Parameter

**Description**
Defines the default send buffer size for a UDP socket. For more information, see max-buf Parameter.

**Default**
57,344 bytes

**Range**
1,024 to the current value of `max-buf`

**Dynamic?**
Yes

**When to Change**
An application can use `setsockopt (SO_SNDBUF)` to change the size for an individual socket. In general, you do not need to change the default value.

**Commitment Level**
Stable

## `smallest-anon-port` Parameter

**Description**
This parameter controls the smallest port number UDP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol but not a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

**Unit**
Port number

**Default**
32,768

**Range**
1,024 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
Stable

# SCTP Tunable Parameters

This section describes parameters related to the stream control transmission protocol.

## _addip_enabled Parameter

**Description**
Enables or disables SCTP dynamic address reconfiguration.

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
The parameter can be enabled if dynamic address reconfiguration is needed. Due to security implications, enable this parameter only for testing purposes.

**Commitment Level**
Unstable

## _cookie_life Parameter

**Description**
Sets the lifespan of a cookie in milliseconds.

**Default**
60,000

**Range**
10 to 60,000,000

**Dynamic?**
Yes

**When to Change**
Generally, you do not need to change this value. This parameter might be changed in accordance with _rto_max Parameter.

**Commitment Level**
Unstable

## _deferred_ack_interval Parameter

**Description**
Sets the time-out value for SCTP delayed acknowledgment (ACK) timer in milliseconds.

**Default**
100 milliseconds

**Range**
1 to 60,000 milliseconds

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 6.2.

**Commitment Level**
Unstable

## _heartbeat_interval Parameter

**Description**
Computes the interval between HEARTBEAT chunks to an idle destination, that is allowed to heartbeat.
An SCTP endpoint periodically sends an HEARTBEAT chunk to monitor the reachability of the idle destinations transport addresses of its peer.

**Default**
30 seconds

**Range**
0 to 86,400 seconds

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 8.3.

**Commitment Level**
Unstable

## _ignore_path_mtu Parameter

**Description**
Enables or disables path MTU discovery.

**Default**
0 (disabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
Enable this parameter if you want to ignore MTU changes along the path. However, doing so might result in IP fragmentation if the path MTU decreases.

**Commitment Level**
Unstable

## _initial_mtu Parameter

**Description**
Determines the initial maximum send size for an SCTP packet including the length of the IP header.

**Default**
1500 bytes

**Range**
68 to 65,535

**Dynamic?**
Yes

**When to Change**
Increase this parameter if the underlying link supports frame sizes that are greater than 1500 bytes.

**Commitment Level**
Unstable

## _initial_out_streams Parameter

**Description**
Controls the maximum number of outbound streams permitted for an SCTP association.

**Default**
32

**Range**
1 to 65,535

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 5.1.1.

**Commitment Level**
Unstable

## _initial_ssthresh Parameter

**Description**
Sets the initial slow start threshold for a destination address of the peer.

**Default**
1,048,576

**Range**
1,024 to 4,294,967,295

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 7.2.1.

**Commitment Level**
Unstable

## _ipv4_ttl Parameter

**Description**
Controls the time to live (TTL) value in the IPv4 header for outbound SCTP messages sent over IPv4. For more information, see the description for ttl Parameter (IPv4).

**Default**
64 bytes

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## _ipv6_hoplimit Parameter

**Description**
Sets the value of the hop limit in the IPv6 header for the outbound SCTP messages sent over IPv6. For more information, see the description for hoplimit Parameter (IPv6).

**Default**
60

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## _maxburst Parameter

**Description**
Sets the limit on the number of segments to be sent in a burst.

**Default**
4

**Range**
2 to 8

**Dynamic?**
Yes

**When to Change**
You do not need to change this parameter. You might change it for testing purposes.

**Commitment Level**
Unstable

## _max_in_streams Parameter

**Description**
Controls the maximum number of inbound streams permitted for an SCTP association.

**Default**
32

**Range**
1 to 65,535

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 5.1.1.

**Commitment Level**
Unstable

## _max_init_retr Parameter

**Description**
Controls the maximum number of attempts an SCTP endpoint should make at resending an INIT chunk. The SCTP endpoint can use the SCTP initiation structure to override this value.

**Default**
8

**Range**
0 to 128

**Dynamic?**
Yes

**When to Change**
The number of INIT retransmissions depend on _pa_max_retr Parameter. Ideally, `_max_init_retr` should be less than or equal to `_pa_max_retr`.

**Commitment Level**
Unstable

## `_new_secret_interval` Parameter

**Description**
Determines when a new secret needs to be generated. The generated secret is used to compute the MAC for a cookie.

**Default**
2 minutes

**Range**
0 to 1,440 minutes

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 5.1.3.

**Commitment Level**
Unstable

## `_pa_max_retr` Parameter

**Description**
Controls the maximum number of retransmissions (over all paths) for an SCTP association. The SCTP association is aborted when this number is exceeded.

**Default**
10

**Range**
1 to 128

**Dynamic?**
Yes

**When to Change**
The maximum number of retransmissions over all paths depend on the number of paths and the maximum number of retransmission over each path. Ideally, `sctp_pa_max_retr` should be set to the sum of _pp_max_retr Parameter over all available paths. For example, if there are 3 paths to the destination and the maximum number of retransmissions over each of the 3 paths is 5, then `_pa_max_retr` should be set to less than or equal to 15. (See the Note in Section 8.2, RFC 2960.)

**Commitment Level**
Unstable

## _pp_max_retr Parameter

**Description**
Controls the maximum number of retransmissions over a specific path. When this number is exceeded for a path, the path (destination) is considered unreachable.

**Default**
5

**Range**
1 to 128

**Dynamic?**
Yes

**When to Change**
Do not change this value to less than 5.

**Commitment Level**
Unstable

## _prsctp_enabled Parameter

**Description**
Enables or disables the partial reliability extension (RFC 3758) to SCTP.

**Default**
1 (enabled)

**Range**
0 (disabled) or 1 (enabled)

**Dynamic?**
Yes

**When to Change**
Disable this parameter if partial reliability is not supported in your SCTP environment.

**Commitment Level**
Unstable

## _rto_initial Parameter

**Description**
Controls the initial retransmission timeout (RTO) in milliseconds for all the destination addresses of the peer.

**Default**
3,000

**Range**
1,000 to 60,000,000

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 6.3.1.

**Commitment Level**
Unstable

## _rto_max Parameter

**Description**
Controls the upper bound for the retransmission timeout (RTO) in milliseconds for all the destination addresses of the peer.

**Default**
60,000

**Range**
1,000 to 60,000,000

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 6.3.1.

**Commitment Level**
Unstable

## _rto_min Parameter

**Description**
Sets the lower bound for the retransmission timeout (RTO) in milliseconds for all the destination addresses of the peer.

**Default**
1,000

**Range**
500 to 60,000

**Dynamic?**
Yes

**When to Change**
Refer to RFC 2960, section 6.3.1.

**Commitment Level**
Unstable

## _shutack_wait_bound Parameter

**Description**
Controls the maximum time, in milliseconds, to wait for a SHUTDOWN ACK after having sent a SHUTDOWN chunk.

**Default**
60,000

**Range**
0 to 300,000

**Dynamic?**
Yes

**When to Change**
Generally, you do not need to change this value. This parameter might be changed in accordance with _rto_max Parameter.

**Commitment Level**
Unstable

## _xmit_lowat Parameter

**Description**
Controls the lower limit on the send window size.

**Default**
8,192

**Range**
8,192 to 1,073,741,824

**Dynamic?**
Yes

**When to Change**
Generally, you do not need to change this value. This parameter sets the minimum size required in the send buffer for the socket to be marked writable.

**Commitment Level**
Unstable

## cwnd-max Parameter

**Description**
Controls the maximum value of the congestion window for an SCTP association.

**Default**
1,048,576

**Range**
128 to 1,073,741,824

**Dynamic?**
Yes

**When to Change**
This parameter does not need to be changed in normal circumstances. If the system needs to communicate with peers far away (round trip time in the order of hundreds of milliseconds) using very fast network (in the order of Gbps), increase the default value to match the bandwidth-delay product to those peers. Note that `max-buf` parameter should also be increased at the same time.

**Commitment Level**
Stable

## `largest_anon_port` Parameter

**Description**
This parameter controls the largest port number SCTP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol but not a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

**Unit**
Port number

**Default**
65,535

**Range**
32,768 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
Stable

## `max-buf` Parameter

**Description**
Controls the maximum send and receive buffer size in bytes. It controls how large the send and receive buffers might be set to by an application that uses `setsockopt`.

**Default**
1,048,576

**Range**
102,400 to 1,073,741,824

**Dynamic?**
Yes

**When to Change**
Increase the value of this parameter to match the network link speed if associations are being made in a high-speed network environment.

**Commitment Level**
able

## `recv-buf` Parameter

**Description**
Defines the default receive buffer size in bytes. See also max-buf Parameter.

**Default**
102,400

**Range**
8,192 to the current value of `max-buf`

**Dynamic?**
Yes

**When to Change**
An application can use `setsockopt (SO_RCVBUF)` to change the individual connection's receive buffer. See the `setsockopt`(3C) man page.

**Commitment Level**
able

## `send-buf` Parameter

**Description**
Defines the default send buffer size in bytes. See also max-buf Parameter.

**Default**
102,400

**Range**
8,192 to the current value of `max-buf`

**Dynamic?**
Yes

**When to Change**
An application can use `setsockopt (SO_SNDBUF)` to change the individual connection's send buffer.

**Commitment Level**
Stable

## `smallest-anon-port` Parameter

**Description**
This parameter controls the smallest port number SCTP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a

specified protocol but not a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

**Unit**
Port number

**Default**
32,768

**Range**
1,024 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
table

# ICMP Tunable Parameters

This section describes parameters related to the Internet control message protocol.

## `_ipv4_ttl` Parameter

**Description**
Controls the time to live (TTL) value in the IPv4 header for outbound ICMP request and error report messages sent over IPv4. For more information, see the description for ttl Parameter (IPv4).

**Default**
64 bytes

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

## `_ipv6_hoplimit` Parameter

**Description**
Sets the value of the hop limit in the IPv6 header for the outbound ICMP request and error report messages sent over IPv6. For more information, see the description for hoplimit Parameter (IPv6).

**Default**
60

**Range**
1 to 255

**Dynamic?**
Yes

**When to Change**
Do not change this value in a normal network environment.

**Commitment Level**
Unstable

# Per-Route Metrics

You can use per-route metrics to associate some properties with IPv4 and IPv6 routing table entries.

For example, a system has two different network interfaces, a fast Ethernet interface and a gigabit Ethernet interface. The system default `recv_maxbuf` is 128,000 bytes. This default is sufficient for the fast Ethernet interface, but may not be sufficient for the gigabit Ethernet interface.

Instead of increasing the system's default for `recv_maxbuf`, you can associate a different default TCP receive window size to the gigabit Ethernet interface routing entry. By making this association, all TCP connections going through the route will have the increased receive window size.

For example, the following is in the routing table (`netstat -rn`), assuming IPv4:

```
Routing Table: IPv4
Destination          Gateway           Flags   Ref     Use     Interface
-------------------- ----------------- ------- ------- ------- ---------
192.0.2.0/27         192.0.2.4/27          U     1       4     net0
192.0.2.32/27        192.0.2.36/27         U     1       4     net1
default              192.0.2.1/27         UG     1       8
```

In this example, do the following:

```
# route change -net 192.0.2.32/27 -recvpipe
        x
```

Then, all connections going to the `192.0.2.32/27` network, which is on the `net1` link, use the receive buffer size *buff-size*, instead of the default `128,000` receive window size.

If the destination is in the *a.b.c.d* network, and no specific routing entry exists for that network, you can add a prefix route to that network and change the metric. For example:

```
# route add -net a.b.c.d 192.0.2.1/27 -netmask a.b.c.d-netmask

# route change -net a.b.c.d -recvpipe
        buff-size
```

Note that the prefix route's gateway is the default router. Then, all connections going to that network use the receive buffer size *y*. If you have more than one interface, use the -ifp argument to specify which interface to use. This way, you can control which interface to use for specific destinations. To verify the metric, use the `route get` command.

# 6
# System Facility Parameters

This chapter describes most of the parameters default values for various system facilities.

For other types of tunable parameters, refer to the following:

- Oracle Solaris kernel tunable parameters – Oracle Solaris Kernel Tunable Parameters
- Oracle Solaris ZFS tunable parameters – Oracle Solaris ZFS Tunable Parameters
- NFS tunable parameters – NFS Tunable Parameters
- Internet Protocol Suite tunable parameters – Internet Protocol Suite Tunable Parameters

## System Default Parameters

The functioning of various system facilities is governed by a set of values that are read by each facility on startup. The values for each facility might be stored in a file for the facility located in the `/etc/default` directory, or in properties of a service instance in the Service Management Facility (SMF) configuration repository. For more information about SMF services and properties, see *Managing System Services in Oracle Solaris 11.4*.

For information about setting power management properties, see *Managing System Information, Processes, and Performance in Oracle Solaris 11.4*.

### `autofs` Property

You can display or configure SMF `autofs` properties by using the `sharectl` command. For example:

```
# sharectl get autofs
timeout=600
automount_verbose=false
automountd_verbose=false
nobrowse=false
trace=0
environment=
# sharectl set -p timeout=200 autofs
```

For details, see the `sharectl`(8) man page.

### `cron` Facility

This facility enables you to disable or enable `cron` logging.

### `devfsadm` File

This file is not currently used.

## fs File

File system administrative commands have a generic and file system-specific portion. If the file system type is not explicitly specified with the -F option, a default is applied. The value is specified in this file. For more information, see the Description section of the default_fs(5) man page.

## ftp Facility

This facility enables you to set the ls command behavior to the RFC 959 NLST command. The default ls behavior is the same as in the previous Oracle Solaris release.

For details, see the ftp(1) man page.

## inetinit Facility

This facility enables you to configure TCP sequence numbers and to enable or disable support for 6to4 relay routers.

## init Service

System initialization properties are now part of the following SMF service:

```
svc:/system/environment:init
```

You can display and configure system initialization properties, such as TZ and LANG, by using similar syntax:

```
# svccfg -s svc:/system/environment:init
svc:/system/environment:init> setprop
Usage:  setprop pg/name = [type:] value
setprop pg/name = [type:] ([value...])

Set the pg/name property of the currently selected entity.  Values may be
enclosed in double-quotes.  Value lists may span multiple lines.

svc:/system/environment:init> listprop
umask                            application
umask/umask                     astring     022
umask/value_authorization       astring     solaris.smf.value.environment
environment                      application
environment/LANG                astring
environment/LC_ALL              astring
.
.
.
```

For more information, see the FILES section of the init(8) man page.

## ipsec Facility

This facility enables you to configure network security parameters, such as IKE daemon debugging information.

## kbd Configuration Properties

Keyboard configuration properties are now part of the following SMF service:

```
svc:/system/keymap:default
```

You display and configure keyboard properties by using similar syntax:

```
# svccfg -s svc:/system/keymap:default
svc:/system/keymap:default> setprop
Usage:  setprop pg/name = [type:] value
setprop pg/name = [type:] ([value...])

Set the pg/name property of the currently selected entity.  Values may be
enclosed in double-quotes.  Value lists may span multiple lines.

svc:/system/keymap:default> listprop
general                          framework
general/complete                 astring
general/enabled                  boolean     false
keymap                           system
keymap/console_beeper_freq       integer     900
keymap/kbd_beeper_freq           integer     2000
keymap/keyboard_abort            astring     enable
keymap/keyclick                  boolean     false
.
.
.
```

For more information, see the kbd(1) man page.

## keyserv Facility

For details, see the /etc/default/keyserv information in the FILES section of the keyserv(8) man page.

## login Facility

For details, see the /etc/default/login information in the FILES section of the login(1) man page.

## mpathd Facility

This facility enables you to set in.mpathd configuration parameters.

For details, see the in.mpathd(8) man page.

## nfs Properties

You can display or configure SMF NFS properties by using the sharectl command. For example:

```
# sharectl get nfs
servers=1024
lockd_listen_backlog=32
```

```
lockd_servers=1024
lockd_retransmit_timeout=5
grace_period=90
server_versmin=2
server_versmax=4
client_versmin=2
client_versmax=4
server_delegation=on
nfsmapid_domain=
# sharectl set -p grace_period=60 nfs
```

For details, see the nfs(5) man page.

## nfslogd Log File

For details, see the Description section of the *nfslogd 8* man page.

## nss Facility

This facility enables you to configure initgroups lookup parameters.

For details, see the nss(5) man page.

## passwd Facility

For details, see the /etc/default/passwd information in the FILES section of the passwd(1) man page.

## su Facility

For details, see the /etc/default/su information in the FILES section of the su(8) man page.

## syslog Facility

For details, see the /etc/default/syslogd information in the FILES section of the syslogd(8) man page.

## tar Facility

For a description of the -f function modifier, see the tar(1) man page.

If the TAPE environment variable is not present and the value of one of the arguments is a number and -f is not specified, the number matching the archive*N* string is looked up in the /etc/default/tar file. The value of the archive*N* string is used as the output device with the blocking and size specifications from the file.

For example:

```
% tar -c 2 /tmp/*
```

This command writes the output to the device specified as archive2 in the /etc/default/tar file.

## telnetd Facility (Deprecated)

This file identifies the default BANNER that is displayed upon a telnet connection.

## utmpd Daemon

The utmpd daemon monitors /var/adm/utmpx (and /var/adm/utmp in earlier Oracle Solaris versions) to ensure that utmp entries inserted by non-root processes by pututxline are cleaned up on process termination.

Two entries in /etc/default/utmpd are supported:

- SCAN_PERIOD – The number of seconds that utmpd sleeps between checks of /proc to see if monitored processes are still alive. The default is 300.
- MAX_FDS – The maximum number of processes that utmpd attempts to monitor. The default value is 4096 and should never need to be changed.

# A
# System Check Script

## Confirming Flush Behavior on the System

You can tune your system's ZFS and flash storage flush behavior by updating the `physical-block-size` and `cache-nonvolatile` property values in the `sd.conf` file. See Ensuring Proper Cache Flush Behavior for Flash and NVRAM Storage Devices.

The following `sdpropcheck.sh` script enables you to confirm that the flush behavior you specified in `sd.conf` is reflected for the specified disk on the running system:

```
#!/bin/ksh
#
# A script to check that cache-nonvolatile and physical-block-size properties in
# sd.conf have been successfully applied to the specified disk that is handled by
# sd(4D) driver. These sd.conf properties match the un_f_suppress_cache_flush
# and un_phy_blocksize sd_lun structure elements on the running system,
# respectively.
#

if [[ $# -ne 1 ]]; then
    echo "Compare the physical-block-size and cache-nonvolatile property values
of a disk on a running system with the values specified in the sd.conf file."
    echo "Usage: $0 cXtYdZ"
    exit 1;
fi

#
# Use the iostat output to obtain the device instance of a disk that uses the
# sd driver.
#

sd=`iostat -x $1 2>&1 | grep sd | nawk '{print $1}' | sed s/sd//`
printf "Values for %s on the running system:\n" $1

#
# Use mdb to obtain the value of the sd_lun structure's un_phy_blocksize element
# in decimal format, which is the form used by the associated physical-block-size
# property value in sd.conf.
#

printf " - physical-block-size (un_phy_blocksize):"
echo '*sd_state::softstate 0t'$sd' | ::print -H struct sd_lun un_phy_blocksize' \
| mdb --kernel | cut -d'=' -f 2-

#
# Use mdb to obtain the sd_lun structure's un_f_suppress_cache_flush element
# value as a Boolean string (true or false), which is the form used by the
# associated cache-nonvolatile property value in sd.conf.
#

printf " - cache-nonvolatile (un_f_suppress_cache_flush): "
```

```
cache_nv=$(echo '*sd_state::softstate 0t'$sd' | ::print struct sd_lun \
un_f_suppress_cache_flush' | mdb --kernel | cut -d'=' -f 2-)
if [ $cache_nv == 0 ]; then
        echo false
else
        echo true
fi

#
# Show the physical-block-size and cache-nonvolatile settings in the sd.conf
# file.
#

echo "\nValue of the physical-block-size property in the sd.conf file:"
grep "physical-block-size" /kernel/drv/sd.conf /etc/driver/drv/sd.conf

echo "\nValue of the cache-nonvolatile property in the sd.conf file:"
grep "cache-nonvolatile" /kernel/drv/sd.conf /etc/driver/drv/sd.conf
```

The `sdpropcheck.sh` script uses the `iostat` and `mdb` commands to obtain configuration information about the disk that is controlled by the `sd` driver.

The `mdb` command obtains the values for the `sd_lun` structure's `un_phy_blocksize` and `un_f_suppress_cache_flush` elements. These elements correspond to the `physical-block-size` and `cache-nonvolatile` properties in the `sd.conf` file, respectively.

Note that the form of the structure element values differs from the `sd.conf` property values. The `un_phy_blocksize` element specifies a hexidecimal value rather than a decimal value. The `un_f_suppress_cache_flush` element specifies a Boolean value rather than `true` or `false`.

The following `sdpropcheck.sh` script output shows a block size of `512` and the cache flush value as `false`. This output shows that neither the `physical-block-size` property nor the `cache-nonvolatile` property is set explicitly in the `sd.conf` file, so the default values of these properties are used.

```
# ./sdpropcheck.sh c0t5000CCA02D101AD0d0
Values for c0t5000CCA02D101AD0d0 on the running system:
 - physical-block-size (un_phy_blocksize): 512
 - cache-nonvolatile (un_f_suppress_cache_flush): false

Value of the physical-block-size property in the sd.conf file:

Value of the cache-nonvolatile property in the sd.conf file:
```

If you set the `physical-block-size` property value to `4096` and set the `cache-nonvolatile` property value to `true`, the `sdpropcheck.sh` script produces the following output:

```
# ./sdpropcheck.sh c0t5000CCA02D101AD0d0
Values for c0t5000CCA02D101AD0d0 on the running system:
 - physical-block-size (un_phy_blocksize): 4096
 - cache-nonvolatile (un_f_suppress_cache_flush): true

Value of the physical-block-size property in the sd.conf file:
/etc/driver/drv/sd.conf:sd-config-list="ATA 2E256-TU2-510B00","disksort:false,
cache-nonvolatile:true, physical-block-size:4096";

Value of the cache-nonvolatile property in the sd.conf file:
```

```
/etc/driver/drv/sd.conf:sd-config-list="ATA 2E256-TU2-510B00","disksort:false,
cache-nonvolatile:true, physical-block-size:4096";
```

# Index

## Symbols


_addip_enabled, *5-34*
_addrs_per_if, *5-2*
_arp_defend_interval, *5-7*
_arp_defend_period, *5-7*
_arp_defend_rate, *5-8*
_arp_fastprobe_count, *5-8*
_arp_fastprobe_interval, *5-9*
_arp_probe_count, *5-9*
_arp_probe_interval, *5-10*
_conn_req_max_q, *5-14*
_conn_req_max_q0, *5-14*
_conn_req_min, *5-15*
_cookie_life, *5-34*
_defend_interval, *5-10*
_deferred_ack_interval, *5-15*, *5-34*
_deferred_acks_max, *5-16*
_dup_recovery, *5-10*
_forwarding_src_routed, *5-2*
_heartbeat_interval, *5-35*
_icmp_err_burst, *5-3*
_icmp_err_interval, *5-3*
_icmp_return_data_bytes, *5-13*
_ignore_path_mtu, *5-35*
_initial_mtu, *5-36*
_initial_out_streams, *5-36*
_initial_ssthresh, *5-36*
_ip_abort_interval, *5-27*
_ipv4_ttl (ICMP), *5-45*
_ipv4_ttl (TCP), *5-16*
_ipv4_ttl (UDP), *5-31*
_ipv4_ttl SCTP, *5-37*
_ipv6_hoplimit (ICMP), *5-45*
_ipv6_hoplimit (SCTP), *5-37*
_ipv6_hoplimit (TCP), *5-17*
_ipv6_hoplimit (UDP), *5-31*
_keepalive_interval, *5-27*
_local_dack_interval, *5-17*
_local_dacks_max, *5-17*
_local_slow_start_initial, *5-18*
_max_defend, *5-11*
_max_in_streams, *5-38*
_max_init_retr, *5-38*
_max_temp_defend, *5-11*
_maxburst, *5-38*
_ndp_defend_interval, *5-7*
_ndp_defend_period, *5-7*
_ndp_defend_rate, *5-8*
_new_secret_interval, *5-39*
_pa_max_retr, *5-39*
_pathmtu_interval, *5-13*
_policy_mask, *5-3*
_pp_max_retr, *5-40*
_prsctp_enabled, *5-40*
_recv_hiwat_minmss, *5-28*
_respond_to_echo_broadcast, *5-4*
_respond_to_echo_multicast, *5-4*
_rev_src_routes, *5-18*
_rexmit_interval_extra, *5-28*
_rexmit_interval_initial, *5-29*
_rexmit_interval_max, *5-29*
_rexmit_interval_min, *5-30*
_rst_sent_rate, *5-19*
_rst_sent_rate_enabled, *5-19*
_rto_max, *5-40*, *5-41*
_rto_min, *5-41*
_shutack_wait_bound, *5-42*
_slow_start_after_idle, *5-20*
_slow_start_initial, *5-20*
_time_wait_interval, *5-20*
_tstamp_always, *5-21*
_tstamp_if_wscale, *5-30*
_wscale_always, *5-21*
_xmit_lowat, *5-42*


## A


arp-publish-count, *5-12*
arp-publish-interval, *5-12*
autofs, *6-1*
autoup, *2-6*

## C

cron, 6-1
cwnd-max (SCTP), 5-42
cwnd-max (TCP), 5-22

## D

dax_stats_flags, 2-28
ddi_msix_alloc_limit parameter, 2-28
default_stksize, 2-2
default_tsb_size, 2-51
desfree, 2-15
dnlc_dir_enable, 2-40
dnlc_dir_max_size, 2-41
dnlc_dir_min_size, 2-42
dnlc_dircache_percent, 2-42
doiflush, 2-7
dopageflush, 2-8

## E

ecn, 5-22
enable_tsb_rss_sizing, 2-52

## F

fastscan, 2-17
fs, 6-2
fsflush, 2-8
ftp, 6-2

## H

handspreadpages, 2-17
hires_tick, 2-50
hoplimit (IPv6), 5-4
hostmodel, 5-5

## I

inetinit, 6-2
init, 6-2
intr_force, 2-32
intr_throttling, 2-33
ip_squeue_fanout, 2-31
ip_squeue_worker_wait, 2-30
ipcl_conn_hash_size, 2-31
ipsec, 6-2

## K

kbd, 6-3
keyserv, 6-3
kmem_flags, 2-26
kmem_stackinfo, 2-27

## L

largest-anon-port (SCTP), 5-43
largest-anon-port (TCP), 5-23
largest-anon-port (UDP), 5-31
lgrp_mem_pset_aware, 2-54
logevent_max_q_sz, 2-3
login, 6-3
lotsfree, 2-18
lpg_alloc_prefer, 2-55
lwp_default_stksize, 2-3

## M

max_nprocs, 2-10
max-buf (SCTP), 5-43
max-buf (TCP), 5-24
max-buf (UDP), 5-32
maxpgio, 2-19
maxphys, 2-37
maxpid, 2-13
maxuprc, 2-11
maxusers, 2-11
min_percent_cpu, 2-20
minfree, 2-20
moddebug, 2-29
mpathd, 6-3
mr_enable, 2-33

## N

ncsize, 2-43
ndp-unsolicit-count, 5-12
ndp-unsolicit-interval, 5-12
nfs_max_threads, 4-16
nfs:nacache, 4-21
nfs:nfs_allow_preepoch_time, 4-1
nfs:nfs_async_clusters, 4-2
nfs:nfs_async_timeout, 4-5
nfs:nfs_cots_timeo, 4-7
nfs:nfs_disable_rddir_cache, 4-9
nfs:nfs_do_symlink_cache, 4-10
nfs:nfs_dynamic, 4-12
nfs:nfs_lookup_neg_cache, 4-14
nfs:nfs_nra, 4-22

nfs:nfs_shrinkreaddir, 4-26
nfs:nfs_write_error_interval, 4-27
nfs:nfs_write_error_to_cons_only, 4-28
nfs:nfs3_async_clusters, 4-3
nfs:nfs3_bsize, 4-6
nfs:nfs3_cots_timeo, 4-8
nfs:nfs3_do_symlink_cache, 4-11
nfs:nfs3_dynamic, 4-12
nfs:nfs3_jukebox_delay, 4-13
nfs:nfs3_lookup_neg_cache, 4-14
nfs:nfs3_max_threads, 4-17
nfs:nfs3_max_transfer_size, 4-18
nfs:nfs3_max_transfer_size_clts, 4-20
nfs:nfs3_max_transfer_size_cots, 4-20
nfs:nfs3_nra, 4-23
nfs:nfs3_pathconf_disable_cache, 4-25
nfs:nfs3_shrinkreaddir, 4-26
nfs:nfs4_async_clusters, 4-4
nfs:nfs4_bsize, 4-6
nfs:nfs4_cots_timeo, 4-8
nfs:nfs4_do_symlink_cache, 4-11
nfs:nfs4_lookup_neg_cache, 4-15
nfs:nfs4_max_threads, 4-17
nfs:nfs4_max_transfer_size, 4-19
nfs:nfs4_nra, 4-23
nfs:nrnode, 4-24
nfslogd, 6-4
nfssrv:rfs_write_async, 4-29
ngroups_max, 2-12
noexec_user_stack, 2-4
nss, 6-4
nstrpush, 2-48

## P

pageout_reserve, 2-21
pages_before_pager, 2-22
pages_pp_maximum, 2-23
passwd, 6-4
physmem, 2-5
pidmax, 2-13
pt_cnt, 2-46
pt_max_pty, 2-47
pt_pctofmem, 2-47

## R

recv-buf (SCTP), 5-44
recv-buf (TCP), 5-24
recv-buf (UDP), 5-32
recv-multicast-scaling, 5-5
reserved_procs, 2-14
rlim_fd_cur, 2-38
rlim_fd_max, 2-39
rlim_fd_sys, 2-40
rpcmod:clnt_idle_timeout, 4-31
rpcmod:clnt_max_conns, 4-30
rpcmod:svc_idle_timeout, 4-31
rx_copy_threshold, 2-34
rx_limit_per_intr, 2-34
rx_queue_number, 2-35
rx_ring_size, 2-35

## S

sack, 5-24
send-buf (SCTP), 5-44
send-buf (TCP), 5-25
send-buf (UDP), 5-33
send-redirects, 5-6
slowscan, 2-23
smallest-anon-port (SCTP), 5-44
smallest-anon-port (TCP), 5-25
smallest-anon-port (UDP), 5-33
strmsgsz, 2-48, 2-49
su , 6-4
syslog , 6-4

## T

tar, 6-4
tcp_cwnd_normal, 5-26
throttlefree, 2-24
timer_max, 2-51
tmpfs_maxkmem, 2-44
tmpfs_minfree, 2-44
tsb_alloc_hiwater, 2-52
tsb_rss_factor, 2-53
ttl (IPv4), 5-6
tune_t_fsflushr, 2-9
tune_t_minarmem, 2-25
tx_copy_threshold, 2-36
tx_queue_number, 2-36
tx_ring_size, 2-37

## U

utmpd, 6-5

## Z

zfs_arc_max, 3-3
zfs_arc_max_percent, 3-4
zfs_arc_min, 3-3

zfs_prefetch_disable, 3-5