Ce logiciel et la documentation qui l’accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d’utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le dé assembler ou de le décompiler, excepté à des fins d’interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu’elles soient exemptes d’erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l’accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l’utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s’applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d’applications de gestion des informations. Ce logiciel ou matériel n’est pas conçu ni n’est destiné à être utilisé dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d’applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l’utilisation de ce logiciel ou matériel pour le cadre d’applications dangereuses.

Oracle et Java sont des marques déposées d’Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d’autres propriétaires qu’Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d’Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées d’SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d’Advanced Micro Devices. UNIX est une marque déposée d’The Open Group.

Ce logiciel ou matériel et la documentation qui l’accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l’accès à des contenus, produits ou services tiers, ou à leur utilisation.
# Contents

<table>
<thead>
<tr>
<th>Preface</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
</tr>
</tbody>
</table>

1 Overview of Oracle Solaris System Tuning ...................................................................................... 17
What's New in Oracle Solaris System Tuning? .................................................................................. 17
Oracle Solaris System Tuning in the Solaris 10 Release ................................................................. 19
  Default Stack Size .............................................................................................................................. 19
  System V IPC Configuration .............................................................................................................. 19
  NFSv4 Parameters ............................................................................................................................... 21
New and Changed TCP/IP Parameters ............................................................................................... 21
SPARC: Translation Storage Buffer (TSB) Parameters ........................................................................ 23
SCTP Tunable Parameters .................................................................................................................. 23
Tuning an Oracle Solaris System ......................................................................................................... 24
Tuning Format of Tunable Parameters Descriptions .......................................................................... 24
Tuning the Oracle Solaris Kernel ......................................................................................................... 26
/etc/system File .................................................................................................................................. 26
/kmdb Command .................................................................................................................................. 27
/mdb Command .................................................................................................................................. 28
Special Oracle Solaris tune and var Structures .................................................................................. 29
Viewing Oracle Solaris System Configuration Information .................................................................. 29
sysdef Command ............................................................................................................................... 29
kstat Utility ......................................................................................................................................... 30

2 Oracle Solaris Kernel Tunable Parameters ...................................................................................... 31
Where to Find Tunable Parameter Information .................................................................................. 32
General Kernel and Memory Parameters .......................................................................................... 32
  physmem .......................................................................................................................................... 32
  default_stksize ................................................................................................................................. 33
lwp_default_stksize ................................................................................................................34
logevent_max_q_sz ........................................................................................................................35
segkpsize ......................................................................................................................................35
noexec_user_stack ....................................................................................................................36
fsflush and Related Parameters .......................................................................................................37
fsflush .........................................................................................................................................37
tune_t_fsflushr ........................................................................................................................38
autoup ...........................................................................................................................................38
dopageflush .................................................................................................................................39
doiflush .......................................................................................................................................40
Process-Sizing Parameters .................................................................................................................41
maxusers .......................................................................................................................................41
reserved_procs ..........................................................................................................................42
pidmax .......................................................................................................................................42
max_nprocs ...................................................................................................................................43
maxuprc ...........................................................................................................................................44
ngroups_max ................................................................................................................................44
Paging-Related Parameters ..............................................................................................................45
lotsfree .......................................................................................................................................46
desfree .........................................................................................................................................47
minfree .........................................................................................................................................48
throttlefree .................................................................................................................................49
pageout_reserve ........................................................................................................................50
pages_pp_maximum ......................................................................................................................51
tune_t_minarmem ........................................................................................................................52
fastscan .......................................................................................................................................52
slowscan .......................................................................................................................................53
min_percent_cpu ........................................................................................................................54
handspreadpages ..........................................................................................................................54
pages_before_pager .......................................................................................................................55
maxpgio .......................................................................................................................................56
Swapping-Related Parameters ...........................................................................................................56
swapfs_reserve ............................................................................................................................57
swapfs_minfree ...........................................................................................................................57
Kernel Memory Allocator .................................................................................................................58
kmem_flags ................................................................................................................................58
General Driver Parameters ................................................................. 60
  moddebug ..................................................................................... 60
  ddi_msix_alloc_limit ................................................................. 61
General I/O Parameters ......................................................................... 62
  maxphys ...................................................................................... 62
  rlim_fd_max ............................................................................. 63
  rlim_fd_cur ............................................................................. 63
General File System Parameters ......................................................... 64
  ncsizE ................................................................. 64
  rstchown ............................................................................... 65
  dnlc_dir_enable ................................................................. 66
  dnlc_dir_min_size ............................................................. 66
  dnlc_dir_max_size ............................................................. 67
UFS Parameters ................................................................................ 67
  bufhwm and bufhwm_pct ...................................................... 67
  ndquot .................................................................................. 69
  ufs_ninode ......................................................................... 70
  ufs_WRITES ......................................................................... 71
  ufs_LW and ufs_HW ........................................................... 72
  freebehind ........................................................................... 73
  smallfile ............................................................................ 73
  ufs_delete_hiwat ............................................................. 74
TMPFS Parameters ........................................................................... 75
  tmpfs:tmpfs_maxkmem ...................................................... 75
  tmpfs:tmpfs_minfree ......................................................... 75
Pseudo Terminals ................................................................................ 76
  pt_cnt .............................................................................. 77
  pt_pctofmem .................................................................... 77
  pt_max_pty........................................................................ 78
STREAMS Parameters ........................................................................ 79
  nstrpush ........................................................................... 79
  strmsgsz ......................................................................... 79
  strctlsz ........................................................................... 80
System V Message Queues ................................................................. 80
System V Semaphores ......................................................................... 81
System V Shared Memory ................................................................. 81
3 Oracle Solaris ZFS Tunable Parameters .................................................................93
   Where to Find Tunable Parameter Information ..................................................93
   Tuning ZFS Considerations .............................................................................94
   ZFS ARC Parameters .....................................................................................94
      zfs_arc_min ...............................................................................................94
      zfs_arc_max .............................................................................................95
   ZFS File-Level Prefetch ................................................................................95
      zfs_prefetch_disable ..................................................................................95
   ZFS Device I/O Queue Depth .........................................................................96
      zfs_vdev_max_pending .............................................................................96
   ZFS and Cache Flushing ...............................................................................97
      zfs_nocacheflush ......................................................................................97
   ZFS Metadata Compression .........................................................................98
      zfs_mdcomp_disable ...............................................................................99
   Tuning ZFS for Database Products ...............................................................99
4 NFS Tunable Parameters ................................................................. 105
   Where to Find Tunable Parameter Information .......................... 105
   Tuning the NFS Environment ..................................................... 105
   NFS Module Parameters .......................................................... 106
      nfs:nfs3_pathconf_disable_cache ......................................... 106
      nfs:nfs4_pathconf_disable_cache ......................................... 106
      nfs:nfs_allow_preepoch_time ............................................. 107
      nfs:nfs_cots_timeo .......................................................... 108
      nfs:nfs3_cots_timeo ......................................................... 108
      nfs:nfs4_cots_timeo ......................................................... 109
      nfs:nfs_do_symlink_cache ................................................. 110
      nfs:nfs3_do_symlink_cache .............................................. 110
      nfs:nfs4_do_symlink_cache .............................................. 111
      nfs:nfs_dynamic ............................................................ 111
      nfs:nfs3_dynamic .......................................................... 112
      nfs:nfs_lookup_neg_cache ................................................. 112
      nfs:nfs3_lookup_neg_cache .............................................. 113
      nfs:nfs4_lookup_neg_cache .............................................. 114
      nfs:nfs_max_threads ....................................................... 115
      nfs:nfs3_max_threads ....................................................... 115
      nfs:nfs4_max_threads ....................................................... 116
      nfs:nfs_nra ................................................................. 117
      nfs:nfs3_nra ............................................................... 117
      nfs:nfs4_nra ............................................................... 118
      nfs:nrrnode .................................................................. 119
      nfs:nfs_shrinkreaddir ....................................................... 120
      nfs:nfs3_shrinkreaddir .................................................... 121
      nfs:nfs_write_error_interval .......................................... 121
      nfs:nfs_write_error_to_cons_only .................................. 122
      nfs:nfs_disable_rddir_cache ........................................... 122
      nfs:nfs3_bsize .............................................................. 123
      nfs:nfs4_bsize .............................................................. 124
Contents

nfs:nfs_async_clusters ................................................................. 125
nfs:nfs3_async_clusters ............................................................. 125
nfs:nfs4_async_clusters ............................................................. 126
nfs:nfs_async_timeout ............................................................... 127
nfs:nacache ............................................................................. 128
nfs:nfs3_jukebox_delay ............................................................. 129
nfs:nfs3_max_transfer_size ........................................................ 129
nfs:nfs4_max_transfer_size ........................................................ 130
nfs:nfs3_max_transfer_size_clts ................................................ 131
nfs:nfs3_max_transfer_size_cots ............................................... 132
rpcmod Module Parameters ....................................................... 132
  rpcmod:clnt_max_conns ......................................................... 132
  rpcmod:clnt_idle_timeout ..................................................... 133
  rpcmod:svc_idle_timeout ..................................................... 134
  rpcmod:svc_default_stksize ................................................. 134
  rpcmod:maxdupreqs ............................................................. 135
  rpcmod:cotsmaxdupreqs ....................................................... 136

5 Internet Protocol Suite Tunable Parameters .................................. 137
Where to Find Tunable Parameter Information .................................. 137
Overview of Tuning IP Suite Parameters ......................................... 137
IP Suite Parameter Validation ....................................................... 138
Internet Request for Comments (RFCs) ......................................... 138
IP Tunable Parameters ................................................................ 138
  ip_icmp_err_interval and ip_icmp_err_burst ................................ 138
  ip_respond_to_echo_broadcast and ip6_respond_to_echo_multicast ........................................ 139
  ip_send_redirects and ip6_send_redirects ................................ 139
  ip_forward_src_routed and ip6_forward_src_routed .................................................. 139
  ip_addrs_per_if .................................................................. 140
  ip_strict_dst_multihoming and ip6_strict_dst_multihoming ................................................ 140
  ip_multidata_outbound ......................................................... 141
  ip_squeue_fanout ................................................................. 141
  ip_soft_rings_cnt ............................................................... 142
  ip_pmtu_min .................................................................... 143
IP Tunable Parameters With Additional Cautions ............................. 143
TCP Tunable Parameters .................................................................................................................. 144
  tcp_defered_ack_interval .................................................................................................................. 144
  tcp_local_dack_interval ................................................................................................................... 145
  tcp_defered_acks_max ...................................................................................................................... 145
  tcp_local_daks_max ......................................................................................................................... 146
  tcp_wscale_always .......................................................................................................................... 146
  tcp_tstamp_always .......................................................................................................................... 147
  tcp_xmit_hiwat ............................................................................................................................... 147
  tcp_recv_hiwat ............................................................................................................................... 147
  tcp_max_buf ...................................................................................................................................... 148
  tcp_cwnd_max .................................................................................................................................. 148
  tcp_slow_start_initial ...................................................................................................................... 149
  tcp_slow_start_after_idle .................................................................................................................. 149
  tcp_sack_permitted .......................................................................................................................... 149
  tcp_rev_src_routes .......................................................................................................................... 149
  tcp_time_wait_interval ..................................................................................................................... 150
  tcp_ecn_permitted ............................................................................................................................ 151
  tcp_conn_req_max_q .......................................................................................................................... 152
  tcp_conn_req_max_q0 ......................................................................................................................... 152
  tcp_conn_req_min ............................................................................................................................ 153
  tcp_rst_sent_rate_enabled .................................................................................................................. 153
  tcp_rst_sent_rate ............................................................................................................................. 154
  tcp_mdt_max_pbufs ........................................................................................................................... 154
  tcp_naglim_def ................................................................................................................................. 155
  tcp_smallest_anon_port ..................................................................................................................... 155
  tcp_largest_anon_port ....................................................................................................................... 156
TCP/IP Parameters Set in the /etc/system File .................................................................................. 156
TCP Parameters With Additional Cautions ......................................................................................... 157
UDP Tunable Parameters ..................................................................................................................... 161
  udp_xmit_hiwat ............................................................................................................................... 161
  udp_recv_hiwat ............................................................................................................................... 161
  udp_smallest_anon_port .................................................................................................................... 162
  udp_largest_anon_port ...................................................................................................................... 162
  udp_do_checksum ............................................................................................................................ 163
UDP Parameter With Additional Caution ............................................................................................ 163
IPQoS Tunable Parameter .................................................................................................................. 163
ip_policy_mask .......................................................................................................................... 163

SCTP Tunable Parameters .............................................................................................................. 164
sctp_max_init_retr ...................................................................................................................... 164
sctp_pa_max_retr ...................................................................................................................... 165
sctp_pp_max_retr ...................................................................................................................... 165
sctp_cwnd_max ....................................................................................................................... 165
sctp_ipv4_ttl .......................................................................................................................... 166
sctp_heartbeat_interval ........................................................................................................... 166
sctp_new_secret_interval ......................................................................................................... 166
sctp_initial_mtu ....................................................................................................................... 167
sctp_deferred_ack_interval ..................................................................................................... 167
sctp_ignore_path_mtu ................................................................................................................ 167
sctp_initial_ssthresh ................................................................................................................ 168
sctp_xmit_hiwat ...................................................................................................................... 168
sctp_xmit_lowat ...................................................................................................................... 168
sctp_recv_hiwat ...................................................................................................................... 169
sctp_max_buf .......................................................................................................................... 169
sctp_ipv6_holimit ..................................................................................................................... 169
sctp_rto_min ............................................................................................................................ 170
sctp_rto_max ............................................................................................................................ 170
sctp_rto_initial ....................................................................................................................... 170
sctp_cookie_life ...................................................................................................................... 171
sctp_max_in_streams .............................................................................................................. 171
sctp_initial_out_streams ........................................................................................................ 171
sctp_shutack_wait_bound ......................................................................................................... 171
sctp_maxburst .......................................................................................................................... 172
sctp_addip_enabled .................................................................................................................. 172
sctp_prsctp_enabled ................................................................................................................. 172
sctp_smallest_anon_port ......................................................................................................... 173
sctp_largest_anon_port .......................................................................................................... 173

Per-Route Metrics .................................................................................................................... 174

6 System Facility Parameters .................................................................................................... 175
System Default Parameters ...................................................................................................... 176
autofs ......................................................................................................................................... 176
cron ................................................................. 176
devfsadm ......................................................... 176
dhcpagent ......................................................... 176
fs ........................................................................... 177
ftp ......................................................................... 177
inetinit .............................................................. 177
init ................................................................. 177
ipsec ............................................................... 177
kbd ................................................................. 177
keyserv ........................................................ 178
login ............................................................... 178
lu ................................................................. 178
mpathd ......................................................... 178
nfs .................................................................... 178
nfslogd .......................................................... 179
nss ................................................................. 179
passwd .......................................................... 179
power ............................................................ 179
rpc.nisd ........................................................ 179
su ................................................................. 179
syslog ........................................................... 179
sys-suspend ...................................................... 179
tar ............................................................... 179
telnetd .......................................................... 180
utmpd .......................................................... 180
yppasswd ....................................................... 180

A Tunable Parameters Change History .......................................................... 181
Kernel Parameters .................................................................................. 181
General Kernel and Memory Parameters (Oracle Solaris 10) ................. 181
Paging-Related Parameters ............................................................... 182
Process-Sizing Tunables ............................................................... 182
UFS Parameter ........................................................................... 182
General Driver Parameter .......................................................... 182
General I/O Tunable Parameters .......................................................... 183
Preface

The Oracle Solaris Tunable Parameters Reference Manual provides reference information about Oracle Solaris OS kernel and network tunable parameters. This manual does not provide tunable parameter information about desktop systems or Java environments.

This manual contains information for both SPARC based and x86 based systems.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the Oracle Solaris Hardware Compatibility List at http://www.oracle.com/webfolder/technetwork/hcl/index.html. This document cites any implementation differences between the platform types.

In this document these x86 terms mean the following:
- “x86” refers to the larger family of 64-bit and 32-bit x86 compatible products.
- “x64” relates specifically to 64-bit x86 compatible CPUs.
- “32-bit x86” points out specific 32-bit information about x86 based systems.

Who Should Use This Book

This book is intended for experienced Oracle Solaris system administrators who might need to change kernel tunable parameters in certain situations. For guidelines on changing Oracle Solaris tunable parameters, refer to “Tuning an Oracle Solaris System” on page 24.

How This Book Is Organized

The following table describes the chapters and appendixes in this book.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1, &quot;Overview of Oracle Solaris System Tuning&quot;</td>
<td>An overview of tuning an Oracle Solaris system. Also provides a description of the format used in the book to describe the kernel tunables.</td>
</tr>
</tbody>
</table>
Chapter 2, "Oracle Solaris Kernel Tunable Parameters"  
A description of Oracle Solaris kernel tunables such as kernel memory, file system, process size, and paging parameters.

Chapter 3, "Oracle Solaris ZFS Tunable Parameters"  
A description of Oracle Solaris ZFS tunable parameters. This chapter also includes ZFS tuning information for database products.

Chapter 4, "NFS Tunable Parameters"  
A description of NFS tunables such as caching symbolic links, dynamic retransmission, and RPC security parameters.

Chapter 5, "Internet Protocol Suite Tunable Parameters"  
A description of TCP/IP tunables such as IP forwarding, source routing, and buffer-sizing parameters.

Chapter 6, "System Facility Parameters"  
A description of parameters used to set default values of certain system facilities. Changes are made by modifying files in the /etc/default directory.

Appendix A, "Tunable Parameters Change History"  
A history of parameters that have changed or are now obsolete.

Appendix B, "Revision History for This Manual"  
A history of this manual’s revisions including the current Oracle Solaris release.

---

**Other Resources for Oracle Solaris Tuning Information**

This table describes other resources for Oracle Solaris tuning information.

<table>
<thead>
<tr>
<th>Tuning Resource</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-depth technical white papers</td>
<td><a href="http://www.oracle.com/technetwork/server-storage/solaris/overview/index.html">http://www.oracle.com/technetwork/server-storage/solaris/overview/index.html</a></td>
</tr>
</tbody>
</table>

---

**Access to Oracle Support**

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

<table>
<thead>
<tr>
<th>Typeface</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories, and onscreen computer output</td>
<td>Edit your .login file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use <code>ls -a</code> to list all files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>machine_name% you have mail.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>What you type, contrasted with onscreen computer output</td>
<td>machine_name% su</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Password:</td>
</tr>
<tr>
<td>aabbcc123</td>
<td>Placeholder: replace with a real name or value</td>
<td>The command to remove a file is <code>rm filename</code>.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Book titles, new terms, and terms to be emphasized</td>
<td>Read Chapter 6 in the User’s Guide.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A cache is a copy that is stored locally.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not save the file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: Some emphasized items appear bold online.</td>
</tr>
</tbody>
</table>

Shell Prompts in Command Examples

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash shell, Korn shell, and Bourne shell</td>
<td>$</td>
</tr>
<tr>
<td>Bash shell, Korn shell, and Bourne shell for superuser</td>
<td>#</td>
</tr>
<tr>
<td>C shell</td>
<td>machine_name%</td>
</tr>
<tr>
<td>C shell for superuser</td>
<td>machine_name#</td>
</tr>
</tbody>
</table>
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.

- “What’s New in Oracle Solaris System Tuning?” on page 17
- “Oracle Solaris System Tuning in the Solaris 10 Release” on page 19
- “Tuning an Oracle Solaris System” on page 24
- “Tuning Format of Tunable Parameters Descriptions” on page 24
- “Tuning the Oracle Solaris Kernel” on page 26
- “Special Oracle Solaris tune and var Structures” on page 29
- “Viewing Oracle Solaris System Configuration Information” on page 29
- “kstat Utility” on page 30

What’s New in Oracle Solaris System Tuning?

This section describes new or changed parameters in the Oracle Solaris 10 release.

- **Oracle Solaris 10 1/13**: Oracle Solaris ZFS tunable information is provided in Chapter 3, “Oracle Solaris ZFS Tunable Parameters.”
- **Oracle Solaris 10 8/11**: The rstchown parameter that was previously set in the /etc/system file is obsolete. If you set this parameter in the /etc/system file, the following error message is displayed:
  
  ```
  sorry, variable 'rstchown' is not defined in the 'kernel'
  ```

  This parameter has been replaced by the ZFS rstchown file system property and a general file system mount option. For more information, see Oracle Solaris ZFS Administration Guide and mount(1M).

- **Oracle Solaris 10 8/11**: This release includes the ngroups_max parameter description. For more information, see “ngroups_max” on page 44.
What's New in Oracle Solaris System Tuning?

- **Solaris 10 10/09**: This release includes the `zfs_arc_min` and `zfs_arc_max` parameter descriptions. For more information, see "zfs_arc_min" on page 94 and "zfs_arc_max" on page 95.

- **Solaris 10 10/09**: Memory locality group parameters are provided in this release. For more information about these parameters, see "Locality Group Parameters" on page 89.

- **Solaris 10 5/08**: The translation storage buffers parameters in the "SPARC System Specific Parameters" on page 85 section have been revised to provide better information. In this release, the following parameters have changed:
  - "default_tsb_size" on page 87
  - "enable_tsb_rss_sizing" on page 88
  - "tsb_rss_factor" on page 88

- **Solaris 10 8/07**: Parameter information was updated to include sun4v systems. For more information, see the following references:
  - "maxphys" on page 62
  - "tmpfs:tmpfs_maxkmem" on page 75
  - "SPARC System Specific Parameters" on page 85

- **Solaris 10 8/07**: The IP instances project enables you to configure a zone as an exclusive-IP zone and assign exclusive access of some LANs or VLANs to that zone.

  The previous behavior of shared-IP zones remains the default behavior. The exclusive-IP zone means that all aspects of the TCP/IP state and policy are per exclusive-IP zone, including TCP/IP tunable parameters.

  The introduction of the IP instances feature means that the following TCP parameters can only be set in the global zone because they require the PRIV_SYS_NET_CONFIG privilege:
  - "ip_squeue_fanout" on page 141
  - "ip_squeue_worker_wait" on page 157

  The other TCP, IP, and SCTP parameters and route metrics only require the PRIV_SYS_IP_CONFIG privilege. Each exclusive-IP zone controls its own set of these parameters. For shared-IP zones, TCP, IP, SCTP, and route parameters are controlled by the global zone since the settings of these parameters are shared between the global zone and all shared IP zones.

Oracle Solaris System Tuning in the Solaris 10 Release

This section describes significant tuning enhancements in the Oracle Solaris 10 release.

- "Default Stack Size" on page 19
- "System V IPC Configuration" on page 19
- "NFSv4 Parameters" on page 21
- "New and Changed TCP/IP Parameters" on page 21
- "SPARC: Translation Storage Buffer (TSB) Parameters" on page 23
- "SCTP Tunable Parameters" on page 23

Default Stack Size

A new parameter, default_stksize, specifies the default stack size of all threads, kernel or user. The lwp_default_stksize parameter is still available, but it does not affect all kernel stacks. If default_stksize is set, it overrides lwp_default_stksize. For more information, see "default_stksize" on page 33.

System V IPC Configuration

In the Oracle Solaris 10 release, all System V IPC facilities are either automatically configured or can be controlled by resource controls. Facilities that can be shared are memory, message queues, and semaphores.

Resource controls allow IPC settings to be made on a per-project or per-process basis on the local system or in a name service environment.

In previous Solaris releases, IPC facilities were controlled by kernel tunables. You had to modify the /etc/system file and reboot the system to change the default values for these facilities.

Because the IPC facilities are now controlled by resource controls, their configuration can be modified while the system is running.

Many applications that previously required system tuning to function might now run without tuning because of increased defaults and the automatic allocation of resources.

The following table identifies the now obsolete IPC tunables and the possible resource controls that could be used as replacements. An important distinction between the obsolete IPC tunables and resource controls is that the IPC tunables were set on a system-wide basis and the resource controls are set on a per-project or per-process basis.
<table>
<thead>
<tr>
<th>Resource Control</th>
<th>Obsolete Tunable</th>
<th>Old Default Value</th>
<th>Maximum Value</th>
<th>New Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-msg-qbytes</td>
<td>msgsys:msginfo_msgmnb</td>
<td>4096</td>
<td>ULONG_MAX</td>
<td>65536</td>
</tr>
<tr>
<td>process.max-msg-messages</td>
<td>msgsys:msginfo_msgtql</td>
<td>40</td>
<td>UINT_MAX</td>
<td>8192</td>
</tr>
<tr>
<td>process.max-sem-ops</td>
<td>semsys:seminfo_semopm</td>
<td>10</td>
<td>INT_MAX</td>
<td>512</td>
</tr>
<tr>
<td>process.max-sem-nsems</td>
<td>semsys:seminfo_semsl</td>
<td>25</td>
<td>SHRT_MAX</td>
<td>512</td>
</tr>
<tr>
<td>project.max-shm-memory</td>
<td>shmsys:shminfo_shmmax*</td>
<td>0x800000</td>
<td>UINT64_MAX</td>
<td>1/4 of physical memory</td>
</tr>
<tr>
<td>project.max-shm-ids</td>
<td>shmsys:shminfo_shmmni</td>
<td>100</td>
<td>$2^{24}$</td>
<td>128</td>
</tr>
<tr>
<td>project.max-msg-ids</td>
<td>msgsys:msginfo_msgmni</td>
<td>50</td>
<td>$2^{24}$</td>
<td>128</td>
</tr>
<tr>
<td>project.max-sem-ids</td>
<td>semsys:seminfo_semmni</td>
<td>10</td>
<td>$2^{24}$</td>
<td>128</td>
</tr>
</tbody>
</table>

* Note that the `project.max-shm-memory` resource control limits the total amount of shared memory of one project, whereas previously, the `shmsys:shminfo_shmmax` parameter limited the size of a single shared memory segment.


Obsolete parameters can still be included in the `/etc/system` file on an Oracle Solaris system. If so, the parameters are used to initialize the default resource control values as in previous Oracle Solaris releases. For more information, see “Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)” on page 186. However, using the obsolete parameters is not recommended.

The following related parameters have been removed. If these parameters are included in the `/etc/system` file on an Oracle Solaris system, the parameters are commented out.

- `semsys:seminfo_semmns`
- `semsys:seminfo_semmnu`
- `semsys:seminfo_semume`
- `semsys:seminfo_semmap`
- `shmsys:shminfo_shmmin`
- `msgsys:msginfo_msgseg`
- `msgsys:msginfo_msgmax`
For the current list of available resource controls, see `rctladm(1M)`. For information about configuring resource controls, see `project(4)`, and Chapter 6, “Resource Controls (Overview),” in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*.

**NFSv4 Parameters**

The following parameters for the NFSv4 protocol are included in the Oracle Solaris 10 release:

- “nfs:nfs4_pathconf_disable_cache” on page 106
- “nfs:nfs4_cots_timeo” on page 109
- “nfs:nfs4_do_symlink_cache” on page 111
- “nfs:nfs4_lookup_neg_cache” on page 114
- “nfs:nfs4_max_threads” on page 116
- “nfs:nfs4_nra” on page 118
- “nfs:nfs4_bsize” on page 124
- “nfs4_async_clusters” on page 126
- “nfs:nfs4_max_transfer_size” on page 130

For information about NFSv4 parameters, see “NFS Module Parameters” on page 106.

**New and Changed TCP/IP Parameters**

The following IP parameters are available in the Oracle Solaris 10 release:

- “ip_squeue_worker_wait” on page 157
- “ip_squeue_fanout” on page 141
- “ipcl_conn_hash_size” on page 156

The following TCP parameters are available in the Oracle Solaris 10 release:

- “tcp_rst_sent_rate_enabled” on page 153
- “tcp_rst_sent_rate” on page 154
- “tcp_mdt_max_pbufs” on page 154

The following TCP/IP parameters are obsolete in this Oracle Solaris release:

- `ipc_tcp_conn_hash_size`
- `tcp_compression_enabled`
- `tcp_conn_hash_size`
- `ip_forwarding`
- `ip6_forwarding`
- `xxx_forwarding`
IP Forwarding Changes

In this Oracle Solaris release, IP forwarding is enabled or disabled by using the routeadm command or the ifconfig commands instead of setting the following tunable parameters with the ndd command:

- `ip_forwarding`
- `ip6_forwarding`
- `xxx_forwarding`

Using the routeadm command and the ifconfig command instead of the ndd command to set IP forwarding provides the following advantages:

- All settings are persistent across reboots
- The new `ifconfig router` and `-router` commands can be placed in the `/etc/hostname.interface` files, along with other ifconfig commands that are run when the interface is initially configured.

To enable IPv4 or IPv6 packet forwarding on all interfaces of a system, you would use the following commands:

```
# routeadm -e ipv4-forwarding
# routeadm -e ipv6-forwarding
```

To disable IPv4 or IPv6 packet forwarding on all interfaces of a system, you would use the following commands:

```
# routeadm -d ipv4-forwarding
# routeadm -d ipv6-forwarding
```

In previous Solaris releases, you would enable IPv4 or IPv6 packet forwarding on all interfaces of a system as follows:

```
# ndd -set /dev/ip ip_forwarding 1
# ndd -set /dev/ip ip6_forwarding 1
```

In previous Solaris releases, you would disable IPv4 or IPv6 packet forwarding on all interfaces of a system as follows:

```
# ndd -set /dev/ip ip_forwarding 0
# ndd -set /dev/ip ip6_forwarding 0
```

If you want to enable IP forwarding on a specific IPv4 interface or IPv6 interface, you would use syntax similar to the following for your interface. The `bge0` interface is used as an example.

```
# ifconfig bge0 router
```
If you want to disable IP forwarding on a specific IPv4 interface or IPv6 interface, you would use syntax similar to the following for your interface. The bge0 interface is used as an example.

```
# ifconfig bge0 -router
# ifconfig bge0 inet6 -router
```

Previously, IP forwarding was enabled on a specific interface as follows:

```
# ndd -set /dev/ip bge0:ip_forwarding 1
# ndd -set /dev/ip bge0:ip_forwarding 1
```

Previously, IP forwarding on a specific interface was disabled as follows:

```
# ndd -set /dev/ip ip_forwarding 0
# ndd -set /dev/ip ip6_forwarding 0
```

If you want any of the preceding `routeadm` settings to take effect on the running system, use the following command:

```
# routeadm -u
```

For more information, see `routeadm(1M)` and `ifconfig(1M)`.

### SPARC: Translation Storage Buffer (TSB) Parameters

New parameters for tuning Translation Storage Buffer (TSB) are included in the Oracle Solaris 10 release. For information about TSB parameters, see "SPARC System Specific Parameters" on page 85.

### SCTP Tunable Parameters

Stream Control Transmission Protocol (SCTP), a reliable transport protocol that provides services similar to the services provided by TCP, is provided in this Oracle Solaris release. For more information about SCTP tunable parameters, see "SCTP Tunable Parameters" on page 164.
**Tuning an Oracle Solaris System**

The Oracle Solaris OS is a multi-threaded, scalable UNIX operating system that runs on SPARC and x86 processors. It is self-adjusting to system load and demands minimal tuning. In some cases, however, tuning is necessary. This book provides details about the officially supported kernel tuning options available for the Oracle Solaris OS.

The Solaris kernel is composed of a core portion, which is always loaded, and a number of loadable modules that are loaded as references are made to them. Many variables referred to in the kernel portion of this guide are in the core portion. However, a few variables are located in loadable modules.

A key consideration in system tuning is that setting system parameters (or system variables) is often the least effective action that can be done to improve performance. Changing the behavior of the application is generally the most effective tuning aid available. Adding more physical memory and balancing disk I/O patterns are also useful. In a few rare cases, changing one of the variables described in this guide will have a substantial effect on system performance.

Remember that one system’s `/etc/system` settings might not be applicable, either wholly or in part, to another system’s environment. Carefully consider the values in the file with respect to the environment in which they will be applied. Make sure that you understand the behavior of a system before attempting to apply changes to the system variables that are described here.

We recommend that you start with an empty `/etc/system` file when moving to a new Oracle Solaris release. As a first step, add only those tunables that are required by in-house or third-party applications. Any tunables that involve System V IPC (semaphores, shared memory, and message queues) have been modified in the Oracle Solaris 10 release and should be changed in your environment. For more information, see “System V IPC Configuration” on page 19. After baseline testing has been established, evaluate system performance to determine if additional tunable settings are required.

---

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

---

**Tuning Format of Tunable Parameters Descriptions**

The format for the description of each tunable parameter is as follows:

- Parameter Name
- Description
- Data Type
- Default
- Range
Parameter Name | Is the exact name that is typed in the /etc/system file, or found in the /etc/default/facility file.

Description | Briefly describes what the parameter does or controls.

Data Type | Indicates the signed or unsigned short integer or long integer with the following distinctions:
- On a system that runs a 32-bit kernel, a long integer is the same size as an integer.
- On a system that runs a 64-bit kernel, 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 | What the system uses as the default value.

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? Yes, if the parameter can be changed on a running system with the \texttt{mdb} or \texttt{kmdb} debugger. No, if the parameter is a boot time initialization only.

Validation Checks that the system applies to the value of the variable either as specified in the \texttt{/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 \texttt{attributes(5)}.

Change History (Optional) Contains a link to the Change History appendix, if applicable.

---

**Tuning the Oracle Solaris Kernel**

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

<table>
<thead>
<tr>
<th>Apply Tunable Parameters In These Ways</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify the \texttt{/etc/system file}</td>
<td>\texttt{/etc/system File} on page 26</td>
</tr>
<tr>
<td>Use the kernel debugger (\texttt{kmdb})</td>
<td>\texttt{kmdb Command} on page 27</td>
</tr>
<tr>
<td>Use the modular debugger (\texttt{mdb})</td>
<td>\texttt{mdb Command} on page 28</td>
</tr>
<tr>
<td>Use the \texttt{ndd} command to set TCP/IP parameters</td>
<td>Chapter 5, &quot;Internet Protocol Suite Tunable Parameters&quot;</td>
</tr>
<tr>
<td>Modify the \texttt{/etc/default files}</td>
<td>Chapter 6, &quot;System Facility Parameters&quot;</td>
</tr>
</tbody>
</table>

**/etc/system File**

The \texttt{/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 are applied. Any changes that are made to the file are not applied to the operating system until the system is rebooted.
One pass is made to set all the values before the configuration parameters are calculated.

**Example-Setting a Parameter in /etc/system**
The following /etc/system entry sets the ZFS ARC maximum (zfs_arc_max) to 30 GB.

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

**Recovering From an Incorrect Value**
Make a copy of the /etc/system file before modifying it so that you can easily recover from incorrect value. For example:

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

If a value specified in the /etc/system file 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 is displayed, 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 on system recovery, see *Oracle Solaris Administration: Basic Administration*.

**kmdb Command**

kmdb is an interactive kernel debugger with the same general syntax as mdb. An advantage of interactive kernel debugger is that you can set breakpoints. When a breakpoint is reached, you can examine data or step through the execution of kernel code.

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 kadb.

For more information, see kmdb(1).
**mdb Command**

The modular debugger, mdb, is unique among Solaris debuggers because it is easily extensible. A programming API is available that allows compilation of modules to perform desired tasks within the context of the debugger.

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

For more information, see mdb(1).

**Example–Using mdb to Display Information**

Display a high-level view of a system’s memory usage. For example:

```
# mdb -k Loading modules: [ unix genunix specfs dtrace zfs sd pcisch sockfs ip hook neti scgp ar
usba fcp fctl md lofs cpc random crypto fcip nca logindmux ptm ufs sppp nfs ] > ::memstat
Page Summary | Pages | MB | %Tot
-------------|-------|----|------
Kernel       | 95193 | 743| 37%
ZFS File Data| 96308 | 752| 38%
Anon         | 28132 | 219| 11%
Exec and libs| 1870  | 14 | 1%
Page cache   | 1465  | 11 | 1%
Free (cachelist)| 4242 | 33 | 2%
Free (freelist)| 28719| 224| 11%
Total        | 255929| 1999|
Physical     | 254495| 1988|
> $q
```

For more information on using the modular debugger, see the Oracle Solaris Modular Debugger Guide.

When using either 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.

For example, ufs:ufs_WRITES would be accessed as ufs_WRITES in each debugger (assuming the UFS module is loaded). The ufs: prefix is required when set in the /etc/system file.
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.

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 the system to 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 mdb(1) or kadb(1M).

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 system with 16 GB of memory is as follows:

20840448 maximum memory allowed in buffer cache (bufhwm)
15898 maximum number of processes (v.v.proc)
99 maximum global priority in sys class (MAXCLSYSPRI)
15893 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 sysdef(1M).
kstat Utility

kstats 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 \texttt{kstat(1M)} or \texttt{kstat(3KSTAT)}. 

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

- “General Kernel and Memory Parameters” on page 32
- “fsflush and Related Parameters” on page 37
- “Process-Sizing Parameters” on page 41
- “Paging-Related Parameters” on page 45
- “Swapping-Related Parameters” on page 56
- “Kernel Memory Allocator” on page 58
- “General Driver Parameters” on page 60
- “General I/O Parameters” on page 62
- “General File System Parameters” on page 64
- “UFS Parameters” on page 67
- “TMPFS Parameters” on page 75
- “Pseudo Terminals” on page 76
- “STREAMS Parameters” on page 79
- “System V Message Queues” on page 80
- “System V Semaphores” on page 81
- “System V Shared Memory” on page 81
- “Scheduling” on page 83
- “Timers” on page 84
- “SPARC System Specific Parameters” on page 85
- “Locality Group Parameters” on page 89
- “Solaris Volume Manager Parameters” on page 91
General Kernel and Memory Parameters

This section describes general kernel parameters that are related to physical memory and stack configuration. The ZFS-related memory parameters have moved to Chapter 3, “Oracle Solaris ZFS Tunable Parameters.”

**physmem**

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
**default_stksize**

**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` on page 34."

**Data Type** Integer

**Default**

- 3 x PAGESIZE on SPARC systems
- 2 x PAGESIZE on x86 systems
- 5 x PAGESIZE on x64 systems

**Range** Minimum is the default values:

- 3 x PAGESIZE on SPARC systems
- 2 x PAGESIZE on x86 systems
- 5 x PAGESIZE on x86 systems

Maximum is 32 times the default value.

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

**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.
<table>
<thead>
<tr>
<th>Commitment Level</th>
<th>Unstable</th>
</tr>
</thead>
</table>

**lwp_default_stksize**

**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.

**Data Type**
Integer

**Default**
- 8192 for x86 platforms
- 24,576 for SPARC platforms
- 20,480 for x64 platforms

**Range**
Minimum is the default values:
- 3 x PAGESIZE on SPARC systems
- 2 x PAGESIZE on x86 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 `getpagesize(3C)`.

**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
Change History For information, see “lwp_default_stksize (Solaris 10 Releases)” on page 182.

**logevent_max_q_sz**

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 ddi_log_sysevent(9F) and sysevent_post_event(3SYSEVENT).

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

Commitment Level Unstable

**segkpsize**

Description Specifies the amount of kernel pageable memory available. This memory is used primarily for kernel thread stacks. Increasing this number allows either larger stacks for the same number of threads or more threads. This parameter can only be set on a system running a 64-bit kernel. A system running a 64-bit kernel uses a default stack size of 24 KB.

Data Type Unsigned long
Default 64-bit kernels, 2 GB
32-bit kernels, 512 MB

Range 64-bit kernels, 512 MB to 24 GB

Units Pages

Dynamic? No

Validation Value is compared to minimum and maximum sizes (512 MB and 24 GB for 64-bit systems). If smaller than the minimum or larger than the maximum, it is reset to 2 GB. A message to that effect is displayed.

The actual size used in creation of the cache is the lesser of the value specified in segkpsize after the validation checking or 50 percent of physical memory.

When to Change Required to support large numbers of processes on a system. The default size of 2 GB. This default size allows creation of 24-KB stacks for more than 65,536 kernel threads. If more than this number is needed, segkpsize can be increased, assuming sufficient physical memory exists.

Commitment Level Unstable

**noexec_user_stack**

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 32-bit applications nonexecutable on systems running 64-bit or 32-bit kernels.

**Note** – This parameter is only effective on 64-bit SPARC and AMD64 architectures.

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 `mprotect(2)`.
Commitment Level: Unstable
Change History: For information, see “noexec_user_stack (Solaris 10 Releases)” on page 181.

**fsflush and Related Parameters**

This section describes `fsflush` and related tunables.

**fsflush**

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` on page 38 and “`autoup` on page 38 for details.

The following features are configurable:
- Frequency of invocation (tune_t_fsflushr)
Whether memory scanning is executed (do page flush)
Whether file system data flushing occurs (do if flush)
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**

- **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

**autoup**

- **Description**: Along with `tune_t_fsflushr`, 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 machines) – If large numbers of files are created and then deleted, fsflush might unnecessarily write data pages for those files to disk.

Commitment Level:  Unstable

**dopageflush**

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)
doiflush

Description Controls whether file system metadata syncs will be executed during fsflush invocations. This synchronization is done every Nth invocation of fsflush where \( N = (\text{autoup} / \text{tune_t_fsflushr}) \).

Because this algorithm is integer division, if \( \text{tune_t_fsflushr} \) is greater than \( \text{autoup} \), 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 \( \text{tune_t_fsflushr} \) or \( \text{autoup} \) 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.
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`.

**maxusers**

**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. A number of 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

**Range**
1 to 2048, based on physical memory if not set in the `/etc/system` file
1 to 4096, if set in the `/etc/system` 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 \texttt{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 "\texttt{ncsize}" on page 64.

**reserved_procs**

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, \texttt{fs\_flush} 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 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.

**pidmax**

Description: Specifies the value of the largest possible process ID.

\texttt{pidmax} sets the value for the \texttt{maxpid} variable. Once \texttt{maxpid} is set, \texttt{pidmax} is ignored. \texttt{maxpid} is used elsewhere in the kernel to determine the maximum process ID and for validation checking.

Any attempts to set \texttt{maxpid} by adding an entry to the /etc/system file 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” on page 43.
Commitment Level: Unstable

**max_nprocs**

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 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 ncsized is not specified)
- Allocating disk quota structures for UFS (if ndquot 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 the /etc/system 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

Change History: For information, see "max_nprocs (Solaris 10 Releases)" on page 182.

### maxuprc

**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

### ngroups_max

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

**Data Type:** Signed integer

**Default:** 16

**Range:** 0 to 1024

**Units:** Groups

**Dynamic?** No
Paging-Related Parameters

The 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” on page 52 and “slowscan” on page 53.) As available memory falls between the range of 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

### 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
When you want to increase the maximum number of groups.

Keep in mind that if a particular user is assigned to more than 16 groups, the user might experience problems with AUTH_SYS credentials in an NFS environment.

### Commitment Level
Unstable

### Change History
For information, see “ngroups_max (Solaris 10 Releases)” on page 182.
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 the /etc/system file can cause the system to use less than optimal values.

---

### Caution – Remove all tuning of the VM system from the /etc/system file. 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. In that case, the value specified in /etc/system is used, unless a constraint on the value of the variable has been violated. In this case, the value is reset.

<table>
<thead>
<tr>
<th>lotsfree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Data Type</strong></td>
</tr>
<tr>
<td><strong>Default</strong></td>
</tr>
<tr>
<td><strong>Range</strong></td>
</tr>
</tbody>
</table>
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

**desfree**

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
### minfree

**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.

<table>
<thead>
<tr>
<th>Units</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic?</td>
<td>Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in the <code>/etc/system</code> file or calculated from the new physical memory value.</td>
</tr>
<tr>
<td>Validation</td>
<td>If <code>minfree</code> is greater than <code>desfree</code>, <code>minfree</code> is set to <code>desfree / 2</code>. No message is displayed.</td>
</tr>
<tr>
<td>Implicit</td>
<td>The relationship of <code>lotsfree</code> being greater than <code>desfree</code>, which is greater than <code>minfree</code>, should be maintained at all times.</td>
</tr>
<tr>
<td>When to Change</td>
<td>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 <code>getpagesize</code>.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

### throttlefree

**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.

<table>
<thead>
<tr>
<th>Units</th>
<th>Pages</th>
</tr>
</thead>
</table>

---

Chapter 2 • Oracle Solaris Kernel Tunable Parameters
### Dynamic?

Yes, unless dynamic reconfiguration operations that add or delete memory occur. At that point, the value is reset to the value provided in the `/etc/system` 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 `getpagesize(3C)`.

### Commitment Level

Unstable

---

### pageout_reserve

**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 the `/etc/system` 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_pp_maximum**

**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 the /etc/system file or was calculated from the new physical memory value.

**Validation**
If the value specified in the /etc/system file 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 the /etc/system 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 `mlock(3C)`, `mlockall(3C)`, and `memcntl(2)`.

Commitment Level Unstable

**tune_t_minarmem**

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

**fastscan**

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 the `/etc/system` file is used.
- The `maxfastscan` value set in the `/etc/system` file is used.
If neither `fastscan` nor `maxfastscan` is set in the `/etc/system` 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` 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

### slowscan

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

**DataType** 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 the `/etc/system` 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

**min_percent_cpu**

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

**handspreadpages**

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

---

### pages\_before\_pager

#### 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 `\texttt{lots\_free + 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
**maxpgio**

**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**
40

**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/Os

**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

**Change History**
For information, see "maxpgio (Solaris 10 Releases)" on page 183.

---

**Swapping-Related Parameters**

Swapping in the Oracle Solaris OS is accomplished by the swapfs pseudo file system. The combination of space on swap devices and physical memory is treated as the pool of space available to support the system for maintaining backing store for anonymous memory. The system attempts to allocate space from disk devices first, and then uses physical memory as backing store. When swapfs is forced to use system memory for backing store, limits are enforced to ensure that the system does not deadlock because of excessive consumption by swapfs.
**swapfs_reserve**

Description: Defines the amount of system memory that is reserved for use by system (UID = 0) processes.

Data Type: Unsigned long

Default: The smaller of 4 MB and 1/16th of physical memory

Range: The minimum value is 4 MB or 1/16th of physical memory, whichever is smaller, 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 10 percent of physical memory. The system does not enforce this range, other than that described in the Validation section.

Units: Pages

Dynamic?: No

Validation: None

When to Change: Generally not necessary. Only change when recommended by a software provider, or when system processes are terminating because of an inability to obtain swap space. A much better solution is to add physical memory or additional swap devices to the system.

Commitment Level: Unstable

**swapfs_minfree**

Description: Defines the desired amount of physical memory to be kept free for the rest of the system. Attempts to reserve memory for use as swap space by any process that causes the system’s perception of available memory to fall below this value are rejected. Pages reserved in this manner can only be used for locked-down allocations by the kernel or by user-level processes.

Data Type: Unsigned long

Default: The larger of 2 MB and 12.5% of physical memory

Range: 1 to amount of physical memory

Units: Pages

Dynamic?: No
Validation None
When to Change Consider reducing this parameter value when processes are failing because of an inability to obtain swap space, yet the system has memory available. For example, change this value to use no more than 6.25% of system memory, but do not reduce it below 5% of system memory.

On SPARC systems, the value should be at least 2 times the value of tsb_alloc_hiwater_factor. For more information, see "tsb_alloc_hiwater_factor" on page 86.

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

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.

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.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIT</td>
<td>0x1</td>
<td>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.</td>
</tr>
<tr>
<td>TEST</td>
<td>0x2</td>
<td>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.</td>
</tr>
<tr>
<td>REDZONE</td>
<td>0x4</td>
<td>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.</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>0x8</td>
<td>The allocator logs up to 256 bytes of buffer contents when the buffer is freed. This flag requires that AUDIT also be set.</td>
</tr>
<tr>
<td>LITE</td>
<td>0x100</td>
<td>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.</td>
</tr>
</tbody>
</table>

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
General Driver Parameters

moddebug

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

**ddi_msix_alloc_limit**

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 the /etc/system file or by setting it with mdb before the device driver attach occurs.

Data Type

Signed integer

Default

2

Range

1 to 16

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

Change History

For information, see “ddi_msix_alloc_limit” on page 182.
<table>
<thead>
<tr>
<th><strong>maxphys</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>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 <code>maxphys</code> sized chunks. File systems can and do impose their own limit.</td>
</tr>
<tr>
<td><strong>Data Type</strong></td>
<td>Signed integer</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>131,072 (sun4u or 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.</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>Machine-specific page size to <code>MAXINT</code></td>
</tr>
<tr>
<td><strong>Units</strong></td>
<td>Bytes</td>
</tr>
<tr>
<td><strong>Dynamic?</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Validation</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>When to Change</strong></td>
<td>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 <code>maxphys</code> does not result in any performance improvement in that case. You might also consider changing this parameter when doing I/O to and from a UFS file system where large amounts of data (greater than 64 KB) are being read or written at any one time. The file system should be optimized to increase contiguity. For example, increase the size of the cylinder groups and decrease the number of inodes per cylinder group. UFS imposes an internal limit of 1 MB on the maximum I/O size it transfers.</td>
</tr>
<tr>
<td><strong>Commitment Level</strong></td>
<td>Unstable</td>
</tr>
<tr>
<td><strong>Change History</strong></td>
<td>For information, see “maxphys (Solaris 10 Releases)” on page 183.</td>
</tr>
</tbody>
</table>
**rlim_fd_max**

**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**  
65,536

**Range**  
1 to MAXINT

**Units**  
File descriptors

**Dynamic?**  
No

**Validation**  
None

**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:

- A 32-bit program using standard I/O is limited to 256 file descriptors. A 64-bit program using standard I/O can use up to 2 billion descriptors. Specifically, standard I/O refers to the `stdio(3C)` functions in `libc(3LIB)`.

- `select` is by default limited to 1024 descriptors per `fd_set`. For more information, see `select(3C)`. 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(1)` 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_cur**

**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.
**General File System Parameters**

### ncs

**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**
\((4 \times (v.v\_proc + \text{maxusers}) + 320) + (4 \times (v.v\_proc + \text{maxusers}) + 320) / 100\)

**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 determine when entries have been removed from the DNLC because it was too small.
The sum of the `pick_heuristic` and the `pick_last` parameters represents otherwise valid entries that were reclaimed because the cache was too small.

Excessive values of `ncsize` have an immediate impact on the system because the system allocates a set of data structures for the DNLC based on the value of `ncsize`. A system running a 32-bit kernel allocates 36-byte structures for `ncsize`, while a system running a 64-bit kernel allocates 64-byte structures for `ncsize`. The value has a further effect on UFS and NFS, unless `ufs_ninode` and `nfs:nrnode` are explicitly set.

Commitment Level: Unstable
Change History: For information, see “ncsize (Solaris 10 Release)” on page 183.

**rstchown**

Description: Indicates whether the POSIX semantics for the `chown` system call are in effect. POSIX semantics are as follows:
- A process cannot change the owner of a file, unless it is running with UID 0.
- A process cannot change the group ownership of a file to a group in which it is not currently a member, unless it is running as UID 0.

For more information, see `chown(2)`.

Data Type: Signed integer
Default: 1, indicating that POSIX semantics are used
Range: `0 = POSIX semantics not in force or 1 = POSIX semantics used`
Units: Toggle (on/off)
Dynamic?: Yes
Validation: None
When to Change: When POSIX semantics are not wanted. Note that turning off POSIX semantics opens the potential for various security holes. Doing so also opens the possibility of a user changing ownership of a file to another user and being unable to retrieve the file without intervention from the user or the system administrator.

Commitment Level: Obsolete
**dnlc_dir_enable**

**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_min_size**

**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. For instance, UFS
limits directories to a minimum of \texttt{ufs\_min\_dir\_cache} bytes (approximately 1024 entries), assuming 16 bytes per entry.

\textbf{Commitment Level} \hspace{1cm} Unstable

\textbf{dnlc\_dir\_max\_size}

\textbf{Description} \hspace{1cm} Specifies the maximum number of entries cached for one directory.

\textbf{Note} – This parameter has no effect on NFS or ZFS file systems.

\begin{tabular}{|l|}
\hline
\textbf{Data Type} \hspace{1cm} Unsigned integer \\
\textbf{Default} \hspace{1cm} MAXUINT (no maximum) \\
\textbf{Range} \hspace{1cm} 0 to MAXUINT \\
\textbf{Dynamic?} \hspace{1cm} Yes, this parameter can be changed at any time. \\
\textbf{Validation} \hspace{1cm} None \\
\textbf{When to Change} \hspace{1cm} If performance problems occur with large directories, then decrease \texttt{dnlc\_dir\_max\_size}. \\
\textbf{Commitment Level} \hspace{1cm} Unstable \\
\hline
\end{tabular}

\textbf{UFS Parameters}

\textbf{bufhwm and bufhwm\_pct}

\textbf{Description} \hspace{1cm} Defines the maximum amount of memory for caching I/O buffers. The buffers are used for writing file system metadata (superblocks, inodes, indirect blocks, and directories). Buffers are allocated as needed until the amount of memory (in KB) to be allocated exceed \texttt{bufhwm}. At this point, metadata is purged from the buffer cache until enough buffers are reclaimed to satisfy the request.

For historical reasons, \texttt{bufhwm} does not require the \texttt{ufs:} prefix.

\textbf{Data Type} \hspace{1cm} Signed integer \\
\textbf{Default} \hspace{1cm} 2 percent of physical memory
Range: 80 KB to 20 percent of physical memory, or 2 TB, whichever is less. Consequently, bufhwm_pct can be between 1 and 20.

Units:
- bufhwm: KB
- bufhwm_pct: percent of physical memory

Dynamic?: No. bufhwm and bufhwm_pct are only evaluated at system initialization to compute hash bucket sizes. The limit in bytes calculated from these parameters is then stored in a data structure that adjusts this value as buffers are allocated and deallocated.

Attempting to adjust this value without following the locking protocol on a running system can lead to incorrect operation.

Modifying bufhwm or bufhwm_pct at runtime has no effect.

Validation: If bufhwm is less than its lower limit of 80 KB or greater than its upper limit (the lesser of 20 percent of physical memory, 2 TB, or one quarter (1/4) of the maximum amount of kernel heap), it is reset to the upper limit. The following message appears on the system console and in the /var/adm/messages file if an invalid value is attempted:

"binit: bufhwm (value attempted) out of range (range start..range end). Using N as default."

"Value attempted" refers to the value specified in the /etc/system file or by using a kernel debugger. N is the value computed by the system based on available system memory.

Likewise, if bufhwm_pct is set to a value that is outside the allowed range of 1 percent to 20 percent, it is reset to the default of 2 percent. And, the following message appears on the system console and in the /var/adm/messages file:

"binit: bufhwm_pct(value attempted) out of range(0..20). Using 2 as default."

If both bufhwm or bufhwm_pct are set to non-zero values, bufhwm takes precedence.

When to Change: Because buffers are only allocated as they are needed, the overhead from the default setting is the required allocation of control structures for the buffer hash headers. These structures consume 52 bytes per potential buffer on a 32-bit kernel and 96 bytes per potential buffer on a 64-bit kernel.
On a 512-MB 64-bit kernel, the number of hash chains calculates to $10316 / 32 = 322$, which scales up to next power of 2, 512. Therefore, the hash headers consume $512 \times 96\text{ bytes}$, or 48 KB. The hash header allocations assume that buffers are 32 KB.

The amount of memory, which has not been allocated in the buffer pool, can be found by looking at the `bfreelist` structure in the kernel with a kernel debugger. The field of interest in the structure is `b_bufsize`, which is the possible remaining memory in bytes. Looking at it with the `buf` macro by using the `mdb` command:

```
# mdb -k
Loading modules: [ unix krtld genunix ip nfs ipc ]
> bfreelist::print "struct buf" b_bufsize
b_bufsize = 0x225800
```

The default value for `bufhwm` on this system, with 6 GB of memory, is 122277. You cannot determine the number of header structures used because the actual buffer size requested is usually larger than 1 KB. However, some space might be profitably reclaimed from control structure allocation for this system.

The same structure on a 512-MB system shows that only 4 KB of 10144 KB has not been allocated. When the `biostats kstat` is examined with `kstat -n biostats`, it is determined that the system had a reasonable ratio of `buffer_cache_hits` to `buffer_cache_lookups` as well. As such, the default setting is reasonable for that system.

Commitment Level

Unstable

**ndquot**

Description

Defines the number of quota structures for the UFS file system that should be allocated. Relevant only if quotas are enabled on one or more UFS file systems. Because of historical reasons, the `ufs:` prefix is not needed.

Data Type  
Signed integer

Default  
$((\text{maxusers} \times 40) / 4) + \text{max_nprocs}$

Range  
0 to MAXINT

Units  
Quota structures

Dynamic?  
No

Validation  
None. Excessively large values hang the system.
When to Change

When the default number of quota structures is not enough. This situation is indicated by the following message displayed on the console or written in the message log:

dquot table full

Commitment Level

Unstable

ufs_ninode

Description

Specifies the number of inodes to be held in memory. Inodes are cached globally for UFS, not on a per-file system basis.

A key parameter in this situation is $ufs_ninode$. This parameter is used to compute two key limits that affect the handling of inode caching. A high watermark of $ufs_ninode / 2$ and a low watermark of $ufs_ninode / 4$ are computed.

When the system is done with an inode, one of two things can happen:

- The file referred to by the inode is no longer on the system so the inode is deleted. After it is deleted, the space goes back into the inode cache for use by another inode (which is read from disk or created for a new file).
- The file still exists but is no longer referenced by a running process. The inode is then placed on the idle queue. Any referenced pages are still in memory.

When inodes are idled, the kernel defers the idling process to a later time. If a file system is a logging file system, the kernel also defers deletion of inodes. Two kernel threads handle this deferred processing. Each thread is responsible for one of the queues.

When the deferred processing is done, the system drops the inode onto either a delete queue or an idle queue, each of which has a thread that can run to process it. When the inode is placed on the queue, the queue occupancy is checked against the low watermark. If the queue occupancy exceeds the low watermark, the thread associated with the queue is awakened. After the queue is awakened, the thread runs through the queue and forces any pages associated with the inode out to disk and frees the inode. The thread stops when it has removed 50 percent of the inodes on the queue at the time it was awakened.

A second mechanism is in place if the idle thread is unable to keep up with the load. When the system needs to find a vnode, it goes through
the `ufs_vget` routine. The first thing `vget` does is check the length of
the idle queue. If the length is above the high watermark, then it takes
two inodes off the idle queue and “idles” them (flushes pages and frees
inodes). `vget` does this before it gets an inode for its own use.

The system does attempt to optimize by placing inodes with no in-core
pages at the head of the idle list and inodes with pages at the end of the
idle list. However, the system does no other ordering of the list. Inodes
are always removed from the front of the idle queue.

The only time that inodes are removed from the queues as a whole is
when a synchronization, unmount, or remount occur.

For historical reasons, this parameter does not require the `ufs:` prefix.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Signed integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td><code>ncsize</code></td>
</tr>
<tr>
<td>Range</td>
<td>0 to <code>MAXINT</code></td>
</tr>
<tr>
<td>Units</td>
<td>Inodes</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>If <code>ufs_ninode</code> is less than or equal to zero, the value is set to <code>ncsize</code>.</td>
</tr>
<tr>
<td>When to Change</td>
<td>When the default number of inodes is not enough. If the <code>maxsize</code> reached field as reported by <code>kstat -n inode_cache</code> is larger than the <code>maxsize</code> field in the <code>kstat</code>, the value of <code>ufs_ninode</code> might be too small. Excessive inode idling can also be a problem. You can identify excessive inode idling by using <code>kstat -n inode_cache</code> to look at the <code>inode_cache</code> kstat. Thread idles are inodes idled by the background threads while <code>vget idles</code> are idles by the requesting process before using an inode.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**ufs_WRITES**

**Description**

If `ufs_WRITES` is non-zero, the number of bytes outstanding for writes
on a file is checked. See `ufs_HW` to determine whether the write should
be issued or deferred until only `ufs_LW` bytes are outstanding. The total
number of bytes outstanding is tracked on a per-file basis so that if the
limit is passed for one file, it won’t affect writes to other files.

| 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 you want UFS write throttling turned off entirely. If sufficient I/O capacity does not exist, disabling this parameter can result in long service queues for disks.
Commitment Level Unstable

ufs_LW and ufs_HW

ufs_HW specifies the number of bytes outstanding on a single file barrier value. If the number of bytes outstanding is greater than this value and ufs_WRITES is set, then the write is deferred. The write is deferred by putting the thread issuing the write to sleep on a condition variable.

ufs_LW is the barrier for the number of bytes outstanding on a single file below which the condition variable on which other sleeping processes are toggled. When a write completes and the number of bytes is less than ufs_LW, then the condition variable is toggled, which causes all threads waiting on the variable to awaken and try to issue their writes.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Signed integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>8 x 1024 x 1024 for ufs_LW and 16 x 1024 x 1024 for ufs_HW</td>
</tr>
<tr>
<td>Range</td>
<td>0 to MAXINT</td>
</tr>
<tr>
<td>Units</td>
<td>Bytes</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
</tbody>
</table>

ufs_LW and ufs_HW have meaning only if ufs_WRITES is not equal to zero. ufs_HW and ufs_LW should be changed together to avoid needless churning when processes awaken and find that either they cannot issue a write (when ufs_LW and ufs_HW are too close) or they might have waited longer than necessary (when ufs_LW and ufs_HW are too far apart).
When to Change: Consider changing these values when file systems consist of striped volumes. The aggregate bandwidth available can easily exceed the current value of ufs_HW. Unfortunately, this parameter is not a per-file system setting.

You might also consider changing this parameter when ufs_throttles is a non-trivial number. Currently, ufs_throttles can only be accessed with a kernel debugger.

Commitment Level: Unstable

### freebehind

**Description**: Enables the freebehind algorithm. When this algorithm is enabled, the system bypasses the file system cache on newly read blocks when sequential I/O is detected during times of heavy memory use.

**Data Type**: Boolean

**Default**: 1 (enabled)

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

**Dynamic?**: Yes

**Validation**: None

**When to Change**: The freebehind algorithm can occur too easily. If no significant sequential file system activity is expected, disabling freebehind makes sure that all files, no matter how large, will be candidates for retention in the file system page cache. For more fine-grained tuning, see smallfile.

Commitment Level: Unstable

### smallfile

**Description**: Determines the size threshold of files larger than this value are candidates for no cache retention under the freebehind algorithm.

Large memory systems contain enough memory to cache thousands of 10-MB files without making severe memory demands. However, this situation is highly application dependent.
The goal of the smallfile and freebehind parameters is to reuse cached information, without causing memory shortfalls by caching too much.

- **Data Type**: Signed integer
- **Default**: 0
- **Range**: 0 to 2,147,483,647
- **Dynamic?**: Yes
- **Validation**: None
- **When to Change**: Increase smallfile if an application does sequential reads on medium-sized files and can most likely benefit from buffering, and the system is not otherwise under pressure for free memory. Medium-sized files are 32 KB to 2 GB in size.
- **Commitment Level**: Unstable

### ufs_delete_hiwat

- **Description**: Specifies the size of UFS file system delete queue for each mounted UFS file system.
- **Data Type**: Unsigned integer
- **Default**: 0 (unlimited)
- **Range**: 0 to $2^{64}$
- **Dynamic?**: No
- **Validation**: None
- **When to Change**: Changing this parameter is generally not recommended because a change might impact UFS delete queue performance. However, the size of delete queue influences sync operation performance when all queued operations need to run.

  In some configurations, consider limiting the size of delete queue so that a UFS file system responds faster to a sync operation. In this case, the value should be set to hundred of thousands. For example, 262,144.
- **Commitment Level**: Unstable
- **Change History**: For information, see “ufs_delete_hiwat” on page 182.
TMPFS Parameters

**tmpfs:tmpfs_maxkmem**

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 sun4u or 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

Change History: For information, see “tmpfs:tmpfs_maxkmem (Solaris 10 Releases)” on page 183.

**tmpfs:tmpfs_minfree**

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
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 by using the `telnet`, `rlogin`, or `rsh` commands
- 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 pty 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 `pt_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 pty 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 pty allocation is allowed. Note that the limit based on `pt_pctofmem` only comes into play if both `pt_cnt` and `ptms_pptymax` 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_pptymax` in `/etc/system` to the preferred number of ptys. The setting of `ptms_pptymem` is not relevant in this case.
To dedicate a different percentage of system memory to pty support and let the operating system manage the explicit limits, do the following:

- Do not set `pt_cnt` or `ptms_ptymax` in `/etc/system`.
- Set `pt_pctofmem` in `/etc/system` 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 pty. Once memory is allocated, it remains allocated.

**pt_cnt**

**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 pty 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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Unsigned integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Range</td>
<td>0 to maxpid</td>
</tr>
<tr>
<td>Units</td>
<td>Logins/windows</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>When you want to explicitly control the number of users who can remotely log in to the system.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**pt_pctofmem**

**Description**  
Specifies the maximum percentage of physical memory that can be consumed by data structures to support /dev/pts entries. A system running a 64-bit kernel consumes 176 bytes per /dev/pts entry. A system running a 32-bit kernel consumes 112 bytes per /dev/pts entry.
Pseudo Terminals

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

**pt_max_pty**

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
STREAMS Parameters

**nstrpush**

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**

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 write(2).

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 putmsg(2).

Commitment Level: Unstable
### strctlSz

<table>
<thead>
<tr>
<th>Description</th>
<th>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</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Signed integer</td>
</tr>
<tr>
<td>Default</td>
<td>1024</td>
</tr>
<tr>
<td>Range</td>
<td>0 to MAXINT</td>
</tr>
<tr>
<td>Units</td>
<td>Bytes</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>At the direction of your software vendor. <em>putmsg</em>(2) calls return ERANGE if they attempt to exceed this limit.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

### 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. Dequeueing removes the first message of a specific type from the queue or the first message if no type is specified.

For information about System V message queues in the Oracle Solaris 10 release, see “System V IPC Configuration” on page 19.

For detailed information on tuning these system resources, see Chapter 6, "Resource Controls (Overview)," in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*.

For legacy information about the obsolete System V message queues, see “Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)” on page 186.
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.

For information about the changes to semaphore resources in the Oracle Solaris 10 release, see "System V IPC Configuration" on page 19.

For detailed information about using the new resource controls in the Oracle Solaris 10 release, see Chapter 6, "Resource Controls (Overview)," in System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones.

For legacy information about the obsolete System V semaphore parameters, see “Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)” on page 186.

System V Shared Memory

System V shared memory allows the creation of a segment by a process. Cooperating processes can attach to the memory segment (subject to access permissions on the segment) and gain access to the data contained in the segment. This capability is implemented as a loadable module. Entries in the /etc/system file must contain the shmsys: prefix.

A special kind of shared memory known as intimate shared memory (ISM) is used by DBMS vendors to maximize performance. When a shared memory segment is made into an ISM segment, the memory for the segment is locked. This feature enables a faster I/O path to be followed and improves memory usage. A number of kernel resources describing the segment are then shared between all processes that attach to the segment in ISM mode.

For information about the changes to shared memory resources in the Oracle Solaris 10 release, see “System V IPC Configuration” on page 19.

For detailed information about using the new resource controls in the Oracle Solaris 10 release, see Chapter 6, "Resource Controls (Overview)," in System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones.

For legacy information about the obsolete System V shared memory parameters, see “Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)” on page 186.
segspt_minfree

**Description**
Identifies pages of system memory that cannot be allocated for ISM shared memory.

**Data Type**
Unsigned long

**Default**
5 percent of available system memory when the first ISM segment is created

**Range**
0 to 50 percent of physical memory

**Units**
Pages

**Dynamic?**
Yes

**Validation**
None. Values that are too small can cause the system to hang or performance to severely degrade when memory is consumed with ISM segments.

**When to Change**
On database servers with large amounts of physical memory using ISM, the value of this parameter can be decreased. If ISM segments are not used, this parameter has no effect. A maximum value of 128 MB (0x4000) is almost certainly sufficient on large memory machines.

**Commitment Level**
Unstable

pr_segp_disable

**Description**
Disables the page lock cache flushing when trying to retire a page that might belong to ISM.

When locked or busy (heavy I/O) pages are in the pending page retirement queue, the page retire thread flushes the segp_cache to encourage retirement of pending pages that might be owned by ISM. Periodic or repeated flushes of the segp_cache can be a bottleneck for high memory machines.

Default behavior is to flush the page cache every 30 seconds and if locked pages are observed in queue, then timeout exponentially backs off until 1 hour in multiples of 2.

Enabling pr_segp_disable does not disable the system’s ability to retire memory pages, such as those that are faulted as a result of system diagnostic measures.

**Data Type**
Boolean
When locked or busy (heavy I/O) pages are in the pending page retirement queue, the page retire thread flushes the segp_cache to encourage retirement of pending pages that might be owned by ISM. Periodic or repeated flushes of the segp_cache can be a bottleneck for high memory machines.

If you have a latency sensitive database or a large shared memory application, consider disabling this parameter to completely skip segp_cache flushing.

Symptoms of locked kernel pages that can’t be retired are as follows:

- Brief database latency or momentary database unresponsive events along with brief periodic elevated SYS CPU events upon successful page retirements. However, locked or busy pages that repeatedly fail to retire might continue to trigger page retirement threads at slower rates.

  For example, locked memory pages that can’t be retired might retry at small intervals and repeat forever at 1 hour intervals. After the system reboots, the scheduled pages might retire, or it might start trying again at 30 seconds, the default rate.

- Brief unexpected or elevated smtx lock contention might be seen when monitoring segspt_shmfault, segspt_softunlock, segspt_shmpagelock, segspt_shmfree, segspt_shmunmap, segspt_shmattach, and segspt_dismfault structures.

Commitment Level: Unstable

Scheduling

rechoose_interval

Description: Specifies the number of clock ticks before a process is deemed to have lost all affinity for the last CPU it ran on. After this interval expires, any CPU is considered a candidate for scheduling a thread. This parameter
is relevant only for threads in the timesharing class. Real-time threads are scheduled on the first available CPU.

**hires_tick**

<table>
<thead>
<tr>
<th>Description</th>
<th>When set, this parameter causes the Oracle Solaris OS to use a system clock rate of 1000 instead of the default value of 100.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Signed integer</td>
</tr>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Range</td>
<td>0 (disabled) or 1 (enabled)</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>When you want timeouts with a resolution of less than 10 milliseconds, and greater than or equal to 1 millisecond.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>
**timer_max**

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

**Data Type**
Signed integer

**Default**
32

**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 System Specific Parameters**

**consistent_coloring**

**Description**
The ability to use different page placement policies on the UltraSPARC platform is available. A page placement policy attempts to allocate physical page addresses to maximize the use of the L2 cache. Whatever algorithm is chosen as the default algorithm, that algorithm can potentially provide less optimal results than another algorithm for a particular application set. This parameter changes the placement algorithm selected for all processes on the system.

Based on the size of the L2 cache, memory is divided into bins. The page placement code allocates a page from a bin when a page fault first occurs on an unmapped page. The page chosen depends on which of the three possible algorithms are used:

- **Page coloring** – Various bits of the virtual address are used to determine the bin from which the page is selected. consistent_coloring is set to zero to use this algorithm. No per-process history exists for this algorithm.

- **Virtual addr=physical address** – Consecutive pages in the program selects pages from consecutive bins. consistent_coloring is set to 1 to use this algorithm. No per-process history exists for this algorithm.
- Bin-hopping – Consecutive pages in the program generally allocate pages from every other bin, but the algorithm occasionally skips more bins. consistent_coloring is set to 2 to use this algorithm. Each process starts at a randomly selected bin, and a per-process memory of the last bin allocated is kept.

Dynamic? Yes

Validation None. Values larger than 2 cause a number of WARNING: AS_2_BIN: bad consistent coloring value messages to appear on the console. The system hangs immediately thereafter. A power-cycle is required to recover.

When to Change When the primary workload of the system is a set of long-running high-performance computing (HPC) applications. Changing this value might provide better performance. File servers, database servers, and systems with a number of active processes (for example, compile or time sharing servers) do not benefit from changes.

Commitment Level Unstable

**tsb_alloc_hiwater_factor**

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:

\[
\text{tsb_alloc_hiwater} = \frac{\text{physical memory (bytes)}}{\text{tsb_alloc_hiwater_factor}}
\]

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

**default_tsb_size**

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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8 KB</td>
</tr>
<tr>
<td>1</td>
<td>16 KB</td>
</tr>
<tr>
<td>3</td>
<td>32 KB</td>
</tr>
<tr>
<td>4</td>
<td>128 KB</td>
</tr>
<tr>
<td>5</td>
<td>256 KB</td>
</tr>
<tr>
<td>6</td>
<td>512 KB</td>
</tr>
<tr>
<td>7</td>
<td>1 Mbyte</td>
</tr>
</tbody>
</table>

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

Change History: For information, see “default_tsb_size (Solaris 10 Releases)” on page 184.

### enable_tsb_rss_sizing

**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

Change History: For information, see “enable_tsb_rss_sizing (Solaris 10 Releases)” on page 184.

### tsb_rss_factor

**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
Change History For information, see “tsb_rss_factor (Solaris 10 Releases)” on page 184.

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).

lpg_alloc_prefer

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)
Locality Group Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
</table>
| 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                                                          |

**lgrp_mem_pset_aware**

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 psrset(1M).

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

Solaris Volume Manager Parameters

md_mirror:md_resync_bufsz

Description Sets the size of the buffer used for resynchronizing RAID 1 volumes
(mirrors) as the number of 512-byte blocks in the buffer. Setting larger
values can increase resynchronization speed.

Data Type Integer
Default The default value is 128. Larger systems could use higher values to
increase mirror resynchronization speed.

Range 128 to 2048
Units Blocks (512 bytes)
Dynamic? No
Validation None
When to Change If you use Solaris Volume Manager RAID 1 volumes (mirrors), and you
want to increase the speed of mirror resynchronizations. Assuming
that you have adequate memory for overall system performance, you
can increase this value without causing other performance problems.

If you need to increase the speed of mirror resynchronizations, increase
the value of this parameter incrementally (using 128-block increments)
until performance is satisfactory. On fairly large or new systems, a
value of 2048 seems to be optimal. High values on older systems might
hang the system.

Commitment Level Unstable
### md:mirrored_root_flag

**Description**
Overrides Solaris Volume Manager requirements for replica quorum and forces Solaris Volume Manager to start if any valid state database replicas are available.

The default value is disabled, which requires that a majority of all replicas are available and synchronized before Solaris Volume Manager will start.

**Data Type**
Boolean values

**Default**
0 (disabled)

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

**Dynamic?**
No

**Validation**
None

**When to Change**
Use of this parameter is not supported.

Some people using Solaris Volume Manager accept the risk of enabling this parameter if all three of the following conditions apply:

- When root (/) or other system-critical file systems are mirrored
- Only two disks or controllers are available
- An unattended reboot of the system is required

If this parameter is enabled, the system might boot with a stale replica that inaccurately represents the system state (including which mirror sides are good or in Maintenance state). This situation could result in data corruption or system corruption.

Change this parameter only if system availability is more important than data consistency and integrity. Closely monitor the system for any failures. You can mitigate the risk by keeping the number of failed, Maintenance, or hot-swapped volumes as low as possible.

For more information about state database replicas, see Chapter 6, “State Database (Overview),” in *Solaris Volume Manager Administration Guide*.

**Commitment Level**
Unstable
Chapter 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” on page 94
- “ZFS ARC Parameters” on page 94
- “ZFS File-Level Prefetch” on page 95
- “ZFS Device I/O Queue Depth” on page 96
- “ZFS and Cache Flushing” on page 97
- “ZFS Metadata Compression” on page 99
- “Tuning ZFS for Database Products” on page 99

Where to Find Tunable Parameter Information

<table>
<thead>
<tr>
<th>Tunable Parameter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris kernel tunable parameters</td>
<td>Chapter 2, “Oracle Solaris Kernel Tunable Parameters”</td>
</tr>
<tr>
<td>NFS tunable parameters</td>
<td>Chapter 4, “NFS Tunable Parameters”</td>
</tr>
<tr>
<td>Internet Protocol Suite tunable parameters</td>
<td>Chapter 5, “Internet Protocol Suite Tunable Parameters”</td>
</tr>
</tbody>
</table>
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 11, "Recommended Oracle Solaris ZFS Practices," in Oracle Solaris ZFS Administration Guide.

- Unless noted otherwise, the tunable parameters are global and impact ZFS behavior across the system.

ZFS ARC Parameters

This section describes parameters related to ZFS ARC behavior.

**zfs_arc_min**

Description: Determines the minimum size of the ZFS Adaptive Replacement Cache (ARC). See also “zfs_arc_max” on page 95.

Data Type: Unsigned Integer (64-bit)

Default: 1/32nd of physical memory or 64 MB, whichever value is larger.

Range: 64 MB 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

Change History: For information, see “zfs_arc_min” on page 181.
**zfs_arc_max**

**Description**
Determine the maximum size of the ZFS Adaptive Replacement Cache (ARC). See also “zfs_arc_min” on page 94.

**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**
64 MB 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

**Change History**
For information, see “zfs_arc_max” on page 181.

---

**ZFS File-Level Prefetch**

**zfs_prefetch_disable**

**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 as described here:
You can disable prefetching by setting `zfs_prefetch_disable` in the `/etc/system` file.

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, then disabling file level prefetching should be considered.

### Commitment Level
- Unstable

## ZFS Device I/O Queue Depth

### zfs_vdev_max_pending

**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 ISP2200, ISP2300, and SP212 family FCI HBA (qlc) driver is described as the execution-throttle parameter in `/kernel/drv/qlc.conf`.

Field  | Description
--- | ---
Commitment Level | Unstable

**ZFS and Cache Flushing**

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.

### zfs_nocacheflush

**Description**

This parameter controls ZFS write cache flushes for the entire system.

Oracle’s Sun hardware should not require tuning this parameter. If you need to tune cache flushing, considering tuning it per hardware device. See the general instructions below. Contact your storage vendor for instructions on how to tell the storage devices to ignore the cache flushes sent by ZFS.

**Data Type**

Boolean

**Default**

0

**Range**

0 (enabled) or 1 (disabled)

**Dynamic?**

Yes

**Validation**

No

**When to Change**

Cache flush tuning was recently shown to help some SSD performance when used as log devices. This tuning syntax can be included in sd.conf but there must be only a single sd-config-list entry per vendor/product. For example:

```
sd-config-list = "ATA TX43E10100GB0LS1","throttle-max:32, disksort:false, cache-nonvolatile:true";
```

As a last resort, 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.

**Commitment Level**

Unstable
ZFS Metadata Compression

**zfs_mdcomp_disable**

Description: This parameter controls compression of ZFS metadata (indirect blocks only). ZFS data block compression is controlled by the ZFS compression property that can be set per file system.

Data Type: Boolean

Default: 0

Range: 0 (enabled) or 1 (disabled)

Dynamic?: Yes

Validation: No

When to Change: In general, metadata compression should be enabled. For metadata intensive loads, this default is expected to gain some amount of space (a few percentages) at the expense of a little extra CPU computation. However, a bigger motivation exists to have metadata compression on. For directories that grow to millions of objects then shrink to just a few, metadata compression saves large amounts of space (greater than 10 times).

I/O on flash storage devices is aligned along 4 KB boundaries. If metadata compression is enabled, the I/O on flash storage devices might become unaligned. You might consider disabling metadata compression to resolve the I/O alignment problem if you are using flash devices for primary storage. Using separate log devices on flash devices are not affected by the alignment problem.

Commitment Level: Unstable

Tuning ZFS for Database Products

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

- If the database uses a fixed disk block or record size for I/O, set the ZFS recordsize property to match it. You can do this on a per-file system basis, even though multiple file systems might share a single pool.

- With ZFS's copy-on-write design, tuning down the recordsize is a way to improve OLTP performance at the expense of batch reporting queries.
- 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.

- UFS direct I/O is used to overcome some of the design deficiencies of UFS and to eliminate double buffering of data. In ZFS, the UFS design deficiencies do not exist and ZFS uses the primary cache and secondary cache properties to manage buffering data in the ARC. Note that using the secondary cache (L2ARC) property to improve random reads also requires the primary cache property to be enabled.

- Keep pool space under 80% utilization to maintain pool performance.

### Tuning ZFS for an Oracle Database

ZFS is recommended for any Oracle database version in single instance mode. ZFS can be used with an Oracle RAC database when it is available as a NFS-shared file system.

Review the following recommendations below for tuning ZFS for an Oracle database:

- **Verify that you are running the latest Oracle Solaris release**
  
  Start with the latest Oracle Solaris 10 or Oracle Solaris 11 release, with the Solaris 10 9/10 release as a minimum starting point.

- **Create LUNs for your ZFS storage pools, if needed**
  
  Use your storage array tools to create LUNs that will be presented to the ZFS storage pool. Or, consider using whole disks for your mirrored ZFS storage pools. For more information, see Chapter 3, “Managing Oracle Solaris ZFS Storage Pools,” in Oracle Solaris ZFS Administration Guide.

- **Create a storage pool for data files for tables, index, undo and temp data**
  
  Consider creating a mirrored storage pool to provide a higher level of data redundancy. For example:

  ```bash
  # zpool status dbpool
  pool: dbpool
  state: ONLINE
  scan: none requested
  config:
  NAME STATE READ WRITE CKSUM
  dbpool ONLINE 0 0 0
  mirror-0 ONLINE 0 0 0
  c0t5000C500335F95E3d0 ONLINE 0 0 0
  mirror-1 ONLINE 0 0 0
  c0t5000C500335BD117d0 ONLINE 0 0 0
  c0t5000C500335DC60Fd0 ONLINE 0 0 0
  ```

  Tuning ZFS for Database Products

Oracle Solaris Tunable Parameters Reference Manual • September 2013
For databases with high redo log activity, such as a typical OLTP database with many commits, use a separate LUN for a separate log device.

- **Create a storage pool for the archivelog**
  If available, a system's internal disk can handle this type of load. The archivelog file system can also be a file system in the dbpool.

  ```
  # zpool create archivepool c0t5000c500335e106bd0
  ```

- **Create the ZFS file systems and set the specific file system properties by using the following guidelines**

  Create separate file systems for redo, archive, undo, and temp database components using the default record size of 128 KB. The general rule is to set the file system `recordsize = db_block_size` for the file systems that contains Oracle data files. For table data and index components, create a file system with an 8 KB record size. Also consider providing metadata caching hints for your database file systems by using the `primarycache` property. For more information about ZFS file system properties, see "Introducing ZFS Properties" in Oracle Solaris ZFS Administration Guide.

  - Create file systems for the table data files and index data files with an 8 KB record size. Use the default value for `primarycache`.

    ```
    # zfs create -o recordsize=8k -o mountpoint=/my_db_path/index dbpool/index
    # zfs set logbias=throughput dbpool/index
    # zfs get primarycache,recordsize,logbias dbpool/index
    NAME PROPERTY VALUE SOURCE
    dbpool/index primarycache all default
    dbpool/index recordsize 8K local
    dbpool/index logbias throughput local
    ```

  - Create file systems for temporary and undo table spaces, using the default record size and `primarycache` values.

    ```
    # zfs create -o mountpoint=/my_db_path/temp dbpool/temp
    # zfs set logbias=throughput dbpool/temp
    # zfs create -o mountpoint=/my_db_path/undo dbpool/undo
    # zfs set logbias=throughput dbpool/undo
    ```

- **Create a storage pool for redo logs with a separate log device.** For databases with high redo log activity, such as a typical OLTP database with many commits, use a separate log device LUN.

  Partition the disk into two slices, a small slice, s0, in the 64 to 150 MB range, for the separate log device. The s1 slice contains the remaining disk space for the redo log.

  ```
  # zpool create redopool c0t500157959486f11d0s1 log c0t500157959486f11d0s0
  # zpool status redopool
  pool: redopool
  state: ONLINE
  scan: none requested
  config:
Create a filesystem for redo logs in the redo pool. Use default filesystem values for recordsize and primarycache.

```
# zfs create -o mountpoint=/my_db_path/redo redopool/redo
# zfs set logbias=latency redopool/redo
```

Create a filesystem for archivelog files in the archive pool, enabling compression, use the default value for recordsize and set primarycache to metadata.

```
# zfs create -o compression=on -o primarycache=metadata -o mountpoint=/my_db_admin_path/archive archivepool/archive
# zfs get primarycache,recordsize,compressratio,compression,available,used,quota archivepool/archive
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>PROPERTY</th>
<th>VALUE</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>archivepool/archive</td>
<td>primarycache</td>
<td>metadata</td>
<td>local</td>
</tr>
<tr>
<td>archivepool/archive</td>
<td>recordsize</td>
<td>128K</td>
<td>default</td>
</tr>
<tr>
<td>archivepool/archive</td>
<td>compressratio</td>
<td>1.32x</td>
<td>-</td>
</tr>
<tr>
<td>archivepool/archive</td>
<td>compression</td>
<td>on</td>
<td>local</td>
</tr>
<tr>
<td>archivepool/archive</td>
<td>available</td>
<td>40.0G</td>
<td>-</td>
</tr>
<tr>
<td>archivepool/archive</td>
<td>used</td>
<td>10.0G</td>
<td>-</td>
</tr>
<tr>
<td>archivepool/archive</td>
<td>quota</td>
<td>50G</td>
<td>local</td>
</tr>
</tbody>
</table>

**Consider tuning storage array I/O queues (for systems with HDS or EMC storage arrays)**

ZFS aggregates read and write I/O and manages the priority of I/O before sending it to the driver level, which handles the device. The `zfs_vdev_max_pending` parameter defines the maximum number of I/Os that ZFS sends to any storage pool device.

In a legacy storage environment, the `ssd_max_throttle` and `sd_max_throttle` parameters define the maximum number of concurrent I/Os that the driver can send to the storage. By setting the `zfs_vdev_max_pending` default value equal to the value of the `ssd_max_throttle` parameter, we prevent ZFS from queuing I/O to yet another unnecessary SD layer.

If you have `ssd:ssd_max_throttle` or `sd:sd_max_throttle` in the `/etc/system` file in your existing environment, then set `zfs:zfs_vdev_max_pending` at the same value. For example, if the storage array administrator asked for the following setting:

```
set ssd:ssd_max_throttle=20
```

Then, also set this parameter as follows:

```
set ssd:ssd_max_throttle=20
set zfs:zfs_vdev_max_pending=20
```

Setting this parameter allows ZFS to control each LUN queue. This means that the total number of pending I/Os in the storage can grow as follows:
number of LUNs * ZFS_VDEV_MAX_PENDING

- **Allocate sufficient memory and swap resources**
  You can reduce ZFS memory consumption by tuning the `zfs_arc_max` parameter to a low value, but we still recommend provisioning enough memory to cache metadata for the actively used portion of the database, which is estimated at 1.5% with an 8 KB ZFS record size and proportionately less or more with larger or smaller records. The file system that hold index files is the one that has the largest benefit from file system caching because it is the last one to invalidate in case of lack of memory. The `zfs_arc_max` parameter is in bytes and accepts decimal or hexadecimal values. The following example sets this parameter to 2 GB:

  set zfs:zfs_arc_max=2147483648
  or
  set zfs:zfs_arc_max=0x80000000

  To prevent applications from failing due to lack of memory, you must configure some amount of swap space. The amount of swap equivalent to all of system memory is always enough for this purpose. This swap space is not expected to be used, but is needed as a reservation area. For information about increasing swap space, see “Managing Your ZFS Swap and Dump Devices” in *Oracle Solaris ZFS Administration Guide*.

- Additional Oracle database configuration recommendations
  - *Configuring Your Oracle Database on ZFS File Systems* in the following white paper:
  - *Dynamic SGA Tuning of Oracle Database on Oracle Solaris with DISM* white paper:
  - Oracle 11g Installation Guides
    - Oracle Database Quick Installation Guide 11g Release 2 (11.2) for Oracle Solaris on SPARC (64-Bit)
      http://docs.oracle.com/cd/E11882_01/install.112/e24349/toc.htm
    - Oracle Database Quick Installation Guide 11g Release 2 (11.2) for Oracle Solaris on x86-64 (64-Bit)
      http://docs.oracle.com/cd/E11882_01/install.112/e24351/toc.htm

### Using ZFS with MySQL Considerations

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.
This section describes the NFS tunable parameters.

- “Tuning the NFS Environment” on page 105
- “NFS Module Parameters” on page 106
- “rpcmod Module Parameters” on page 132

Where to Find Tunable Parameter Information

<table>
<thead>
<tr>
<th>Tunable Parameter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris kernel tunable parameters</td>
<td>Chapter 2, “Oracle Solaris Kernel Tunable Parameters”</td>
</tr>
<tr>
<td>Oracle Solaris ZFS tunable parameters</td>
<td>Chapter 3, “Oracle Solaris ZFS Tunable Parameters”</td>
</tr>
<tr>
<td>Internet Protocol Suite tunable parameters</td>
<td>Chapter 5, “Internet Protocol Suite Tunable Parameters”</td>
</tr>
</tbody>
</table>

Tuning the NFS Environment

You can define NFS parameters in the `/etc/system` file, which is read during the boot process. Each parameter includes the name of its associated kernel module. For more information, see “Tuning an Oracle Solaris System” on page 24.

**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:nfs3_pathconf_disable_cache**

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 server can change the information for a specific file dynamically, use this parameter to disable caching. There is no mechanism for the client to validate its cache entry.

Commitment Level: Unstable

**nfs:nfs4_pathconf_disable_cache**

Description: Controls the caching of pathconf information for NFS Version 4 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 server can change the information for a specific file dynamically, use this parameter to disable caching. There is no mechanism for the client to validate its cache entry.
Commitment Level: Unstable

**nfs:nfs_allow_preepoch_time**

**Description**

Controls whether files with incorrect or negative time stamps should be made visible on the 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.

However, on a system running a 32-bit Solaris kernel, the timestamp values are signed and 32-bits long. Thus, it would be possible to have a timestamp representation that appeared to be prior to January 1, 1970, or pre-epoch.

The problem on a system running a 64-bit Solaris kernel is slightly different. The timestamp values on the 64-bit 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_cots_timeo**

**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** Signed integer (32-bit)

**Default** 600 (60 seconds)

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

**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**

**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** Signed integer (32-bit)

**Default** 600 (60 seconds)

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

**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

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 client responds to certain events, such as detecting a forced unmount operation or detecting how quickly the server fails over to a new server.

Data Type Signed integer (32-bit)

Default 600 (60 seconds)

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

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_do_symlink_cache**

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 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 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**

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 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 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**
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 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 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**
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 server response times and then adjusting RPC timeouts and read- and write- transfer sizes.
**nfs:nfs3_dynamic**

**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 server response times and then adjusting RPC timeouts and read- and write- transfer sizes.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer (32-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0 (disabled)</td>
</tr>
<tr>
<td>Range</td>
<td>0 (disabled) or 1 (enabled)</td>
</tr>
<tr>
<td>Units</td>
<td>Boolean values</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>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.</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>Do not change this parameter.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**nfs:nfs_lookup_neg_cache**

**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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer (32-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>1 (enabled)</td>
</tr>
<tr>
<td>Range</td>
<td>0 (disabled) or 1 (enabled)</td>
</tr>
<tr>
<td>Units</td>
<td>Boolean values</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>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 server is not changing or is changing very slowly, and that it is okay for such changes to propagate slowly to the 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 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 &quot;nfs:nfs_disable_rddir_cache&quot; on page 122.</td>
</tr>
</tbody>
</table>

**Commitment Level** | Unstable

### nfs:nfs3_lookup_neg_cache

**Description**

Controls whether a negative name cache is used for NFS version 3 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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer (32-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>1 (enabled)</td>
</tr>
<tr>
<td>Range</td>
<td>0 (disabled) or 1 (enabled)</td>
</tr>
<tr>
<td>Units</td>
<td>Boolean values</td>
</tr>
</tbody>
</table>
**nfs:nfs4_lookup_neg_cache**

**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 server is not changing or is changing very slowly, and that it is okay for such changes to propagate slowly to the 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 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" on page 122.

**Commitment Level**
Unstable
**nfs:nfs_max_threads**

Description: Controls the number of kernel threads that perform asynchronous I/O 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 client performs when it stops using a file.

Data Type: Unsigned short

Default: 8

Range: 0 to $2^{15} - 1$

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 client and 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**

Description: Controls the number of kernel threads that perform asynchronous I/O 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.
**nfs:nfs4_max_threads**

**Description**
Controls the number of kernel threads that perform asynchronous I/O 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 $2^{15} - 1$

**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 client and 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.
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 client and 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:nfs_nra**

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 client and 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**

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.

**DataType** Integer (32-bit)

**Default** 4

**Range** 0 to \(2^{31} - 1\)

**Units** Logical blocks. (See "nfs:nfs3_bsize" on page 123.)

**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 client and 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

**Change History** For information, see "nfs:nfs3_nra (Solaris 10 Releases)" on page 184.

---

**nfs:nfs4_nra**

**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.

**DataType** Integer (32-bit)

**Default** 4

**Range** 0 to \(2^{31} - 1\)

**Units** Logical blocks. (See "nfs:nfs4_bsize" on page 124.)

**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 client and 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**

Description  

Controls the size of the rnode cache on the NFS client.

The rnode, used by both 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 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 rnodes 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 of nrnode should be set to the value of the ncsiz parameter. Actually, any non positive value of nrnode results in nrnode being set to the value of ncsiz.

Range  

1 to \(2^{31} - 1\)

Units  

rnodes

Dynamic?  

No. This value can only be changed by adding or changing the parameter in the `/etc/system` 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 rnodes 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 \( rnodes \).

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:nfs_shrinkreaddir

**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 a 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**

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 a 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 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 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**

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 (32 bits on 32-bit platforms and 64 bits on 64-bit platforms)
Default: 5 seconds
Range: 0 to 2^{31} - 1 on 32-bit platforms
0 to 2^{63} - 1 on 64-bit platforms
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 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 server is being actively used.

Commitment Level Unstable

**nfs:nfs_write_error_to_cons_only**

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:nfs_disable_rddir_cache**

Description Controls the use of a cache to hold responses from READDIR and READDIRPLUS requests. This cache avoids over-the-wire calls to the 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 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" on page 112
- "nfs:nfs3_lookup_neg_cache" on page 113
- "nfs:nfs4_lookup_neg_cache" on page 114

Commitment Level: Unstable

**nfs:nfs3_bsize**

**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 server when it needs to do an I/O.

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

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

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

**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 the `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**

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 server when it needs to do an I/O.

Data Type: Unsigned integer (32-bit)

Default: 32,768 (32 KB)

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

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 the `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_async_clusters**

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 \(2^{31} - 1\)
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 server functionality that depends upon clusters of requests coming from the client.
Commitment Level Unstable

**nfs:nfs3_async_clusters**

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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Unsigned integer (32-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>1</td>
</tr>
<tr>
<td>Range</td>
<td>0 to $2^{31} - 1$</td>
</tr>
<tr>
<td>Units</td>
<td>Asynchronous requests</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>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.</td>
</tr>
<tr>
<td>Validation</td>
<td>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.</td>
</tr>
<tr>
<td>When to Change</td>
<td>To increase the number of each type of asynchronous operation that is generated before switching to the next type. Doing so might help with server functionality that depends upon clusters of operations coming from the client.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**nfs:nfs4_async_clusters**

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.

**DataType** Unsigned integer (32-bit)

**Default** 1

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

**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 server functionality that depends upon clusters of requests coming from the client.

**Commitment Level** Unstable

---

**nfs:nfs_async_timeout**

**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.

**DataType** 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:nacache

#### 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 `nrnode` 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 the `/etc/system` 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

---

NFS Module Parameters

<table>
<thead>
<tr>
<th>Dynamic?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>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.</td>
</tr>
<tr>
<td>When to Change</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>- By making the threads expire more quickly, thus freeing up kernel resources more quickly</td>
</tr>
<tr>
<td></td>
<td>- By making the threads expire more slowly, thus avoiding thread create and destroy overhead</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

---

**nfs:nacache**

- 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 `nrnode` 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 the `/etc/system` 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:nfs3_jukebox_delay**

**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 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 (32 bits on 32-bit platforms and 64 bits on 64-bit
platforms)

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

**Range** 0 to $2^{31}$ - 1 on 32-bit platforms

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 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:nfs3_max_transfer_size**

**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 server returns
as well as the maximum size of the request that the client generates.

**Data Type** Integer (32-bit)
**nfs:nfs3_bsize**

**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 server returns as well as the maximum size of the request that the client generates.

**DataType**

Integer (32-bit)

**Default**

1,048,576 (1 Mbyte)

**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 server to 0 is likely to cause 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 client and the server.

**When to Change**

To tune the size of data transmitted over the network. In general, the **nfs:nfs3_bsize** 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. For more information, see "**nfs:nfs3_bsize** on page 123."

If you want to use a smaller transfer size than the default transfer size, use the mount command's **-wsize** or **-rsize** option on a per-file system basis.

**Commitment Level**

Unstable

---

**nfs:nfs4_max_transfer_size**

**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 server returns as well as the maximum size of the request that the client generates.

**DataType**

Integer (32-bit)

**Default**

32,768 (32 KB)

**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 server to 0 is likely to cause clients to malfunction or just decide not to attempt to talk to the server.
When to Change To tune the size of data transmitted over the network. In general, the nfs:nfs4_bsize 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. For more information, see “nfs:nfs4_bsize” on page 124.

If you want to use a smaller transfer size than the default transfer size, use the mount command’s -wsize or -rsize option on a per-file system basis.

Commitment Level Unstable

nfs:nfs3_max_transfer_size_clts

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 server returns as well as the maximum size of the request that the client generates.
Data Type Integer (32-bit)
Default 32,768 (32 KB)
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 server to 0 is likely to cause 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

**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 server returns as well as the maximum size of the request that the client generates.

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

**Default**
1,048,576 bytes

**Range**
0 to 2³¹ - 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 server to 0 is likely to cause 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 Mbyte are preferred.

**Commitment Level**
Unstable

---

### rpcmod Module Parameters

This section describes NFS parameters for the rpcmod module.

### rpcmod:clnt_max_conns

**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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Integer (32-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>1</td>
</tr>
<tr>
<td>Range</td>
<td>1 to $2^{31} - 1$</td>
</tr>
<tr>
<td>Units</td>
<td>Connections</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>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 client and the 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.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**rpcmod:clnt_idle_timeout**

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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Long integer (32 bits on 32-bit platforms and 64 bits on 64-bit platforms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>300,000 milliseconds (5 minutes)</td>
</tr>
<tr>
<td>Range</td>
<td>0 to $2^{31} - 1$ on 32-bit platforms</td>
</tr>
<tr>
<td></td>
<td>0 to $2^{63} - 1$ on 64-bit platforms</td>
</tr>
<tr>
<td>Units</td>
<td>Milliseconds</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>Use this parameter to change the time that idle connections are allowed to exist on the client before being closed. You might want to close connections at a faster rate to avoid consuming system resources.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>
**rpcmod:svc_idle_timeout**

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

Data Type: Long integer (32 bits on 32-bit platforms and 64 bits on 64-bit platforms)

Default: 360,000 milliseconds (6 minutes)

Range: 0 to $2^{31} - 1$ on 32-bit platforms

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 server before being closed. You might want to close connections at a faster rate to avoid consuming system resources.

Commitment Level: Unstable

**rpcmod:svc_default_stksize**

Description: Sets the size of the kernel stack for kernel RPC service threads.

Data Type: Integer (32-bit)

Default: The default value is 0. This value means that the stack size is set to the system default.

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

Units: Bytes

Dynamic?: Yes, for all new threads that are allocated. The stack size is set when the thread is created. Therefore, changes to this parameter do not affect existing threads but are applied to all new threads that are allocated.

Validation: None

When to Change: Very deep call depths can cause the stack to overflow and cause red zone faults. The combination of a fairly deep call depth for the transport, coupled with a deep call depth for the local file system, can cause NFS service threads to overflow their stacks.
Set this parameter to a multiple of the hardware page size on the platform.

**Commitment Level**  Unstable

### rpcmod:maxdupreqs

**Description**  Controls the size of the duplicate request cache that detects RPC-level retransmissions on connectionless transports. This cache is indexed by the client network address and the RPC procedure number, program number, version number, and transaction ID. This cache avoids processing retransmitted requests that might not be idempotent.

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

**Default**  1024

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

**Units**  Requests

**Dynamic?**  The cache is dynamically sized, but the hash queues that provide fast access to the cache are statically sized. Making the cache very large might result in long search times to find entries in the cache.

Do not set the value of this parameter to 0. This value prevents the NFS server from handling non idempotent requests.

**Validation**  None

**When to Change**  Examine the value of this parameter if false failures are encountered by NFS clients. For example, if an attempt to create a directory fails, but the directory is actually created, perhaps that retransmitted `MKDIR` request was not detected by the server.

The size of the cache should match the load on the server. The cache records non idempotent requests and so only needs to track a portion of the total requests. The cache does need to hold the information long enough to be able to detect a retransmission by the client. Typically, the client timeout for connectionless transports is relatively short, starting around 1 second and increasing to about 20 seconds.

**Commitment Level**  Unstable
**rpcmod:cotsmaxdupreqs**

**Description**
Controls the size of the duplicate request cache that detects RPC-level retransmissions on connection-oriented transports. This cache is indexed by the client network address and the RPC procedure number, program number, version number, and transaction ID. This cache avoids processing retransmitted requests that might not be idempotent.

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

**Default**
8192

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

**Units**
Requests

**Dynamic?**
Yes

**Validation**
The cache is dynamically sized, but the hash queues that provide fast access to the cache are statically sized. Making the cache very large might result in long search times to find entries in the cache.

Do not set the value of this parameter to 0. It prevents the NFS server from handling non-idempotent requests.

**When to Change**
Examine the value of this parameter if false failures are encountered by NFS clients. For example, if an attempt to create a directory fails, but the directory is actually created, it is possible that a retransmitted `MKDIR` request was not detected by the server.

The size of the cache should match the load on the server. The cache records non-idempotent requests and so only needs to track a portion of the total requests. It does need to hold the information long enough to be able to detect a retransmission on the part of the client. Typically, the client timeout for connection oriented transports is very long, about 1 minute. Thus, entries need to stay in the cache for fairly long times.

**Commitment Level**
Unstable
Internet Protocol Suite Tunable Parameters

This chapter describes various Internet Protocol suite parameters, such as TCP, IP, UDP, and SCTP.

- "IP Tunable Parameters" on page 138
- "TCP Tunable Parameters" on page 144
- "UDP Tunable Parameters" on page 161
- "IPQoS Tunable Parameter" on page 163
- "SCTP Tunable Parameters" on page 164
- "Per-Route Metrics" on page 174

Where to Find Tunable Parameter Information

<table>
<thead>
<tr>
<th>Tunable Parameter</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris kernel tunable parameters</td>
<td>Chapter 2, “Oracle Solaris Kernel Tunable Parameters”</td>
</tr>
<tr>
<td>Oracle Solaris ZFS kernel tunable parameters</td>
<td>Chapter 3, “Oracle Solaris ZFS Tunable Parameters”</td>
</tr>
<tr>
<td>NFS tunable parameters</td>
<td>Chapter 4, &quot;NFS Tunable Parameters&quot;</td>
</tr>
</tbody>
</table>

Overview of Tuning IP Suite Parameters

For new information about IP forwarding, see “New and Changed TCP/IP Parameters” on page 21.

You can set all of the tuning parameters described in this chapter by using the `nld` command except for the following parameters:

- "ipcl_conn_hash_size" on page 156
IP Tunable Parameters

- “ip_queue_worker_wait” on page 157

These parameters can only be set in the /etc/system file.

Use the following syntax to set TCP/IP parameters by using the ndd command:

```
# ndd -set driver parameter
```

For more information, see ndd(1M).

Although the SMF framework provides a method for managing system services, ndd commands are still included in system startup scripts. For more information on creating a startup script, see “Using Run Control Scripts” in Oracle Solaris Administration: Basic Administration.

IP Suite Parameter Validation

All parameters described in this section 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 (RFCs)

Internet protocol and standard specifications are described in RFC documents. You can get copies of RFCs from ftp://ftp.rfc-editor.org/in-notes. Browse RFC topics by viewing the rfc-index.txt file at this site.

IP Tunable Parameters

**ip_icmp_err_interval and ip_icmp_err_burst**

<table>
<thead>
<tr>
<th>Description</th>
<th>Controls the rate of IP in generating IPv4 or IPv6 ICMP error messages. IP generates only up to ip_icmp_err_burst IPv4 or IPv6 ICMP error messages in any ip_icmp_err_interval.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>100 milliseconds for ip_icmp_err_interval 10 error messages for ip_icmp_err_burst</td>
</tr>
<tr>
<td>Range</td>
<td>0 – 99,999 milliseconds for ip_icmp_err_interval 1 – 99,999 error messages for ip_icmp_err_burst</td>
</tr>
</tbody>
</table>
Dynamic?  Yes
When to Change  If you need a higher error message generation rate for diagnostic purposes.
Commitment Level  Unstable

**ip_respond_to_echo_broadcast and ip6_respond_to_echo_multicast**

Description  Controls whether IPv4 or IPv6 responds to a broadcast ICMPv4 echo request or a multicast 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

**ip_send_redirects and ip6_send_redirects**

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  Unstable

**ip_forward_src_routed and ip6_forward_src_routed**

Description  Controls whether IPv4 or IPv6 forwards packets with source IPv4 routing options or IPv6 routing headers.
Default  0 (disabled)
Range  0 (disabled) or 1 (enabled)
Dynamic? Yes
When to Change Keep this parameter disabled to prevent denial of service attacks.
Commitment Level Unstable
Change History For information, see “ip_forward_src_routed and ip6_forward_src_routed (Solaris 10 Releases)” on page 184.

### ip_addrs_per_if

**Description** Defines the maximum number of logical interfaces associated with a real interface.

**Default** 256

**Range** 1 to 8192

**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

### ip_strict_dst_multihoming and ip6_strict_dst_multihoming

**Description** Determines whether a packet arriving on a non forwarding interface can be accepted for an IP address that is not explicitly configured on that interface. If ip_forwarding is enabled, or xxx:ip_forwarding for the appropriate interfaces is enabled, then this parameter is ignored, because the packet is actually forwarded.

Refer to RFC 1122, 3.3.4.2.

**Default** 0 (loose multihoming)

**Range**

0 = Off (loose multihoming)

1 = On (strict multihoming)

**Dynamic?** Yes

**When to Change** If a machine has interfaces that cross strict networking domains (for example, a firewall or a VPN node), set this parameter to 1.
Commitment Level  Unstable

**ip_multidata_outbound**

Description  Enables the network stack to send more than one packet at one time to the network device driver during transmission.

Enabling this parameter reduces the per-packet processing costs by improving host CPU utilization, network throughput, or both.

This parameter now controls the use of multidata transmit (MDT) for transmitting IP fragments. For example, when sending out a UDP payload larger than the link MTU. When this tunable is enabled, IP fragments of a particular upper-level protocol, such as UDP, are delivered in batches to the network device driver. Disabling this feature results in both TCP and IP fragmentation logic in the network stack to revert back to sending one packet at a time to the driver.

The MDT feature is only effective for device drivers that support this feature.

See also “**tcp_mdt_max_pbufs**” on page 154.

Default  1 (Enabled)

Range  0 (disabled) or 1 (enabled)

Dynamic?  Yes

When to Change  If you do not want this parameter enabled for debugging purposes or for any other reasons, disable it.

Commitment Level  Unstable

Change History  For information, see “**ip_multidata_outbound (Solaris 10 Releases)**” on page 185.

**ip_squeue_fanout**

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. The number of squeues that are used to fanout the connection is based upon “**ip_soft_rings_cnt**” on page 142
Default 0
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
Change History For information, see “ip_squeue_fanout (Solaris 10 11/06 Release)” on page 185.

**ip_soft_rings_cnt**

Description Determines the number of queues to be used to fanout the incoming TCP/IP connections.

---

**Note** – The incoming traffic is placed on one of the rings. If the ring is overloaded, packets are dropped. For every packet that gets dropped, the kstat dls counter, dls_soft_ring_pkt_drop, is incremented.

Default 2
Range 0 - nCPUs, where nCPUs is the maximum number of CPUs in the system
Dynamic? No. The interface should be plumbed again when changing this parameter.
When to Change Consider setting this parameter to a value greater than 2 on systems that have 10 Gbps NICs and many CPUs.
Zone Configuration This parameter can only be set in the global zone.
Commitment Level Obsolete
Change History For information, see “ip_soft_rings_cnt (Solaris 10 11/06 Release)” on page 185.
**ip_pmtu_min**

**Description**
Determines the minimum value upon which the Path MTU Discovery (PMTUD) protocol operates.

This parameter has no direct interactions with other tunables except in the case of the TCP protocol when the highest of the two tunables, tcps_mss_min and ip_pmtu_min are used to decide the lowest MTU at which to perform PMTUD for TCP.

**Default**
576

**Range**
68 to 65,535

**Dynamic?**
No

**When to Change**
This parameter can be adjusted to override PMTUD behavior, but do not tune this parameter below 68 (bytes). When changing this parameter, consider the following:

- Match the lowest MTU size in the environment for which PMTUD is operating.
- When ip_pmtu_min is below this threshold, PMTUD is turned off.
- Set this parameter to 68 to bring back the former PMTUD behavior on systems that run a pre-Solaris 10 8/11 release or a system with a patch level prior to 144488-12 or 144489-12.

**Commitment Level**
Unstable

### IP Tunable Parameters With Additional Cautions

Changing the following parameters is not recommended.

**ip_ire_pathmtu_interval**

**Description**
Specifies the interval in milliseconds when IP flushes the path maximum transfer unit (PMTU) discovery information, and tries to rediscover PMTU.

Refer to RFC 1191 on PMTU discovery.

**Default**
10 minutes

**Range**
5 seconds to 277 hours

**Dynamic?**
Yes
ip_icmp_return_data_bytes and ip6_icmp_return_data_bytes

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 bytes (ip_icmp_return_data_bytes)
1280 bytes (ip6_icmp_return_data_bytes)

Range: 8 to 65,536 bytes

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

TCP Tunable Parameters

tcp_deferred_ack_interval

Description: Specifies the time-out value for the TCP-delayed acknowledgment (ACK) timer for hosts that are not directly connected.

Default: 100 milliseconds

Range: 1 millisecond to 1 minute

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

**tcp_local_dack_interval**

Description Specifies the time-out value for TCP-delayed acknowledgment (ACK) timer for hosts 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

Change History For information, see “tcp_local_dack_interval (Solaris 10 Releases)” on page 185.

**tcp_deferred_acks_max**

Description Specifies the maximum number of TCP segments received from remote destinations (not directly connected) 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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Range</th>
<th>Dynamic?</th>
<th>When to Change</th>
<th>Commitment Level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When to Change</strong></td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unstable</td>
</tr>
<tr>
<td><strong>Commitment Level</strong></td>
<td>Unstable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>tcp_local_dacks_max</strong></td>
<td>Specifies the maximum number of TCP segments received from directly connected destinations 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.</td>
<td>8</td>
<td>0 to 16</td>
<td>Yes</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>0 to 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dynamic?</strong></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>When to Change</strong></td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Commitment Level</strong></td>
<td>Unstable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>tcp_wscale_always</strong></td>
<td>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.</td>
<td>1 (enabled)</td>
<td>0 (disabled) or 1 (enabled)</td>
<td>Yes</td>
<td>Refer to RFC 1323 for the window scale option.</td>
<td></td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>1 (enabled)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>0 (disabled) or 1 (enabled)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dynamic?</strong></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

### tcp_tstamp_always

**Description** | If set to 1, TCP always sends a SYN segment with the timestamp option. 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) or 1 (enabled)
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.
---|---
Commitment Level | Unstable

### tcp_xmit_hiwat

**Description** | Defines the default send window size in bytes. Refer to “Per-Route Metrics” on page 174 for a discussion of setting a different value on a per-route basis. See also “tcp_max_buf” on page 148.
---|---
Default | 49,152
Range | 4096 to 1,073,741,824
Dynamic? | Yes
When to Change | An application can use setsockopt(3XNET) SO_SNDBUF to change the individual connection’s send buffer.
---|---
Commitment Level | Unstable

### tcp_recv_hiwat

**Description** | Defines the default receive window size in bytes. Refer to “Per-Route Metrics” on page 174 for a discussion of setting a different value on a
### TCPTunable Parameters

- **tcp_max_buf**

  **Description**: Defines the maximum buffer size in bytes. This parameter controls how large the send and receive buffers are set to by an application that uses `setsockopt(3XNET)`.

  **Default**: 1,048,576

  **Range**: 8192 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.

  **Commitment Level**: Unstable

- **tcp_cwnd_max**

  **Description**: Defines the maximum value of the TCP congestion window (cwnd) in bytes.

  For more information on 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**: Even if an application uses `setsockopt(3XNET)` to change the window size to a value higher than `tcp_cwnd_max`, the actual window used can never grow beyond `tcp_cwnd_max`. Thus, `tcp_max_buf` should be greater than `tcp_cwnd_max`.

  **Commitment Level**: Unstable
Commitment Level: Unstable

**tcp_slow_start_initial**

Description: Defines the maximum initial congestion window (cwnd) 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: 4

Range: 1 to 4

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

**tcp_slow_start_after_idle**

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 “tcp_slow_start_initial” on page 149.

Commitment Level: Unstable

**tcp_sack_permitted**

Description: If set to 2, 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 1, TCP responds with a SACK-permitted option. If the parameter is set to 0, 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 on the SACK option.

Default 2 (active enabled)
Range 0 (disabled), 1 (passive enabled), or 2 (active enabled)
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 1 so that SACK processing is enabled only when incoming connections allow SACK processing.

Commitment Level Unstable

tcp_rev_src_routes

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

tcp_time_wait_interval

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 10 minutes
Dynamic? Yes
When to Change Do not set the value lower than 60 seconds.
For information on changing this parameter, refer to RFC 1122, 4.2.2.13.
Commitment Level Unstable

tcp_ecn_permitted

Description Controls Explicit Congestion Notification (ECN) support.

If this parameter is set to 0, TCP does not negotiate with a peer that supports the ECN mechanism.

If this parameter is set to 1 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 2, 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 on ECN.

Default 1 (passive enabled)
Range 0 (disabled), 1 (passive enabled), or 2 (active enabled)
Dynamic? Yes
When to Change ECN can help TCP better handle congestion control. However, there are existing TCP implementations, firewalls, NATs, and other network devices that are confused by this mechanism. These devices do not comply to the IETF standard.

Because of these devices, the default value of this parameter is set to 1. In rare cases, passive enabling can still cause problems. Set the parameter to 0 only if absolutely necessary.

Commitment Level Unstable
### tcp_conn_req_max_q

**Description**
Specifies the default maximum number of pending TCP connections for a TCP listener waiting to be accepted by `accept(3SOCKET)`. See also “tcp_conn_req_max_q0” on page 152.

**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 `tcp_conn_req_max_q` does not mean that applications can have that many pending TCP connections. Applications can use `listen(3SOCKET)` 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 `tcp_conn_req_max_q`, depending on the value used in `listen()`.

**Commitment Level**
Unstable

### tcp_conn_req_max_q0

**Description**
Specifies the default maximum number of incomplete (three-way handshake not yet finished) pending TCP connections for a TCP listener.

For more information on TCP three-way handshake, refer to RFC 793. See also “tcp_conn_req_max_q” on page 152.

**Default**
1024

**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 \texttt{tcp\_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 \texttt{tcp\_conn\_req\_max\_q0}. If it does not, the request is accepted. Otherwise, the oldest incomplete pending TCP request is dropped.

Commitment Level

Unstable

\textbf{tcp\_conn\_req\_min}

\textbf{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 \texttt{listen(3SOCKET)} that an application can use.

\textbf{Default} \hspace{1cm} 1

\textbf{Range} \hspace{1cm} 1 to 1024

\textbf{Dynamic?} \hspace{1cm} Yes

\textbf{When to Change}

This parameter can be a solution for applications that use \texttt{listen(3SOCKET)} 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

\textbf{tcp\_rst\_sent\_rate\_enabled}

\textbf{Description}

If this parameter is set to 1, the maximum rate of sending a RST segment is controlled by the \texttt{ndd} parameter, \texttt{tcp\_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

### tcp_rst_sent_rate

**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

### tcp_mdt_max_pbuds

**Description** Specifies the number of payload buffers that can be carried by a single M_MULTIDATA message that is generated by TCP. See also “ip_multidata_outbound” on page 141.

**Default** 16

**Range** 1 to 16

**Dynamic?** Yes

**When to Change** Decreasing this parameter might aid in debugging device driver development by limiting the amount of payload buffers per M_MULTIDATA message that is generated by TCP.

**Commitment Level** Unstable
**tcp_naglim_def**

**Description**
This parameter controls the Nagle algorithm threshold. TCP uses the minimum of this parameter and the MSS of a connection to determine when the Nagle algorithm should kick in. For example, if the amount of new data is more than 1 MSS, the data is sent out regardless of the value of this parameter. If this parameter is set to 1, the Nagle is disabled for all TCP connections.

**Default**
4,096

**Range**
1 to 65,535

**Dynamic?**
Yes

**When to Change**
Real-time applications that need to send data without delay should use `setsockopt()` to set `TCP_NODELAY` to 1 for the sockets needing fast transmission rather than setting the `tcp_naglim_def` parameter.

**Commitment Level**
Unstable

**Change History**
For information, see “tcp_naglim_def (Solaris 10 Releases)” on page 186.

---

**tcp_smallest_anon_port**

**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 and it does not specify 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**
Unstable

**Change History**
For information, see “[tcp,sctp,udp]_smallest_anon_port and [tcp,sctp,udp]_largest_anon_port” on page 185.
**tcp_largest_anon_port**

**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 and it does not specify 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.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Port number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>65,535</td>
</tr>
<tr>
<td>Range</td>
<td>1,024 to 65,535</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>When to Change</td>
<td>When a larger ephemeral port range is required.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
<tr>
<td>Change History</td>
<td>For information, see “[tcp, sctp, udp]_smallest_anon_port and [tcp, sctp, udp]_largest_anon_port” on page 185.</td>
</tr>
</tbody>
</table>

**TCP/IP Parameters Set in the /etc/system File**

The following parameters can be set only in the /etc/system file. 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_sizes=value
```

**ipcl_conn_hash_size**

**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.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Unsigned integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td>Range</td>
<td>0 to 82,500</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>No. The parameter can only be changed at boot time.</td>
</tr>
</tbody>
</table>
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

**ip_squeue_worker_wait**

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 machine serves mostly interactive network traffic.

The default value usually works best on a network file server, a web server, or any server that has substantial network traffic.

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

Commitment Level  Unstable

Change History  For information, see “ip_squeue_worker_wait (Solaris 10 11/06 Release)” on page 185.

---

**TCP Parameters With Additional Cautions**

Changing the following parameters is not recommended.

**tcp_keepalive_interval**

Description  This ndd 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Range</th>
<th>Units</th>
<th>Dynamic?</th>
<th>When to Change</th>
<th>Commitment Level</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tcp_keepalive_interval</code></td>
<td>2 hours</td>
<td>10 seconds to 10 days</td>
<td>Unsigned integer (milliseconds)</td>
<td>Yes</td>
<td>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.</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**tcp_ip_abort_interval**

Description: Specifies the default total retransmission timeout value for a TCP connection. For a given TCP connection, if TCP has been retransmitting for `tcp_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 "tcp_rexmit_interval_max" on page 159.

<table>
<thead>
<tr>
<th>Default</th>
<th>5 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>500 milliseconds to 1193 hours</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>Yes</td>
</tr>
<tr>
<td>When to Change</td>
<td>Do not change this value. See “tcp_rexmit_interval_max” on page 159 for exceptions.</td>
</tr>
</tbody>
</table>

Commitment Level: Unstable

**tcp_rexmit_interval_initial**

Description: Specifies the default initial retransmission timeout (RTO) value for a TCP connection. Refer to “Per-Route Metrics” on page 174 for a discussion of setting a different value on a per-route basis.

<table>
<thead>
<tr>
<th>Default</th>
<th>3 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>1 millisecond to 20 seconds</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tcp_rexmit_interval_max</td>
<td>Defines the default maximum retransmission timeout value (RTO). The calculated RTO for all TCP connections cannot exceed this value. See also “tcp_ip_abort_interval” on page 158.</td>
</tr>
<tr>
<td>tcp_rexmit_interval_min</td>
<td>Specifies the default minimum retransmission timeout (RTO) value. The calculated RTO for all TCP connections cannot be lower than this value. See also “tcp_rexmit_interval_max” on page 159.</td>
</tr>
</tbody>
</table>
Commitment Level Unstable

**tcp_rexmit_interval_extra**

**Description**
Specifies a constant added to the calculated retransmission time out value (RTO).

**Default**
0 milliseconds

**Range**
0 to 2 hours

**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

**tcp_tstamp_if_wscale**

**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

**tcp_recv_hiwat_minmss**

**Description**
Controls the default minimum receive window size. The minimum is tcp_recv_hiwat_minmss times the size of maximum segment size (MSS) of a connection.

**Default**
4

**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

**UDP Tunable Parameters**

**udp_xmit_hiwat**

Description: Defines the default maximum UDP socket datagram size. For more information, see “udp_max_buf” on page 163.

Default: 57,344 bytes

Range: 1,024 to 1,073,741,824 bytes

Dynamic?: Yes

When to Change: Note that an application can use `setsockopt(3XNET) SO_SNDBUF` to change the size for an individual socket. In general, you do not need to change the default value.

Commitment Level: Unstable

**udp_recv_hiwat**

Description: Defines the default maximum UDP socket receive buffer size. For more information, see “udp_max_buf” on page 163.

Default: 57,344 bytes

Range: 128 to 1,073,741,824 bytes

Dynamic?: Yes

When to Change: Note that an application can use `setsockopt(3XNET) SO_RCVBUF` to change the size for an individual socket. In general, you do not need to change the default value.

Commitment Level: Unstable
**udp_smallest_anon_port**

**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 and it does not specify 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**
Unstable

**Change History**
For information, see "[tcp,sctp,udp]_smallest_anon_port and [tcp,sctp,udp]_largest_anon_port" on page 185.

**udp_largest_anon_port**

**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 and it does not specify 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**
1,024 to 65,535

**Dynamic?**
Yes

**When to Change**
When a larger ephemeral port range is required.

**Commitment Level**
Unstable

**Change History**
For information, see "[tcp,sctp,udp]_smallest_anon_port and [tcp,sctp,udp]_largest_anon_port" on page 185.
udp_do_checksum

Description This parameter controls whether UDP calculates the checksum on outgoing UDP/IPv4 packets.

Default 1 (enabled)

Range 0 (disabled) or 1 (enabled)

Dynamic? Yes

When to Change Do not change this parameter.

Commitment Level Unstable

Change History For information, see “udp_do_checksum (Solaris 10 Releases)” on page 186.

UDP Parameter With Additional Caution

Changing the following parameter is not recommended.

udp_max_buf

Description Controls how large send and receive buffers can be for a UDP socket.

Default 2,097,152 bytes

Range 65,536 to 1,073,741,824 bytes

Dynamic? Yes

When to Change Do not change the value. If this parameter is set to a very large value, UDP socket applications can consume too much memory.

Commitment Level Unstable

IPQoS Tunable Parameter

ip_policy_mask

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:
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

SCTP Tunable Parameters

**sctp_max_init_retr**

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 “sctp_pa_max_retr” on page 165. Ideally, sctp_max_init_retr should be less than or equal to sctp_pa_max_retr.

Commitment Level: Unstable
**sctp_pa_max_retr**

**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 "sctp_pp_max_retr" on page 165 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 `sctp_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

---

**sctp_pp_max_retr**

**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

---

**sctp_cwnd_max**

**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

Even if an application uses `setsockopt(3XNET)` to change the window size to a value higher than `sctp_cwnd_max`, the actual window used can never grow beyond `sctp_cwnd_max`. Thus, "`sctp_max_buf` on page 169 should be greater than `sctp_cwnd_max`.

**Commitment Level**: Unstable

---

#### sctp_ipv4_ttl

**Description**: Controls the time to live (TTL) value in the IP version 4 header for the outbound IP version 4 packets on an SCTP association.

- **Default**: 64
- **Range**: 1 to 255
- **Dynamic?**: Yes
- **When to Change**: Generally, you do not need to change this value. Consider increasing this parameter if the path to the destination is likely to span more than 64 hops.

**Commitment Level**: Unstable

---

#### sctp_heartbeat_interval

**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

---

#### sctp_new_secret_interval

**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

**sctp_initial_mtu**

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

**sctp_deferred_ack_interval**

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

**sctp_ignore_path_mtu**

Description: Enables or disables path MTU discovery.

Default: 0 (disabled)
Range: 0 (disabled) or 1 (enabled)
Dynamic?: Yes
### sctp_initial_ssthresh

- **Description**: Sets the initial slow start threshold for a destination address of the peer.
- **Default**: 102,400
- **Range**: 1024 to 4,294,967,295
- **Dynamic?**: Yes
- **When to Change**: Refer to RFC 2960, section 7.2.1.
- **Commitment Level**: Unstable

### sctp_xmit_hiwat

- **Description**: Sets the default send window size in bytes. See also “sctp_max_buf” on page 169.
- **Default**: 102,400
- **Range**: 8,192 to 1,073,741,824
- **Dynamic?**: Yes
- **When to Change**: An application can use `getsockopt(3SOCKET) SO_SNDBUF` to change the individual association’s send buffer.
- **Commitment Level**: Unstable

### sctp_xmit_lowat

- **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. If required, consider changing this parameter in accordance with "sctp_xmit_hiwat" on page 168.

Commitment Level Unstable

**sctp_recv_hiwat**

Description Controls the default receive window size in bytes. See also "sctp_max_buf" on page 169.

Default 102,400
Range 8,192 to 1,073,741,824
Dynamic? Yes
When to Change An application can use getsockopt(3SOCKET) SO_RCVBUF to change the individual association’s receive buffer.

Commitment Level Unstable

**sctp_max_buf**

Description Controls the maximum buffer size in bytes. It controls how large the send and receive buffers are set to by an application that uses getsockopt(3SOCKET).

Default 1,048,576
Range 8,192 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 Unstable

**sctp_ipv6_hoplimit**

Description Sets the value of the hop limit in the IP version 6 header for the outbound IP version 6 packets on an SCTP association.

Default 60
Range 0 to 255
Dynamic? Yes
**When to Change**  Generally, you do not need to change this value. Consider increasing this parameter if the path to the destination is likely to span more than 60 hops.

**Commitment Level**  Unstable

<table>
<thead>
<tr>
<th><strong>sctp_rto_min</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Sets the lower bound for the retransmission timeout (RTO) in milliseconds for all the destination addresses of the peer.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>1,000</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>500 to 60,000</td>
</tr>
<tr>
<td><strong>Dynamic?</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>When to Change</strong></td>
<td>Refer to RFC 2960, section 6.3.1.</td>
</tr>
<tr>
<td><strong>Commitment Level</strong></td>
<td>Unstable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>sctp_rto_max</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Controls the upper bound for the retransmission timeout (RTO) in milliseconds for all the destination addresses of the peer.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>60,000</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>1,000 to 60,000,000</td>
</tr>
<tr>
<td><strong>Dynamic?</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>When to Change</strong></td>
<td>Refer to RFC 2960, section 6.3.1.</td>
</tr>
<tr>
<td><strong>Commitment Level</strong></td>
<td>Unstable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>sctp_rto_initial</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Controls the initial retransmission timeout (RTO) in milliseconds for all the destination addresses of the peer.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>3,000</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>1,000 to 60,000,000</td>
</tr>
<tr>
<td><strong>Dynamic?</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>When to Change</strong></td>
<td>Refer to RFC 2960, section 6.3.1.</td>
</tr>
<tr>
<td><strong>Commitment Level</strong></td>
<td>Unstable</td>
</tr>
</tbody>
</table>
## sctp_cookie_life

**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 "sctp_rto_max" on page 170.

**Commitment Level**
Unstable

## sctp_max_in_streams

**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

## sctp_initial_out_streams

**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

## sctp_shutack_wait_bound

**Description**
Controls the maximum time, in milliseconds, to wait for a SHUTDOWN ACK after having sent a SHUTDOWN chunk.
| **sctp_rto_max** | 
|---|---|
| **Description** | Sets the limit on the number of segments to be sent in a burst. |
| **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 “sctp_rto_max” on page 170. |
| **Commitment Level** | Unstable |

| **sctp_maxburst** | 
|---|---|
| **Description** | Enables or disables the partial reliability extension (RFC 3758) to SCTP. |
| **Default** | 1 (enabled) |
| **Range** | 0 (disabled) or 1 (enabled) |
| **Dynamic?** | Yes |
| **Commitment Level** | Unstable |

| **sctp_addip_enabled** | 
|---|---|
| **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 |

| **sctp_prsctp_enabled** | 
|---|---|
| **Description** | Enables or disables the partial retransmission (RFC 3758) to SCTP. |
| **Default** | 0 (disabled) |
| **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

**sctp_smallest_anon_port**

**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 and it does not specify 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**

Unstable

**sctp_largest_anon_port**

**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 and it does not specify 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**

1,024 to 65,535

**Dynamic?**

Yes

**When to Change**

When a larger ephemeral port range is required.

**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 tcp_recv_hiwat is 49,152 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 tcp_recv_hiwat, 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:

```
192.123.123.0 192.123.123.4 U 1 4 hme0
192.123.124.0 192.123.124.4 U 1 4 ge0
default 192.123.123.1 UG 1 8
```

In this example, do the following:

```
# route change -net 192.123.124.0 -recvpipe x
```

Then, all connections going to the 192.123.124.0 network, which is on the ge0 link, use the receive buffer size x, instead of the default 49,152 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 192.123.123.1 -netmask w.x.y.z
# route change -net a.b.c.d -recvpipe y
```

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(1M) get command.
This chapter describes most of the parameters default values for various system facilities.

- "autofs" on page 176
- "cron" on page 176
- "devfsadm" on page 176
- "dhcpagent" on page 176
- "fs" on page 177
- "ftp" on page 177
- "inetinit" on page 177
- "init" on page 177
- "ipsec" on page 177
- "kbd" on page 177
- "keyserv" on page 178
- "login" on page 178
- "lu" on page 178
- "mpathd" on page 178
- "nfs" on page 178
- "nfslogd" on page 179
- "nss" on page 179
- "passwd" on page 179
- "power" on page 179
- "rpc.nisd" on page 179
- "su" on page 179
- "syslog" on page 179
- "sys-suspend" on page 179
- "tar" on page 179
- "telnetd" on page 180
- "utmpd" on page 180
- "yppasswdd" on page 180
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 stored in a file for each facility are located in the `/etc/default` directory. Not every system facility has a file located in this directory.

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 on SMF services and properties, see "Managing SMF Services" in Oracle Solaris Administration: Basic Administration.

For information about setting power management properties, see Chapter 3, "Managing Serial Ports With the Service Access Facility (Tasks)," in System Administration Guide: Advanced Administration.

autofs

This facility enables you to configure autofs parameters such as automatic timeout, displaying or logging status messages, browsing autofs mount points, and tracing. For details, see autofs(4).

cron

This facility enables you to disable or enable cron logging.

devfsadm

This file is not currently used.

dhcpagent

Client usage of DHCP is provided by the dhcpagent daemon. When `ifconfig` identifies an interface that has been configured to receive its network configuration from DHCP, it starts the client daemon to manage that interface.

For more information, see the `/etc/default/dhcpagent` information in the FILES section of dhcpagent(1M).
**fs**

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 `default_fs(4)`.

**ftp**

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 Solaris release.

For details, see `ftp(4)`.

**inetinit**

This facility enables you to configure TCP sequence numbers and to enable or disable support for 6to4 relay routers.

**init**

For details, see the `/etc/default/init` information in the FILES section of `init(1M)`.

All values in the file are placed in the environment of the shell that `init` invokes in response to a single user boot request. The `init` process also passes these values to any commands that it starts or restarts from the `/etc/inittab` file.

**ipsec**

This facility enables you to configure parameters, such as IKE daemon debugging information and the `ikeadm` privilege level.

**kbd**

For details, see the Extended Description section of `kbd(1)`.

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 kbd(1).

**keyserv**

For details, see the /etc/default/keyserv information in the FILES section of keyserv(1M).

**login**

For details, see the /etc/default/login information in the FILES section of login(1).

**lu**

This file contains default settings for the Oracle Solaris Live Upgrade feature.

**mpathd**

This facility enables you to set in.mpathd configuration parameters.

For details, see in.mpathd(1M).

**nfs**

This facility enables you to set NFS daemon configuration parameters.

For details, see nfs(4).
**nfslogd**
For details, see the Description section of `nfslogd(1M)`.

**nss**
This facility enables you to configure `initgroups(3C)` lookup parameters.
For details, see `nss(4)`.

**passwd**
For details, see the `/etc/default/passwd` information in the FILES section of `passwd(1)`.

**power**
For details, see the `/etc/default/power` information in the FILES section of `pmconfig(1M)`.

**rpc.nisd**
For details, see the `/etc/default/rpc.nisd` information in the FILES section of `rpc.nisd(1M)`.

**su**
For details, see the `/etc/default/su` information in the FILES section of `su(1M)`.

**syslog**
For details, see the `/etc/default/syslogd` information in the FILES section of `syslogd(1M)`.

**sys-suspend**
For details, see the `/etc/default/sys-suspend` information in the FILES section of `sys-suspend(1M)`.

**tar**
For a description of the `-f` function modifier, see `tar(1)`.
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 archiveN string is looked up in the /etc/default/tar file. The value of the archiveN 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

This file identifies the default BANNER that is displayed upon a telnet connection.

utmpd

The utmpd daemon monitors /var/adm/utmpx (and /var/adm/utmp in earlier Solaris versions) to ensure that utmp entries inserted by non-root processes by putxline(3C) 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.

yppasswdd

This facility enables you to configure whether a user can successfully set a login shell to a restricted shell when using the passwd -r nis -e command.

For details, see rpc.yppasswdd(1M).
This chapter describes the change history of specific tunable parameters. If a parameter is in this section, it has changed from a previous release. Parameters whose functionality has been removed are listed also.

- "Kernel Parameters" on page 181
- "NFS Tunable Parameters" on page 184
- "TCP/IP Tunable Parameters" on page 184
- "Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)" on page 186

Kernel Parameters

General Kernel and Memory Parameters (Oracle Solaris 10)

**zfs_arc_min**

The `zfs_arc_min` parameter information has moved to Chapter 3, "Oracle Solaris ZFS Tunable Parameters."

**zfs_arc_max**

The `zfs_arc_max` parameter information has moved to Chapter 3, "Oracle Solaris ZFS Tunable Parameters."

**noexec_user_stack (Solaris 10 Releases)**

The Solaris 10 description section was updated by removing the text “and sun4m” and adding the text “64–bit SPARC and AMD64.”
The Solaris 10 description section was updated by adding default and maximum values for AMD64.

The Solaris 10 default value for SPARC platforms was changed to 24,576.

**Paging-Related Parameters**

**fastscan**
The default value of `fastscan` was clarified. For more information, see “`fastscan`” on page 52.

**Process-Sizing Tunables**

**ngroups_max (Solaris 10 Releases)**
This parameter was undocumented in previous Solaris releases. The default maximum has been increased to 1024 groups. For more information, see “`ngroups_max`” on page 44.

**max_nprocs (Solaris 10 Releases)**
The Solaris 10 description section was updated by removing the text "sun4m.”

**UFS Parameter**

**ufsd_delete_hiwat**
This parameter is new in the Solaris 10 1/13 release. For more information, see “`ufsd_delete_hiwat`” on page 74.

**General Driver Parameter**

**ddi_msix_alloc_limit**
This parameter is newly documented. For more information, see “`ddi_msix_alloc_limit`” on page 61.
General I/O Tunable Parameters

maxphys (Solaris 10 Releases)
The default value is updated to include sun4v systems. For more information, see “maxphys” on page 62.

fsflush and Related Parameters

dopageflush (Solaris 10 Releases)
The description was clarified by including that number of physical memory pages are examined.

Paging-Related Tunable Parameters

maxpgio (Solaris 10 Releases)
In the Solaris 10 versions, the range value was incorrectly documented as 1 to 1024. The actual range depends on system architecture and I/O subsystems. For more information, see “maxpgio” on page 56.

General File System Parameters

ncsize (Solaris 10 Release)
In previous Solaris 10 releases, the default value of the ncsiz parameter was incorrectly described as follows:

\[4 \times (v.v_{proc} + maxusers) + 320 / 100\]

The correct default value is as follows:

\[(4 \times (v.v_{proc} + maxusers) + 320) + (4 \times (v.v_{proc} + maxusers) + 320 / 100)\]

For more information, see “ncsize” on page 64.

TMPFS Parameters

tmpfs:tmpfs_maxkmem (Solaris 10 Releases)
The range description is updated to include sun4v systems. For more information, see “tmpfs:tmpfs_maxkmem” on page 75.
SPARC System Specific Parameters (Solaris 10 Releases)

The title of the SPARC System Specific Parameters section was revised in the Solaris 10 8/07 release to include sun4v systems.

**default_tsb_size (Solaris 10 Releases)**

The default description has changed. For more information, see “default_tsb_size” on page 87.

**enable_tsb_rss_sizing (Solaris 10 Releases)**

The description and default and range values have changed. For more information, see “enable_tsb_rss_sizing” on page 88.

**tsb_rss_factor (Solaris 10 Releases)**

The when to change example text was changed to this:

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.

NFS Tunable Parameters

**nfs:nfs3_nra (Solaris 10 Releases)**

The default value was incorrectly documented in previous Solaris 10 releases. The default value is 4.

TCP/IP Tunable Parameters

**ip_forward_src_routed and ip6_forward_src_routed (Solaris 10 Releases)**

The default value of these parameters was incorrectly documented in previous Solaris 10 releases. The correct default value is disabled. For more information, see “ip_forward_src_routed and ip6_forward_src_routed” on page 139.
**ip_multidata_outbound (Solaris 10 Releases)**

This parameter was enhanced in the Solaris 10 release to deliver IP fragments in batches to the network driver. For more information, see “ip_multidata_outbound” on page 141.

**ip_squeue_fanout (Solaris 10 11/06 Release)**

Zone configuration information was added in the Solaris 10 8/07 release. For more information, see “ip_squeue_fanout” on page 141.

**ip_squeue_worker_wait (Solaris 10 11/06 Release)**

Zone configuration information was added in the Solaris 10 8/07 release. For more information, see “ip_squeue_worker_wait” on page 157. In addition, this parameter was moved to “TCP/IP Parameters Set in the /etc/system File” on page 156.

**ip_soft_rings_cnt (Solaris 10 11/06 Release)**

Zone configuration information was added in the Solaris 10 8/07 release. For more information, see “ip_soft_rings_cnt” on page 142.

**ip_squeue_write (Solaris 10 Releases)**

This parameter was incorrectly documented in the Solaris 10 release. It has been removed.

**tcp_local_dack_interval (Solaris 10 Releases)**

The range of this parameter was incorrectly documented in previous Solaris releases. The correct range is 10 milliseconds to 1 minute.

**[tcp,sctp,udp]_smallest_anon_port and [tcp,sctp,udp]_largest_anon_port**

These parameters are newly documented in the Solaris 10 8/11 release.

- “sctp_smallest_anon_port” on page 173
- “sctp_largest_anon_port” on page 173
- “tcp_smallest_anon_port” on page 155
Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)

The following section describes parameters that are obsolete or have been removed from more recent Oracle Solaris releases.

rstchown

This parameter is obsolete starting in the Oracle Solaris 10 8/11 release.

Description
Indicates whether the POSIX semantics for the chown system call are in effect. POSIX semantics are as follows:

- A process cannot change the owner of a file, unless it is running with UID 0.
- A process cannot change the group ownership of a file to a group in which it is not currently a member, unless it is running as UID 0.

For more information, see chown(2).

Data Type
Signed integer

Default
1, indicating that POSIX semantics are used

Range
0 = POSIX semantics not in force or 1 = POSIX semantics used

Units
Toggle (on/off)

Dynamic?
Yes

tcp_naglim_def (Solaris 10 Releases)
The “tcp_naglim_def” on page 155 parameter is newly documented in the Solaris 10 8/11 release.

udp_do_checksum (Solaris 10 Releases)
The “udp_do_checksum” on page 163 parameter is newly documented in the Solaris 10 8/11 release.
Validation: None
When to Change: When POSIX semantics are not wanted. Note that turning off POSIX semantics opens the potential for various security holes. Doing so also opens the possibility of a user changing ownership of a file to another user and being unable to retrieve the file without intervention from the user or the system administrator.
Commitment Level: Obsolete

System V Message Queue Parameters

**msgsys:msginfo_msgmni**
Obsolete in the Solaris 10 release.
Description: Maximum number of message queues that can be created.
Data Type: Signed integer
Default: 50
Range: 0 to MAXINT
Dynamic?: No. Loaded into msgmni field of msginfo structure.
Validation: None
When to Change: When msgget(2) calls return with an error of ENOSPC or at the recommendation of a software vendor.
Commitment Level: Unstable

**msgsys:msginfo_msgtql**
Obsolete in the Solaris 10 release.
Description: Maximum number of messages that can be created. If a msgsnd call attempts to exceed this limit, the request is deferred until a message header is available. Or, if the request has set the IPC_NOWAIT flag, the request fails with the error EAGAIN.
Data Type: Signed integer
Default: 40
Range: 0 to MAXINT
Dynamic?: No. Loaded into msgtql field of msginfo structure.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>When <code>msgsnd()</code> calls block or return with error of <code>EGAIN</code>, or at the recommendation of a software vendor.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

### `msgsys:msginfo_msgmnb`

**Obsolete in the Solaris 10 release.**

- **Description**: Maximum number of bytes that can be on any one message queue.
- **Data Type**: Unsigned long
- **Default**: 4096
- **Range**: 0 to amount of physical memory
- **Units**: Bytes
- **Dynamic?**: No. Loaded into `msgmnb` field of `msginfo` structure.
- **Validation**: None
- **When to Change**: When `msgsnd()` calls block or return with an error of `EGAIN`, or at the recommendation of a software vendor.
- **Commitment Level**: Unstable

### `msgsys:msginfo_msgssz`

**Removed in the Solaris 10 release.**

- **Description**: Specifies size of chunks system uses to manage space for message buffers.
- **Data Type**: Signed integer
- **Default**: 40
- **Range**: 0 to MAXINT
- **Dynamic?**: No. Loaded into `msgtql` field of `msginfo` structure.
- **Validation**: The space consumed by the maximum number of data structures that would be created to support the messages and queues is compared to 25% of the available kernel memory at the time the module is loaded. If the number is too big, the message queue module refuses to load and the facility is unavailable. This computation does include the space that might be consumed by the messages. This situation occurs only when the module is first loaded.
| When to Change | When the default value is not enough. Generally changed at the recommendation of software vendors. |
| Commitment Level | Obsolete |

**msgs:msginfo_msgmap**

Removed in the Solaris 10 release.

- **Description**: Number of messages the system supports.
- **Data Type**: Signed integer
- **Default**: 100
- **Range**: 0 to MAXINT
- **Dynamic?**: No
- **Validation**: The space consumed by the maximum number of data structures that would be created to support the messages and queues is compared to 25% of the available kernel memory at the time the module is loaded. If the number is too big, the message queue module refuses to load and the facility is unavailable. This computation does include the space that might be consumed by the messages. This situation occurs only when the module is first loaded.

| When to Change | When the default value is not enough. Generally changed at the recommendation of software vendors. |
| Commitment Level | Obsolete |

**msgs:msginfo_msgseg**

Removed in the Solaris 10 release.

- **Description**: Number of `msginfo_msgsz` segments the system uses as a pool for available message memory. Total memory available for messages is `msginfo_msgseg * msginfo_msgssz`.
- **Data Type**: Signed short
- **Default**: 1024
- **Range**: 0 to 32,767
- **Dynamic?**: No
- **Validation**: The space consumed by the maximum number of data structures that would be created to support the messages and queues is compared to 25% of the available kernel memory at the time the module is loaded. If
the number is too big, the message queue module refuses to load and the facility is unavailable. This computation does not include the space that might be consumed by the messages. This situation occurs only when the module is first loaded.

When to Change  When the default value is not enough. Generally changed at the recommendation of software vendors.

Commitment Level  Obsolete

**msgsys:msginfo_msgmax**

Removed in the Solaris 10 release.

Description  Maximum size of System V message.

Data Type  Unsigned long

Default  2048

Range  0 to amount of physical memory

Units  Bytes

Dynamic?  No. Loaded into msgmax field of msginfo structure.

Validation  None

When to Change  When msgsnd(2) calls return with error of EINVAL or at the recommendation of a software vendor.

Commitment Level  Unstable

**System V Semaphore Parameters**

**semsys:seminfo_semmni**

Obsolete in the Solaris 10 release.

Description  Specifies the maximum number of semaphore identifiers.

Data Type  Signed integer

Default  10

Range  1 to 65,535

Dynamic?  No
### semsys:seminfo_semmsl

Obsolete in the Solaris 10 release.

**Description**
Specifies the maximum number of System V semaphores per semaphore identifier.

**DataType**
Signed integer

**Default**
25

**Range**
1 to MAXINT

**Dynamic?**
No

**Validation**
The amount of space that could possibly be consumed by the semaphores and their supporting data structures is compared to 25 percent of the kernel memory available at the time the module is first loaded. If the memory threshold is exceeded, the module refuses to load and the semaphore facility is not available.

**When to Change**
When the default value is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when an attempt is made to create more semaphores in a set than are currently configured. The application sees a return code of EINVAL from a `semget(2)` call.

**Commitment Level**
Unstable

### semsys:seminfo_semopm

Obsolete in the Solaris 10 release.

**Validation**
Compared to SEMA_INDEX_MAX (currently 65,535) and reset to that value if larger. A warning message is written to the console, messages file, or both.

**When to Change**
When the default number of sets is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when an attempt is made to create more sets than are currently configured. Instead, the application receives a return code of ENOSPC from a `semget` call.

**Commitment Level**
Unstable
### semsys:seminfo_semmns

Removed in the Solaris 10 release.

<table>
<thead>
<tr>
<th>Description</th>
<th>Specifies the maximum number of System V semaphores on the system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Signed integer</td>
</tr>
<tr>
<td>Default</td>
<td>60</td>
</tr>
<tr>
<td>Range</td>
<td>1 to MAXINT</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>The amount of space that could possibly be consumed by the semaphores and their supporting data structures is compared to 25% of the kernel memory available at the time the module is first loaded. If the memory threshold is exceeded, the module refuses to load and the semaphore facility is not available.</td>
</tr>
<tr>
<td>When to Change</td>
<td>When the default number of semaphores is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when an attempt is made to create more semaphores in a single semop call than are currently allowed. Instead, the application receives a return code of E2BIG from a semop() call.</td>
</tr>
</tbody>
</table>

### Parameters That Are Obsolete or Have Been Removed (Oracle Solaris 10)

<table>
<thead>
<tr>
<th>Description</th>
<th>Specifies the maximum number of System V semaphore operations per semop call. This parameter refers to the number of sembufs in the sops array that is provided to the semop() system call. For more information, see semop().</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Signed integer</td>
</tr>
<tr>
<td>Default</td>
<td>10</td>
</tr>
<tr>
<td>Range</td>
<td>1 to MAXINT</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>The amount of space that could possibly be consumed by the semaphores and their supporting data structures is compared to 25 percent of the kernel memory available at the time the module is first loaded. If the memory threshold is exceeded, the module refuses to load and the semaphore facility is not available.</td>
</tr>
<tr>
<td>When to Change</td>
<td>When the default value is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when an attempt is made to perform more semaphore operations in a single semop call than are currently allowed. Instead, the application receives a return code of E2BIG from a semop() call.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>
semaphores than are currently configured. The application sees a
return code of ENOSPC from a semget(2) call.

Commitment Level Unstable

sem.sys:seminfo_semnu

Description Total number of undo structures supported by the System V semaphore system.
Data Type Signed integer
Default 30
Range 1 to MAXINT
Dynamic? No
Validation The amount of space that could possibly be consumed by the
semaphores and their supporting data structures is compared to 25% of
the kernel memory available at the time the module is first loaded. If the
memory threshold is exceeded, the module refuses to load and the
semaphore facility is not available.
When to Change When the default value is not enough. Generally changed at the
recommendation of software vendors. No error message is displayed
when an attempt is made to perform more undo operations than are
currently configured. The application sees a return value of ENOSPC
from a semop(2) call when the system runs out of undo structures.

Commitment Level Unstable

sem.sys:seminfo_semume

Description Removed in the Solaris 10 release.

Maximum number of System V semaphore undo structures that can be
used by any one process.
Data Type Signed integer
Default 10
Range 1 to MAXINT
Dynamic? No
Validation The amount of space that could possibly be consumed by the
semaphores and their supporting data structures is compared to 25% of
the kernel memory available at the time the module is first loaded. If the memory threshold is exceeded, the module refuses to load and the semaphore facility is not available.

When to Change
When the default value is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when an attempt is made to perform more undo operations than are currently configured. The application sees a return code of EINVAL from a semop(2) call.

Commitment Level Unstable

**semsys:seminfo_semvmx**

Removed in the Solaris 10 release.

Description Maximum value a semaphore can be set to.
Data Type Unsigned short
Default 32,767
Range 1 to 65,535
Dynamic? No
Validation None
When to Change When the default value is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when the maximum value is exceeded. The application sees a return code of ERANGE from a semop(2) call.

Commitment Level Unstable

**semsys:seminfo_semaem**

Removed in the Solaris 10 release.

Description Maximum value that a semaphore’s value in an undo structure can be set to.
Data Type Unsigned short
Default 16,384
Range 1 to 65,535
Dynamic? No
Validation None
When to Change  When the default value is not enough. Generally changed at the recommendation of software vendors. No error messages are displayed when an attempt is made to perform more undo operations than are currently configured. The application sees a return code of EINVAL from a `semop(2)` call.

Commitment Level  Unstable

**System V Shared Memory Parameters**

**shmsys:shminfo_shmmni**

Obsolete in the Solaris 10 release.

Description  System wide limit on number of shared memory segments that can be created.

Data Type  Signed integer

Default  100

Range  0 to MAXINT

Dynamic?  No. Loaded into `shmni` field of `shminfo` structure.

Validation  The amount of space consumed by the maximum possible number of data structures to support System V shared memory is checked against 25% of the currently available kernel memory at the time the module is loaded. If the memory consumed is too large, the attempt to load the module fails.

When to Change  When the system limits are too low. Generally changed on the recommendation of software vendors.

Commitment Level  Unstable

**shmsys:shminfo_shmmax**

Obsolete in the Solaris 10 release.

Description  Maximum size of system V shared memory segment that can be created. This parameter is an upper limit that is checked before the application sees if it actually has the physical resources to create the requested memory segment.

Attempts to create a shared memory section whose size is zero or whose size is larger than the specified value will fail with an EINVAL error.
This parameter specifies only the largest value the operating system can accept for the size of a shared memory segment. Whether the segment can be created depends entirely on the amount of swap space available on the system and, for a 32-bit process, whether there is enough space available in the process’s address space for the segment to be attached.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Unsigned long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>8,388,608</td>
</tr>
<tr>
<td>Range</td>
<td>0 - MAXUINT32 on 32-bit systems, 0 – MAXUINT64 on 64-bit systems</td>
</tr>
<tr>
<td>Units</td>
<td>Bytes</td>
</tr>
<tr>
<td>Dynamic?</td>
<td>No. Loaded into <code>shmmax</code> field of <code>shminfo</code> structure.</td>
</tr>
<tr>
<td>Validation</td>
<td>None</td>
</tr>
<tr>
<td>When to Change</td>
<td>When the default value is too low. Generally changed at the recommendation of software vendors, but unless the size of a shared memory segment needs to be constrained, setting this parameter to the maximum possible value has no side effects.</td>
</tr>
<tr>
<td>Commitment Level</td>
<td>Unstable</td>
</tr>
</tbody>
</table>
Revision History for This Manual

This section describes the revision history for this manual.

- “Current Version: Oracle Solaris 10 1/13 Release” on page 197
- “New or Changed Parameters in the Oracle Solaris Release” on page 197

Current Version: Oracle Solaris 10 1/13 Release

The current version of this manual applies to the Oracle Solaris 10 1/13 release.

New or Changed Parameters in the Oracle Solaris Release

The following sections describe new, changed, or obsolete kernel tunables.

- **Oracle Solaris 10 1/13**: Oracle Solaris ZFS tunable information is provided in Chapter 3, “Oracle Solaris ZFS Tunable Parameters.”
- **Solaris 10 8/11**: The rstchown parameter is obsolete. For more information, see “What’s New in Oracle Solaris System Tuning?” on page 17.
- **Oracle Solaris 10 8/11**: This release includes the ngroups_max parameter description. For more information, see “ngroups_max” on page 44.
- **Solaris 10 10/09**: This release includes the zfs_arc_min and zfs_arc_max parameter descriptions. For more information, see “zfs_arc_min” on page 94 and “zfs_arc_max” on page 95.
- **Solaris 10 10/09**: This release includes the ddi_msix_alloc_limit parameter that can be used to increase the number of MSI-X interrupts that a device instance can allocate. For more information, see “ddi_msix_alloc_limit” on page 61.
- **Solaris 10 10/09**: Memory locality group parameters are provided in this release. For more information about these parameters, see “Locality Group Parameters” on page 89.
New or Changed Parameters in the Oracle Solaris Release

- **Solaris 10 5/08**: The translation storage buffers parameters in the “SPARC System Specific Parameters” on page 85 section have been revised to provide better information. In this release, the following parameters have changed:
  - “default_tsb_size” on page 87 – The default text has been clarified.
  - “enable_tsb_rss_sizing” on page 88 – The default text was incorrect and has been corrected.
  - “tsb_rss_factor” on page 88 – The example section referred to percentages rather than the more appropriate parameter units. This issue has been resolved.
- **Solaris 10 8/07**: Parameter information was updated to include sun4v systems. For more information, see the following references:
  - “maxphys” on page 62
  - “tmpfs:tmpfs_maxkmem” on page 75
  - “SPARC System Specific Parameters” on page 85
## Index

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>autofs, 176</td>
<td>bufhwm, 67</td>
<td>consistent_coloring, 85</td>
<td>ddi_msix_alloc_limit parameter, 61</td>
<td>enable_tsb_rss_sizing, 88</td>
<td>fastscan, 52</td>
<td>handspreadpages, 54</td>
<td>inetinit, 177</td>
</tr>
<tr>
<td>autoup, 38</td>
<td>bufhwm_pct, 67</td>
<td>cron, 176</td>
<td>default_stksize, 33</td>
<td></td>
<td>freebehind, 73</td>
<td>hires_tick, 84</td>
<td>init, 177</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default_tcb_size, 87</td>
<td></td>
<td>fs, 177</td>
<td></td>
<td>ip_addrs_per_if, 140</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>desfree, 47</td>
<td></td>
<td>fsflush, 37</td>
<td></td>
<td>ip_forward_src_routed, 139</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dhcpagent, 176</td>
<td></td>
<td>ftp, 177</td>
<td></td>
<td>ip_icmp_err_burst, 138</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dnlc_dir_enable, 66</td>
<td></td>
<td></td>
<td></td>
<td>ip_icmp_err_interval, 138</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dnlc_dir_max_size, 67</td>
<td></td>
<td></td>
<td></td>
<td>ip_icmp_return_data_bytes, 144</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dnlc_dir_min_size, 66</td>
<td></td>
<td></td>
<td></td>
<td>ip_ire_pathmtu_interval, 143</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>doiflush, 40</td>
<td></td>
<td></td>
<td></td>
<td>ip_multidata_outbound, 141</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dopageflush, 39, 183</td>
<td></td>
<td></td>
<td></td>
<td>ip_pmtu_min, 143</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ip_policy_mask, 163</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ip_respond_to_echo_broadcast, 139</td>
</tr>
</tbody>
</table>
Index

ip_send_redirects, 139
ip_soft_rings_cnt, 142
ip_squeue_fanout, 141
ip_squeue_worker_wait, 157
ip_strict_dst_multihoming, 140
ip6_forward_src_routed, 139
ip6_icmp_return_data_bytes, 144
ip6_respond_to_echo_multicast, 139
ip6_send_redirects, 139
ip6_strict_dst_multihoming, 140
ipcl_conn_hash_size, 156
ipsec, 177
K
kbd, 177
keyserv, 178
kmem_flags, 58
L
lgrp_mem_pset_aware, 90
logevent_max_q_sz, 35
login, 178
lotsfree, 46
lpg_alloc_prefer, 89
lu, 178
lwp_default_stksize, 34
M
max_nprocs, 43, 182
maxpgio, 56, 183
maxphys, 62, 183
maxpid, 42
maxuprc, 44
maxusers, 41
md_mirror:md_resync_bufsz, 91
md:mirrored_root_flag, 92
min_percent_cpu, 54
minfree, 48
moddebug, 60
mpathd, 178
msgsys:msginfo_msgmax, 190
msgsys:msginfo_msgmb, 188
msgsys:msginfo_msgmbi, 187
msgsys:msginfo_msgseg, 189
msgsys:msginfo_mgszsz, 188
msgsys:msginfo_msgtql, 187

N
ncsize, 64
ndd, 138
ndquot, 69
nfs_max_threads, 115
nfs:nacache, 128
nfs:nfs_allow_preepoch_time, 107
nfs:nfs_async_clusters, 125
nfs:nfs_async_timeout, 127
nfs:nfs_cots_timeo, 108
nfs:nfs_disable_rddir_cache, 122
nfs:nfs_do_symlink_cache, 110
nfs:nfs_dynamic, 111
nfs:nfs_lookup_neg_cache, 112
nfs:nfs_nra, 117
nfs:nfs_shrinkreaddir, 120
nfs:nfs_write_error_interval, 121
nfs:nfs_write_error_to_cons_only, 122
nfs:nfs3_async_clusters, 126
nfs:nfs3_bsize, 123
nfs:nfs3_cots_timeo, 108
nfs:nfs3_do_symlink_cache, 110
nfs:nfs3_dynamic, 112
nfs:nfs3_jukebox_delay, 129
nfs:nfs3_lookup_neg_cache, 113
nfs:nfs3_max_threads, 115
nfs:nfs3_max_transfer_size, 129
nfs:nfs3_max_transfer_size_clts, 131
nfs:nfs3_max_transfer_size_cots, 132
nfs:nfs3_nra, 117, 184
nfs:nfs3_pathconf_disable_cache, 106
nfs:nfs3_shrinkreaddir, 121
nfs:nfs4_async_clusters, 127
nfs:nfs4_bsize, 124
nfs:nfs4_cots_timeo, 109
<table>
<thead>
<tr>
<th>System Call</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfs:nfs4_do_symlink_cache</td>
<td>111</td>
</tr>
<tr>
<td>nfs:nfs4_lookup_neg_cache</td>
<td>114</td>
</tr>
<tr>
<td>nfs:nfs4_max_threads</td>
<td>116</td>
</tr>
<tr>
<td>nfs:nfs4_max_transfer_size</td>
<td>130</td>
</tr>
<tr>
<td>nfs:nfs4_nra</td>
<td>118</td>
</tr>
<tr>
<td>nfs:nfs4_pathconf_disable_cache</td>
<td>106</td>
</tr>
<tr>
<td>nfs:nrnnode</td>
<td>119</td>
</tr>
<tr>
<td>nfslogd</td>
<td>179</td>
</tr>
<tr>
<td>ngrups_max</td>
<td>44,182</td>
</tr>
<tr>
<td>noexec_user_stack</td>
<td>36,181</td>
</tr>
<tr>
<td>nss</td>
<td>179</td>
</tr>
<tr>
<td>nstrpush</td>
<td>79</td>
</tr>
</tbody>
</table>

**O**

Oracle database tuning, ZFS file systems, 100

**P**

pageout_reserve, 50
pages_before_pager, 55
pages_pp_maximum, 51
passwd, 179
physmem, 32
pidmax, 42
power, 179
pr_segp_disable, 82
primarycache, ZFS file system property, 100
pt.cnt, 77
pt.max_ptx, 78
pt.pctofmem, 77

**R**

rechoose_interval, 83
recordsize, ZFS file system property, 99
reserved_procs, 42
rlim_fd_cur, 63
rlim_fd_max, 63
routeadm, 22
rpc.nisd, 179
rpcmod:clnt_max_conns, 133
rpcmod:clnt_max_dupreqs, 136
rpcmod:maxdupreqs, 135
rpcmod:svc_default_stksize, 134
rpcmod:svc_idle_timeout, 134
rstchown, 65,186

**S**

sctp_addip_enabled, 172
sctp_cookie_life, 171
sctp_cwnd_max, 165
sctp_deferred_ack_interval, 167
sctp_heartbeat_interval, 166
sctp_ignore_path_mtu, 167
sctp_initial_mtu, 167
sctp_initial_out_streams, 171
sctp_initial_ssthresh, 168
sctp_ipv4_ttl, 166
sctp_ipv6_hoplimit, 169
sctp_largest_anon_port, 173
sctp_max_mbuf, 169
sctp_max_in_streams, 171
sctp_max_init_retr, 164
sctp_maxburst, 172
sctp_new_secret_interval, 166
sctp_pp_max_retr, 165
sctp_prsctp_enabled, 172
sctp_recv_hiwat, 169
sctp_rto_max, 170
sctp_rto_min, 170
sctp_shutack_wait_bound, 171
sctp_smallest_anon_port, 173
sctp_xmit_hiwat, 168
sctp_xmit_lowat, 168
secondarycache, ZFS file system property, 100
segspt_minfree, 82
semsys:seminfo_semaem, 194
semsys:seminfo_semmni, 190
semsys:seminfo_semmns, 192
semsys:seminfo_semnum, 193
semsys:seminfo_semmsl, 191
semsys:seminfo_sempm, 192
semsys:seminfo_semume, 193
Index

semsys:seminfo_semvmx, 194
shmsys:shminfo_shmmax, 195
shmsys:shminfo_shmmni, 195
slowscan, 53
smallfile, 73
strmsgsz, 79, 80
su, 179
sun4u, 184
sun4v, 85, 184
swapfs_minfree, 57
swapfs_reserve, 57
sys-suspend, 179
syslog, 179

tcp_conn_req_max_q, 152
tcp_conn_req_max_q0, 152
tcp_conn_req_min, 153
tcp_cwnd_max, 148
tcp_defered_ack_interval, 144
tcp_defered_acks_max, 145
tcp_ecn_permitted, 151
tcp_socket_abort_interval, 158
tcp_socket_keepalive_interval, 158
tcp_socket_largest_anon_port, 156
tcp_socket_local_dack_interval, 145, 185
tcp_socket_local_dacks_max, 146
tcp_socket_max_buf, 148
tcp_mdt_max_pbefs, 154
tcp_natlim_def, 155
tcp_recv_hiwat, 147
tcp_recv_hiwat_minmss, 160
tcp_recv_src_routes, 150
tcp_rexmit_interval_extra, 160
tcp_rexmit_interval_initial, 158
tcp_rexmit_interval_max, 159
tcp_rexmit_interval_min, 159
tcp_rst_sent_rate, 154
tcp_rst_sent_rate_enabled, 153
tcp_sack_permitted, 149
tcp_slow_start_after_idle, 149
tcp_slow_start_initial, 149
tcp_smallest_anon_port, 155
tcp_time_wait_interval, 150
tcp_tstamp_always, 147
tcp_tstamp_if_wscale, 160
tcp_wscale_always, 146
tcp_xmit_hiwat, 147
throttlefree, 49
timer_max, 85
tmpfs_maxkmem, 75
tmpfs_minfree, 75
tmpfs:tmpfs_maxkmem, 183
tsb_alloc_hiwater, 86
tsb_rss_factor, 88
tune_t_fsflushr, 38
tune_t_minarmem, 52

U
udp_do_checksum, 163
udp_largest_anon_port, 162
udp_max_buf, 163
udp_recv_hiwat, 161
udp_smallest_anon_port, 162
udp_xmit_hiwat, 161
ufs_delete_hiwat, 74
ufs_HW, 72
ufs_LW, 72
ufs_ninode, 70
ufs:ufs_WRITES, 71
utmpd, 180

Y
yppasswdd, 180

Z
zfs_arc_max, 95
zfs_arc_min, 94
ZFS file system property
primarycache, 100
recordsize, 99
ZFS file system property (Continued)
  secondarycache, 100
ZFS file systems, tuning for an Oracle database, 100
  zfs_mdcomp_disable, 99
  zfs_nocacheflush, 98
  zfs_prefetch_disable, 96