

# Dynamic System V Semaphore Tunables



Executive Summary .....	2
Introduction .....	2
Interfaces .....	2
Tunables .....	3
System V Semaphores Dynamic Tunables .....	3
Performance Benefits .....	4
Deployment Scenario .....	4
Scenario 1 .....	4
Scenario 2 .....	5
Installing Dynamic System V Semaphore Tunables .....	7
Updating the HP-UX Operating System .....	7
Updating the Man Pages .....	7
Cautions on the Use and Uninstallation of the DynSysVSem Bundle.....	7
Summary .....	7
For more information.....	7

## Executive Summary

The Dynamic System V Semaphore Tunables product bundle (`DynSysVSem`, revision B.11.31.01) enables the system administrator to modify the system V semaphore tunable values dynamically (with a few exceptions). A system reboot is not required for the modified values to take effect. The system administrator can invoke the `ktune` command to change the tunable values.

Without this product bundle, only the `SEMMSL` tunable is dynamic and the remaining system V semaphore tunables are static, and any change in their values requires a reboot of the system.

This product bundle also provides better system performance due to improved multi processor scaling algorithms with respect to allocation and data structure management of semaphore objects.

## Introduction

Third-party applications, databases in particular, use semaphores (commonly referred to as System V IPC) extensively available in the HP-UX 11i v1 and higher operating systems. The installation guides for third party products often recommend modifying the tunable values associated with the semaphores.

A semaphore is a construct that is used to control access to a set of resources. It maintains a count that determines the number of available resources. Resources are allocated by decrementing the count and released by incrementing the count. All updates to the semaphore are done atomically through system calls. When all resources are consumed, the next process that attempts to allocate a resource goes to sleep until a resource is freed, at which point the process is awakened and granted access.

## Interfaces

The following interfaces are available to manage system V semaphores:

<code>semget(2)</code>	Allocates a set of semaphores. You can allocate more than one semaphore at a time and operations can be conducted on these semaphores. A successful call to <code>semget(2)</code> returns a semaphore identifier.
<code>semctl(2)</code>	Operates on a set of semaphores.
<code>semop(2)</code>	Performs control operations on the semaphore set.

For more information on these interfaces, see the `semget(2)`, `semctl(2)`, and `srefer(2)` to the man pages for these calls.

# Tunables

Table 1 lists the tunables related to system V semaphores.

Table 1: Tunables Related to System V Semaphones

<b>Tunable</b>	<b>Description</b>
<code>semaem</code>	Specifies the maximum adjustment value of any System V IPC semaphore at the time of process exit.
<code>semnmi</code>	Specifies the number of System V IPC systemwide semaphore identifiers.
<code>semnms</code>	Specifies the number of System V IPC systemwide semaphores .
<code>semnmu</code>	Specifies the maximum number of processes that can have undo operations pending.
<code>semmsl</code>	Specifies the maximum number of System V IPC semaphores per identifier.
<code>semume</code>	Specifies the maximum number of System V IPC undo entries per process.
<code>semvmx</code>	Specifies the maximum value of any single System V IPC semaphore.
<code>sysv_hash_locks</code>	Specifies the System V IPC hashed spinlock pool size.

The `semaem` and `semvmx` tunables represent the limits associated with the values in the semaphores. These tunables do not affect the semaphore memory consumption. The `semnmi`, `semnms`, `semnmu`, `semmsl` and `semume` tunables are related to memory utilization. The `sysv_hash_locks` tunable specifies the size of the hashed spinlock (kernel data structures used for synchronization) pool (number of available spinlocks).

## System V Semaphores Dynamic Tunables

Until HP-UX 11i v3 Update 2, all the semaphore tunables controlling the behavior and memory allocation of system V semaphores were static in nature and any change in their values required a system reboot. This was due to restrictions imposed by the earlier design of system V semaphores. The semaphore descriptor tables for the semaphore sets and the undo tables were statically sized; therefore, the tables were allocated at the time of system boot. New semaphore descriptors were allocated from the semaphore table using the `semget(2)` system call. The `semget(2)` system call returns a semaphore ID from the semaphore ID table, which is used to access the defined set of semaphores. In the earlier design, the fixed size of the ID table imposed a limit on the maximum number of IDs that could be returned by the `semget(2)` call. Additionally, the fixed size of the undo tables limited the number of `SEM_UNDO` operations (`semop(2)` is called with an option for `SEM_UNDO`) for a process. Semaphore undo entries were allocated from the semaphore undo table whenever a process performed a `SEM_UNDO` operation using the `semop(2)` system call.

With the release of Dynamic System V Semaphore Tunables product bundle in HP-UX 11iv3 Update 3, the internal design of system V semaphores has changed and the previously mentioned tables are no longer statically allocated at boot time. The new design offers the following advantages:

- Better manageability through reduction in the need for unscheduled downtime.

- Dynamically increase and decrease the value of the `semnmi`, `semnms`, and `semmsl` tunables.

- Dynamically increase the value of the `semaem` and `semvmx` tunables but decreasing their value requires a system reboot.

Unlimited number of semaphore undo operations. As a result, unnecessary adjustment of the `semmnu` and `semume` tunables is not required when this product is installed and configured.

Improved internal locking obviates the need to adjust the `sysv_hash_locks` tunable value when this product is installed and configured.

Better system performance due to improved multi processor scaling and efficient algorithms.

Table 2 summarizes the tunable behavior and their minimum and maximum limits.

Table 2: Tunable Behavior

<b>Tunable</b>	<b>Behavior</b>	<b>Minimum Value*</b>	<b>Default Value*</b>	<b>Maximum Value*</b>
<code>semaem</code>	Dynamic (Decrease requires a reboot)	0	16384	32767
<code>semni</code>	Dynamic	2	2048	32767
<code>semns</code>	Dynamic	2	4096	335534080
<code>semmnu</code>	Ignored	1	256	<code>nproc** - 4</code>
<code>semmsl</code>	Dynamic	1	2048	10240
<code>semume</code>	Ignored	1	100	<code>semns</code>
<code>semvmx</code>	Dynamic (Decrease requires a reboot)	1	32767	65535
<code>sysv_hash_locks</code>	Ignored	64	128	32768

\* New minimum, default, maximum values are same the old ones.

\*\* `nproc` = Maximum number of processes on the system.

## Performance Benefits

The new design provides significantly better performance in the following scenarios:

- Faster search for a semaphore identifier using a key, passed to the `semget(2)` system call.

- Faster allocation of new semaphore identifiers.

- Less contention on a semaphore set during `semop(2)` (without the `SEM_UNDO` flag) involving multiple threads/processes with single or multiple semaphores in the set.

- Less contention on a semaphore set during `semop(2)` (with the `SEM_UNDO` flag) involving multiple threads/processes with multiple semaphores in the set.

## Deployment Scenario

This section discusses scenarios that describe how dynamic system V semaphore tunables are useful in a business context.

### Scenario 1

Consider an application that exceeds the limit on the number of semaphore sets available in the system and any subsequent `semget(2)` call results in the `ENOSPC` as error. A semaphore identifier must be created, but the system-imposed limit on the maximum number of allowed semaphore

identifiers system wide would be exceeded. Therefore, to maintain the application running, the system administrator would want to increase the value of `semmni` or `semmns` tunable(s).

Without the Dynamic System V Semaphore Tunables product bundle installed, changing the values of `semmni` or `semmns` would require a reboot. Example 1 lists the output of the `kctune` command

#### Example 1: The `kctune` Command Output Without the DynSysVSem product Bundle Installed

```
# kctune semmni=4096 semmns=5144
NOTE:   The configuration being loaded contains the following change(s)
        that cannot be applied immediately and which will be held for
        the next boot:
        -- The tunable semmns cannot be changed in a dynamic fashion.
        -- The tunable semmni cannot be changed in a dynamic fashion.
WARNING: The automatic 'backup' configuration currently contains the
        configuration that was in use before the last reboot of this
        system.
==> Do you wish to update it to contain the current configuration
        before making the requested change? n
NOTE:   The backup will not be updated.
WARNING: The previous change of 'semmni' to '2048', which was being held
        until next boot, has been discarded.
        * The requested changes have been saved, and will take effect at
        next boot.
```

Tunable		Value	Expression
semmni	(now)	2048	Default
	(next boot)	4096	4096
semmns	(now)	4096	Default
	(next boot)	5144	5144

With the Dynamic System V Semaphore Tunables product bundle installed, the system administrator can change the values of the `semmni` or `semmns` tunable without performing a reboot for the application to work.

#### Example 2. The `kctune` Command output with the DynSysVSem Product Bundle Installed

```
# kctune semmni=4096 semmns=5144
==> Update the automatic 'backup' configuration first? n
        * Future operations will ask whether to update the backup.
        * The requested changes have been applied to the currently
        running configuration.
```

Tunable		Value	Expression	Changes
semmni	(before)	2048	Default	Immed
	(now)	4096	4096	
semmns	(before)	4096	Default	Immed
	(now)	5144	5144	

## Scenario 2

Consider an application that exceeds the system limit on the maximum number of processes performing undo operations and any subsequent `semop(2)` system call with `SEM_UNDO` flag with fail with `ENOSPC` as error. The limit on the number of individual processes requesting an `SEM_UNDO` would be exceeded or results in `EINVAL` as error. The number of individual semaphores for which the calling process requests a `SEM_UNDO` would exceed the limit. Therefore, to maintain the application running, the system administrator would want to increase the value of the `semmnu` or `semume` tunable.

Without the Dynamic System V Semaphore Tunables product bundle installed, changing the values of `semmnu` or `semume` will require a reboot.

### Example 3: : The `kctune` Command Output Without the `DynSysVSem` product Bundle Installed

```
# kctune semmnu=512 semume=200
NOTE:   The configuration being loaded contains the following change(s)
        that cannot be applied immediately and which will be held for
        the next boot:
        -- The tunable semume cannot be changed in a dynamic fashion.
        -- The tunable semmnu cannot be changed in a dynamic fashion.
WARNING: The automatic 'backup' configuration currently contains the
        configuration that was in use before the last reboot of this
        system.
==> Do you wish to update it to contain the current configuration
        before making the requested change? n
NOTE:   The backup will not be updated.
        * The requested changes have been saved, and will take effect at
        next boot.
```

Tunable		Value	Expression
semmnu	(now)	256	Default
	(next boot)	512	512
semume	(now)	100	Default
	(next boot)	200	200

With the Dynamic System V Semaphore Tunables product bundle installed, the system administrator does not need to change the value of `semmni` or `semmns` for the application to work. The application will not fail as the system imposed limits are removed with the new design. Changing the value of these tunables does not have an on the system.

### Example 4: The `kctune` Command Output with The `DynSysVSem` Product Bundle Installed.

```
# kctune semmnu=512 semume=200
==> Update the automatic 'backup' configuration first? n
        * Future operations will ask whether to update the backup.
WARNING: With DynSysVSem product bundle installed, semmnu tunable is
        ignored and any change in its value has no effect on the
        system.
WARNING: With DynSysVSem product bundle installed, semume tunable is
        ignored and any change in its value has no effect on the
        system.
        * The requested changes have been applied to the currently
        running configuration.
```

Tunable		Value	Expression	Changes
semmnu	(before)	256	Default	Immed
	(now)	512	512	
semume	(before)	100	Default	Immed
	(now)	200	200	

# Installing Dynamic System V Semaphore Tunables

The software to enable the dynamic system V semaphore tunables enhancement is available in Update 3 to the HP-UX 11i v3 operating system.

## Updating the HP-UX Operating System

All the files required to update the HP-UX core operating system for dynamic system V semaphore tunables are installed as a consequence of installing the `DynSysVSem` product bundle. A reboot following the installation is required.

## Updating the Man Pages

Updates to the man pages of the system V semaphore tunables can be obtained by installing the PHKL\_37849 patch. The patch provides important additional information about the behavior of the system V semaphore tunables when the `DynSysVSem` product bundle is installed. This patch is not included in the `DynSysVSem` product bundle and is not automatically installed.

## Cautions on the Use and Uninstallation of the DynSysVSem Bundle

Following are cautions to be noted while using and uninstalling the `DynSysVSem` product bundle:

Before uninstalling this product, the tunable settings must be reviewed. For instance, with the product installed, the `semnu` and `semume` tunables have no effect on boot-time kernel memory consumption. However, if they are set to high values and the `DynSysVSem` product bundle is removed, the system may fail to boot or perform as desired. For more information, refer the man pages of system V semaphore tunables by installing the PHKL\_37849 patch. You can also check the HP IT Resource Center for the latest information:  
<http://itrc.hp.com>

Care must be taken when increasing the value of the `semaem` and `semvmx` tunables because only increase in the value of these tunables is dynamic and any subsequent decrease in the value requires a reboot of the system. For more information, refer the man pages of `semaem(5)` and `semvmx(5)` by installing the PHKL\_37849 patch. You can also check the HP IT Resource Center for the latest information.

## Summary

The Dynamic System V Semaphore enhancement to HP-UX 11i v3 Update 3 changes the tunable behavior from static to dynamic and also provides performance improvements. This enhancement enables system administrators to manage system V semaphore tunables better and reduces the need for unscheduled reboots.

## For more information

For additional information, see the HP-UX man pages `semget(2)`, `semop(2)`, `semctl(2)`, `sema(5)`, `semaem(5)`, `semnns(5)`, `semmni(5)`, `semmsl(5)`, `semume(5)`, `semvmx(5)`, `sysv_hash_locks(5)`.

To help us improve our documents, please provide feedback at [www.hp.com/solutions/feedback](http://www.hp.com/solutions/feedback)