NAME

sad – STREAMS Administrative Driver

SYNOPSIS

#include <sys/types.h>
#include <sys/conf.h>
#include <sys/sad.h>
#include <sys/stropts.h>

int ioctl (int *fildes*, **int** *command*, **int** *arg***);**

DESCRIPTION

The STREAMS Administrative Driver provides an interface for applications to perform administrative operations on STREAMS modules and drivers. The interface is provided through **ioctl**(2) commands. Privileged operations may access the **sad** driver using **/dev/sad/admin**. Unprivileged operations may access the **sad** driver using **/dev/sad/user**.

fildes is an open file descriptor that refers to the **sad** driver. *command* determines the control function to be performed as described below. *arg* represents additional information that is needed by this command. The type of *arg* depends upon the command, but it is generally an integer or a pointer to a *command*-specific data structure.

COMMAND FUNCTIONS

The autopush facility (see **autopush**(1M)) allows one to configure a list of modules to be automatically pushed on a stream when a driver is first opened. Autopush is controlled by the following commands:

SAD_SAP Allows the administrator to configure the given device's autopush information. *arg* points to a **strapush** structure, which contains the following members:

```
uint   sap_cmd;
long   sap_major;
long   sap_minor;
long   sap_lastminor;
long   sap_npush;
uint   sap_list [MAXAPUSH] [FMNAMESZ + 1];
```

The **sap_cmd** field indicates the type of configuration being done. It may take on one of the following values:

SAP_ONE Configure one minor device of a driver.

SAP_RANGE Configure a range of minor devices of a driver.

SAP_ALL Configure all minor devices of a driver.

SAP_CLEAR Undo configuration information for a driver.

The **sap_major** field is the major device number of the device to be configured. The **sap_minor** field is the minor device number of the device to be configured. The **sap_lastminor** field is used only with the **SAP_RANGE** command, which configures a range of minor devices between **sap_minor** and **sap_lastminor**, inclusive. The minor fields have no meaning for the **SAP_ALL** command. The **sap_npush** field indicates the number of modules to be automatically pushed when the device is opened. It must be less than or equal to **MAXAPUSH** , defined in **sad.h**. It must also be less than or equal to **NSTRPUSH**, the maximum number of modules that can be pushed on a stream, defined in the kernel master file. The field **sap_list** is an array of NULL-terminated module names to be pushed in the order in which they appear in the list.

When using the **SAP_CLEAR** command, the user sets only **sap_major** and **sap_minor**. This will undo the configuration information for any of the other commands. If a previous entry was configured as **SAP_ALL**, **sap_minor** should be set to zero. If a previous entry was configured as **SAP_RANGE** , **sap_minor** should be set to the lowest minor device number in the range configured.

On failure, **errno** is set to the following value:

EFAULT *arg* points outside the allocated address space.

EINVAL The major device number is invalid, the number of modules is invalid, or the list of module names is invalid.

ENOSTR The major device number does not represent a STREAMS driver.

EEXIST The major-minor device pair is already configured.

ERANGE The command is **SAP_RANGE** and **sap_lastminor** is not greater than **sap_minor**, or the command is **SAP_CLEAR** and **sap_minor** is not equal to the first minor in the range.

ENODEV The command is **SAP_CLEAR** and the device is not configured for autopush.

ENOSR An internal autopush data structure cannot be allocated.

SAD_GAP Allows any user to query the **sad** driver to get the autopush configuration information for a given device. *arg* points to a **strapush** structure as described in the previous command.

The user should set the **sap_major** and **sap_minor** fields of the **strapush** structure to the major and minor device numbers, respectively, of the device in question. On return, the **strapush** structure will be filled in with the entire information used to configure the device. Unused entries in the module list will be zero-filled.

On failure, **errno** is set to one of the following values:

EFAULT *arg* points outside the allocated address space.

EINVAL The major device number is invalid.

ENOSTR The major device number does not represent a STREAMS driver.

ENODEV The device is not configured for autopush.

SAD_VML Allows any user to validate a list of modules (that is, to see if they are installed on the system). *arg* is a pointer to a **str_list** structure with the following members:

int	**sl_nmods;**
struct str_mlist	***sl_modlist;**

The **str_mlist** structure has the following member:

char	**l_name[FMNAMESZ+1];**

sl_nmods indicates the number of entries the user has allocated in the array and **sl_modlist** points to the array of module names. The return value is 0 if the list is valid, 1 if the list contains an invalid module name, or −1 on failure. On failure, **errno** is set to one of the following values:

EFAULT *arg* points outside the allocated address space.

EINVAL The **sl_nmods** field of the **str_list** structure is less than or equal to zero.

SEE ALSO

intro(2), **ioctl**(2), **open**(2)

STREAMS Programming Guide

DIAGNOSTICS

Unless otherwise specified, the return value from **ioctl** is 0 upon success and −1 upon failure with **errno** set as indicated.