

# Driver Functions

## DSP Functions

### Start the DSP

kern\_return\_t      **snddriver\_dsp\_boot**(port\_t *commandPort*, int \**bootImage*, int *imageSize*, int *priority*)

kern\_return\_t      **snddriver\_dsp\_reset**(port\_t *commandPort*, int *priority*)

### Transfer data to and from the DSP via DMA

kern\_return\_t      **snddriver\_dsp\_dma\_write**(port\_t *commandPort*, int *elementCount*, int *dataFormat*, pointer\_t *data*)

kern\_return\_t      **snddriver\_dsp\_dma\_read**(port\_t *commandPort*, int *elementCount*, int *dataFormat*, pointer\_t *data*)

### Enqueue a DSP command

kern\_return\_t      **snddriver\_dsp\_host\_cmd**(port\_t *commandPort*, u\_int *hostCommand*, u\_int *priority*)

### Set the sound driver's protocol vis-a-vis the DSP

kern\_return\_t s    **nddriver\_dsp\_protocol**(port\_t *devicePort*, port\_t *ownerPort*, int *protocol*)

### Set the DSP host flags

kern\_return\_t      **snddriver\_dsp\_set\_flags**(port\_t *commandPort*, u\_int *flagMask*, u\_int *flagValue*, u\_int *priority*)

### Transfer data to and from the DSP

kern\_return\_t      **snddriver\_dsp\_write**(port\_t *commandPort*, void \**buffer*, int *elementCount*, int *elementSize*, int *priority*)

kern\_return\_t      **snddriver\_dsp\_read**(port\_t *commandPort*, void \**buffer*, int *elementCount*, int *elementSize*, int *priority*)

kern\_return\_t      **snddriver\_dsp\_read\_messages**(port\_t *commandPort*, void \**buffer*, int *elementCount*, int *elementSize*, int *priority*)

kern\_return\_t      **snddriver\_dsp\_read\_data**(port\_t *commandPort*, void \*\**buffer*, int *elementCount*, int *elementSize*, int *priority*)

### Request a DSP host interface register condition

kern\_return\_t      **snddriver\_dspcmd\_req\_condition**(port\_t *commandPort*, u\_int *registerMask*, u\_int *conditionFlags*, int *priority*, port\_t *replyPort*)

### Request the contents of the DSP-reply buffers

|               |  |
|---------------|--|
| kern_return_t | <b>snddriver_dspcmd_req_msg</b> (port_t <i>commandPort</i> , port_t <i>replyPort</i> ) |
| kern_return_t | <b>snddriver_dspcmd_req_err</b> (port_t <i>commandPort</i> , port_t <i>replyPort</i> ) |

### Get the DSP command port

|               |  |
|---------------|--|
| kern_return_t | <b>snddriver_get_dsp_cmd_port</b> (port_t <i>devicePort</i> , port_t <i>ownerPort</i> , port_t<br>* <i>commandPort</i> ) |
|---------------|--|

## Driver Setup and Access

### Acquire ownership of sound resources

|               |   |
|---------------|---|
| kern_return_t | <b>snddriver_set_dsp_owner_port</b> (port_t <i>devicePort</i> , port_t <i>ownerPort</i> , port_t<br>* <i>negotiationPort</i> )    |
| kern_return_t | <b>snddriver_set_sndin_owner_port</b> (port_t <i>devicePort</i> , port_t <i>ownerPort</i> , port_t<br>* <i>negotiationPort</i> )  |
| kern_return_t | <b>snddriver_set_sndout_owner_port</b> (port_t <i>devicePort</i> , port_t <i>ownerPort</i> , port_t<br>* <i>negotiationPort</i> ) |

### Reallocate the sound driver device port

|               |   |
|---------------|---|
| kern_return_t | <b>snddriver_new_device_port</b> (port_t <i>devicePort</i> , port_t <i>superuserPort</i> , port_t<br>* <i>newDevicePort</i> ) |
|---------------|---|

### Respond to asynchronous sound driver messages

|               |  |
|---------------|--|
| kern_return_t | <b>snddriver_reply_handler</b> (msg_header_t * <i>reply</i> , snddriver_handlers_t * <i>handlers</i> ) |
|---------------|--|

### Set and get sound playback attributes

|               |   |
|---------------|---|
| kern_return_t | <b>snddriver_set_device_parms</b> (port_t <i>devicePort</i> , boolean_t <i>speakerOn</i> , boolean_t<br>* <i>filterOn</i> , boolean_t <i>zerofill</i> )     |
| kern_return_t | <b>snddriver_get_device_parms</b> (port_t <i>devicePort</i> , boolean_t * <i>speakerOn</i> , boolean_t<br>* <i>filterOn</i> , boolean_t * <i>zerofill</i> ) |
| kern_return_t | <b>snddriver_set_volume</b> (port_t <i>devicePort</i> , int <i>leftVolume</i> , int <i>rightVolume</i> )  |
| kern_return_t | <b>snddriver_get_volume</b> (port_t <i>devicePort</i> , int * <i>leftVolume</i> , int * <i>rightVolume</i> )  |
| kern_return_t | <b>snddriver_set_ramp</b> (port_t <i>devicePort</i> , int <i>rampOn</i> )   |

## Stream Setup and Access

### Configure stream transfer buffers

|               |  |
|---------------|--|
| kern_return_t | <b>snddriver_set_sndout_bufcount</b> (port_t <i>devicePort</i> , port_t <i>sndoutPort</i> , int <i>count</i> ) |
| kern_return_t | <b>snddriver_set_sndout_bufsize</b> (port_t <i>devicePort</i> , port_t <i>sndoutPort</i> , int <i>size</i> )   |
| kern_return_t | <b>snddriver_stream_ndma</b> (port_t <i>streamPort</i> , int <i>regionTag</i> , int <i>count</i> )             |

### Control and query a stream

|               |   |
|---------------|---|
| kern_return_t | <b>snddriver_stream_control</b> (port_t <i>streamPort</i> , int <i>regionTag</i> , int <i>control</i> ) |
| kern_return_t | <b>snddriver_stream_nsamples</b> (port_t <i>streamPort</i> , int * <i>byteCount</i> )                   |

**Configure a sound stream**

kern\_return\_t      **snddriver\_stream\_setup**(port\_t *devicePort*, port\_t *ownerPort*, int *dataPath*, int *sampleCount*, int *sampleSize*, int *lowWater*, int *highWater*, int *\*protocol*, port\_t *\*streamPort*)

**Send data to and retrieve data fom a stream**

kern\_return\_t      **snddriver\_stream\_start\_writing**(port\_t *streamPort*, void *\*data*, int *sampleCount*, int *regionTag*, boolean\_t *preempt*, boolean\_t *deallocateWhenDone*, boolean\_t *msgStarted*, boolean\_t *msgCompleted*, boolean\_t *msgAborted*, boolean\_t *msgPaused*, boolean\_t *msgResumed*, boolean\_t *msgUnderrun*, port\_t *replyPort*)

kern\_return\_t      **snddriver\_stream\_start\_reading**(port\_t *streamPort*, char *\*filename*, int *sampleCount*, int *regionTag*, boolean\_t *msgStarted*, boolean\_t *msgCompleted*, boolean\_t *msgAborted*, boolean\_t *msgPaused*, boolean\_t *msgResumed*, boolean\_t *msgOverrun*, port\_t *replyPort*)