Defined Types

MIDIAlarmReplyFunction

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

typedef void (***MIDIAlarmReplyFunction**)(port_t *replyPort*, int *requestedTime*, int *actualTime*);

DESCRIPTION This function is used to handle requests for alarm registered by the **MIDIRequestAlarm()** function. *replyPort* represents the port passed by the **MIDIRequestAlarm()** function. *requestedTime* represents the time passed by the **MIDIRequestAlarm()** function. *actualTime* represents the actual time at which the alarm is sent.

MIDIDataReplyFunction

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

typedef void (***MIDIDataReplyFunctio**n)(port_t *replyPort*, short *unit*, MIDIRawEvent **events*, unsigned int *count*);

DESCRIPTION This function is used to handle requests for data registered by the **MIDIRequestData()** function. *replyPort* represents the port passed by the **MIDIRequestData()** function. *events* represents an array of MIDIRawEvent data. *count* represents the number of elements in *events*.

MIDIExceptionReplyFunction

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

typedef void (***MIDIExceptionReplyFunction**)(port_t *replyPort*, int *exception*);

DESCRIPTION This function is used to handle requests for exceptions registered by the **MIDIRequestExceptions()** function. *replyPort* represents the port passed by the **MIDIRequestExceptions()** function. *exception* represents the exception sent by the driver.

MIDIQueueReplyFunction

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

typedef void (***MIDIQueueReplyFunction**)(port_t *replyPort*, short *unit*);

DESCRIPTION This function is used to handle requests for queue information registered by the **MIDIRequestQueueNotification()** function. *replyPort* represents the port passed by the

MIDIRequestQueueNotification() function. *unit* represents the serial port with which the queue is associated.

MIDIRawEvent

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

```
typedef struct {
    int   time;
    unsigned char   byte;
} MIDIRawEvent;
```

DESCRIPTION *time* is the timestamp associated with the MIDI data. *byte* is the actual MIDI data.

MIDIReplyFunctions

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

```
typedef struct _MIDIReplyFunctions {
    MIDIDataReplyFunction   dataReply;
    MIDIAlarmReplyFunction   alarmReply;
    MIDIExceptionReplyFunction   exceptionReply;
    MIDIQueueReplyFunction   queueReply;
} MIDIReplyFunctions;
```

DESCRIPTION This structure is used as an argument to the **MIDIAwaitReply()** and **MIDIHandleReply()** functions to allow an application to handle replies to requests to the MIDI driver.

Symbolic Constants

Clock Modes

DECLARED IN mididriver/midi_driver.h

SYNOPSIS

```
MIDI_CLOCK_MODE_INTERNAL
MIDI_CLOCK_MODE_MTC_SYNC
```

Controller Definitions

DECLARED IN mididriver/midi_spec.h

SYNOPSIS

```
MIDI_EXTERNALEFFECTSDEPTH
MIDI_TREMELODEPTH
MIDI_CHORUSDEPTH
MIDI_DETUNEDEPTH
MIDI_PHASERDEPTH
```

(from original 1.0 MIDI spec)

MIDI_EFFECTS1
MIDI_EFFECTS2
MIDI_EFFECTS3
MIDI_EFFECTS4
MIDI_EFFECTS5
MIDI_DATAINCREMENT
MIDI_DATADECREMENT
(From June 1990 spec)

Error Codes

DECLARED IN mididriver/midi_driver.h

SYNOPSIS MIDI_ERROR_BUSY
MIDI_ERROR_NOT_OWNER
MIDI_ERROR_QUEUE_FULL
MIDI_ERROR_BAD_MODE
MIDI_ERROR_UNIT_UNAVAILABLE
MIDI_ERROR_ILLEGAL_OPERATION
MIDI_ERROR_UNKNOWN_ERROR

Event Count

DECLARED IN mididriver/midi_driver.h

SYNOPSIS MIDI_MAX_EVENT 100

Event Size

DECLARED IN mididriver/midi_driver.h

SYNOPSIS MIDI_MAX_MSG_SIZE 1024

Exception Codes

DECLARED IN mididriver/midi_driver.h

SYNOPSIS MIDI_EXCEPTION_MTC_STOPPED
MIDI_EXCEPTION_MTC_STARTED_FORWARD
MIDI_EXCEPTION_MTC_STARTED_REVERSE

General MIDI Constants

DECLARED IN mididriver/midi_spec.h

SYNOPSIS MIDI_RESETCONTROLLERS
MIDI_LOCALCONTROL

MIDI_ALLNOTESOFF
MIDI_OMNIOFF
MIDI_OMNION
MIDI_MONO
MIDI_POLY
MIDI_NOTEOFF
MIDI_NOTEON
MIDI_POLYPRES
MIDI_CONTROL
MIDI_PROGRAM
MIDI_CHANPRES
MIDI_PITCH
MIDI_CHANMODE
MIDI_CONTROL
MIDI_SYSTEM
MIDI_SYSEXCL
MIDI_TIMECODEQUARTER
MIDI_SONGPOS
MIDI_SONGSEL
MIDI_TUNEREQ
MIDI_EOX
MIDI_CLOCK
MIDI_START
MIDI_CONTINUE
MIDI_STOP
MIDI_ACTIVE
MIDI_RESET
MIDI_MAXDATA
MIDI_MAXCHAN
MIDI_NUMCHANS
MIDI_NUMKEYS
MIDI_ZEROBEND
MIDI_DEFAULTVELOCITY

DESCRIPTION These constants represent various MIDI-specified messages.

Ignores

DECLARED IN mididriver/midi_driver.h

SYNOPSIS
MIDI_IGNORE_CLOCK
MIDI_IGNORE_START
MIDI_IGNORE_CONTINUE
MIDI_IGNORE_STOP
MIDI_IGNORE_ACTIVE
MIDI_IGNORE_RESET
MIDI_IGNORE_REAL_TIME

DESCRIPTION Used with the **MIDISetSystemIgnores()** function.

Least Significant Bit for Controller Numbers

DECLARED IN mididriver/midi_spec.h

SYNOPSIS MIDI_MODWHEELLSB

MIDI_BREATHLSB
MIDI_FOOTLSB
MIDI_PORTAMENTOTIMELSB
MIDI_DATAENTRYLSB
MIDI_MAINVOLUMELSB
MIDI_BALANCELSB
MIDI_PANLSB
MIDI_EXPRESSIONLSB

Masks for MIDI Status Bytes

DECLARED IN mididriver/midi_spec.h

SYNOPSIS MIDI_STATUSBIT

MIDI_STATUSMASK
MIDI_SYSRTBIT

Miscellaneous

DECLARED IN mididriver/midi_driver.h

SYNOPSIS MIDI_NO_TIMEOUT

MIDI Controller Numbers

DECLARED IN mididriver/midi_spec.h

SYNOPSIS MIDI_MODWHEEL

MIDI_BREATH
MIDI_FOOT
MIDI_PORTAMENTOTIME
MIDI_DATAENTRY
MIDI_MAINVOLUME
MIDI_BALANCE
MIDI_PAN
MIDI_EXPRESSION
MIDI_EFFECTCONTROL1
MIDI_EFFECTCONTROL2
MIDI_DAMPER
MIDI_PORTAMENTO
MIDI_SOSTENUTO
MIDI_SOFTPEDAL
MIDI_HOLD2

Port Constants

DECLARED IN mididriver/midi_driver.h

SYNOPSIS MIDI_PORT_A_UNIT

MIDI_PORT_B_UNIT

DESCRIPTION Used to identify the port claimed for the application.