MiscTime

Inherits From:	Object
Conforms To:	NXTransport
Declared In:	misckit/MiscTime.h

Class Description

This documentation is still incomplete, and the class itself is only partway implemented. Sorry.

MiscTime is used to store a time, accurate from years down to microseconds. You can add and subtract MiscTimes from each other and specify how to generate a string value for the time. It can calculate the day of the week for a given date, as well. If used to measure time in a relative sense, then the day of week functions are disabled since they don't make sense semantically. A MiscTime object may easily be initialized to an arbitrary time or set to the current time on the system clock. It may also be sent bycopy over a distributed objects link and archived to a file.

You can use the format: method to display a MiscTime as a string, using a variety of different *printf*-like formatting codes. For instance, [aMiscTime format:"%A %B %d %H:%M"] returns a string in the style of "Thursday September 29 02:28". See *man strftime* for the full details of the format conversions supported by the format: method.

The day of the week will be recalculated as needed. If it is calculated and then the date is changed MiscTime will do the right thing and recalculate it. Note that if you override any of the 'set', 'add', or 'subtract' methods you should probably call the super's method, otherwise the day of week functioning may not work.

There was some confusion in earlier implementations about the yearÐsometimes it was number of years since 1900, or 1970, or the actual year. Now, use the actual year for everything to work right. If it is 1995, set the year as 1995, and *not* 95!

Instance Variables

BOOL **isRelative**;
long **microsecond**;
long **second**;
long **minute**;
long **hour**;
long **dow**;
long **day**;
long **month**;
long **year**;

isRelative	Whether the MiscTime object is an absolute or relative time.

microsecond	The microsecond (0 ± 999,999).
second	The second (0 ± 59).
minute	The minute (0 ± 59).
hour	The hour (0 ± 23).
dow	The day of the week (-1 ±€6). dow is -1 if the day of the week is unknown; for example, if it has not yet been calculated.
day	The day of the month (0 ± 27/28/29/30).
month	The month (0 ± 11).
year	The year.

Method Types

Information on days, months, years	+ (int)daysInMonth:(int)aMonth ofYear:(int)aYear + (int)daysUpToMonth:(int)aMonth ofYear:(int)aYear + (int)isLeapYear:(int)aYear
Initializing and freeing a MiscTime	± init ± initWithCurrentTime
Formatting a MiscTime as a string	± (const char *)format:(const char *)format

Changing a MiscTime	± setDay: ± setDayOfWeek: ± setHour: ± setMicrosecond: ± setMinute: ± setMonth: ± setRelative: ± setSecond: ± setYear:
Manipulating a MiscTime	± addDays: ± addHours: ± addMicroseconds: ± addMinutes: ± addMonths: ± addSeconds: ± addTime: ± addYears: ± calcDayOfWeek ± subtractDays: ± subtractHours: ± subtractMicroseconds: ± subtractMinutes: ± subtractMonths: ± subtractSeconds: ± subtractTime: ± subtractYears:

Querying attributes	± day
	± dayOfWeek
	± doubleValue
	± floatValue
	± hour
	± intValue
	± isRelative
	± microsecond
	± minute
	± month
	± second
	± stringValue
	± year
Archiving	± read:
	± write:

Class Methods

daysInMonth:ofYear:

+ (int)**daysInMonth**:(int)*aMonth*
  ofYear:(int)*aYear*

Returns the number of days in the month *aMonth* for the year *aYear*.

daysUpToMonth:ofYear:
  + (int)**daysUpToMonth**:(int)*aMonth*
        ofYear:(int)*aYear*

Returns the number of days before *aMonth* that must pass for the year *aYear.* (eg 151 days pass before the first day of June unless it's a leapyear in which case it is 152 days)

isLeapYear:aYear
  + (int)**isLeapYear**:(int)*aYear*

Returns YES if *aYear* is a leap year, NO otherwise.

Instance Methods

addDays:
  - **addDays:**(long)*t*

Adds t days to the MiscTime.

See also: - **myReference**

addHours:

- **addHours:**(long)*t*

Adds t hours to the MiscTime.

See also: - **myReference**

addMicroseconds:
- **addMicroseconds:**(long)*t*

Adds t microseconds to the MiscTime.

See also: - **myReference**

addMinutes:
- **addMinutes:**(long)*t*

Adds t minutes to the MiscTime.

See also: - **myReference**

addMonths:
- **addMonths:**(long)*t*

Adds t months to the MiscTime.

See also: - **myReference**

addSeconds:
- **addSeconds:**(long)*t*

Adds t seconds to the MiscTime.

See also: - **myReference**

addTime:
- **addTime:***aTime*

Adds the MiscTime values of aTime to the instance. If aTime is not a MiscTime this method simply returns nil. Note that this method does not test to make sure aTime is a relative time.

See also: - **myReference**

addWeeks:
- **addWeeks:**(long)*t*

Adds t weeks to the MiscTime.

See also: - **myReference**

addYears:
- **addYears:**(long)*t*

Adds t years to the MiscTime.

See also: - **myReference**

copyTimeFrom:

- **copyTimeFrom:***aTime*

Copies the time from aTime.

See also: - **myReference**

dayOfWeek

- (int)**dayOfWeek**

Returns the day of the MiscTime. Note that the first day of the month is the 0th. eg, the first of January would be represented by a MiscTime as having month=0, day=0.

See also: - **myReference**

doubleValue

- (double)**doubleValue**

Returns the 0.0. **BUG:** not useful.

See also: - **myReference**

first.
    - **calcDayOfWeek**   **// set the year, month, and day first.**

Method description here.

See also: - **myReference**

floatValue
    - (float)**floatValue**

Returns the 0.0.   **BUG:** not useful.

See also: - **myReference**

format:
    - (const char *)**format:**(const char *)*fmtString*

Returns a string int the Unix function *strftime* format for the MiscTime.

See also: - **myReference**

hour
    - (int)**hour**

Returns the hour of the MiscTime.

See also: - **myReference**

initWithCurrentTime

- **initWithCurrentTime**

Initializes the new MiscTime instance. All date values are set to represent the current date, which is found using the Unix functions *gettimeofday* and *localtime*. The year is set to be the complete year, however, not the year relative to 1900.

See also: - initWithTime_t

initWithTime_t:

- **initWithTime_t:** (time_t)*aTime*

Initializes the new MiscTime instance. All date values are set to represent the date passed in (time_t) aTime. aTime is parsed using the Unix function and *localtime*. The year is set to be the complete year, however, not the year relative to 1900.

See also: - **myReference**

intValue

- (int)**intValue**

Returns the number of seconds since 1970. **BUG:** does not include leapyears (or seconds).

See also: - **myReference**

isAfter:
- (BOOL)**isAfter:***aTime*

Returns YES if   the instance of the MiscTime is after aTime (ie at a later date).

See also: - **myReference**

isEqual:
- (BOOL)**isEqual:***aTime*

Returns YES if   the instance of the MiscTime at the same aTime (ie the same date).

See also: - **myReference**

isRelative
- (BOOL)**isRelative**

Returns YES if the date has been set to be relative, NO otherwise.

See also: - **setRelative**

microsecond
- (int)**microsecond**

Returns the microsecond of the MiscTime.

See also: - **myReference**

minute

- (int)**minute**

Returns the minute of the MiscTime.

See also: - **myReference**

month

- (int)**month**

Returns the month of the MiscTime. Note that January is counted as the 0th month, February is the 1st.

See also: - **myReference**

read:

- **read:**(NXTypedStream *)*stream*

Reads the MiscTime from the typed stream *stream*.

See also: - **myReference**

second

- (int)**second**

Returns the second of the MiscTime.

See also: - **myReference**

setDay:

- **setDay:**(int)*t*

Sets the day to be t.

See also: - **myReference**

setDayOfWeek:

- **setDayOfWeek:**(int)*t*

Sets the day of the week to be t. **Note**: If you set the day, month, or year after you set the day of week the day of week will be recalculated next time it is requested.

See also: - **myReference**

setHour:

- **setHour:**(int)*t*

Sets the hour to be t.

See also: - **myReference**

setMicrosecond:

- **setMicrosecond:**(int)*t*

Sets the microsecond to be t.

See also: - **myReference**

setMinute:

- **setMinute:**(int)*t*

Sets the minute to be t.

See also: - **myReference**

setMonth:

- **setMonth:**(int)*t*

Sets the month to be t. Sets the day to be t.

See also: - **myReference**

setRelative:

- **setRelative:**(BOOL)*t*

Sets wether the date is relative.   Note that being a relative date does not affect the functioning of the MiscTime in any way except that it will reply YES if it is sent an isRelative message.

See also: - **isRelative**

setSecond:

- **setSecond:**(int)*t*

Sets the second to be t.

See also: - **myReference**

setYear:

- **setYear:**(int)*t*

Sets the year to be t.   Sets the day to be t.

See also: - **myReference**

stringValue

- (const char *)**stringValue**

Returns a string of the Unix function *strftime* format "%H : %M : %S" which is the (24 hour) hour : minute : second of the MiscTime.

See also: - **format**

subtractDays:
- **subtractDays:**(long)*t*

Subtracts t days from the MiscTime.

See also: - **myReference**

subtractHours:
- **subtractHours:**(long)*t*

Subtracts t hours from the MiscTime.

See also: - **myReference**

subtractMicroseconds:
- **subtractMicroseconds:**(long)*t*

Subtracts t microseconds from the MiscTime.

See also: - **myReference**

subtractMinutes:

- **subtractMinutes:**(long)*t*

Subtracts t minutes from the MiscTime.

See also: - **myReference**

subtractMonths:

- **subtractMonths:**(long)*t*

Subtracts t months from the MiscTime.

See also: - **myReference**

subtractSeconds:

- **subtractSeconds:**(long)*t*

Subtracts t seconds from the MiscTime.

See also: - **myReference**

subtractTime:

- **subtractTime:***aTime*

Subtracts the MiscTime values of aTime from the instance. If aTime is not a MiscTime this method simply returns nil. Note that this method does not test to make sure aTime is a relative time.

See also: - **myReference**

subtractWeeks:
- **subtractWeeks:**(long)*t*

Subtracts t weeks from the MiscTime.

See also: - **myReference**

subtractYears:
- **subtractYears:**(long)*t*

Subtracts t years from the MiscTime.

See also: - **myReference**

usecs.
- (int)_**nintValue // private method, does intValue w/o usecs.**

Returns the number of microseconds of the MiscTime instance.

See also: - **myReference**

write:
- **write:**(NXTypedStream *)*stream*

Writes the MiscTime to the typed stream *stream*.

See also: - **myReference**

year

- (int)**year**

Returns the year of the MiscTime.

See also: - **myReference**