

# SayTime - The Verbal Clock ver 1.7

## Overview

- \* **Tells** you what time it is via your nifty sound card.
- \* Randomly plays potentially obnoxious wave files after telling you the time.
- \* Let's you choose you're own font color (background color is window background color).
- \* Automatically adjusts the clock frame to best fit the time and date display.
- \* Can be *INVISIBLE!* Wooooooo!
- \* A desert topping *AND* a floor wax! Impress your friends!
- \* Version 1.6 tested by at least 20 Windows Testers for 1 year!

## Menu Items

The following menu items can be accessed by double clicking the left mouse button or single clicking the right mouse button on the clock's client area.

Options...

Font...

About...

Recheck User Waves!

Recheck Wave Device!

Exit

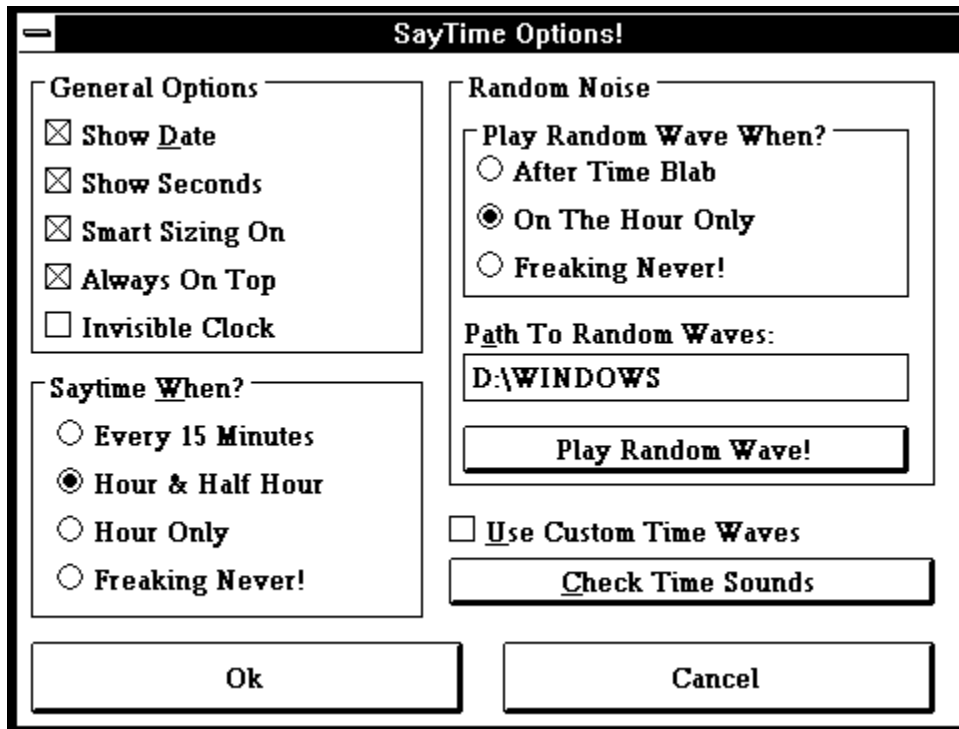
## New For Version 1.7:

- \* *Fixed user defined wave files to REALLY work.*
- \* *Old SAYTIME.INI file has now become SAYTIME.DAT. You can use the old one, just rename it.*
- \* *Wonder What's Been Fixed/Added? Search for "Revision" in the help system...*

## Who Made This Infernal Thing?

Send bugs, suggestions, wishes or chocolate to MikeLi. This thing is my baby, it's completely free. No obligations or whining about "*Please send me \$2.38 if you find it useful, etc.*" I see it as my donation to the bit bucket. I still haven't seen anything I like better...

# Options Dialog



The Options Dialog contains the controls for many of the user defined features of SayTime including how the time is displayed and how the time is spoken.

## Time Display - General Options

<u>Show Date</u>	Causes the date to be displayed when checked.
<u>Show Seconds</u>	Causes the seconds to be displayed when checked
<u>Smart Sizing On</u>	If checked, the clock window will automatically be resized so that the largest possible font will just fit with minimal excess. The upper left portion of the clock window will remain fixed.
<u>Always On Top</u>	If checked, the clock will be a topmost window. Note: some screen savers like After Dark are not topmost windows (silly!). In these cases the clock will still be topmost (and look stupid, but it's not my fault!!).
<u>Invisible Clock</u>	Causes the clock window to be invisible. The clock won't appear anywhere on the desktop or the task list. NOTE: If you want the clock to reappear, run another instance. This will make the original clock once again visible. For more information, see <a href="#">Invisible Clock</a> .

## Time Display - Verbal Options

In order for the following options to have an affect, you must have a wave output device installed (sound card driver or PC SPEAKER DRIVER). SayTime will warn you on startup if you don't have a suitable waveform output device installed.

<u>SayTime When</u>	This set of radio buttons controls when SayTime says the time. If you don't like being told the time, you can disable speech output by selecting the "Freaking Never!" radio button.
<u>Play Random Wave When</u>	This set of radio buttons controls when SayTime plays a random wave file.

For more information, see [Playing Random Waves](#).

Path To Random Waves

This edit control specifies the path showing where to look for the nest 'o random wave files. For more information, see [Playing Random Waves](#).

Use Custom Time Waves

When checked, this instructs SayTime to use wave files provided by YOU to say the time. For more information, see [Using Custom Time Waves](#).

# Invisible Clock

When you've got a clock that talks, you might not want to have it taking up space on your desktop because *it can tell you what time it is*. So, by checking the **Invisible Clock** check box in the Options Dialog you can make your clock invisible. So invisible that it doesn't even show up in the task list (Oooo!).

## **Making An Invisible Clock Visible Again**

If the last time SayTime was running, it was invisible, it's going to be invisible next time too. It's going to stay that way until you try to run a second instance. When you run the second instance, SayTime will become visible again.

## Second Instance

Running a second instance can be accomplished by running SayTime (or any other application) when there's already a copy of SayTime running. If there's no previous copy of the application running, you'd be running the first instance.

# Playing Random Waves

Random waves are played after the time is spoken. Random waves are played **asynchronously** so you don't have to worry about having to sit and wait while Saytime plays 7 minutes and 55 seconds of Stairway To Heaven (yes, that's how long it really is - [trust me](#)).

The following controls of the [Options Dialog](#) govern the use of random wave files:

<a href="#">Play Random Wave When</a>	This set of radio buttons controls when SayTime plays a random wave file. If not set to "Freaking Never!", SayTime will play a wave file from the directory specified in the <b>Path To Random Waves</b> edit control described below.
<a href="#">Path To Random Waves</a>	This edit control defaults to the Windows directory. If you try to set this to a bogus directory that doesn't contain any files matching *.wav, then SayTime will tell you so and allow you to correct your mistake. It only tells you once per Option Dialog session, so if you don't fix it - you won't get to hear random waves. Presently you're limited to the first 150 wave files in the dir - I'll fix that next time around.
<a href="#">Play Random Wave!</a>	This push button will play a random wave file contained in the directory specified by <b>Path To Random Waves</b> above.

In order to play random waves, you must:

Have a properly installed waveform output device (aka a sound card and driver).

Make sure the **Play Random Wave When?** radio buttons (of the [Options Dialog](#)) are not set to "*Freaking Never!*".

Have a valid path to \*.wav files in the **Path To Random Waves** edit control.

# Using Custom Time Waves

By default, SayTime will use wave files built into the SayTime EXE file when saying the time. If you want to get cute and don't like my stinking voice, I have graciously provided SayTime with the ability to use other time wave files provided by you the mortal user.

**NOTE:** *You don't have to redo all of the time wave files - if you don't provide all the clock numbers, SayTime will use the default / built in wave file.*

## Rolling Your Own Time Wave Files

Your best bet is to get a microphone, plug it into your nifty sound card and use Sound Recorder to record your yackings (that's how I did it). You might want to make one mondo wave file that contains all the numbers and then chop it up into the individual wave files described below.

## What To Call Your Homegrown Wave Files

SayTime will use wave files following the format: <number>.wav, where <number> is the digits 1 through 12 (for the hours) and 0, 15, 30, 45 for the minutes. Note that 0.wav generally corresponds to *o'clock*.

**NOTE:** *Once you've got your army of custom wave files, put them in the directory where SayTime.exe lives or it won't find them.*

## Using And Testing Your Custom Wave Files

Once you've created your custom wave files and put them in the SayTime directory, check the **Use Custom Time Waves** box in the Options Dialog.

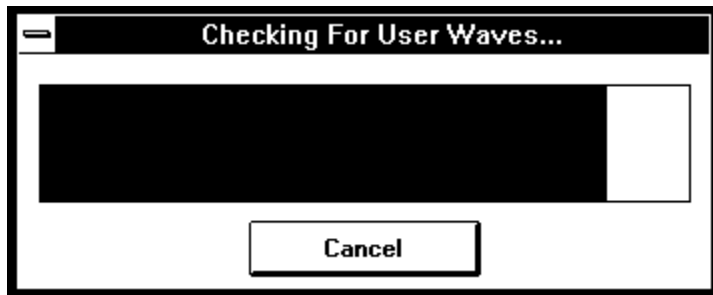
To hear how they sound you can push the **Check Time Sounds** push button. This will go through all hours and minutes using the wave files just as they would be used when saying the normal time (using default wave files for missing or invalid custom wave files). You can't cancel this action once started - but I disabled the button so you'll only have to suffer through this once.

If you add a new custom wave file (for example 1.wav where 1.wav didn't exist when SayTime was run), you need to notify SayTime of it's presence. See [Rechecking User Waves](#) for information on this.

## Rechecking User Waves

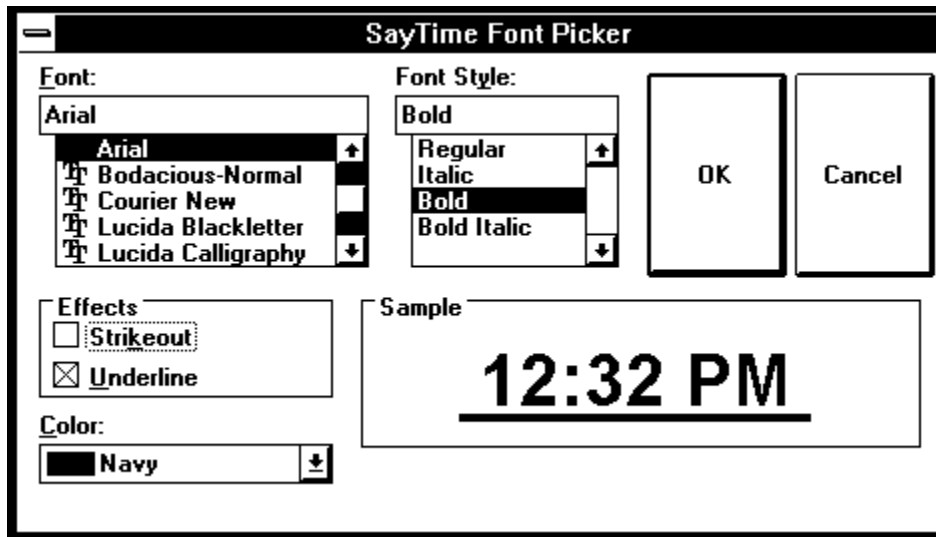
In order to be efficient, SayTime only checks what custom time wave's you're going to use when it starts or you check the box and then exit the Options Dialog. If you add a new custom wave file you can instruct SayTime to sniff around and recheck what custom wave files you've got by selecting the **Recheck UserWaves!** menu item from the main menu.

By the way, there's a neato gasgauge mechanism to show you the progress while it's checking (so I got bored...).





# Font Dialog



This is a common dialog and works just as most of the other common dialogs you've encountered.

This font dialog differs from the CLOCK.EXE dialog in that you can pick a font color and make the font strikeout and underline if you wish.

The buttons are mondo because, ahh... they were on sale.

Saves the current settings and quits SayTime (oh really??).

It's **about** this big >[ ]<.

# Changing The Time/Date Format

SayTime can show the time and date in different formats. This might be considered useful by our international friends, I'm not sure. You can change the format of the time and date via the Control Panel and the International applet.

This is also where you specify the AMPM gizmo (or lack of).

When you change the settings of the international display, SayTime will automatically react and show the time and date accordingly.

# Revision History

<b>Version</b>	<b>Date</b>	<b>Fixes And Non Heinous Improvements</b>
1.01	11/10/92	Rewrote time display functions to support proportional fonts and only repaint only the part of the time display that has changed. This dramatically reduces the repaint flicker when displaying large point sizes.
1.02	12/12/92	Saves window position when app closes (instead of only when Windows closes). Doesn't say "Back from realm of invisibility" unless it was invisible when second instance is run. Help file now uses 1bpp bitmaps - this decreased the size of the help file by about 150K!
1.50	01/31/93	Added commdlg for window color selection. People whined about this long enough, so I caved in. This required some special PatBlt crap to paint the background, boy what fun! I also put in a fudge factor of 2 pixels for those wiley italic fonts whose charWidths aren't 100% accurate. I was amazed that my CalcTimeSize() really works (you might notice only the numbers that changed are repainted!) I used the Windows Sound System to normalize the sounds ("to make them as loud as possible without adding distortion" - so QuickRecorder says...). Also fixed the AM not changing to PM problem, added a bit more sensitivity for when the system time changes. You can always grab and resize a bit to cause it to repaint if you find a bug like the AM --> PM thang until I git a chance to fix the dang thing. Waa!
1.60	03/13/93	Got rid of the nasty msg box that came up each time the clock was started if you didn't have a wave device. Saytime still checks each time, but doesn't tell you if you don't have a sound device. Chances are, you're already aware of this... Also added a menu option so you can do the sound card check manually (so you don't have to close and restart the clock if you all of sudden install a wave device). Don't know if this is useful because 99% of the time, if you install a sound device, you'll have to restart Windows for it to take affect. Whatever.
1.70	09/06/93	Time to put this baby on the BBS things seeing how I like it better than any of the other ones I've seen. The clock dude now looks for and uses SAYTIME.DAT (because I was bored). Don't worry, it will automagically rename an old SAYTIME.INI if it finds one. I fixed the userWaves (they work again - was missing a wsprintf(), minor detail). I also fixed it so it would save its position better and better detect when someone changed the resolution so that the previous position was off the screen. I moved the check for user defined waves during startup to a better spot during initialization. I stuck in GetModuleFilename() goodies so now it will look in the startup dir (whether or not that was the WORKING DIR) for the INI file (now SAYTIME.DAT), user defined number waves and the help file. Pretty nifty. I also added some code so that you can't minimize or maximize the clock (try and post it a msg, you evil dudes). Thwarting the TILE ALL and CASCADE ALL stuff from the task man is more difficult and didn't make it this round. Probably did some other minor stuff - see if you can find them all!

# Recheck Wave Device

If you don't have a sound device installed (PC SPEAKER driver or a sound card and driver capable of playing digitized sounds like WAVE files), Saytime won't talk. Each time Saytime starts, it checks to see if you've got a suitable sound device installed. If you do - sounds will be enabled. If you install the driver and want to tell Saytime about it, you can choose the "**Recheck Wave Device**" menu option and Saytime will tell you if it found something to play wave files or not.

If you can hear wave files using Sound Recorder or you have system events working, then you have a wave output device installed.

Here's the code that does the checking, so if there's a better way let me know:

```
if (waveOutGetNumDevs() <= 0)
    gwOptions |= FLG_NOWAVEDEVICE;
else
    gwOptions &= ~FLG_NOWAVEDEVICE;
```



