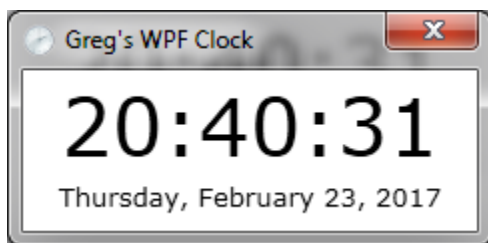


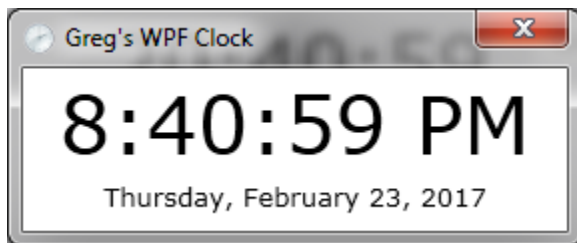
# WPF Clock App

I wrote this app in C# and WPF, beginning with a sample clock app downloaded from the Web. Since getting the original version built and running, I've made numerous enhancements to it based on using it and getting ideas for improvements. One of the first enhancements was adding a date in 3 different formats:

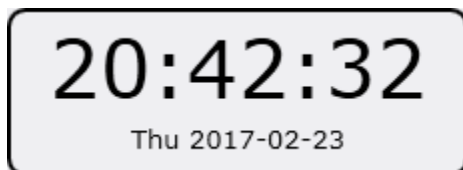
1. Long: Thursday, February 23, 2017
2. Short: Thu 2/23/2017
3. ISO 8601 (with day of week): Thu 2017-02-23
4. ISO 8601 by itself: 2017-02-23



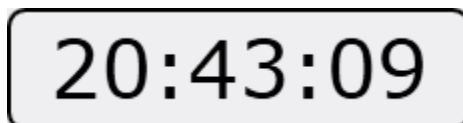
Also added 12-hour display as an option:



Then I got rid of the title bar to reduce the desktop real estate it took up:

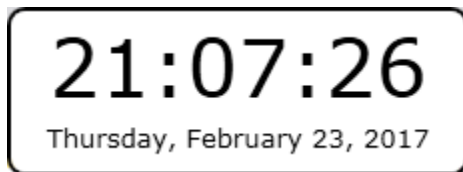


... made the date optional

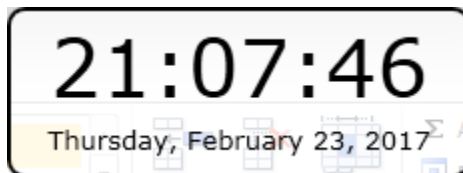


... and added variable opacity to the background

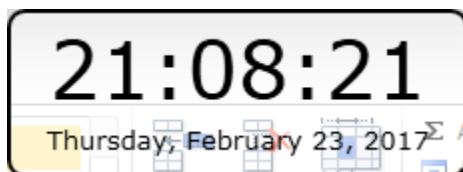
100%



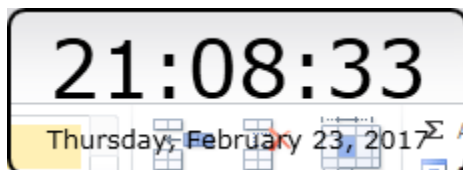
75%



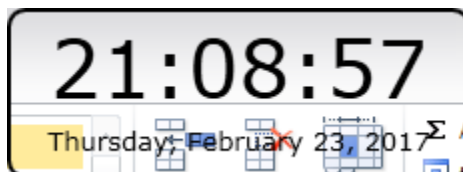
50%



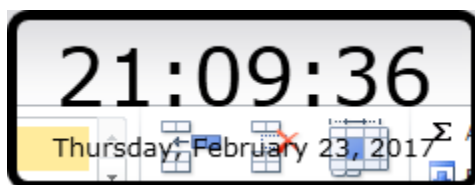
25%



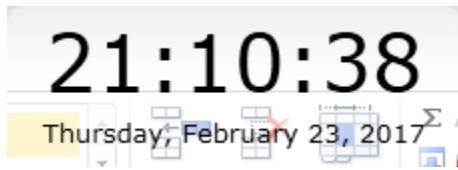
See-through



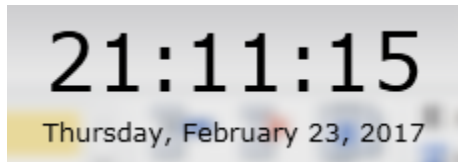
Then I added code to increase the border thickness ...



... and decrease it, all the way to 0



Then I added the Vista “glass” background



The user interface is through a context (right-click) menu. It contains all the options for adjusting the appearance to one’s preferences:

- Always on top (ctrl T), defaults to off.
- Draggable (defaults to off).
- Increase border thickness (+ key).
- Decrease border thickness (- key).
- Increase corner radius (ctrl +).
- Decrease corner radius (ctrl -).
- Display Mode to select the display modes shown above (ctrl P cycles through them all).
- Ctrl-M toggles an option to highlight the background when the mouse moves over the window.
- Ctrl-I toggles showing an icon in the taskbar (default is off).
- Ctrl-H hides the clock and creates a systray icon. The clock can be restored by either clicking on the systray icon (at which point the icon is removed from the systray since that’s its only purpose) or by launching the app again, which will reactivate the clock but not create another instance of it (unless the /m command line switch is used, described below).
- Select between 12-hour and 24-hour time formats.
- Ctrl-S toggles the option to display the seconds in the time.
- Select from all the date format options described above (ctrl D will cycle through them all).
- Copy the date, time, or both to the clipboard, either in the currently displayed format, or in the ISO 8601 format shown above.
- Ctrl C will copy the date and time in their currently displayed format to the clipboard as a string in a single line.
- Select font, font style (Normal, Italic, or Oblique), and font size. The font size for the time and date may each be set separately.
- Select foreground and background colors.

All of these settings are stored in the registry; the clock doesn’t ever write a file to the system. The settings are saved when the clock is shut down. There are menu options for saving the settings to the

registry immediately, and also for shutting down the clock and removing the registry settings (to make it easy to uninstall by closing the clock with this option and then deleting the WpfClock exe).

In addition to showing the system fonts for selection, WpfClock also has 4 embedded fonts, 2 of which are an image of the magnetic ink font used for checking account and routing numbers on checks. The other 2 are 7-segment and 14-segment fonts like those used in digital displays. The 7-segment font is only really useable for numeric characters, so when this font is selected, the date is switched to ISO 8601 without the day-of-week text, time is switched to 24-hour format, and all other date and time options are disabled. Switching to any other font will re-enable the other date and time options and restore the previously-selected time and date formats.

Another feature allows the clock to be effectively docked to the edges and corners on monitors (both on systems with single and multiple monitors), and keep them there when the border thickness is changed and when the date format or font is changed and the window is resized as a result. For systems with multiple monitors, this feature will also prevent the clock from straddling the border between monitors to keep the window on only one monitor. The clock's last position will be saved in the registry and the clock will reposition itself there when restarted. If multiple instances of the clock are run, the settings from the most recently closed (or saved) instance will over-write the other's settings.

Two versions are available: 32-bit and 64-bit. The 64-bit version is recommended over the 32-bit for 64-bit versions of Windows because its code for detecting screensaver operation is more reliable. The 32-bit version is fine for 32-bit Windows versions.

There are also a few command line switches that are available:

- `/m` enables multiple instances of the clock application. If another instance of the clock is hidden, this option will still allow that instance to be brought to the desktop. If multiple instances of the clock have been hidden, this will only reactivate one instance and the others will remain hidden. This appears to be a limitation of the Windows operating system.
- `/t` shows the title bar and hides the menu options for border thickness, corner radius, and opacity.
- `/c` enables a "click-through" context menu option. In this mode and the "mouse-over" option is disabled, mouse clicks pass through the clock window to the app behind it when the mouse is positioned in a place where the app behind the clock is visible through the clock window.
- `/t` and `/c` are mutually exclusive; `/c` overrides `/t` if both are used.
- `/?` will display a message box with an abbreviated form of the command-line switches listed above.

Comments and suggestions may be sent to me at [greg02@pobox.com](mailto:greg02@pobox.com).