

## **Status Bar Component: Overview**

The Status Bar Component can be inserted in a project to obtain one of the following features on the status bar of the main window of an MFC SDI or MDI application. Not all three features need be added at the same time. However, a particular feature cannot be added more than once. If the main frame window does not already have a status bar, the component will add one.

Status Bar Component - Specifics

Status Bar Component - Results

## Status Bar Component: Specifics

The status bar component has the following features:

- Time
- Date
- Menu Prompts

### Time

Selecting this feature adds a pane to the status bar in which the current time will be displayed. The time will be formatted based on specifications made in the **Time** dialog box while inserting the component.

### Date

Selecting this feature adds a pane to the status bar in which the current date will be displayed. The format in which the date is displayed can be set to the default system setting, or a custom setting can be selected.

### Menu Prompts

The first pane of the status bar gets resized to fit in the space left after all the other panes are positioned on the status bar. Sometimes status bars get so crowded that the menu prompts do not have enough room for display. An example is the prompt text displayed inside the first pane of the status bar (when the user navigates the menus with the mouse or arrow keys). Including this feature adds code to your application which forces the first pane of the status bar to become as wide as the main frame window every time menu prompts are displayed. It is automatically resized to its original size as soon as menu navigation is over.

### Tips

You can swap the panes for the time and date. The order of the panes in an MFC AppWizard generated application is determined by a `static UINT` array in `mainfrm.cpp`. This array, named `indicators[]`, is copied in the `CMainFrame::OnCreate()` and passed to `CMainFrame::CGStatusBarInit()`. The IDs corresponding to the two new panes are added here as follows:

```
1 BOOL CMainFrame::CGStatusBarInit(UINT *puIndicators, int nSize)
2 {
3     // Create an index for the TIME pane
4     m_nTimePaneNo = nSize++;
5     puIndicators[m_nTimePaneNo] = ID_INDICATOR_TIME;
6     // Create an index for the DATE pane
7     m_nDatePaneNo = nSize++;
8     puIndicators[m_nDatePaneNo] = ID_INDICATOR_DATE;
9
10    return m_wndStatusBar.Create(this) &&
        m_wndStatusBar.SetIndicators(puIndicators, nSize);
}
```

Swapping the positions of lines 4 and 5 with 7 and 8 above will swap the positions of the two panes.

### Why did a `SetTimer()` get added?

The status bar panes get updated via the **COMMAND\_UPDATE\_UI** mechanism. Update **\_UI**

functions get called by the **CWinThread::OnIdle()** which is called once every time the message queue becomes empty. The function of the timer is to generate **WM\_TIMER** messages so that the status bar, especially the time pane, is updated regularly.

If your application already has a timer, the following line can be taken out from `CMainFrame::OnCreate()` since a second timer is not needed:

```
m_wndStatusBar.SetTimer(1, 1000, NULL);
```

### **Status Bar Component: Results**

This component automatically adds all the code that is needed for the desired effects. However, if you wish to modify any of the three features, please refer to the log file that gets created during the component insertion. You can select the name of this log file inside the **Finish** dialog while adding the component.

