Using custom-draw in tooltips to adjust the font

devblogs.microsoft.com/oldnewthing/20060627-22

June 27, 2006

Raymond Chen

Last time, we looked at in-place tooltips. In that example, we finessed the font problem by simply setting the destination font into the tooltip control. We got away with that since we had only one tool. But if you have multiple tools with different fonts, then you can’t set a font into the tooltip control and expect it to work for every tool. That’s where custom draw comes in.

Start with the program from last time, but this time, we’ll set the font via custom-draw instead of setting it globally.

```
BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
 ...
 ~~// SetWindowFont(g_hwndTT, g_hfTT, FALSE);~~
 ...
}
LRESULT
OnTooltipCustomDraw(HWND hwnd, NMHDR *pnm)
{
 LPNMTTCUSTOMDRAW pcd = (LPNMTTCUSTOMDRAW)pnm;
 if (pcd->nmcd.dwDrawStage == CDDS_PREPAINT) {
  SelectFont(pcd->nmcd.hdc, g_hfTT);
  return CDRF_NEWFONT;
 }
 return 0;
}
LRESULT
OnNotify(HWND hwnd, int idFrom, NMHDR *pnm)
{
 if (pnm->hwndFrom == g_hwndTT) {
  switch (pnm->code) {
  case NM_CUSTOMDRAW:
   return OnTooltipCustomDraw(hwnd, pnm);
  case TTN_SHOW:
   return OnTooltipShow(hwnd, pnm);
  }
 }
 return 0;
}
```

Of course, doing this is overkill in our case where we have only one tool, so you’ll have to imagine that the tooltip is managing multiple tool regions, each with a different font. When we get the `NM_CUSTOMDRAW` notification, we respond to the `CDDS_PREPAINT` stage by changing the font and returning the `CDRF_NEWFONT` flag (which is necessary when changing the font).

Raymond Chen

Follow