

How do I programmatically reposition monitors in a multiple-monitor system?

devblogs.microsoft.com/oldnewthing/20211222-00

December 22, 2021



Raymond Chen

Say you're looking to write a tool to manipulate monitor positions. What functions should you be calling?

The function to call is `ChangeDisplaySettingsEx` with a `DEVMODE` whose `dmFields` has the `DM_POSITION` flag set. Put the desired upper left corner of the monitor in the `dmPosition.x` and `dmPosition.y` fields. You're probably also going to want to update the monitor size, which you do by setting the `DM_PELSWIDTH` and `DM_PELSHEIGHT` flags, putting the corresponding values in the `dmPelsetWidth` and `dmPelsHeight` members. To remove a monitor from the desktop, set the position to (0,0) and the width and height to zero.

The overall pattern is to use `EnumDisplayDevices` to get all the available display devices. For each one, call `EnumDisplaySettingsEx` to get the current `DEVMODE` for each monitor, using `ENUM_CURRENT_SETTINGS` to get whatever the monitor setting is right now, or `ENUM_REGISTRY_SETTINGS` to get the saved settings. (The current settings may be different from the registry settings if the display is in a temporary mode, such as when playing a fullscreen DirectX game.)

After you gather all of the `DEVMODE` s, you modify the things you want to modify, and then stage the values by calling `ChangeDisplaySettingsEx` to set the updated values with the `CDS_UPDATEREGISTRY` and `CDS_NORESET` flags.

Finally, once you're done setting all the coordinates, perform a final

```
ChangeDisplaySettingsEx(NULL, NULL, NULL, 0, NULL);
```

to apply all the settings at once.

MSDN provides [a sample function that detaches all secondary monitors](#) to give you an idea of how the whole thing fits together.

Raymond Chen

Follow

