



In the coming 10.3 release of RAD Studio, we are adding support for Microsoft Per Monitor v2 mode available in recent releases of Windows 10.

Supporting multiple monitors with different resolutions in Windows applications, including VCL ones, is often fairly complex. One of the issues faced in the past was that asking for the size of platform elements with Windows APIs like `GetSystemMetrics` used to return only information for the primary monitor. So it was difficult, for example, to have a custom scroll bar of the right size on secondary monitors with a different resolution.

Microsoft introduced new APIs in Windows 10 to simplify this work. Namely, they added a new `GetSystemMetricsForDPI` in Windows 10 Creator's Update (build 1703). The VCL library makes extensive use of this API, however you cannot just use the new call, as this will break compatibility with Windows 7 and with older versions of Windows 10. Therefore the solution required some extra indirection.

First we added a new global function in `VCL.Classes` unit, called `GetSystemMetricsForWindow`. It wraps a call to the new `GetSystemMetricsForDPI` if available or to the traditional `GetSystemMetrics` if not. The new function has an additional handle parameter, passed to the API, indicating the handle of the windows you are interested in (and indirectly the information about the monitors it is displayed on).

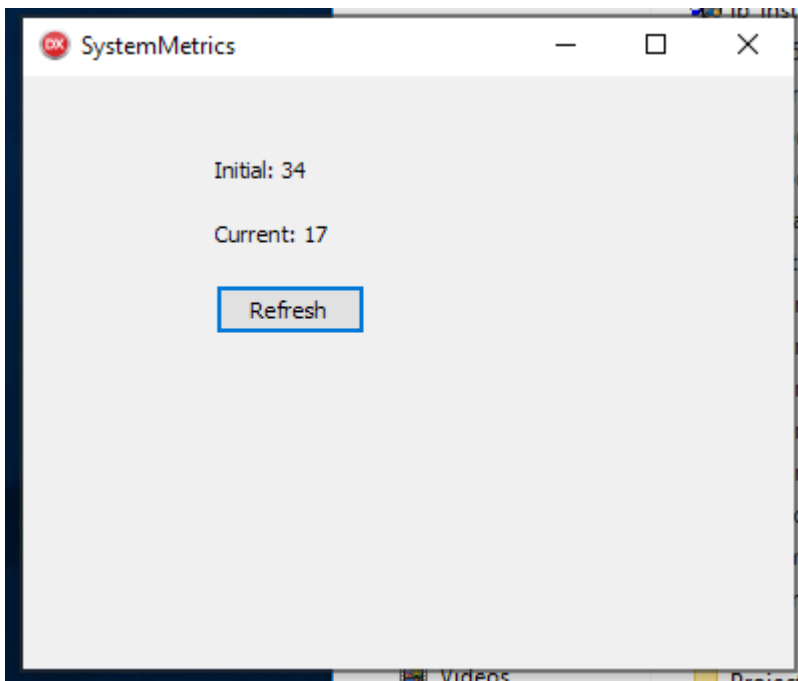
Second, to simplify the work needed to update the existing VCL code (the VCL library code itself and your custom components or application code), we've added a new method in `TControl`, called `GetSystemMetrics`. This is compatible in terms of parameters with the Windows API of the same name, so an existing call to the API in one of your components gets redirected to the method, which in turn calls `GetSystemMetricsForWindow` passing the control's parent handle as parameter. In this way, most of your code gets "migrated" with no effort. Finally, we've added a `CurrentPPI` read only property to the `TControl` class to get the DPI for the control, depending the current monitor.

As an example of the use, I've created a simple VCL application with the following code for the `OnClick` event of a button:

```
procedure TFormSystemMetrics.btnRefreshClick(Sender: TObject);
30 begin
    Label2.Caption := 'Current: ' +
        GetSystemMetrics (SM_CXVSCROLL).ToString;
end;
```

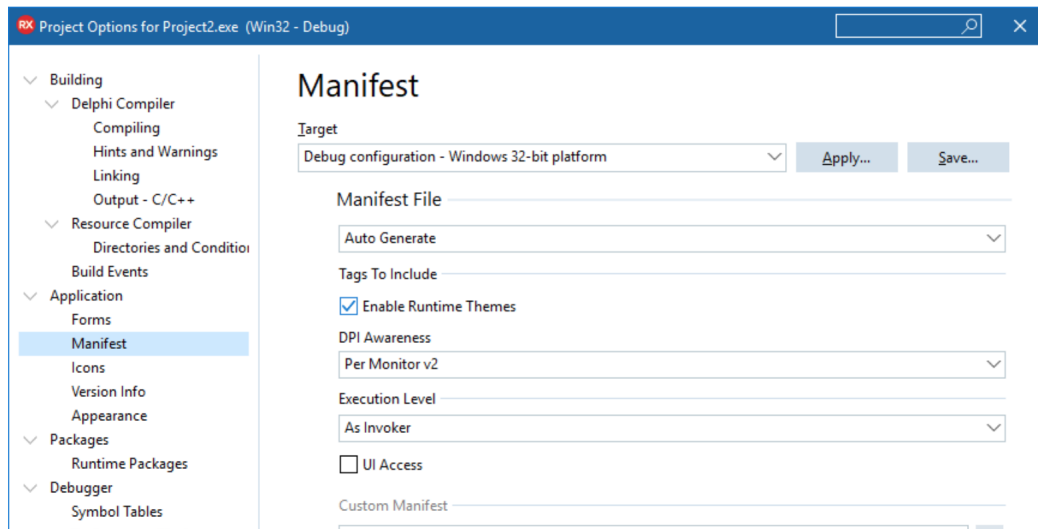
Notice this code has been correct since early versions of Delphi, involving the Windows API function

monitor specific result as it hits a different, newer Windows API function. I have the same call in the OnCreate event of the form, updating another label with the default for my main, high resolution monitor (34 pixels). After moving the form to my secondary lower resolution monitor (on which scrollbars take 17 pixels), you'll see the result of the same call changes:



Beside this core enablement, we've made significant updates to the VCL and its styling support adopting the new model, and recommend any component writer to do the same for supporting this new Microsoft DPI model. In most cases, all we had to do was leave the code as it. Some of the metrics are monitor independent or depend invariably on the primary monitor, and in that case you might want to force a call to the Windows api (by prefixing the call with the Windows unit, `Windows.GetSystemMetrics (SM_XXX)`). Also if you have any call on a data module or a global function or a class not inheriting from `TControl`, you might have to reconsider your code structure.

Finally we've made it easier to enable the feature in your applications manifest, by adding it to the Project Options in the IDE, as shows below. For a new VCL project, this is the default:



Finally there is a caveat: The Per Monitor V2 (and many of the other High-DPI features in Windows) have no support for the MDI child windows model. Microsoft has stopped addressing bugs reported for MDI, and we recommend migrating to a different multi-windows model (multiple floating windows, docked panes, tabbed windows, etc.)

0 comments 1 member is here