vbforums.com

[VB6] Manifest Creator II-VBForums

1. Apr 9th, 2017, 03:20 PM #1

Thread Starter

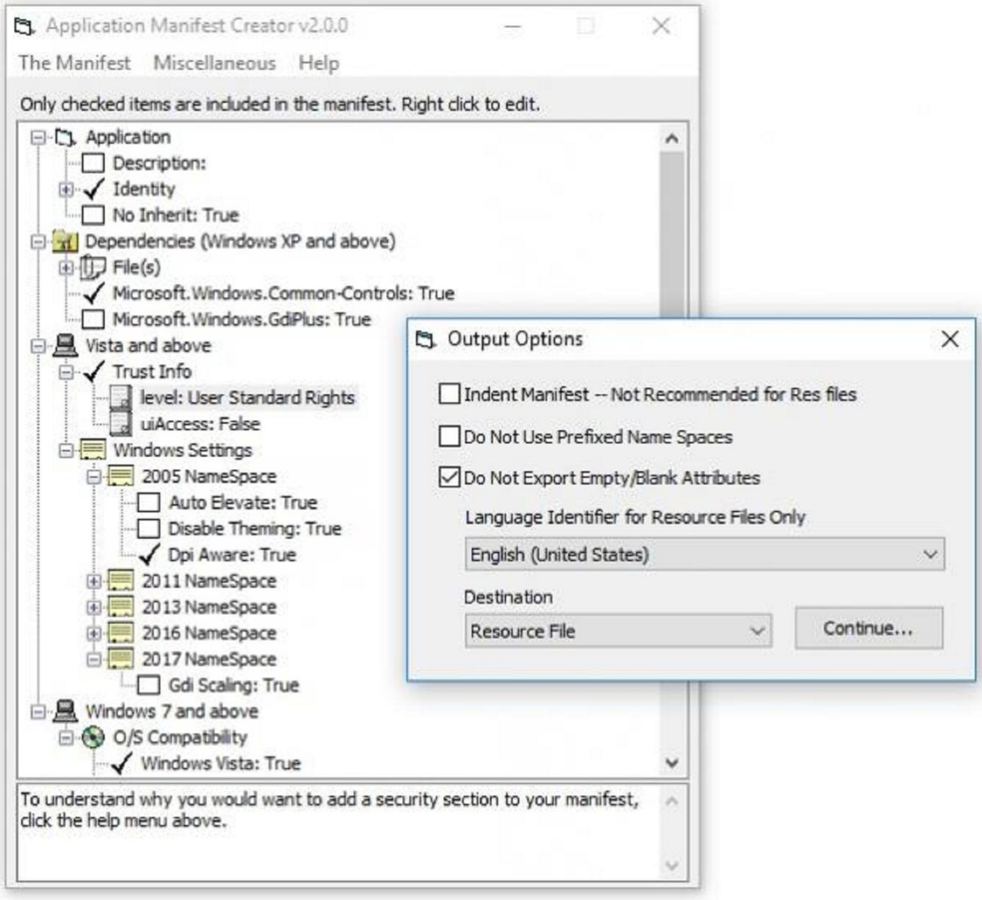
VB-aholic & Lovin' It

[VB6] Manifest Creator II

> This is the second version of this application. The previous version will no longer be maintained.
>
> The project simply creates application-manifests that can be written to disk, clipboard, or VB resource files. The project includes a default/dummy manifest with all known options, as of this date. You would select whichever manifest options you want to include and modify those option values as needed. If you wish to view or replace an existing manifest file, you can import it into the project via the clipboard, disk, or VB resource file. The tool will attempt to parse VB vbp project files and create a new manifest, filling in the Identity element from that file.
>
> The major reasons for revision are several
>
> - Previous version was create only. This version can also edit existing manifests.
> - Use Microsoft XML in the background to create and modify the XML
> - Use XML stylesheets to help transform the manifest based on export options
> - More intuitive design and more modular to assist with future updates by you or me
> - Will no longer ignore hand-jammed manifests within VB resource files, i.e., #24 custom entries
> - Offers ability to update unrecognized (future) manifest entries imported from elsewhere
>
> The manifest is designed to make use of XML qualified names, internally, and is the default mode. However, when exporting the manifest, you can opt to convert to traditional name spaces if you choose.
>
> Caveat: Not 100% of the manifest options available have been fully tested. Some

options can't be tested by me because I don't have the system(s) to perform the desired tests. However, all the options should be technically correct and in accordance with the Microsoft schema for application manifests.

Comments, bug reports, and suggested enhancements are always welcomed.



Only checked items are included in the manifest. Right click to edit.

- Application
  - □ Description:
  - ✓ Identity
  - □ No Inherit: True
- Dependencies (Windows XP and above)
  - File(s)
  - ✓ Microsoft.Windows.Common-Controls: True
  - □ Microsoft.Windows.GdiPlus: True
- Vista and above
  - ✓ Trust Info
    - level: User Standard Rights
    - uiAccess: False
  - Windows Settings
    - 2005 NameSpace
      - □ Auto Elevate: True
      - □ Disable Theming: True
      - ✓ Dpi Aware: True
    - 2011 NameSpace
    - 2013 NameSpace
    - 2016 NameSpace
    - 2017 NameSpace
      - □ Gdi Scaling: True
- Windows 7 and above
  - O/S Compatibility
    - ✓ Windows Vista: True

To understand why you would want to add a security section to your manifest, click the help menu above.

□ Indent Manifest -- Not Recommended for Res files

□ Do Not Use Prefixed Name Spaces

☑ Do Not Export Empty/Blank Attributes

Language Identifier for Resource Files Only: English (United States)

Destination: Resource File

Update History

9 Apr 17: Realigned questionable elements to their correct namespace
10 Apr 17: Fixed error noted in post #4 below
13 Apr 17: Fixed new-found error & added minor enhancement. See post #9 below
15 Apr 17: Optional update. See post #11 below
26 Apr 17: Added support to allow editing Assembly Manifests. See post #12 below
29 Apr 17: Optional update. See post #13

Attached Files

Last edited by LaVolpe; Apr 30th, 2017 at 02:51 PM. Reason: uploaded lastest changes

2. Apr 9th, 2017, 03:21 PM #2

○

Thread Starter

VB-aholic & Lovin' It

Manifest Creator II - Tips, Notes, Etc

Note: Adding a manifest with the Common Controls option allows your form's standard controls to be skinned using the current O/S theme. When adding this element, also start your project with the Sub Main located in a bas module, that routine is used to initialize the common controls. Do know that v5 of the common controls in your toolbox is skinnable, v6 in the toolbox is not. **Tip**: The Help menu can copy a typical Sub Main to the clipboard for pasting into that bas module. **Tip**: This option allows you to add alpha blended icons to buttons if desired. See post #10 below for an example.

Tip: When exporting manifest to a RES file, the file does not need to exist. The tool will create the new RES file and then insert the manifest. However, don't export to a RES file belonging to a project that is currently open, write access will likely be denied.

Note: Many options prevent modifying their values. This is simply because only one value is permitted and it is hard-coded. **Tip**: To show any attributes hidden in the tree view, edit values that are locked, or manually provide a value not in a displayed list of values, hold the shift key down while activating the context menu. Regardless, you cannot enter a blank value if the element/attribute has been flagged as its value cannot be blank.

Tip: Attributes cannot be selected/unselected. How to prevent optional ones from being exported? Set their value to empty/blank. In the "Output" window/form, choose option to not export empty attributes. Optional attributes have the context menu item "Erase". Several can have their values manually entered as a null string. Empty attributes that are exported will have a null value, i.e., <assemblyIdentity ... publicKeyToken=""/>. Optional is not always optional. For example, the publicKeyToken attribute is not optional if it applies. You, of course, are responsible for providing correct values for any item that can be manually entered.

Note: The Identity item cannot be unselected. It is the only required manifest XML element.

Tip: Prefixes for known namespaces are hard-coded, i.e., asmv1, ws2005, etc. If desired, you can change them in the cManifestEx class' pvCreateLUT routine.

> They are not referenced anywhere else directly.
>
> **Tip**: When displaying the Qualified Name context menu item, the base name and NameSpace submenus are FYI only. Since the treeview is designed to be human-friendly for the most part, the actual manifest element name and namespace are not necessarily displayed as they would be if reading the raw XML, if displayed at all. Those submenus give you that info.
>
> **Tip**: VB is mostly DPI-aware. Adding the DPIAware option to your manifest will prevent your forms being stretched in higher DPI settings your users may be running your app in. However, VB6 is only mostly aware, and there are details that must be addressed. There is a tutorial of sorts on things to consider; here are a few:
>
> - Use TrueType fonts everywhere. First step: Assign the form a TrueType font as that is the font that is inherited by most controls added to the form afterwards, negating the need to individually set the font for most controls
> - Image controls should have its Stretch property set to True else manually scaled as needed
> - PictureBox controls should NOT have its AutoSize property set to True else scaled as needed
> - Metafiles are DPI-aware. All other image types are not & may need to be scaled by you
> - Controls without Top/Left, Width/Height properties (i.e., Line controls, etc) can fail to position/size correctly unless its form is saved in ScaleMode of twips. Always leave the form in twips ScaleMode and only change its ScaleMode at run-time if desired, i.e., in Form_Load
> - Always test your app in highest DPI setting available when using the DPIAware option. Minimally, test at 175% DPI. If you are going to see issues, it will be there since internally, VB is not running at the same exact DPI.
>
> **Note**: For those of you that are familiar with SetProcessDPIAware & SetProcessDPIAwareness APIs, Win10 can prevent their usage with the dpiAwareness and dpiAware elements. May only apply if your app, no embedded manifest, calls those APIs and an external manifest is applied to prevent the API usage.

> Last edited by LaVolpe; Apr 30th, 2017 at 02:51 PM. Reason: constantly updated to add tips & notes

3. Apr 9th, 2017, 04:34 PM #3

Thread Starter

VB-aholic & Lovin' It

Embedded vs. External Manifests

> Pretty much external manifests are history. These manifests are external files with the same name as your application and appending a .manifest extension. The reason why they are becoming history is that any embedded manifest will take priority unless certain registry keys are changed.
>
> To embed a manifest in VB6 project, you can use a tool like the one here or you can create an external manifest and add it to your resource file, in the CUSTOM section. Once added, double click on it and change the Type Field to: #24 and change the Id Field to: 1. Tip: Those can be manually extracted by opening the RES file in NotePad.
>
> If you want your VB6.exe to use an external manifest, unless you plan on using some 3rd party software to embed one into the exe (and hope it doesn't corrupt it), some annoyances are expected. Simply adding an external manifest along side of VB6.exe doesn't guarantee it will be used. You may very well need to change a registry setting to make it happen. After you are sure your manifest is being recognized, you will likely want to go and restore the setting:
>
> Key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows \CurrentVersion\SideBySide
> Value: PreferExternalManifest
> Data: 1 = prefer, 0 = don't prefer
>
> After making the change and manifest still appears to not take effect, reboot. It is getting harder and harder to use external manifests within Windows -- and that's probably not a bad thing.

> Last edited by LaVolpe; Apr 16th, 2017 at 12:33 PM.

4. Apr 10th, 2017, 05:07 AM #4

Addicted Member

Re: [VB6] Manifest Creator II

> Code:

```
1. Start program
2. Select "Application" => "Description"
3. Context menu => "Qualified Name" => "Base Name:
description"

Run-time error '9':
Subscript out of range
```

5. Apr 10th, 2017, 11:08 AM #5

Re: [VB6] Manifest Creator II

Say LaVolpe,

I'm not entirely sure when it was all fixed, but resources are correctly unpacked with the correct byte size (not rounded to 4-byte multiples) so long as they're smaller than 65520 bytes (which most .manifest files would certainly be smaller). Once you go over 65520 bytes, they do start getting rounded up to the nearest 4-byte boundary.

Best Regards,

Elroy

EDIT1: Also, great work on the above.

Last edited by Elroy; Apr 11th, 2017 at 04:30 PM.

6. Apr 10th, 2017, 05:56 PM #6

Thread Starter

VB-aholic & Lovin' It

Re: [VB6] Manifest Creator II

Elroy... Correct. After testing it, it doesn't matter if the manifest manually inserted into the RES file is word-aligned or not. I know that when my code inserts the manifest, it must ensure the entry is word-aligned

Edited: removed some comments, regarding another project, where I may have been misinformed. No clarification will be provided

Last edited by LaVolpe; Apr 11th, 2017 at 03:39 PM.

7. Apr 10th, 2017, 10:56 PM #7

Re: [VB6] Manifest Creator II

Interesting about the alignment if you're doing it through code. I suppose though, since you're also creating the manifest through code, it's easy enough to align/pad it.

(Removed my related comments as well.) Peace, to all.
🙂

All The Best,

Elroy

Last edited by Elroy; Apr 11th, 2017 at 04:29 PM.

8. Apr 11th, 2017, 03:37 PM #8

Re: [VB6] Manifest Creator II

I have removed a few posts from this thread, including one of my own. It was taking the thread off topic, though it raised some points I feel are worth pursuing. I'll attempt to pursue them by other means.

My usual boring signature: Nothing

9. Apr 13th, 2017, 06:30 PM #9

Thread Starter

VB-aholic & Lovin' It

Re: [VB6] Manifest Creator II

Uploaded modified project. The modifications include a bug fix and a minor enhancement:

> Bug: Adding a new <file> element from the Miscellaneous menu could result in a duplicate key error. Fixed
>
> Enhancement: Holding shift key down while activating the context menu (right click) unlocks any locked item and/or attributes
>
> -- Any item with 'hidden' attributes displays a new menu item: Show Attributes
> -- Any item where the Value or Text menu item is disabled, will be enabled
> -- Any item where a new value must be selected from a finite list will have "New Value..." appended to the list, allowing manual entries
>
> Some code changes, but a noticeable one should you already have modified the project is that the various cManifestEntryEx class methods that accepts an optional attribute index parameter did accept -1 meaning the method applies to the element vs. an attribute. That has been removed, -1 is no longer a valid index. The MenuActions_Timer event modified due to that change.

Last edited by LaVolpe; Apr 13th, 2017 at 06:59 PM.

10. Apr 15th, 2017, 10:14 AM #10

○

Thread Starter

VB-aholic & Lovin' It 🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩

Using Alpha Blended Icons in Buttons

> Just a FYI item. Another benefit of adding manifests is the ability to add nice alpha-blended icons to your buttons. This method will NOT work if controls are not themed.
>
> The button icon assignment in the screenshot above was accomplished with a couple lines of code
> 1) Load your alpha-blended icon via your favorite method. You will need to destroy the icon at some point
>
> 2) Here are the APIs
>
> Code:

```
Private Const BM_SETIMAGE As Long = &HF7&
Private Declare Function SendMessage Lib "user32.dll"
Alias "SendMessageA" (ByVal hWnd As Long, ByVal wMsg As
Long, ByVal wParam As Long, ByRef lParam As Any) As Long
Private Declare Function DestroyIcon Lib "user32.dll"
(ByVal hIcon As Long) As Long
```

3) Here's the simple assignment after you've loaded your icon

Code:

```
SendMessage Command1.hwnd, BM_SETIMAGE, 1&, ByVal hIcon
```

4) Finally, as needed, destroy your icon during form_unload
How did I use a 48x48 icon? Easy way: use your favorite icon editor/parser and single it out to its own file.

If you are asking how to load an icon, suggest starting with LoadImage API

Last edited by LaVolpe; Apr 15th, 2017 at 10:21 AM.

11. Apr 16th, 2017, 12:42 PM #11

Thread Starter

VB-aholic & Lovin' It

Re: [VB6] Manifest Creator II

Project uploaded to address undesirable effects when running in 200% DPI. This is an optional update since most will never be running this application in 200% DPI. Since I talk about manifesting for DPI-awareness and how VB is mostly DPI-aware, least I should do is make sure the app works well in high DPI.

Should you want to download the changes, here's what changed

- Form size is no longer fixed. At 200% DPI, the form was taller than the screen. So form is resizable, but to prevent resizing the width (which is not what I want), I subclassed the form. Subclassing is optional when in IDE via a message box confirming you will allow it. There is a specific line that can be rem'd out to prevent the message box from displaying (see modWork.SetSizeRestrictions). It will never display once the application is

> compiled.
>
> - The TreeView control would not scale properly in 200% DPI. Added tweak in Form_Resize event to address that.
>
> - Since form can now be resized vertically, needed to add code to resize the treeview and reposition the textbox
>
> By the way, app looks fine at 200% DPI
> 😉

12. Apr 26th, 2017, 07:44 PM [#12](#12)

○

Thread Starter

VB-ahoic & Lovin' It ▮▮▮▮▮▮▮▮▮▮▮

Re: [VB6] Manifest Creator II

> Added moderate support for Assembly Manifests also. This required including nearly 10 new elements, along with ability to merge elements common between both Assembly & Application manifests. This project does not auto-generate assembly manifests; however, you can now edit existing ones or create them from scratch. The Miscellaneous menu has been modified to allow insertion of any Assembly-Manifest related elements.
>
> Other minor changes include defaulting to exclude prefixed element names during exporting of manifests. Can be toggled in the Output window. The "File" menu now has an option to select secondary manifests to be merged with the currently loaded manifest. Since many elements can only exist once in a manifest, any subsequently loaded manifest elements will not overwrite existing elements. For those elements that can exist multiple times, they will be appended. Recommend if using this option, always load the primary/base manifest first, then append other manifests if you want to import their special/custom elements.

13. Apr 29th, 2017, 12:50 PM [#13](#13)

Thread Starter

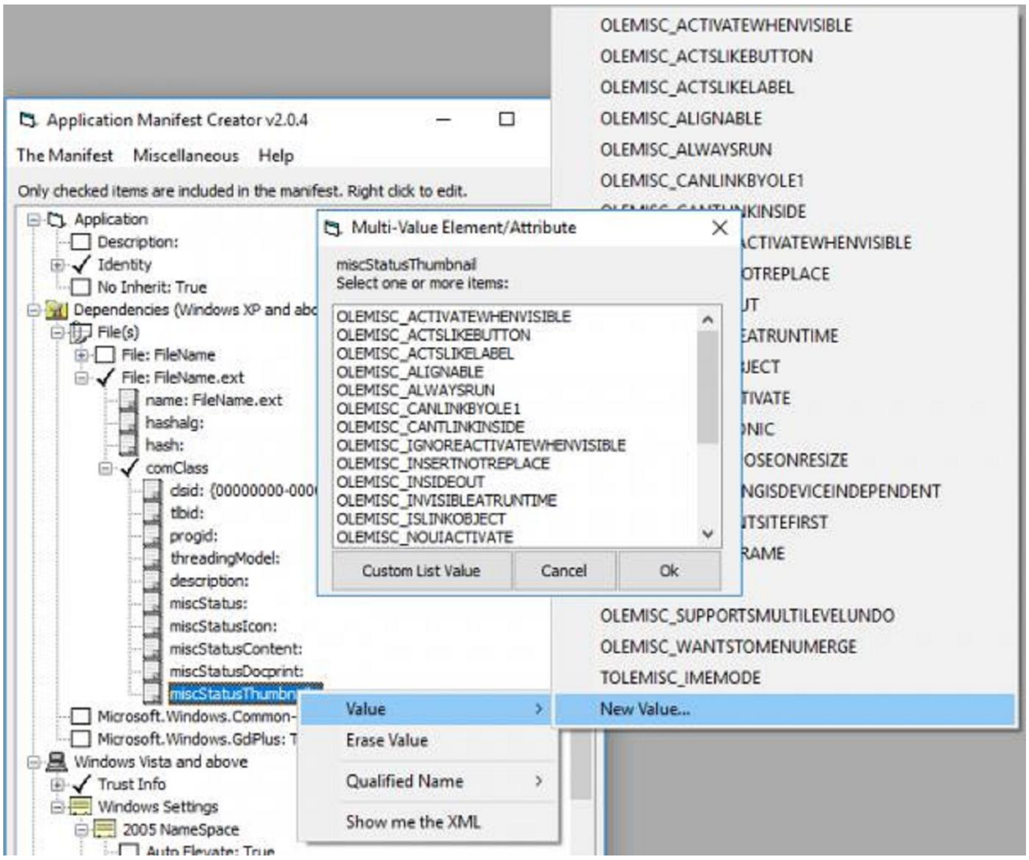
VB-aholic & Lovin' It

Re: [VB6] Manifest Creator II

Updates applied, no need to update your previous version if these options are not desired:

1. Added support for optional assembly-manifest windowClass element.

2. Updated a few more help tips for assembly-manifest related elements
3. All manifest elements can be deleted that were appended via the menu: Manifest|Append/Merge Manifest
4. Any elements that offer a list of possible values now offer those values in a separate submenu
5. Added new form to make selection of comma-delimited element values easier vs. relying on the standard vb InputBox() function. See screenshot below. When the "New Value" menu item is clicked, it generates the displayed form with its multi-select listbox. This additional fluff currently applies only to 2 elements:
- windowsSettings/2016/dpiAwareness
- assembly/file/comClass/miscXXX attributes.



Application Manifest Creator v2.0.4

Only checked items are included in the manifest. Right click to edit.

Application
☐ Description:
✓ Identity
☐ No Inherit: True
Dependencies (Windows XP and abo
File(s)
☐ File: FileName
✓ File: FileName.ext
name: FileName.ext
hashalg:
hash:
✓ comClass
clsid: {00000000-000
tlbid:
progid:
threadingModel:
description:
miscStatus:
miscStatusIcon:
miscStatusContent:
miscStatusDocprint:
miscStatusThumbn
☐ Microsoft.Windows.Common-
☐ Microsoft.Windows.GdiPlus: T
Windows Vista and above
✓ Trust Info
Windows Settings
2005 NameSpace
☐ Auto Elevate: True

Multi-Value Element/Attribute

miscStatusThumbnail
Select one or more items:

Thread Information

Users Browsing this Thread

There are currently 1 users browsing this thread. (1 members and 0 guests)

1. dz32

Posting Permissions

- You **may** post new threads
- You **may** post replies
- You **may** post attachments
- You **may** edit your posts
-

Forum Rules

Click Here to Expand Forum to Full Width

Survey posted by VBForums.