代码基本从网上搜集而来，整理成以下文件：   
包括屏幕截图（和屏幕上看到的一致）；   
以及控件截图（只要该控件在本窗口内显示完全且不被其他控件遮挡就可正确截图） 

**C#代码  [收藏代码](javascript:void())**

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Drawing;
4. **using** System.Linq;
5. **using** System.Runtime.InteropServices;
6. **using** System.Text;
7. **using** System.Threading.Tasks;
8. **using** System.Windows.Forms;
10. **namespace** LC
11. {
12. **class** ScreenCapture
13. {
14. #region 抓取屏幕
15. /// <summary>
16. /// 抓取屏幕(层叠的窗口)
17. /// </summary>
18. /// <param name="x">左上角的横坐标</param>
19. /// <param name="y">左上角的纵坐标</param>
20. /// <param name="width">抓取宽度</param>
21. /// <param name="height">抓取高度</param>
22. /// <returns></returns>
23. **public** **static** Bitmap captureScreen(**int** x, **int** y, **int** width, **int** height)
24. {
25. Bitmap bmp = **new** Bitmap(width, height);
26. **using** (Graphics g = Graphics.FromImage(bmp))
27. {
28. g.CopyFromScreen(**new** Point(x, y), **new** Point(0, 0), bmp.Size);
29. g.Dispose();
30. }
31. //bit.Save(@"capture2.png");
32. **return** bmp;
33. }
35. /// <summary>
36. ///  抓取整个屏幕
37. /// </summary>
38. /// <returns></returns>
39. **public** **static** Bitmap captureScreen()
40. {
41. Size screenSize = Screen.PrimaryScreen.Bounds.Size;
42. **return** captureScreen(0,0,screenSize.Width,screenSize.Height);
43. }
44. #endregion
46. #region 使用BitBlt方法抓取控件，无论控件是否被遮挡
47. /// <summary>
48. /// 控件(窗口)的截图，控件被其他窗口(而非本窗口内控件)遮挡时也可以正确截图，使用BitBlt方法
49. /// </summary>
50. /// <param name="control">需要被截图的控件</param>
51. /// <returns>该控件的截图，控件被遮挡时也可以正确截图</returns>
52. **public** **static** Bitmap captureControl(Control control)
53. {
54. //调用API截屏
55. IntPtr hSrce = GetWindowDC(control.Handle);
56. IntPtr hDest = CreateCompatibleDC(hSrce);
57. IntPtr hBmp = CreateCompatibleBitmap(hSrce, control.Width, control.Height);
58. IntPtr hOldBmp = SelectObject(hDest, hBmp);
59. **if** (BitBlt(hDest, 0, 0, control.Width, control.Height, hSrce, 0, 0, CopyPixelOperation.SourceCopy | CopyPixelOperation.CaptureBlt))
60. {
61. Bitmap bmp = Image.FromHbitmap(hBmp);
62. SelectObject(hDest, hOldBmp);
63. DeleteObject(hBmp);
64. DeleteDC(hDest);
65. ReleaseDC(control.Handle, hSrce);
66. // bmp.Save(@"a.png");
67. // bmp.Dispose();
68. **return** bmp;
69. }
70. **return** **null**;
72. }
74. //         /// <summary>
75. //         /// 有问题！！！！！用户区域坐标不对啊
76. //         /// 控件(窗口)的用户区域截图，控件被其他窗口(而非本窗口内控件)遮挡时也可以正确截图，使用BitBlt方法
77. //         /// </summary>
78. //         /// <param name="control">需要被截图的控件</param>
79. //         /// <returns>控件(窗口)的用户区域截图</returns>
80. //         public static Bitmap captureClientArea(Control control)
81. //         {
82. //
83. //             Size sz = control.Size;
84. //             Rectangle rect = control.ClientRectangle;
85. //
86. //
87. //             //调用API截屏
88. //             IntPtr hSrce = GetWindowDC(control.Handle);
89. //             IntPtr hDest = CreateCompatibleDC(hSrce);
90. //             IntPtr hBmp = CreateCompatibleBitmap(hSrce, rect.Width, rect.Height);
91. //             IntPtr hOldBmp = SelectObject(hDest, hBmp);
92. //             if (BitBlt(hDest, 0, 0, rect.Width, rect.Height, hSrce, rect.X, rect.Y, CopyPixelOperation.SourceCopy | CopyPixelOperation.CaptureBlt))
93. //             {
94. //                 Bitmap bmp = Image.FromHbitmap(hBmp);
95. //                 SelectObject(hDest, hOldBmp);
96. //                 DeleteObject(hBmp);
97. //                 DeleteDC(hDest);
98. //                 ReleaseDC(control.Handle, hSrce);
99. //                 // bmp.Save(@"a.png");
100. //                 // bmp.Dispose();
101. //                 return bmp;
102. //             }
103. //             return null;
104. //
105. //         }
106. #endregion

109. #region 使用PrintWindow方法抓取窗口，无论控件是否被遮挡
110. /// <summary>
111. /// 窗口的截图，窗口被遮挡时也可以正确截图，使用PrintWindow方法
112. /// </summary>
113. /// <param name="control">需要被截图的窗口</param>
114. /// <returns>窗口的截图，控件被遮挡时也可以正确截图</returns>
115. **public** **static** Bitmap captureWindowUsingPrintWindow(Form form)
116. {
117. **return** GetWindow(form.Handle);
118. }

121. **private** **static** Bitmap GetWindow(IntPtr hWnd)
122. {
123. IntPtr hscrdc = GetWindowDC(hWnd);
124. Control control = Control.FromHandle(hWnd);
125. IntPtr hbitmap = CreateCompatibleBitmap(hscrdc, control.Width, control.Height);
126. IntPtr hmemdc = CreateCompatibleDC(hscrdc);
127. SelectObject(hmemdc, hbitmap);
128. PrintWindow(hWnd, hmemdc, 0);
129. Bitmap bmp = Bitmap.FromHbitmap(hbitmap);
130. DeleteDC(hscrdc);//删除用过的对象
131. DeleteDC(hmemdc);//删除用过的对象
132. **return** bmp;
133. }
134. #endregion
136. #region  DLL calls
137. [DllImport("gdi32.dll")]
138. **static** **extern** **bool** BitBlt(IntPtr hdcDest, **int** xDest, **int** yDest, **int**
139. wDest, **int** hDest, IntPtr hdcSource, **int** xSrc, **int** ySrc, CopyPixelOperation rop);
140. [DllImport("gdi32.dll")]
141. **static** **extern** IntPtr DeleteDC(IntPtr hDc);
142. [DllImport("gdi32.dll")]
143. **static** **extern** IntPtr DeleteObject(IntPtr hDc);
144. [DllImport("gdi32.dll")]
145. **static** **extern** IntPtr CreateCompatibleBitmap(IntPtr hdc, **int** nWidth, **int** nHeight);
146. [DllImport("gdi32.dll")]
147. **static** **extern** IntPtr CreateCompatibleDC(IntPtr hdc);
148. [DllImport("gdi32.dll")]
149. **static** **extern** IntPtr SelectObject(IntPtr hdc, IntPtr bmp);
150. [DllImport("user32.dll")]
151. **public** **static** **extern** IntPtr GetDesktopWindow();
152. [DllImport("user32.dll")]
153. **public** **static** **extern** IntPtr GetWindowDC(IntPtr ptr);
154. [DllImport("user32.dll")]
155. **public** **static** **extern** **bool** PrintWindow(IntPtr hwnd, IntPtr hdcBlt, UInt32 nFlags);
156. [DllImport("user32.dll")]
157. **static** **extern** **bool** ReleaseDC(IntPtr hWnd, IntPtr hDc);
158. #endregion
159. }
160. }