

VisualBasic代码优化的六条军规 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/166/2021_2022_VisualBasi_c97_166174.htm 计算机等级考试训练软件《百宝箱》在优化程序代码大小的诸多技术中，大多包括从代码中删除不必要的元素。在编译应用程序时，visual basic自动删除某些元素。而标识符名称、注释、空行的长度或数量是无须限制的，当应用程序作为一个.exe文件运行时，以上这些元素都不会影响应用程序所占内存的大小。其他元素，如变量、窗体和过程，确实要占据内存的一部分空间。最好将它们精简以使效率更高。下面介绍6种方法可用于缩小应用程序所需内存和减少代码大小，希望能对初学者有所帮助。

- 1、减少加载窗体、控件数目和用标签代替文本框每一个加载的窗体，无论可视与否，都要占据一定数量的内存(其数量随窗体上控件的类型和数量，以及窗体上位图的大小等的不同而变化)。只在需要显示时才加载窗体，不再需要时卸载窗体(而不是隐藏窗体)。记住，任何对窗体的属性、方法或控件的引用，或对用new声明的窗体变量的引用，都会导致visual basic加载该窗体。使用unload方法卸载窗体时，只能释放部分窗体所占空间。要释放所有空间，可用关键字nothing使窗体的引用无效：set form=nothing 当设计应用程序时，窗体应尽量少用控件。实际的限制取决于控件的类型和系统，但实际上，含有大量控件的窗体将运行缓慢。一项与之相关的技术是：设计时尽可能地使用控件数组，而不是在窗体上放置大量同类型的控件。控件数组是一组具有共同名称和类型的控件。它们的事件过程也相同。在设计时，使用控件数组添加控件所消耗的资

源比直接向窗体添加多个相同类型的控件消耗的资源要少。当希望若干控件共享代码时，控件数组也很有用。标签控件label占用的windows资源比文本框textbox少，因此在可能的情况下，应使用标签代替文本框。例如，当窗体上需要一个隐藏的控件保存文本时，使用标签更有效。

2、使用磁盘文件或资源和组织模块在设计时，直接放入应用程序的数据(象属性或代码中的文字字符串和数值)将增加运行时应用程序占用的内存。运行时从磁盘文件或资源中加载数据可减少占用内存。这对大位图和字符串特别有价值。资源文件实际上是由一系列独立的字符串、位图或者其他项目组成的，其中每一项都有一个唯一的标识符。可以使用类似于在microsoft visual c 中提供的文本编辑器和资源编译器创建资源文件。编译过的资源文件带有.res扩展名。visual basic只在需要时才加载模块即当代码调用模块中的一个过程时，模块才被加载到内存。如果从未调用一特定模块中的过程，visual basic决不加载该模块。因此，尽量把相关的过程放在同一模块中，让visual basic只在需要时才加载模块。

3、考虑替换variant数据类型variant数据类型使用极其灵活，但是比其他数据类型所占内存大。当要压缩应用程序多余的空间时，应考虑用其他数据类型替代variant变量，特别是替代variant变量数组。每一个variant占用16个字节，而integer占2个字节，double占8个字节。变长字符串变量占用4个字节加上字符串中每一个字符占用1个字节，但是，每一个包含字符串的variant都要占用16个字节加上字符串中每一个字符占用1个字节。因为它们太大，因此在用作局部变量或过程的参数时，variant变量是特别烦人的，这是因为它们消耗堆栈空间太快。但在有些情况下，使用其他数

据类型替代variant，灵活性降低了，为弥补损失的灵活性，不得不增加更多的代码。结果是大小没有真正的减小。

4、使用动态数组并在删除时回收内存使用动态数组代替固定数组。当不再需要动态数组的数据时，用erase或redim preserve放弃不需要的数据，并回收数组所用内存。例如，用以下代码可回收动态数组所用空间：erase myarray 里，erase完全删除数组，redim preserve则只缩短数组而不丢失其内容：redim preserve myarray(10, smallernum) 删除了固定大小数组，也不能回收该数组所占空间--只是简单地清除数组每一元素中的值。如果元素是字符串，或包含字符串或数组的variant变量，那么删除数组可回收这些字符串或variants所占内存，而不是数组本身所占内存。

5、回收被字符串或对象变量用过的空间当过程结束时，可自动回收(非静态)局部字符串和数组变量所用空间。但是，全局和模块级的字符串和数组变量一直存活到整个程序结束。要想应用程序尽量小，就得尽可能回收这些变量所用空间。将零长度字符串赋给字符串变量，可回收其空间：somestring var= 回收空间 同样，将对象变量设置成nothing可回收该对象所用的部分(而不是全部)空间。例如，删除一个form对象变量：global f as new statusformf.show 1 form加载并以模态显示x=f.text1.text 用户按下按钮,隐藏窗体unload f 删除窗体可视部分set f=nothing 回收空间(模块数据)即使没有使用显式窗体变量，也应注意将不再用的窗体卸载，而不是简单地隐藏。

6、消除死代码和无用的变量在开发和修改应用程序时，可能遗留了死代码--代码中的一个完整过程，而它并没有被任何地方调用。也可能声明了一些不用的变量。虽然在创建.exe文件中，visual basic确实可删除无用的

常数，但不能删除无用的变量和死代码。注意要复查代码，查找并删除无用的变量和死代码。如debug.print语句，在运行.exe时被忽略，可它常常出现在.exe文件中。当创建.exe文件时，含有字符串和变量作为参数的debug.print语句不会被编译。但对于含有函数作为参数的debug.print语句，其本身被编译器忽略，而函数则被编译。因此，在应用程序运行时，函数被调用，但返回值被忽略。因为在.exe文件中，函数作为debug.print的参数出现时，将占用空间和cpu周期时间，所以在生成exe文件前，最好删除这些语句。在编辑菜单中使用查找命令搜索特定变量的引用。或者当每个模块都含有option explicit语句时，通过删除或注释该变量的声明，并运行应用程序，可迅速发现变量是否被使用。若该变量被使用，则visual basic将出错。若不出错，则该变量没被使用。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com