

RE: Maintaining Maximum Platform Compatibility

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2006-08/msg00803.html>

- *From:* Jim Thomlinson <James_Thomlinson@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 4 Aug 2006 11:27:02 -0700
-

Early binding is tied to the exact reference that you select in the references list. If the end user does not have that EXACT reference then the code crashes (as you have seen). I choose Word because that reference is neither backwards nor forwards compatible (ain't that just a kick in the shorts). If you have Excel 10 then your reference will (in all likelihood) be Word 10. If you send that to your boss who is on Word 9 or Word 11 then... Boom... In this case Late Binding is really the only good option. Late binding creates an instance of the Word application. It does not care about versions.

Right now I assume you have chosen some references from the Tools -> References list. What you want to do is un-select the references you have chosen. Now if you try to compile your code it should crash because you are trying to create things that the compiler no longer understand. Where it has those issues you want to change to a generic object and then use createobject to instantiate the object you require...

What references do you have that are causing you trouble???

--

HTH...

Jim Thomlinson

"Bob the Kart Racing fool" wrote:

I can't thank you enough for your quick response! I am not using any pivot tables in my workbook, so there is no issue there. The part that I'm worried about is the VB code that I am using.

So I will want to use late binding? Because, early binding would automatically update to newer versions of a reference and thus cause that saved file to stop working on earlier versions, correct?

So this would solve compatibility issues:

```
Sub LateBinding()
```

RE: Maintaining Maximum Platform Compatibility

Dim appExcel As Object

```
Set appExcel = CreateObject("Excel.Application")
appExcel.Visible = True
End Sub
```

When would I have LateBinding() execute? Also, How do I remove Early Binding references? I can't simply uncheck their box... I receive an error. I'm guessing this is done through code.

One last thing... I would like to use User Forms, do I need to reference a library for this as well or is that taken care of in the above code?

Sorry about all of the questions! I'm so used to programming robots... their easy! Just compile the code with the libraries you need and voila, working robot

Thanks again Jim for all your help!

"Jim Thomlinson" wrote:

There are two aspects to compatibility. One is that Excel has changed over the years and new features have been added. This means that code written for the current version of Excel may not work on older versions. You are best off to write to the lowest version of Excel possible. The one area that can cause difficulty is code for Pivot Tables as the Pivot Table Engine changed in xl2002 and the code is different for the different versions.

The other issue (that I think you are alluding to) is the compatibility of references. This gets into early or late binding. Early binding is Selecting Tools -> References and binding a reference to your project at design time. The other method is late binding where you create your reference to the library at run time. Early binding makes the code run a tad faster and it allows you to use the intelligence while coding (a very handy feature). Late binding has the advantage of not being tied to a specific instance of a reference. Check out these two examples...

```
Sub EarlyBinding()
'Must Reference Word Library (9.0, 10.1, 11.0, ...)
Dim appWord As Word.Application
```

```
Set appWord = New Word.Application
appWord.Visible = True
End Sub
```

```
Sub LateBinding()
Dim appWord As Object
```

```
Set appWord = CreateObject("Word.Application")
appWord.Visible = True
```

RE: Maintaining Maximum Platform Compatibility

End Sub

If you can not gurantee which references an end user system will have then late binding is great. Speed is not usually too big of an issue with late binding. I find the loss of intellisense to be a bit of an issue. I will normally reference the project to the library right up front and write my code assuming early binding. When the code is up and running I will then switch it to late binding (a little bit of rewrite but usually not too bad).

--
HTH...

Jim Thomlinson

"Bob the Kart Racing fool" wrote:

I am just learning about all the issues associated with maintaining compatibility between different software years, discovered when I sent the completed file to my boss for approval& low and behold, a reference error. So now, trying desperately to gather my pride from the floor, I come to you because this forum has helped me so much in the past two weeks.

I need to maintain compatibility of my excel 2003 document that I have created. I need it to work on Office 2000 and up. I have read all the posts that I can find about binding but, I still don t understand how to use this to automatically maintain my files compatibility. How is this code written and implemented? If the file is saved, and opened again on an earlier version of excel& will it still work?

I desperately need guidance&

Thank you for your help in advance!

~Josh