



Understanding How Word Looks for and Loads Templates

By: Karen Walker, Senior Developer with Payne Consulting Group, Inc.

Recently I had an experience that changed my perspective on how Word manages templates. I was troubleshooting a problem on a client's Desktop in which there were multiple globally-loaded templates (or "add-ins") loading in Microsoft Word. As my add-in was attempting to insert an AutoText entry into the header of the active document, a runtime error was consistently being triggered (Run-time error '91': "Object variable or With block variable not set").

I was finally able to pinpoint the exact line of code where my macro was breaking. It was immediately after setting an object variable to point to the first page header range of the active document. The first page header existed, but after I had instantiated it (as well as checking to be sure that it was indeed a valid object), I then tried to insert an AutoText into the range and *BOOM!* - Error 91. There was no logical reason for the pointer to the header range to be self-destructing at that point in my code.

After spending several hours going 'round and 'round with different ways of instantiating the header (for instance, declaring it as a late-bound "Object" rather than as an early-bound object) and coming up empty-handed, I recalled something I had heard long ago about template loading order and potential conflicts between global templates.

I exited Word and placed an underscore (_) before the name of my global template. When Word restarted and attempted to reproduce the error, it no longer happened. Great, I thought. I fixed the problem by forcing my template to load first. (I only hoped that nothing else would break now that the template loading order had been changed.) I still would have liked to know why the problem was happening and why changing the loading order of my template resolved it.

After more experimentation, I realized that I had it all wrong: When the name of the template was changed, it was actually making it load last, not first. This led me to question how Word handles the loading of multiple global templates on startup.

I'd like to share my findings with you, as well as additional information I've learned (sometimes the hard way) about how Word looks for and manages user templates. Hopefully this will come in handy when you're dealing with environments where multiple Word add-ins and user templates are in use.

Word Loads Global Templates In Reverse Alphabetical Order

You can view the list of globally-loaded templates—if there are any—by launching Word and going to the Templates and Add-ins command under the Tools menu. The list appears in alphabetical order, sort of. You're actually looking at the entire "layer" of globally-loaded templates, from the top down, starting with the last template that loaded. (I said "sort of" because if you have global templates loading from a series of different folders, the list will consist of multiple alphabetical sequences.)

Target Audience:

Word Developers, Trainers,
Support Staff, Power Users

Level:

Intermediate to Advanced

d.dot was the first template to load during this work session, and a.dot loaded last. The templates are listed in the order that they loaded as Word was launching, from the bottom up.

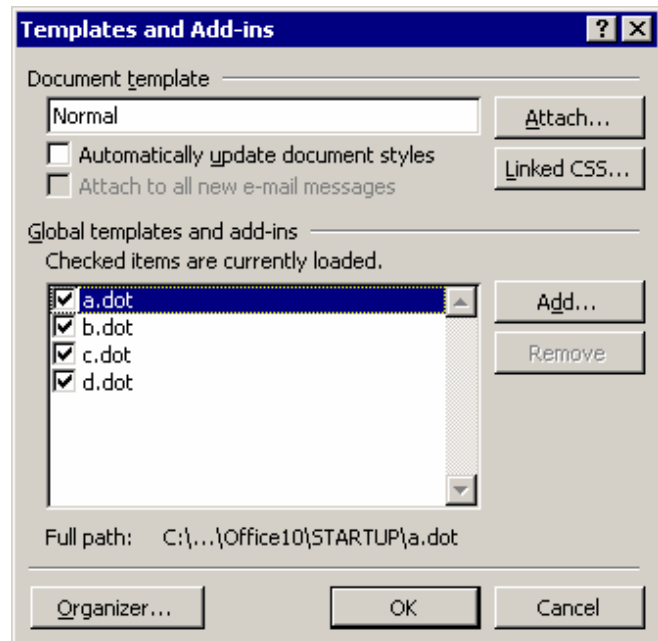


Figure 1: Templates Loaded

Let's say I have four templates: "a.dot", "b.dot", "c.dot", and "d.dot". These templates are located in Word's Startup folder, and each contains a little AutoExec procedure as well as a custom menu item. The code within each template's AutoExec procedure automatically runs when the template is loaded.

```
Sub AutoExec ()
    MsgBox "AutoExec firing in " & ThisDocument.FullName
End Sub
```

As Word launches, a message box appears for each template that is loaded, giving a good indication of the order in which they are loaded.

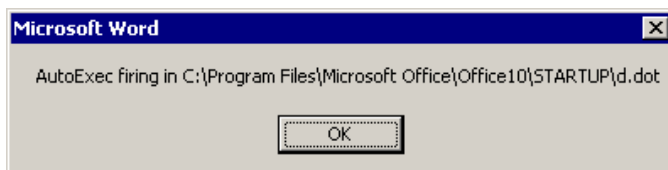


Figure 2: An example of the message box that appears as a result of AutoExec routine.

The positioning of my custom menu bar items also reveals that the templates were loaded into the Word environment in reverse alphabetical sequence (see Figure 3).

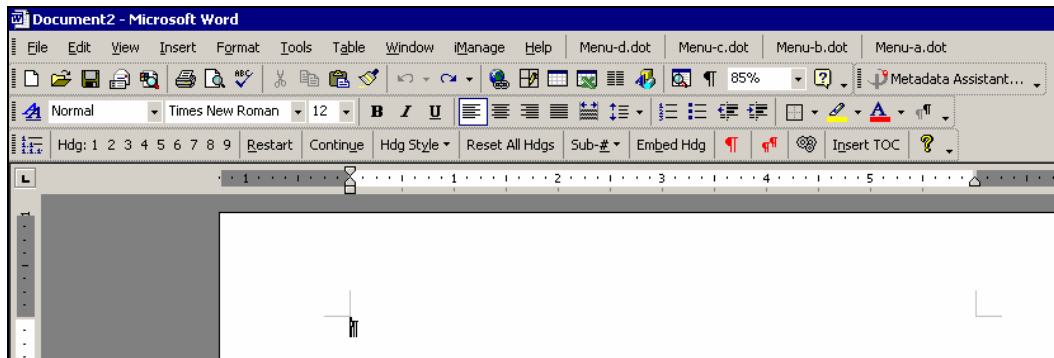


Figure 3: The position of custom menu bars is influenced by the order in which the templates are loaded into the Word environment.

Global Templates in Multiple Folders are Loaded in a Predictable Sequence

Here is the logic, in as much as I can surmise:

1. Shortcuts (.lnk files) pointing to templates stored in alternate folder locations are loaded first, in reverse alphabetical sequence (based on the name of the source template, not the shortcut), within each folder. For example, the Payne Metadata Assistant has a global add-in that is loaded when Word starts up. The actual MetadataAssistant.dot file is located in a folder named C:\Program Files\Payne Consulting Group\Metadata Assistant. The reason that the .dot loads when Word launches—despite the fact that it is not located directly in one of Word’s startup folders—is because the Startup folder contains a shortcut pointing to this template.
2. Next, Word looks to the “traditional” Startup folder located in the Microsoft Office program files folder.* For Word 97 and Word 2000, this is typically C:\Program Files\Microsoft Office\Office\Startup. For Word 2002, it is C:\Program Files\Microsoft Office\Office10\Startup. (Note that in this example, Microsoft Office is installed to the “usual” location of C:\Program Files\Microsoft Office, but if you have installed Office to a different location, just substitute the correct path of the Microsoft Office program files location.) Any .dot files located in this Startup location are loaded, again, in reverse alphabetical order.

Note: Although this location may not be indicated as the “official” Startup location under Tools, Options in Word, Word always looks there; therefore, this folder is a handy location for storing global templates that will be shared by all users who log into a workstation (as opposed to the user profile-based location mentioned in the next paragraph).

3. Finally, if there is another path defined as the Startup location in Tools, Options (such as the default path associated with a user’s Windows 2000 or XP profile of C:\Documents and Settings\[username]\Application Data\Microsoft\Word\Startup), Word then loads any .dot files in that folder in reverse alphabetical sequence.

Note: You might decide to create a folder called C:\Documents and Settings\All Users\Application Data\Microsoft\Word\Startup, hoping that the templates will be loaded by Word no matter what user is logged into the workstation. However, unless you explicitly define this as the Startup file location in Tools, Options, it won’t be acknowledged by Word. You’re better off using the C:\Program Files\Microsoft Office\Office\Startup location (or for Word 2002, Program Files\Microsoft Office\Office10\Startup), which is created by default and automatically recognized by Word.

Maybe you’re wondering: why does it matter what order Word startup files load? Usually it doesn’t matter (and I prefer it that way). But if several of the .dot files being loaded during Startup are triggering their own AutoExec

routines or altering Word's menus, conflicts can occur. Unfortunately, conflicts between global add-ins may lead to unexpected behavior (such as the header object “breaking” when I tried to use it programmatically) or even GPFs. Therefore, understanding the order in which global templates are being introduced into the Word environment might be useful in times of crisis.

The Significance of File Locations for User Templates and Workgroup Templates

Unlike global templates, which typically comprise only code (or “macros”) and are sometimes referred to as add-ins, user or workgroup templates are used primarily as boilerplates for creating new documents (although they may also contain code). In order to determine what templates are available to you during a given work session, Word looks at the settings for User templates and Workgroup templates under Word's Tools, Options, File Locations tab.

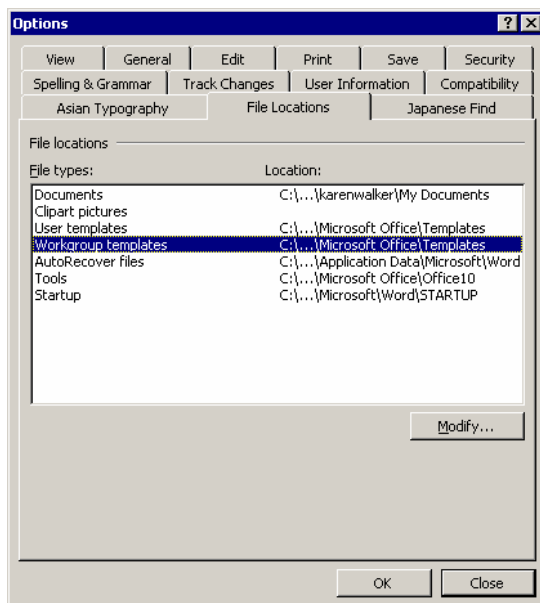


Figure 4: Word's file locations in the Tools, Options dialog box

In previous versions of Word, the “traditional” or default location for templates was usually C:\Program Files\Microsoft Office\Templates. But after you install Word 2002, the setting you'll find under Tools, Options will look something like this: C:\Documents and Settings\[username]\Application Data\Microsoft\Templates.

Whereas Word 2002 kindly checks for the existence of files in the “common” default Startup location to see if there's anything there that needs to be loaded, it does not automatically acknowledge the existence of the C:\Program Files\Microsoft Office\Templates location.

As a rule, the only folders Word looks in for the existence of user and workgroup templates are those explicitly defined under Tools, Options, File Locations. That's why it's useful to define the Workgroup templates' location in addition to the User templates' location. You can place templates that will be shared by all users in the shared Workgroup location (either on the local machine or on the network), and leave the User templates location set to the profile-specific path. (One advantage of keeping the User template location associated with the profile of the person who's logged in is that it allows for management of each user's Normal.dot file or personal templates – where users may keep their own AutoText entries, macros, and other customizations they have made to the Word environment.)



The Importance of the Attached Template

It's helpful to understand the concept of the "attached template," especially if you create or distribute templates that are designed to run macros in addition to providing boilerplate text.

You can see what Word considers to be the "attached template," or the originating template used to create a document, by going to the Tools, Templates and Add-ins command when you have the document open.

However, you can't count on Word to automatically resolve a document's attached template when the document is opened, especially if your document is opened on a different user's machine or if the template that it was created from has been moved. Normally, this does not pose much of a risk, but if there are macros or AutoText entries that you want to make sure are available to a document whenever it is open and being edited, the attached template needs to be resolved. (Even worse than not resolving at all, a document may resolve to the wrong attached template altogether when it is opened.)

For example, you create a new memo from C:\Documents and Settings**<your name>**\Application Data\Microsoft\Templates\Memo.dot. You then save this memo and e-mail it to a co-worker. The co-worker opens it, and because Word can't find the Memo.dot in the exact path that it wants to find it in (the one under your user profile, which of course is not on your co-worker's machine), the document attaches to the first Memo.dot it finds in the User templates location. (It does not look in the equivalent folder on your co-worker's machine, C:\Documents and Settings**<your co-worker's name>**\Application Data\Microsoft\Templates\.)

How Word Attempts to Resolve a Document's Attached Template When It is Opened

1. Word checks for the attached template in the exact path that it is supposed to be in (the creation location).
2. If not found in the expected location, Word then looks in the folder where the document has been opened, hoping to find the attached template there.
3. Next, Word looks in the location specified for User templates under Tools, Options, File Locations.

There are a couple of catches. First, Word will not look into sub-folders when trying to resolve the attached template. For example, if Word gets to step three in the above sequence of steps, and you have your User templates location set to "C:\Program Files\Microsoft Office\Templates," but the template is actually in "C:\Program Files\Microsoft Office\Templates\Firm Templates," it will fail to resolve. That's because it's a folder below the expected location.

Also, if you have both a User templates and a Workgroup templates location defined, and a different version of Memo.dot resides in both places, the one in User templates will triumph as the attached template.

Finally, if the attached template can't be resolved, Word falls back to Normal.dot as the attached template. But if you look at Tools, Templates and Add-ins, you may still see the original path of the missing attached template. To find out what the actual attached template is after Word's attempt to resolve the original template failed, you can enter the following in the Immediate window in the VBA Editor:

```
?ActiveDocument.AttachedTemplate.FullName
```



In conclusion, there are some nuances to how the various versions of Word manage both global and user templates. Hopefully, you now have a more intimate understanding of what to expect, and how to overcome challenges that might arise when troubleshooting template-related issues in client environments.

About Payne Consulting Group, Inc.

Payne Consulting Group is a software training and development company headquartered in Seattle, WA. Payne has authored ten books on Microsoft products including: *Word 2002*, *Word 2000*, and *Word 97 for Law Firms*, *Essential Excel 97* and *Essential Excel 2000*, *Excel Expert Solutions*, *Office Expert Solutions*, and more. The company also specializes in developing software and includes the Metadata Assistant featured on the front page of the Wall Street Journal (Oct. 20, 2000) which removes embedded and often confidential information buried in documents, the Forms Assistant that automates the creation of letter, memo, fax, agreement and pleading documents, and the Numbering Assistant which simplifies the process of numbering. Payne offers high end Master Series and Visual Basic for Applications classes, end-user training, courseware, essential project management as defined by the Project Management Institute, and custom development.

Payne is a member of the Microsoft Legal Advisory Council, the American Bar Association, and the American Society of Journalists and Authors. Payne is a frequent speaker at Microsoft conferences and events.

Web Address: www.payneconsulting.com. Telephone: 206-344-8966.

Default File Locations for Templates	Word 97	Word 2000	Word 2002
Startup	C:\Program Files\Microsoft Office\Office\Startup	C:\Program Files\Microsoft Office\Office\Startup	C:\Documents and Settings\[username]\Application Data\Microsoft\Word\Startup * (Plus the undocumented additional location of: C:\Program Files\Microsoft Office\Office10\Startup)
User Templates	C:\Program Files\Microsoft Office\Templates	C:\Program Files\Microsoft Office\Templates	C:\Documents and Settings\[username]\Application Data\Microsoft\Templates *
Workgroup Templates	<Not designated by default>	<Not designated by default>	<Not designated by default>

Table 1 Places where the various versions of Word looks for templates unless you specify otherwise under Tools, Options, File Locations

***Note:** File locations are typically on the C: drive, but this depends on what drive Microsoft Office is installed on. In your environment, Word may be looking to the D: drive or some other location. If trying to navigate to this folder on Windows 2000 or XP, make sure Show Hidden Files and Folders is turned on under the Tools, Folder Options, View tab of the Explorer folder window.

Equivalent Registry Keys for these settings (substitute 8.0, 9.0, or 10.0 for the version):

User Templates:

HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Common\General\UserTemplates

Workgroup Templates:

HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Common\General\SharedTemplates

Startup:

HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Word\Options\STARTUP-PATH