

An Esquire Innovations, Inc.  
White Paper

**ESQUIRE**  
INNOVATIONS, INC.

Temecula, CA  
951-506-5641  
[www.Esqlnc.com](http://www.Esqlnc.com)

**The Load behavior of COM Add-ins, Normal.Dotm and  
Global Templates in Microsoft Word 2010**

*By Randall Farrar, President*

June 16, 2011

## Contents

---

<b>Contents .....</b>	<b>2</b>
<b>Introduction.....</b>	<b>2</b>
COM Add-ins.....	2
Identify Word COM Add-ins .....	2
Global Templates .....	3
Normal.dotm .....	3
<b>Microsoft Word's Load Behavior .....</b>	<b>3</b>
Load Behavior Test Setup.....	3
COM Add-in	
<i>ThisAddIn_Startup</i> subroutine.....	4
Normal.dotm AutoExec.....	4
Global Template AutoExec .....	4
Running the Microsoft Word	
Load Behavior Test.....	4
Microsoft Word Load Behavior	
Test Results .....	4
Why the Load Behavior is	
Important.....	5
Business Problem/Solution	
Examples .....	5
Problems With Using	
Normal.dotm .....	5
<b>Summary .....</b>	<b>6</b>

## Introduction

The load behavior of Microsoft Word 2010<sup>1</sup> refers to the way extended applications, or **Add-ins**, are loaded on startup. Add-ins contain automation that allow developers to build in features that are not part of the out-of-the-box installation of Word. These Add-ins can be in the form of COM Add-ins, adding to the existing Normal.dotm, or Global Templates.

It is important for support staff and developers to understand the load behavior of Add-ins because this knowledge can be used to:

- Troubleshoot Add-ins that are causing issues
- Create solutions that may or may not fit a particular type of Add-in and its behavior
- Update and support applications

### COM Add-ins

A COM add-in is a COM DLL that implements the IDTExtensibility2 COM interface. IDTExtensibility2 provides event notifications that occur to add-ins, such as when they are loaded, unloaded, updated, etc. COM Add-ins always run within the process of their container application, i.e., Microsoft Word.

### Identify Word COM Add-ins

1. Go to the File tab in Word 2010
2. Select Options
3. Select Add-ins
4. The Add-ins and application extensions are categorized as follows:

**Active Application Add-ins:** Lists the extensions that are registered and currently running in your Word program.

---

<sup>1</sup> Actually, the basic load behavior of Microsoft Word hasn't changed since Word 97.

**Inactive Application Add-ins:** Lists the add-ins that are present on your computer, but are not currently loaded.

The **Type** column shows what type of Add-In it is. A COM Add-in will be listed as *COM Add-in*.

COM Add-ins can contain:

- .NET Classes, modules, WinForms, Windows Presentation Foundation (WPF) and pretty much anything you can do with .NET.<sup>2</sup>
- Ribbon Customizations
- Built-in Word Command Keyboard binding (setting keyboard shortcuts)
- Code that can redefine built-in word functionality

### **Global Templates**

Global Templates are Macro Enabled Templates (.dotm) that are placed in the user's Startup folder (C:\Users\%username%\AppData\Roaming\Microsoft\Word\STARTUP)

or

Program Files Startup folder (C:\Program Files (x86)\Microsoft Office\Office14\STARTUP)<sup>3</sup>.

To view your Global Templates use the same steps outlined in **Identify Word COM Add-ins** on page 2.

The **Type** column shows what type of Add-In it is. A Global Template will be listed as *Template*.

Global Templates can contain:

- Macros

---

<sup>2</sup> This is what makes COM Add-ins so powerful.

<sup>3</sup> It would be C:\Program Files Microsoft Office\Office14\STARTUP for 32 bit Windows OS.

- Autotext/Building blocks
- Ribbon Customizations
- Advanced Keyboard Shortcuts

### **Normal.dotm**

The Normal template is always loaded. It can be used for all new blank documents created. The Normal template can contain:

- Macros
- Styles
- Autotext/Building blocks
- Ribbon Customizations
- Advanced Keyboard Shortcuts

The location of the Normal template is:  
C:\Users\%username%\AppData\Roaming\Microsoft\Templates

### **Microsoft Word's Load Behavior**

Now that we're familiar with Microsoft Word's three startup Add-ins, let's look at the order in which they are loaded when Word starts.

#### **Load Behavior Test Setup**

To test Microsoft Word's load behavior I programmatically set up three simple tests that gave me a pop up message to let me know in which order each Add-in loaded. These were the tests I set up:

1. Create a COM Add-in called COMAddinLoadTest (using Microsoft Visual Studio 2010) that will have a message box in the *ThisAddIn\_Startup* subroutine that behaves the same as the VBA AutoExec subroutine (see #2 below) when Microsoft Word starts.
2. Create an *AutoExec* VBA subroutine in the Normal.dotm. An AutoExec subroutine is a Microsoft Word auto macro that will automatically run when word starts the template it is contained in, in this case Normal.dotm.
3. Create a Global Template called GlobalTemplateTest.dotm, and place it in the user's Startup folder. Create an AutoExec VBA subroutine in this template.

### COM Add-in *ThisAddIn\_Startup* subroutine

I created a very simple COM Add-in using Microsoft Visual Studio 2010 called COMAddinLoadTest. The only line of code I added is as follows:

```
Private Sub ThisAddIn_Startup() _  
    Handles Me.Startup  
    MsgBox("COM Addin loaded")  
End Sub
```

When Microsoft Word starts and when the COM Add-in loads, the message box in the *ThisAddIn\_Startup* subroutine will pop up on the screen.

### Normal.dotm AutoExec

This is an even easier code change in the Normal.dotm. In any module in the Normal project, add the following lines of code:

```
Public Sub AutoExec()  
    MsgBox "Normal.dotm loaded"  
End Sub
```

When Microsoft Word starts and when the Normal.dotm template loads, the message box in the AutoExec subroutine will pop up on screen.

### Global Template AutoExec

Just as in the Normal.dotm, I created a global template called GlobalTemplateTest.dotm and placed it in the user's Startup folder (see **Global Templates** on page 3). In any module in the GlobalTemplateTest project, add the following lines of code:

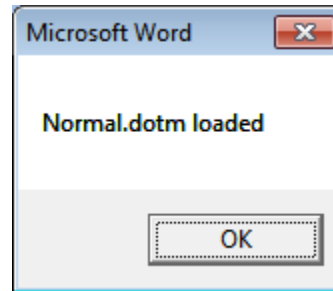
```
Public Sub AutoExec()  
    MsgBox "Global Template loaded"  
End Sub
```

When Microsoft Word starts and when the GlobalTemplateTest.dotm template loads, the message box in the AutoExec sub subroutine will pop up on screen.

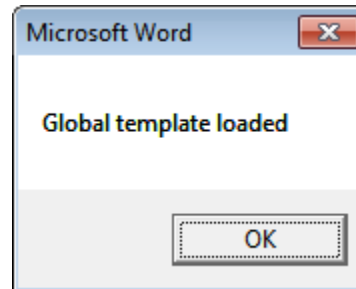
### Running the Microsoft Word Load Behavior Test

I made sure Microsoft Word was closed, and then started Word to see in which order the message boxes appeared.

1.



2.



3.



### Microsoft Word Load Behavior Test Results

Therefore, from this test we can see the load behavior of Microsoft Word occurs in the following order:

1. Normal.dotm
2. Global Templates
3. COM Add-ins

## ***Why the Load Behavior is Important***

Knowing when the startup code in the application loads can often help determine the best development approach to use.

### **Business Problem/Solution Examples**

#### **Problem**

Suppose there's a particular vendor's COM Add-in that your firm has invested in and it solves an incredible business problem. BUT, there is one problem with this COM Add-in – when Microsoft Word starts up for the first time the default “Document1” is closed, which is fine, but you want a firm blank document open that is based on the firm's blank template not the Normal template (which Document1 is based on).

#### **Solution**

You would create an AutoExec subroutine in a Global Template<sup>4</sup> that creates a new blank document based on the firm's blank template. For this example, the blank template is called “iBlank Portrait.dotx”

The following example adds a new blank document for the user and sets the window caption to BlankDocument1.

```
Public Sub AutoExec()  
    Dim oDocument As Document  
    Set oDocument = _  
Documents.Add(Template:= _  
"C:\Program Files (x86)\Esquire  
Innovations\iCreate\iTemplates\iBlank  
Portrait.dotx")  
    oDocument.ActiveWindow.Caption  
= _ "BlankDocument1"  
End Sub
```

---

<sup>4</sup> A COM Add-in could also be used, but a Global Template is simpler and easier to manage for in-house staff.

#### **Problem**

You need to bind a keyboard short cut (Ctrl+Shift+C) to an internal macro called ***ConflictsCheck***. In order to bind keyboard short cuts to a custom subroutine you have to use a Global Template, because you cannot bind to custom methods in COM Add-in.

#### **Solution**

You would create an AutoExec subroutine (or add to an existing AutoExec) in a Global Template that binds a keyboard short cut Ctrl+Shift+C to the subroutine ***ConflictsCheck***.

The following example binds Ctrl+Shift+C to the macro ***ConflictsCheck***.

```
Public Sub AutoExec()  
    CustomizationContext = NormalTemplate  
KeyBindings.Add _  
KeyCode:=BuildKeyCode(wdKeyC, wdKeyControl,  
wdKeyShift), _  
KeyCategory:=wdKeyCategoryCommand, _  
Command:="ConflictsCheck"  
End Sub
```

#### **Problems With Using Normal.dotm**

Using the Normal template to place a firm's (or vendor's) automation was necessary in Word 2.0 (circa 1991) because that was the only option. There were no Global templates or COM Add-in capabilities. Today, development best practices for Microsoft Word dictates DO NOT DO IT!

Managing code in the Normal template can turn into a nightmare for a firm for the following reasons:

1. When code changes occur, all users will need to be given a new Normal.dotm, overwriting their existing Normal.dotm.
2. If the vendor's software makes changes without replacing the normal template, the code is being changed in a template that is already loaded. This is not a good practice if the document that is opened is based on Normal.
3. For trouble shooting Microsoft Word problems, Normal.dotm may have to be renamed (will not load) or deleted (see KB article "[How to troubleshoot](#)")

[problems that occur when you start or use Word 2010, Word 2007, Word 2003, or Word 2002](http://support.microsoft.com/kb/921541)  
<http://support.microsoft.com/kb/921541>).

## **Summary**

I was prompted to write this paper in response to two things. First, I recently saw a presentation that was offered to a wide audience of legal technology leaders, staff, and trainers. The presentation erroneously stated that Microsoft Word's load order was COM Add-in, Normal.dotm, and then Global Templates. Second, I received a panicked call from a client, who was told by a vendor that all their code should be in Normal.dotm. I thought this was poor advice based on current best practices, as discussed above. We follow the approach that the Normal.dotm should be left for the user, and that any firm customizations and additional add-ons to Word should be either COM Add-ins or Global Templates.