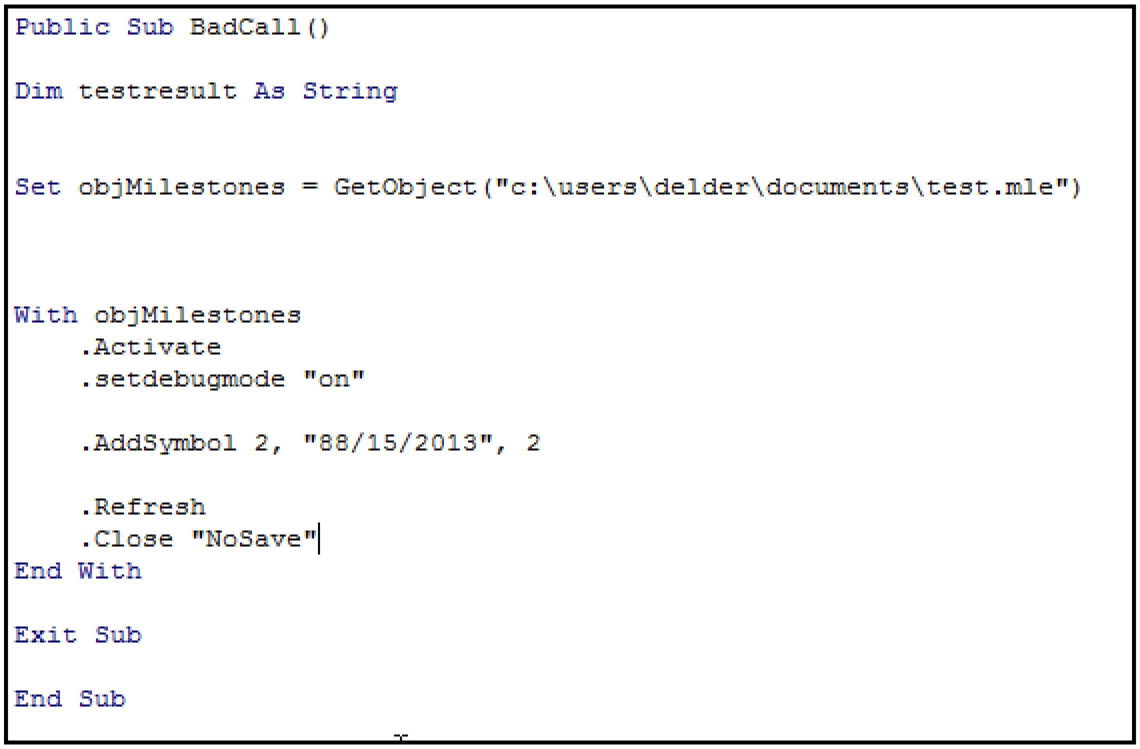
# Debugging Milestones Automation Applications on a Server (and Desktop)

## Desktop Debugging

Debugging Milestones Automation programs on your desktop is fairly easy, using a combination of error messages from your scripting language that helps you identify the line of code rejected by Milestones, and the optional Milestones Automation Debug window.

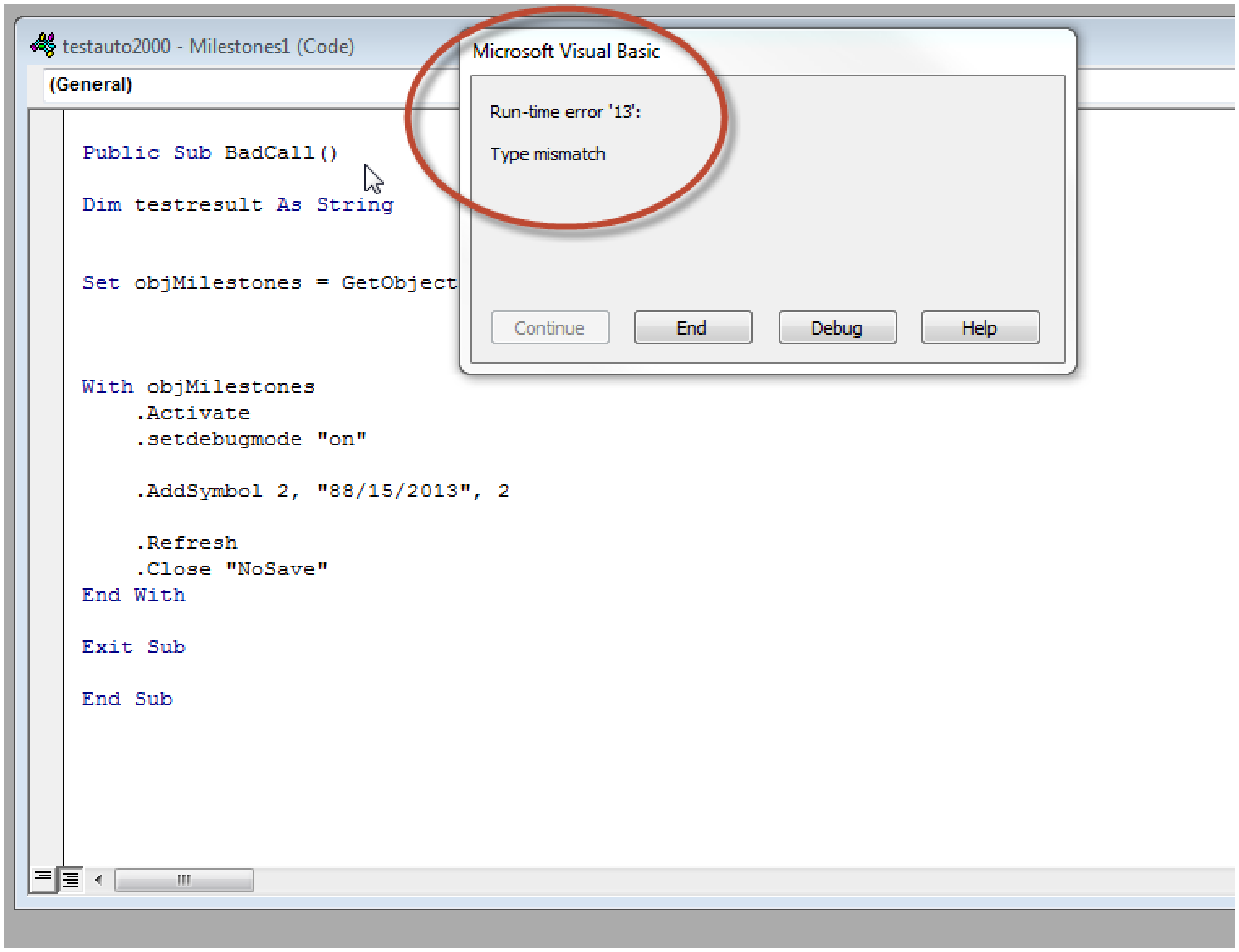
For example, the following code:



```
Public Sub BadCall()

Dim testresult As String


Set objMilestones = GetObject("c:\users\delder\documents\test.mle")



With objMilestones
    .Activate
    .setdebugmode "on"

    .AddSymbol 2, "88/15/2013", 2

    .Refresh
    .Close "NoSave"
End With

Exit Sub

End Sub
```

has an error in it in a call to Milestones.

When you run it under an Office program, such as Access or Excel, as a VBA script, you will get the following error screen:

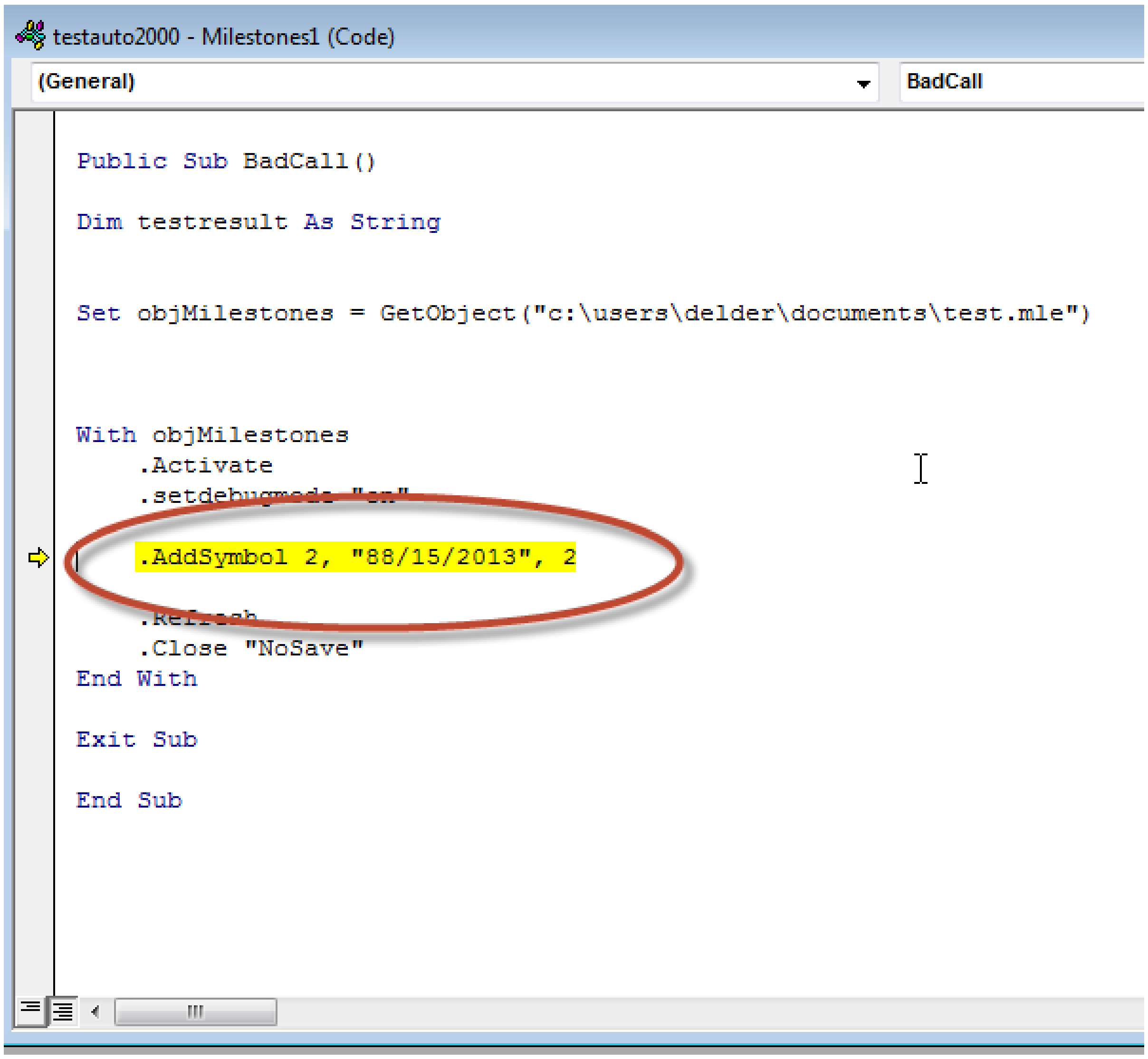


```
Run-time error '13':

Type mismatch
```

```
Public Sub BadCall()

Dim testresult As String


Set objMilestones = GetObject


With objMilestones
    .Activate
    .setdebugmode "on"

    .AddSymbol 2, "88/15/2013", 2

    .Refresh
    .Close "NoSave"
End With

Exit Sub

End Sub
```

which is somewhat limited.

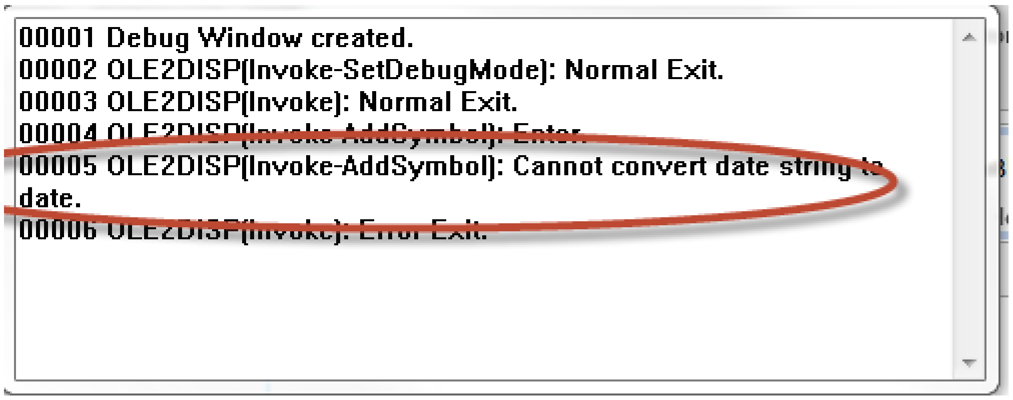
If you click on the Debug button, at least the line that caused the problem is highlighted:



```
Public Sub BadCall()

Dim testresult As String


Set objMilestones = GetObject("c:\users\delder\documents\test.mle")



With objMilestones
    .Activate
    .setdebugmode "on"

    .AddSymbol 2, "88/15/2013", 2

    .Refresh
    .Close "NoSave"
End With

Exit Sub

End Sub
```

But the actual error is still not identified.

Luckily, this program also made a call to the Milestones method “SetDebugMode”, so if we look at the Milestones Automation debug window, more information is found:

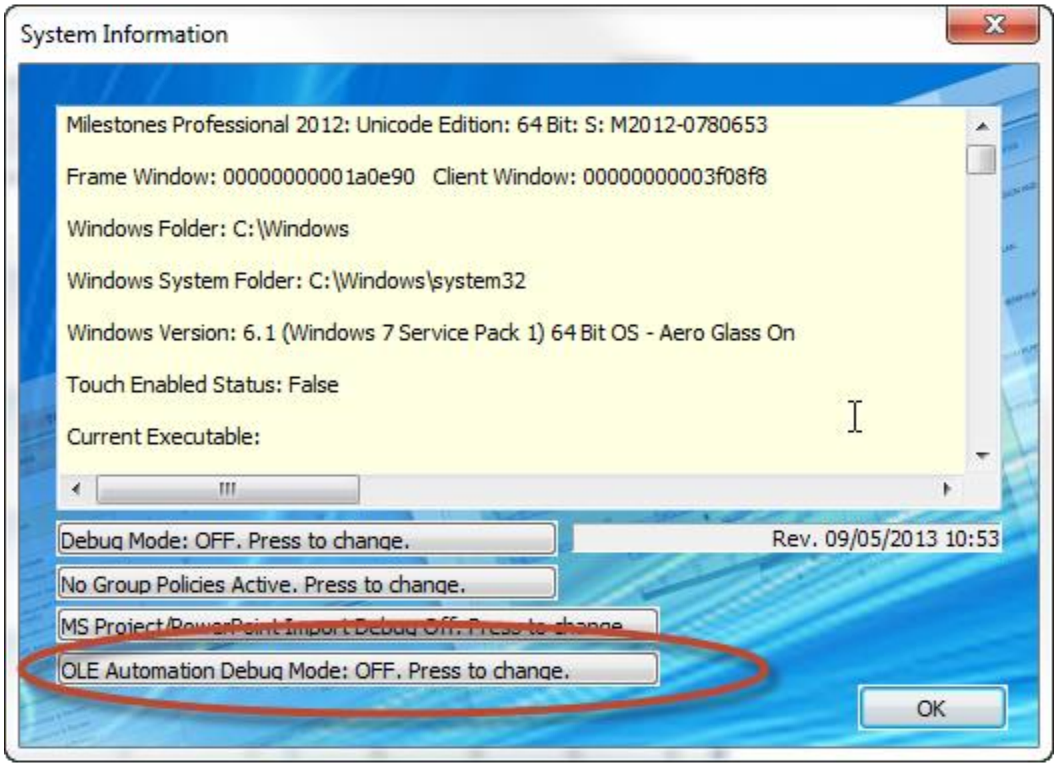


```
00001 Debug Window created.
00002 OLE2DISP(Invoke-SetDebugMode): Normal Exit.
00003 OLE2DISP(Invoke): Normal Exit.
00004 OLE2DISP(Invoke-AddSymbol): Enter.
00005 OLE2DISP(Invoke-AddSymbol): Cannot convert date string to
date.
00006 OLE2DISP(Invoke): Error Exit.
```

Which will eventually lead us to conclude that “88” is not a proper month in a date string.

The automation interface allows only limited information to be sent back via an error return, just a failure notification and a very limited set of failure codes, thus most of the time, the Milestones interface will issue a “Type Mismatch” error for almost all errors.

This is why the Automation debug window was added, and which can be turned on or off via the SetDebugMode method or via the button in the Milestones | help | View System Info screen:



```
Milestones Professional 2012: Unicode Edition: 64 Bit: S: M2012-0780653

Frame Window: 0000000001a0e90   Client Window: 00000000003f08f8

Windows Folder: C:\Windows

Windows System Folder: C:\Windows\system32

Windows Version: 6.1 (Windows 7 Service Pack 1) 64 Bit OS - Aero Glass On

Touch Enabled Status: False

Current Executable:
```

```
Debug Mode: OFF. Press to change.          Rev. 09/05/2013 10:53
No Group Policies Active. Press to change.
MS Project/PowerPoint Import Debug Off. Press to change
OLE Automation Debug Mode: OFF. Press to change.
```

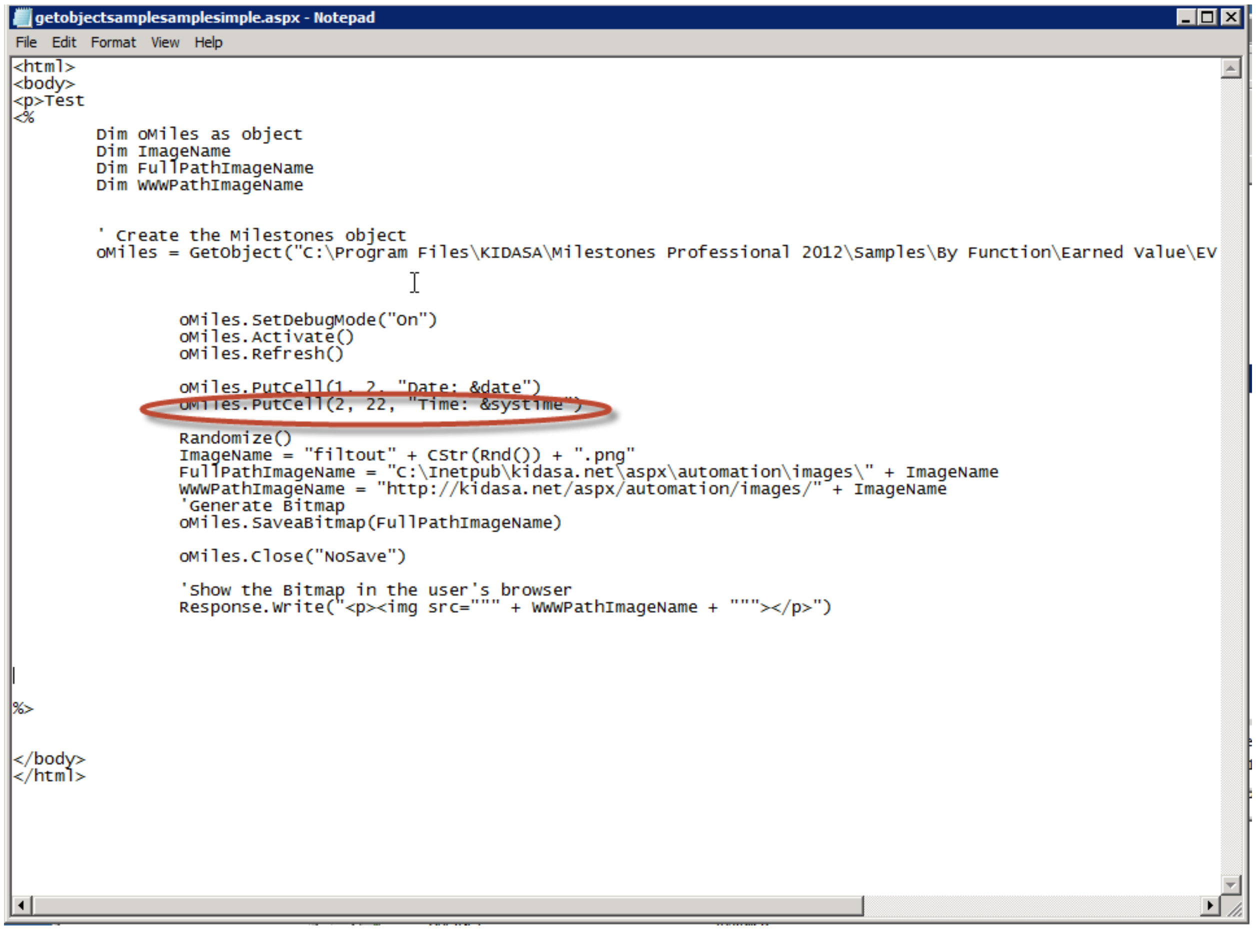
When you press this button, the Automation debug window appears, and any future automation calls will show up in the debug window. This can be useful for already deployed desktop applications that are failing, and have no way of turning on debug mode by themselves.

## Server Debugging

Since, when you are running Milestones on a server, most likely under IIS, the Milestones GUI is not available, so there is no way to directly turn on debugging via the System Info screen, or even to see the debug window, since it is running under a different process and has no desktop window anyway.

While it is possible that you have a development environment that lets you remotely debug server based applications from within your development environment, in many cases that sort of remote debugging is not possible due to security or other issues. The rest of this paper discusses how to debug a server based application in the situation where the running code cannot be debugged from within the development environment.

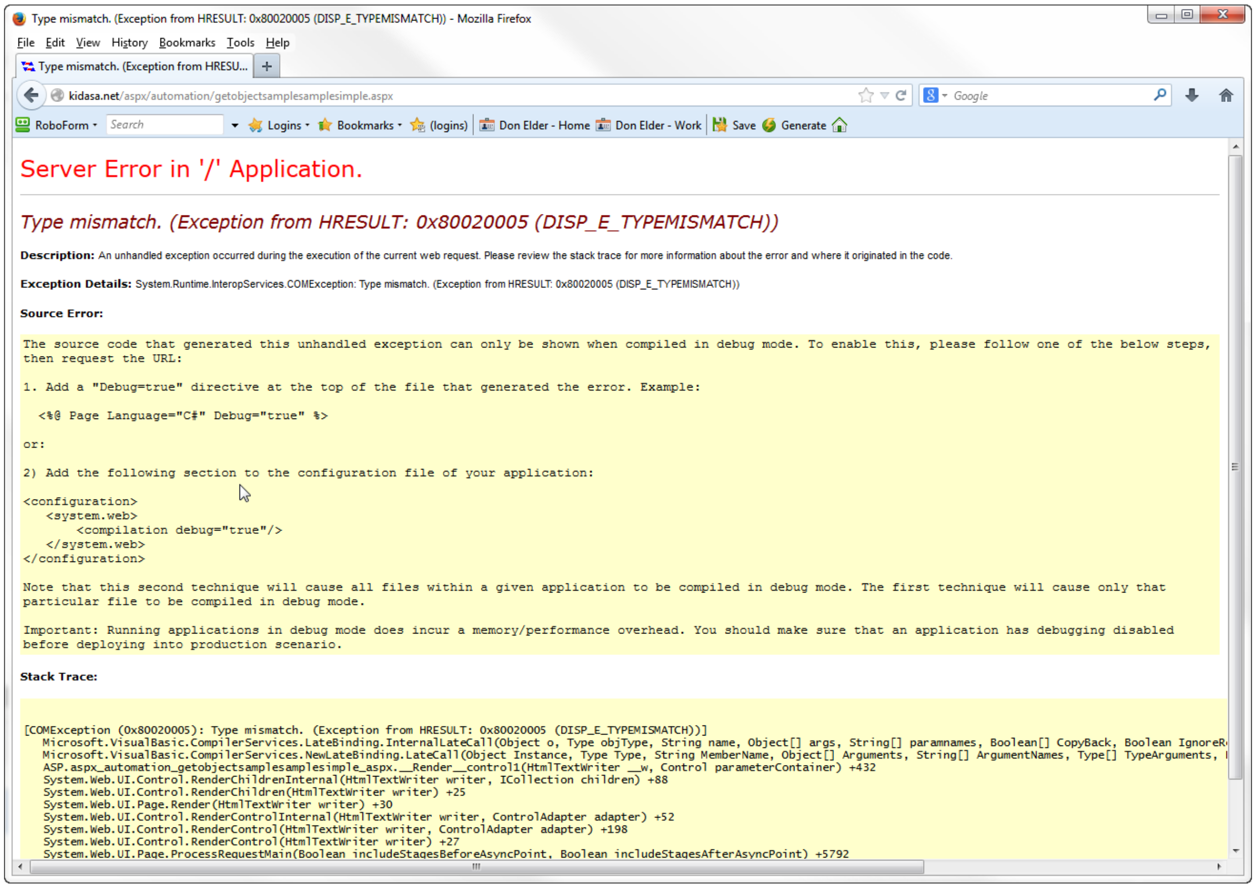
So, if we have the following program on a Windows 2008 server that we are running under ASP.NET:


getobjectsamplesamplesimple.aspx - Notepad

File Edit Format View Help

```
<html>
<body>
<p>Test
<%
        Dim oMiles as object
        Dim ImageName
        Dim FullPathImageName
        Dim WWwPathImageName


        ' Create the Milestones object
        oMiles = GetObject("C:\Program Files\KIDASA\Milestones Professional 2012\Samples\By Function\Earned Value\EV


                oMiles.SetDebugMode("On")
                oMiles.Activate()
                oMiles.Refresh()

                oMiles.PutCell(1, 2, "Date: &date")
                oMiles.PutCell(2, 22, "Time: &systime")

                Randomize()
                ImageName = "filtout" + CStr(Rnd()) + ".png"
                FullPathImageName = "C:\Inetpub\kidasa.net\aspx\automation\images\" + ImageName
                WWwPathImageName = "http://kidasa.net/aspx/automation/images/" + ImageName
                'Generate Bitmap
                oMiles.SaveaBitmap(FullPathImageName)

                oMiles.Close("NoSave")

                'Show the Bitmap in the user's browser
                Response.Write("<p><img src=""" + WWwPathImageName + """></p>")




%>


</body>
</html>
```

Note that the second PutCell has an error. However, if we simply run this from a browser, the default error page again tells us that we have a Type Mismatch error:


Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH)) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Type mismatch. (Exception from HRESU...
kidasa.net/aspx/automation/getobjectsamplesamplesimple.aspx
Google
RoboForm Search Logins Bookmarks (logins) Don Elder - Home Don Elder - Work Save Generate

Server Error in '/' Application.

*Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))*

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Runtime.InteropServices.COMException: Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))

**Source Error:**

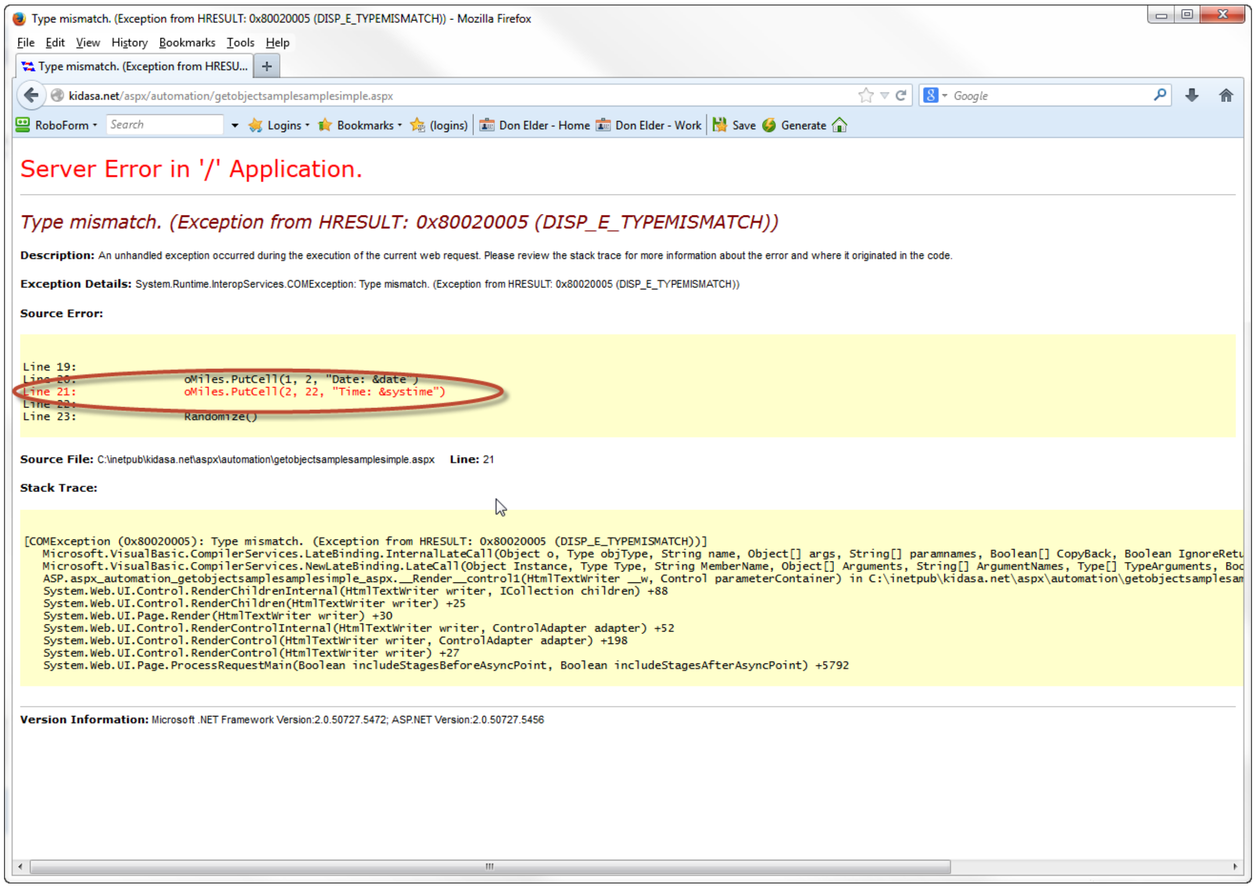
```
The source code that generated this unhandled exception can only be shown when compiled in debug mode. To enable this, please follow one of the below steps,
then request the URL:

1. Add a "Debug=true" directive at the top of the file that generated the error. Example:

  <%@ Page Language="C#" Debug="true" %>

or:

2) Add the following section to the configuration file of your application:

<configuration>
   <system.web>
       <compilation debug="true"/>
   </system.web>
</configuration>

Note that this second technique will cause all files within a given application to be compiled in debug mode. The first technique will cause only that
particular file to be compiled in debug mode.

Important: Running applications in debug mode does incur a memory/performance overhead. You should make sure that an application has debugging disabled
before deploying into production scenario.
```

**Stack Trace:**

```
[COMException (0x80020005): Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))]
   Microsoft.VisualBasic.CompilerServices.LateBinding.InternalLateCall(Object o, Type objType, String name, Object[] args, String[] paramnames, Boolean[] CopyBack, Boolean IgnoreR
   Microsoft.VisualBasic.CompilerServices.NewLateBinding.LateCall(Object Instance, Type Type, String MemberName, Object[] Arguments, String[] ArgumentNames, Type[] TypeArguments, 
   ASP.aspx_automation_getobjectsamplesamplesimple_aspx.__Render__control1(HtmlTextWriter __w, Control parameterContainer) +432
   System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children) +88
   System.Web.UI.Control.RenderChildren(HtmlTextWriter writer) +25
   System.Web.UI.Page.Render(HtmlTextWriter writer) +30
   System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter) +52
   System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +198
   System.Web.UI.Control.RenderControl(HtmlTextWriter writer) +27
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +5792
```

on a “Late Binding” call, but no line number or actual method that failed is shown in the Stack Trace. The page on this server does suggest though that we add <compilation debug="true"/> to the web.config file.

If you are running under ASP.NET, you can use the web.config file to at least get some line information. So if add the compilation debug=true to the web.config file we get a better error screen:


Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH)) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Type mismatch. (Exception from HRESU...
kidasa.net/aspx/automation/getobjectsamplesamplesimple.aspx
Google
RoboForm
Search
Logins
Bookmarks
(logins)
Don Elder - Home
Don Elder - Work
Save
Generate

Server Error in '/' Application.

*Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))*

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Runtime.InteropServices.COMException: Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))

**Source Error:**

```
Line 19:
Line 20:                oMiles.PutCell(1, 2, "Date: &date")
Line 21:                oMiles.PutCell(2, 22, "Time: &systime")
Line 22:
Line 23:                Randomize()
```

**Source File:** C:\inetpub\kidasa.net\aspx\automation\getobjectsamplesamplesimple.aspx **Line:** 21

**Stack Trace:**

```
[COMException (0x80020005): Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))]
   Microsoft.VisualBasic.CompilerServices.LateBinding.InternalLateCall(Object o, Type objType, String name, Object[] args, String[] paramnames, Boolean[] CopyBack, Boolean IgnoreRetu
   Microsoft.VisualBasic.CompilerServices.NewLateBinding.LateCall(Object Instance, Type Type, String MemberName, Object[] Arguments, String[] ArgumentNames, Type[] TypeArguments, Boo
   ASP.aspx_automation_getobjectsamplesamplesimple_aspx.__Render__control1(HtmlTextWriter __w, Control parameterContainer) in C:\inetpub\kidasa.net\aspx\automation\getobjectsamplesam
   System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children) +88
   System.Web.UI.Control.RenderChildren(HtmlTextWriter writer) +25
   System.Web.UI.Page.Render(HtmlTextWriter writer) +30
   System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter) +52
   System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +198
   System.Web.UI.Control.RenderControl(HtmlTextWriter writer) +27
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +5792
```

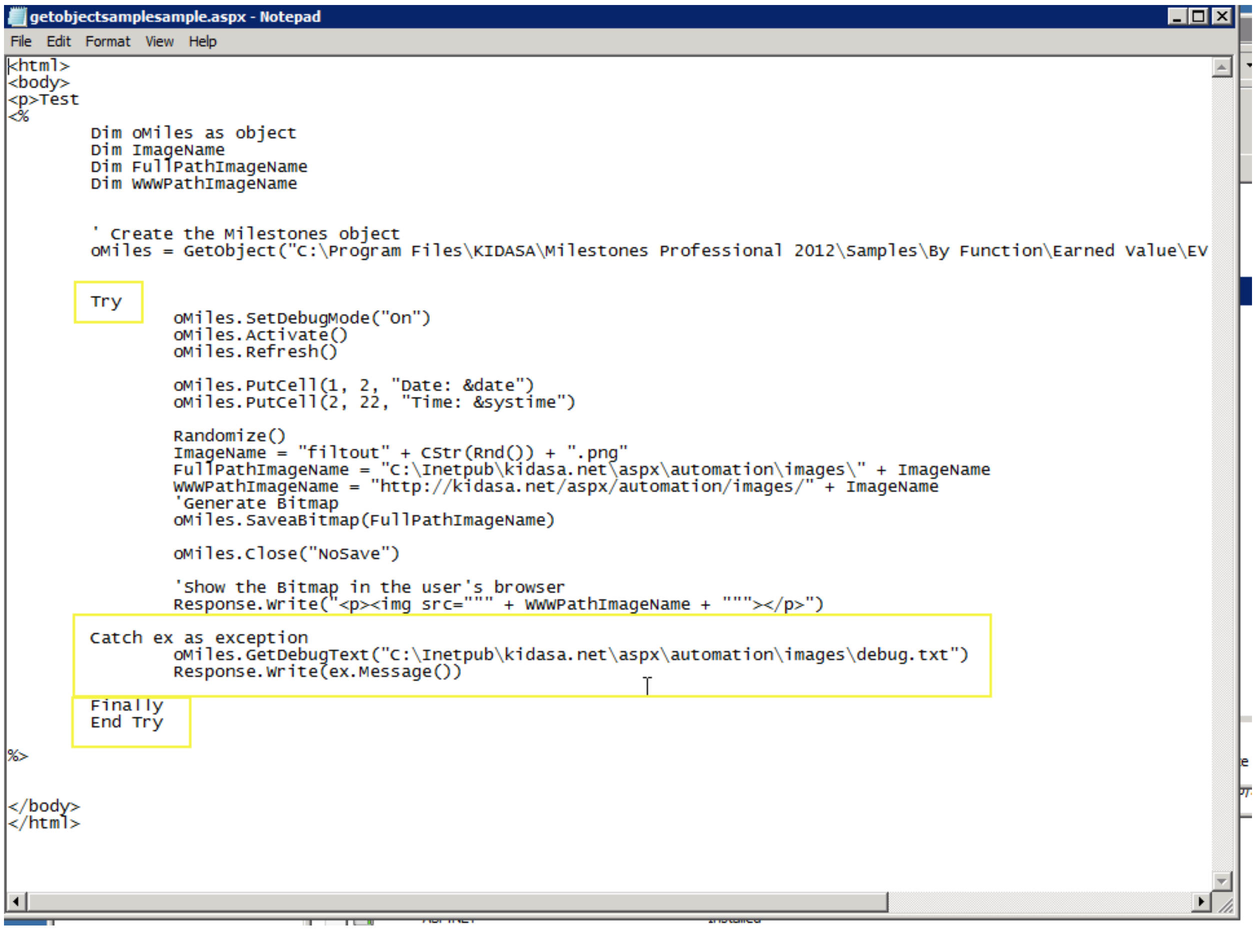
**Version Information:** Microsoft .NET Framework Version:2.0.50727.5472; ASP.NET Version:2.0.50727.5456

Now at least the failing line number is highlighted and we know the error was in a call to PutCell.

If that still is not enough information to determine the problem, it is possible to get the same Milestones automation debug window information by using a Try/Catch sequence.

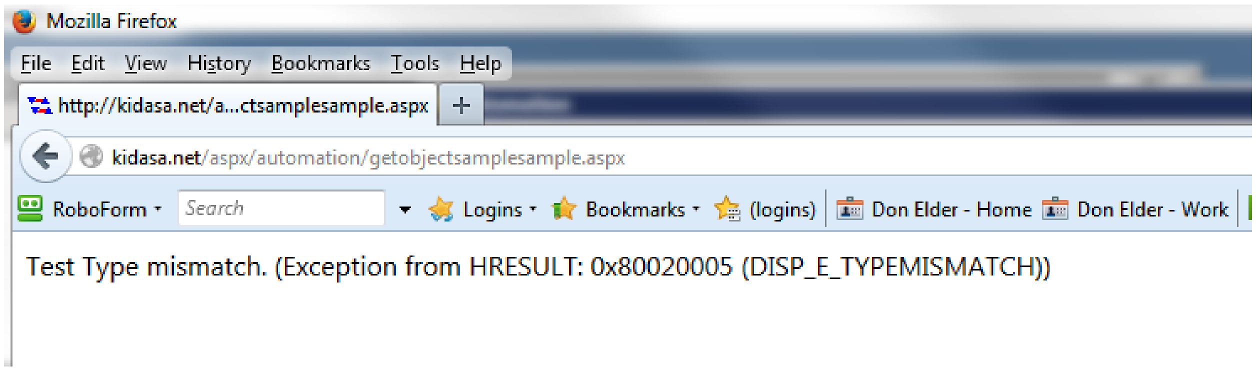
There is a method in Milestones Automation called “GetDebugText”, which when called, will dump the last 4000 characters in the debug window to a text file, so even though you cannot see the debug window, you can get its text.

So, if we enhance our sample application to have a Try/Catch sequence it will look like this:



```
<html>
<body>
<p>Test
<%
        Dim oMiles as object
        Dim ImageName
        Dim FullPathImageName
        Dim WWWPathImageName


        ' Create the Milestones object
        oMiles = GetObject("C:\Program Files\KIDASA\Milestones Professional 2012\Samples\By Function\Earned Value\EV

        Try

                oMiles.SetDebugMode("On")
                oMiles.Activate()
                oMiles.Refresh()

                oMiles.PutCell(1, 2, "Date: &date")
                oMiles.PutCell(2, 22, "Time: &systime")

                Randomize()
                ImageName = "filtout" + CStr(Rnd()) + ".png"
                FullPathImageName = "C:\Inetpub\kidasa.net\aspx\automation\images\" + ImageName
                WWWPathImageName = "http://kidasa.net/aspx/automation/images/" + ImageName
                'Generate Bitmap
                oMiles.SaveaBitmap(FullPathImageName)

                oMiles.Close("NoSave")

                'Show the Bitmap in the user's browser
                Response.Write("<p><img src=""" + WWWPathImageName + """></p>")

        Catch ex as exception
                oMiles.GetDebugText("C:\Inetpub\kidasa.net\aspx\automation\images\debug.txt")
                Response.Write(ex.Message())

        Finally
        End Try

%>


</body>
</html>
```

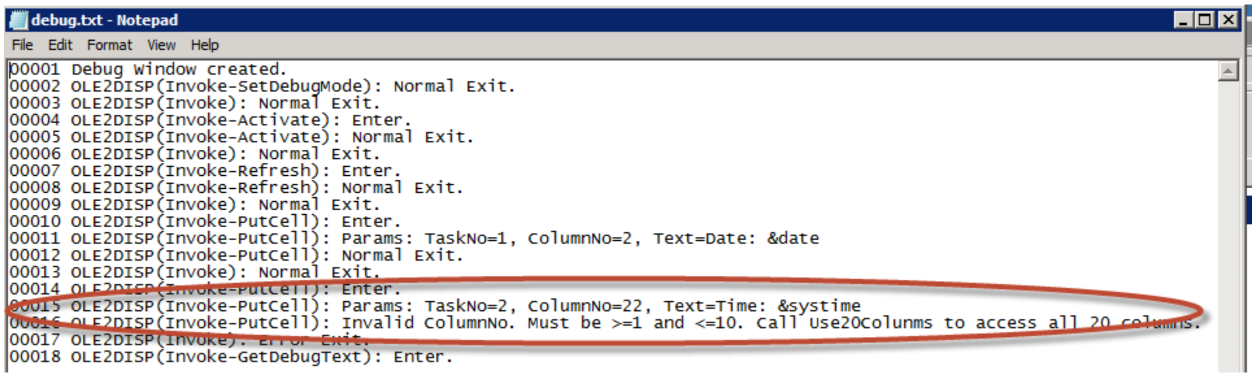
Note that we have added the code boxed in yellow. Now when we run this we get the following error screen:



Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://kidasa.net/a...ctsamplesample.aspx
kidasa.net/aspx/automation/getobjectsamplesample.aspx
RoboForm
Search
Logins
Bookmarks
(logins)
Don Elder - Home
Don Elder - Work

Test Type mismatch. (Exception from HRESULT: 0x80020005 (DISP_E_TYPEMISMATCH))

which is not too useful since it just writes out the error code, but if we look at the debug.txt file that the call to GetDebugText in the Catch section created:


debug.txt - Notepad

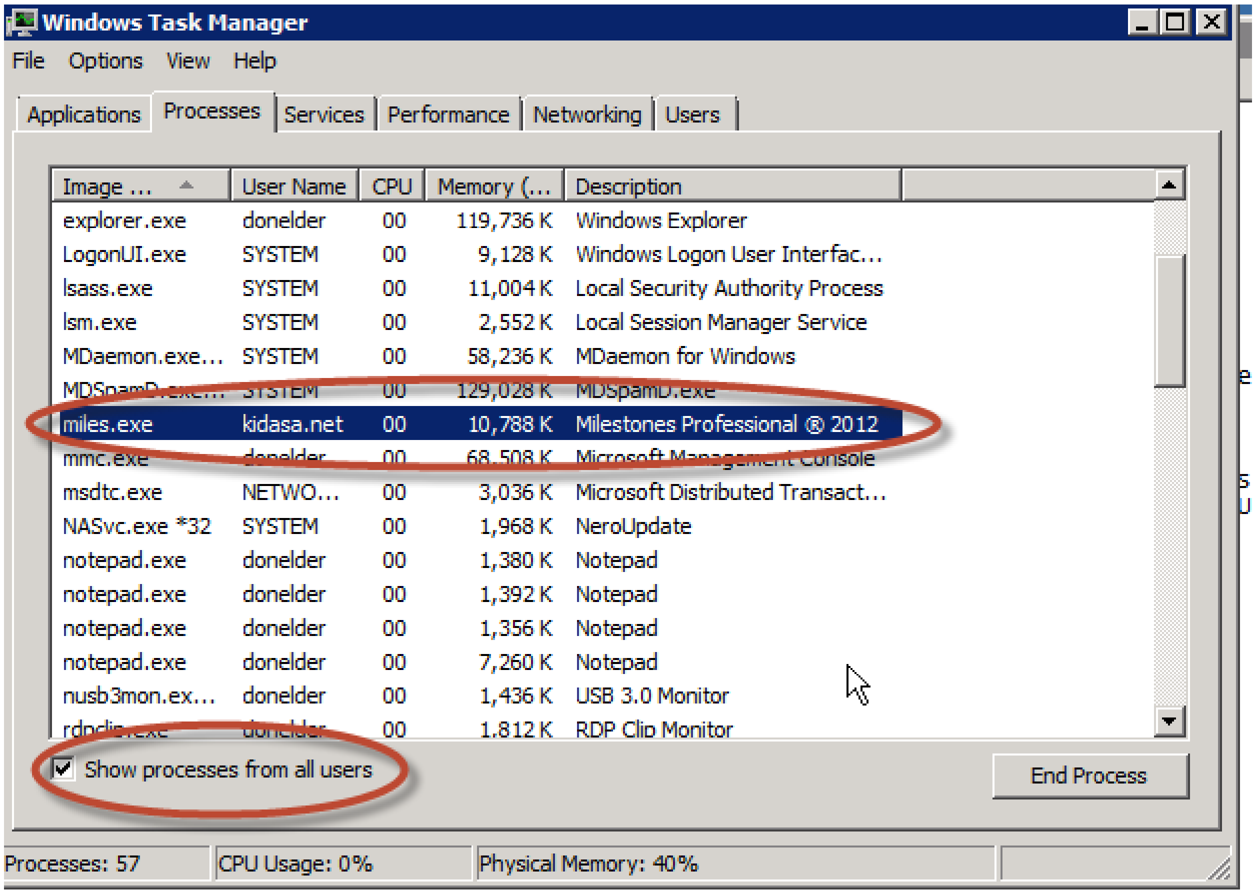
```
00001 Debug window created.
00002 OLE2DISP(Invoke-SetDebugMode): Normal Exit.
00003 OLE2DISP(Invoke): Normal Exit.
00004 OLE2DISP(Invoke-Activate): Enter.
00005 OLE2DISP(Invoke-Activate): Normal Exit.
00006 OLE2DISP(Invoke): Normal Exit.
00007 OLE2DISP(Invoke-Refresh): Enter.
00008 OLE2DISP(Invoke-Refresh): Normal Exit.
00009 OLE2DISP(Invoke): Normal Exit.
00010 OLE2DISP(Invoke-PutCell): Enter.
00011 OLE2DISP(Invoke-PutCell): Params: TaskNo=1, ColumnNo=2, Text=Date: &date
00012 OLE2DISP(Invoke-PutCell): Normal Exit.
00013 OLE2DISP(Invoke): Normal Exit.
00014 OLE2DISP(Invoke-PutCell): Enter.
00015 OLE2DISP(Invoke-PutCell): Params: TaskNo=2, ColumnNo=22, Text=Time: &systime
00016 OLE2DISP(Invoke-PutCell): Invalid ColumnNo. Must be >=1 and <=10. Call Use20Colunms to access all 20 columns.
00017 OLE2DISP(Invoke): Error Exit.
00018 OLE2DISP(Invoke-GetDebugText): Enter.
```

We can see that the second call to PutCell had an invalid column number (22).

**Dangling miles.exe processes**

If you have a situation where your Milestones automation application has terminated unexpectedly and the Milestones Close method was never called, you most likely have left some miles.exe processes running.

You should always check to see under Task Manager to see if any are running:


Windows Task Manager

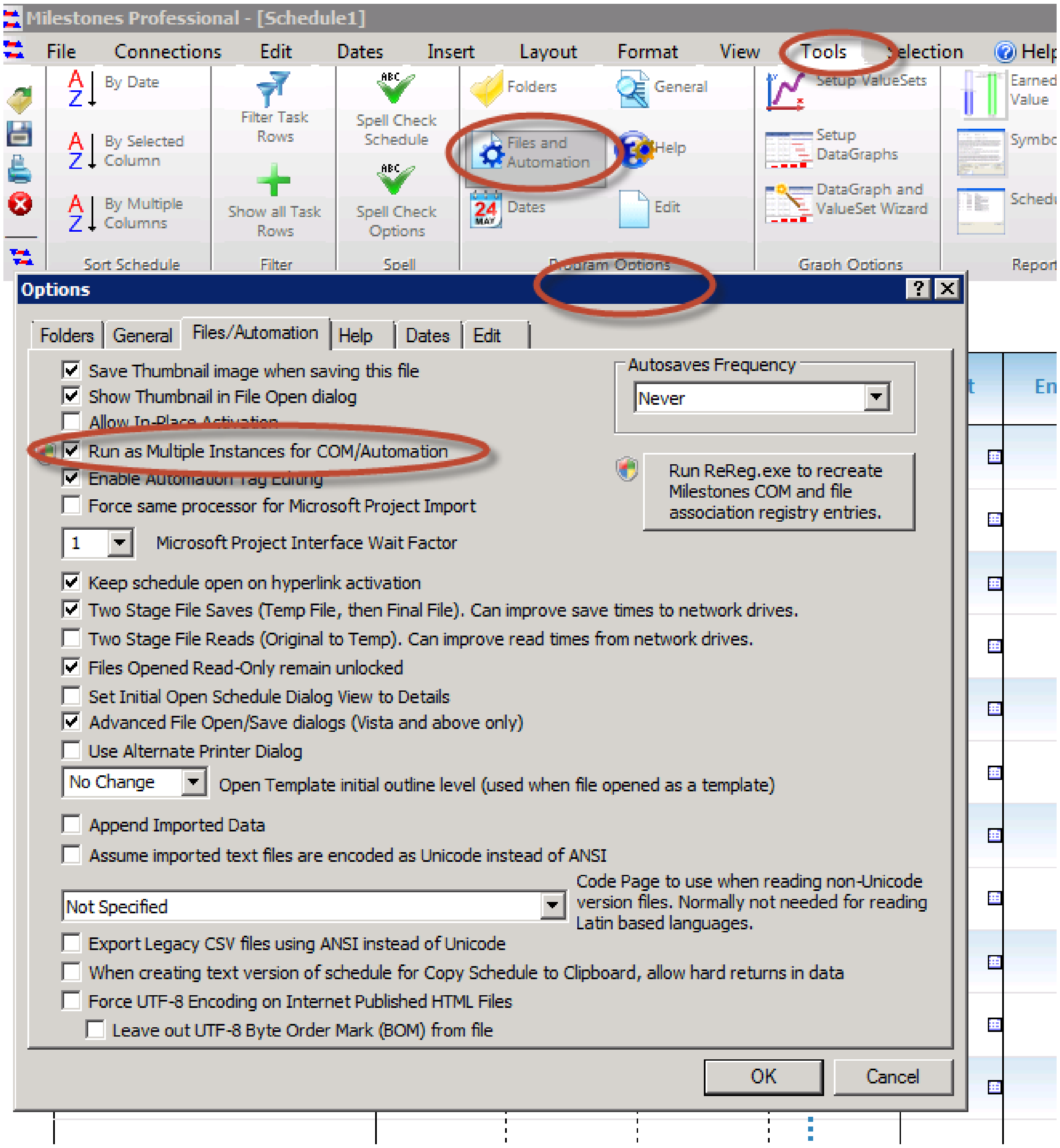
| Image ... | User Name | CPU | Memory (... | Description |
|---|---|---|---|---|
| explorer.exe | donelder | 00 | 119,736 K | Windows Explorer |
| LogonUI.exe | SYSTEM | 00 | 9,128 K | Windows Logon User Interfac... |
| lsass.exe | SYSTEM | 00 | 11,004 K | Local Security Authority Process |
| lsm.exe | SYSTEM | 00 | 2,552 K | Local Session Manager Service |
| MDaemon.exe... | SYSTEM | 00 | 58,236 K | MDaemon for Windows |
| MDSpamD.exe... | SYSTEM | 00 | 129,028 K | MDSpamD.exe |
| miles.exe | kidasa.net | 00 | 10,788 K | Milestones Professional ® 2012 |
| mmc.exe | donelder | 00 | 68,508 K | Microsoft Management Console |
| msdtc.exe | NETWO... | 00 | 3,036 K | Microsoft Distributed Transact... |
| NASvc.exe *32 | SYSTEM | 00 | 1,968 K | NeroUpdate |
| notepad.exe | donelder | 00 | 1,380 K | Notepad |
| notepad.exe | donelder | 00 | 1,392 K | Notepad |
| notepad.exe | donelder | 00 | 1,356 K | Notepad |
| notepad.exe | donelder | 00 | 7,260 K | Notepad |
| nusb3mon.ex... | donelder | 00 | 1,436 K | USB 3.0 Monitor |
| rdpclip.exe | donelder | 00 | 1,812 K | RDP Clip Monitor |

[✓] Show processes from all users

Processes: 57 CPU Usage: 0% Physical Memory: 40%

Be sure to click on the all users checkbox, since miles.exe will be running under a web site ID and not yours.

Normally a dangling process does not matter that much, unless you forgot to have the "Use Multiple Instances for COM/Automation checked in Milestones Tools | Program Options | Files and Automation screen:



Options

Folders | General | Files/Automation | Help | Dates | Edit

- [✓] Save Thumbnail image when saving this file
- [✓] Show Thumbnail in File Open dialog
- [ ] Allow In-Place Activation
- [✓] Run as Multiple Instances for COM/Automation
- [✓] Enable Automation Tag Editing
- [ ] Force same processor for Microsoft Project Import

Autosaves Frequency: Never

Run ReReg.exe to recreate Milestones COM and file association registry entries.

1 Microsoft Project Interface Wait Factor

- [✓] Keep schedule open on hyperlink activation
- [✓] Two Stage File Saves (Temp File, then Final File). Can improve save times to network drives.
- [ ] Two Stage File Reads (Original to Temp). Can improve read times from network drives.
- [✓] Files Opened Read-Only remain unlocked
- [ ] Set Initial Open Schedule Dialog View to Details
- [✓] Advanced File Open/Save dialogs (Vista and above only)
- [ ] Use Alternate Printer Dialog

No Change Open Template initial outline level (used when file opened as a template)

- [ ] Append Imported Data
- [ ] Assume imported text files are encoded as Unicode instead of ANSI

Not Specified Code Page to use when reading non-Unicode version files. Normally not needed for reading Latin based languages.

- [ ] Export Legacy CSV files using ANSI instead of Unicode
- [ ] When creating text version of schedule for Copy Schedule to Clipboard, allow hard returns in data
- [ ] Force UTF-8 Encoding on Internet Published HTML Files
  - [ ] Leave out UTF-8 Byte Order Mark (BOM) from file

OK Cancel

If this option was not checked, then a possibly unstable copy of miles.exe will attempt to act as the automation server for any future Milestones objects. Also, if an open Milestones file is stuck in this dangling process, then other users may not be able to update it, or since it will be in the Running Object Table it may not be possible to open another copy of it via another automation session.

Issuing a .Close without the “NoSave” parameter can leave a Milestones file open, as the GUI may be displaying the “Do You Want to Save Your File” dialog in a hidden window with no way to get a response.

**Late Binding**

Earlier in this document a reference was made to “Late Binding”. The alternative to “Late Binding” is “Early Binding”. Binding has to do with how a program references the API of another program. Early binding allows the references to be “locked up” at compile time and no extra work is required to figure out API calls when the program actually runs. Late Binding leaves the API references open, and unchecked, until the program actually runs. Milestones uses “Late Binding” and that is why there is no reference file to aid you in typing in the calls to the various calls available via the Milestones API.

For more info on the technical details of early vs. late binding, please see this Microsoft document: http://support.microsoft.com/kb/245115

Milestones is an OLE Automation (COM) server designed to run as an independent executable. Other applications can instantiate a Milestones object and then control it via calls to the Milestones API.

The nice thing about a “Late Bound” program is that it is version independent, i.e. even though the internals of Milestones get updated with each new version, all prior automation programs that use Milestones continue to function even though the underlying application has been updated, i.e. you are not forced to recompile your automation app with the latest reference file or worry that your app will no longer function just because you have updated Milestones to a new version.

Since Milestones is distributed as an EXE file and not a DLL you don’t have to worry about whether or not the Milestones exe is 32 or 64 bit or whether the calling app’s bitness matches.