



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
TestStand, version 3.6.197.29569
Begin @ 05/31/2014 17:38:14.316 (05/31/2014 22:38:14.316 UTC)

Each of the next four screens displays all 18 overloads of a method.
The program pauses to allow you to capture each screen.
Press the Enter key when you are ready to see the first screen.
```

Figure 1 is the initial screen displayed by version **3.6.197.29569** of the test program for class library `WizardWrx.ConsoleAppAids`.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Overload 1: True
Overload 2: E
Overload 3: FGH
Overload 4: 1.25
Overload 5: 3.333333
Overload 6: 3.3333
Overload 7: -255
Overload 8: -125000
Overload 9: Stuff 1
Overload 10: This is it.
Overload 11: 512
Overload 12: 250000
Overload 13: object arg0: object arg0 = Stuff 1
Overload 14: params object [ ] arg: params object [ ] arg = Stuff 1
Overload 15: GH
Overload 16: object arg0, object arg1: object arg0 = Stuff 1, object arg1 =
Stuff 2
Overload 17: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3
Overload 18: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3, object arg3 = Stuff 4
Static MessageInColor.WriteLine() Tests End
Capture this screen, then press the ENTER key.
```

Figure 2 is the static `MessageInColor.WriteLine` test set, which demonstrates all 18 of its overloads.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...

Overload 1: True
Overload 2: E
Overload 3: FGH
Overload 4: 1.25
Overload 5: 3.333333
Overload 6: 3.3333
Overload 7: -255
Overload 8: -125000
Overload 9: Stuff 1
Overload 10: This is it.
Overload 11: 512
Overload 12: 250000
Overload 13: object arg0: object arg0 = Stuff 1
Overload 14: params object [ ] arg: params object [ ] arg = Stuff 1
Overload 15: GH
Overload 16: object arg0, object arg1: object arg0 = Stuff 1, object arg1 =
Stuff 2
Overload 17: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3
Overload 18: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3, object arg3 = Stuff 4
Static MessageInColor.Write() Tests End
Capture this screen, then press the ENTER key.
```

Figure 3 is the static `MessageInColor.Write` test set, which demonstrates all 18 of its overloads.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...

Overload 1: True
Overload 2: E
Overload 3: FGH
Overload 4: 1.25
Overload 5: 3.333333
Overload 6: 3.3333
Overload 7: -255
Overload 8: -125000
Overload 9: Stuff 1
Overload 10: This is it.
Overload 11: 512
Overload 12: 250000
Overload 13: object arg0: object arg0 = Stuff 1
Overload 14: params object [ ] arg: params object [ ] arg = Stuff 1
Overload 15: GH
Overload 16: object arg0, object arg1: object arg0 = stuff 1, object arg1 =
stuff 2
Overload 17: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3
Overload 18: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3, object arg3 = Stuff 4
Instance MessageInColor.ColorWriteLine() Tests End
Capture this screen, then press the ENTER key.
```

Figure 4 demonstrates all 18 overloads of the `MessageInColor.ColorWriteLine` method of an instance of the `MessageInColor` class.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Overload 1: True
Overload 2: E
Overload 3: FGH
Overload 4: 1.25
Overload 5: 3.333333
Overload 6: 3.3333
Overload 7: -255
Overload 8: -125000
Overload 9: Stuff 1
Overload 10: This is it.
Overload 11: 512
Overload 12: 250000
Overload 13: object arg0: object arg0 = Stuff 1
Overload 14: params object [ ] arg: params object [ ] arg = Stuff 1
Overload 15: GH
Overload 16: object arg0, object arg1: object arg0 = Stuff 1, object arg1 =
Stuff 2
Overload 17: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3
Overload 18: object arg0 = Stuff 1, object arg1 = Stuff 2, object arg2 = Stu
ff 3, object arg3 = Stuff 4
Instance MessageInColor.ColorWrite() Tests End
Capture this screen, then press the ENTER key.
```

Figure 5 demonstrates all 18 overloads of the `MessageInColor.ColorWrite` method of an instance of the `MessageInColor` class.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 1 of 5 End

    An exception was caught in routine ExerciseColorExceptionReporting:
        Message = This error message appears in the default text
on the default background.
        Exception Method = Void ExerciseColorExceptionReporting()
        Exception Source = TestStand

Stack Trace Begin:
    at TestStand.Program.ExerciseColorExceptionReporting() in C:/Documents and Se
ttings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAid
s/TestStand/Program.cs:line 491
End of Stack Trace

    There are no further exceptions to report.

Error reporting test 1 of 5 End

Press the ENTER key to display the next example.
█
```

Figure 6 is the first of five examples of the deprecated static `WizardWrx.ConsoleAppAids.ExceptionReporting.ReportException` method, which shall remain, at least for a short while, for the benefit of `NBReplT.exe`, the only production program that has calls to written into it.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 2 of 5 End
    An exception was caught in routine ExerciseColorExceptionReporting:
    Message      = This error message appears in white text on Red background.
    Exception Method = Void ExerciseColorExceptionReporting()
    Exception Source = TestStand
Stack Trace Begin:
    at TestStand.Program.ExerciseColorExceptionReporting() in C:\Documents and Settings\DAG\My Documents\Visual Studio 2010\Projects\WizardWrx_Libs\ConsoleAppAids\TestStand\Program.cs:line 491
End of Stack Trace
    There are no further exceptions to report.
Error reporting test 2 of 5 End
Press the ENTER key to display the next example.
```

Figure 7 is the second of five examples of the deprecated static method `WizardWrx.ConsoleAppAids.ExceptionReporting.ReportException`, using white text on a red background, which is way over the top.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 3 of 5 End
    An exception was caught in routine ExerciseColorExceptionReporting:
    Message      = This error message appears in Yellow text on Red background.
    Exception Method = Void ExerciseColorExceptionReporting()
    Exception Source = TestStand
Stack Trace Begin:
    at TestStand.Program.ExerciseColorExceptionReporting() in C:\Documents and Settings\DAG\My Documents\Visual Studio 2010\Projects\WizardWrx_Libs\ConsoleAppAids\TestStand\Program.cs:line 491
End of Stack Trace
    There are no further exceptions to report.
Error reporting test 3 of 5 End
Press the ENTER key to display the next example.
```

Figure 8 is the third of the five examples of the deprecated static method `WizardWrx.ConsoleAppAids.ExceptionReporting.ReportException`, rendered in yellow on red, which is only a tad better than the white on red shown in **Figure 7**.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 4 of 5 End
    An exception was caught in routine ExerciseColorExceptionReporting:
    Message           = This error message appears in white text on DarkRed background.
    Exception Method = Void ExerciseColorExceptionReporting()
    Exception Source = TestStand
Stack Trace Begin:
    at TestStand.Program.ExerciseColorExceptionReporting() in C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/Program.cs:line 491
End of Stack Trace
    There are no further exceptions to report.
Error reporting test 4 of 5 End
Press the ENTER key to display the next example.
```

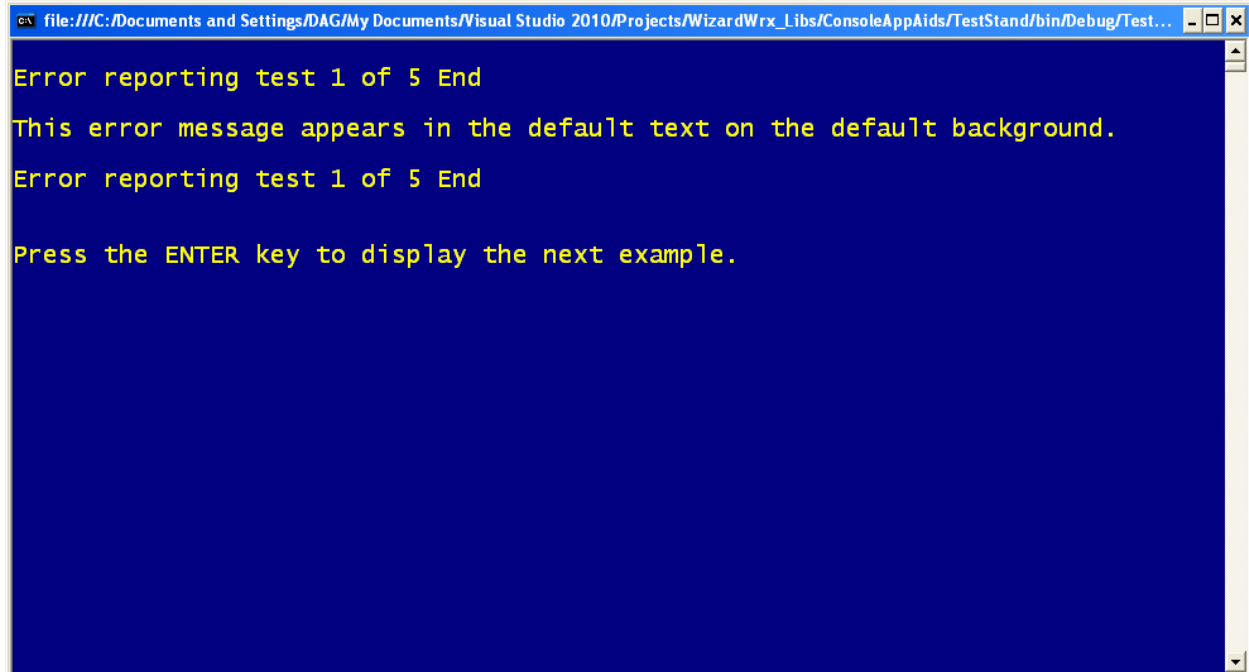
Figure 9 is the fourth of the five examples of the deprecated static method `WizardWrx.ConsoleAppAids.ExceptionReporting.ReportException`, rendered in the foreground color shown in **Figure 7** on a much more appropriate dark red background.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 5 of 5 End
    An exception was caught in routine ExerciseColorExceptionReporting:
    Message           = This error message appears in Yellow text on DarkRed background.
    Exception Method = Void ExerciseColorExceptionReporting()
    Exception Source = TestStand
Stack Trace Begin:
    at TestStand.Program.ExerciseColorExceptionReporting() in C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/Program.cs:line 491
End of Stack Trace
    There are no further exceptions to report.
Error reporting test 5 of 5 End
Press the ENTER key to display the next example.
```

Figure 10 is the fifth of the five examples of the deprecated static method `WizardWrx.ConsoleAppAids.ExceptionReporting.ReportException`, rendered in the foreground color shown in **Figure 8** on a much more appropriate dark red background.

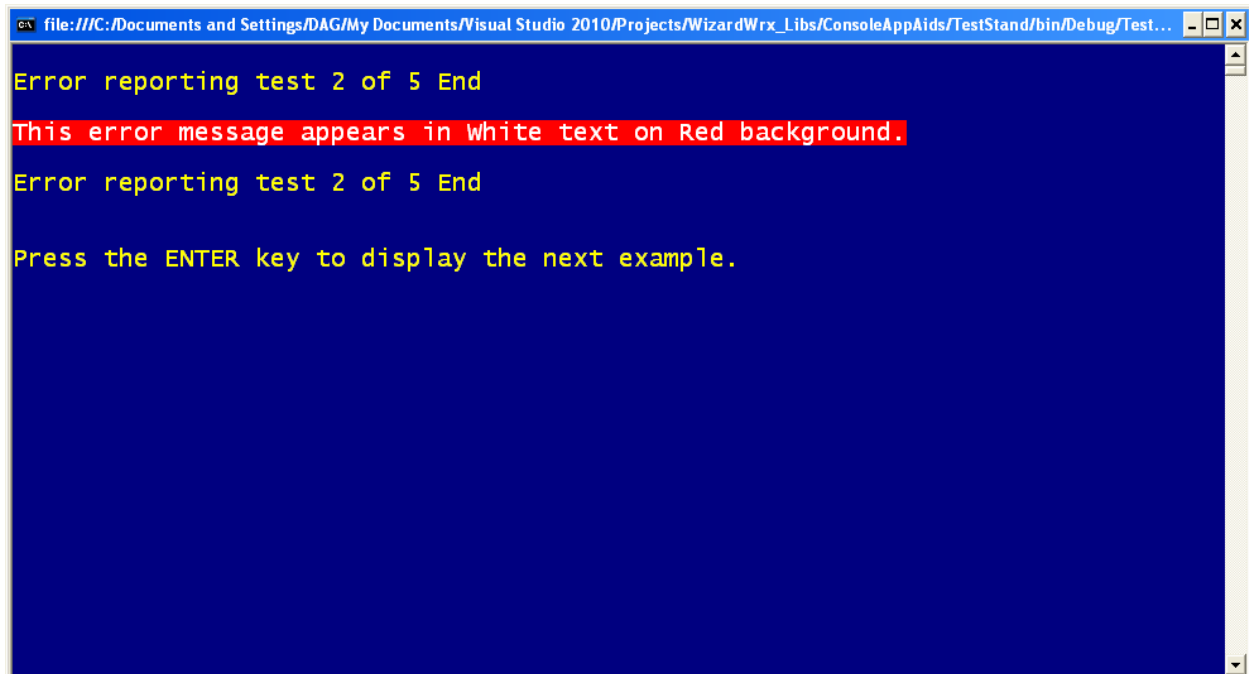


DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION



```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...  
Error reporting test 1 of 5 End  
This error message appears in the default text on the default background.  
Error reporting test 1 of 5 End  
Press the ENTER key to display the next example.
```

Figure 11 is the first of five tests, using the same foreground and background colors used in the last five screens to display a one-line message.

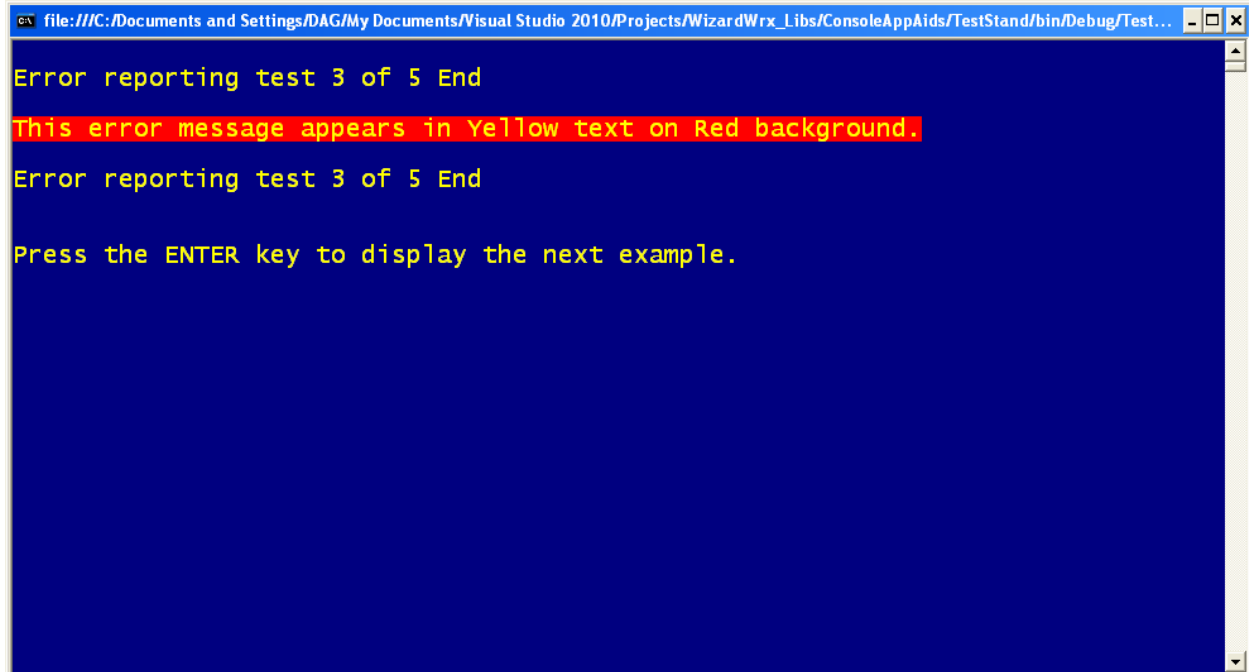


```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...  
Error reporting test 2 of 5 End  
This error message appears in white text on Red background.  
Error reporting test 2 of 5 End  
Press the ENTER key to display the next example.
```

Figure 12 is the second of the five one-line message tests. Although the color combination is as harsh as ever, it is more tolerable for one line than for a whole screen full of text, as in **Figure 7**.

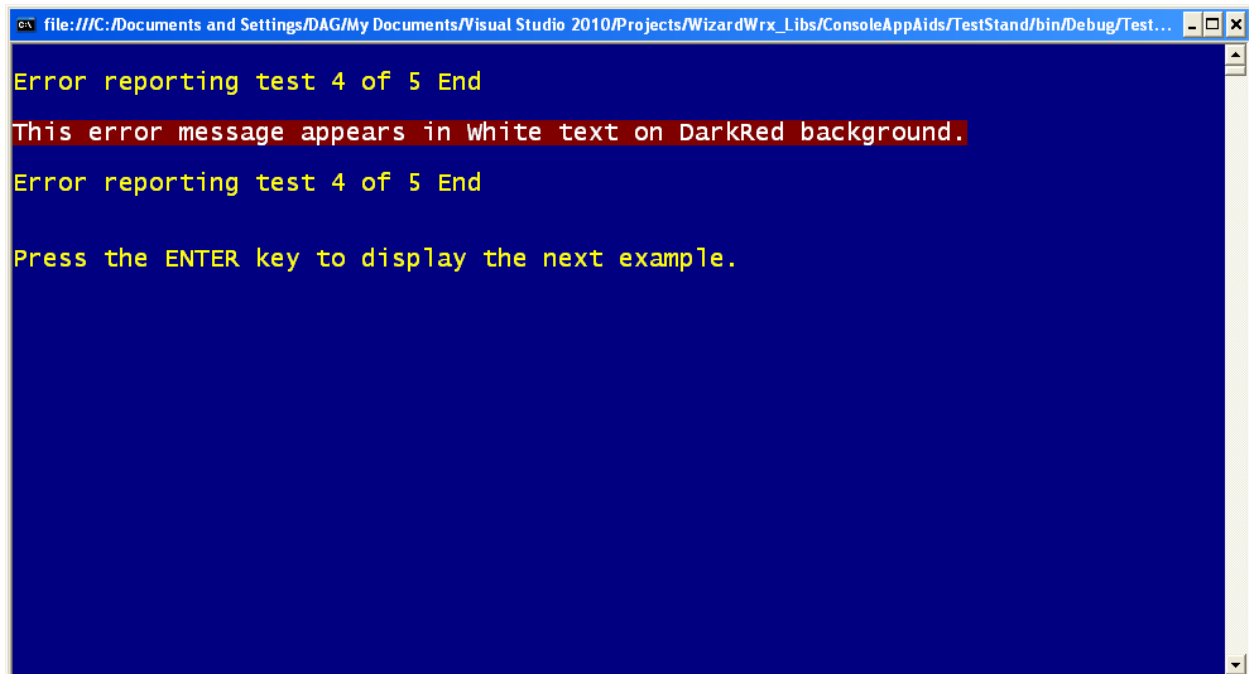


DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION



```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrX_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 3 of 5 End
This error message appears in Yellow text on Red background.
Error reporting test 3 of 5 End
Press the ENTER key to display the next example.
```

Figure 13 is the third of the five one-line message tests. The observations noted in the caption of **Figure 12** apply equally to this example.

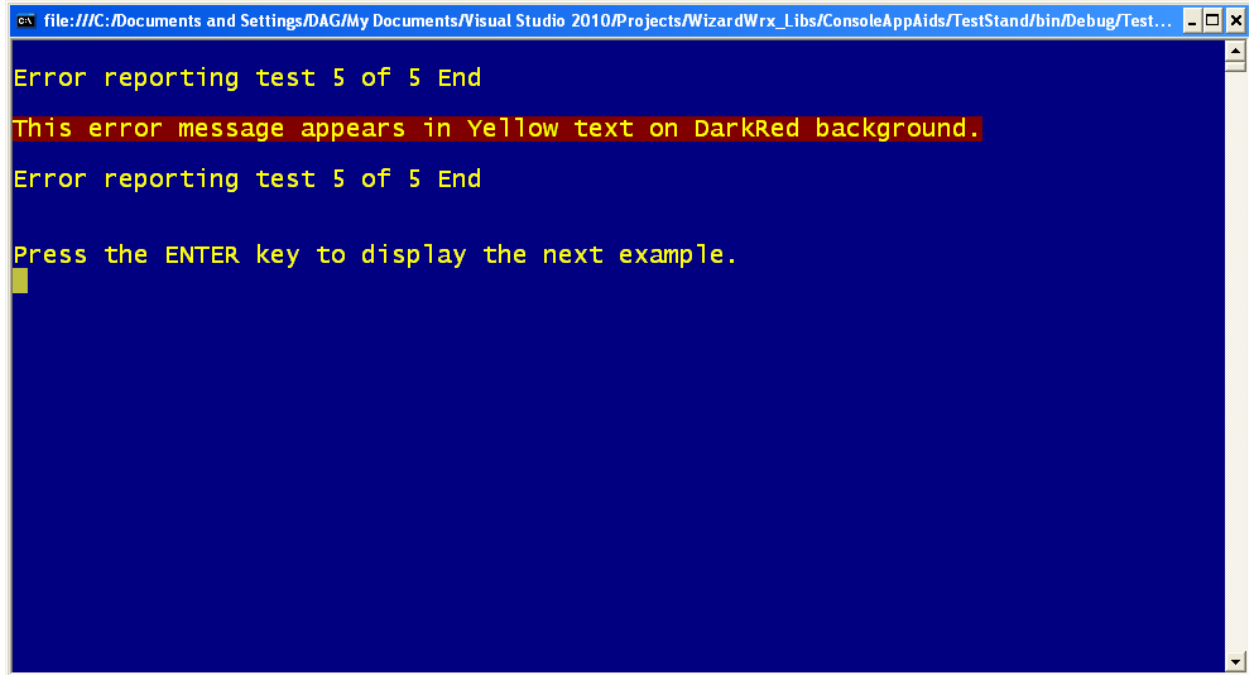


```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrX_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Error reporting test 4 of 5 End
This error message appears in White text on DarkRed background.
Error reporting test 4 of 5 End
Press the ENTER key to display the next example.
```

Figure 14 is white text on the more subdued dark red background, applied to a limited amount of text. About four months of regular use in a production utility that frequently generates brief error messages in normal use confirm the effectiveness of this color combination.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION



```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...  
Error reporting test 5 of 5 End  
This error message appears in Yellow text on DarkRed background.  
Error reporting test 5 of 5 End  
Press the ENTER key to display the next example.  
■
```

Figure 15 is yellow text on the more subdued background that the Console class, and MS-DOS, long before it, calls Dark Red. The yellow text is a little more jarring, and it might be appropriate for a really serious error. However, since my wife thinks this combination blurs the text, I have been reluctant to use it in production code.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
DarkMagenta background color.
This is a 4th test.
Test # 19 end
The next 5 tests use the following two parameters.
    Count    =    10 (Default =    10)
    Interval = 1,000 (Default = 1,000)
Test # 20 begin
    Test a FixedConsoleWriter with Black foreground color and
    White background color. Text is indented 0 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 20 end
Test # 21 begin
    Test a FixedConsoleWriter with Black foreground color and
    White background color. Text is indented 4 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 4.
```

Figure 16 displays the first two examples of the `FixedConsoleWriter` class, which displays text at a fixed position relative to the left margin. The first test demonstrates that flush left is a legal starting position. The Fixed attribute actually refers to the vertical position on the screen.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
    Count    =    10 (Default =    10)
    Interval = 1,000 (Default = 1,000)
Test # 20 begin
    Test a FixedConsoleWriter with Black foreground color and
    White background color. Text is indented 0 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 20 end
Test # 21 begin
    Test a FixedConsoleWriter with Black foreground color and
    White background color. Text is indented 4 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 21 end
Test # 22 begin
    Test a FixedConsoleWriter with Black foreground color and
    White background color. Text is indented 8 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 4.
```

Figure 17 displays the first three tests of the `FixedConsoleWriter` class, moving the starting position progressively further from the left edge, with the goal of forcing a line wrap.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test # 21 end
Test # 22 begin
    Test a FixedConsoleWriter with Black foreground color and
    white background color. Text is indented 8 spaces from the left.
    Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 22 end
Test # 23 begin
    Test a FixedConsoleWriter with Black foreground color and
    white background color. Text is indented 12 spaces from the left.
    Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 23 end
Test # 24 begin
    Test a FixedConsoleWriter with Black foreground color and
    white background color. Text is indented 16 spaces from the left.
    Waiting for 1,000 milliseconds (1 seconds). This is pause 1 of 1
0.
```

Figure 18 is the first of three illustrations of how the `FixedConsoleWriter` class behaves when the text wraps around.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test # 22 begin
    Test a FixedConsoleWriter with Black foreground color and
    white background color. Text is indented 8 spaces from the left.
    Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 22 end
Test # 23 begin
    Test a FixedConsoleWriter with Black foreground color and
    white background color. Text is indented 12 spaces from the left.
    Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 23 end
Test # 24 begin
    Test a FixedConsoleWriter with Black foreground color and
    white background color. Text is indented 16 spaces from the left.
    Waiting for 1,000 milliseconds (1 seconds). This is pause 6.
```

Figure 19 is the second of three illustrations of how the `FixedConsoleWriter` class behaves when the text wraps around. The vertical position is unchanged from that of the wrapped message shown in **Figure 18**, because the carat retreated when it wrote the shorter line.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
white background color. Text is indented 8 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 22 end
Test # 23 begin
Test a FixedConsoleWriter with Black foreground color and
white background color. Text is indented 12 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of 10.
Test # 23 end
Test # 24 begin
Test a FixedConsoleWriter with Black foreground color and
white background color. Text is indented 16 spaces from the left.
Waiting for 1,000 milliseconds (1 seconds). This is pause 10 of
10.
Test # 24 end
Test # 24: This test displays a countdown of 30 seconds.
Test ending in 13 seconds
```

Figure 20 is the second of three illustrations of how the `FixedConsoleWriter` class behaves when the text wraps around. The carat is positioned so that the next line is written below the last line occupied by part of the last fixed message, and the screen resumes scrolling.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
AppExceptionLogger properties:
AppEventSourceID = WizardWrx
AppStackTraceDisposition = Omit
AppSubsystem = CUI
MessageBackgroundColor = White
MessageForegroundColor = Red
EventLoggingState = Enabled
Test TimeedWaitTest 24:
Test Case 1 Properties: TwTC_puintWaitSeconds = 5
TwTC_pstrCountdownWaitingFor = A TimeedWaitTest
TwTC_pclrTextColor = Green
TwTC_pclrTextBackgroundColor = Black
TwTC_penmInterruptCriterion = None
A TimeedWaitTest NOW
Test Case 1 completed.
Test Case 2 Properties: TwTC_puintWaitSeconds = 10
TwTC_pstrCountdownWaitingFor = This and That
TwTC_pclrTextColor = Green
TwTC_pclrTextBackgroundColor = Black
TwTC_penmInterruptCriterion = None
This and That in 8 seconds
```

Figure 21 is the first of a series of demonstrations of a `TimedWait` method that combines the `FixedConsoleWriter` class with a countdown clock built around the `Thread.Sleep` method to freeze the screen for a maximum number of seconds.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test TimeedWaitTest 24:
Test Case 1 Properties: TwTC_puintWaitSeconds      = 5
                       TwTC_pstrCountdownWaitingFor = A TimeedWaitTest
                       TwTC_pclrTextColor          = Green
                       TwTC_pclrTextBackgroundColor = Black
                       TwTC_penmInterruptCriterion  = None
A TimeedWaitTest NOW
Test Case 1 completed.
Test Case 2 Properties: TwTC_puintWaitSeconds      = 10
                       TwTC_pstrCountdownWaitingFor = This and That
                       TwTC_pclrTextColor          = Green
                       TwTC_pclrTextBackgroundColor = Black
                       TwTC_penmInterruptCriterion  = None
This and That NOW
Test Case 2 completed.
Test Case 3 Properties: TwTC_puintWaitSeconds      = 15
                       TwTC_pstrCountdownWaitingFor =
                       TwTC_pclrTextColor          = Green
                       TwTC_pclrTextBackgroundColor = Black
                       TwTC_penmInterruptCriterion  = None
Program ending in 11 seconds
```

Figure 22 displays more permutations of the argument values that can be passed into the `TimedWait` method.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test Case 2 Properties: TwTC_puintWaitSeconds      = 10
                       TwTC_pstrCountdownWaitingFor = This and That
                       TwTC_pclrTextColor          = Green
                       TwTC_pclrTextBackgroundColor = Black
                       TwTC_penmInterruptCriterion  = None
This and That NOW
Test Case 2 completed.
Test Case 3 Properties: TwTC_puintWaitSeconds      = 15
                       TwTC_pstrCountdownWaitingFor =
                       TwTC_pclrTextColor          = Green
                       TwTC_pclrTextBackgroundColor = Black
                       TwTC_penmInterruptCriterion  = None
Program ending NOW
Test Case 3 completed.
Test Case 4 Properties: TwTC_puintWaitSeconds      = 20
                       TwTC_pstrCountdownWaitingFor = Exiting
                       TwTC_pclrTextColor          = Green
                       TwTC_pclrTextBackgroundColor = Black
                       TwTC_penmInterruptCriterion  = None
Exiting in 16 seconds
```

Figure 23 displays even more permutations of `TimedWait`, the most interesting being the one where argument `TwTC_pstrCountdownWaitingFor` is left blank, causing the display to say "Program ending."



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...

Test Case 3 Properties: TwTC_puintWaitSeconds      = 15
                      TwTC_pstrCountdownWaitingFor = 
                      TwTC_pclrTextColor          = Green
                      TwTC_pclrTextBackgroundColor = Black
                      TwTC_penmInterruptCriterion  = None

Program ending NOW

Test Case 3 completed.

Test Case 4 Properties: TwTC_puintWaitSeconds      = 20
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = Green
                      TwTC_pclrTextBackgroundColor = Black
                      TwTC_penmInterruptCriterion  = None

Exiting NOW

Test Case 4 completed.

Test Case 5 Properties: TwTC_puintWaitSeconds      = 25
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = White
                      TwTC_pclrTextBackgroundColor = Black
                      TwTC_penmInterruptCriterion  = None

Exiting in 20 seconds
```

Figure 24 demonstrates control over foreground (text) and background colors used by `TimedWait`.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...

Test Case 4 Properties: TwTC_puintWaitSeconds      = 20
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = Green
                      TwTC_pclrTextBackgroundColor = Black
                      TwTC_penmInterruptCriterion  = None

Exiting NOW

Test Case 4 completed.

Test Case 5 Properties: TwTC_puintWaitSeconds      = 25
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = White
                      TwTC_pclrTextBackgroundColor = Black
                      TwTC_penmInterruptCriterion  = None

Exiting NOW

Test Case 5 completed.

Test Case 6 Properties: TwTC_puintWaitSeconds      = 30
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = White
                      TwTC_pclrTextBackgroundColor = Blue
                      TwTC_penmInterruptCriterion  = None

Exiting in 28 seconds
```

Figure 25 demonstrates yet another color combination that can be used with `TimedWait`; the last example changes the foreground and background colors.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test Case 5 Properties: TwTC_puintWaitSeconds      = 25
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = White
                      TwTC_pclrTextBackgroundColor = Black
                      TwTC_penmInterruptCriterion  = None
Exiting NOW
Test Case 5 completed.
Test Case 6 Properties: TwTC_puintWaitSeconds      = 30
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = White
                      TwTC_pclrTextBackgroundColor = Blue
                      TwTC_penmInterruptCriterion  = None
Exiting NOW
Test Case 6 completed.
Test Case 7 Properties: TwTC_puintWaitSeconds      = 0
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = Blue
                      TwTC_pclrTextBackgroundColor = Blue
                      TwTC_penmInterruptCriterion  = AnyKey
Exiting in 27 seconds
```

Figure 26 demonstrates that specifying the same colors for foreground and background causes `TimedWait` to use the default screen colors. Although this example shows Blue, *any* color has the same effect.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test Case 6 Properties: TwTC_puintWaitSeconds      = 30
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = White
                      TwTC_pclrTextBackgroundColor = Blue
                      TwTC_penmInterruptCriterion  = None
Exiting NOW
Test Case 6 completed.
Test Case 7 Properties: TwTC_puintWaitSeconds      = 0
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = Blue
                      TwTC_pclrTextBackgroundColor = Blue
                      TwTC_penmInterruptCriterion  = AnyKey
Exiting NOW
Test Case 7 completed.
Test Case 8 Properties: TwTC_puintWaitSeconds      = 10
                      TwTC_pstrCountdownWaitingFor = Exiting
                      TwTC_pclrTextColor          = Blue
                      TwTC_pclrTextBackgroundColor = Blue
                      TwTC_penmInterruptCriterion  = CarriageReturn
Exiting in 10 seconds
```

Figure 27 illustrates an early exit from `TimedWait` in case 7. Currently, there is no provision for indicating that the countdown was interrupted.



DEMONSTRATION OF VERSION 3.6.197.29569 MESSAGEINCOLOR CLASS DOCUMENTATION

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
TWTC_pclrTextBackgroundColor = Blue
TWTC_penmInterruptCriterion = CarriageReturn
Exiting NOW
Test Case 8 completed.
Test Case 9 Properties: TWTC_puintWaitSeconds = 0
TWTC_pstrCountdownWaitingFor = Exiting
TWTC_pclrTextColor = Blue
TWTC_pclrTextBackgroundColor = Blue
TWTC_penmInterruptCriterion = None
Exiting NOW
Test Case 9 completed.
Test Case 10 Properties: TWTC_puintWaitSeconds = 10
TWTC_pstrCountdownWaitingFor = Exiting
TWTC_pclrTextColor = White
TWTC_pclrTextBackgroundColor = White
TWTC_penmInterruptCriterion = None
Exiting NOW
Test Case 10 completed.
```

Figure 28 displays the remaining permutations of `TimedWait` parameters.

```
file:///C:/Documents and Settings/DAG/My Documents/Visual Studio 2010/Projects/WizardWrx_Libs/ConsoleAppAids/TestStand/bin/Debug/Test...
Test Case 9 Properties: TWTC_puintWaitSeconds = 0
TWTC_pstrCountdownWaitingFor = Exiting
TWTC_pclrTextColor = Blue
TWTC_pclrTextBackgroundColor = Blue
TWTC_penmInterruptCriterion = None
Exiting NOW
Test Case 9 completed.
Test Case 10 Properties: TWTC_puintWaitSeconds = 10
TWTC_pstrCountdownWaitingFor = Exiting
TWTC_pclrTextColor = White
TWTC_pclrTextBackgroundColor = White
TWTC_penmInterruptCriterion = None
Exiting NOW
Test Case 10 completed.
Test TimeedWaitTest 24 is over.
TestStand End
05/31/2014 19:04:57.466 (06/01/2014 00:04:57.466 UTC)
Elapsed time: 0:01:26:45.806
Please press the ENTER (Return) key to exit the program.
```

Figure 29 is the final screen displayed by the test program. Since it is running in debug mode, it waits for a carbon unit.