# Overview

In this exercise you will modify the **Orders.dbv** program that you created in [Exercise – 1](file:///E:\Documents\Custom%20Office%20Templates\Exercise%20-%201.docx). You will create an external subroutine, which will create and display a Synergy Window containing a simple message. The message to be displayed will be passed to the subroutine as an argument.

# Resources

* [Synergy Best Practices - Coding Standards](http://jobfunc2.cu.net/Job%20Functions/Programmer/Programmer%20Handbook/Tims%20Best%20Practices%20-%20Standards/Synergy%20Best%20Practices%20-%20Coding%20Standards.docx)

# Exercise

1. Using Visual Studio, open the previously created “Orders” project.
2. Using Visual Studio, open “Orders.dbv”.
3. Using Visual Studio, create a new file.
4. Create a new external subroutine called DisplayMessage. The subroutine should accept one argument, the text to be displayed in the window.
5. As the subroutine will use Synergy Windows routines, it must .include “WND:windows.def” in its data division.
6. Add code to the subroutine to consider the size of the passed message. Using the size of the passed message to determine the size and position of the required window, which should be centered approximately on the screen.
7. When the window has been built and displayed, wait for a key press before removing the window.
8. Save the new file as **DisplayMessage.dbl**. Add this file to your project.
9. Using Visual Studio, open the **Orders.dbv** source file.
10. Modify the pProcess routine in the main program to make use of the new subroutine. Remove the code that manually displays a message window.
11. Compile, link, and run the program.

# Discussion

The functionality of the subroutine can be broadly based on the existing code from [Exercise – 5](file:///E:\Documents\Synergy%20Tutorials\Synergy%20Training\Language%20Essentials\Exercise%20-%205.docx), which is already displaying a message window.

The changes that we have made in this exercise will not alter the operation of the program, which should continue to operate exactly as in the previous exercise. Now however, you have a general-purpose routine that can easily be called whenever you need to display a message window to the screen.