# 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 be adding three internal subroutines, which will perform the application start-up, main processing, and shutdown logic in turn.

# 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. In the procedure division, create three internal subroutines called:
   * “pInitialize”
   * “pProcess”
   * “pShutDown”
4. At this point, simply have each routine use the RETURN statement.
5. Add code to call each of these routines in turn.
6. Compile, link and run the program.

# Discussion

In this instance the internal subroutines are being used to partition parts of the code into logical segments. For example, all code associated with starting up the application will be collected together in the “pInitialize” subroutine.

You will need to pay careful attention to the placement of the subroutines, and the calls to those routines, within the overall code of the procedure division. The code in each subroutine should be executed once and only once. When each subroutine has been executed the program should terminate.

At this point, the subroutines are not actually doing anything (other than returning). We will go on to make the routines actually do something in later exercises.