# Overview

To become familiar with basic input method processing.

# Resources

* [Synergy DBL Language Reference](https://www.synergex.com/docs/index.htm)
* [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)
* [Traditional Synergy in Visual Studio - CU Wiki](http://echo.cu.net/cuwiki/Traditional_Synergy_in_Visual_Studio)
* [Traditional Synergy in Visual Studio Common Terminology - CU Wiki](http://echo.cu.net/cuwiki/Traditional_Synergy_in_Visual_Studio_Common_Terminology)
* [Installing Traditional Synergy in Visual Studio Templates - CU Wiki](http://echo.cu.net/cuwiki/Installing_Traditional_Synergy_in_Visual_Studio_Templates)
* [Creating a New DBR Project (TSVS) - CU Wiki](http://echo.cu.net/cuwiki/Creating_a_New_DBR_Project_(TSVS))
* [Debugging (TSVS) - CU Wiki](http://echo.cu.net/cuwiki/Debugging_(TSVS))
* [CU Toolkit Manual](http://jobfunc2.cu.net/Job%20Functions/Programmer/Programmer%20Handbook/Synergy%20Reference%20-%20CUToolkit%20Reference%20Manual.doc)
* [Getting Started With Repository](https://www.synergex.com/docs/versions/v111/index.htm#rps/1_WELCOME_RPS.htm)
* [Synergy Data Language](https://www.synergex.com/docs/versions/v111/index.htm#rps/6_SDL.htm)

# Exercise

1. For each of the input methods you write, they will all have the same functionality:
   * Using **E\_SECT:**
     + This will allow you to run the program and see when the various methods are being called in the processing loop.
   * Place the subroutine name in the info bar.
   * Place the current value of **G\_ENTNAM** in the info bar.
   * With the text left justified.
   * Specifying to clear the info line each time.
2. Using Visual Studio, add the following methods to your “DemoMntG.dbl” file:
3. Input menu method (**imnu\_**).
   * In the input menu method, create a using statement, and trap **G\_ENTNAM** for “**DEMO\_ACTIVE**”, and “**DEMO\_DESCR**”.
   * When you receive “**DEMO\_ACTIVE**”, display a message stating the value of **G\_ENTNAM**, and clear **G\_SELECT**.
   * When you receive “**DEMO\_DESCR**”, display a message.
4. Set default method (**isdf\_**):
   * The parameters are the same as an input arrive method.
5. Add a field level arrive method (**am\_**) to the active field.
6. A change method (**cm\_**) to the active field, which will signal “**DEMO\_ACTIVE**”.
7. A field dependency method (**fdep\_**) to the active field:
   * When the active field (check box) is selected have the change method call the active field dependency method, which will display the message “Active” to the screen.
   * Remember to add a call to the active field dependency method in your input display method.
8. Add a display method (**dsp\_**) to the description field.
9. A change method (**cm\_**) to the description field, which will signal “**DEMO\_DESCR**”.
10. In “DemoMnt.dbv”, add a program level menu method “**DemoMnt\_Menu**”:
    * Add logic to trap for the signal “**DEMO\_DESCR**”, and “**DEMO\_ACTIVE**”:
      + Have the menu method display a message with the current value of **G\_ENTNAM** when either of the signals is received.

# Discussion

This is just a brief introduction to CU Toolkit Methods. The methods in this exercise are intentionally simple, and are not meant to represent the complexity found in a typical production program.