# Overview

The purpose of this practice set is to test your knowledge of the CU Toolkit, data access, sequences, drill methods, reporting, and basic program design.

# 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

To complete this exercise you should complete the following steps, in the order shown:

1. Create a schema for an insurance file with the following fields:
   * Insurance File:
     + The primary key for this file will be the insurance number.
     + No duplicates will be allowed.
     + Insurance number:
       - Alphanumeric field with a length of 3.
       - Lookup available
     + Insurance name:
       - Alphanumeric field with a length of 50.
     + Insurance address #1:
       - Alphanumeric field with a length of 50.
     + Insurance address #2:
       - Alphanumeric field with a length of 50.
     + Insurance city:
       - Alphanumeric field with a length of 30.
     + Insurance state:
       - Alphanumeric field with a length of 4.
     + Insurance zip code:
       - Alphanumeric field with a length of 9.
     + Insurance phone number:
       - Alphanumeric field with a length of 10.
     + Insurance fax number:
       - Alphanumeric field with a length of 10.
     + Insurance assignment flag:
       - Decimal field with a length of 1.
       - Y/N field.
     + Insurance percentage:
       - Decimal field with a length of 5.
     + Insurance type flag:
       - Alphanumeric field with a length of 1.
     + Insurance form type:
       - Decimal field with length of 3.
     + Insurance supplier id:
       - Alphanumeric field with a length of 15.
     + Insurance provider id:
       - Alphanumeric field with a length of 15.
     + Insurance receiver/submitter id:
       - Alphanumeric field with a length of 15.
     + Insurance branch:
       - Decimal field with length of 3.
     + Insurance flag to print retail on claim:
       - Decimal field with length of 1.
       - Y/N field.
2. Create the table creation script based on the schema.
3. Create the four standard data access routines (delete, insert, select, and update).
4. Create a script containing all necessary input, and text windows.
5. Create a Maintenance program for the insurance file where records can be added, changed, inquired upon, and deleted.
6. When user is adding a new insurance record, if they hit return, get the next available insurance number.
7. Do not allow them to enter in a blank insurance number or an insurance number of "000".
8. If user is dealing with any of the other modes and they hit return, get the next available insurance record:
   * In other words, if for example they have looked up an insurance record in CHANGE mode all ready, if they hit return, it will take them to the next insurance.
9. Do not allow the entry of zip codes other the 6 digit or 9 digit standard zip codes.
10. Insurance assignment flag should default to a value of "Yes".
11. The insurance percentage should not be greater than 100 percent.
12. The percentage field has an implied 2 decimal place.
13. After user enters in value, the value formatted with the decimal should display in its place.
14. The insurance type field should only allow the following values and display their corresponding descriptions beside them:
    * R-Medicare
    * D-Medicaid
    * C-Medi-Cal
    * O-Other
15. The insurance form type field should allow values between:
    * 100-199
    * 300-399
    * 999
16. The user should not be allowed to enter in a blank value for the insurance receiver/submitter id field.
17. The insurance flag to print retail on claim should default to "No".
18. Create a listing for the Insurance Company File that displays the following information:
    * Insurance number
    * Insurance name
    * Insurance address #1
    * Insurance address #2
    * Insurance city
    * Insurance state
    * Insurance zip code
    * Insurance phone number

# Discussion

This is a very large and complex exercise. Complete the exercise one step at a time and test the application as you go. Make sure that you complete each step fully before moving on to the next step.