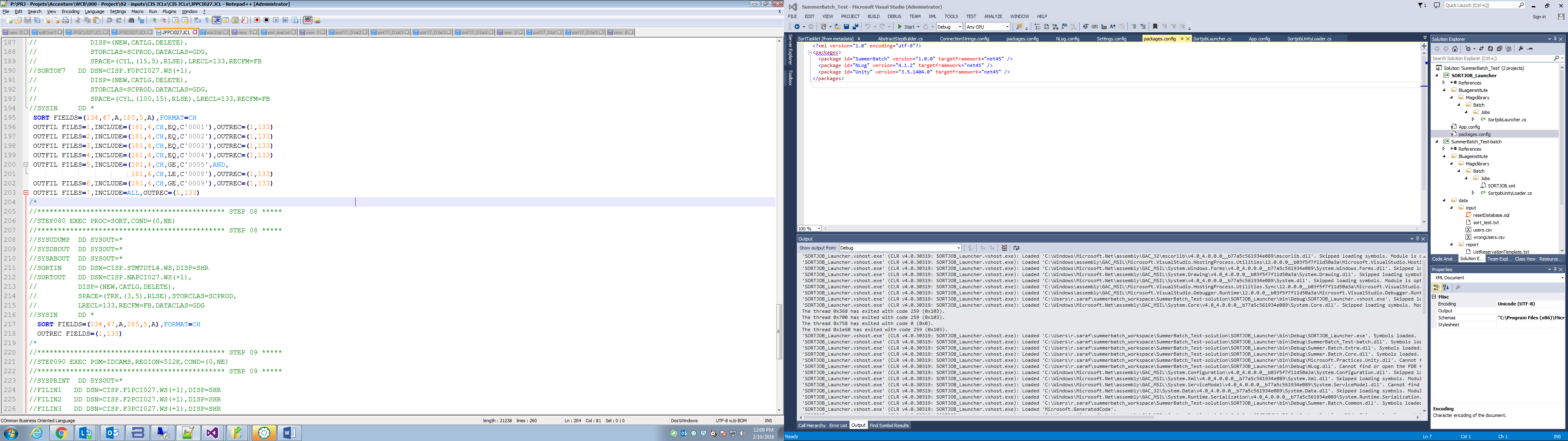
SummerBatch – CustomSortTasklet

1. Requirements –

In JCL job (JPPCI027), requirement is to sort the input files, merge these input files and create output files as per ‘INCLUDE’ filters.



Explanation –

*SORT FIELDS= (134, 47, A, 185, 5, A), FORMAT=CH*

This means sorting needs to be done on two fields, first field – 47 character long starts with index 134 in ascending order. Second field – 5 character long starts with index 185 in ascending order. Both fields are in Character format.

*OUTFIL FILES=1, INCLUDE= (181, 4, CH, EQ, C’0001’), OUTREC= (1,133)*

This means output file 1 should contain records satisfy below criteria –

Character content starting at index 181 of length 4 should be equal to ‘0001’ – output only first 133 characters.

1. Old Sort Step –

Existing sort step sorts based on the fields provided and it writes all the records satisfying ‘INCLUDE’ or ‘OMIT’ criteria to a single file.

Generated code – after bluage generation –

|  |
| --- |
| container.StepScopeRegistration<ITasklet, SortTasklet>("step0Batchlet") .Property("Input").Resources("#{settings['SORTJOB.step0.inputFiles']}")  .Property("Output").Resource("#{settings['SORTJOB.step0.outputFile']}")  .Property("Include").Value("75,2,CH,EQ,C'RD'")  .Property("HeaderSize").Value(0)  .Property("Separator").Value(Environment.NewLine)  .Property("SortCard").Value("FORMAT=CH,FIELDS=(121,14,CH,A)")  .Register(); |

1. Custom Sort Step –

This step sorts the records based on the fields provided and based on the number of ‘INCLUDE’ or ‘OMIT’ filters it generated the output files.

|  |
| --- |
| IList<OutputFile> list = new List<OutputFile>();  var outputFile1 = new OutputFile  {  Include = "75,2,CH,EQ,C'RD'",  Outrec = "25,200"  };  var outputFile2 = new OutputFile  {  Include = "75,2,CH,EQ,C'XD'"  };  var outputFile3 = new OutputFile  {  Include = "75,2,CH,EQ,C'CD'"  };  var outputFile4 = new OutputFile  {  Include = "75,2,CH,EQ,C'YD'"  };  var outputFile5 = new OutputFile  {    };  list.Add(outputFile1);  list.Add(outputFile2);  list.Add(outputFile3);  list.Add(outputFile4);  list.Add(outputFile5);  container.StepScopeRegistration<ITasklet,CustomSortTasklet>("step0Batchlet")  .Property("Input").Resources("#{settings['SORTJOB.step0.inputFiles']}")  .Property("Output").Resource("#{settings['SORTJOB.step0.outputFile']}")  .Property("outputFile").Value(list)  .Property("HeaderSize").Value(0)  .Property("Separator").Value(Environment.NewLine)  .Property("SortCard").Value("FORMAT=CH,FIELDS=(121,14,CH,A)")  .Register(); |

Highlighted part is incorporated for CustomSortTasklet.

1. Step by step test batch generation to test CustomSortTasklet –
   1. Create a bluage project.
   2. Create a step ‘sort’ with Stereotype – ‘SORT\_STEP’
   3. Enter following values in Tags under SORT\_STEP
      1. For include - (75,2,CH,EQ,C'RD')
      2. For sort - FORMAT=CH,FIELDS=(121,14,CH,A)
   4. Add this step to the job.
   5. Add following properties for input and output files in Settings.config
      1. <add key="SORTJOB.step0.Input" value="D:\SummerBatch\Summer.Batch.CoreTests\bin\Debug\TestData\Sort\Input\sort17.txt" />
      2. <add key="SORTJOB.step0.Output" value="D:\SummerBatch\Summer.Batch.CoreTests\bin\Debug\TestData\Sort\Input\sort17\_O.txt" />
   6. Generate the bluage project, open SummerBatch\_Test.sln to view the code in visual studio.
   7. Before proceeding for CustomSortTasklet, we need to add dll files for incorporating the new changes made to SummerBatch.

In the generated bluage project, open packages.config

|  |
| --- |
| <packages>  <package id="SummerBatch" version="1.0.0" targetFramework="net45" />  <package id="NLog" version="4.1.2" targetFramework="net45" />  <package id="Unity" version="3.5.1404.0" targetFramework="net45" />  </packages> |

And open packages.config from SummerBatch project,

|  |
| --- |
| <package id="NLog" version="4.2.3" targetFramework="net45" />  <package id="Unity" version="4.0.1" targetFramework="net45" />  </packages> |

Because of different versions, we need to replace all the dll files from SummerBatch project to test project.

1. First build SummerBatch project to generate the dll files, we need to copy below dll files from location –

D:\SummerBatch\Summer.Batch.Extra\bin\Debug

Dll files – Summer.Batch.Common.dll, Summer.Batch.Core.dll, Summer.Batch.Data.dll, Summer.Batch.Extra.dll, Summer.Batch.Infrastructure.dll

1. Paste all these files to below location of test project –

C:\Users\r.saraf\summerbatch\_workspace\SummerBatch\_Test-solution\packages\SummerBatch.1.0.0\lib\net45

1. We need to copy Nlog dll and Unity dll as well.

Copy NLog.dll and Nlog.xml file from D:\SummerBatch\packages\NLog.4.2.3\lib\net45 location, and put these to C:\Users\r.saraf\summerbatch\_workspace\SummerBatch\_Test-solution\packages\NLog.4.1.2\lib\net45

Copy all files from D:\SummerBatch\packages\Unity.4.0.1\lib\net45 to C:\Users\r.saraf\summerbatch\_workspace\SummerBatch\_Test-solution\packages\Unity.3.5.1404.0\lib\net45.

* 1. Now place the input file (sort17.txt)to below location –

D:\SummerBatch\Summer.Batch.CoreTests\bin\Debug\TestData\Sort\Input

* 1. Run the batch job.

1. Attachments –

Input & SummerBatch changes –



Output –



Dll files –

1. SummerBatch files –



1. Unity files –



1. NLog files –



Test Batch job file –

