# Creating a XML file using - Composer and HJoin steps within Hierarchical Data stage

Here we will use the HJoin step to create a hierarchical structure from the multiple relational data files (lists) based on the parent and child keys.

We will be using 3 different input flat files - employee.txt, address.txt and phoneNumber.txt and the schema file Employee.xsd to compose source data into an XML file.

Please find below link document for step by step Process

**Stages Used**: Sequential File and Hierarchical Data

**Input Files Used**:

1. **Address File**



1. **Employee File**

****

1. **Phone Number Details File**

****

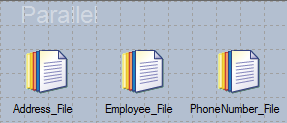
**Schema File used:**

****

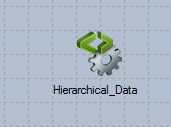
**Job Design:**

Step 1: Create a New Parallel Job (FileToXMl)

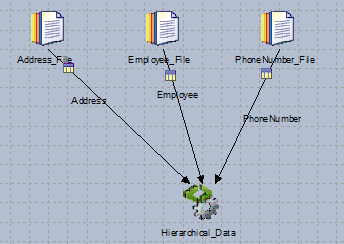
Step2: Add 3 sequential file stages and name each stage as: Address\_File, Employee\_File and PhoneNumber\_File



Step3: Add the Hierarchical Data from Real Time



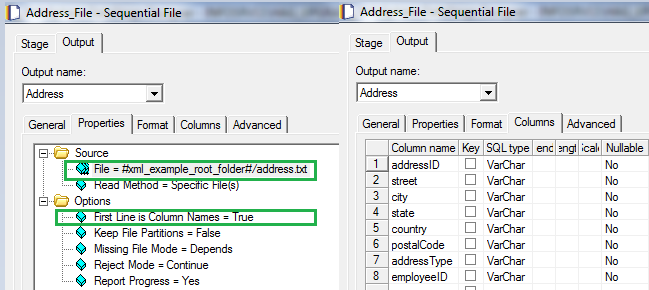
Step4: Connect each sequential file stage to the Hierarchical Data stage and give meaningful link names as seen below in the screenshot

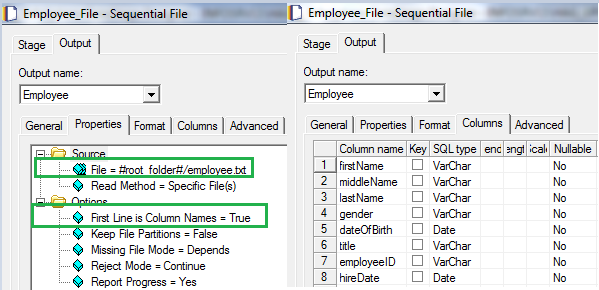


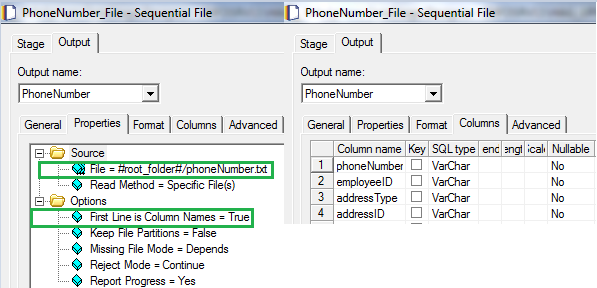
Step5: Use the following information to add a job parameter by name root\_folder

|  |  |
| --- | --- |
| Prompt | Input path |
| Type | String |
| Help Text | Point to the root of the file tree folder |

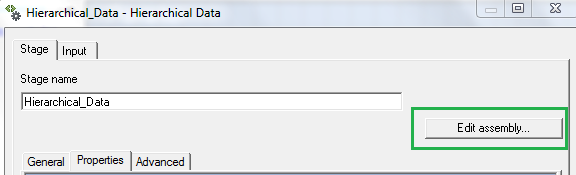
Step6: Configure the sequential file input stage by giving – column names, file path/file name to use and turn on first line is column names option to True

**Address** 

**Employee**

**PhoneNumber**

Step7: Open the Hierarchical\_Data stage and click on Edit Assembly



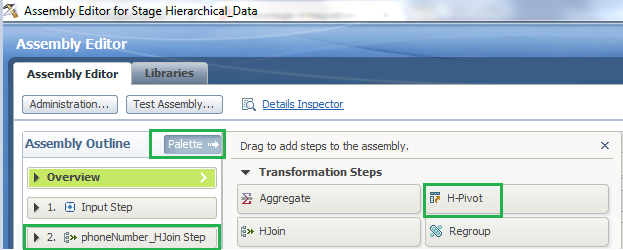
Before proceeding further, we need to import the schema for our example (employee.xsd) file into the library to be used later. This can be done by following the steps below

**Step7.1**: From the Assembly Editor, click the Libraries tab to open the Schema Library Manager.

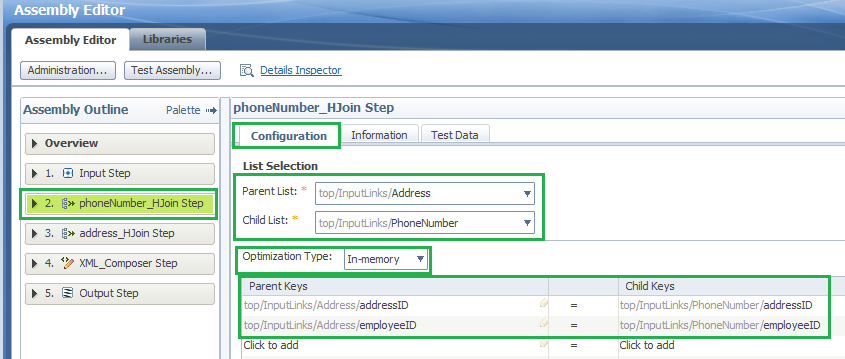
**Step7.2**: To create a library for the example schemas, click New Library. For the library name, enter Schemas\_for\_XML\_examples. For the category, enter Examples. The library is now categorized under Examples

**Step7.3**: Expand the Examples category, select Schemas\_for\_XML\_examples, and then click Import New Resource. Find the Employee.xsd and import into the library.

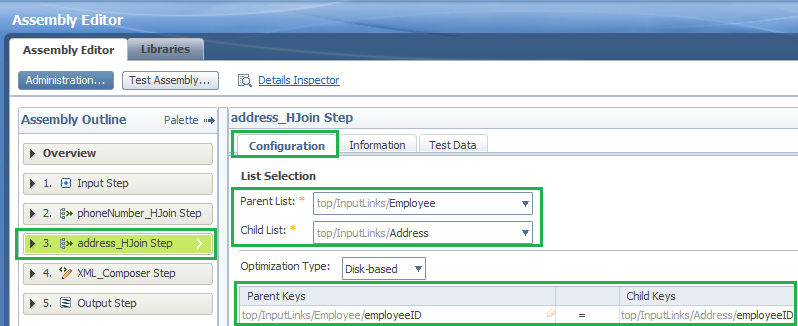
Step8: Within assembly editor, click on Palette and add step – H-Pivot and rename the step to phoneNumber\_HJoin Step

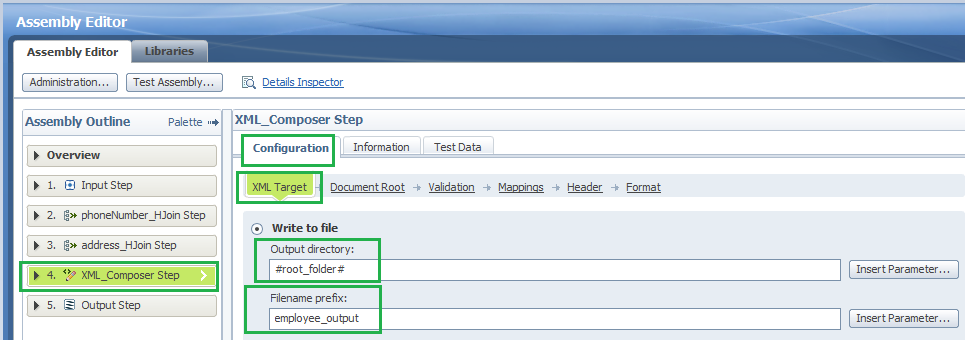
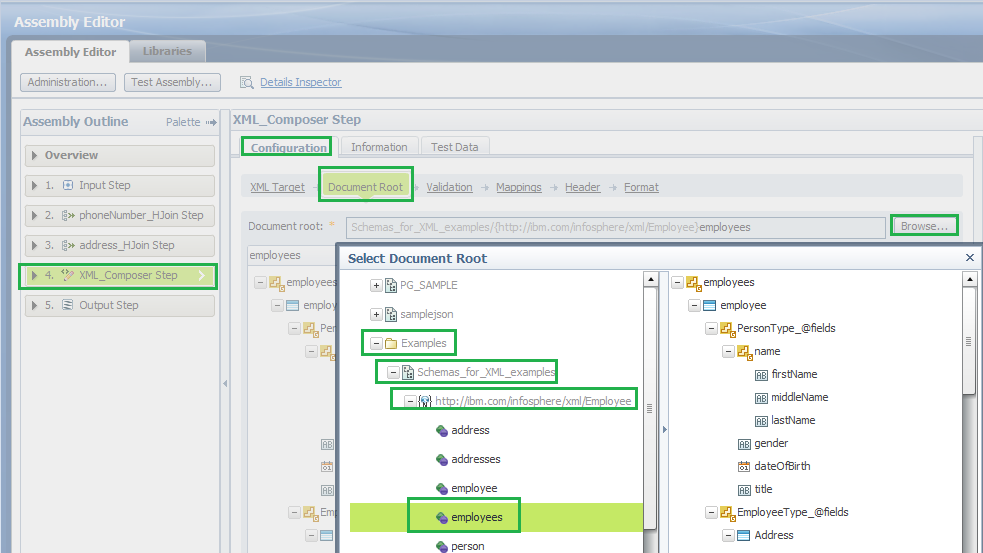


Step9: Click on phoneNumber\_HJoin Step and configure it as shown in the screenshot below by selecting the options highlighted as below. This is to create a list of Phone Numbers within the Address for a particular addressID and employeeID

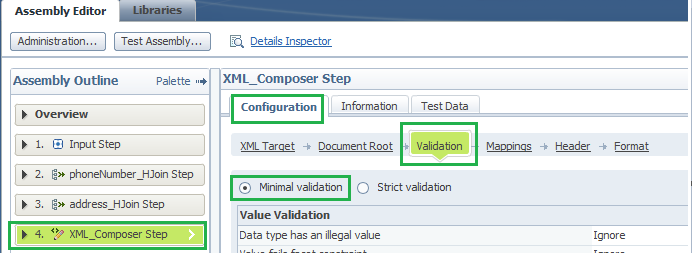


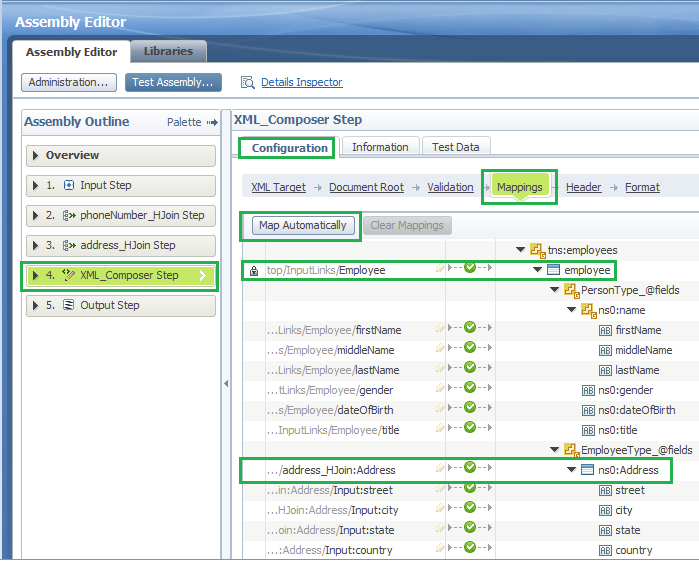
Step10: Similar to step 8, click on Palette and add step – H-Pivot and rename the step to address\_HJoin Step and configure it as shown in the screenshot below by selecting the options highlighted as below. This is to create a list of addresses within the employee for a particular employeeID



Step11: Click on palette and add step XML\_Composer. Configure the XML\_Composer step as shown below. Under XML Target option under configuration tab, provide the output path (click on insert parameter and select the job parameter root\_folder) and output XML file name needed as shown Step12: Next , configure the DocumentRoot option under XML\_Composer step configuration tab, by selecting the schema employee.xsd which you imported into library previously 

Step13: Configure the validation option to Minimal validation as shown in the screenshot below:



Step14: Next is to map input vs output by using the Map Automatically option under the Mappings. This step is to translate the relational data we have in our sequential files into a hierarchical structure

Step15: Save the options, click ok in the stage to save changes. Save, compile and run the job by passing valid value to the parameter root\_folder (the path where your input files are located)

Step16: Open the XML output file created by the job by browsing to root\_folder path and opening the file employee\_output.xml. Below is a screenshot of the XML file created by the job

