Talend 사용자 매뉴얼

2014년 1월

솔루션팀

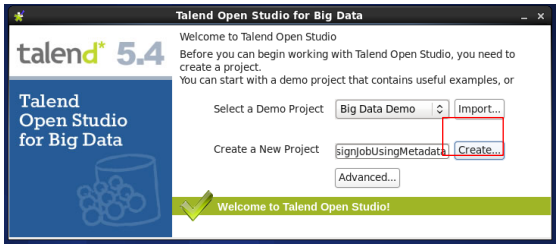
innotree
Total IT Solution & Service

목 차

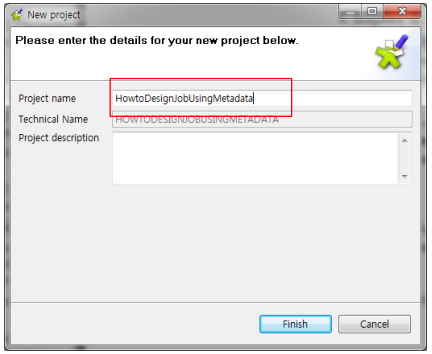
1. 프로젝트 생성

①

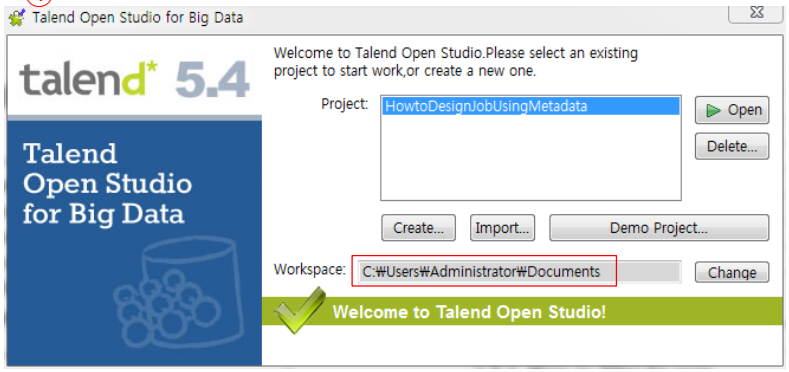
②



③



④



1. 탈렌드 BigData 프로그램을 실행하면 로고 화면이 표시된다.

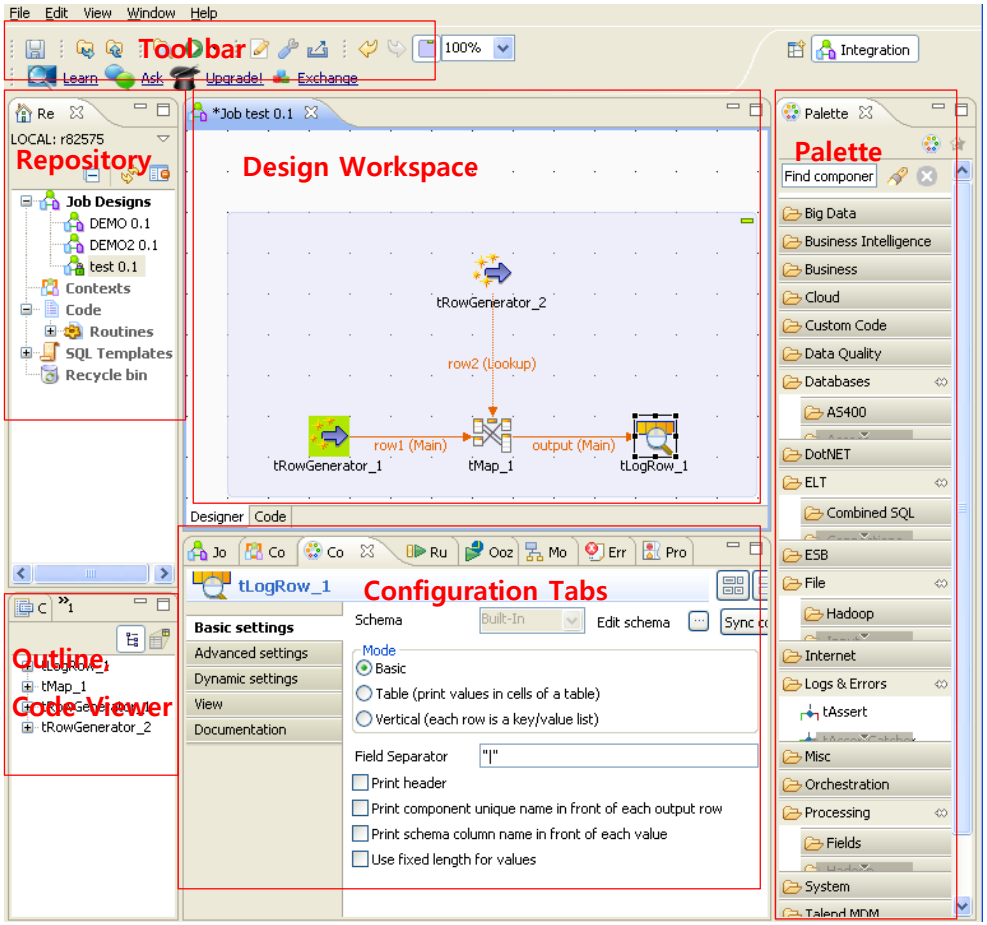
2. 데모 프로젝트를 열거나 새로운 프로젝트를 열수 있는 창이 나타난다. 이 창에서 프로젝트를 생성하기 위하여 Create 버튼을 클릭한다.

3. 새로운 프로젝트의 이름과 프로젝트 설명을 입력한 후 Finish버튼을 클릭한다. Workspace 폴더 안에 프로젝트 이름으로 폴더가 생성된다.

4. 새로 만든 프로젝트가 표시되고, 열거나 삭제할 수 있다. 이 화면에서 추가로 새로운 프로젝트를 만들거나 다른 프로젝트를 Import할 수 있고, 데모 프로젝트를 불러올 수 있다. Workspace를 변경하고자 하는 경우에는 Change버튼을 클릭하여 다른 폴더를 지정해 주면 된다. 새로 생성한 프로젝트를 선택한 후 Open을 클릭하면 Job을 Design할 수 있는 화면이 나타난다.

2. Main Windows 구성 설명

2.1 Main Windows 구성



Main Windows 구성

- **menu bar**
  - 기본적인 이클립스의 기능과 함께 다양한 Talend Studio의 메뉴를 제공한다.
- **toolbar**
  - Save, export items, Run job과 같은 일반적으로 사용하는 작업을 빠르게 엑세스할 수 있는 아이콘을 제공한다.
- **Repository tree view**
  - 잡을 디자인할 때 사용할 수 있는 모든 기술 항목, JobDesigns, Contexts, 재사용 가능한 루틴 등을 모아서 보여준다.
- **Design workspace**
  - Designer tab에서 수행할 잡을 디자인할 수 있다. Code tab에서는 java code를 보여준다.
- **Palette**
  - Design workspace에 가져올 수 있는 컴포넌트들을 보여준다.
- **Configuration views**
  - Design workspace에서 디자인한 컴포넌트의 여러가지 값을 설정 할 수 있다.
- **Outline, Code Viewer**
  - Outline tab은 열려져 있는 Job에서 사용되고 있는 컴포넌트들을 리스트한다.
  - Code Viewer tab은 Design workspace에서 선택한 컴포넌트에 해당하는 코드를 보여준다.

2. Main Windows 구성 설명

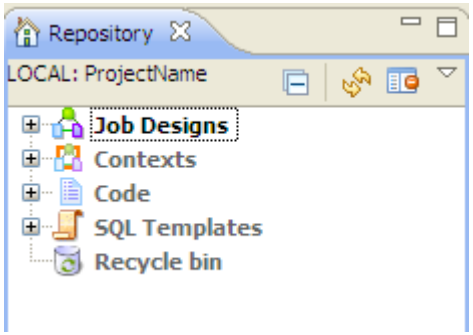
2.2 Toolbar 설명

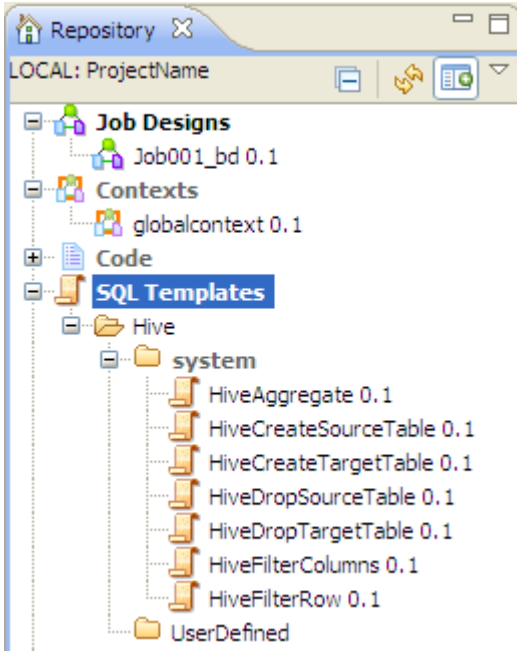
메뉴명	아이콘	설명
Save		디자인한 잡을 저장한다.
Save as		새로운 잡으로 저장한다.
Export items		레파지토리 아이템을 아카이브 파일로 export 한다. 새로운 버전의 Talend Studio나 다른 workstation에서 import하여 사용할 경우 소스파일이 포함되는지 확인해야 한다.
Import items		아카이브 파일로부터 import한다.
Find a specific job		리포지토리 트리 뷰에 나열된 모든 작업을 열 수있는 대화 상자를 표시한다.
Run job		현재 디자인한 내용을 실행한다.
Create		생성 마법사를 시작한다. 이 메뉴를 통해, 잡 디자인, 컨텍스트 및 루틴을 포함한 모든 레파지토리 항목을 만들 수 있다.
Project settings		[프로젝트 설정] 대화 상자를 실행하여 현재 프로젝트에 대한 설명을 추가하고 팔레트의 표시를 사용자 정의 할 수 있다.
Detect and update all jobs		작업에 사용할 수있는 모든 업데이트를 검색한다.
Export Talend projects		[Export Talend projects] 마법사를 시작한다.

2. Main Windows 구성 설명

2.3 Repository 설명

노드명	설명
Job Designs	잡 디자인 폴더는 현재 프로젝트에서 디자인 되어 있는 잡을 트리형식으로 보여준다.
Contexts	파일경로나 DB접속정보 같은 여러 잡에서 재사용할 변수들을 저장한다.
Code	이 프로젝트와 다른 프로젝트에서 사용할 수 있는 코드를 그룹화한 라이브러리이다.
SQL Templates	표준화한 사용자 정의 SQL templates를 생성할 수 있는 폴더이다.
Recycle bin	Repository에서 삭제한 내용을 보관한다. Recycle bin icon을 오른쪽 클릭해서 recycle bin 을 비우기 전까지는 파일 시스템에 삭제된 내용이 존재하고, Recycle bin 에서 바로 삭제된 내용을 복구하거나 완전히 삭제가 가능하다.





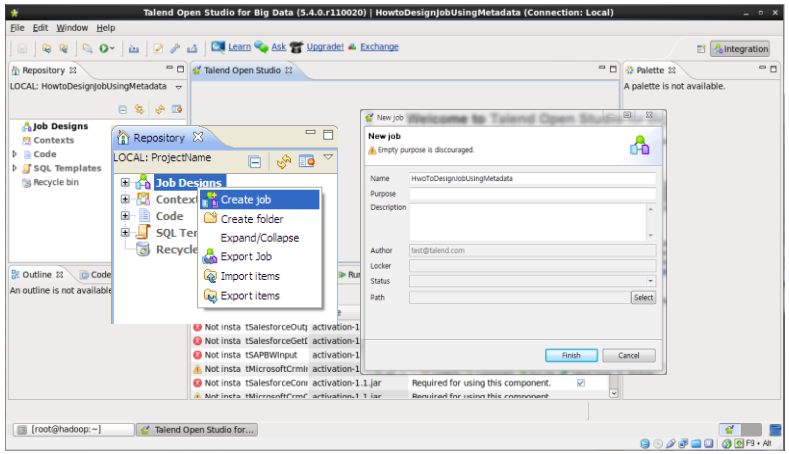
2. Main Windows 구성 설명

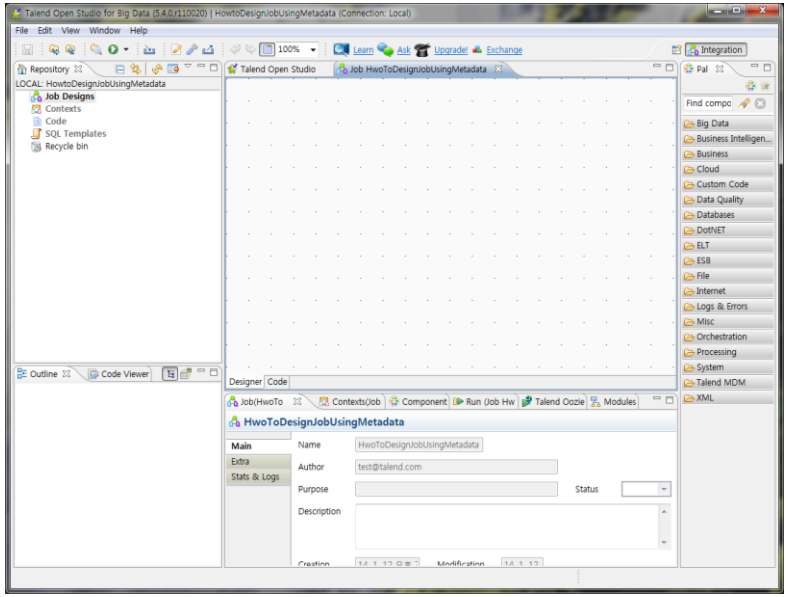
2.4 Configuration 설명

View	설명
Component	Component 뷰는 각 구성 요소에 대한 특정 매개 변수의 자세한 사항들이다. 작업을 구축하기 위해 각각의 컴포넌트가 필요한 필드를 작성해야 한다.
Run Job	Run Job 뷰는 현재 작업의 실행을 보여주고, 실행 후에는 로그 결과를 보여준다.
Oozie scheduler	현재 작업을 실행하거나 원격 HDFS 서버에 주기적으로 실행되도록 예약할 수 있다.
Error Log	Error Log 뷰는 주로 작업 실행 오류에 사용되고 경고나 작업 실행 중에 발생하는 오류의 history를 보여준다. 로그 탭은 또한 Java 구성 요소 운영 진행에 대한 유익한 기능을 가지고 있다. 예를 들어 오류 로그 탭은 기본적으로 숨겨져 있고 Window > Show views 창에서 General 폴더를 확장 후 Error Log를 선택하면 탭이 보여진다.
Modules	작업을 원활하게 수행하기 위하여 컴포넌트에서 필요하거나 참조되는 모듈을 보여준다. 필요한 외부 모듈을 다운받아 설치할 수 있다.
Job view	Job view는 design workspace에 열려 있는 작업과 관련된 다양한 정보를 표시한다. 여기에는 다음과 같은 Tab이 있다.
	Main tab 이 탭은 design workspace에 열려있는 작업에 대한 기본 정보 예를들면 이름, 저자, 버전 번호등을 표시하고, 읽기 전용이고, 이 탭을 편집하기 위해서는 Job을 닫은 후 저장소 트리 뷰에 레이블을 마우스 오른쪽 버튼으로 클릭하고 드롭다운 목록에서 속성 편집을 클릭한다.
	Extra tab 이 탭은 멀티 스레드 실행과 암시적 컨텍스트 로딩 기능을 포함하여 추가 매개 변수를 표시한다.
	Stats/Log tab 이 탭은 모든 작업에 대한 통계 및 로그를 활성화 또는 비활성화 할 수 있고 통계, Log, Volumetrics를 사용여부를 간단히 설정할 수 있다. 또는 Catcher 컴포넌트을 사용하지 않고 한 번에 전체 활성 작업에 대해 이 기능을 설정할 수 있는데, 모든 컴포넌트는 설정에 따라 추적하고 로그를 파일이나 데이터베이스 테이블에 로그를 남길 수 있다. 또한 프로젝트 세팅을 다시 읽어오거나 저장할 수 있다.
	Version tab design workspace에 열려져 있는 잡의 다른 버전에 대하여 생성일자와 변경일자를 보여준다.
Problems	Problems 뷰는 설정의 일부가 누락 된 것과 같은 경우 컴포넌트에 도킹 아이콘에 링크된 메시지를 표시한다. 아이콘 / 메시지에는 Error, Warning and Infos 세 가지 유형이 있다.
Job Hierarchy	선택한 잡의 하위 잡을 보여준다. 이 뷰를 표시하려면, Repository 트리 뷰에서 상위 작업을 오른쪽 클릭하고 드롭 다운 목록에서 Open Job Hierarchy 을 선택하거나, Window > Show view 창에서 Talend > Job Hierarchy을 선택하면 된다. tRunJob 컴포넌트를 통해 상위 잡 및 하나 이상의 하위 잡을 만들 경우에만이 Job Hierarchy를 볼 수 있다.

3. Job Design 실습

3.1 CSV -> XML 파일로 변환





개요 : CSV 파일을 읽어서 XML파일로 변환한다.

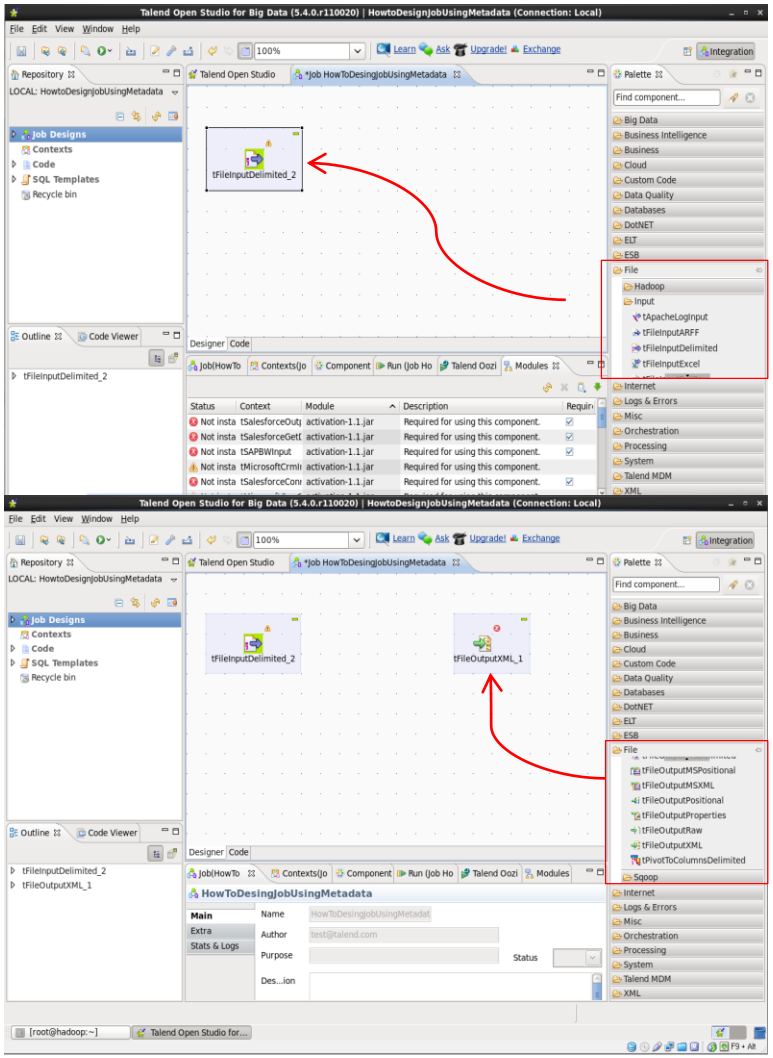
1. 새로운 잡을 만들기 위해서 Job Designs에서 마우스 오른쪽 버튼 클릭 후 Create Job을 클릭한다.

2. Name에 새로운 Job 이름, Purpose, Description을 입력한다.
   Workspace 지정 경로 밑에
   HOWTODESIGNJOBUSINGMETADATA₩process에
   HwoToDesignJobUsingMetadata_0.1.item
   HwoToDesignJobUsingMetadata_0.1.properties
   HwoToDesignJobUsingMetadata_0.1.screenshot
   3개의 파일이 생성된다

3. Job을 Design할 수 있는 메인 화면이 표시된다

3. Job Design 실습

3.1 CSV -> XML 파일로 변환

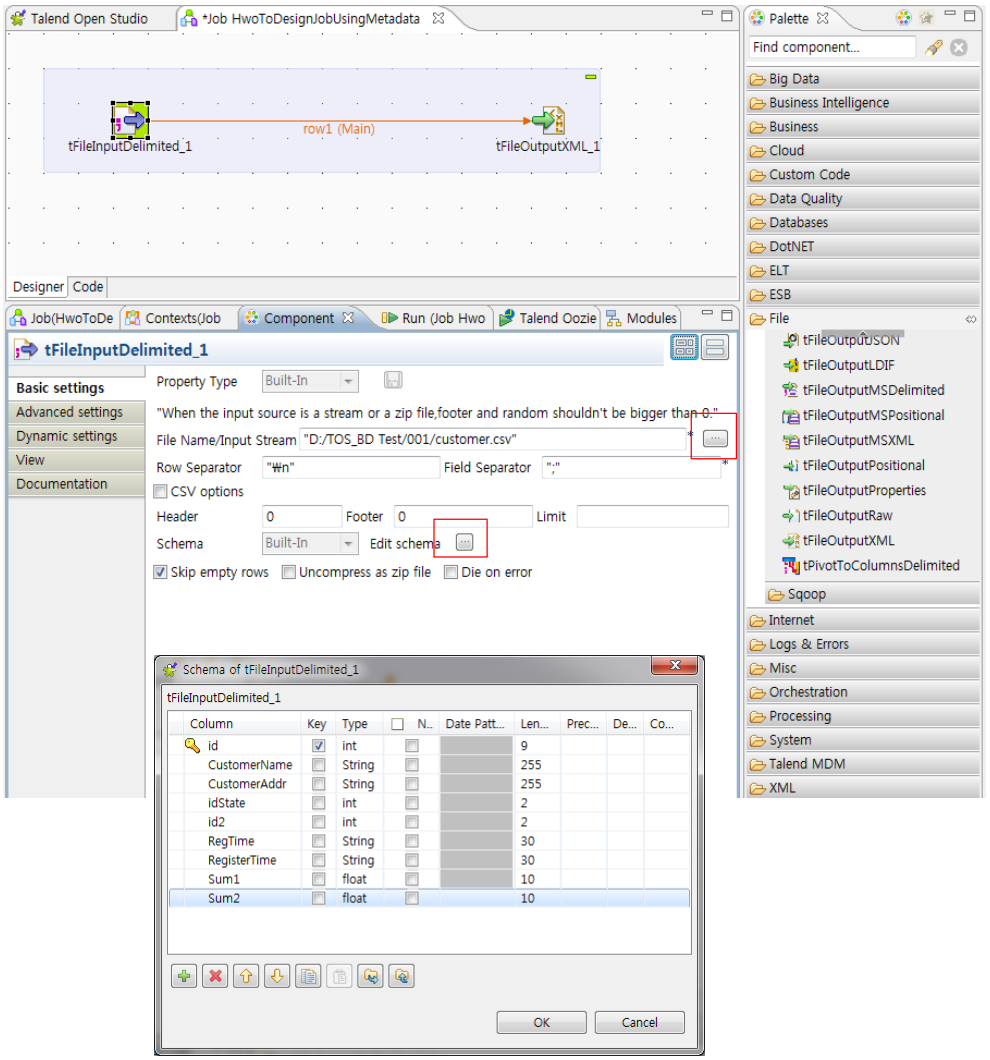


1. 콤마로 구분되어 있는 Input 파일을 사용하기 위하여 Palette-File-Input-tFileInputDelimited Component 선택 후 Job Designer에 드롭한다.

2. XML 형식의 출력파일을 만들기 위하여 Palette-File-Output-tFileOutputXML Component 선택 후 Job Designer에 드롭한다

3. Job Design 실습

3.1 CSV -> XML 파일로 변환



1. 두 컴포넌트를 연결하기 위하여 tFileInputDelimited에서 마우스 오른쪽 버튼을 클릭한 채로 tFileOutputXML로 드래그한다.

2. 버튼을 놓으면 row1(Main) 연결선이 생긴다. tFileInputDelimited에서 마우스 오른쪽 버튼을 클릭 후 row-main을 선택하고 tFileOutputXML을 선택하여도 연결선이 생긴다.

3. Job Designer에서 tFileInputDelimited를 더블클릭하면 아래 부분에Component 판넬이 보인다.

4. Basic settings의 File name/Stream의 [...]버튼에서 input file로 사용할 파일을 선택한다.

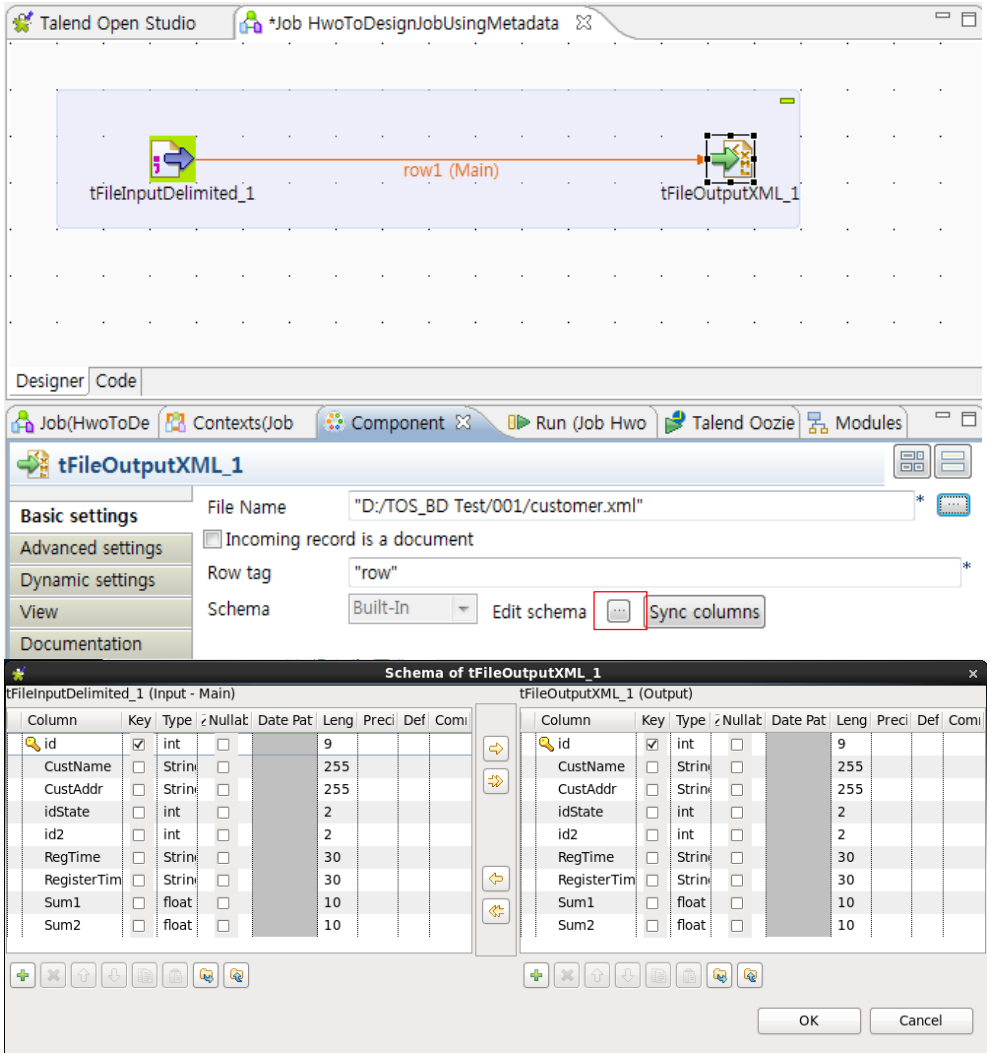
5. Input File의 헤더가 6라인이므로 Header에 6을 입력한다.

6. Edit Schema [...] 버튼을 클릭한다

7. tFileInputDelimited 스키마 창의 아래쪽의 + 키를 눌러서 빈 라인을 생성한 후 Input File에서 사용하는 컬럼 정의 내역을 그림과 같이 입력한다

3. Job Design 실습

3.1 CSV -> XML 파일로 변환

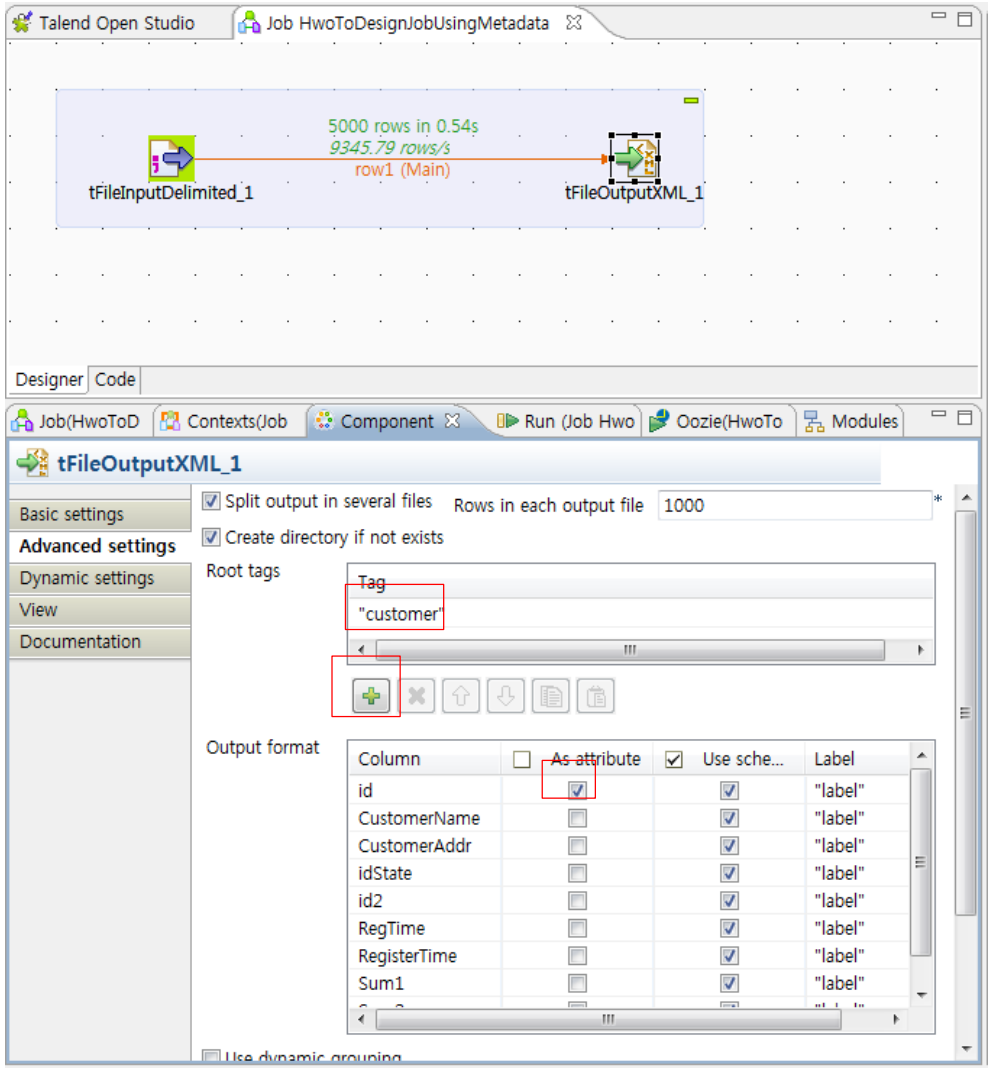


1. tFileOutputXML을 더블클릭한 후 아래의 Component 판넬에서 생성할 XML 파일의 경로와 이름을 입력한다

2. Edit schema의 [...] 을 클릭하여 input / output column 을 확인한다

3. Job Design 실습

3.1 CSV -> XML 파일로 변환



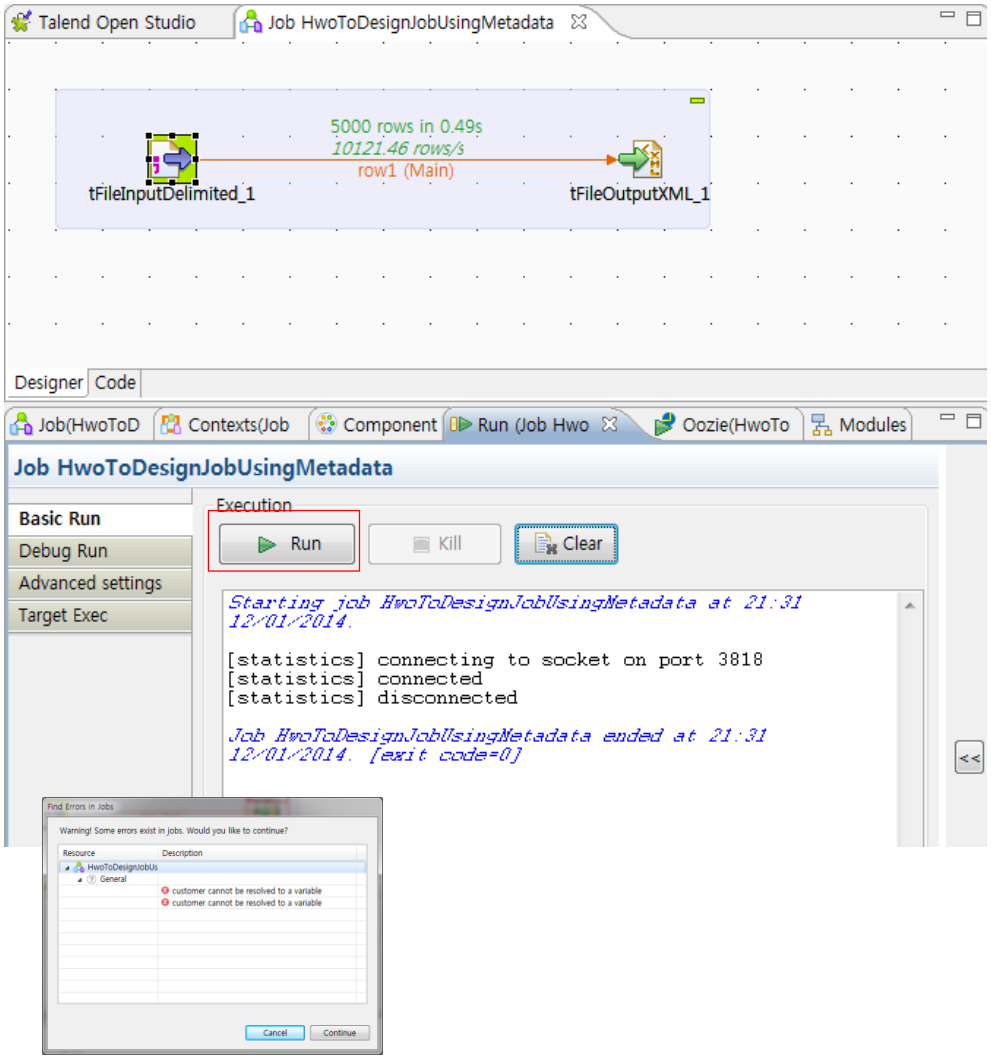
1. Advanced settings에서 1000 Rows마다 파일을 분리하기 위하여 "Split outpout in several files"를 체크한다.

2. Root tags를 입력하기 위하여 [+] 추가 후 "customer"를 입력한다. "를 입력하지 않으면 실행시 오류가 발생하므로 주의해서 입력한다.

3. Output format에서 id 컬럼의 As attribute를 체크한다.

3. Job Design 실습

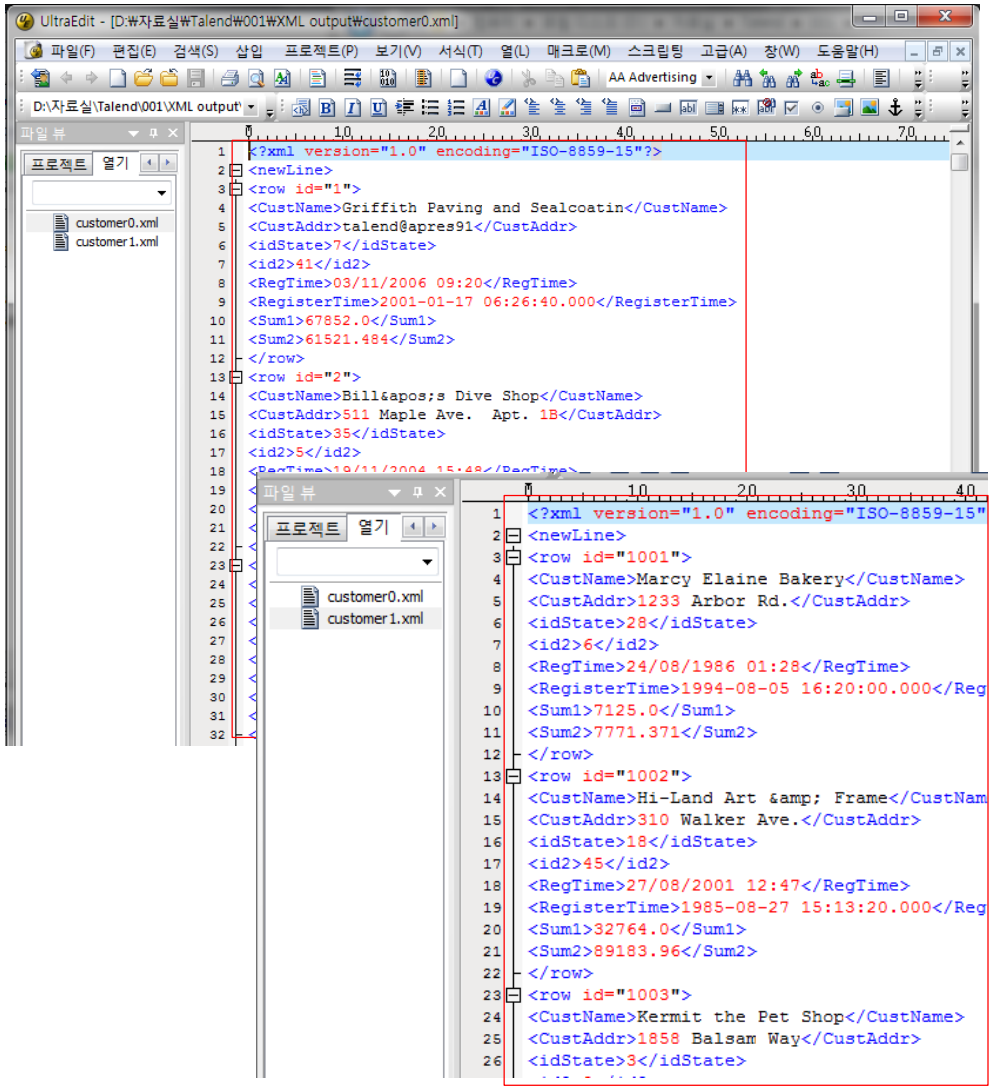
3.1 CSV -> XML 파일로 변환



1. Run 판넬의 Run 버튼을 클릭하여 정상적으로 실행되는지 확인한다. 정상적으로 실행되면 화면과 같은 결과가 표시되고, 오류가 발생한 경우 오류 팝업창이 나타난다.

3. Job Design 실습

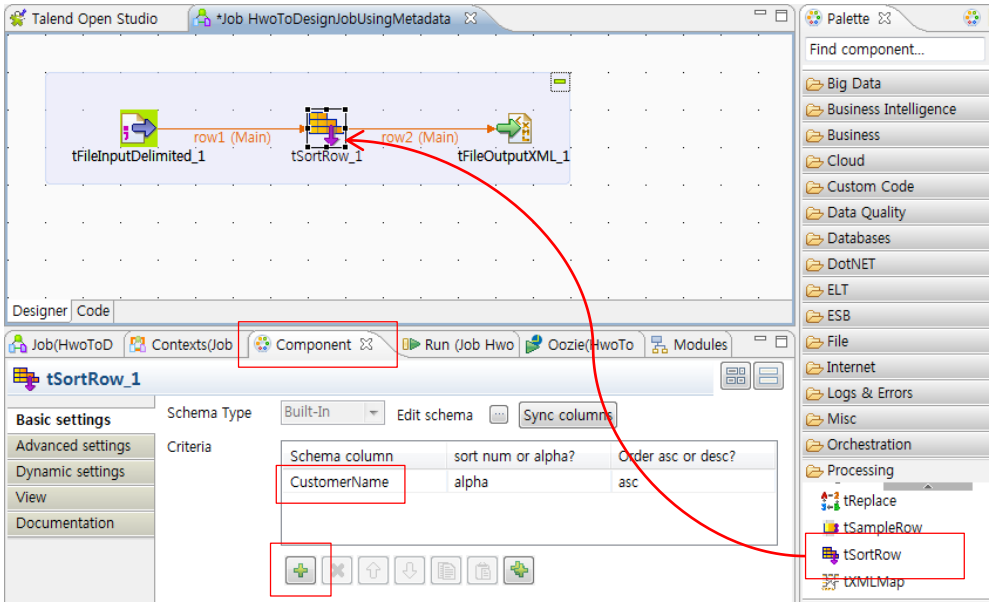
3.1 CSV -> XML 파일로 변환

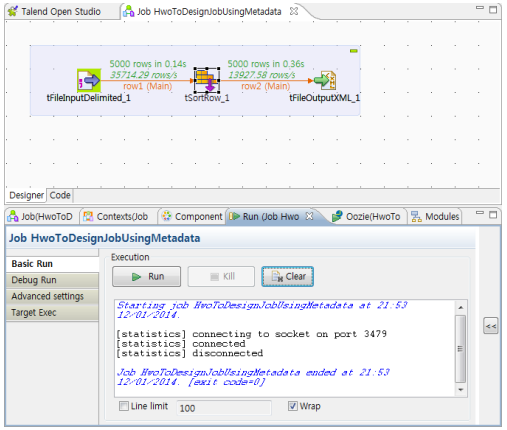


1. tFileOutputXML에서 지정한 저장경로에 가면 customer0.xml ~ customer4.xml 5개의 xml 파일이 생성되어 있고 내용을 확인해 보면 하나의 파일에 id가 1000개씩 저장되어 있다.

3. Job Design 실습

3.2 CSV -> XML 파일로 변환시 정렬하기





개요 : XML파일로 생성하기 전에 고객명으로 정렬한 후 XML 파일로 생성한다.

1. CSV 파일을 XML 파일로 변환시 고객명으로 정렬하여 저장한다
2. Palette/Processing/tSortRow를 드래그하여 row1에 드롭한다
3. tSortRow 이 추가되고 Component에서 "+"를 눌러서 정렬할 컬럼을 추가하고 콤보박스에서 정렬할 컬럼인 CustName와 숫자, 문자 중 alpha를 선택한다.
4. 오름차순은 asc, 내림차순은 desc를 선택하면 된다.
5. Run 판넬의 Run 버튼을 클릭하여 정상적으로 실행되는지 확인한다.

3. Job Design 실습

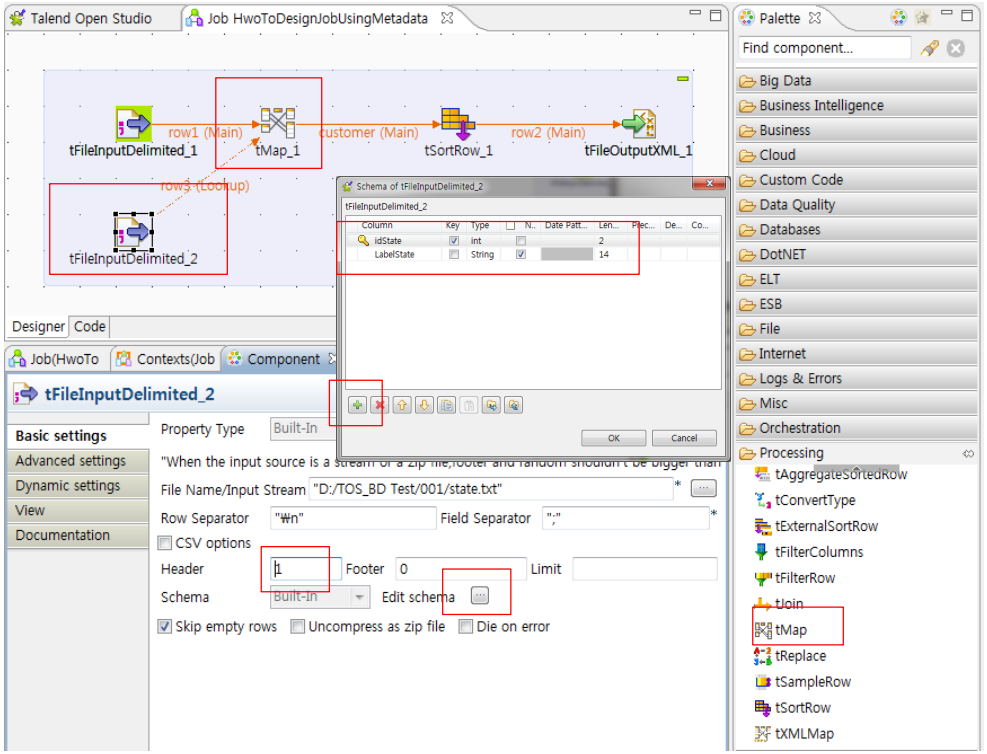
3.2 CSV -> XML 파일로 변환시 정렬하기

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<customer>
<row id="224">
<CustomerName>AA Advertising</CustomerName>
<CustomerAddr>35 Glenlake Dr.</CustomerAddr>
<idState>13</idState>
<id2>30</id2>
<RegTime>18/05/2004 04:47</RegTime>
<RegisterTime>2005-03-03 16:28:16.000</RegisterTime>
<Sum1>75136.0</Sum1>
<Sum2>13600.888</Sum2>
</row>
<row id="313">
<CustomerName>AA Advertising</CustomerName>
<CustomerAddr>245 Oakridge Ave.</CustomerAddr>
<idState>7</idState>
<id2>1</id2>
<RegTime>04/09/2001 16:08</RegTime>
<RegisterTime>2002-10-18 12:06:40.000</RegisterTime>
<Sum1>43120.0</Sum1>
<Sum2>78881.51</Sum2>
</row>
<row id="393">
<CustomerName>AA Advertising</CustomerName>
<CustomerAddr>899 Harvard Ct</CustomerAddr>
<idState>38</idState>
<id2>27</id2>
<RegTime>14/03/2005 02:28</RegTime>
<RegisterTime>1987-09-22 10:06:40.000</RegisterTime>
<Sum1>22565.0</Sum1>
<Sum2>94625.33</Sum2>
</row>
<row id="578">
<CustomerName>AA Advertising</CustomerName>
<CustomerAddr>673 Calais Drive</CustomerAddr>
<idState>25</idState>
<id2>34</id2>
<RegTime>01/10/1997 11:00</RegTime>
<RegisterTime>2005-09-14 15:54:56.000</RegisterTime>
<Sum1>24902.0</Sum1>
<Sum2>62266.684</Sum2>
</row>
```

1. tFileOutputXML에서 지정한 저장경로에 가면 customer0.xml ~ customer4.xml 5개의 xml 파일이 생성되어 있고 내용을 확인해 보면 하나의 파일에 id가 1000개씩 저장되어 있고, 정렬순서로 지정한 CustomerName 순서로 정렬되어 있다

3. Job Design 실습

3.3 CSV -> XML 파일로 맵핑 후 정렬하기



개요 : State id가 있는 컬럼을 State이름으로 맵핑한 후 고객명으로 정렬하여 xml파일로 생성한다.

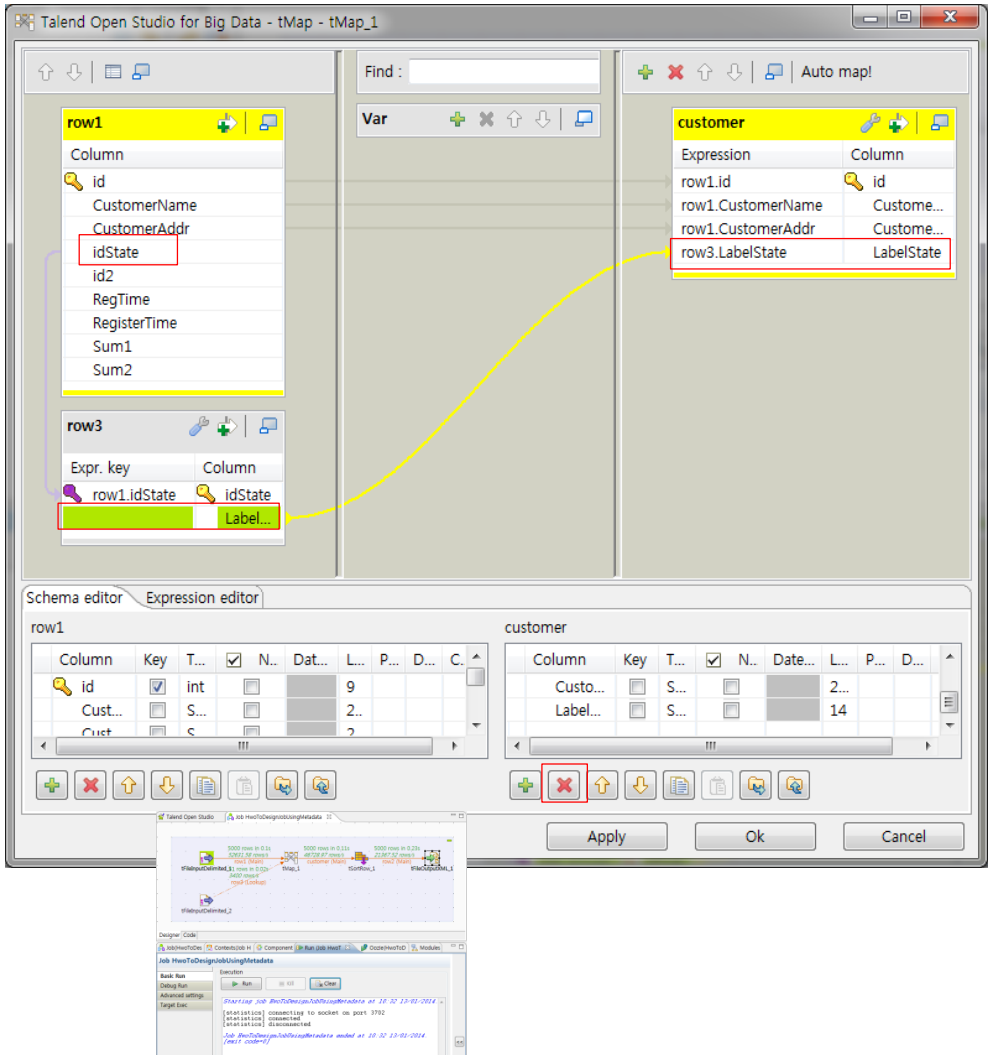
1. State 이름이 있는 Input 파일을 사용하기 위하여 Palette-File-Input-tFileInputDelimited Component 선택 후 Job Designer에 드롭하면 tFileInputDelimited_2가 생성된다. tFileInputDelimited_2를 클릭한 후 Component Tab으로 이동한다. File Name의 … 버튼을 클릭하여 state.txt 파일이 들어있는 폴더를 지정한다. Header 필드에는 헤더라인이 1라인이므로 1을 입력한다. Edit schema 의 … 버튼을 클릭하여 + 버튼을 2번 클릭하여 새로운 라인을 2개 추가 후 State 이름 파일의 스키마를 그림과 같이 입력한다.

2. 고객파일과 State이름을 매핑하기 위하여 Processing-tMap Component를 선택하여 row1 위에 드롭하고, new Output name을 지정하라는 팝업창이 나오면 customer을 입력한다.

3. State 이름이 있는 Input 파일 Component인 tFileInputDelimited_2를 클릭한 후 tMap_1에 드래그&드롭하면 row3(Lookup) 라인이 생성된다.

3. Job Design 실습

3.3 CSV -> XML 파일로 맵핑 후 정렬하기



1. Input 파일을 맵핑하기 위해서 tMap_1을 더블클릭하면 맵핑화면이 나타난다.

2. row1의 idState컬럼을 클릭한 후 row3의 idState에 드롭하면 row1에서 row3를 참조하는 라인이 생긴다.

3. 오른쪽 customer 필드에서 row1.CustomerAddr의 아래 필드를 선택한 후 아래의 X버튼을 눌러서 맵핑필드에서 제외하기 위하여 필드를 삭제한다.

4. idState에 해당하는 이름을 출력하기 위하여 Row3의 Label을 클릭한 후 오른쪽의 customer row1.CustomerAddr아래에 드래그&드롭하면 row3.LavelState가 추가된다.

5. 화면과 같이 설정이 끝나면 Ok 버튼을 클릭하여 Design 화면으로 돌아간다.

6. Run Tab에서 실행하여 정상적으로 실행되는지 확인한다.

3. Job Design 실습

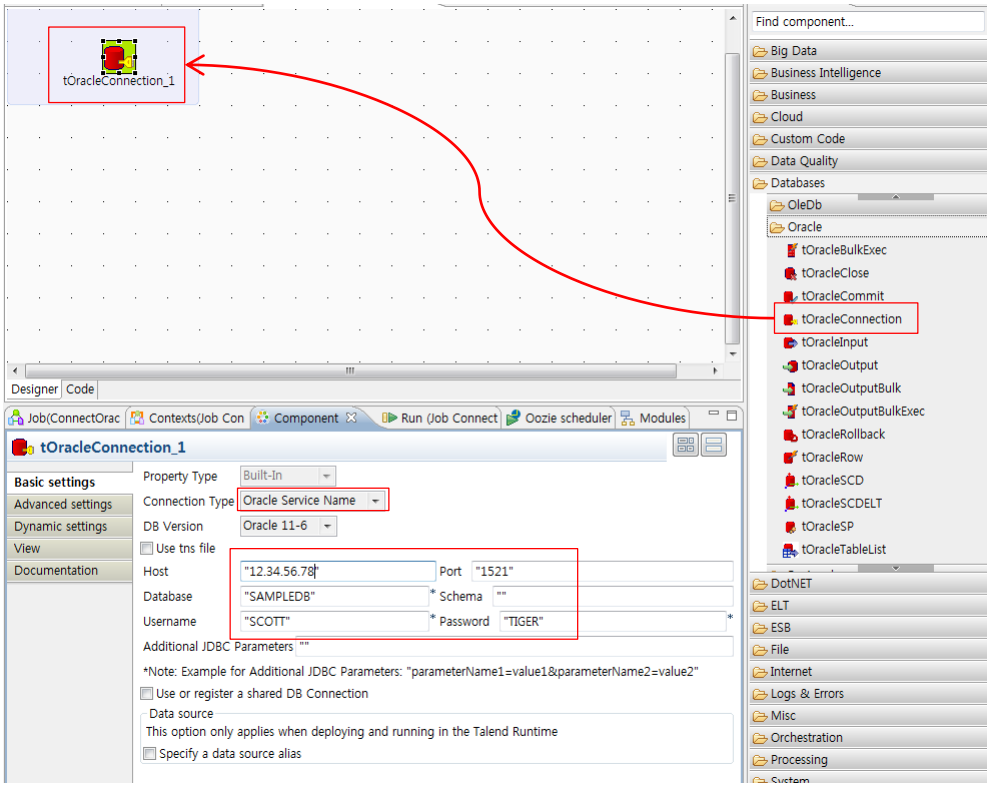
3.3 CSV -> XML 파일로 맵핑 후 정렬하기

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<customer>
<row id="1354">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>750 Homewood Ave.</CustomerAddr>
<LabelState>Hawaii</LabelState>
</row>
<row id="1491">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>43078 Kennedy Dr.</CustomerAddr>
<LabelState>Pennsylvania</LabelState>
</row>
<row id="1722">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>1882 St. Johns</CustomerAddr>
<LabelState>Wisconsin</LabelState>
</row>
<row id="1837">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>1126 Hillcrest Ave</CustomerAddr>
<LabelState>Rhode Island</LabelState>
</row>
<row id="2067">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>14836 W. River Oaks Dr.</CustomerAddr>
<LabelState>Iowa</LabelState>
</row>
<row id="2099">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>1077 Court Ave.</CustomerAddr>
<LabelState>New Hampshire</LabelState>
</row>
<row id="2261">
<CustomerName>Pac Builders</CustomerName>
<CustomerAddr>225 Jeffrys Place</CustomerAddr>
<LabelState>Indiana</LabelState>
</row>
```

1. tFileOutputXML에서 지정한 저장경로에 가면 customer0.xml ~ customer4.xml 5개의 xml 파일이 생성되어 있고 내용을 확인해 보면 하나의 파일에 id가 1000개씩 저장되어 있고, LabelState에 State이름이 추가되어 있는지 확인한다.

4. Oracle 접속 실습

4.1 Oracle DB 접속하기



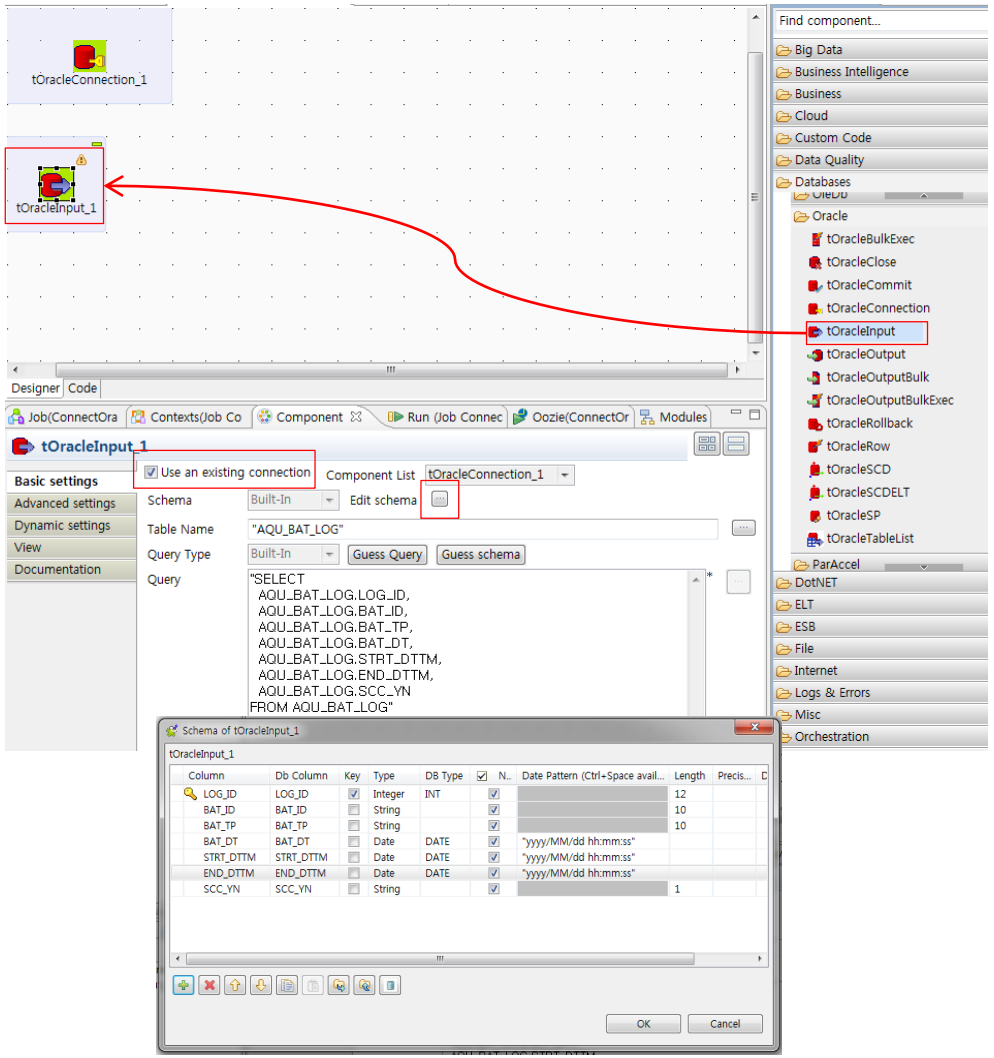
개요 : 오라클 A테이블에서 B테이블로 데이터를 추가한다.

1. 오라클에 접속하기 위해서는 팔레트의 Databases/Oracle/tOracleConnection component를 사용한다. 그림과 같이 tOracleConnection을 Job Designer에 드롭한다.

2. 오라클 접속 방식에 따라서 Connection Type을 선택한다. 여기에서는 Oracle Service Name을 선택한 후 Host, Port, Database, Schema, UserName, Password등 오라클 접속 정보를 입력한다.

4. Oracle 접속 실습

4.1 Oracle DB 접속하기



1. 테이블의 데이터를 가져오기 위해서 팔레트의 Databases/Oracle/tOracleInput component를 선택한 후 Job Designer에 드롭한다.

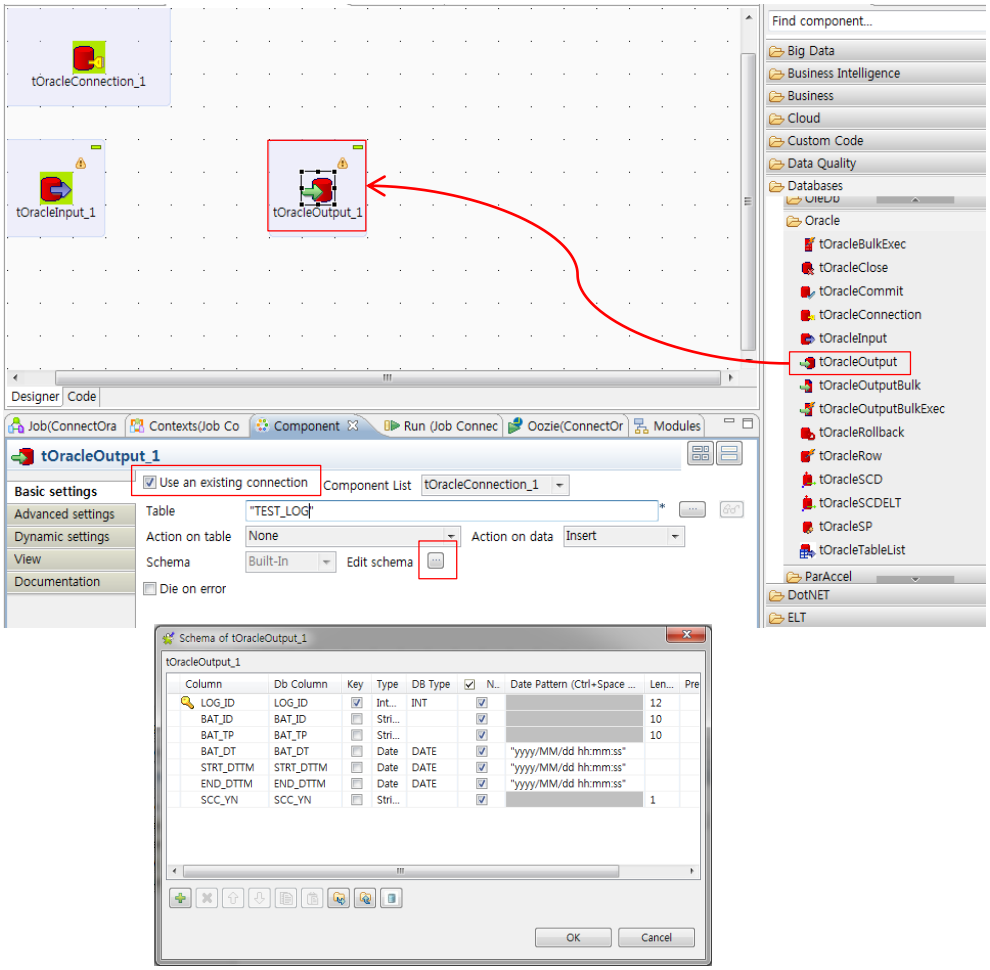
2. tOracleInput _1의 Component 탭에서 tOracleConnection_1 의 오라클 접속정보를 사용하기 위해서 "Use an existing connect"을 체크한다.

3. 조회할 테이블 정보의 스키마를 편집하기 위해 Edit schema 를 누른 후 팝업창에서 테이블의 정보를 입력한다.

4. Guest Query 버튼을 누르면 해당 Query가 표시된다.

4. Oracle 접속 실습

4.1 Oracle DB 접속하기



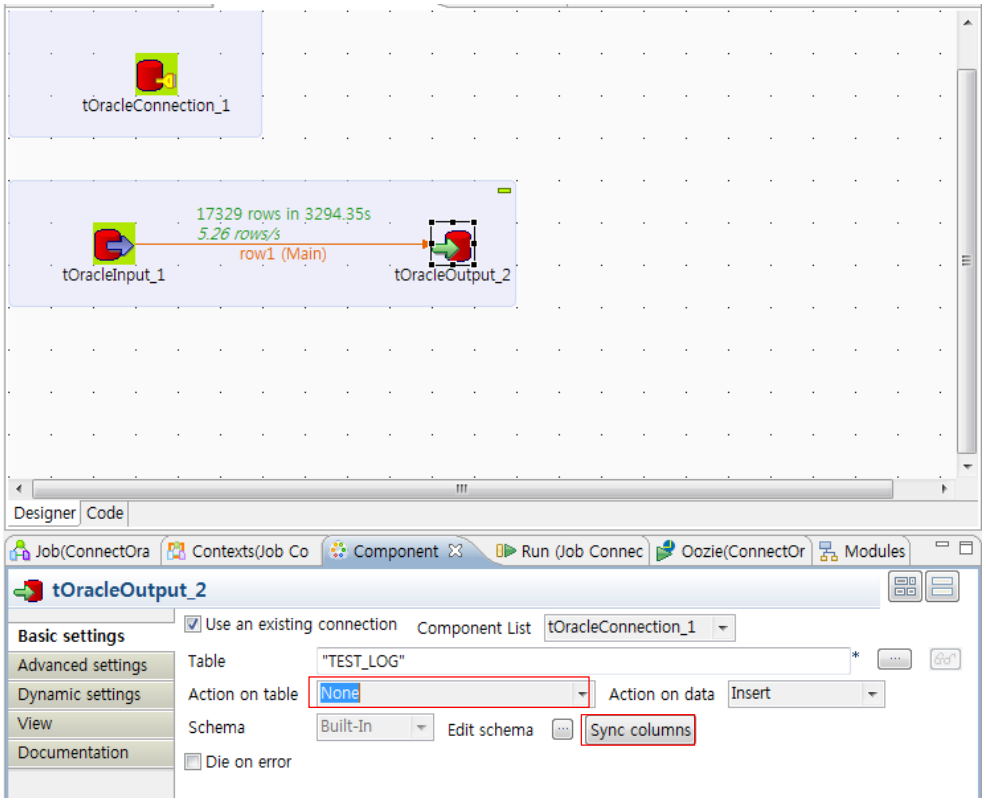
1. 데이터를 저장할 테이블을 위해서 팔레트의 Databases/Oracle/tOracleoOutput component를 선택한 후 Job Designer에 드롭한다.

2. tOracleOutput _1의 Component 탭에서 tOracleConnection_1 의 오라클 접속정보를 사용하기 위해서 "Use an existing connect"을 체크한다.

3. 테이블 정보의 스키마를 편집하기 위해 Edit schema 를 누른 후 팝업창에서 테이블의 정보를 입력한다.

4. Oracle 접속 실습

4.1 Oracle DB 접속하기



1. 두 컴포넌트를 연결하기 위하여 tOracleInput _1 에서 마우스 오른쪽 버튼을 클릭한 채로 tOracleOutput _1 로 드래그한다.

2. 버튼을 놓으면 row1(Main) 연결선이 생긴다. tOracleInput _1 에서 마우스 오른쪽 버튼을 클릭 후 row-main 을 선택하고 tOracleOutput _1 을 선택하여도 연결선이 생긴다.

3. Input data와 Output data를 맵핑하기 위하여 Sync columns을 누르면 자동으로 맵핑하여 준다.

4. Action on table 종류 : Drop and Create table, Create table, Create table if not exists, Drop table if exists and create, Clear table, Truncate table, Truncate table with reuse storage

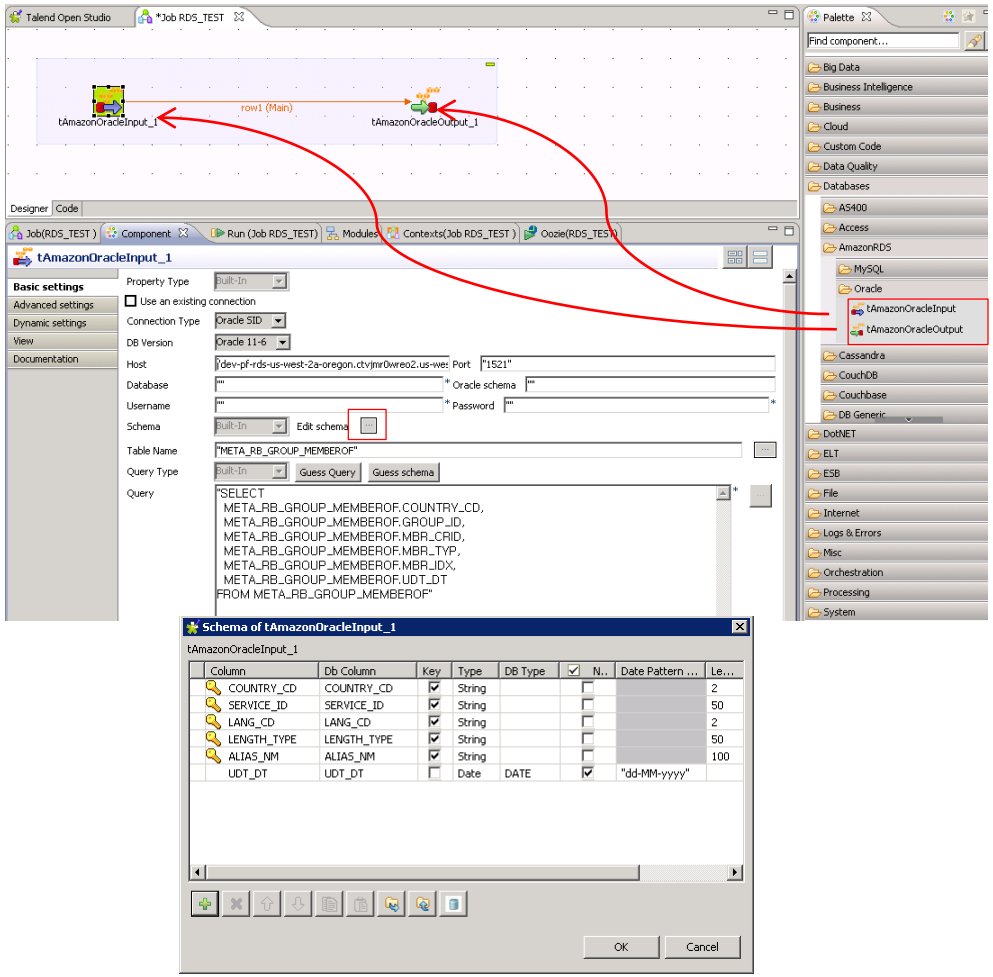
5. Action on data 종류 : Insert, Update, Insert or update, Update or Insert, Delete

6. Run Tab에서 실행

* Input Table schema db에서 가져오는 방법 찾기
* 데이터 추가속도는 초당 5.26건 : 너무 느림!!!

4. Oracle 접속 실습

4.2 AmazonRDS 접속하기



1. AmazonRDS 접속 컴포넌트 tAmazonOracleInput, tAmazonOracleOutput를 드래그하여 Job Designer에 드롭한다.

2. 두 컴포넌트를 연결하기 위하여 tAmazonOracleInput_1 에서 마우스 오른쪽 버튼을 클릭한 채로 tAmazonOracleOutput_1 로 드래그 한다.

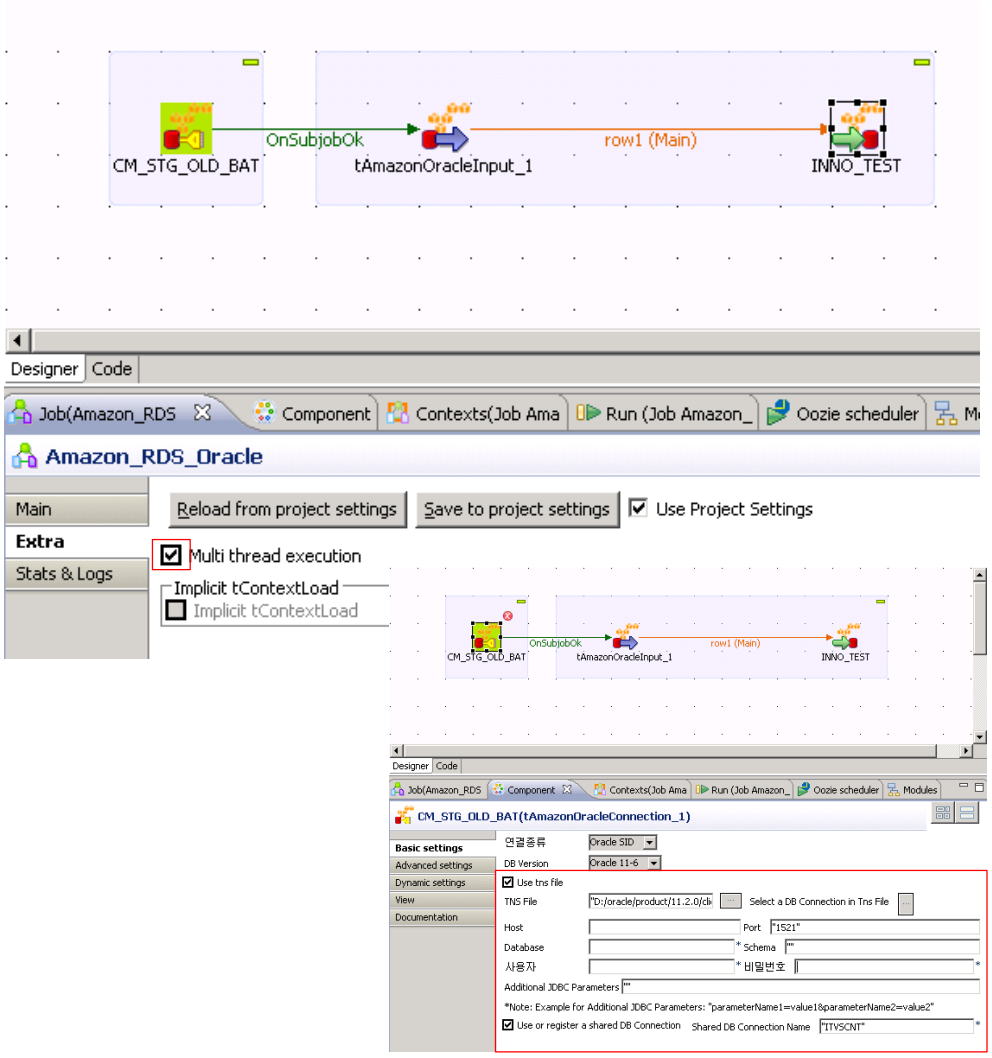
3. AmazonRDS에 접속하기 위하여 tnsname.ora 에 있는 접속정보를 tAmazonOracleInput_1, tAmazonOracleOutput_1 각 필드에 입력 한다.

4. Edit schema 를 누른 후 팝업창에서 테이블의 정보를 입력한 후 Guest Query 버튼을 누르면 해당 Query가 표시된다. 일부 데이터만 가져오려는 경우 Query에서 조건절을 추가로 입력한다.

* 데이터 추가속도는 179,524건 추가시 초당 12,974건 INSERT됨

4. Oracle 접속 실습

4.3 Amazon RDS - Multi thread execution 사용하기



1. AmazonRDS 접속 컴포넌트 tAmazonOracleConnection, tAmazonOracleInput, tAmazonOracleOutput를 드래그하여 Job Designer에 드롭한다.

2. 두 컴포넌트를 연결하기 위하여 tAmazonOracleConnection_1 에서 마우스 오른쪽 버튼을 클릭한 후 trigger/OnSubjobOK 을 클릭 후 tAmazonOracleInput_1 을 클릭하면 OnSubjobOK 선이 연결된다.

3. Multi thread execution을 사용하기 위하여 Job tab의 Extra화면에서 Multi thread execution을 체크한다.

4. tAmazonOracleConnection_1 에 접속하기 위하여 tnsname.ora 에 있는 접속정보를 tAmazonOracleConnection_1 각 필드에 입력한다.

4. Oracle 접속 실습

4.3 Amazon RDS - Multi thread execution 사용하기

1. tAmazonOracleInput_1 Component 탭 Basic settings 화면에서 tAmazonOracleConnection_1을 사용하기 위해서 Use an existing connection을 체크한다.

2. 테이블 정보의 스키마를 편집하기 위해 Edit schema 를 누른 후 팝업창에서 테이블의 정보를 입력한 후 Guest Query 버튼을 누르면 해당 Query가 표시된다.

3. Input data와 Output data를 맵핑하기 위하여 Sync columns을 누르면 자동으로 맵핑하여 준다.

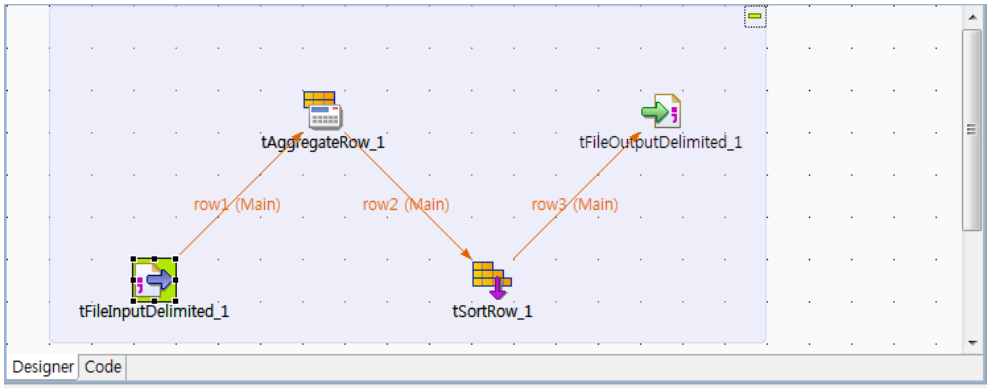
4. 해당 테이블에 들어있는 데이터를 삭제하고 새로 추가하기 위하여 Action on table은 Truncate table을, Action on data는 Insert를 선택한다.

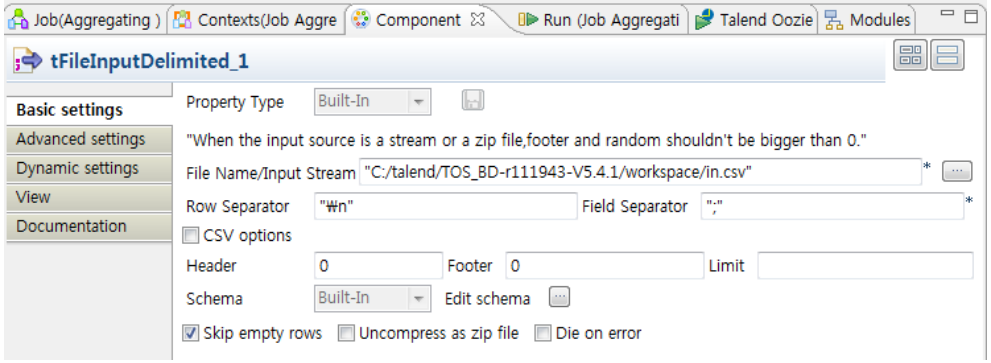
* 623,794건 추가 311.75s초 소요

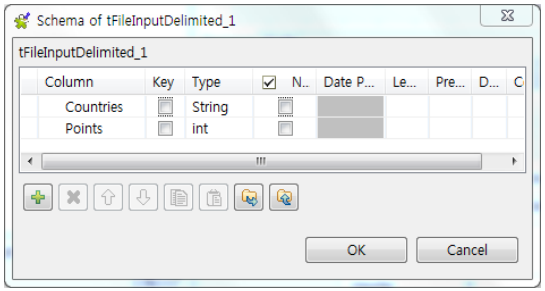
2000.94 rows/s

5. 컴포넌트 사용 실습

5.1 데이터 집계와 정렬하기







국가와 값으로 구성된 CSV파일을 읽어서 국가별로 평균, 최대값, 최소값을 구한 후 정렬하여 CSV파일로 저장한다.

1. 팔레트의 File 폴더에서 tFileInputDelimited 컴포넌트를 design workspace에 drop한다.

2. Basic settings tab 에서 입력할 파일경로를 선택한다

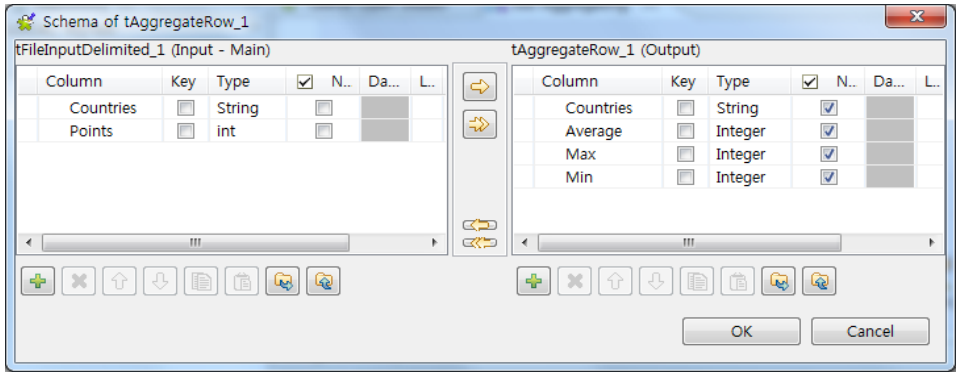
3. Edit schema를 클릭하여 파일구조와 같은 Counties와 Points컬럼을 입력한다.

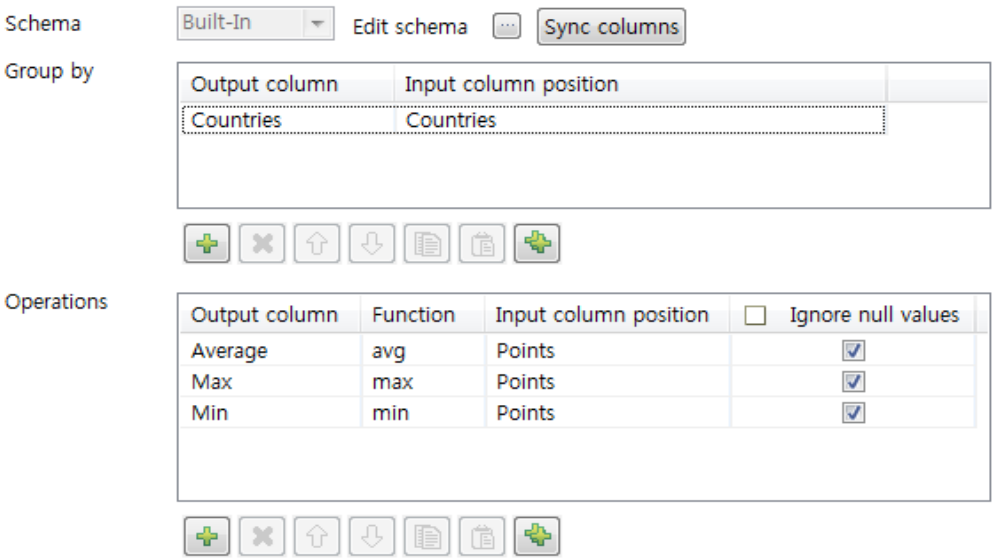
4. Input data

```
England;123
France;5405
England;123
France;540
Germany;937
Ireland;932
Italy;2932;
Spain;3932;
Korea;4339;
Germany;935;
Ireland;932;
Italy;2934;
England;123;
France;540;
Germany;934;
Ireland;932;
England;1232;
France;5404;
Germany;937;
Ireland;9328;
Italy;29358;
Spain;393;
Korea;43397;
Italy;2938;
Spain;3935;
Korea;4339;
Spain;3939;
Korea;4339;
```

5. 컴포넌트 사용 실습

5.1 데이터 집계와 정렬하기

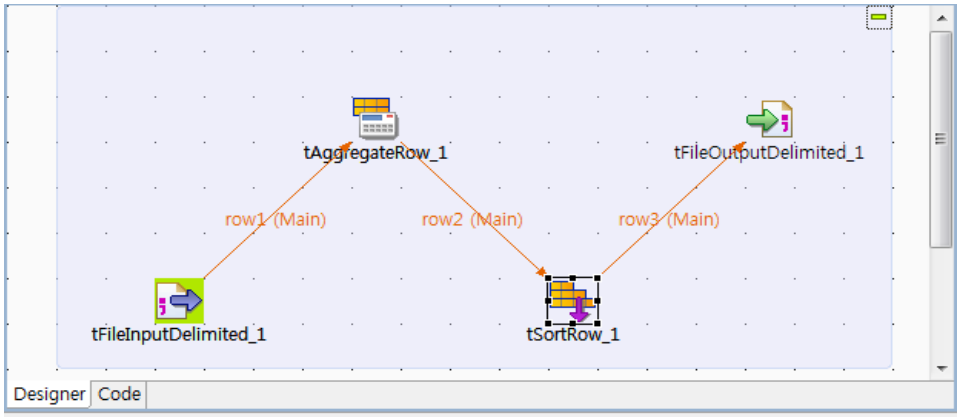


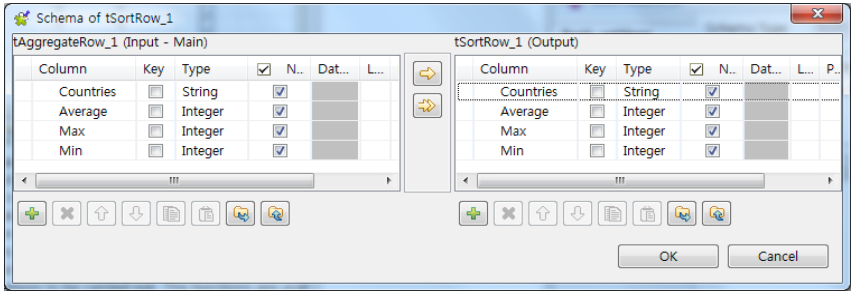


1. 팔레트의 Processing 폴더에서 tAggregateRow 컴포넌트를 design workspace에 drop한다.
2. tFileInputDelimited에서 오른쪽 클린한 후 tAggregateRow 에 드롭하면 row1(Main)선이 연결된다
3. 속성값을 정의하기 위하여 tAggregateRow 컴포넌트를 더블클릭한다.
4. Edit schema 버튼을 눌러서 Output 컬럼을 그림과 같이 추가한다
5. 국가별로 그룹을 주기 위하여 Group by의 Output column과 Input column position을 Countries를 선택한다.
6. 아래의 Operations에서 Output컬럼을 선택한 후 국가별로 평균, 최대값, 최소값을 구하기 위하여 이름에 맞는 Function를 선택하고, Input값으로 받을 Points컬럼을 Input column position에서 Points를 선택한다. Null 값은 무시하기 위하여 Ignore null values를 체크한다.

5. 컴포넌트 사용 실습

5.1 데이터 집계와 정렬하기





1. 팔레트의 Processing 폴더에서 tSortRow 컴포넌트를 design workspace에 drop한다.

2. tAggregateRow 에서 오른쪽 클린한 후 tSortRow 에 드롭하면 row2(Main)선이 연결된다

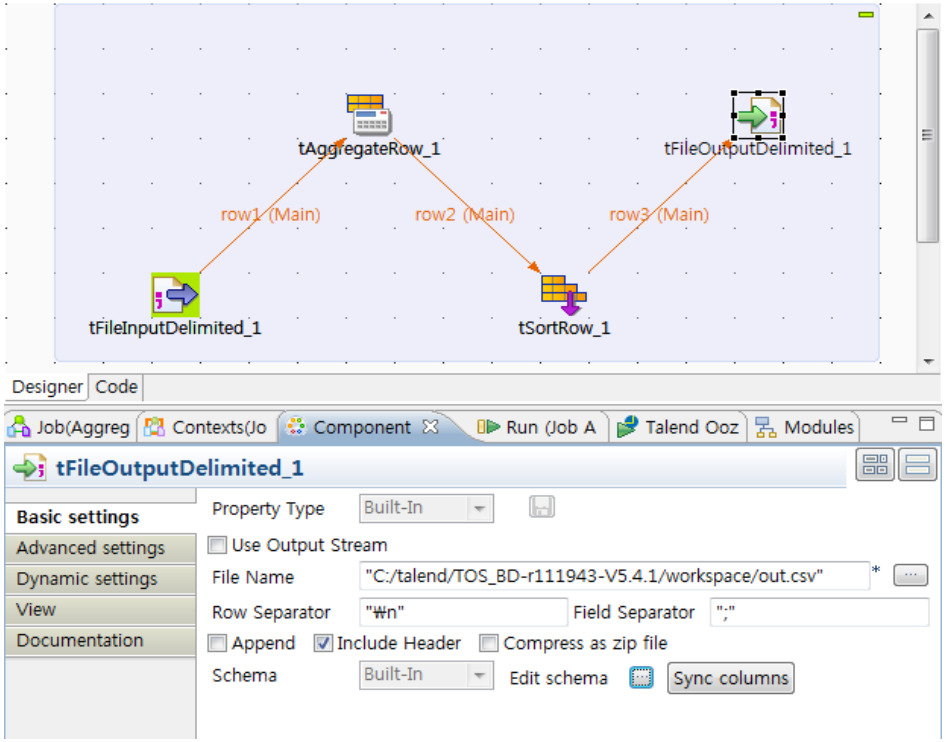
3. 속성값을 정의하기 위하여 tSortRow 컴포넌트를 더블클릭한다.

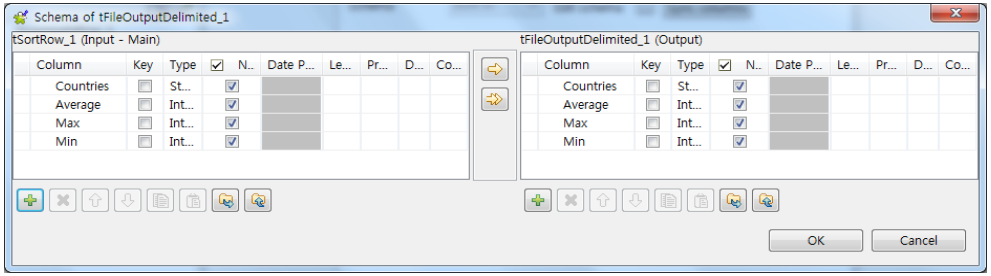
4. Input column과 output column을 맞추기 위하여 Sync columns 버튼을 누른 후 Edit schema 버튼을 눌러서 Input, Output 컬럼이 제대로 정의되었는지 확인한다.

5. 국가별로 정렬하기 위하여 Criteria의 Schema column에서 Country를 선택하고 문자, 숫자로 정렬할 지 선택하고, 오름차순, 내림차순을 선택한다.

5. 컴포넌트 사용 실습

5.1 데이터 집계와 정렬하기



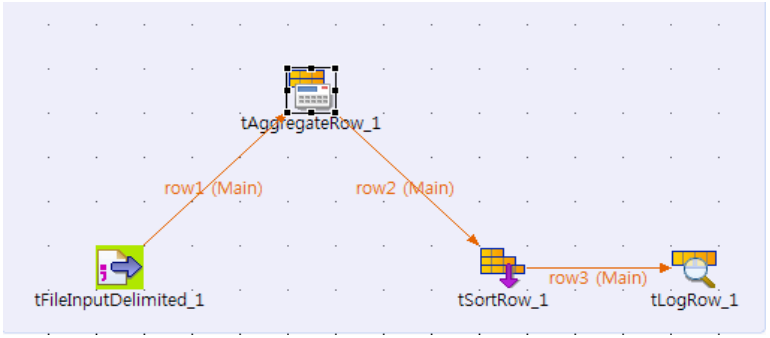


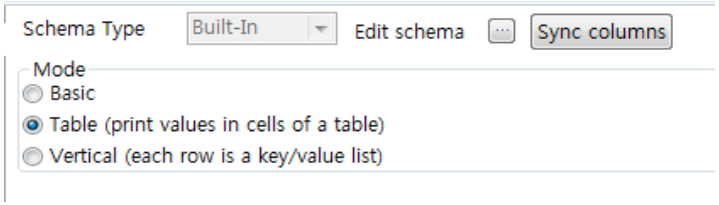
1. 팔레트의 Processing 폴더에서 tFileOutputDelimited 컴포넌트를 design workspace에 drop한다.
2. tSortRow 에서 오른쪽 클린한 후 tFileOutputDelimited 에 드롭하면 row3(Main)선이 연결된다
3. 속성값을 정의하기 위하여 tFileOutputDelimited 컴포넌트를 더블클릭한다.
4. File Name에서 출력할 폴더와 파일이름을 선택한다.
5. 출력파일에서 헤더를 추가하기 위해서 Include Header를 클릭한다.
6. Input column과 output column을 맞추기 위하여 Sync columns 버튼을 누른 후 Edit schema 버튼을 눌러서 Input, Output 컬럼이 제대로 정의되었는지 확인한다.
7. 실행을 하면 위에서 정의한 파일에 그림과 같은 결과파일이 생성된다.
8. Input data

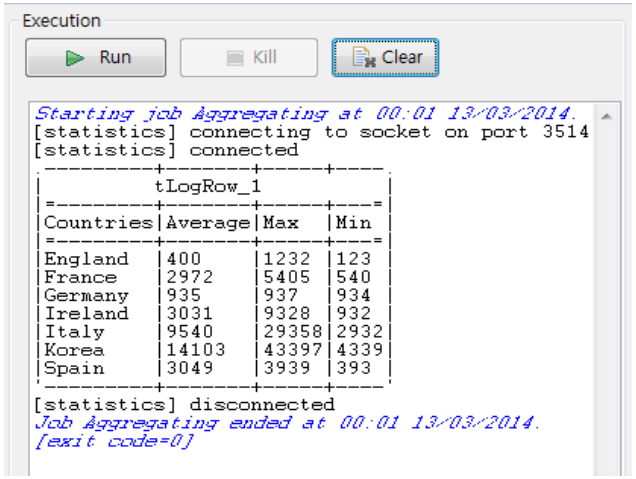
```
Countries;Average;Max;Min
England;400;1232;123
France;2972;5405;540
Germany;935;937;934
Ireland;3031;9328;932
Italy;9540;29358;2932
Korea;14103;43397;4339
Spain;3049;3939;393
```

5. 컴포넌트 사용 실습

5.1 데이터 집계와 정렬하기



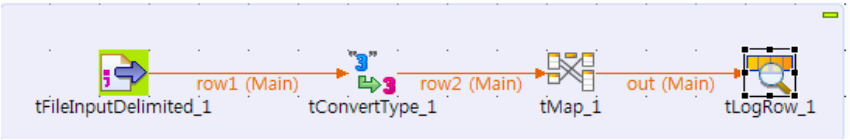


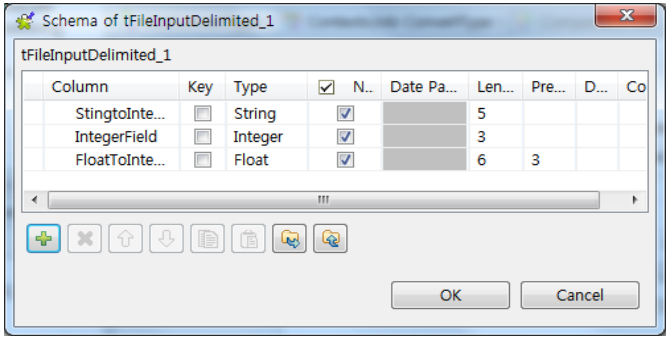


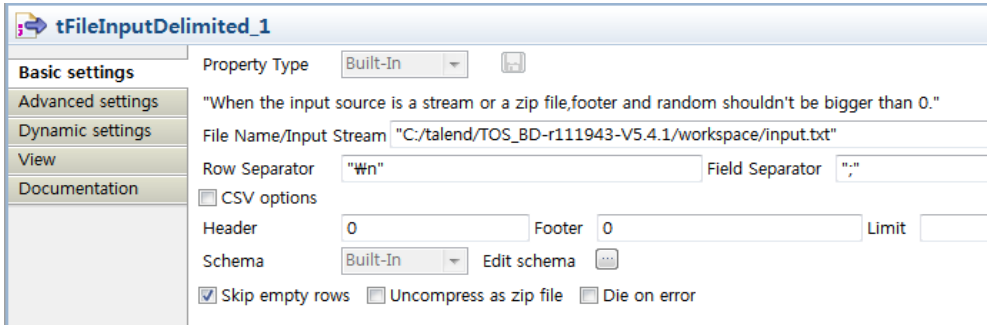
1. 결과를 화면으로 보고 싶으면 팔레트의 Logs&Errors폴더에서 tLogRow 컴포넌트를 design workspace에 drop한다.
2. tSortRow 에서 오른쪽 클린한 후 tLogRow 에 드롭하면 row3(Main) 선이 연결된다
3. 속성값을 정의하기 위하여 tLogRow 컴포넌트를 더블클릭한다.
4. Input column과 output column을 맞추기 위하여 Sync columns 버튼을 누른 후 Edit schema 버튼을 눌러서 Input, Output 컬럼이 제대로 정의되었는지 확인한다.
5. 출력모드에서 테이블 양식으로 출력하기 위해서 Table을 선택한다.
6. 실행을 하면 그림과 같이 결과화면이 출력된다.

5. 컴포넌트 사용 실습

5.2 데이터 타입 변환

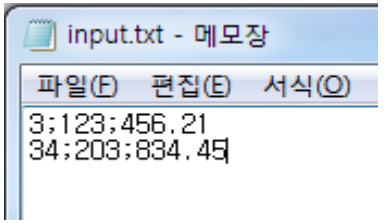






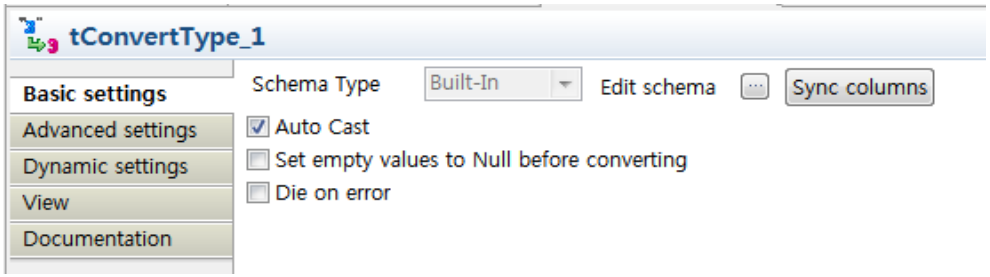
3개의 컬럼으로 구성된 CSV파일을 읽어서 첫번째 컬럼은 숫자타입으로, 두번째 컬럼과 세번째 컬럼의 합계를 구하여 출력파일로 저장한다.

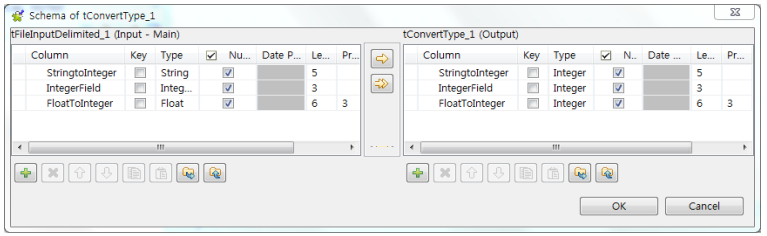
1. 팔레트의 File 폴더에서 tFileInputDelimited 컴포넌트를 design workspace에 drop한다.
2. Basic settings tab 에서 입력할 파일경로를 선택한다
3. Edit schema를 클릭하여 파일구조와 같이 컬럼과 타입을 입력한다.
4. 팔레트의 File 폴더에서 tConvertType_1컴포넌트를 design workspace에 drop한다.
5. Basic settings tab 에서 입력할 파일경로를 선택한 tFileInputDelimited에서 오른쪽 클린한 후 tConvertType_1 에 드롭하면 row1(Main)선이 연결된다
6. Input data

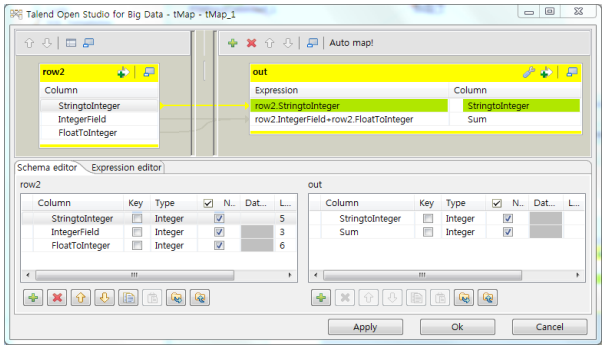


5. 컴포넌트 사용 실습

5.2 데이터 타입 변환



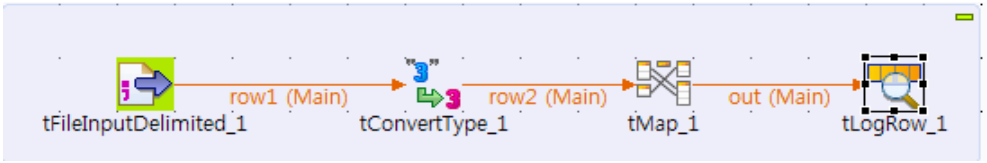


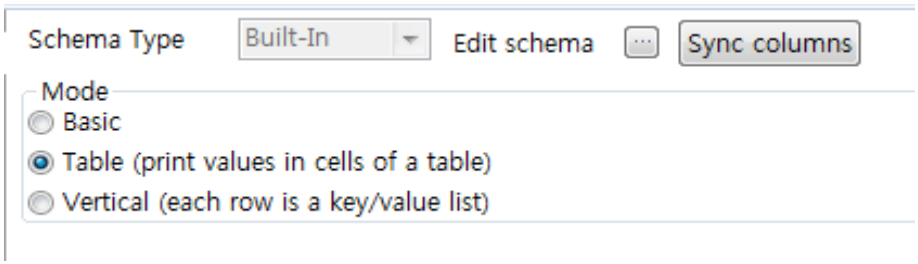


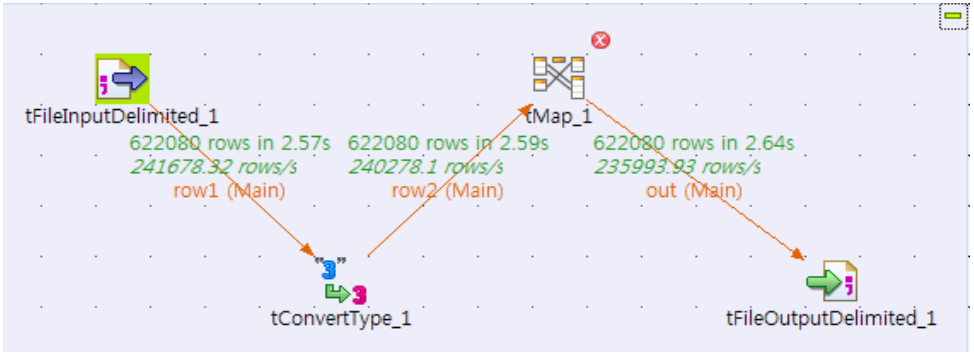
1. 속성값을 정의하기 위하여 tConvertType_1 컴포넌트를 더블 클릭한다.
2. Input과 Output 컬럼을 맞추기 위하여 Sync columns을 클릭 후 Edit schema를 누른다
3. 그림과 같이 컬럼이 매칭되는지 확인하고 Output 컬럼의 Type는 Integer로 변경한다.
4. 팔레트의 Processing 폴더에서 tMap컴포넌트를 design workspace에 drop한다.
5. tConvertType_1 에서 오른쪽 클린한 후 tMap 에 드롭하면 row2(Main)선이 연결된다.
6. tMap을 더블 클릭한 후 팝업창이 뜨면 그림과 같이 설정한다. Output의 Sum Column은 입력 컬럼 IntegerField와 FloatToInteger의 합계를 구하기 위하여 Expression에 row2.IntegerField + row2.FloatToInteger 를 입력한다.

5. 컴포넌트 사용 실습

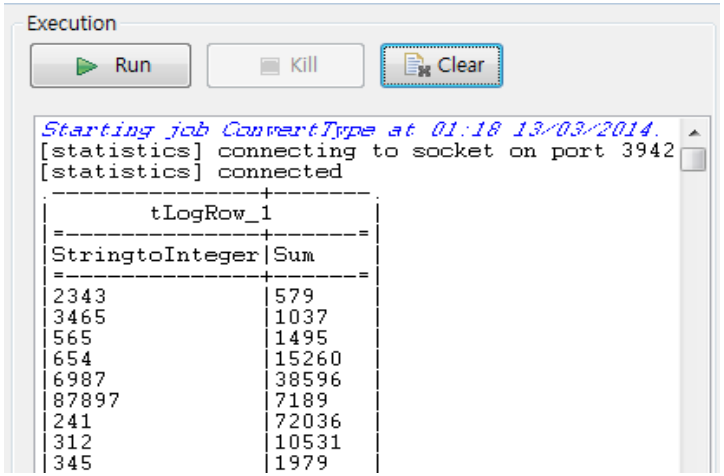
5.2 데이터 타입 변환





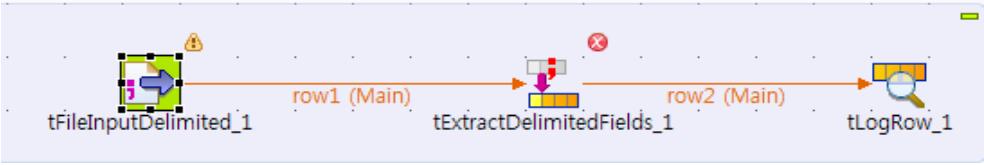


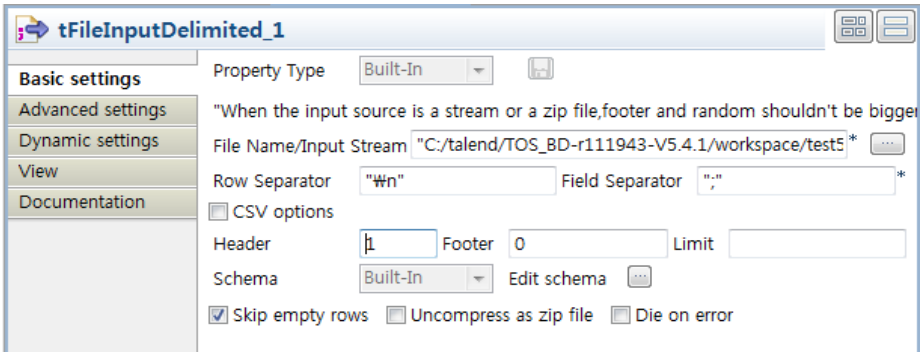
1. 결과를 화면으로 출력하기 위하여 팔레트의 Logs&Errors폴더에서 tLogRow 컴포넌트를 design workspace에 drop한다.
2. tMap에서 오른쪽 클린한 후 tLogRow 에 드롭하면 out(Main)선이 연결된다
3. 속성값을 정의하기 위하여 tLogRow 컴포넌트를 더블클릭한다.
4. Input column과 output column을 맞추기 위하여 Sync columns 버튼을 누른 후 Edit schema 버튼을 눌러서 Input, Output 컬럼이 제대로 정의되었는지 확인한다.
5. 출력모드에서 테이블 양식으로 출력하기 위해서 Table을 선택한다.
6. 실행을 하면 그림과 같이 결과화면이 출력된다.

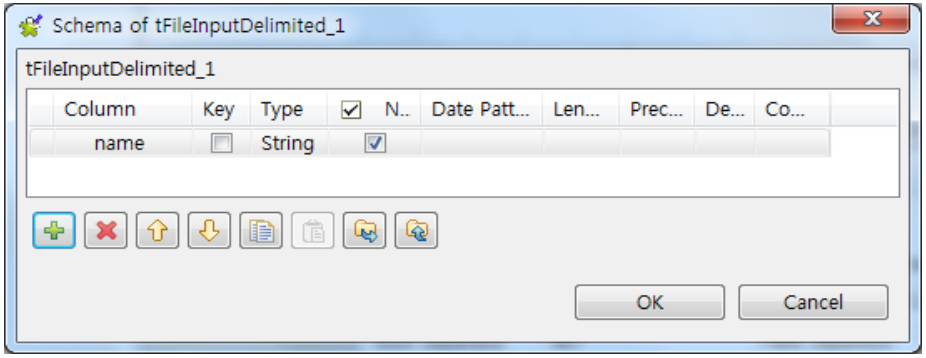


5. 컴포넌트 사용 실습

5.3 한 필드가 컴마로 구분된 파일 추출하기







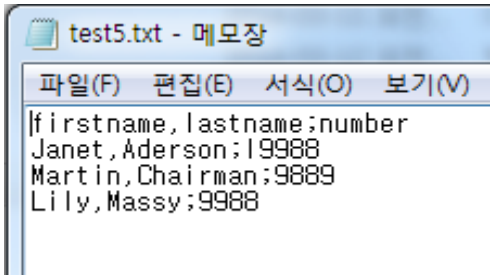
컴마로 구분되어 있는 파일을 읽어서 두 컬럼을 추출한다.

1. 팔레트의 File폴더에서 tFileInputDelimited 컴포넌트를 design workspace에 drop한다.

2. Basic settings tab 에서 입력할 파일경로를 선택한다. Input 파일에 헤더가 1라인 있으므로 Header에 1을 입력한다.

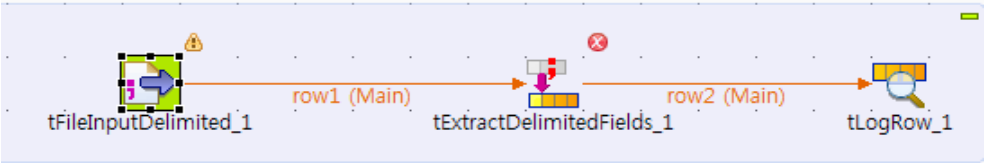
3. Edit schema를 클릭하여 파일구조와 같이 컬럼과 타입을 입력한다.

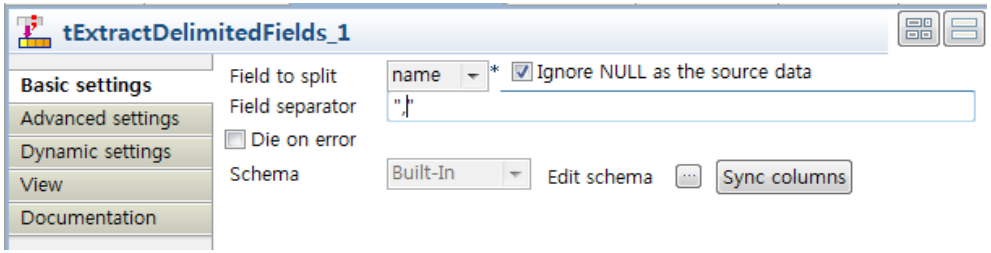
4. Input data

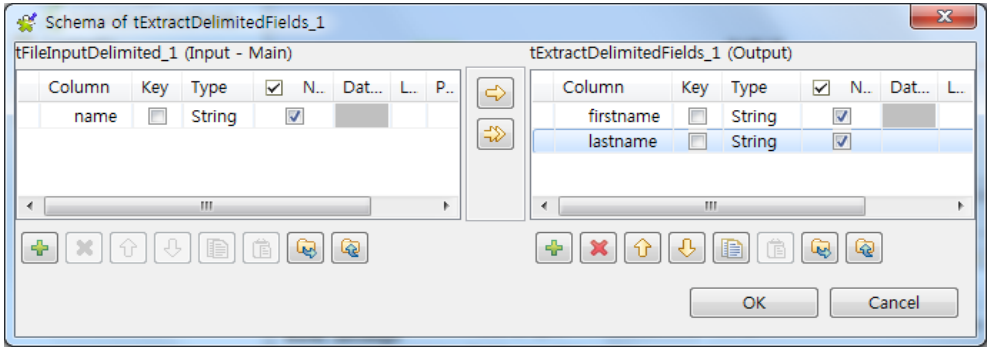


5. 컴포넌트 사용 실습

5.3 한 필드가 컴마로 구분된 파일 추출하기





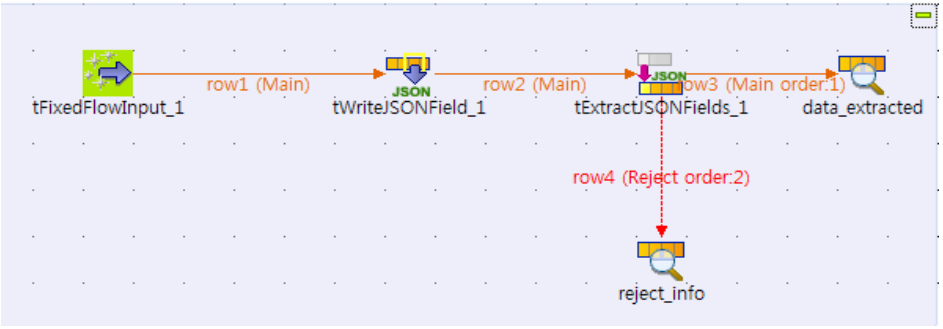


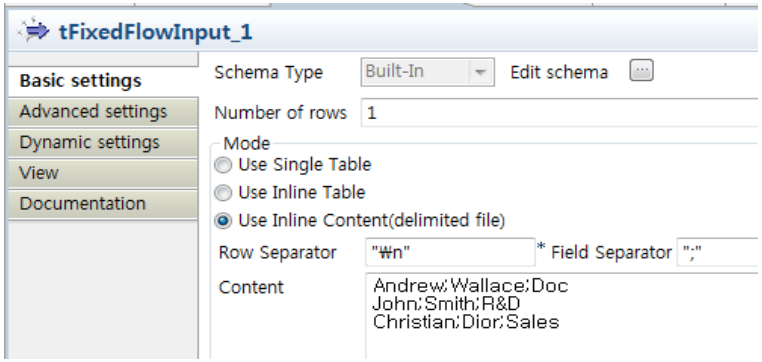
1. 팔레트의 Processing/Fields폴더에서 **tExtractDelimitedFields** 컴포넌트를 design workspace에 drop한다.
2. tFileInputDelimited에서 오른쪽 클린한 후 **tExtractDelimitedFields**에 드롭하면 row1(Main)선이 연결된다
3. 속성값을 정의하기 위하여 **tExtractDelimitedFields** 컴포넌트를 더블클릭한다.
4. 분리할 필드명을 name을 선택하고, 필드 구분자는 "," 를 입력한다.
5. Edit schema를 클릭하여 Output 에 firstname, lastname을 추가한다.
6. 실행을 하면 그림과 같이 firstname과 lastname이 구분되어 출력된다.

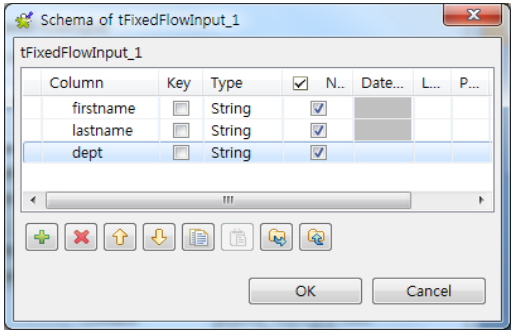
```
Starting job ExtractDelimitedFields at 01:56
13/03/2014.
[statistics] connecting to socket on port 3646
[statistics] connected
.---------+--------.
|     tLogRow_1      |
|=---------+-------=|
|firstname|lastname|
|=---------+-------=|
|Janet     |Aderson  |
|Martin    |Chairman|
|Lily      |Massy    |
'---------+--------'

[statistics] disconnected
Job ExtractDelimitedFields ended at 01:56
13/03/2014. [exit code=0]
```

5. 컴포넌트 사용 실습

5.4 Error messages 출력하기







1. 팔레트의 Misc폴더에서 tFixedFlowInput 컴포넌트를, Misc폴더에서 tWriteJSONField를 Processing/Fields폴더에서 tExtractJSONFields 컴포넌트를, Logs&Errors폴더에서 tLogRow 2개를 design workspace에 drop한다.

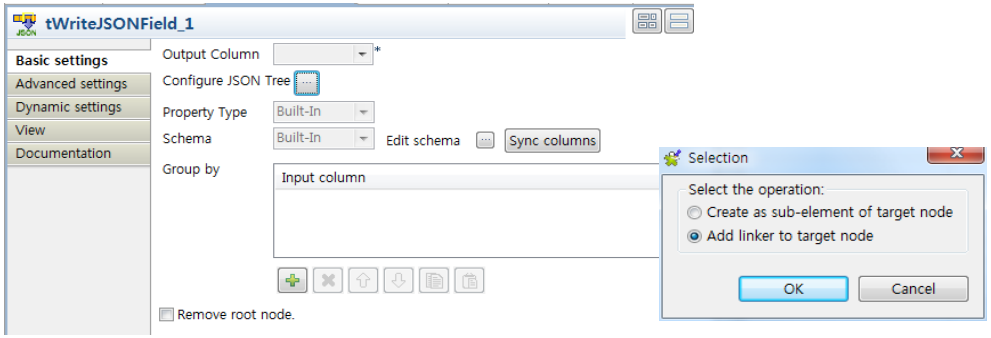
2. 그림과 같이 각 컴포넌트를 Main으로 연결하고, tExtractJSONFields and reject_info 는 Reject로 연결한다.

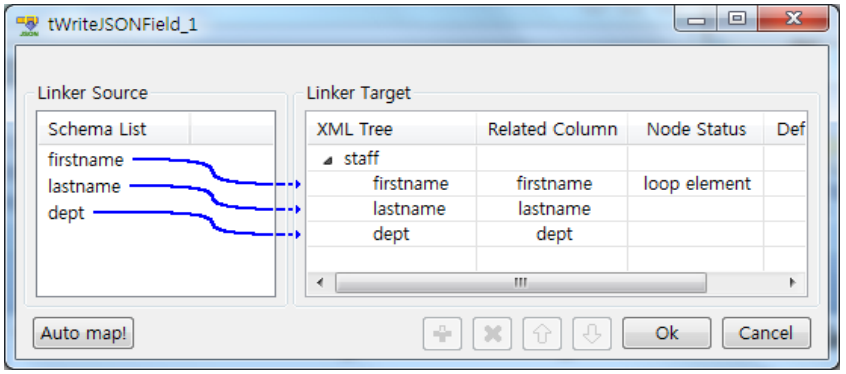
3. tFixedFlowInput_1의 Basic settings tab 에서 Use Inline Content를 선택하고, content 내용을 그림과 같이 입력한다.

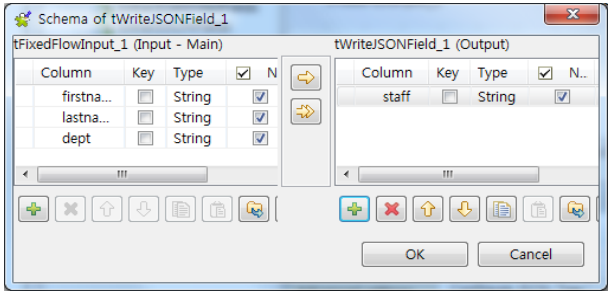
4. Edit schema를 클릭하여 파일구조와 같이 컬럼과 타입을 입력한다.

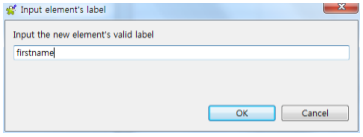
5. 컴포넌트 사용 실습

5.4 Error messages 출력하기





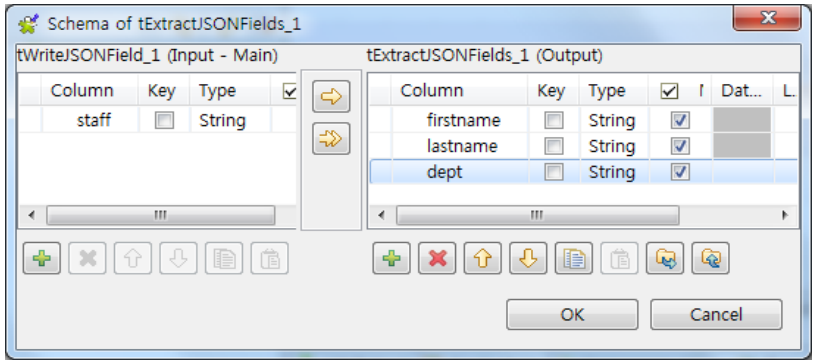


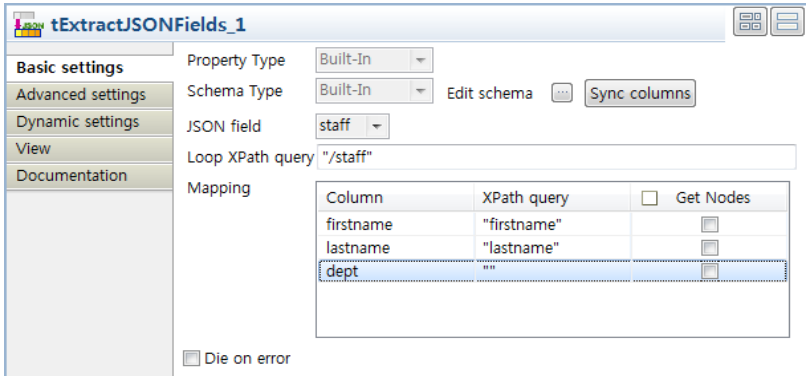


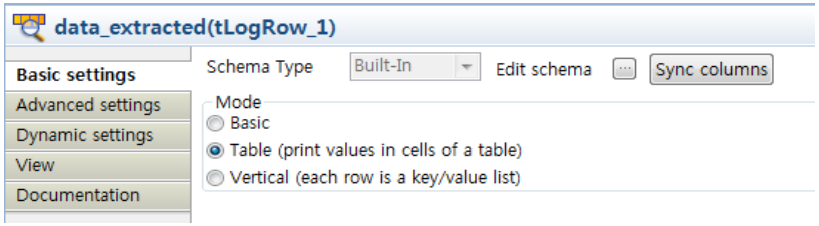
1. Basic settings tab 에서 Configure JSON Tree 를 클릭하여 XML tree editor를 연다
2. tFixedFlowInput 스키마 창의 Linker target 판넬에서 XML Tree에 rootTag로 되어 있는 부분을 staff로 수정한다. 이 부분이 JSON field 의 루트 노드이다
3. Staff에서 오른쪽 클릭하고 context menu 에서 Add Sub-element 를 선택한다.
4. 팝업창에서 sub-node 이름으로 firstname을 입력한다. 같은 방법으로 *lastname과 dept*를 입력한다.
5. firstname 에서 오른쪽 클릭 후 나온 context menu 에서 Set As Loop Element 를 선택한다.
6. Linker source에서 firstname 을 Linker target 의 firstname 에 드롭한다.
7. Selection pop-up dialog box가 나타나면 Add linker to target node 를 선택한다. 같은 방법으로 *lastname과 dept도 연결한다.*
8. Edit schema을 눌러서 오른쪽 판널 tWriteJSONField_1에서 JSON 데이터를 가져올 필드로 staff라고 입력한다.

5. 컴포넌트 사용 실습

5.4 Error messages 출력하기







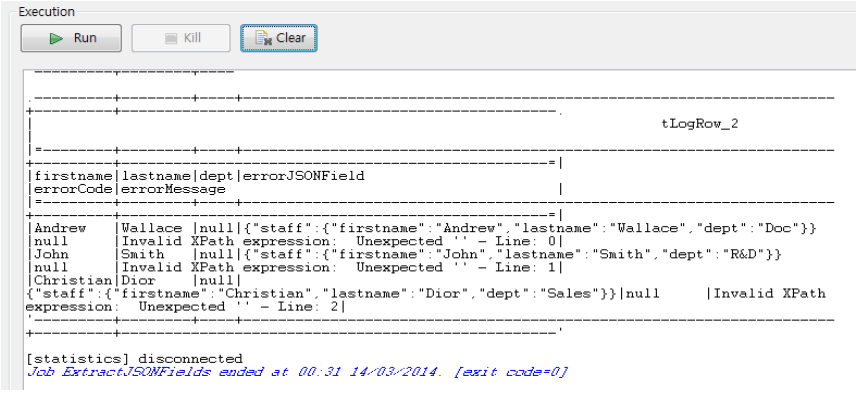
1. tExtractJSONFields_1의 Basic settings tab 에서 Edit schema 창에서 오른쪽 판넬에 *firstname, lastname,dept를 입력한다. 이 부분은 JSON 필드 staff의 노드가 된다*

2. Loop XPath query field에서 JSON data의 루트 노드로 *"/staff"*를 입력한다.

3. Mapping 부분의 XPath query컬럼에 JSON data 의 노드이름을 입력한다. Dept 부분의 ""은 에러 처리부분으로 빠질것이다.

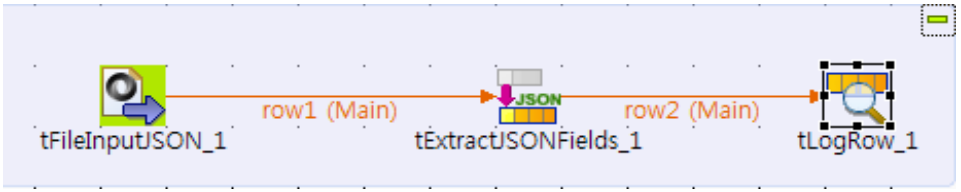
4. data_extracted 의 Basic settings에서 Mode 를 결과를 보기 좋게 볼 수 있도록 Table로 선택한다.

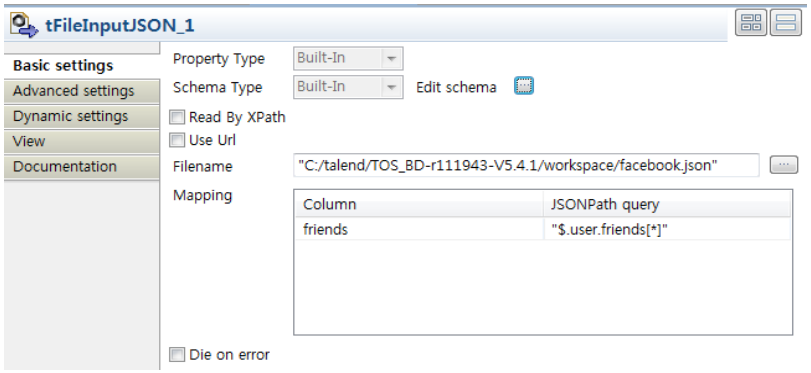
5. 실행을 하게 되면 그림과 같이 extraction에 실패한 reject된 부분의 원인과 상세 에러 메시지가 출력된다.

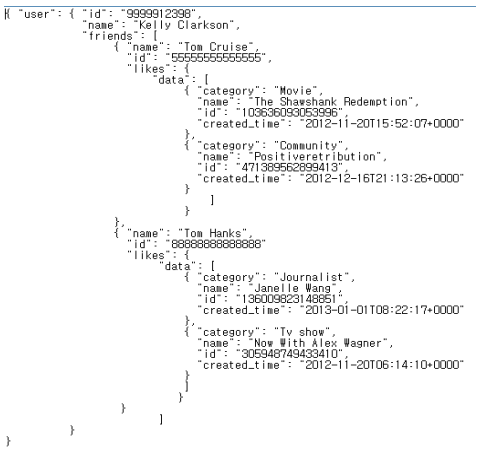


5. 컴포넌트 사용 실습

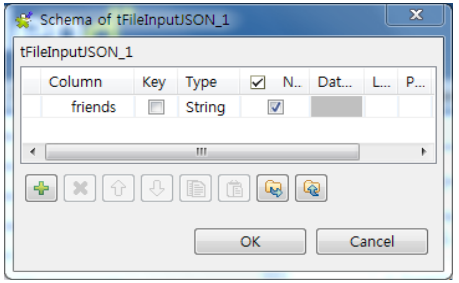
5.5 소셜 네트워크에서 데이터 추출하기







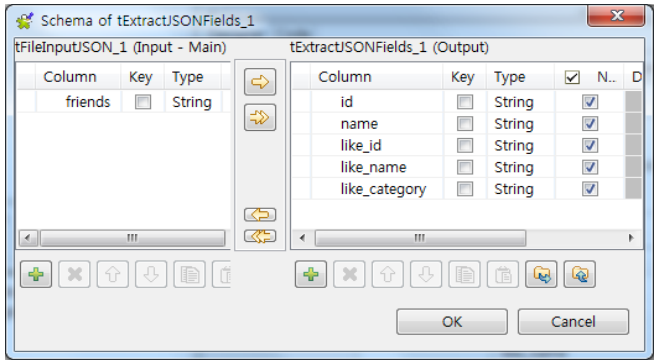
```
{ "user": { "id": "9999912398",
            "name": "Kelly Clarkson",
            "friends": [
                { "name": "Tom Cruise",
                  "id": "55555555555555",
                  "likes": {
                      "data": [
                          { "category": "Movie",
                            "name": "The Shawshank Redemption",
                            "id": "103636093053996",
                            "created_time": "2012-11-20T15:52:07+0000"
                          },
                          { "category": "Community",
                            "name": "Positiveretribution",
                            "id": "471389562899413",
                            "created_time": "2012-12-16T21:13:26+0000"
                          }
                      ]
                  }
                },
                { "name": "Tom Hanks",
                  "id": "88888888888888"
                  "likes": {
                      "data": [
                          { "category": "Journalist",
                            "name": "Janelle Wang",
                            "id": "136009823148851",
                            "created_time": "2013-01-01T08:22:17+0000"
                          },
                          { "category": "Tv show",
                            "name": "Now With Alex Wagner",
                            "id": "305948749433410",
                            "created_time": "2012-11-20T06:14:10+0000"
                          }
                      ]
                  }
                }
            ]
        }
}
```

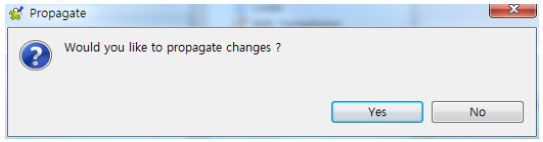


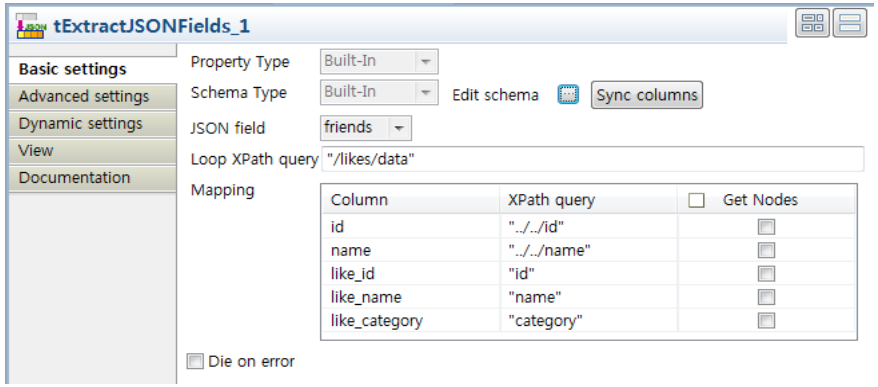
1. 팔레트의 File/Input폴더에서 tFileInputJSON 컴포넌트, Processing/Fields폴더에서 tExtractJSONFields 컴포넌트, Logs&Errors폴더에서 tLogRow를 각각 design workspace에 drop한다.
2. 그림과 같이 각 컴포넌트를 Main으로 연결한다.
3. tFileInputJSON_1 의 Basic settings tab 에서 입력할 파일경로를 선택한다. 선택한 JSON file *facebook.json* 의 내용을 그림과 같다
4. Edit schema를 클릭하여 컬럼에 friends를 입력한다.
5. Mapping table의 friends 컬럼 오른쪽 JSONPath query에 *"$.user.friends[*]"를 입력한다.*
6. Read by XPath check box는 선택하지 않는다.

5. 컴포넌트 사용 실습

5.5 소셜 네트워크에서 데이터 추출하기







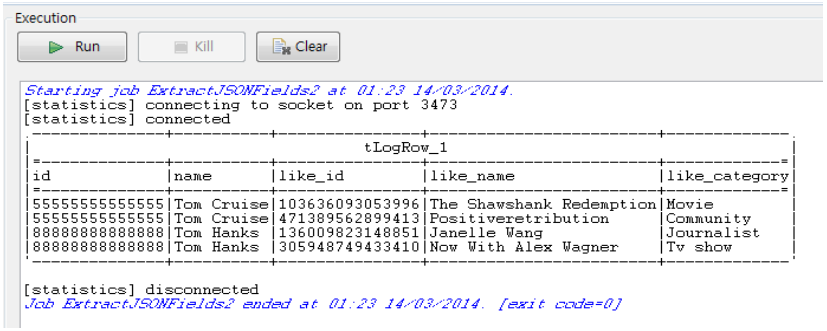
1. tExtractJSONFields 의 Basic settings에서 Edit schema창을 열어 JSON field *friends의 노드인 id, name, like_id, like_name, like_category* 을 입력한다.

2. Propagate 팝업창이 나오면 Yes를 클릭한다.

3. Loop XPath query field에 *"/likes/data"*를 입력한다.

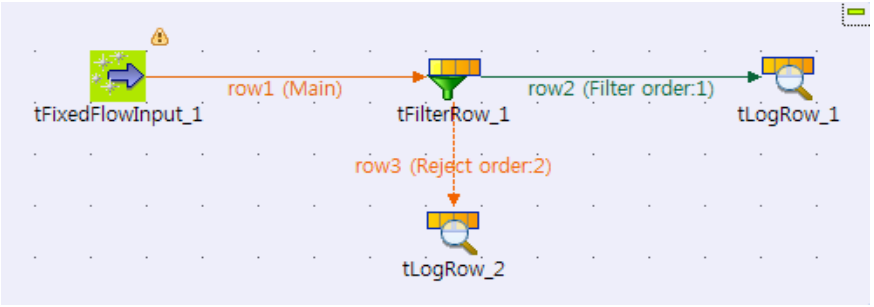
4. Mapping 부분의 XPath query column에 그림과 같이 입력한다. *"../../id"* 는 *"/friends/id"* 노드, *"../../name"* 는 *"/friends/name"* 노드를 가져온다.

5. tLogRow_1의 Basic settings에서 Mode 를 결과를 보기 좋게 볼 수 있도록 Table로 선택한 후 실행하면 그림과 같은 결과가 출력된다.



5. 컴포넌트 사용 실습

5.6 이름으로 필터링하기



Schema Type: Built-In — Edit schema

Number of rows: 1

Mode

- Use Single Table
- Use Inline Table
- Use Inline Content(delimited file)

Row Separator: "₩n" * Field Separator: ";" *

Content: *

```
romain;m;french;16
roman;m;russian,polish,czech;55
romano;n;italian41
romeo;m;italian;29
romolo;m;italian;0
romuluslmlroamn mythology;42
```

tFixedFlowInput_1

Column	Key	Type	N..	Date Patt...	Len...	P
firstname	☐	String	☑			
gender	☐	String	☑			
language	☐	String	☑			
frequency	☐	String	☑			

컴마로 구분되어 있는 파일을 읽어서 두 컬럼을 추출한다.

1. 팔레트의 Misc폴더에서 tFixedFlowInput컴포넌트, Processing폴더에서 tFilterRow 컴포넌트, Logs$Errors폴더에서 tLogRow 컴포넌트 2개를 가져와서 design workspace에 drop한다.

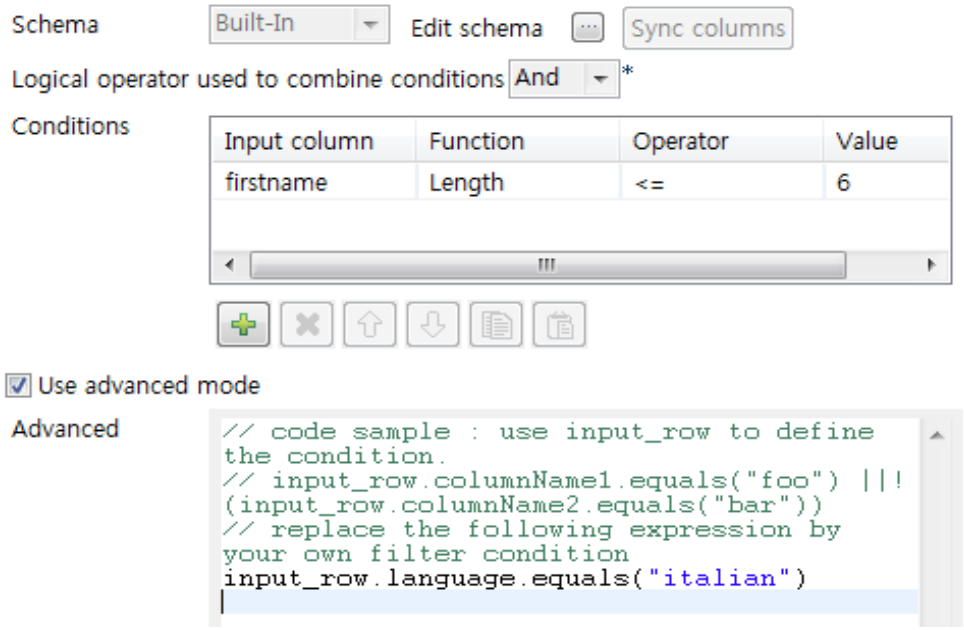
2. tFixedFlowInput_1에서 tFilterRow_1로는 Row > Main을 이용하여 연결하고, tFilterRow_1에서tLogRow_1로는 Row > Filter 를 이용하여 연결한다. tFilterRow_1에서 tLogRow_2로는 Row > Reject를 이용하여 연결한다. tLogRow_2는 필터에서 리젝트된 내용을 출력한다.

3. tFixedFlowInput_1의 Basic settings View에서 Edit schema를 클릭하여 파일구조와 같이 컬럼과 타입을 입력한다.

4. Mode 부분은 Use Inline Content(delimited file) 을 선택하고 샘플 데이터로 그림가 같이 입력한다.

5. 컴포넌트 사용 실습

5.6 이름으로 필터링하기




```
Starting job FilterRow at 01:40 17/03/2014.
[statistics] connecting to socket on port 3485
[statistics] connected
.---------+------+---------+---------.
|            tLogRow_1                |
|=---------+------+---------+---------=|
|firstname|gender|language|frequency|
|=---------+------+---------+---------=|
|romeo    |m     |italian |29       |
|romolo   |m     |italian |0        |
'---------+------+---------+---------'
.---------+------+-----------------------+---------+-----------------------------------------------------------.
|                                          tLogRow_2                                                          |
|=---------+------+-----------------------+---------+-----------------------------------------------------------=|
|firstname|gender|language               |frequency|errorMessage                                               |
|=---------+------+-----------------------+---------+-----------------------------------------------------------=|
|romain   |m     |french                 |16       |advanced condition failed                                  |
|roman    |m     |russian,polish,czech   |55       |advanced condition failed                                  |
|romano   |n     |italian41              |null     |advanced condition failed                                  |
|romulus  |m     |roamn mythology        |42       | firstname.length() <= 6 failed|advanced condition failed|
'---------+------+-----------------------+---------+-----------------------------------------------------------'

[statistics] disconnected
Job FilterRow ended at 01:40 17/03/2014. [exit code=0]
```

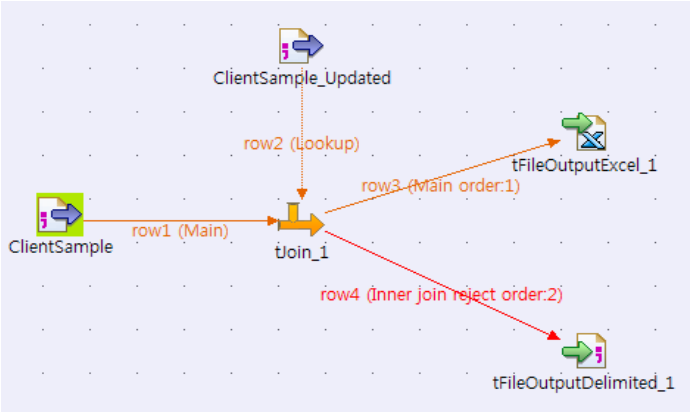
1. tFilterRow_1의 Basic settings View에서 Conditiions 테이블의 + 버튼을 눌러 1라인을 추가한 후 InputColumn에서 firstname, Function에서는 Length, Operator에서는 <= 을 선택하고, Value 는 6을 입력한다.

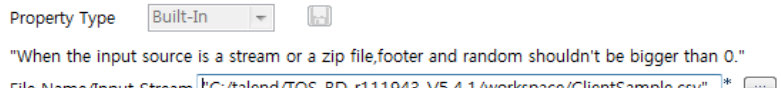
2. Language가 italian을 찾기 위해서 Use advanced mode 를 체크하고, Advanced 부분에 input_row.language.equals("italian")를 입력한다.

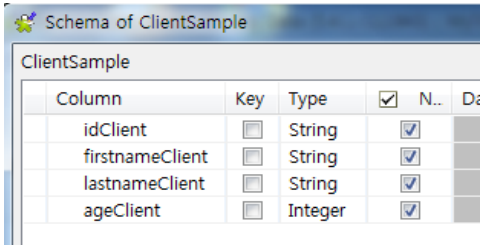
3. 실행을 하면 그림과 같이 첫번째 테이블에는 firstname이 6자 이하이고, language가 italian 인 결과가 나타나고, 두번째 테이블에는 리젝트된 내용과 이유가 표시된다.

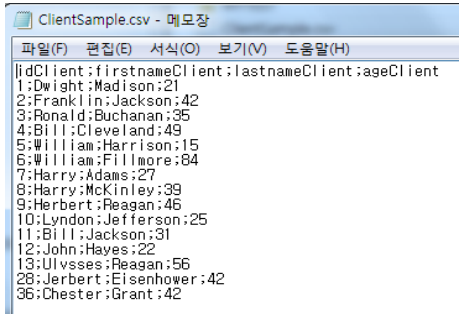
5. 컴포넌트 사용 실습

5.7 두개의 파일 조인하여 엑셀과 리젝트 파일로 출력하기









컴마로 구분되어 있는 파일을 읽어서 두 컬럼을 추출한다.

1. 팔레트의 File/Input폴더에서 tFileInputDelimited 컴포넌트 2개, Processing폴더에서 tJoin컴포넌트, File/Output 폴더에서 tFileOutputExcel, tFileOutputDelimited 컴포넌트를 가져와서 design workspace에 drop한다.

2. tFileInputDelimited_1은 ClientSample, tFileInputDelimited_2는 ClientSample_Updated 로 이름을 변경한다

3. 여기서 입력으로 사용하는 ClientSample File은 네개의 컬럼으로 되어 있고, *firstnameClient* 과*lastnameClient* 이 포함되어 있고 이 두 컬럼들이 정확하게 매치되도록 할 것이다

4. ClientSample에서 tJoin_1로 Main link를 이용하여 연결한다. ClientSample_Update에서 tJoin_1로 Main link를 이용하여 연결하면 Lookup link로 연결된다

5. tJoin_1에서 tFileOutputExcel_1로 Main link, tFileOutputDelimited_1로 Inner join reject link로 연결한다.

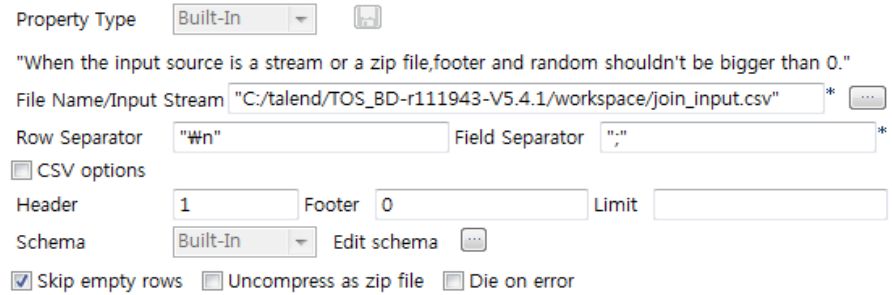
6. ClientSample의 의 Basic settings View에서 File Name에서 InputFile을 선택하고, 헤더라인이 1이므로 Header는 1을 입력한다.

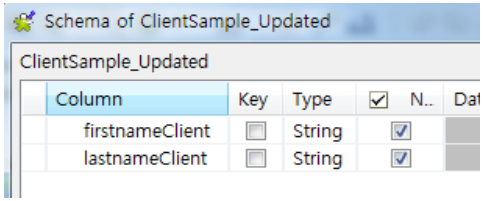
7. Edit schema를 클릭하여 세번째 그림과 같이 입력필드를 정의한다.

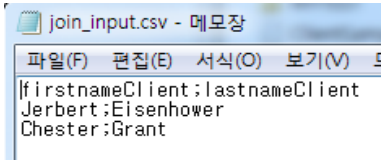
8. ClientSample.csv파일의 내용은 네번째 그림과 같다

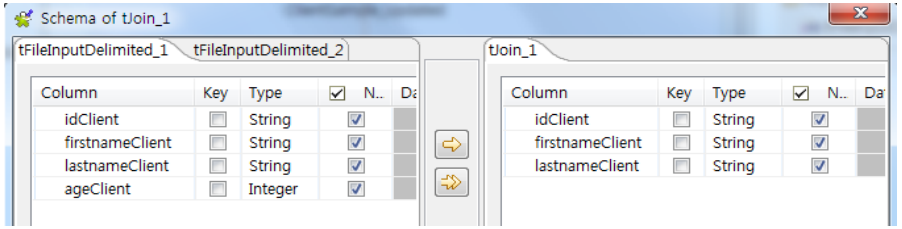
5. 컴포넌트 사용 실습

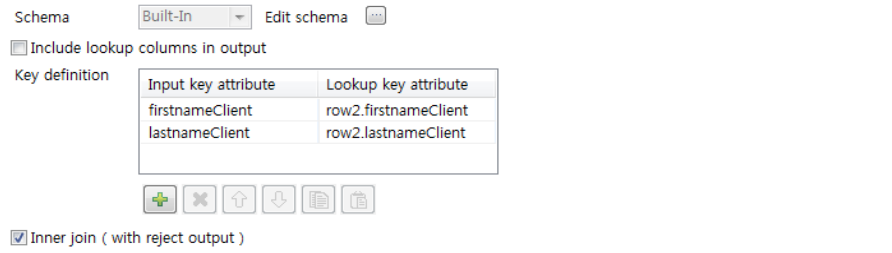
5.7 두개의 파일 조인하여 엑셀과 리젝트 파일로 출력하기







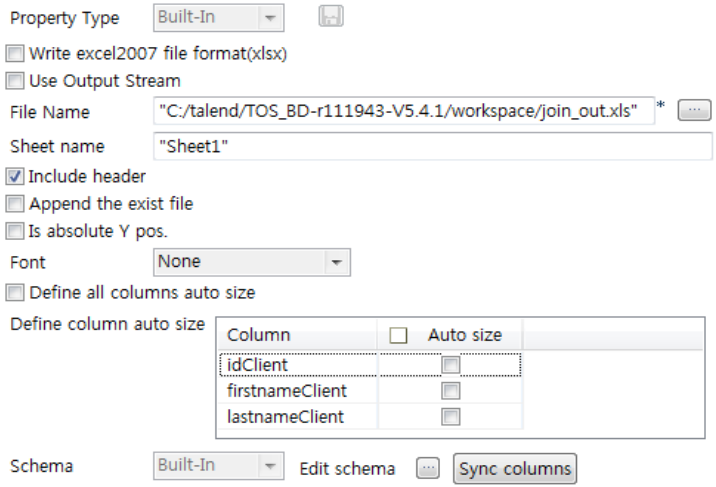


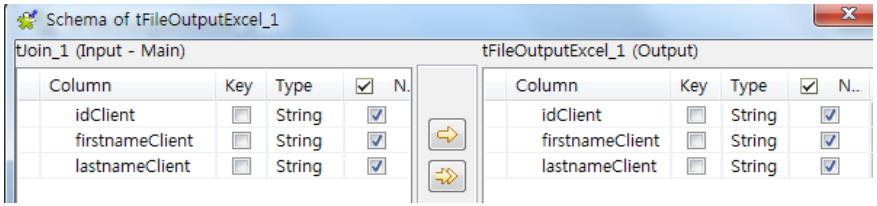


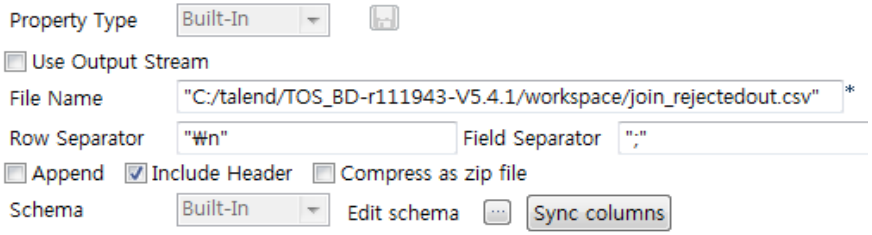
1. ClientSample_Update의 Basic settings View에서 File Name에서 Join_input.csv를 선택하고, 헤더라인이 1이므로 Header는 1을 입력한다.
2. Edit schema를 클릭하여 두번째 그림과 같이 입력필드를 정의한다.
3. Join_input.csv 파일의 내용은 세번째 그림과 같다
4. tJoin_1의 Basic settings View에서 Edit schema를 클릭하여 네번째 그림과 같이 입력한다.
5. Basic setting View의 Key definition에서 Input key attribute와 Lookup key attribute를 그림과 같이 지정한다. Input key attribute와 Lookup key attribute가 일치하는 내용은 Excel 파일로 출력이 되고, 그 이외의 데이터는 csv output 파일로 생성할 것이다.
6. 일치하지 않는 내용을 파일로 생성하기 위하여 Inner join (with reject output)를 체크한다.

5. 컴포넌트 사용 실습

5.7 두개의 파일 조인하여 엑셀과 리젝트 파일로 출력하기





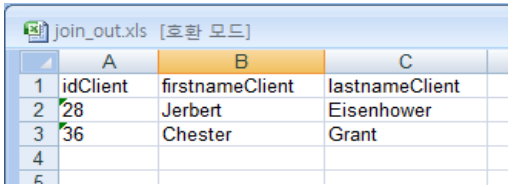


1. tFileOutputExcel 의 Basic settings View에서 File Name에서 엑셀 파일로 출력할 Join_out.xls를 입력하고, 헤더라인을 같이 출력하기 위하여 Include header 를 체크한다.

2. Edit schema를 클릭하여 두번째 그림과 같이 입력필드를 정의한다.

3. tFileOutputDelimited 의 Basic settings View에서 File Name에서 리젝트된 내용을 CSV파일로 출력할 Join_rejectedout.csv를 입력하고, 헤더라인을 같이 출력하기 위하여 Include header 를 체크한다.

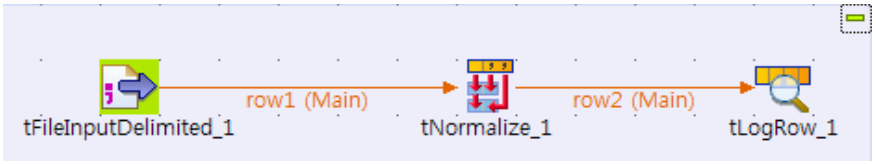
4. 실행 후 나온 결과 출력은 아래 그림과 같다.

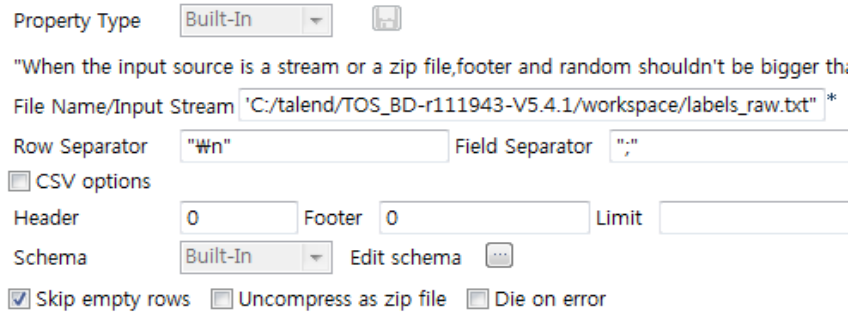


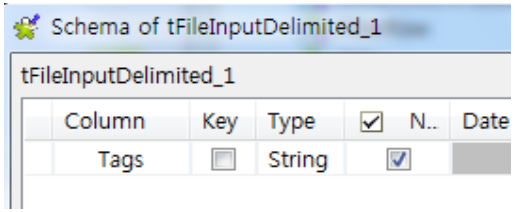
idClient;firstnameClient;lastnameClient
1;Dwight;Madison
2;Franklin;Jackson
3;Ronald;Buchanan
4;Bill;Cleveland
5;William;Harrison
6;William;Fillmore
7;Harry;Adams
8;Harry;McKinley
9;Herbert;Reagan
10;Lyndon;Jefferson
11;Bill;Jackson
12;John;Hayes
13;Ulysses;Reagan

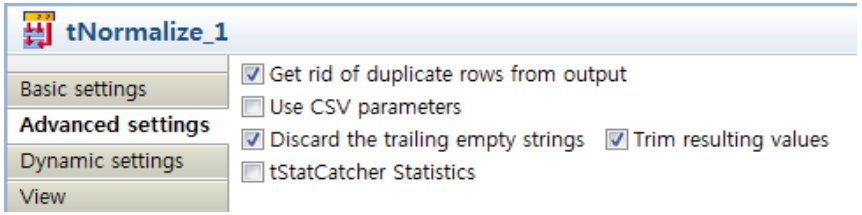
5. 컴포넌트 사용 실습

5.8 데이터 표준화하기









1. 팔레트의 File/Input폴더에서 tFileInputDelimited 컴포넌트, Processing/Field폴더에서 tNormalize컴포넌트, Logs&Errors 폴더에서 tLogRow컴포넌트를 가져와서 design workspace에 drop한다.

2. tFileInputDelimited 에서 tNormalize 로 Main link를 이용하여 연결하고, 다시 tNormalize 에서 tLogRow 로 Main link를 이용하여 연결한다

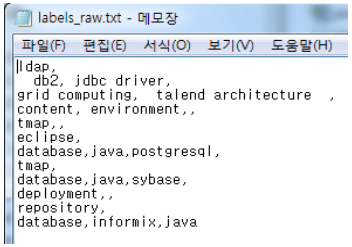
3. tFileInputDelimited의 Basic settings View의 File Name에서 입력파일로 사용할 labels_raw.txt을 선택한다.

4. Edit schema 창을 열어서 Column에 Tags를 추가한다.

5. tNormalize의 Basic settings 에서 입력 컴포넌트와 스키마를 맞추기 위해서 Sync columns 을 클릭한다.

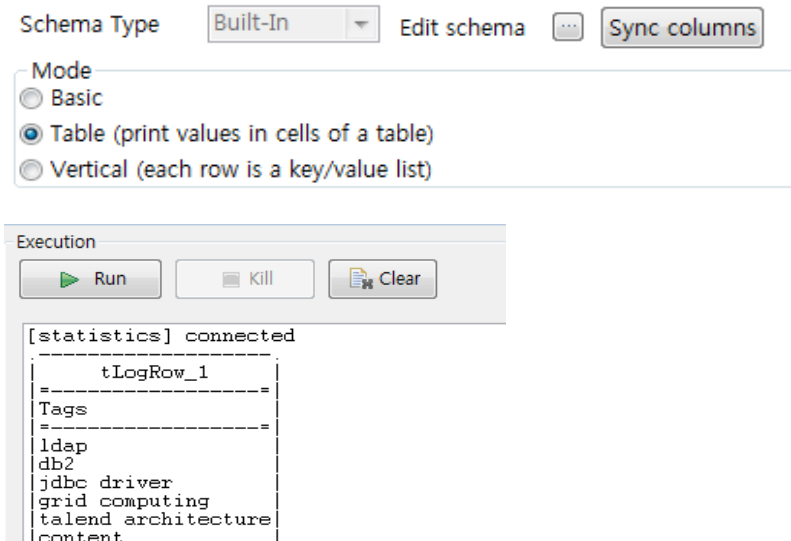
6. Advanced settings view에서 Get rid of duplicate rows from output, Discard the trailing empty strings, Trim resulting values 를 체크한다.

7. 입력파일 내용은 아래와 같다



5. 컴포넌트 사용 실습

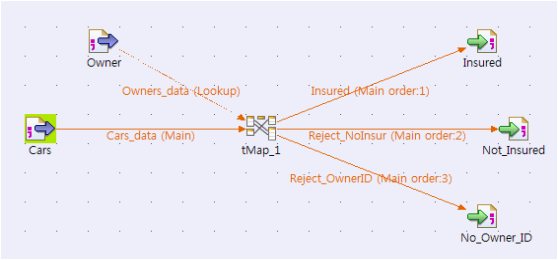
5.8 데이터 표준화하기

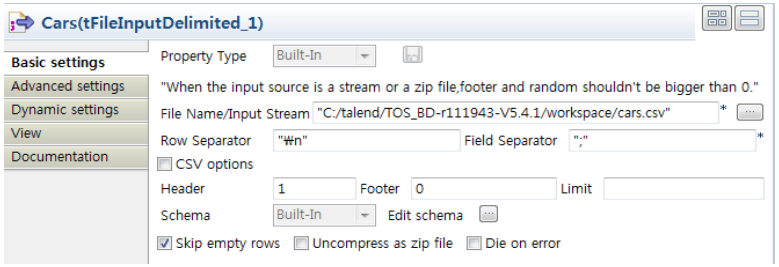


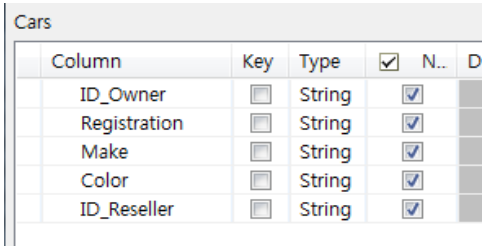
1. tLogRow 에서 Print values in the cells of table 을 선택한다.
2. 실행을 하면 그림과 같이 입력데이타가 표준화 되어서 출력이 된다.

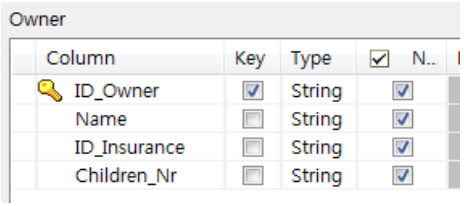
5. 컴포넌트 사용 실습

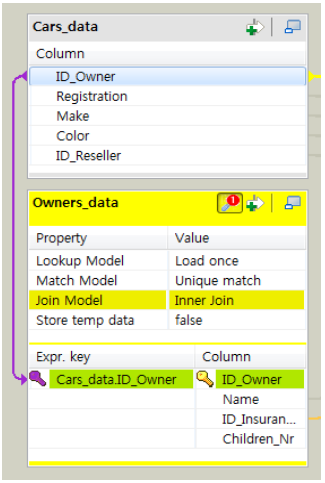
5.9 데이터 매핑과 Reject하기











1. 팔레트의 File/Input폴더에서 tFileInputDelimited 컴포넌트 2개, Processing폴더에서 tMap 컴포넌트, File/Output 폴더에서 tFileOutputDelimited 컴포넌트 3개를 가져와서 design workspace에 drop한다.

2. tFileInputDelimited_1은 Cars, tFileInputDelimited_2는 Owner로 이름을 변경한다.

3. Cars에서 tMap 로 Main link를 이용하여 연결하고 이름을 Cars_data, Owner에서 tMap으로 Lookup link를 이용하여 연결하고 이름은 Owners_data로 변경한다.

4. Cars 컴포넌트의 Basic settings View의 File Name에서 입력파일로 사용할 cars.cvs를 선택하고, Owner도 같은 방식으로 File Name에 owner.csv를 선택한다.

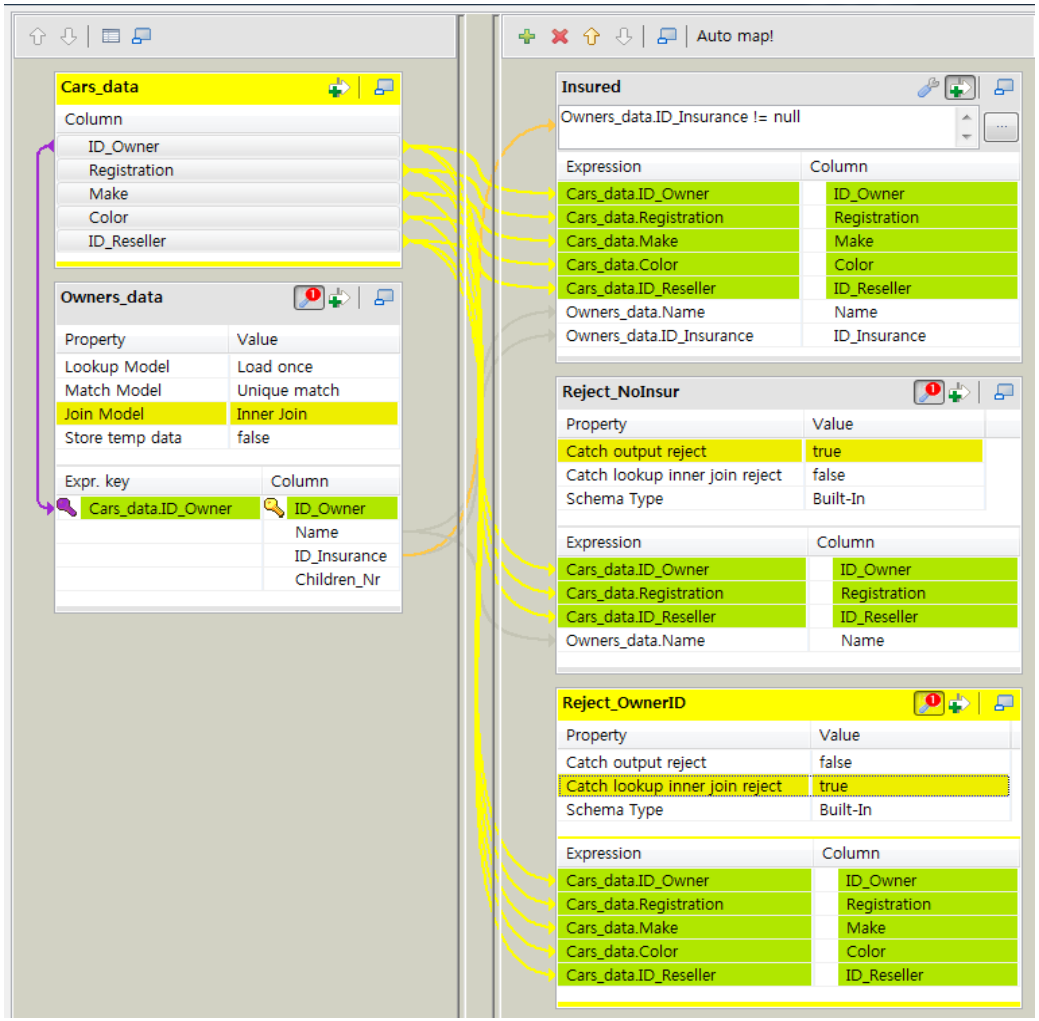
5. Cars와 Owner의 Edit schema 창을 열어서 그림과 같이 컬럼을 추가한다.

6. tMap컴포넌트의 맵 에디터 창을 열어서 Cars_data 테이블의 ID_Owner 컬럼을 Owner_data 테이블에 drop하여 두 테이블간의 조인을 생성한다.

7. Owner_data 테이블의 tMap settings 버튼을 클릭한 후 Join Model은 Inner Join을 선택한다.

5. 컴포넌트 사용 실습

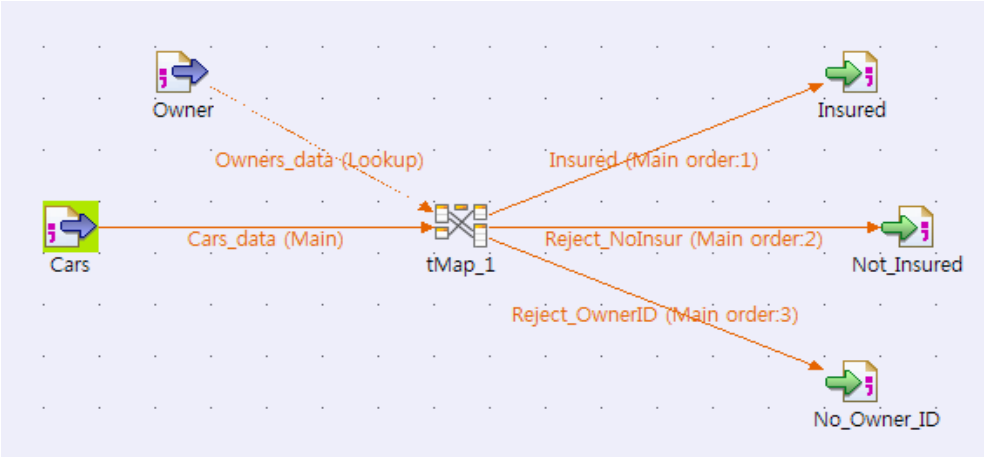
5.9 데이터 매핑과 Reject하기

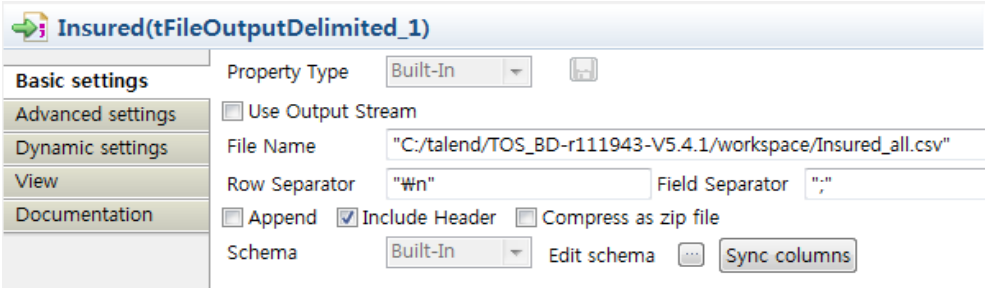


1. Map Editor의 output 영역에 + 키를 눌러서 Insured, Reject_NoInsur, Reject_OwnerID 라는 3개의 output 테이블을 생성한다.
2. Cars_data의 모든 컬럼을 Insured 테이블에 drag&drop한다.
3. Insured 테이블의 우측 상단의 +키를 클릭하여 필터 row를 추가한 후 Owners_data 테이블의 *ID_Insurance* 컬럼을 필터 조건에 drop하고 " != null "을 입력한다. insurance ID가 있는 모든 레코드는 이 테이블에 모여지게 된다.
4. Cars_data테이블의 ID_Owner, Registration, ID_Reseller 컬럼과 Owners_data 테이블의 Name, ID_Insurance 컬럼을 Reject_NoInsur 테이블에 drag&drop한다.
5. Cars_data의 모든 컬럼을 Reject_OwnerID 테이블에 drag&drop한다.
6. Reject_NoInsur 테이블의 우측 상단의 +키를 클릭하여 Catch output reject의 value를 true로 변경한다. insurance ID가 없는 레코드는 이 테이블에 모여지게 된다
7. Reject_OwnerID 테이블의 우측 상단의 +키를 클릭하여 Catch lookup inner join reject의 value를 true로 변경한다. Cars_data에서 owner ID가 매치되지 않거나 빠지는 레코드들이 모여지게 된다.

5. 컴포넌트 사용 실습

5.9 데이터 매핑과 Reject하기





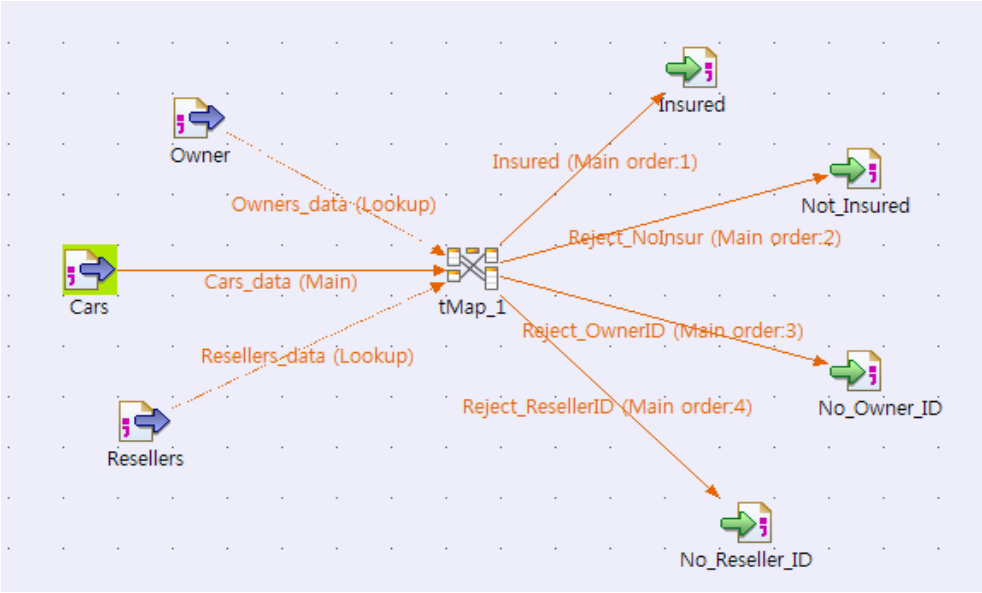
1. 앞에서 가져온 3개의 tFileOutputDelimited 컴포넌트 이름을 Insured, Not_Insured, No_Owner_ID로 변경한다.

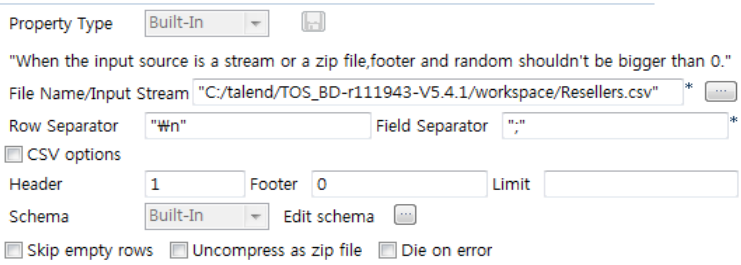
2. tMap_1에서 오른쪽 클릭하여 Row/Insured는 Insured output 컴포넌트를 클릭하면 Insured link가 연결된다. 같은 방식으로 Reject_NoInsur, Reject_OwnerID를 연결한다.

3. Insured 컴포넌트의 Basic settings View의 File Name에서 출력파일로 사용할 insured_all.cvs를 입력하고 Include Header를 체크한다. 같은 방식으로 Not_Insured, No_Owner_ID를 설정한다.

5. 컴포넌트 사용 실습

5.10 Inner join rejection

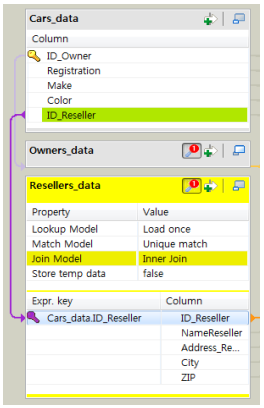


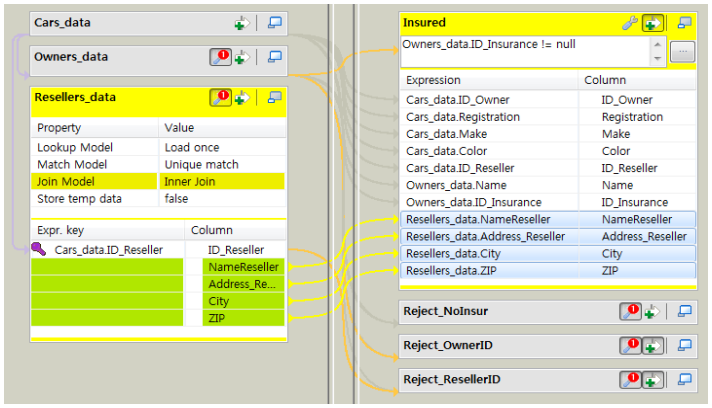


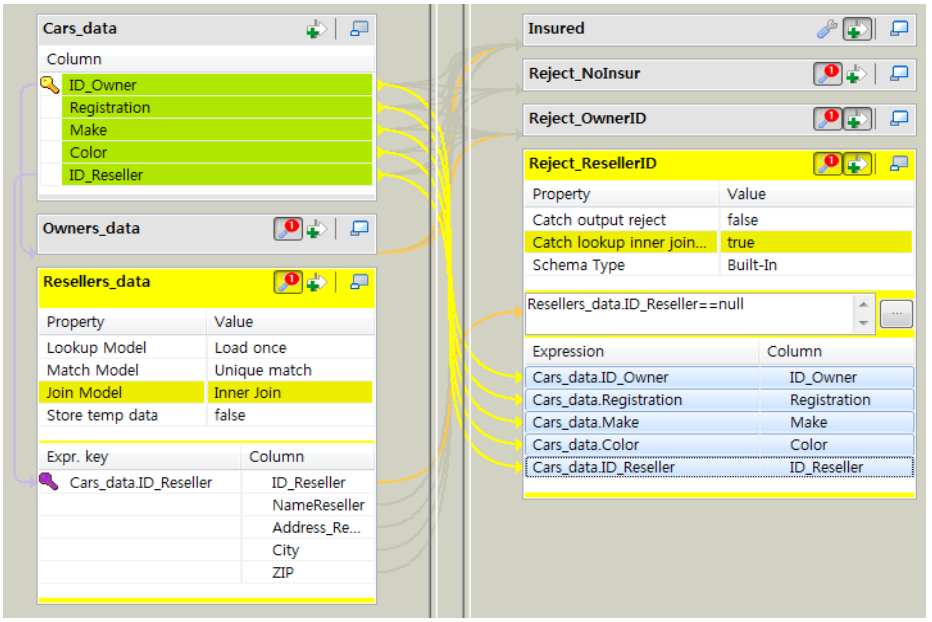
1. 앞의 시나리오에 이어서 작업한다.
2. 팔레트의 File/Input폴더에서 tFileInputDelimited 컴포넌트, File/Output 폴더에서 tFileOutputDelimited 컴포넌트를 가져와서 design workspace에 drop한다.
3. tFileInputDelimited 이름은 Resellers, tFileOutputDelimited 이름은 No_Reseller_ID로 변경한다.
4. Resellers에서 tMap 로 Main link를 이용하여 연결하고 이름을 Resellers_data로 변경하고, tMap_1에서 오른쪽 클릭하여 Row/No_Reseller_ID를 클릭한 후 No_Reseller_ID 컴포넌트를 클릭하면 output 이름을 입력하는 창이 표시되고 Reject_ResellerID라고 입력한 후 OK를 누르면 Reject_ResellerID link가 연결된다.
5. Resellers 컴포넌트의 Basic settings View의 File Name에서 출력파일로 사용할 resellers.cvs를 입력하고 Include Header를 체크하고 Header는 1을 입력한다.

5. 컴포넌트 사용 실습

5.10 Inner join rejection



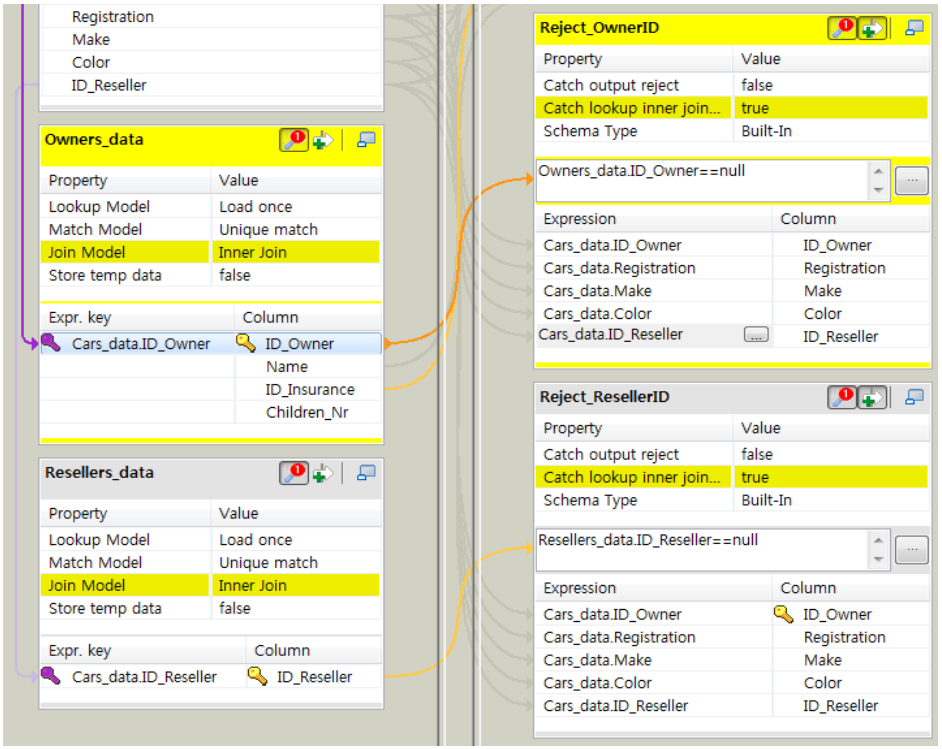


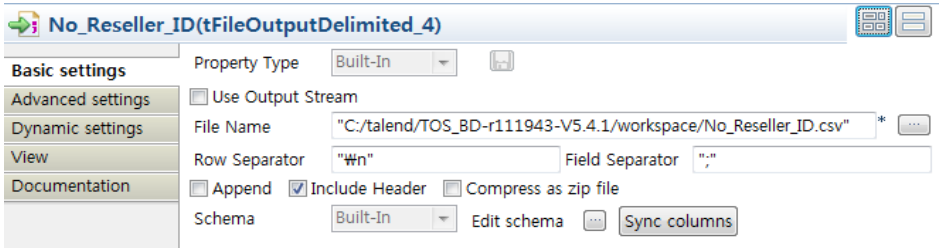


1. Cars_data 와 Resellers_data 를 조인하기 위하여 Cars_data 테이블의 ID_Reseller 컬럼을 Resellers_data 테이블에 드롭하면 Resellers_data 테이블에 ID_Reseller 컬럼이 추가된다
2. 화면 하단의 Schema Editor를 이용하여 Name_Reseller, Address_Reseller, City, ZIP 컬럼을 추가한다.
3. Resellers_data 테이블의 우측 상단의 +키를 클릭하여 Join Model 속성을 Inner Join으로 변경한다
4. Resellers_data 테이블의 모든 컬럼을 메인 output 테이블인 Insured 에 드롭한다.
5. Cars_data의 모든 컬럼을 선택하여 Reject_ResellerID테이블에 드롭하면 컬럼이 추가된다.
6. Reject_ResellerID 테이블의 우측 상단의 +키를 클릭하여 Catch lookup inner join reject 속성을 true로 변경한다

5. 컴포넌트 사용 실습

5.10 Inner join rejection





1. 첫번째 Inner Join output 테이블인 Reject_OwnerID테이블의 +화살표 버튼을 눌러서 필터를 추가할 수 있는 라인을 추가하고, reject할 조건 Owners_data.ID_Owner==null 을 입력한다.

2. 두번째 Inner Join output 테이블인 Reject_ResellerID 의 필터 조건은 Resellers_data.ID_Reseller==null 을 입력한다.

3. No_Reseller_ID컴포넌트의 Basic settings View의 File Name에서 출력파일로 사용할 No_Reseller_ID.cvs를 입력하고 Include Header를 체크한다.

4. 잡을 저장한 후 실행을 하면 출력 파일 지정한 폴더에 4개의 출력파일이 생성된다.