

Cleo Integration Cloud™ | Clarify

Chapter | **Transforming multi-format flat file to database**

Overview | This topic demonstrates how to transform data in a flat file to database format.

The source is a delimited multi-format flat file.

Features, functions, and tips not discussed in previous demos include: how a Project can reference another, thus making its resources available for re-use. Also, if the Project that is referenced does not exist in the Workspace, the process of checking it out from an SVN Repository is introduced.

Contents	Scenario	2
	Resources	4
	Project: com.training.demo04.fftodb	5
	File Adapter: ReadTxtFA	6
	Schema: DelimitedFF	7
	Data Source: Demo04DS	8
	Database Schema: Demo04DB	11
	Database Adapter: Demo04DBA	12
	Transformation Settings: PipeDelimitedSourceTS	14
	Ruleset: DelimitedFfToDbRS	15
	Business Process: DelimitedFfToDbBPS	22
	Deploy, Launch, Audit	25

Scenario

The source data arrives in a multi-format pipe-delimited flat file, illustrated here.

```
PO|BestValue_22|1/1/2013
CI|HI-BVGS|BEST VALUE GOODSTUFF|1 BEST VALUE BLVD||HONOLULU|HAWAII|96815
ST|Same as Company
BT|Same as Company
CN|BOB SMITH|(555) 555-5555|BSMITH@BVGS.COM
SI|
LI|1|PINGBALL|WHITE PING PONG BALLS (100)|TABLE GAME ITEMS|500682|200|Case|16|3200
LI|2|BARSOAP|DECORATIVE SOAP BAR|KEEP IT CLEAN|500227|900|Each|0.75|675
TO|2|1100|3875
PO|GoodStuff 303|1/2/2013
CI|HI-BVGS|BEST VALUE GOOD STUFF|1 BEST VALUE BOULEVARD||HONOLULU|HAWAII|96815
ST|New|ABC DISTRIBUTION|1234 MAIN STREET|PO BOX 938|CLEVELAND|OHIO|44109-1234
BT|Same as Shipping
CN|SUSAN JOHNSON|(555) 555-6666|SJOHNSON@BVGS.COM
SI|Please note the new SHIPTO address
LI|1|LMWG01|LARGE MANS WORK GLOVES BLACK|GLOVELY STUFF|500151|160|Pair|9.25|1480
LI|2|LMWC01|LARGE MANS WINTER COAT BLACK|MANITZ COLDOUT|500155|6|Each|100|600
LI|3|SWG-RED|SMALL WOMENS GLOVES - RED|GLOVELY STUFF|500001|24|Pair|12.5|300
TO|3|190|2380
PO|FV-2013_0047|1/3/2013
CI|WY-FV|FAMILIAR VASES|100 INDUSTRIAL WAY|SUITE 123|CASPER|WYOMING|82640
ST|Same as Company
BT|New|BOB SMITH|23 SOUTH CENTER ST||POTTSVILLE|PENNSYLVANIA|17901
CN|BOB SMITH|(555) 555-5555|BSMITH@BVGS.COM
SI|New Billing Address|Please handle with care|
LI|1|PV-28 EQ|28-INCH PORCELAIN VASE|FANCY VASES||1|Each|375|375
TO|1|1|375
```



The target database's four tables should be populated as below.

HEADER

PO_Number	Date	ContactName	Phone	Email	Lines	Quantity	Cost	Processed	DateProcessed
BestValue_22	1/1/2013	BOB SMITH	(555) 555-5555	BSMITH@BVGS.COM	2	1100	3875 U	(null)	(null)
GoodStuff 303	1/2/2013	SUSAN JOHNSON	(555) 555-6666	SJOHNSON@BVGS.COM	3	190	2380 U	(null)	(null)
FV-2013_0047	1/3/2013	BOB SMITH	(555) 555-5555	BSMITH@BVGS.COM	1	1	375 U	(null)	(null)

NOTES

PO_Number	Note1	Note2	Note3
BestValue_22	(null)	(null)	(null)
GoodStuff 303	New Billing Address	Please handle with care	(null)
FV-2013_0047	New Billing Address	Please handle with care	(null)

ADDRESS

PO_Number	Type	Customer_Indicator	Name	Address1	Address2	City	State	ZipCode
BestValue_22	CI	HI-BVGS	BEST VALUE GOOD STUFF	1 BEST VALUE BOULEVARD	(null)	HONOLULU	HAWAII	96815
BestValue_22	ST	Same as Company	(null)	(null)	(null)	(null)	(null)	(null)
BestValue_22	BT	Same as Company	(null)	(null)	(null)	(null)	(null)	(null)
GoodStuff 303	CI	HI-BVGS	BEST VALUE GOOD STUFF	1 BEST VALUE BOULEVARD	(null)	HONOLULU	HAWAII	96815
GoodStuff 303	ST	New	ABC DISTRIBUTION COMPANY	1234 MAIN STREET	PO BOX 938	CLEVELAND	OHIO	44109-1234
GoodStuff 303	BT	Same as Shipping	(null)	(null)	(null)	(null)	(null)	(null)
FV-2013_0047	CI	WY-FV	FAMILIAR VASES	100 INDUSTRIAL WAY	SUITE 123	CASPER	WYOMING	82640
FV-2013_0047	ST	Same as Company	(null)	(null)	(null)	(null)	(null)	(null)
FV-2013_0047	BT	New	BOB SMITH	23 SOUTH CENTER ST	(null)	POTTSVILLE	PENNSYLVANIA	17901

DETAIL

PO_Number	LineNumber	YourNumber	Description	Brand	OurNumber	Qty	Units	UnitPrice	ExtendedPrice
BestValue_22	1	PINGBALL	WHITE PING PONG BALLS (100)	TABLE GAME ITEMS	500682	200	Case	16	3200
BestValue_22	2	BARSOAP	DECORATIVE SOAP BAR	KEEP IT CLEAN	500227	900	Each	0	675
GoodStuff 303	1	LMWG01	LARGE MANS WORK GLOVES BLACK	GLOVELY STUFF	500151	160	Pair	9	1480
GoodStuff 303	2	LMWC01	LARGE MANS WINTER COAT BLACK	MANITZ COLDOUT	500155	6	Each	100	600
GoodStuff 303	3	SWG-RED	SMALL WOMENS GLOVES - RED	GLOVELY STUFF	500001	24	Pair	12	300
FV-2013_0047	1	PV-28 EQ	28-INCH PORCELAIN VASE	FANCY VASES	(null)	1	Each	375	375

Resources

The resources needed to accomplish this transformation are meant to provide a basic understanding of how Clarify objects work together. All resources are created and defined on the *Clarify Workbench / Project Explorer*.

- Project [**com.training.demo04.fftodb**] – A Project is a directory within the Workspace.
- Package [**com.training.demo04.fftodb**] – A Package is a directory within a Project.
- File Adapter [**ReadTxtFA**] – This File Adapter brings the source data to Clarify.
- Flat File Schema [**DelimitedFF**] – This Schema describes the delimited flat file data. As this Schema already exists in the *com.training.demo02.multifftoff* Project, that Project will be instructed to share its resources, and then this Project will be instructed to depend on that Project.



If you did not create that Project, check it out from Cleo's SVN Repository.

- Data Source [**Demo04DS**] – This Data Source defines the location and login credentials for the database.
 - Database Schema [**Demo04DB**] – This Schema describes the database data.
 - Database Adapter [**Demo04DBA**] – This Schema allows for inserting data into the tables.
 - Transformation Settings [**PipeDelimitedSourceTS**] – This Transformation Settings object informs the Business Process of the delimiter to use when reading the source file. No delimiter is necessary for writing the target file, as it has a database format.
 - Ruleset [**DelimitedFfToDbRS**] – This Ruleset transforms the data from the source format to the target format.
 - Business Process [**DelimitedFfToDbBPS**] – This Business Process sequences three user-defined objects to read the source data file, transform its data to the target data format, and write the output to another file.
-
-
-
-

Project: com.training.demo04.fftodb

Select *File* | *New* | *Clarify Project*.

Type **com.training.demo04.fftodb** in the *Project name* field.

Click **Finish**.

When Clarify creates a Project, it automatically creates a Package named the same as the Project. All resources for this Project are created in its *com.training.demo04.fftodb* Package.



This Project will use a resource shared by another Project. These instructions describe how this “guest” Project depends on resources explicitly shared by a “host” Project.

If you did not create that Project, check it out from Cleo’s SVN Repository.

Host Project

Expand the *com.training.demo02.multifftoff* Project’s *META-INF* folder.

Double-click its *MANIFEST.MF* file to display its editor, click the *Runtime* tab at the bottom, and click **Add** in the *Exported Packages* section.

Select the *Show non-Java packages* checkbox, select the *com.training.demo02.multifftoff* Package, and click **OK**.

Save the object and close the editor.

Guest Project

Expand the *com.training.demo04.fftodb* Project’s *META-INF* folder.

Double-click its *MANIFEST.MF* file to display its editor, click the *Dependencies* tab at the bottom, and click **Add** in the *Required Plug-ins* section.

Type **demo** in the *Select a Plug-in* field, select the *com.training.demo02.multifftoff* Project in the *Matching items* list, and click **OK**.

Save the object and close the editor.

File Adapter: ReadTxtFA

Select the Package.

Select *File* | *New* | *File Adapter*.

Choose *Read* and click **Next**.

Type **ReadTxtFA** in the *Name* field.

Click **Finish**. The object is created and its editor appears.

Click in the *File Name* field. Click the **Open** button, locate/select the *source_data.txt* file in the *C:\Clarify_demos\demo04\in* folder, and click **Open**.

Save the object and close the editor.

Schema: DelimitedFF



Resources created for one Project may be referenced by another Project.

For this demo, the *com.training.demo02.multifftoff* Project's *DelimitedFF* flat file Schema will be re-used in this Project.

Data Source: Demo04DS

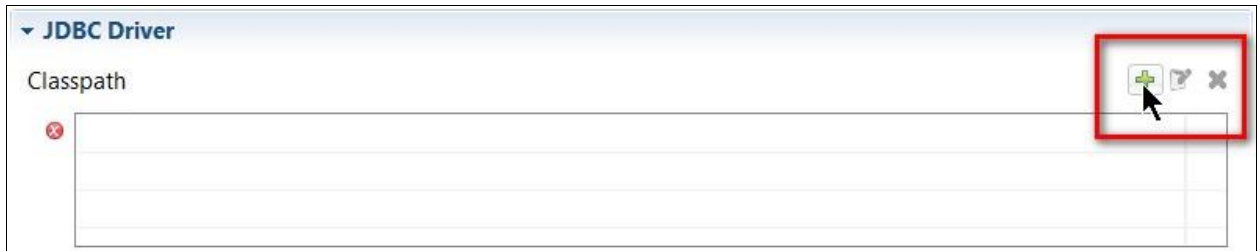
Select the Package.

Select *File | New | Data Source*.

Type **Demo04DS** in the *Name* field.

Click **Finish**. The object is created, and its Editor appears.

Click the *JDBC Driver* section's **Add** button.



Click the **Open** button, locate/select the *hsqldb-2.3.3.jar* file in the *C:\Clarify_demos\DEMO_DB* folder, and click **Open**. Click **OK**.

Click in the *Driver Name* field and double-click the *org.hsqldb.jdbc.JDBCdriver* entry.

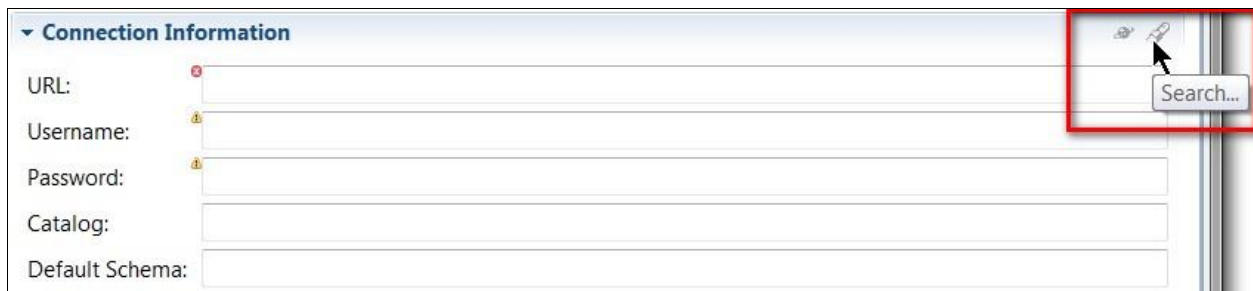
Provide values for the fields in the *Connection Information* section as below.

Field	Value
URL	jdbc:hsqldb:file:C:\Clarify_demos\DEMO_DB\database\dbschematest; hsqldb.lock_file=false;shutdown=true
Username	atl
Password	atl
Catalog	PUBLIC
Default Schema	PUBLIC

You may wish to copy the line below from this PDF document and paste it into the URL field.

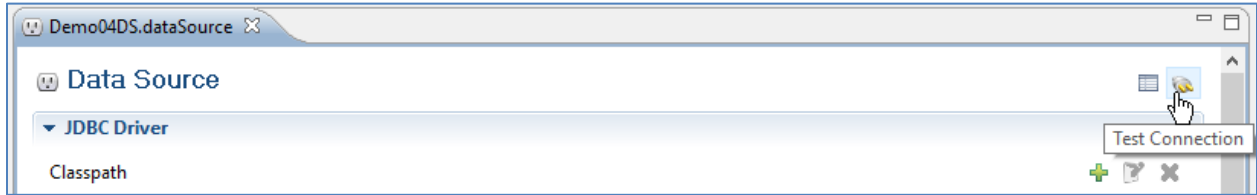
`jdbc:hsqldb:file:C:\Clarify_demos\DEMO_DB\database\dbschematest;hsqldb.lock_file=false;shutdown=true`

Note that the **Search** button may be used to produce a list of Schemas to populate the *Default Schema* field.

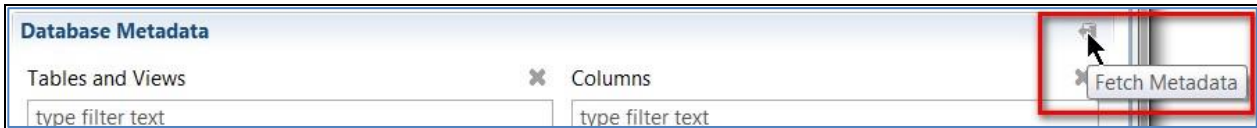


The screenshot shows a web form titled "Connection Information" with the following fields: URL, Username, Password, Catalog, and Default Schema. A red box highlights a "Search..." button located to the right of the Username and Password fields. The button has a magnifying glass icon and a mouse cursor pointing to it.

Click the **Test Connection** button to ensure Clarify can connect to the database. Click **OK**.



Click the *Database Metadata* section's **Fetch Metadata** button to display the tables of the database's default Schema.



Save the object and close the editor.

Database Schema: Demo04DB

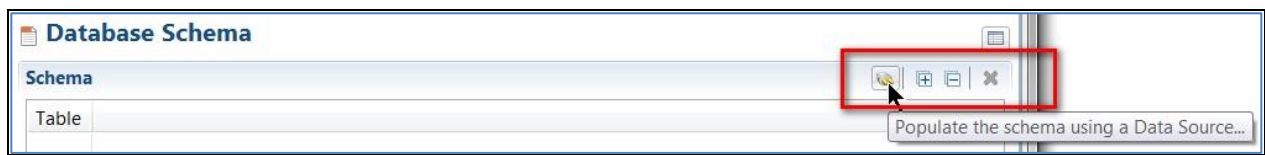
Select the Package.

Select *File | New | Database Schema*.

Type **Demo04DB** in the *Name* field.

Click **Finish**. The object is created, and its Editor appears.

Click the **Populate** button.



Select the *Demo04DS* object and click **OK**.

Drag the *NOTES* table and drop it on the *HEADER* table to create a parent-child relationship.

Repeat for the *ADDRESS* and *DETAIL* tables.

Save the object and close the editor.

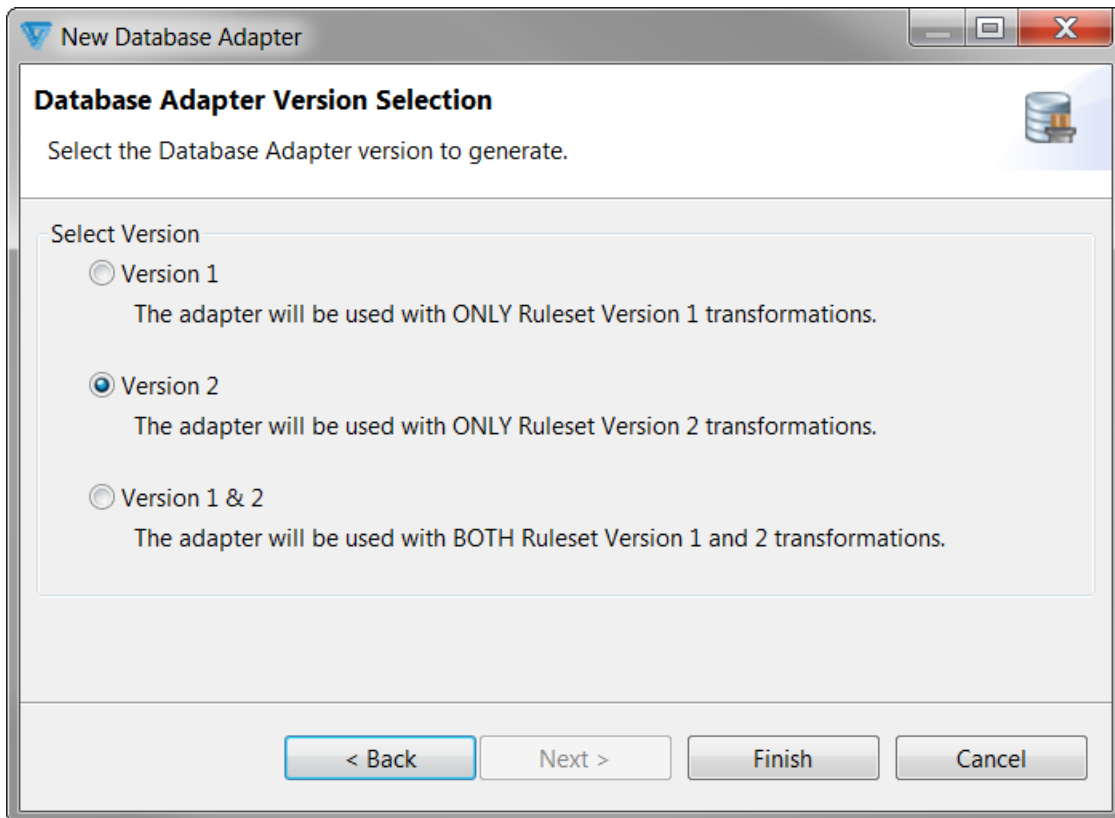
Database Adapter: Demo04DBA

Select the Package.

Select *File | New | Database Adapter*.

Type **Demo04DBA** in the *Name* field.

Accept **Version 2** as the version.



Click **Finish**. The object is created, and its Editor appears.

Click in the *Clarify Database Schema* field and select the *Demo04DB* object.

Click in the *Data Source* field and select the *Demo04DS* object.

The *Data Bindings* section lists the Schema's four tables in the *Schema* column. However, these Schemas must be related (or, "bound") to their corresponding Data Source table names.

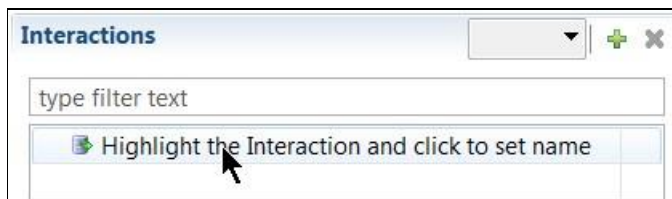
Double-click in the *Data Source* column's cell associated with the *ADDRESS* node and double-click the *ADDRESS* table.

Repeat for the *DETAIL*, *HEADER*, and *NOTES* tables, associating each Schema with the proper Data Source table.

Click the *Interactions* section's **Add** button.



Double-click the *Highlight the Interaction and click to set name* entry.



Type **insertPO**.

Click the dropdown and select *Insert* for the interaction type.



Multi-select all four tables in the *Data Bindings* section and drop them on the *insertPO* interaction.

Save the object and close the editor.

Transformation Settings: PipeDelimitedSourceTS

Select the Package.

Select *File* | *New* | *Transformation Settings*.

Type **PipeDelimitedSourceTS** in the *Name* field.

Click **Finish**. The object is created, and its editor appears.

Type a pipe in the *Delimiter* field of the *Source Connector* section.

Save the object and close the editor.

Ruleset: DelimitedFfToDbRS

This Ruleset transforms the data from the source format to the target format.

Select the Package.

Select *File | New | Ruleset*.

Type **DelimitedFfToDbRS** in the *Name* field.

Click **Next**. A wizard assists with defining this object.

The *Schema Types* panel allows for selecting both the source and target Schemas. Choose *Flat File* for the source and *Database* for the target. Click **Next**.

The *Source Schema* panel allows for selecting an existing Schema. Click **Browse**, select the *DelimitedFF* object, and click **OK**. Click **Next**.

The *Target Schema* panel allows for selecting an existing Schema. Click **Browse**, select the *Demo04OrderDB* object, and click **OK**.

Accept **Version 2** for the *Ruleset Version*.

Click **Finish**.

The object is created, and its editor appears.

Expand the *Source* section's *File: DelimitedFF* node and then its *Record Group: Source* node to display all record nodes. When a record's fields are required to create/define a rule, expand that record node as needed.

Click the **Expand All** button in the *Target* section to display all levels.

This Ruleset requires several Composite Rules and several Simple Rules for each.

Composite Rule 1: Record PO

To create and define this Composite Rule, drag the source's *Record: PO* node and drop it on the target's *Table: HEADER* node. This rule, which relates the source and target Schemas at the respective record levels, appears in the *Rules* section.

Simple Rules for Composite Rule 1

Ensure that this Composite Rule is selected before creating its child Simple Rules, as indicated here.

Source Record: Field	Target Table: Column
Value: U	HEADER: PROCESSED
PO: POnumber	HEADER: PO_NUMBER
PO: POdate	HEADER: DATE
CN: CNname	HEADER: CONTACTNAME
CN: CNphone	HEADER: PHONE
CN: CNemail	HEADER: EMAIL
TO: TOlines	HEADER: LINES
TO: TOquantity	HEADER: QUANTITY
TO: TOcost	HEADER: COST

As all purchase orders to be inserted into the database are initially unprocessed by the back-end application, this Ruleset's first Rule will mark them as such.

To create the first rule, click the **New Rule** button. In the list of available actions, type **mov** in the *Filter* field to restrict the display to actions containing that text. Double-click *Move* to select that action. The undefined rule appears in the *Rules* section.

To define the first rule, display the Properties view's *Rule* tab, type **U** in the *From* property and press **Enter**. Re-select this partially-defined Rule and drag the target's *Table: HEADER: Processed* node to the *Return Assignments* area.

To create and define the remaining rules, drag the source Schema node to the target Schema node.

Composite Rule 2: Record SI

To create this Composite Rule, right-click the previous Composite Rule and select *Add child | Composite Rule | Composite Rule*. The undefined rule appears in the *Rules* section.

To define this Composite Rule, drag the source's *Record: SI* node to the *Properties* view's *Source* property, and then drag the target's *Table: NOTES* node to the *Target* property.

Simple Rules for Composite Rule 2

Source Record: Field	Target Table: Column
PO: POnumber	NOTES: PO_NUMBER
SI: SInote1	NOTES: NOTE1
SI: SInote2	NOTES: NOTE2
SI: SInote3	NOTES: NOTE3

To create and define these rules, drag the source Schema node to the target Schema node.

Composite Rule 3: Record LI

To create and define this Composite Rule, right-click the previous Composite Rule and select *Add sibling / Composite Rule / Composite Rule*. The undefined rule appears in the *Rules* section.

To define this Composite Rule, drag the source's *Record: LI* node to the *Properties* view's *Source* property, and then drag the target's *Table: DETAIL* node to the *Target* property.

Simple Rules for Composite Rule 3

Ensure that this Composite Rule is selected before creating its child Simple Rules, as indicated here.

Source Record: Field	Target Table: Column
PO: POnumber	DETAIL: PO_NUMBER
LI: LInumber	DETAIL: LINENUMBER
LI: LyourSKU	DETAIL: YOURNUMBER
LI: LIdesc	DETAIL: DESCRIPTION
LI: LIbrand	DETAIL: BRAND
LI: LlourSKU	DETAIL: OURNUMBER
LI: LIqty	DETAIL: QTY
LI: Llunit	DETAIL: UNITS
LI: Llcost	DETAIL: UNITPRICE
LI: Llectcost	DETAIL: EXTENDEDPRICE

To create and define these rules, drag the source Schema node to the target Schema node.

Composite Rule 4: Record CI

To create and define this Composite Rule, right-click the previous Composite Rule and select *Add sibling / Composite Rule / Composite Rule*. The undefined rule appears in the *Rules* section.

To define this Composite Rule, drag the source's *Record: CI* node to the *Properties* view's *Source* property, and then drag the target's *Table: ADDRESS* node to the *Target* property.

Simple Rules for Composite Rule 4

Ensure that this Composite Rule is selected before creating its child Simple Rules, as indicated here.

Source Record: Field	Target Table: Column
PO: POnumber	ADDRESS: PO_NUMBER
CI: Clformat	ADDRESS: TYPE
CI: Clcode	ADDRESS: CUSTOMER_INDICATOR
CI: Clname	ADDRESS: NAME
CI: Cladd1	ADDRESS: ADDRESS1
CI: Cladd2	ADDRESS: ADDRESS2
CI: Clcity	ADDRESS: CITY
CI: Clstate	ADDRESS: STATE
CI: Clzip	ADDRESS: ZIP

To create and define these rules, drag the source Schema node to the target Schema node.

Composite Rule 5: Record ST

To create this Composite Rule, right-click the previous Composite Rule and select *Add sibling | Composite Rule | Composite Rule*. The undefined rule appears in the *Rules* section.

To define this Composite Rule, drag the source's *Record: ST* node to the *Properties* view's *Source* property, and then drag the target's *Table: ADDRESS* node to the *Target* property.

Simple Rules for Composite Rule 5

Ensure that this Composite Rule is selected before creating its child Simple Rules, as indicated here.

Source Record: Field	Target Table: Column
PO: POnumber	ADDRESS: PO_NUMBER
ST: STformat	ADDRESS: TYPE
ST: STcode	ADDRESS: CUSTOMER_INDICATOR
ST: STname	ADDRESS: NAME
ST: STadd1	ADDRESS: ADDRESS1
ST: STadd2	ADDRESS: ADDRESS2
ST: STcity	ADDRESS: CITY
ST: STstate	ADDRESS: STATE
ST: STzip	ADDRESS: ZIP

To create and define these rules, drag the source Schema node to the target Schema node.

Composite Rule 6: Record BT

To create this Composite Rule, right-click the previous Composite Rule and select *Add sibling | Composite Rule | Composite Rule*. The undefined rule appears in the *Rules* section.

To define this Composite Rule, drag the source's *Record: BT* node to the *Properties* view's *Source* property, and then drag the target's *Table: ADDRESS* node to the *Target* property.

Simple Rules for Composite Rule 6

Ensure that this Composite Rule is selected before creating its child Simple Rules, as indicated here.

Source Record: Field	Target Table: Column
PO: POnumber	ADDRESS: PO_NUMBER
BT: BTformat	ADDRESS: TYPE
BT: BTcode	ADDRESS: CUSTOMER_INDICATOR
BT: BTname	ADDRESS: NAME
BT: BTadd1	ADDRESS: ADDRESS1
BT: BTadd2	ADDRESS: ADDRESS2
BT: BTcity	ADDRESS: CITY
BT: BTstate	ADDRESS: STATE
BT: BTzip	ADDRESS: ZIP

To create and define these rules, drag the source Schema node to the target Schema node.

Save the object and close the editor.

Business Process: DelimitedFfToDbBPS

Select the Package.

Select *File | New | Business Process*.

Choose *No Template* and click **Next**.

Type **DelimitedFfToDbBPS** in the *Name* field.

Click **Finish**. The object is created, and its editor appears.

This Business Process requires three tasks to (1) read the source data, (2) transform the data, and (3) write the target data.

Task 1: Read the source data

To create this task, click the **Add** button in the *Script* section. Click that step's *Click to select task* entry. In the list of available tasks, type **read** in the *Filter* field to restrict the display to tasks containing that text. Double-click the *ReadTxtFA* user-defined object in this Project (*com.training.demo04.fftodb*) to select it. That task is displayed on the top line.

To define this task, click the *Properties* view tab.

This task has one parameter.

Click in the *storageNodes* parameter's field. This parameter represents the AdapterPayload (the data of the source file). The bold-italic text indicates that this is a mandatory parameter. A list of available parameters and variables appears. As none yet exist, that list is empty. Click **New Variable**. An AdapterPayload may be an Array, Object, or StorageNode. Double-click *StorageNode* to indicate it as the type for this variable. *newVariable* appears in the *Properties* view and also appears in the *Variables* section, with its name pre-selected, ready for a name change. Type **source** and press **Enter** to rename this variable. This new name is reflected in the *Properties* view.

Task 2: Transform the data

To create this task, click the **Add** button in the *Script* section. Click that step's *Click to select task* entry. In the list of available tasks, type **rs** in the *Filter* field to restrict the display to tasks containing that text. Double-click the *DelimitedFfToDbRS* user-defined object to select it. That task is displayed on the second line.

To define this task, click the *Properties* view tab.

This task has several parameters. Provide values for the first three; the last three are not necessary here.

Click in the *sourceNode* parameter's field. This parameter represents the StorageNode containing the data to be transformed. In the list of available parameters and variables, double-click the previously-defined *source* variable, as the data read by the first task is the data to be transformed.

Click in the *targetNode* parameter's field. This parameter represents the StorageNode containing the transformed data. A list of available parameters and variables appears. Click **New Variable**. *newVariable* appears in the *Properties* view and also appears in the *Variables* section, with its name pre-selected, ready for a name change. Type **target** and press **Enter** to rename this variable. This new name is reflected in the *Properties* view.

Click in the *settings* parameter's field. This parameter represents the Transformation Settings object to be used during transformation.

Click the *TransformationSettings Literal* tab and double-click the previously-defined *PipeDelimitedSourceTS* object in this Project (*com.training.demo04.fftodb*) to select it.

That task is displayed on the top line.

Task 3: Write the target data

To create this task, click the **Add** button in the *Script* section. Click that step's *Click to select task* entry. In the list of available tasks, type **insert** in the *Filter* field to restrict the display to tasks containing that text. Double-click the *Demo04DBA - insertPO* object to select it. That task is displayed on the third line.

To define this task, click the *Properties* view tab.

This task has only one parameter.

Click in the *adapterPayload* parameter's field. This parameter represents the AdapterPayload (the data to be inserted into the database). In the list of available parameters and variables, double-click the previously-defined *target* variable, as the data transformed by the second task is the data to be written.

Save the object and close the editor.

Deploy, Launch, Audit

Now that all of the Project's resources are created on the *Clarify Workbench | Project Explorer*, the remaining activities take place on various views of the *Admin Console* perspective.

The Project's top-level object – in this case, the Business Process – must be deployed to the *Local Test* server. Then, that Business Process can be launched and the result audited.

Deploy

Access *Admin Console | Projects*.

In the *Workspace* section, expand the *com.training.demo04.fftodb* Project, and then expand its Package node.

Drag the *DelimitedFfToDbBPS* object to the *Selected Scenarios* section.

Start the *Local Test* server by ensuring that it is displayed in the *Server Environment* dropdown and then click the **Start** button.

When fully-expanded, the *Server Projects* section shows all installed objects.

Launch

Access *Admin Console | Business Process*.

Select *com.training.demo04.fftodb.DelimitedFfToDbBPS* and click the **Launch** button.

Audit

Access *Admin Console | Auditor*.

The *Log Entries* section shows the results.
