Oracle® Endeca Information Discovery

Integrator Getting Started Guide

Version 2.3.0 • June 2012 • Revision A

ORACLE

Copyright and disclaimer

Table of Contents

Preface

Oracle® Endeca Information Discovery is an enterprise data discovery platform for advanced, yet intuitive, exploration and analysis of complex and varied data.

Information is loaded from disparate source systems and stored in a faceted data model that dynamically supports changing data. This integrated and enriched data is made available for search, discovery, and analysis via interactive and configurable applications.

Oracle Endeca Information Discovery enables an iterative “model-as-you-go” approach that simultaneously frees IT from the burdens of traditional data modeling and supports the broad exploration and analysis needs of business users.

About this guide

This guide discusses how to get started with the Oracle Endeca Information Discovery Integrator.

This guide provides a high-level overview of the Integrator, and then outlines how to use the Integrator on a single Windows machine to create your first Integrator project.

Who should use this guide

This guide is written for ETL developers and data architects who want to explore the basics of the Oracle Endeca Information Discovery Integrator.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ¬

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Oracle Customer Support

Oracle Customer Support provides registered users with important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at *https://support.oracle.com*.

Chapter 1

Welcome to the Oracle Endeca Information Discovery Integrator

The Integrator is a high-performance data integration platform that extracts source records from a variety of source types (from flat files to databases) and sends that data to the Endeca data store.

About the Integrator

The Oracle Endeca Information Discovery Integrator is a component-based ETL tool that allows you to build powerful data transformations in an easy-to-use graphical interface. Through merging, joining, filtering, mapping, reading, and writing data, you can build simple or complex transformations.

Within the Integrator, you drag components and connect them into graphs of varying complexity. A graph is essentially a pipeline of components that processes the data. The simplest graph has one **Reader** component to read in the source data and one of the Endeca components to write (send) the data to an Endeca data store. More complex graphs will use additional components, such as **Transformer** and **Joiner** components.

Integrator concepts

This topic lists some important concepts that will help you understand the Integrator.

Transformation *components* are graphical objects that represent data processing steps. *Graph* is the formal term for the graphical layout that contains a set of transformation components.

An *edge* is the join line that connects two components by way of output and input *data ports*. Every component has one or more input ports, and one or more output ports. (The only exception is the **Trash** component, which has no output port.) Different components use ports differently.

Metadata describes the format of the data, and must be assigned to each edge. When connected with an edge, metadata automatically defines the output data format of one component and the input data format of the component it is connected to.

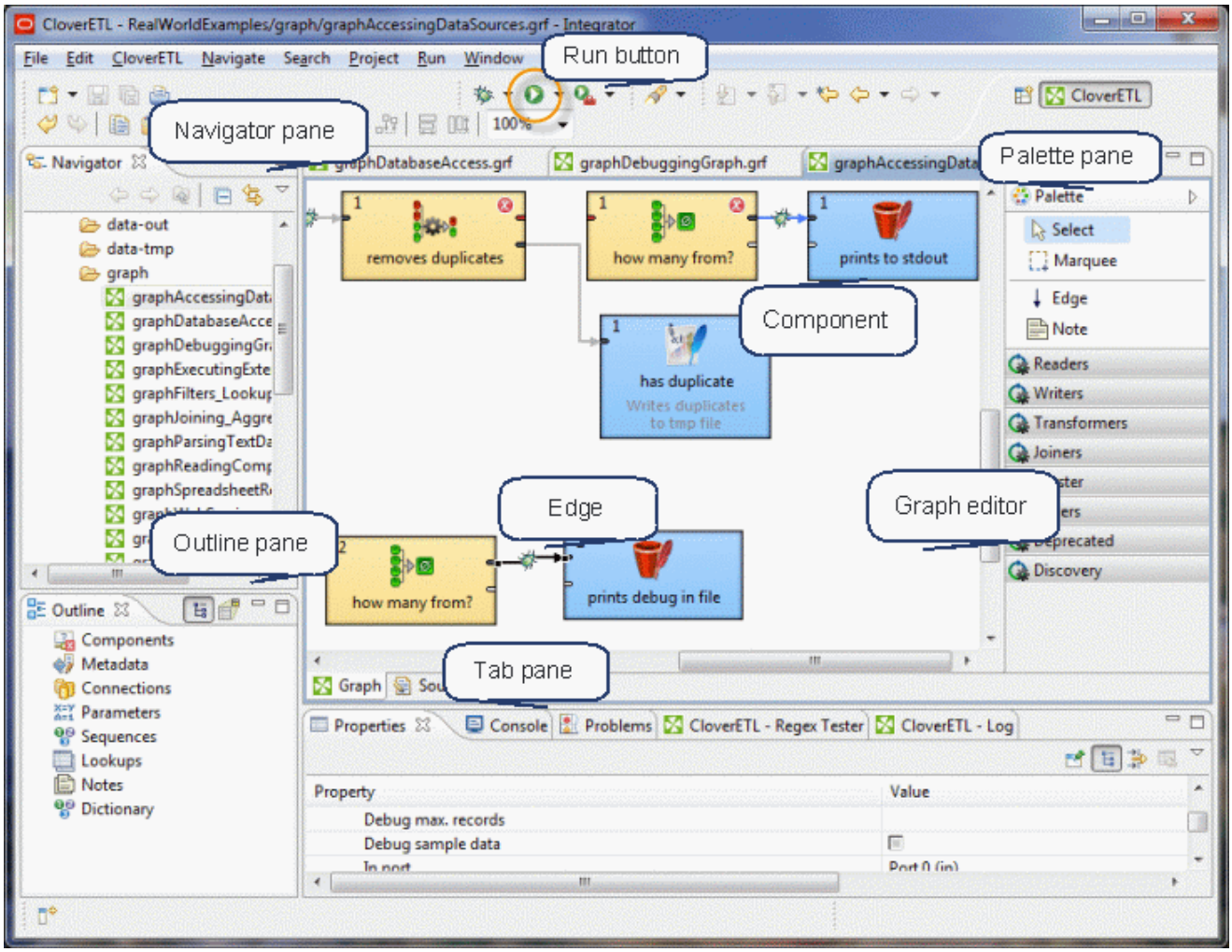
Integrator components process data in rows. By default, Integrator passes a row of data to the next component as soon as that single row has been processed. When a component has processed data, it will almost always send the data to one or more output ports.

Integrator is *multi-threaded*. In practical terms this means it tries to run components in parallel. As a row of data is processed, it is passed immediately to the next component. Integrator does not wait until all rows have been read in from the file or the previous component. This makes it very scalable across processor cores.

Integrator can also run *phases*, which allow you to sequence activities. You can specify that one phase must be completed before another can begin. This allows Integrator to make sure that certain tasks are fully complete before another task begins.

Parts of the Integrator window

This topic provides an overview of the Integrator application. The Integrator is where you construct your data transformations.



The definitions below correspond to the highlighted areas in the image above:

- The **Graph** editor is where you construct your graphs. Graphs consist of transformation **components** linked by **edges**. You can toggle between a graphical and an XML source view of your graph.
- The **Navigator** pane lists Integrator project files.

- The **Outline** pane lists all of the components in the selected graph. It allows you to access and edit things like components properties and metadata definitions.
- The **Palette** tool contains a library of available components clustered by type, as well as the edge and note controls. You drag components from the **Palette** to the **Graph** editor.
- The **Run** button kicks off pipeline processing (the set of operations that manipulate your data).
- The **Tab** pane consists of a series of tabs (such as the **Properties** tab and the **Log** tab) that provide information about the components and the results of graph executions. The illustration shows the **Console** tab logging pipeline processing in real time.

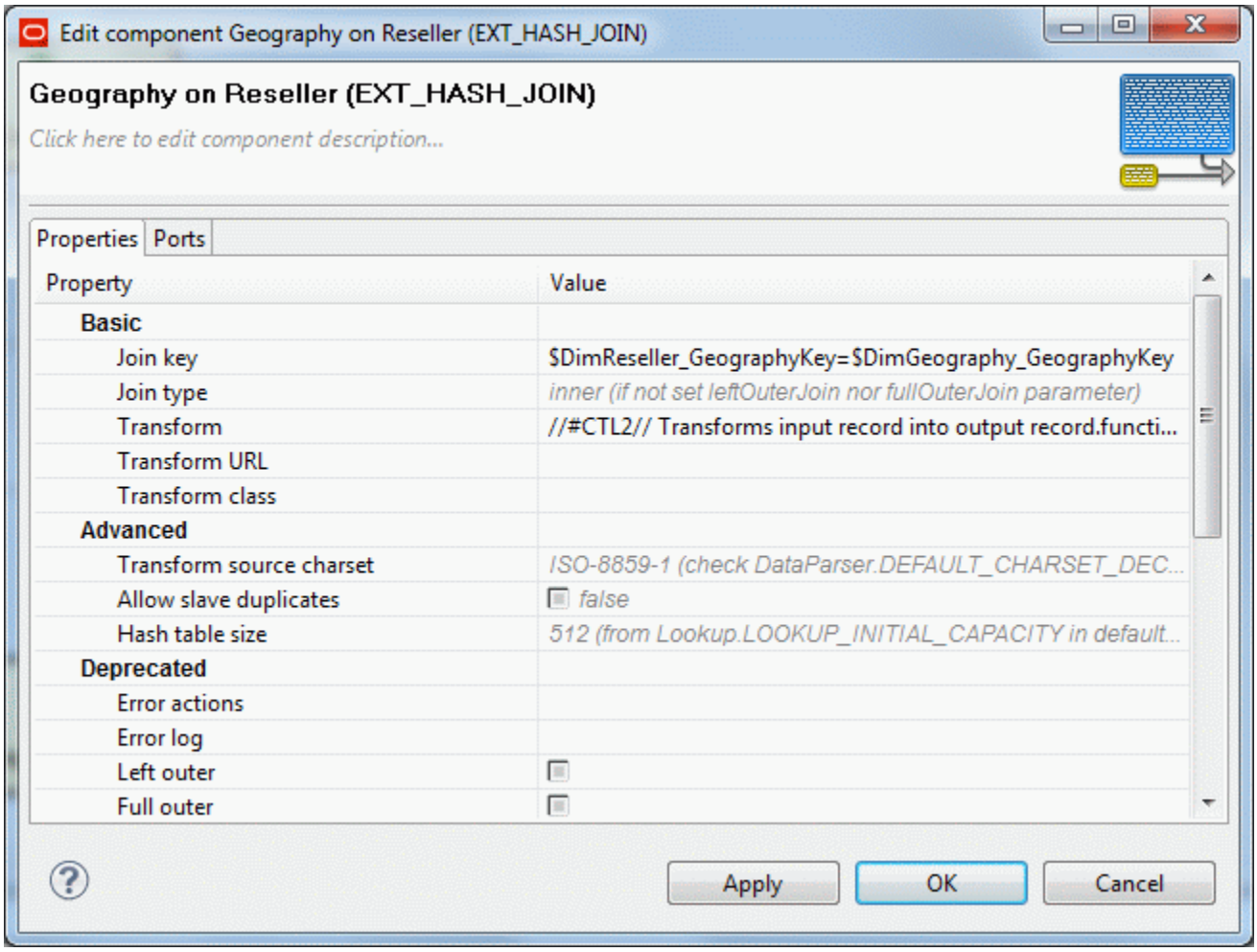
Types of components

Integrator provides several types of transformation components.

Transformation components are the fundamental processes that operate on data. They are divided up into the following sections in the **Palette** pane:

- The **Readers** section contains the various components for reading in data from any external data source.
- The **Writers** section contains components that are responsible for outputting data from the transformation into databases, files, LDAP, and other targets.
- The **Transformers** section contains components that help you transform your data from one format to another. Transformers make it possible to map fields, sort, merge, aggregate, normalize, denormalize, and so on.
- The **Joiners** section contains components that combine various data sources into one or more data outputs.
- The **Clusters** section contains components that merge data flows from cluster nodes.
- The **Others** section includes the **WebServiceClient** component for accessing Web services, as well as less frequently used components like the **Sequence Checker** and the **Speed Limiter**.
- The **Discovery** section provides access to Endeca-specific components like **Add Managed Values**.

Double-clicking on any component allows you to view and edit component properties, as shown in the example below:



Obtaining more information about the Integrator

This guide provides a quick and simple introduction to the Integrator.

For more in-depth information about the Oracle Endeca Information Discovery Integrator, see the following guides:

- The *Oracle Endeca Information Discovery Integrator Guide* provides information about Integrator components and functionality.
- The *Oracle Endeca Information Discovery Integrator Components Guide* provides details about Endeca-specific components.
- The *Oracle Endeca Information Discovery Integrator Server Guide* provides information about the Oracle Endeca Information Discovery Integrator Server, which provides centralized ETL job management and integration into enterprise workflows.

All of these guides are part of the downloadable Oracle Endeca Information Discovery documentation set.

Chapter 2

Running the Integrator Quick Start Project

The Integrator comes with a sample project and data, which you can use to get started.

About the Integrator Quick Start project

The Integrator comes with a sample project and data, which you can launch from the Integrator Welcome screen.

The Integrator Quick Start project demonstrates the Oracle Endeca Information Discovery product in action, using sales and product data from a fictitious bicycle manufacturer.

Note: For details of working with the Integrator Quick Start project, see the *Oracle Endeca Information Discovery Quick Start Guide*.

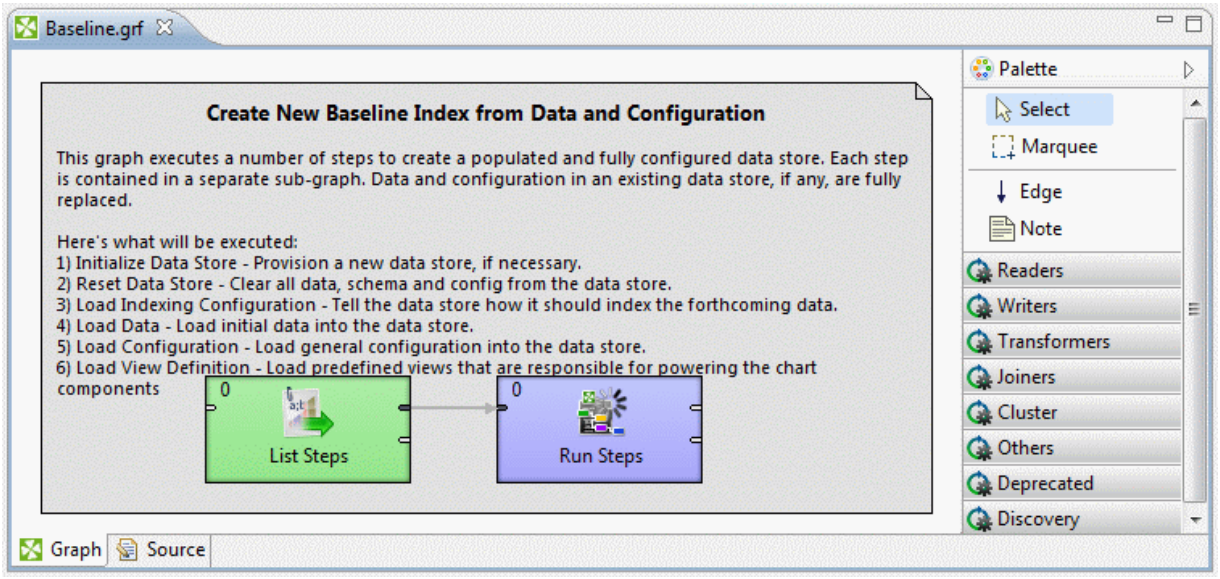
Launching the Integrator Quick Start project

The Quick Start project is installed with the Integrator.

To launch the Integrator Quick Start project:

1. Do one of the following:
   - In the Integrator Welcome screen, click "Open and view the Quick Start project."
   - In the Integrator, select **File>New>Quick Start Example**.

   Integrator opens with a fully-formed project, ready for you to explore or run.


Baseline.grf
Create New Baseline Index from Data and Configuration
This graph executes a number of steps to create a populated and fully configured data store. Each step is contained in a separate sub-graph. Data and configuration in an existing data store, if any, are fully replaced.
Here's what will be executed:
1) Initialize Data Store - Provision a new data store, if necessary.
2) Reset Data Store - Clear all data, schema and config from the data store.
3) Load Indexing Configuration - Tell the data store how it should index the forthcoming data.
4) Load Data - Load initial data into the data store.
5) Load Configuration - Load general configuration into the data store.
6) Load View Definition - Load predefined views that are responsible for powering the chart components
0
List Steps
0
Run Steps
Palette
Select
Marquee
Edge
Note
Readers
Writers
Transformers
Joiners
Cluster
Others
Deprecated
Discovery
Graph
Source

Chapter 3

Building a Simple Project from Scratch

Instead of launching the sample project, you can build your own simple project from scratch, as described in this section.

Starting the Integrator

This topic describes how to start the Integrator.

To start the Integrator:

1. From the **Start** menu, choose **All Programs>Oracle>EID Integrator 2.3.0>Oracle EID Integrator**.
2. Depending on how your Integrator is configured, you may be asked to select or confirm your workspace.

   The workspace is the directory where Integrator creates and stores your projects.
3. The first time you launch Integrator, a **Welcome** screen appears. Use this screen to navigate to launch Integrator or to load the Quick Start sample application.

Note: You can return to the **Welcome** screen at any time by clicking **Help>Welcome**.

Creating a project

This topic describes how to create a new Integrator project.

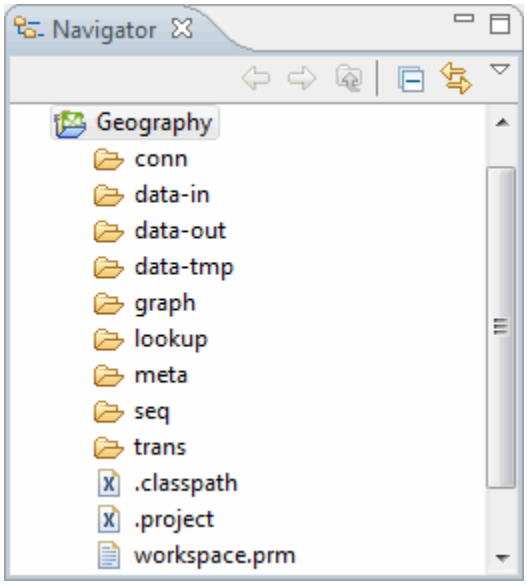
To create a new Integrator project:

1. In the Integrator, select **File>New >Clover ETL Project**.

If you are running Eclipse for the first time, you may not see **Clover ETL Project** on the menu. In this case, select **Other>CloverETL>Clover ETL Project**.

2. In the **Create a new Clover project** dialog box, type the project name (we use **Geography**), set the directory location, and then click **Next**.
3. In the **Configure CloverETL project subdirectories** dialog box, accept the default project directory locations and click **Finish**.
4. If you are asked about using the Clover Perspective, say **Yes**.

A project called **Geography** appears in the **Navigator** pane. You can expand this to see the folders beneath it, all of which are currently empty.



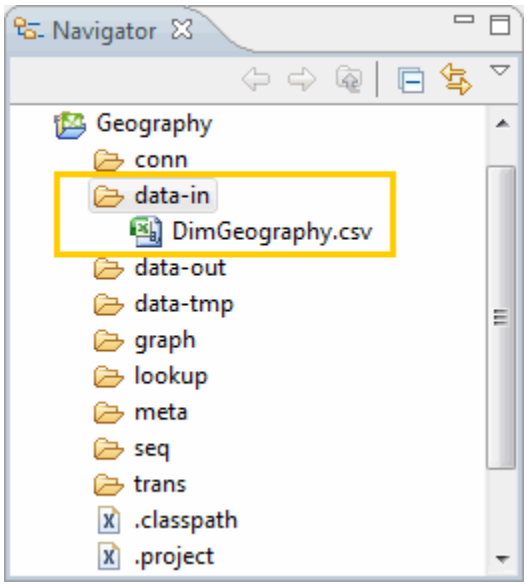
Adding the sample data

In this topic, you add some data to the project.

To get going quickly, you will copy a sample data file from the Quick Start project.

To load the sample data:

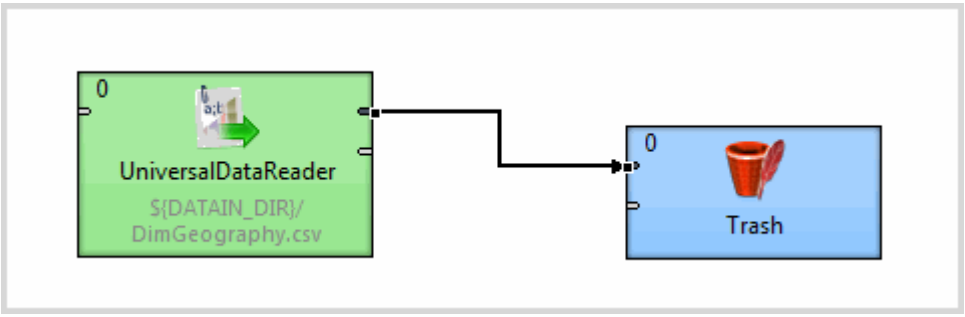
1. In Integrator, load the Quick Start application with **File>New>QuickStart Example**.
2. In the Quick Start project, open the `data-in` folder and copy the `DimGeography.csv` file.
3. Paste the `DimGeography.csv` file in the `data-in` directory of your new **Geography** project. Your `data-in` directory should look like this:



Building your first graph

We are now ready to start building a transformation graph.

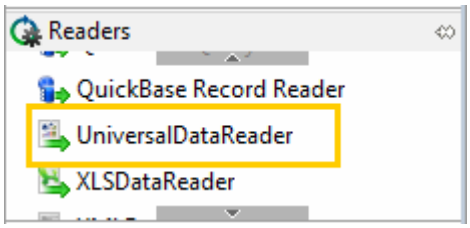
This simple graph contains two components connected by a single edge.



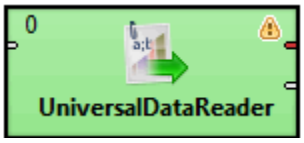
Adding a new component

In this topic, you add a component to read the geography data.

1. In the Navigator, right-click **Geography** and select **New>ETL Graph**.
2. Name it **LoadGeography**.
3. Click **Next** and then click **Finish**.
   An empty graph called **LoadGeography.grf** appears in the **Graph** editor.
4. In the **Palette**, click the section called **Readers** to open it.
5. Select **Universal Data Reader** and drag it onto the **Graph** editor.



The **LoadGeography.grf** now contains a single **UniversalDataReader** component.

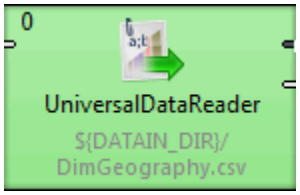


Adding data to the component

After creating the **UniversalDataReader** component, we need to associate data with it.

To add data to the component:

1. Double-click the **UniversalDataReader** component to open the **Edit Component** dialog box.
2. Click the **File URL** property.
3. Click the browse **(...)** button to the right of the **File URL** property.
4. In the **URL Dialog** dialog box, double-click the **data-in** folder to open it, and then select `DimGeography.csv`.
5. Click **OK** to return to the **Edit Component** dialog box.
6. Check the **Quoted strings** property to set it to true.
   This step is necessary because the `DimGeography.csv` data contains quoted strings.
7. Locate the **Number of skipped records** property and set this to 1.
   This ensures that header field names are not read in as proper data.
8. Click **OK** to return to the graph. The **UniversalDataReader** contains a reference to its data source.

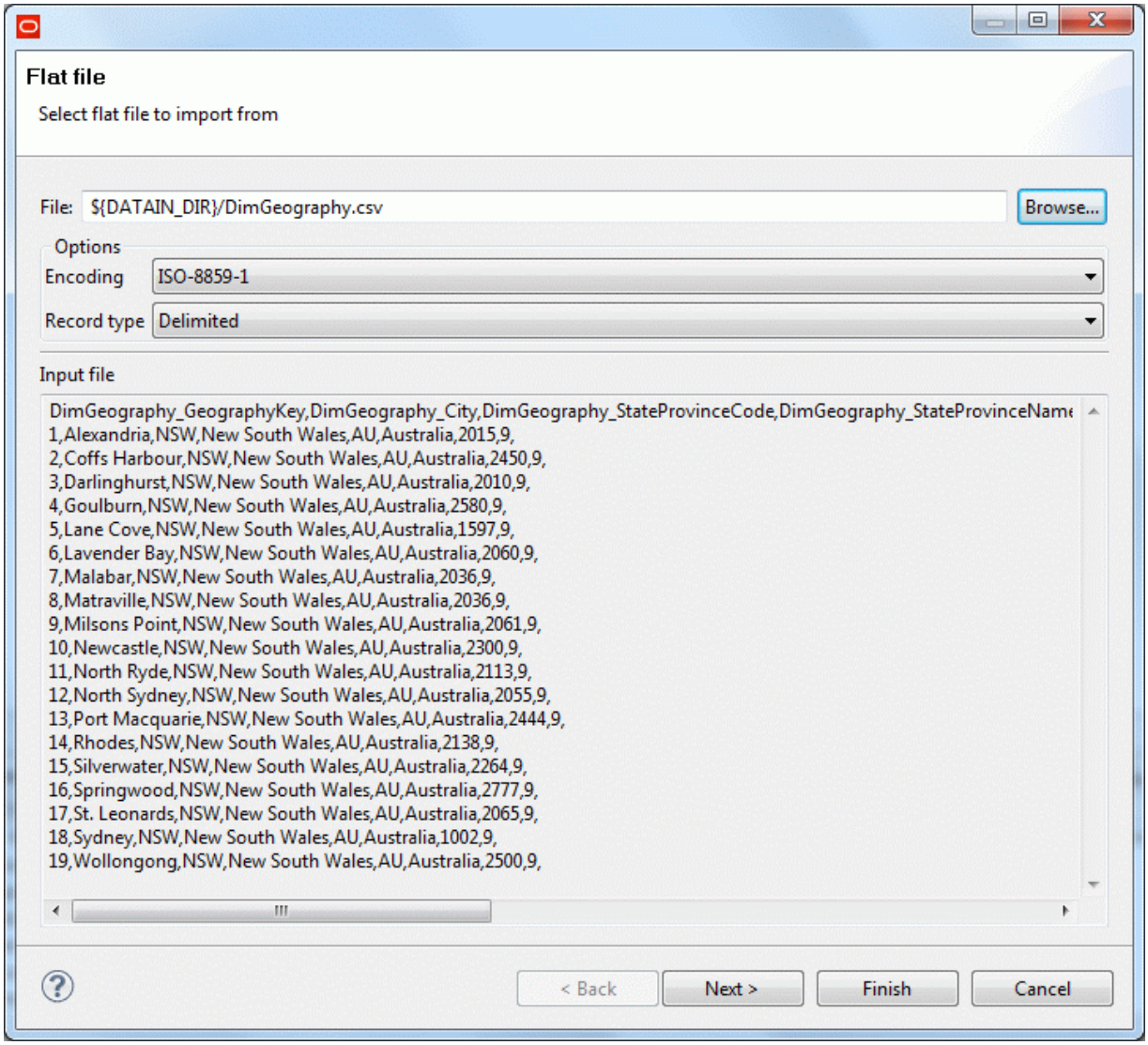
9. Save the **LoadGeography** graph.

Defining metadata for the geography data

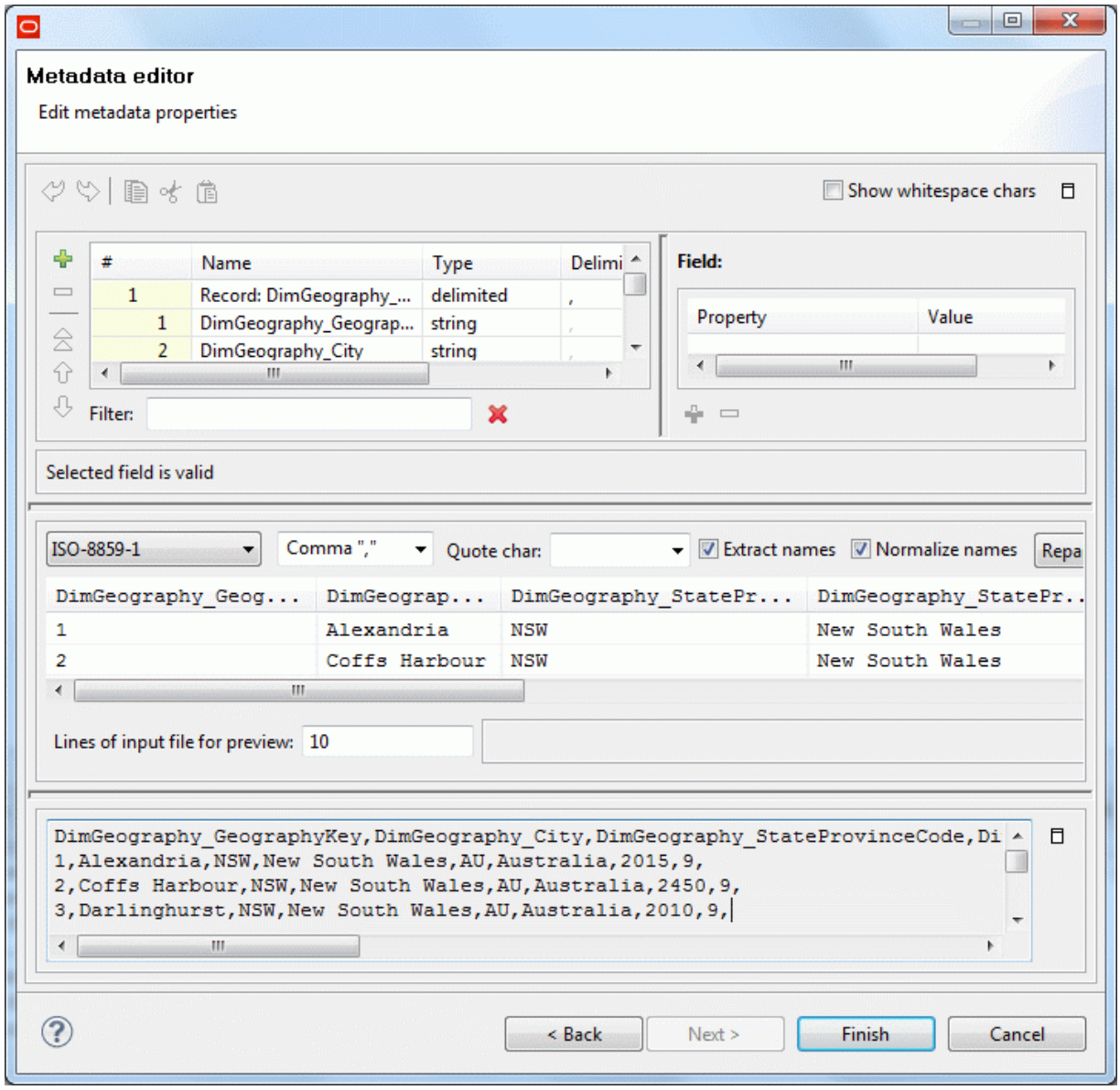
In order to pass data from the **UniversalDataReader** to another component, you have to define metadata that can be assigned to the edge that will join them together.

To define metadata:

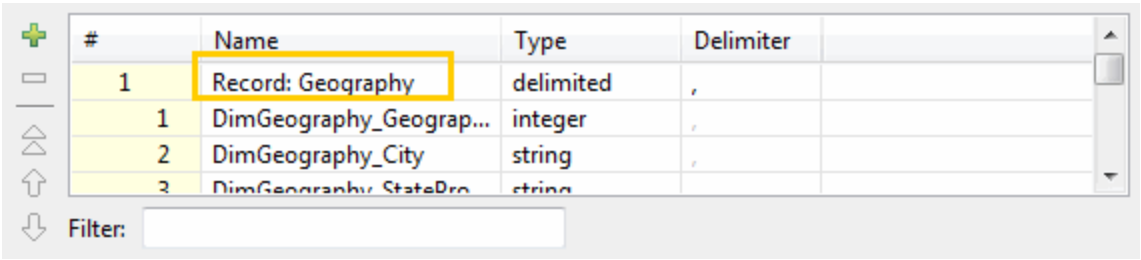
1. In the **Outline**, right-click **Metadata** and select **New metadata>Extract from flat file**.
2. In the **File** text box, type or browse to the full path to your `DimGeography.csv` file and then press **Enter**.



3. Click **Next** to see the **Metadata** editor, where you can edit metadata properties.

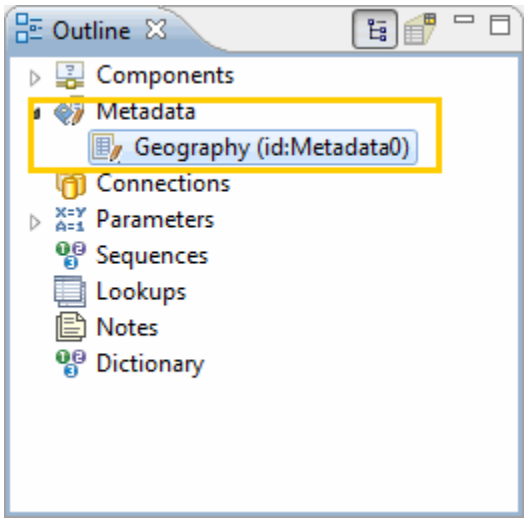


4. To give the metadata a useful name, rename the topmost record in the **Fields** list to **Geography**.



Note: Make sure you tab out of this field. Otherwise it will not be saved correctly.

5. Click **Finish**.



The **Geography** metadata item now appears in the **Metadata** collection. To edit the metadata, double-click it.

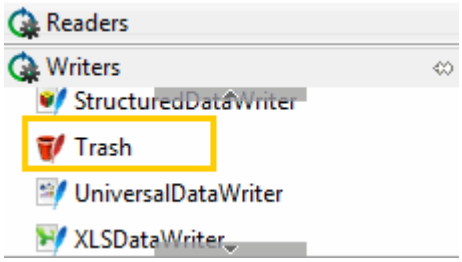
Adding a Trash component

In this topic, you add a **Trash** component.

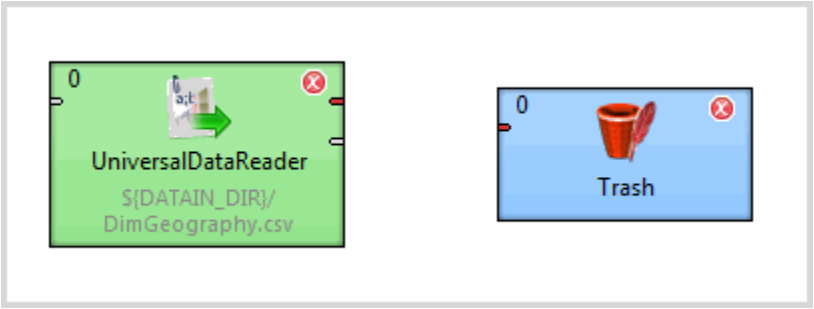
The **Trash** component tests the end points in a graph. Any data that arrives in the **Trash** component is discarded, which means that there is no need to create a file or database output. The **Trash** component also allows you to use some of the debugging capabilities of the Integrator to monitor graph execution.

To add a **Trash** component:

1. In the **Palette**, click the section called **Writers** to open it.
2. Select **Trash** and drag it onto the **Graph** editor.



The **LoadGeography.grf** now contains two unconnected components.

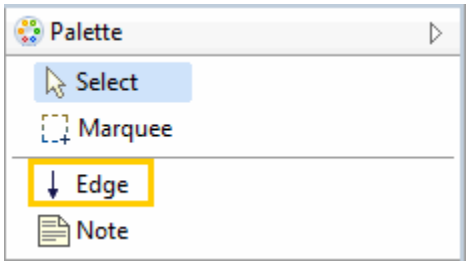


Connecting two components with an edge

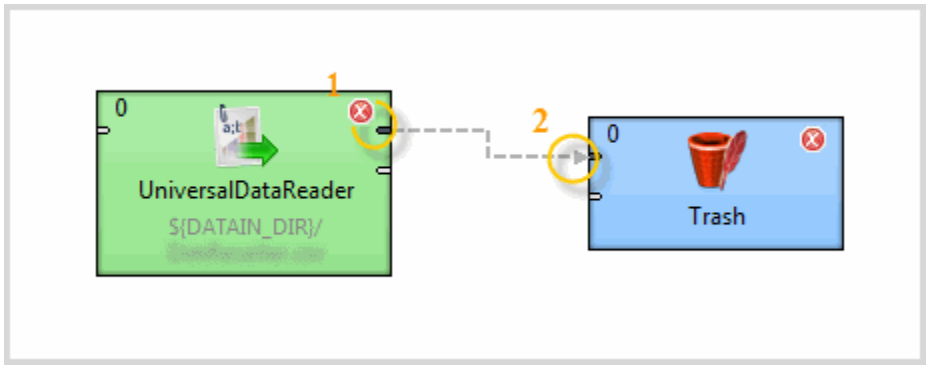
In this topic, you connect the **Trash** component to the **UniversalDataReader** component with an edge.

To connect two components with an edge:

1. In the **Palette**, select the **Edge** tool.

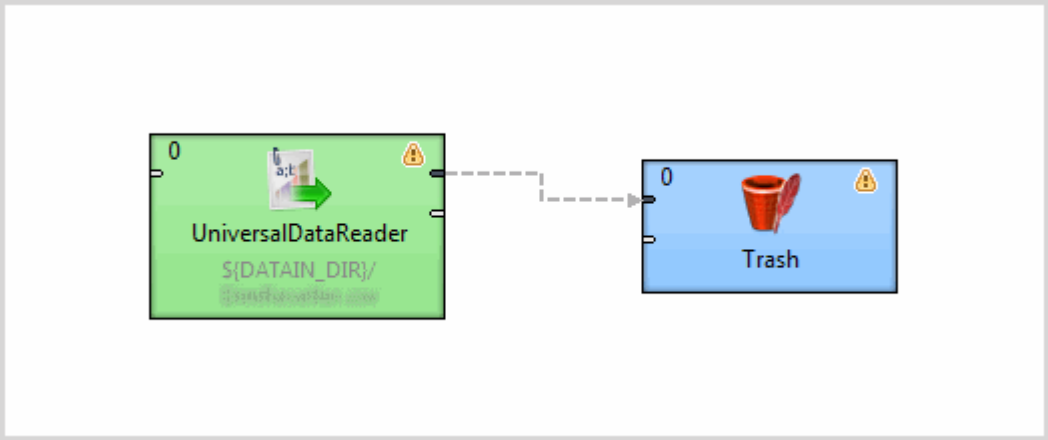


2. Click on the upper output port of the **UniversalDataReader** (number 1 in the image below) and drag across to the upper input port of the **Trash** component (number 2 in the image below).
You have to click on the target component to connect the edge.



3. Press **Esc** to change from Edge mode back to Select mode.
4. Save the graph file.

The **LoadGeography.grf** now contains two components connected by an edge.



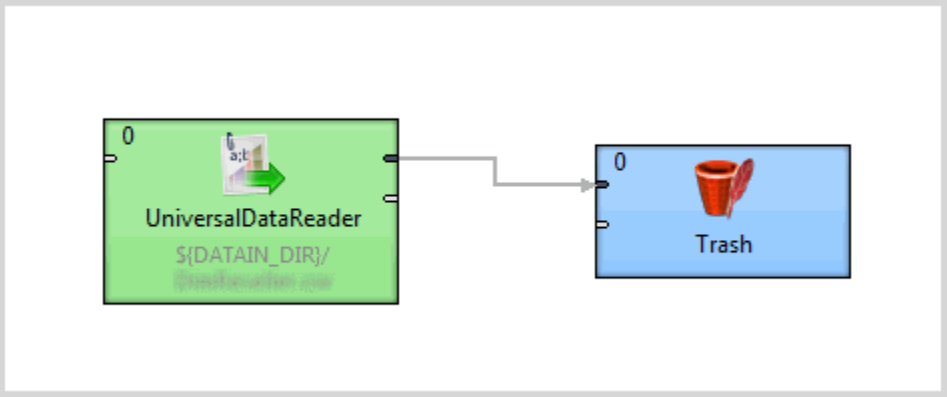
Assigning metadata to the edge

In this step, you apply the metadata that you created earlier to the edge.

To assign metadata to an edge:

1. Right-click on the edge connecting the **UniversalDataReader** and **Trash** components.
2. Choose **Select Metadata>Geography**.
   The edge becomes a solid, rather than a dashed, line, which indicates that metadata is associated with it.
3. Save your graph.

The **LoadGeography.grf** now is now ready to be run.



Running the graph

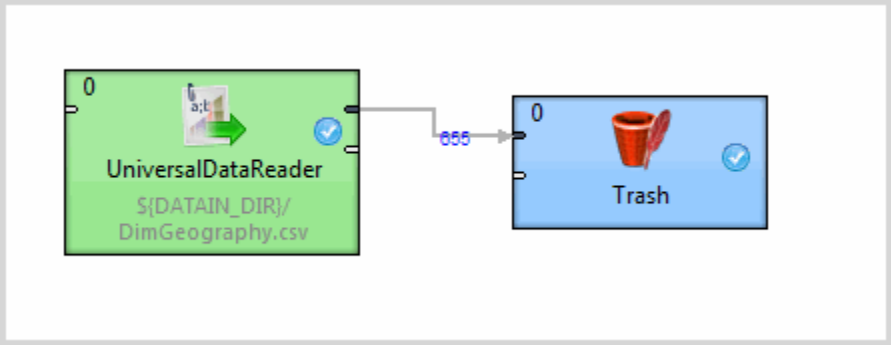
After creating the graph and configuring the components, you can run the graph.

To run your Integrator graph:

1. Run your graph in one of three ways:
   - Select **Run>Run As>CloverETL graph** from the main menu.

- Right-click in the **Graph** editor and select **Run As>CloverETL graph**.
- Click the green circle with white triangle icon in the toolbar.

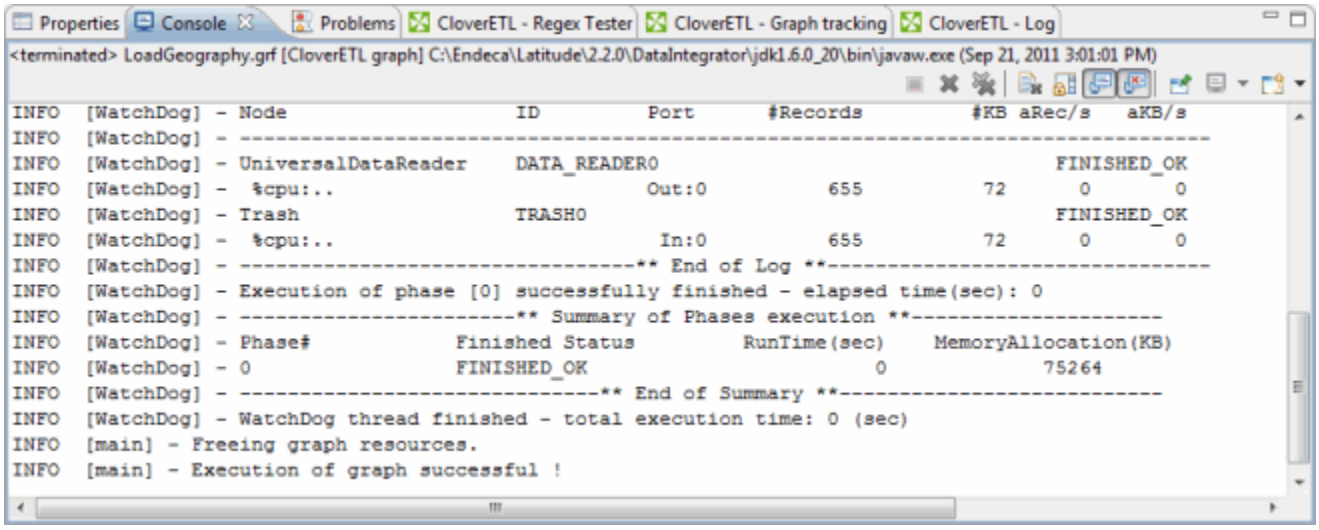
Upon successful execution, the components are flagged with a check mark, and the edge displays the number of records processed.



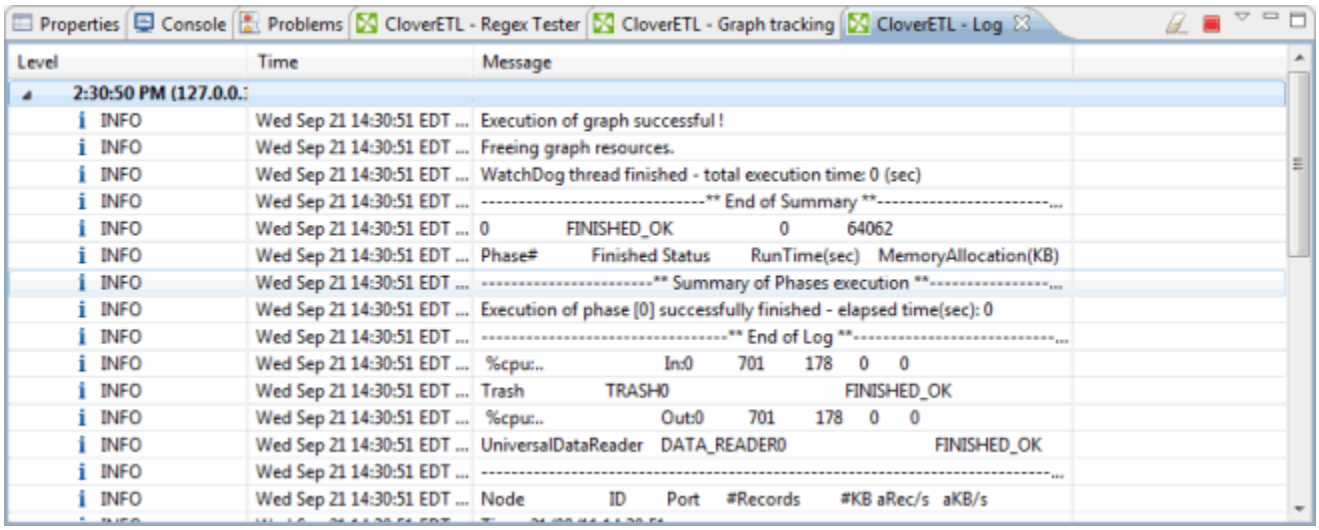
Checking the output

Using the **Console** and the **Clover Log**, you can check the details of graph execution.

When a job runs, a **Console** window opens up at the bottom of the Eclipse window below the graph workspace. The **Console** logs the output for graph execution. The image below shows the **Console**:



Alternately, click the **CloverETL - Log** tab to view more concise output. The image below shows the **CloverETL - Log**:



Note: If you cannot see both of these tabs, go to the **Window** menu option and select **Reset Perspective**.

Debugging the graph

Debugging is a vital (and easy-to-use) feature that lets you see exactly what data was passed along any edge in a graph.

Note: When debugging an Integrator graph, keep in mind that all components in the same phase run in parallel and are multi-threaded. Therefore, make sure you start with components that are flagged as errors (with a red exclamation point) and not with warnings (with a yellow question mark), even if the warnings appear to occur logically before the errors.

To debug a graph:

1. Right-click the edge and select **Enable Debug**.
   The debug icon appears on the edge.
2. Re-run the graph so that Integrator can generate the debug data.
3. Right-click the edge and select **View data**.
   The **View data** window shows all of the data fields correctly parsed and loaded.

Viewing the XML source for the graph

When you create a graph, it is saved as XML. You can view and edit this XML source.

To see the XML source for the graph:

1. At the bottom of the **Graph** editor, click the **Source** tab.
2. In the XML version of **LoadGeography.grf** (shown below), scroll to view the data. Any changes you make are automatically applied to the graphical version.

Sending data to the Endeca data store

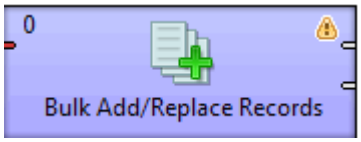
In this topic, you will replace the **Trash** component with a component that sends records to the Endeca data store.

The procedure below assumes that the Endeca Server is running (using the default 7770 port) and that you have created a running Endeca data store named **geography**. Creating Endeca data stores is documented in the *Oracle Endeca Server Administrator's Guide*.

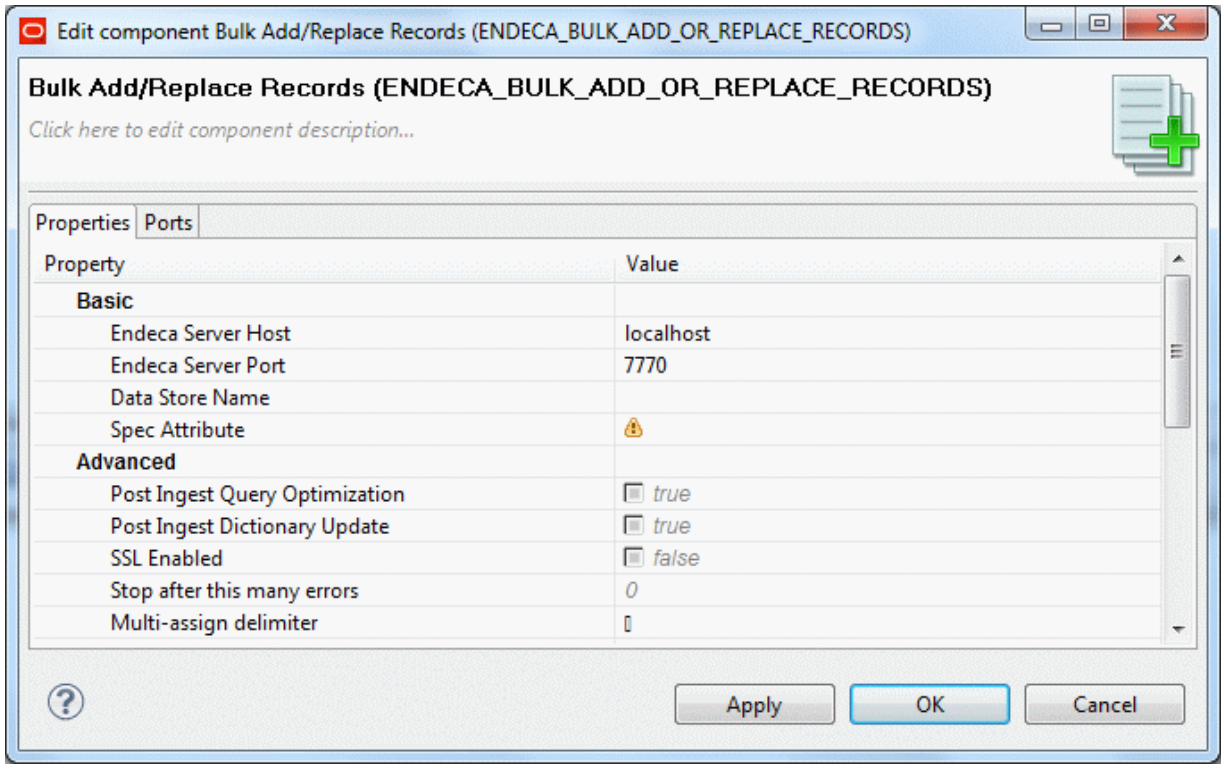
For a detailed walkthrough of the Endeca data store load process, see the " Full Initial Load of Records" chapter in the *Oracle Endeca Information Discovery Integrator Components Guide*.

To send data to an Endeca data store:

1. In the **Navigator**, select the **graph** folder in the **Geography** project.
2. In the **Palette**, click the section called **Discovery** to open it.
3. Select **Bulk Add/Replace Records** and drag it onto the **Graph** editor.

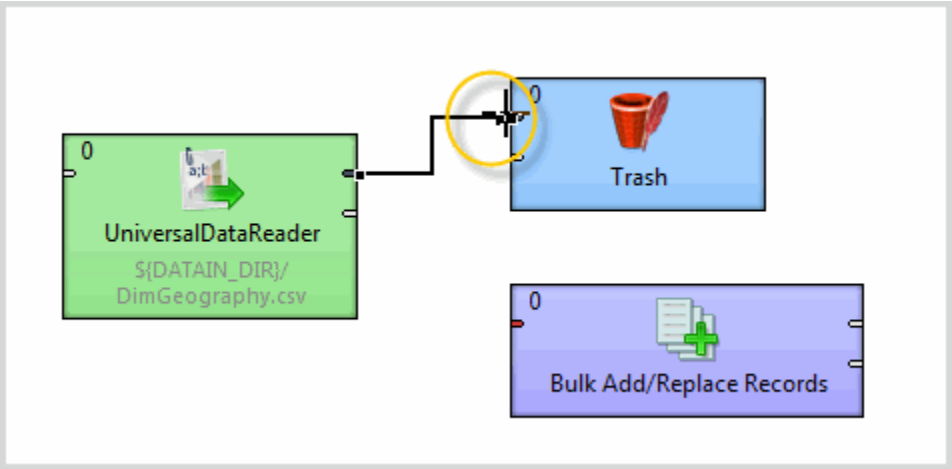


4. Double-click the **Bulk Add/Replace Records** component to open the **Edit Component** dialog box.

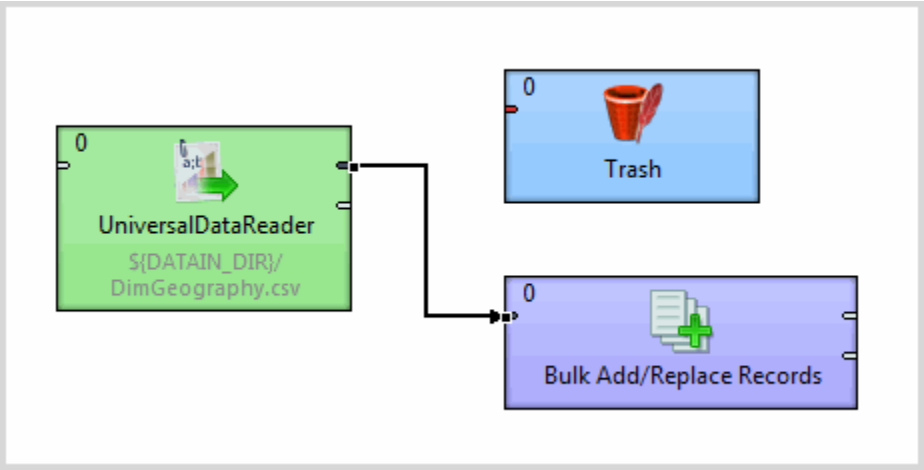


5. Set the following mandatory properties in the **Basic** section, and then click **OK**:
   (a) **Endeca Server Host**: The name or IP address of the machine. The default value (`localhost`) can be used as the name.
   (b) **Endeca Server Port**: 7770 (the value of the default Endeca Server port)
   (c) **Data Store Name**: The name of the Endeca data store. In this example, `geography` is the name.
   (d) **Spec Attribute**: The name of the primary key. In this example, `DimGeography_GeographyKey` is the name.

6. In the **Graph** editor, position the cursor over the input port of the **Trash** component so that the hand cursor becomes a +.



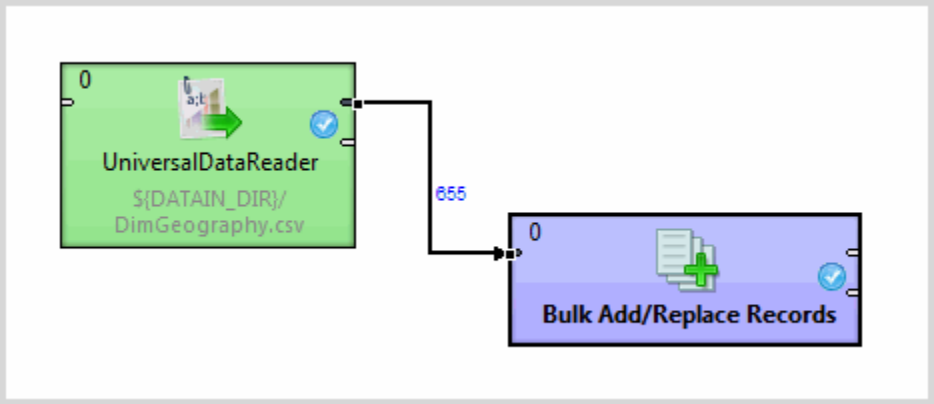
7. Drag the edge endpoint from the **Trash** component to the **Bulk Add/Replace Records** component that you just configured.



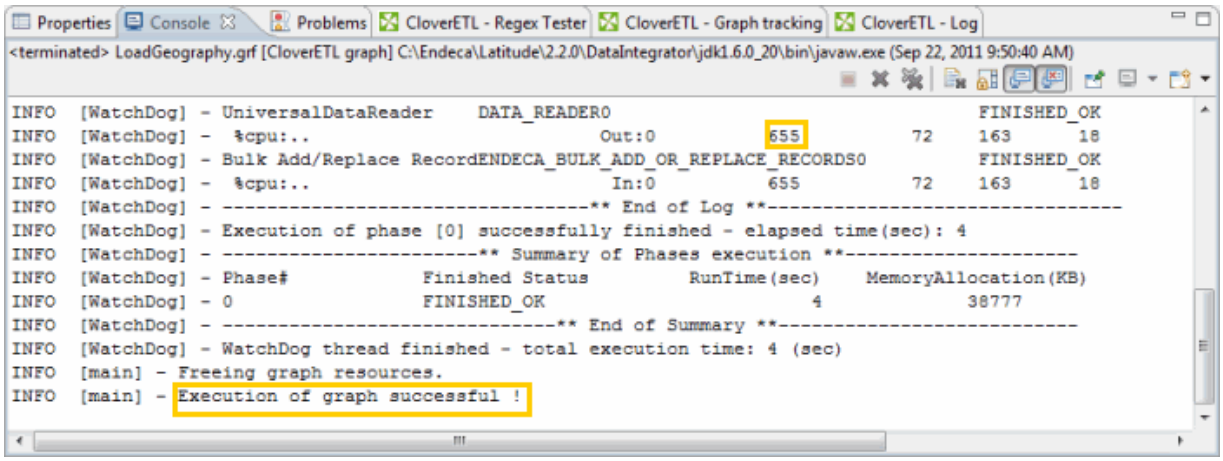
When you move the edge, its associated metadata is also moved.

8. Delete the **Trash** component.
9. Save the graph.
10. Run the graph.

Upon successful execution, the components are flagged with a check mark, and the edge displays the number of records processed.



In addition, you will see the following success message in the **Console**:


Properties
Console
Problems
CloverETL - Regex Tester
CloverETL - Graph tracking
CloverETL - Log
<terminated> LoadGeography.grf [CloverETL graph] C:\Endeca\Latitude\2.2.0\DataIntegrator\jdk1.6.0_20\bin\javaw.exe (Sep 22, 2011 9:50:40 AM)
INFO [WatchDog] - UniversalDataReader DATA_READER0 FINISHED_OK
INFO [WatchDog] - %cpu:.. Out:0 655 72 163 18
INFO [WatchDog] - Bulk Add/Replace RecordENDECA_BULK_ADD_OR_REPLACE_RECORDS0 FINISHED_OK
INFO [WatchDog] - %cpu:.. In:0 655 72 163 18
INFO [WatchDog] - ----------** End of Log **----------
INFO [WatchDog] - Execution of phase [0] successfully finished - elapsed time(sec): 4
INFO [WatchDog] - ----------** Summary of Phases execution **----------
INFO [WatchDog] - Phase# Finished Status RunTime(sec) MemoryAllocation(KB)
INFO [WatchDog] - 0 FINISHED_OK 4 38777
INFO [WatchDog] - ----------** End of Summary **----------
INFO [WatchDog] - WatchDog thread finished - total execution time: 4 (sec)
INFO [main] - Freeing graph resources.
INFO [main] - Execution of graph successful !

Chapter 4

Troubleshooting Tips

This section contains some tips to keep in mind as you begin working with the Integrator.

Editing Reader component properties

When editing **Reader** component properties, keep in mind the following details.

Set "Number of skipped records" for Reader components

If your data file has a header row, make sure to set **Number of skipped records** to 1. If it has no header row, it should be set to 0.

Be aware of quoted strings in your data

If there are quoted strings in your data, make sure you check **Quoted strings**.

Adding and editing metadata

When adding and editing metadata for a component, keep in mind the following details.

When it begins processing your data, Integrator makes some assumptions about your data, but it is up to you to confirm that these assumptions are correct.

For example, when loading metadata from a flat file, Integrator will attempt to ascertain whether the file is fixed length or delimited. Make sure that Integrator has in fact assigned the correct type. You may also want to check the data type assigned to each field within your data, and the delimiter used to separate values.

Index