

|  |
| --- |
|  |

**Pentaho Data Integration**

Neo4J Output

Step Documentation

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Date** | **Version** | **Reason** |
| Bart Maertens | 2017-09-02 | 1.0 | Initial Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Description

The Pentaho Kettle Neo4J output step allows you to load nodes and properties into a Neo4J graph database.

Neo4J is a graph property database that stores data as a set of nodes and relationships.

Nodes can have labels (representing a node’s role in the database) and properties (attributes to the node).

Relationships provide directed, named semantically relevant connections between two node-entities. A relationship has a direction, a start node and an end node.

Like nodes, relationship can have properties.

This step allows you to create your entire graph database:

* **nodes** with their labels and properties
* **relationships** between two nodes and the relationship’s properties

The step uses the Neo4J binary BOLT protocol

# Options

The table below describes the available options for the Neo4J Output step.

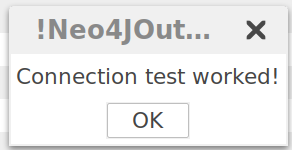
## Connection properties

  
Image 1: Neo4J connection properties

| **Property** | **Description** | **Default** |
| --- | --- | --- |
| Host | The Neo4J server host name or IP address | localhost |
| Port | The Neo4J server (BOLT) port | 7687 |
| Username | The Neo4J server username | - |
| Password | The Neo4J server password | - |
| Test | Button to test the Neo4J connection parameters |  |

## After entering all connection parameters, the connection can be verified through the ‘Test’ button:

## 

  
Image 2: Connection Test

## Properties

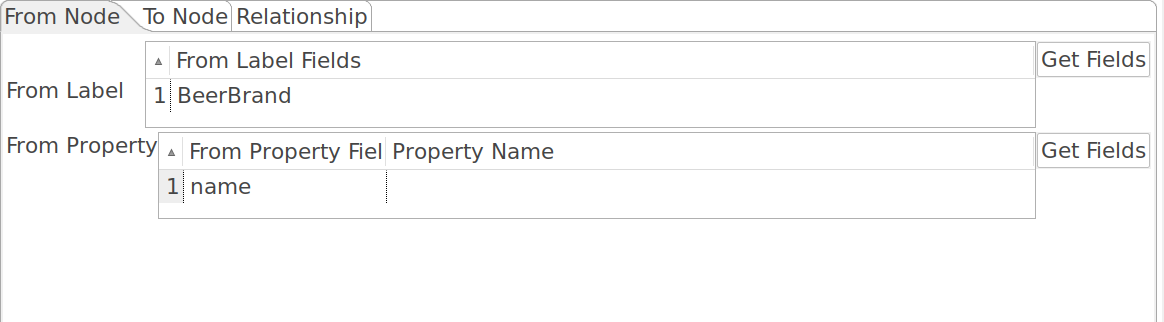
## Three tabs can be used to provide from and to node labels and properties and relationship names and properties.

  
Image 3: Node and relationship properties

## From/To Nodes

Node labels and properties (for ‘From’ and ‘To’ nodes) can be entered in the dialogs.

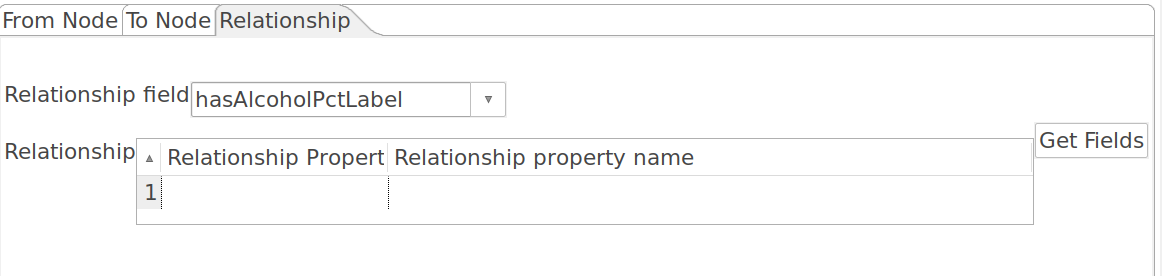
* **Labels** use the selected field’s value as the node’s label
* **Properties** use the selected field’s value as the property value and the field’s name as the property name. A property’s name can be overruled in the ‘Property Name’ field .

  
Image 4: Node labels and properties

## Relationships

When ‘From’ and ‘To’ nodes have been specified, relationship can be created.

* A **Relationship** field’s value will be used to name the relationship
* **Properties** can be added to a relationship. Similar to node properties, a relationship property’s field value will be used for the property value. The field name will be used for the property name, unless overruled in the Property Name field.

  
Image 5: Relationship properties

# TODO

* Repository support