



WhereGroup

---

# Geodatenmanagement und -harmonisierung mit GeoKettle



## Agenda

- Kurze Vorstellung der WhereGroup
- Was ist GeoKettle
- Beschreibung des Anwendungsbeispiels
- GeoKettle in action
- Ergebnis
- Fazit

Dienstleister in den Bereichen WebGIS, GDI, Kataster, Datenbanken mit Freier Software

rund 20 Mitarbeiter  
(Geographen, Informatiker, Vermessungsingenieure, Softwareentwickler)

Sitz in Bonn, Niederlassung in Berlin

Schulungen, Infoveranstaltungen, Konferenzen

Eigenes Schulungsinstitut: FOSS Academy

eigene Open Source-Projekte: Mapbender, PostNAS



**Mapbender**



**PostNAS**  
ALKIS ® mit freier Software



WhereGroup

## ***Vorstellung GeoKettle***



- GeoKettle ist eine ETL-Programm für räumliche Daten
  - E => Extract
  - T => Transform
  - L => Load





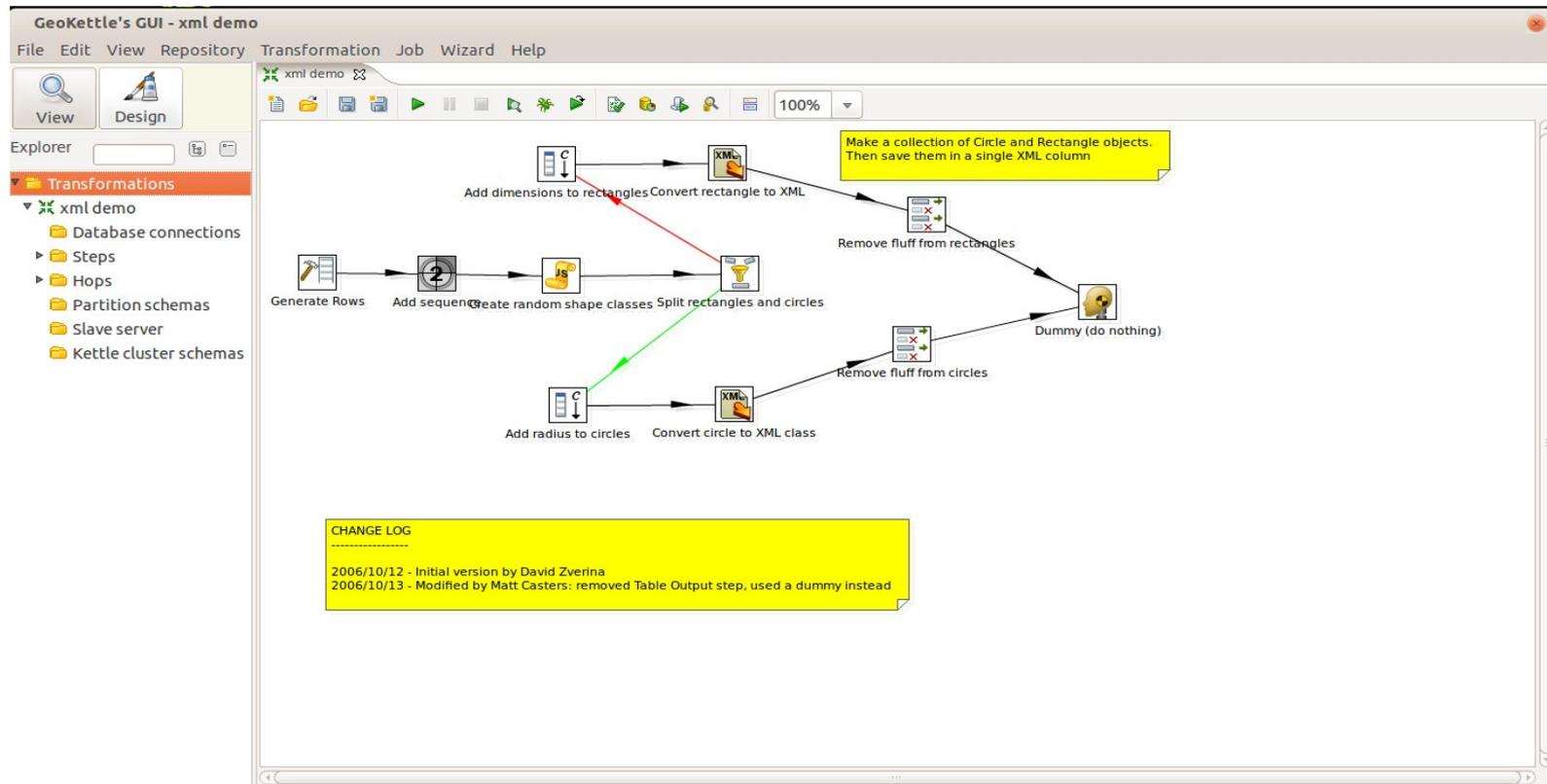
- GeoKettle basiert auf der OpenSource-Software Pentaho Data Integration (Kettle)
- GeoKettle ist OpenSource (Lizenz LGPL)
- GeoKettle unterstützt viele OpenSource Bibliotheken
  - JTS
  - GeoTools
  - Degree
  - gdal/ogr
  - sextante



- Es existieren mehrere Möglichkeiten mit GeoKettle zu arbeiten:
  - Spoon
  - Pan
  - Kitchen
  - Carte



- Graphische Oberfläche, um Transformationen und Jobs zu erstellen.
- Die hier angelegten Transformationen und Jobs werden entweder als xml (.ktr) oder in einer Datenbank (Repository) abgespeichert.

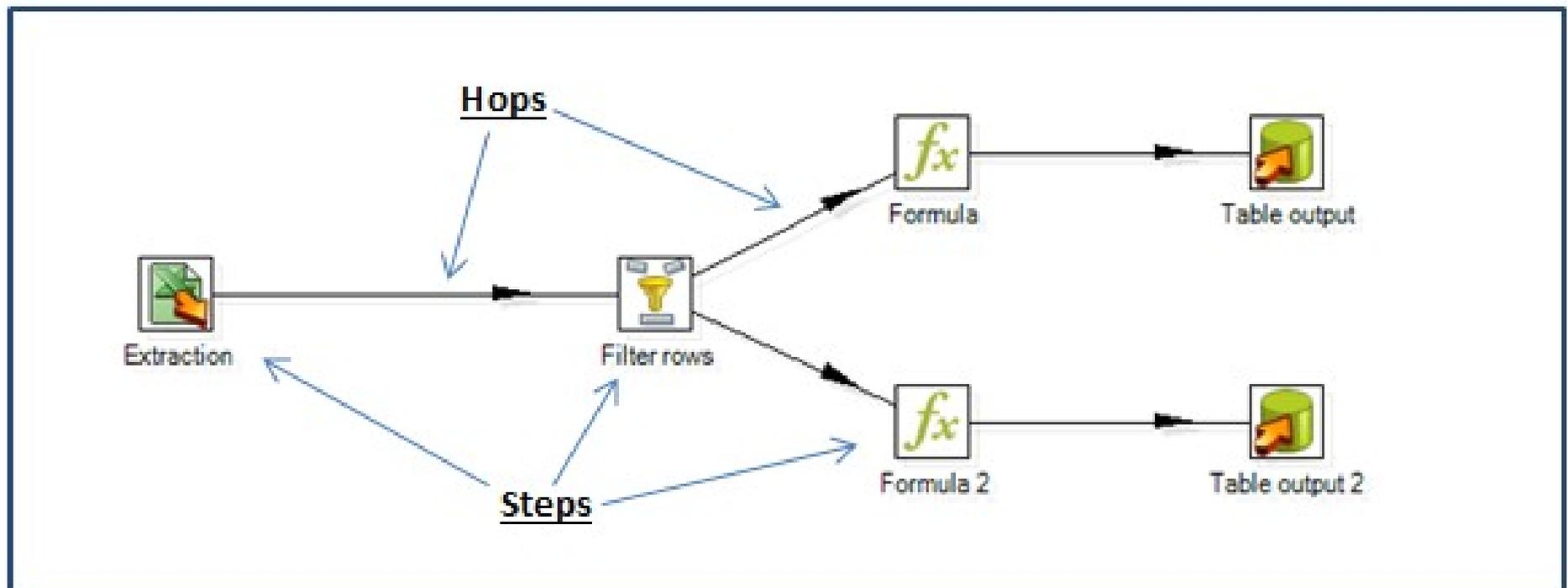




- Pan ist ein Programm, das Transformationen, die in spoon erstellt wurden, ausführen kann.
- Kitchen ist ein Programm, das Jobs, die in spoon erstellt wurden ausführen kann.
- Beide arbeiten mit der Kommandozeile und werden unter Windows über mit pan.bat/kitchen.bat und unter Linux und OSX mit pan.sh/kitchen.sh aufgerufen.

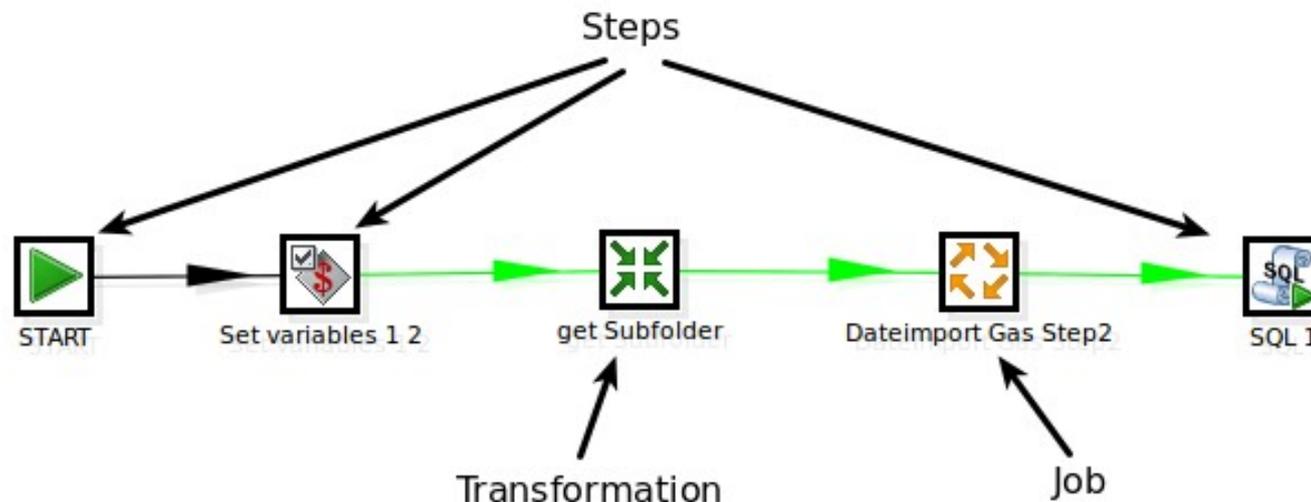


- Eine Transformation besteht aus einem Satz an Operationen, die mit den Daten vollzogen werden sollen.
- Sie bestehen aus mehreren **Steps**, die durch **Hops** verbunden werden.
- In den Transformationen wird mit den Daten „gearbeitet“.





- Jobs sind den Transformationen „übergeordnet“.
- Sie kontrollieren den Ablauf von Transformationen und/oder anderer Jobs.
- Die Schritte in den Jobs werden immer nacheinander abgearbeitet (Innerhalb einer Transformationen kann auch parallel gearbeitet werden).





WhereGroup

## *Beschreibung des Anwendungsbeispiel*



- Die Werraenergie muss ein neues CAD anschaffen.
- Die Daten wurden im alten System in einer Oracle-Datenbank gespeichert und sollen in eine PostgreSQL-Datenbank überführt werden.
- Ein Datenbankmodell ist nicht dokumentiert und quasi nicht vorhanden.
- Die Daten können nur mit Hilfe eines CAD als Shapefiles exportiert werden.



- Es existieren nach dem Export **247** verschiedene Shapefiles.
- Das Versorgungsgebiet der Werraenergie wurde in 40 sich teilweise räumlich überlagernde Gebiete aufgeteilt.
- Die Shapefiles können sowohl Objekte als auch Hilfsgeometrien oder Kartentextinformationen (mit jeweils identischen Ids und Attributdaten) enthalten.



- Die 247 verschiedene Shapefiles sollen in 2 verschiedene Datenbanken importiert werden
  - Gas
  - Grundkarte.
- Es sollen keine redundanten Daten vorhanden sein.
- Objekte und Beschriftungen sind zu trennen bzw. in den Attributen der Objekte zu übernehmen.



WhereGroup

***GeoKettle in action***



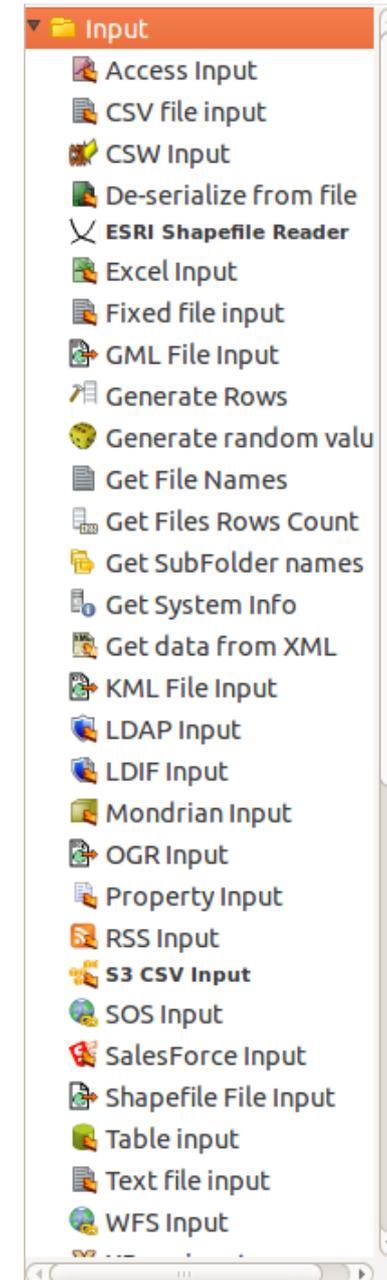
- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden.
- Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden.
- Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden.
- Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.).
- Entfernen von Duplikaten.



- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden
- Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden
- Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden
- Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.)
- Entfernen von Duplikaten



- Es gibt eine Vielzahl von Input-Datenformaten u.a.:
  - Über 35 Datenbanken (u.a. PostgreSQL, Oracle, MySQL, ...)
  - Shapefiles, GML, KML, OGR
  - OGC Web services
  - XML
  - MS Access
  - LDAP
  - Excel, CSV, txt
  - Etc.

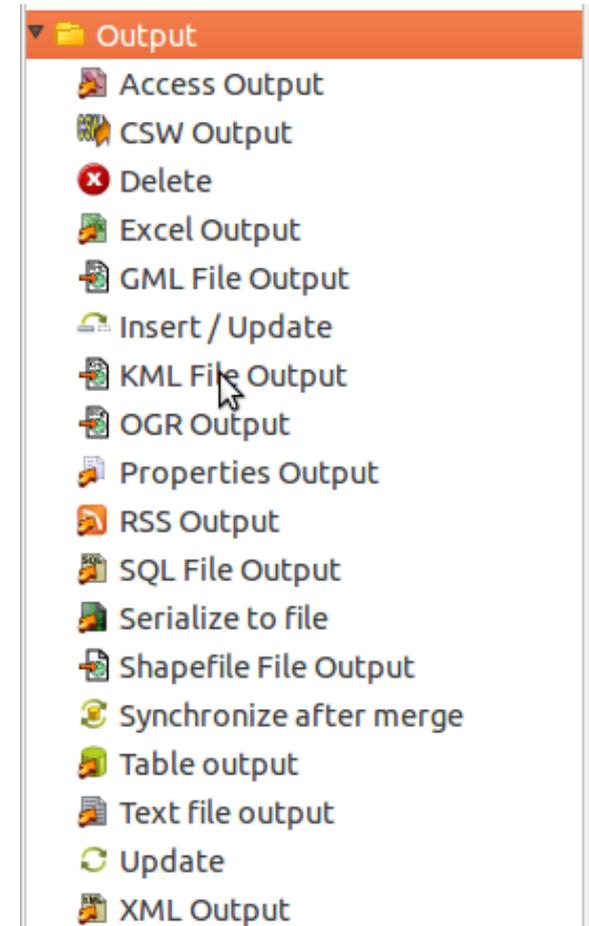




- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden
- Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden
- Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden
- Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.)
- Entfernen von Duplikaten



- Es gibt eine Vielzahl an Outputformate, z. B.
  - Räumliche Datenbanken
  - Datenformate
    - shp
    - GML
    - KML
    - OGR
  - uvm.



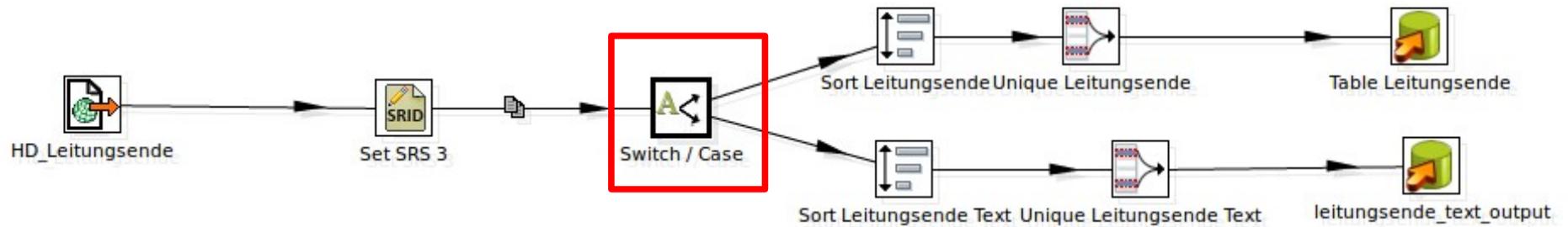


- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden





- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden
- **Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden**
- Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden
- Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.)
- Entfernen von Duplikaten



Switch / case

Step name: Switch / Case

Field name to switch: STIL

Use string contains comparison:

Case value data type: String

Case value conversion mask:

Case value decimal symbol:

Case value grouping symbol:

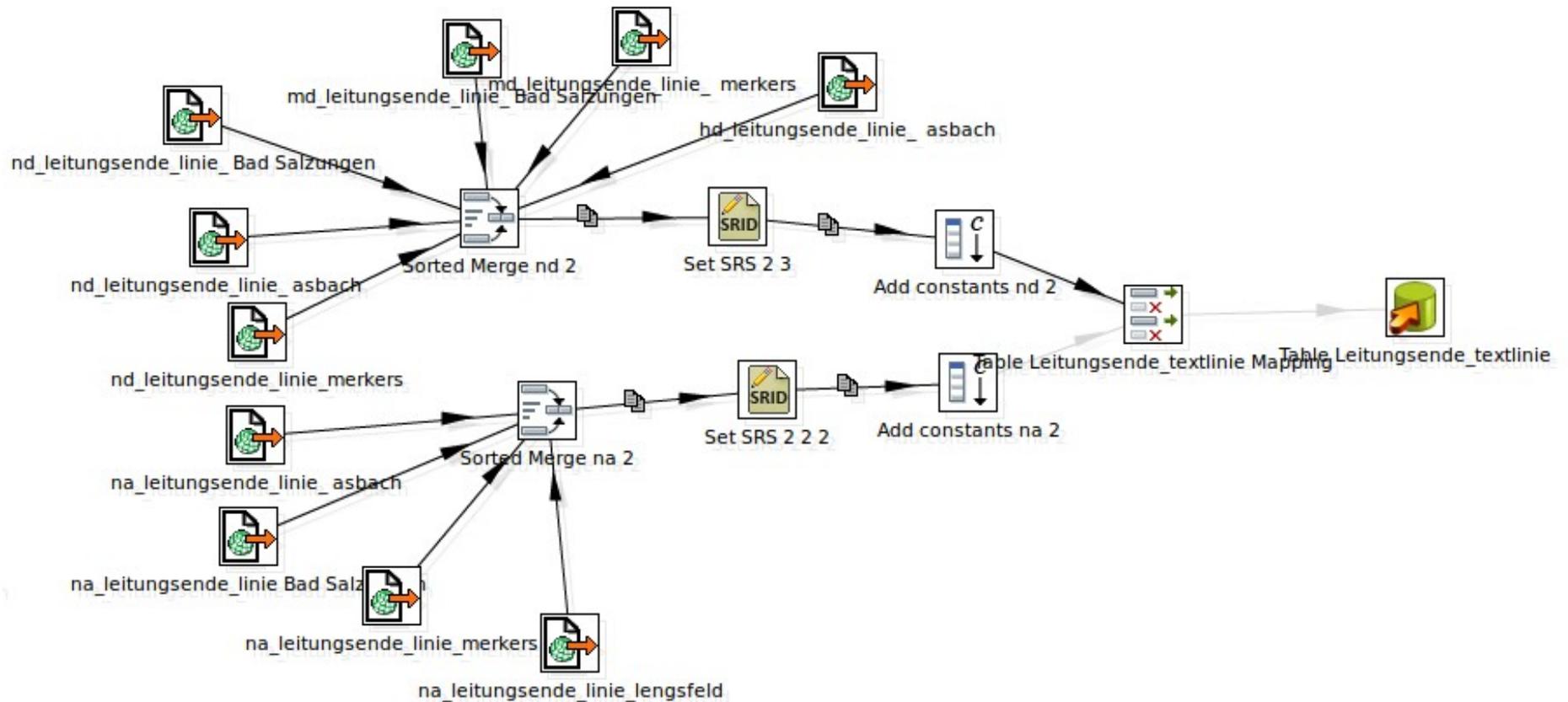
| # | Value                   | Target step            |
|---|-------------------------|------------------------|
| 1 | Style-werragas_gas_symb | Sort Leitungsende      |
| 2 | Style-Umlaut gerade     | Sort Leitungsende Text |
| 3 | Style-DIN_2             | Sort Leitungsende Text |
| 4 |                         |                        |

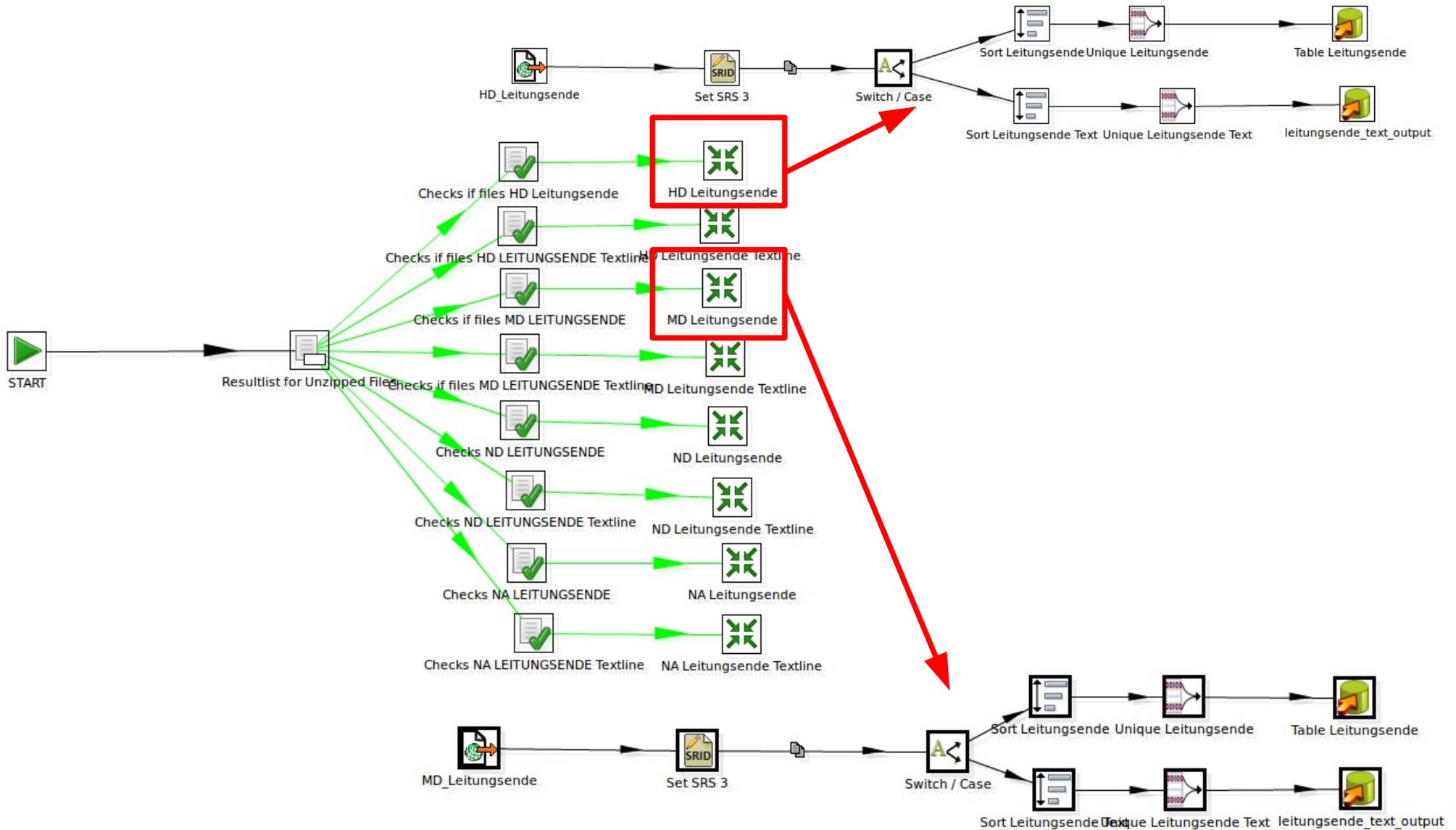
Default target step: Sort Leitungsende Text

OK Cancel



- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden
- Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden
- **Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden**
- Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.)
- Entfernen von Duplikaten



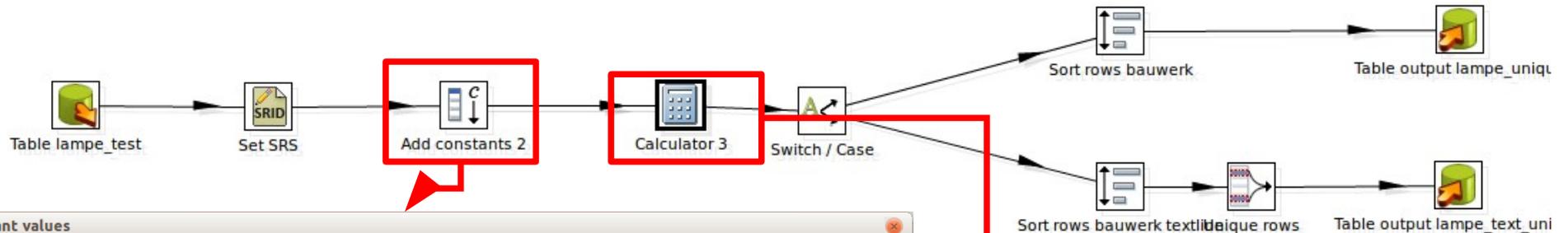




- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden
- Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden
- Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden
- **Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.)**
- Entfernen von Duplikaten



# Beispiel: Umrechnung von Rad nach Grad



**Add constant values**

Step name:

Fields:

| # | Name | Type   | Format | Length | Precision | Currency | Decimal | Group | Value |
|---|------|--------|--------|--------|-----------|----------|---------|-------|-------|
| 1 | grad | Number |        | 0      | 0         |          | .       |       | 57.3  |

**Calculator**

Step name:

Fields:

| # | New field | Calculation | Field A | Field B | Field C | Field D | Field E | Value type | Length | Precision | Remove | Conversion m |
|---|-----------|-------------|---------|---------|---------|---------|---------|------------|--------|-----------|--------|--------------|
| 1 | zwischen  | A * B       | drehung | grad    |         |         |         | Number     |        |           | N      |              |

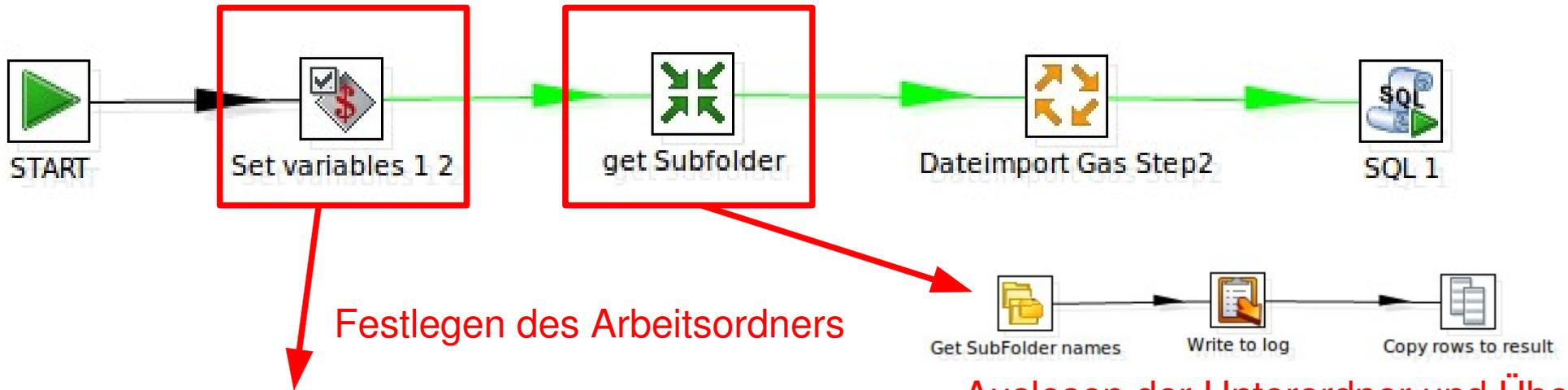
OK Cancel



- Die 247 verschiedenen Shapefiles müssen aus den 40 Ordnern in 2 verschiedene Datenbanken importiert werden
- Es müssen Objekte aus einem Shapefiles auf mehrere Tabellen verteilt werden
- Es müssen mehrere Shapefiles mit unterschiedlichen Attributen in einer Tabelle zusammengefasst werden
- Es müssen Attribute verändert werden (Feldinhalte, Feldformate etc.)
- Entfernen von Duplikaten



1. Erstellen der 247 Einzeltransformationen für die Shapefiles
2. Die 40 Ordner zu Gruppen von je 6 Ordnern zusammenfügen.
3. Erstellen von Jobs und Transformationen, um das automatisierte Auslesen der Ordner und das Anstoßen der einzelnen Transformationen zu steuern



Festlegen des Arbeitsordners

Auslesen der Unterordner und Übergabe von Attributen an nächsten Job

Set variables...

Job entry name: Set variables 1 2

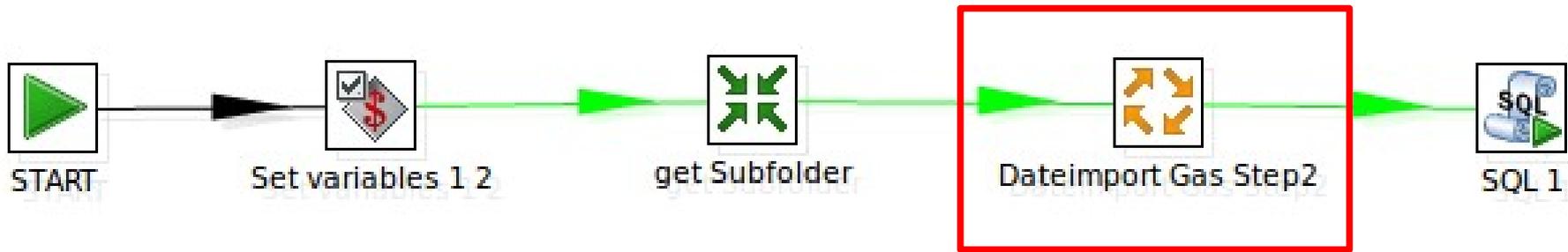
Settings

Variable substitution?

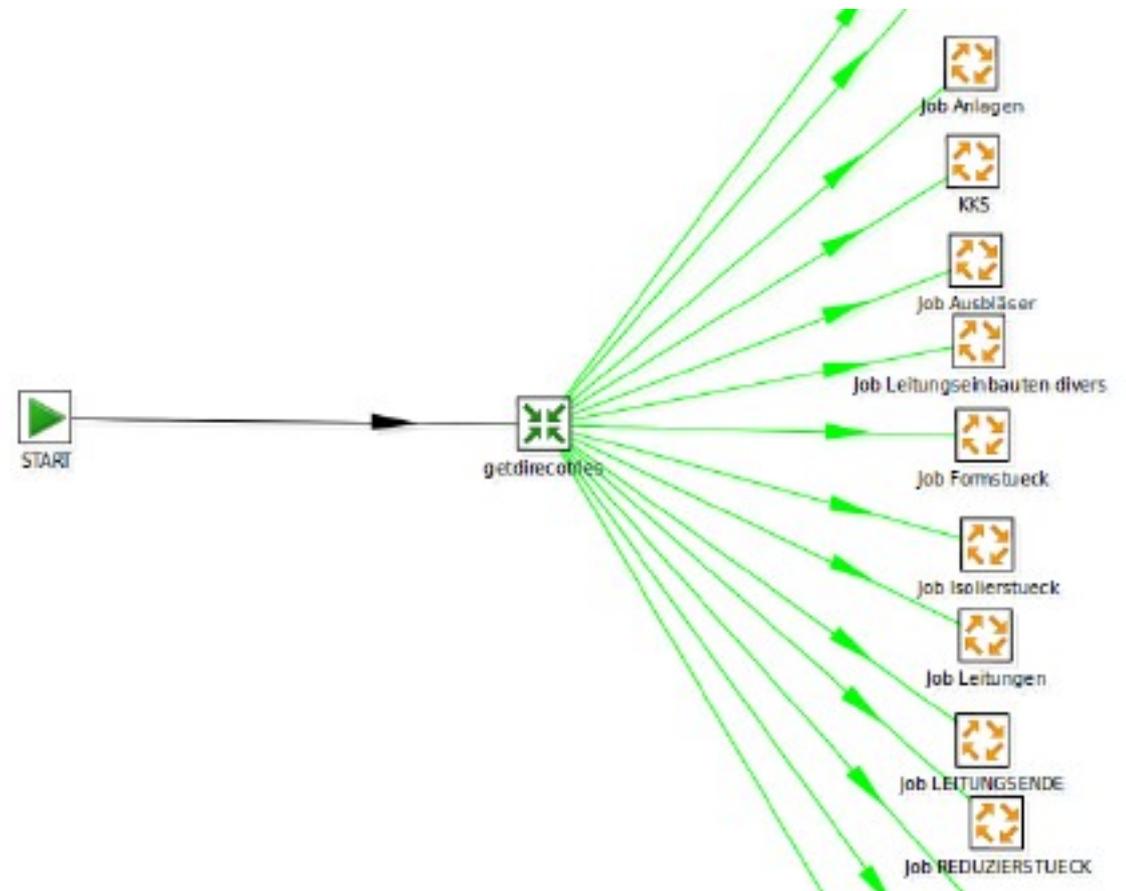
Variables:

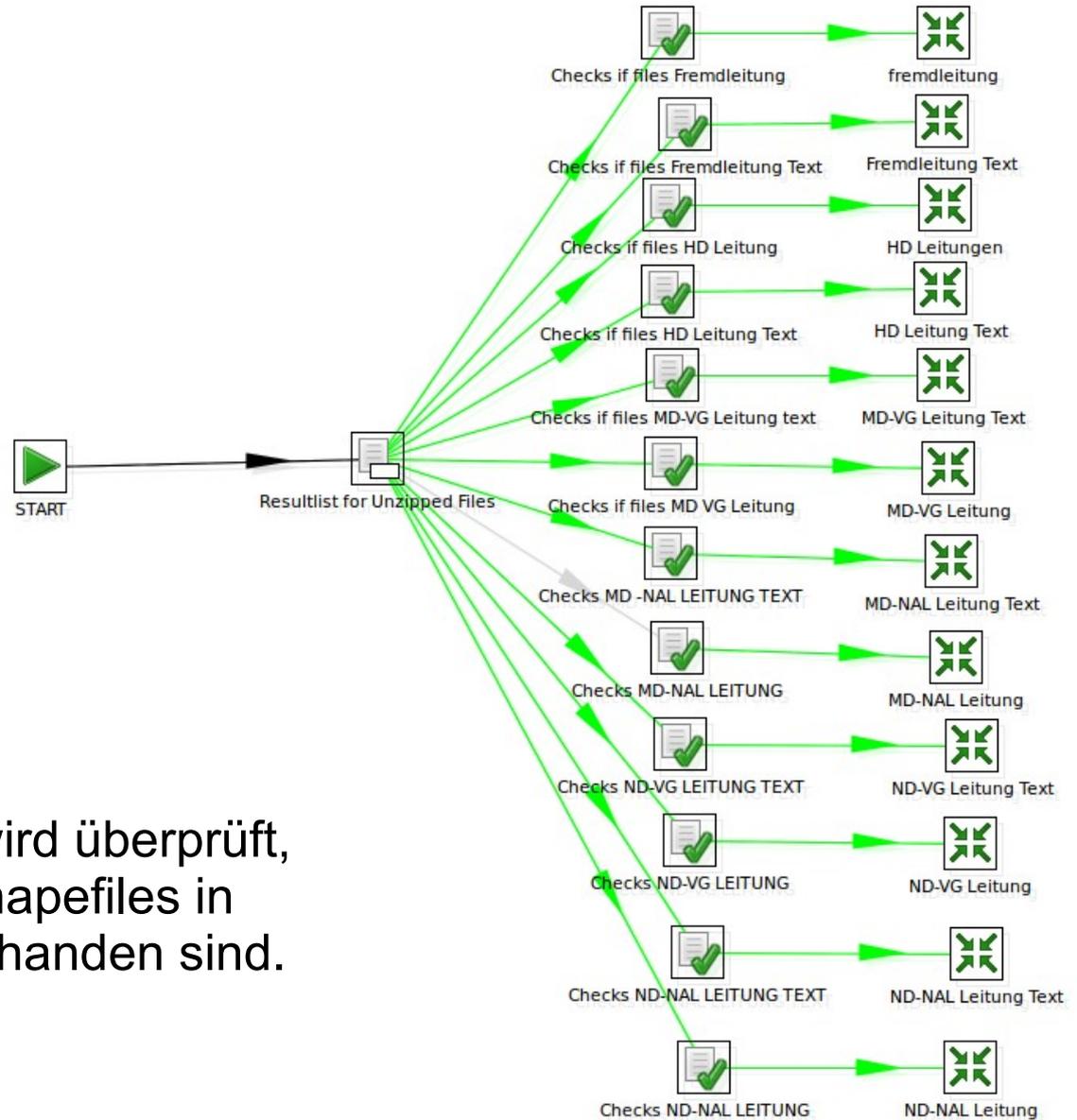
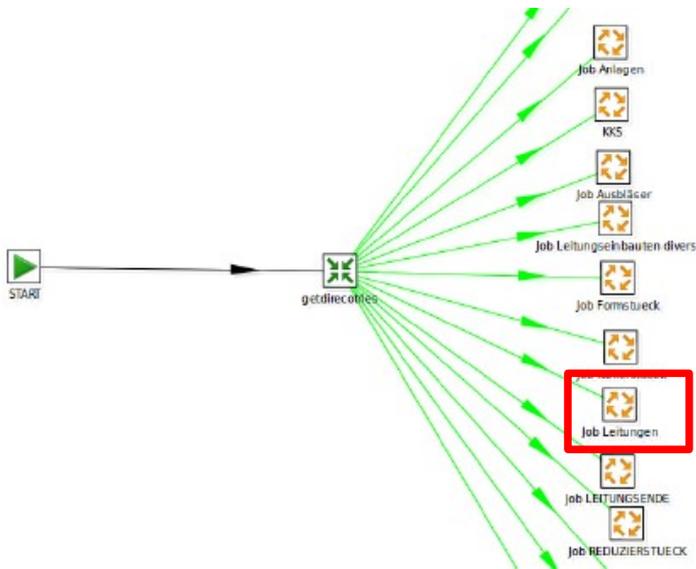
| # | Variable name | Value   | Variable scope type               |
|---|---------------|---|-----------------------------------|
| 1 | root.path     | /home/schaeff/center/wheregroup/Projects/Werraenergie/8_Daten/SHP/produktivdaten/ | Valid in the Java Virtual Machine |

OK Cancel

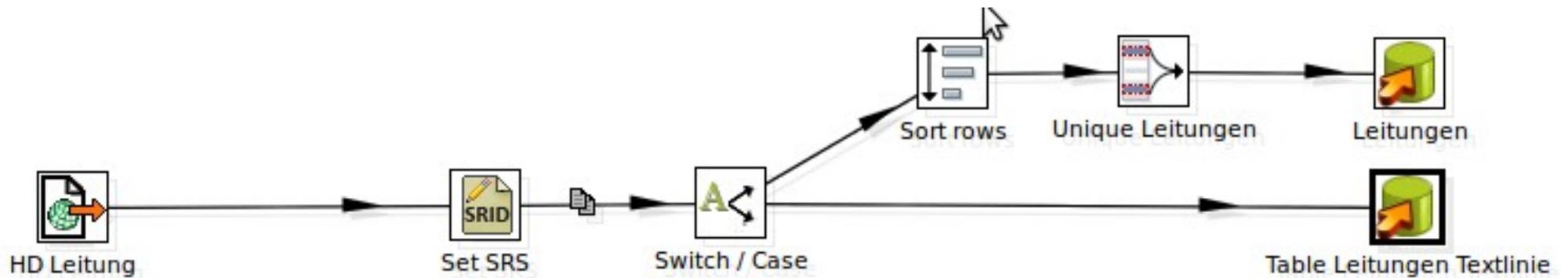
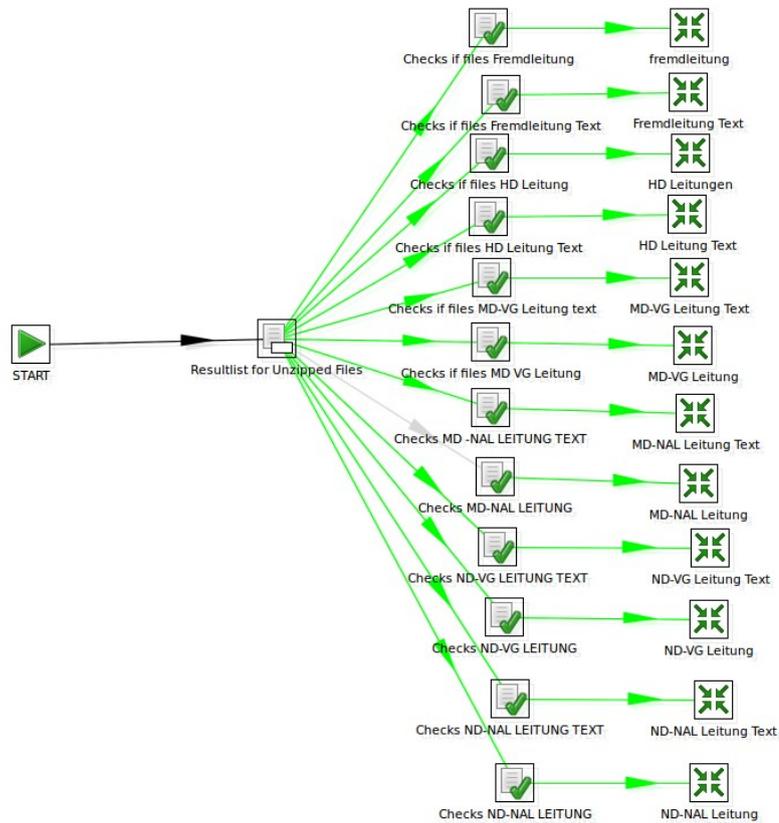


Unterordner werden geöffnet und es werden nacheinander weitere Jobs angestoßen zum Beispiel Job „Leitungen“





In diesen Jobs wird überprüft, ob bestimmte Shapefiles in den Ordnern vorhanden sind.





WhereGroup

***Ergebnis***



- Nach dem Import der Daten in die PostgreSQL befinden sich in
  - der Datenbank „Gas“ 76 Tabellen
  - der Datenbank „Grundkarte“ 74 Tabellen
- Die Modellierung des Datenimports dauerte ca 2 Tage.
- Der Importprozess dauerte ca 3 Stunden



WhereGroup

***Fazit***



- Ohne den Einsatz von GeoKettle wäre dieser Datenimport in dieser Form und in diesem Zeitraum nicht durchführbar gewesen.
- GeoKettle ist eine gut dokumentierte und schnell zu erlernende Open Source Software.
- **GeoKettle ist eine lohnende Alternative zur FME!**

**Save the Date:**

**„Einf. In GeoKettle“ am 19.11.2013 in Bonn**



WhereGroup

**Vielen Dank für Ihre Aufmerksamkeit ...**

**Fragen?**

WhereGroup GmbH & Co. KG  
Eifelstraße 7 | 53121 Bonn

Tel.: +49 (0)228 909038-0  
Fax: +49 (0)228 909038-11

info@wherergroup.com  
<http://www.wherergroup.com>

