

# Integrated Data Repository Toolkit (IDRT)

TMF-Projekt V091-MI\_03

## IDRT Import-Tool

- Anleitung -

Benjamin Baum  
Christian Bauer  
Prof. Dr. Ulrich Sax

## Autoren

Autor 1: Benjamin Baum  
Universitätsmedizin Göttingen, Abteilung Medizinische Informatik  
Georg-August-Universität Göttingen  
Robert-Koch-Straße 40  
37075 Göttingen  
Tel.: +49 551 39 22996  
Fax: +49 551 39 22493  
E-Mail: benjamin.baum@med.uni-goettingen.de

Autor 2: Christian Bauer  
Universitätsmedizin Göttingen, Abteilung Medizinische Informatik  
Georg-August-Universität Göttingen  
Robert-Koch-Straße 40  
37075 Göttingen  
Tel.: +49 551 39 8227  
Fax: +49 551 39 22493  
E-Mail: christian.bauer@med.uni-goettingen.de

Autor 3: Prof. Dr. Ulrich Sax  
Universitätsmedizin Göttingen, Abteilung Medizinische Informatik  
Georg-August-Universität Göttingen  
Robert-Koch-Straße 40  
37075 Göttingen  
Tel.: +49 551 39 13148  
Fax: +49 551 39 22493  
E-Mail: Ulrich.Sax@med.uni-goettingen.de

## **Inhaltsverzeichnis**

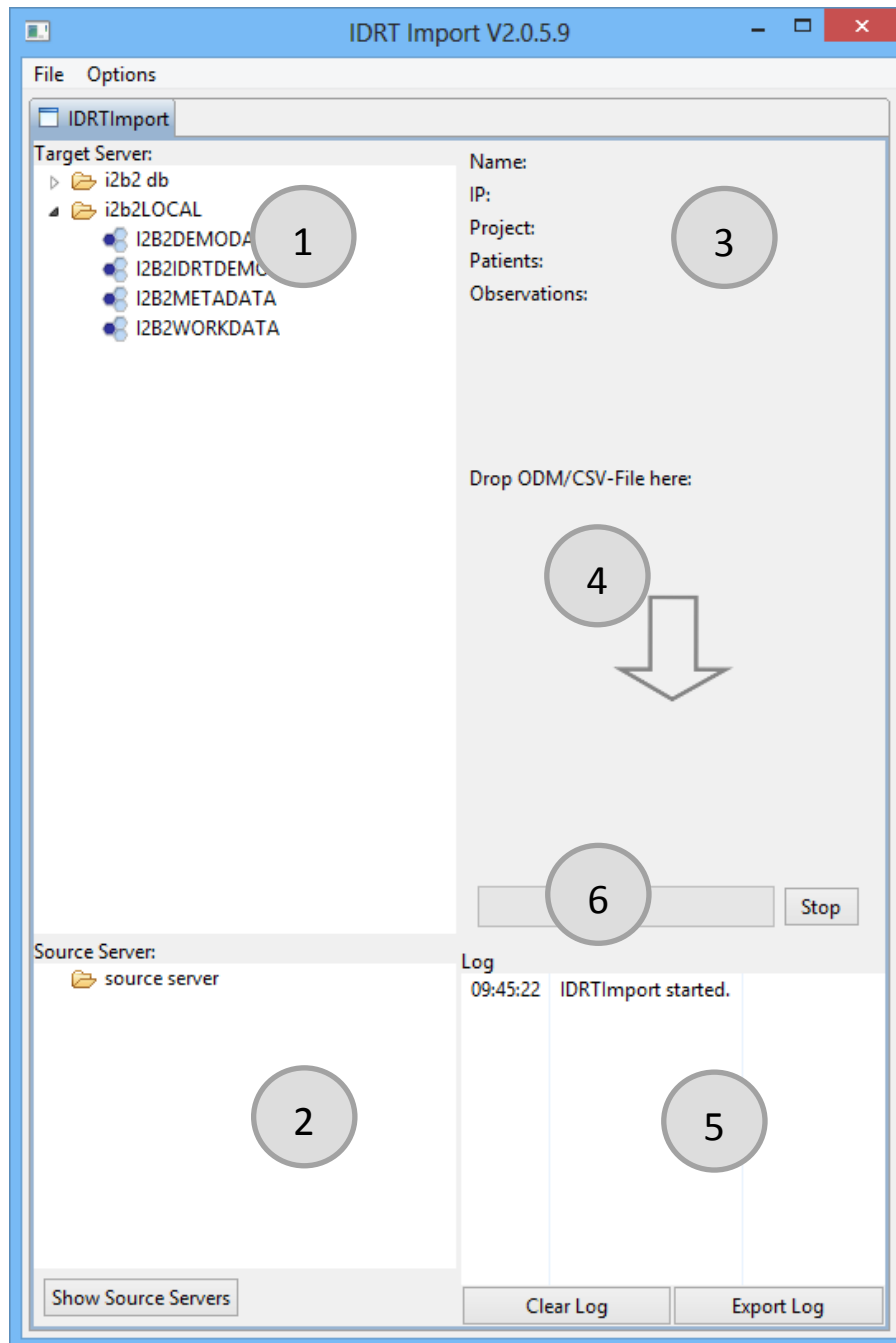
<b>Inhaltsverzeichnis.....</b>	<b>3</b>
<b>Einleitung.....</b>	<b>4</b>
<b>Überblick.....</b>	<b>5</b>
<b>ODM-Import .....</b>	<b>7</b>
<b>CSV-Import.....</b>	<b>8</b>
Vorbereitungen .....	8
Import.....	9
<b>Datenbank Import.....</b>	<b>11</b>
<b>PID-Generator Anbindung .....</b>	<b>13</b>
<b>Talend Open Studio .....</b>	<b>15</b>
<b>TOS-Job: CSV-Import.....</b>	<b>15</b>
Vor- und Nachbereitung.....	15
Extraktion .....	16
Transformation.....	18
Load .....	18
<b>TOS-Job: DB-Import .....</b>	<b>19</b>
DB Konfigurationsdateien .....	19
<b>TOS-Job: ODM-Import.....</b>	<b>20</b>
Extraktion - Ontologie .....	21
Extraktion – Patientendaten .....	22
Transformation.....	23
Load.....	23
<b>Fehlerbehandlung .....</b>	<b>25</b>

## ***Einleitung***

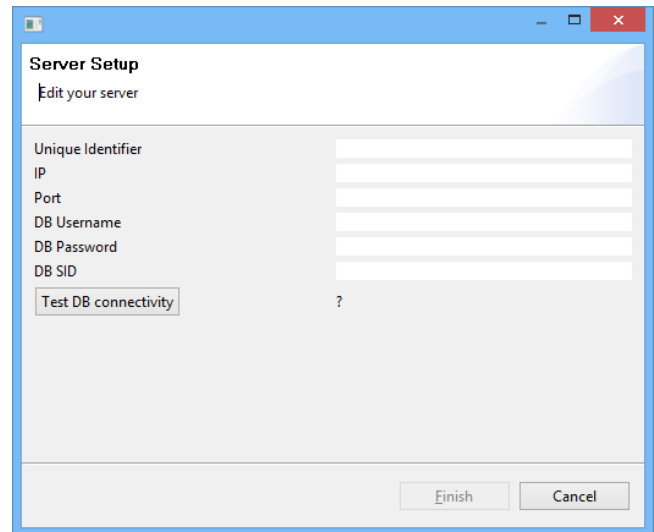
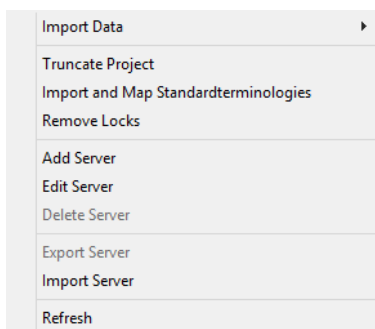
In dieser Anleitung wird die Benutzung des IDRT-Import-Tools erläutert. Das IDRT-Import-Tool ist ein Java RCP-Programm, das die TOS (Talend Open Studio)-Jobs aus den Arbeitspaketen AP2.3 Modul Extraktion von ODM-Datenquellen und AP2.4 Modul Extraktion für tabellarische Datenquellen ausführt. Die beiden Jobs extrahieren Patienten- und Strukturdaten aus den vorgegebenen Formaten, transformieren diese Daten in die i2b2-Struktur und laden diese anschließend in ein angegebenes i2b2-Projekt. Dabei bietet es die Möglichkeit des Imports von Standardterminologien aus dem Arbeitspaket AP4.1 Standardterminologien.

## Überblick

Das Hauptfenster des IDRT-Import-Tools ist in fünf Abschnitte unterteilt. Jeder dieser Abschnitte hat eine eigene Aufgabe. Alle Abschnitte werden nun erläutert.



- (1) Target Server: Hier werden alle i2b2 Server aufgelistet, die der Nutzer eingetragen hat. Über das Kontext-Menü können Server hinzugefügt, gelöscht, editiert oder exportiert bzw. Jeder Server braucht einen einzigartigen Name, IP-Adresse, Port, Datenbank-Nutzernamen inklusive Passwort und den Bezeichner der Datenbank (SID). Nach dem Eintragen der Daten kann die Verbindung getestet werden und der Nutzer wird bei einer fehlerhaften Eingabe darauf aufmerksam gemacht.

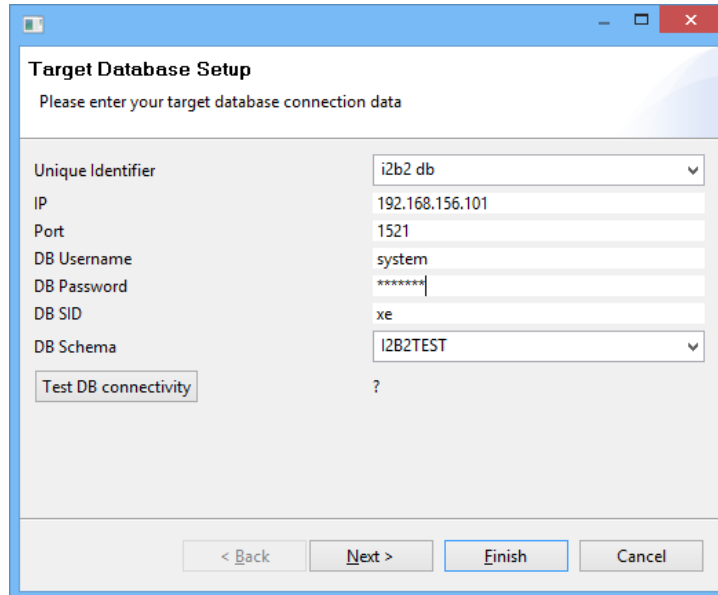
Mit einem Rechtsklick auf ein i2b2 Projekt kann der Nutzer im Kontext-Menü einen Import für CSV, ODM, Datenbank oder §21 starten. Des Weiteren lässt sich das Projekt leeren und die Standardterminologien können importiert werden. Sollte ein i2b2-Benutzer zum Zeitpunkt des Imports angemeldet sein, sperrt i2b2 für den Import wichtige Tabellen, die mit dem Menüpunkt Remove Locks entsperrt werden können. Hier werden alle vorhandenen Locks auf die Datenbank gelöst und ein Datenbank-Administrator Account (system/sys) muss dabei in den Servereinstellungen verwendet werden.

- (2) Source Server: Hier können Server gespeichert werden, die als Quell-Server dienen sollen, um Importe aus fremden Datenbanken und Tabellen zu ermöglichen. Hier können Server ähnlich dem Target Server Abschnitt hinzugefügt, editiert, gelöscht, exportiert bzw. importiert werden.
- (3) Dieses Fenster gibt eine kurze Information über den aktuell ausgewählten Server, das Projekt und die IP-Adresse.
- (4) In diesen Abschnitt kann der Nutzer CSV- oder ODM-Dateien mittels Drag and Drop ablegen, was den jeweiligen Import automatisch startet.
- (5) Log: Hier werden Status-Meldungen ausgegeben. Fehler werden rot markiert.
- (6) Status-Leiste

## ODM-Import

Wählt man im Kontext-Menü den Punkt ODM-Import aus, öffnet sich zunächst ein Fenster, in dem man das Ziel i2b2-Projekt auswählen kann. Hierbei werden die Server aus dem Target Server Abschnitt verwendet.

IP der Datenbank  
Port der Datenbank  
Benutzername für den DB-Login  
Passwort für den DB-Login  
Oracle System Identifier  
Oracle Nutzer des i2b2-Projektes  
Testet die Konfiguration



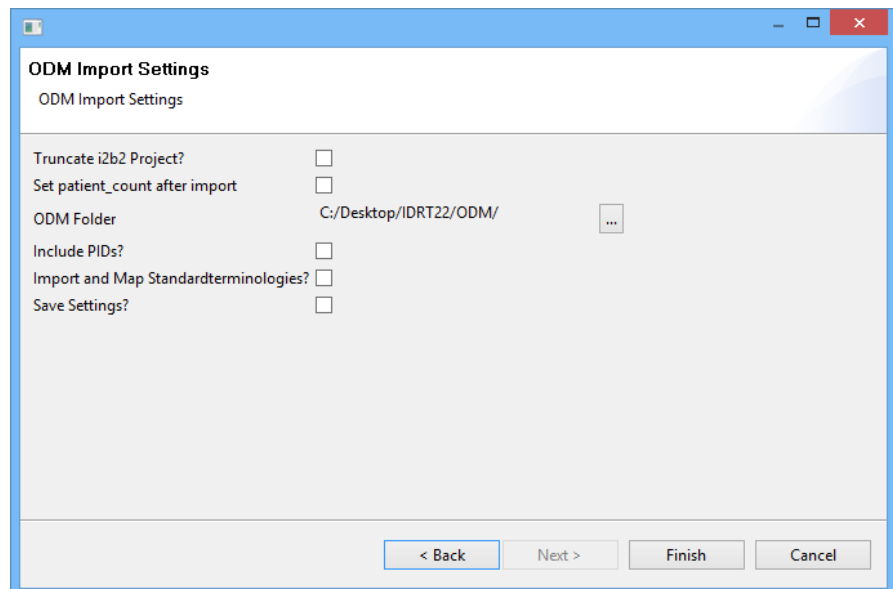
**Target Database Setup**  
Please enter your target database connection data

Unique Identifier	i2b2 db
IP	192.168.156.101
Port	1521
DB Username	system
DB Password	*****
DB SID	xe
DB Schema	I2B2TEST
Test DB connectivity	?

< Back   Next >   Finish   Cancel

Auf der nächsten Seite werden nun die ODM-Spezifischen Optionen angegeben.

Leert das i2b2-Projekt  
Setzt die patient\_count Spalte  
ODM-Dateien  
Patienten IDs als Item mit importieren  
Standardterminologien importieren  
Einstellungen speichern

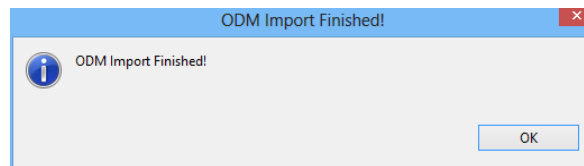


**ODM Import Settings**  
ODM Import Settings

Truncate i2b2 Project?	<input type="checkbox"/>
Set patient_count after import	<input type="checkbox"/>
ODM Folder	C:/Desktop/IDRT22/ODM/ ...
Include PIDs?	<input type="checkbox"/>
Import and Map Standardterminologies?	<input type="checkbox"/>
Save Settings?	<input type="checkbox"/>

< Back   Next >   Finish   Cancel

Bei einem Klick auf **Finish** wird der Importvorgang gestartet. Der Importvorgang lässt sich durch einen Klick auf Knopf „Stop“ im Hauptfenster abbrechen. Nach dem Import-Vorgang bekommt der Nutzer eine Meldung, ob der Import erfolgreich war, oder nicht.



## CSV-Import

### Vorbereitungen

Um CSV-Dateien in ein i2b2-Projekt zu importieren, können einige Vorarbeiten geleistet werden. In diesen Vorarbeiten wird zu jeder Datei, die medizinische Daten enthält, eine Config Datei erstellt, die Metadaten über die zu importierende Datei vorhält.

**Diese Config Datei kann von Hand, oder semi-automatisch im letzten Schritt des CSV-Imports erstellt werden.**

Die optionalen Vorarbeiten werden an einem Beispiel einer Studie mit fünf Patienten und mehreren Items illustriert.

PID	Date1	Date2	Date3	Source	Item1	Item2	Item3	Item4
1	01.01.01	02.01.01	03.01.01	Source1	1001	0,22	11	Value13
2	02.01.01	03.01.01	04.01.01	Source1	1002	1,22	12	Value23
3	03.01.01	04.01.01	05.01.01	Source1	1003	2,22	13	Value33
4	04.01.01	05.01.01	06.01.01	Source1	1004	3,22	14	Value43
5	05.01.01	06.01.01	07.01.01	Source1	1005	4,22	15	Value53

Für diese CSV-Datei muss nun vom Nutzer eine Config-Datei erstellt werden, die Metadaten zu dieser Studie enthält. Diese Config-Datei muss denselben Namen haben, wie die CSV-Datei der Studie, nur mit einem .cfg vor der Endung:

**Studie:** Beispielstudie.csv

**Config:** Beispielstudie.cfg.csv

Inhalt der Config Datei:

**Spaltenname:** Spaltennamen der Ausgangsdatei.

**Datentyp:** Der Datentyp der Spalte (Integer, Date, Float, String).

**Name:** Der Name des Items, so wie er später angezeigt werden soll.

**Metainformationen:** Festlegung der Metainformationen.

PatientID: Die Patientenidentifikation.

EncounterID: Fallnummer des Patienten.

ImportDate: Datum des ursprünglichen Imports in das Quell-System.

UpdateDate: Datum der letzten Änderung.

DownloadDate: Datum des Exports aus dem Quell-System.

Sourcesystem: Name des Quell-Systems.



StartDate: Start Datum des Faktes.

ignore: Spalte wird beim Import ignoriert.

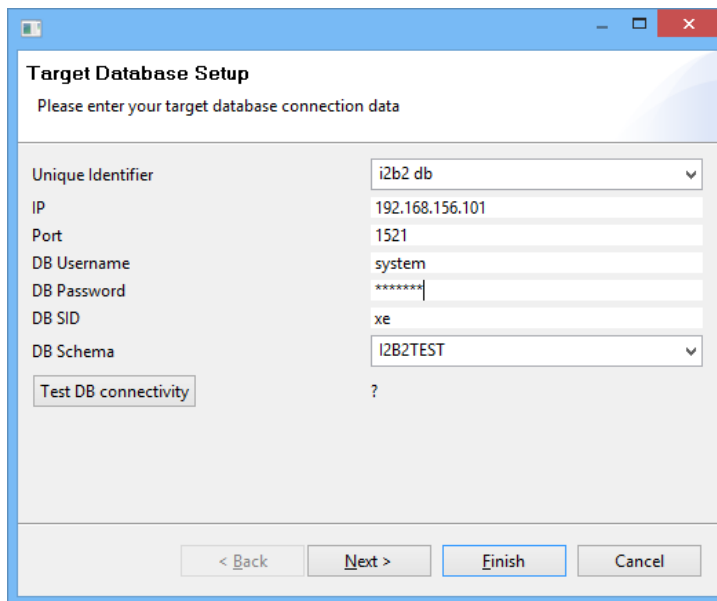
Die PatientID muss in jeder Zeile vorhanden sein, wenn die EncounterID der i2b2-Datenbank noch unbekannt ist, ansonsten reicht die EncounterID.

Spaltenname (Pflicht)	PID	Date1	Date2	Date3	Source	Item1	Item2	Item3	Item4
Datentyp (Pflicht)	Integer	Date	Date	Date	String	Integer	Float	Integer	String
Name (kann leer sein)	PID	UpdateDate	ImportDate	DownloadDate	Sourcesystem	EncounterID	Name1	Name1	Name3
Metainformationen (mind. PatientID oder EncounterID)	PatientID	UpdateDate	ImportDate	DownloadDate	Sourcesystem	EncounterID			

## Import

Wählt man Import CSV aus dem Kontext-Menü aus, öffnet sich wiederum das Fenster, aus dem man das i2b2-Projekt auswählen kann.

IP der Datenbank  
Port der Datenbank  
Benutzername für den DB-Login  
Passwort für den DB-Login  
Oracle System Identifier  
Oracle Nutzer des i2b2-Projektes  
Testet die Konfiguration



**Target Database Setup**  
Please enter your target database connection data

Unique Identifier	i2b2 db
IP	192.168.156.101
Port	1521
DB Username	system
DB Password	*****
DB SID	xe
DB Schema	I2B2TEST

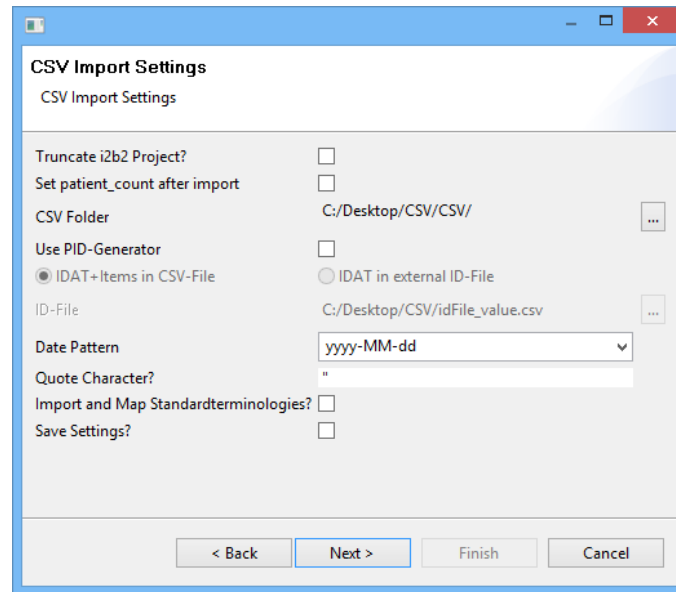
Test DB connectivity ?

< Back   Next >   Finish   Cancel

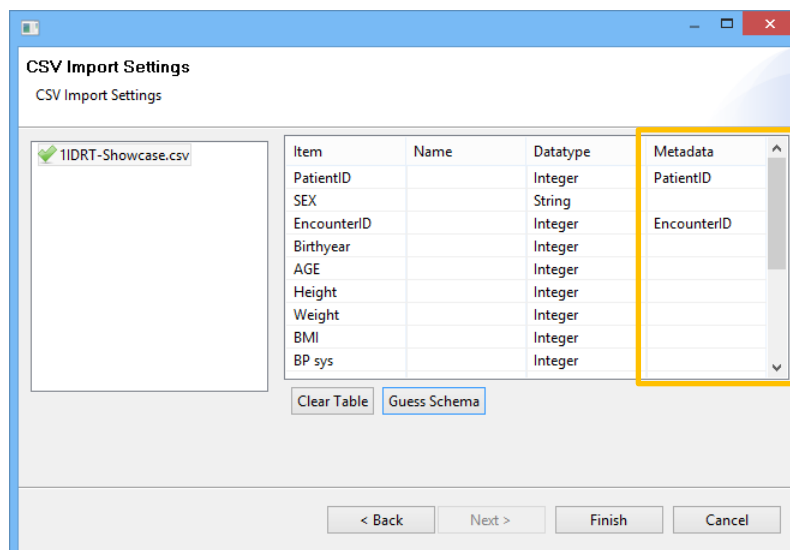
Auf der nächsten Seite werden nun die CSV-Spezifischen Optionen angegeben.

Leert das i2b2-Projekt  
Setzt die patient\_count Spalte  
Ordner mit den zu importierenden CSV-Dateien  
TMF-PIDGenerator (siehe eigenen Abschnitt)

Datei mit IDATs  
Datumsformat  
Textumschließendes Zeichen  
Standardterminologien importieren  
Einstellungen speichern



Bei einem Klick auf **Next** wird die letzte Seite des Imports angezeigt.

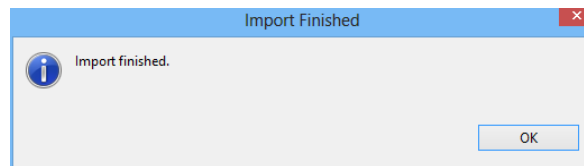


Item	Name	Datatype	Metadata
PatientID		Integer	PatientID
SEX		String	
EncounterID		Integer	EncounterID
Birthyear		Integer	
AGE		Integer	
Height		Integer	
Weight		Integer	
BMI		Integer	
BP sys		Integer	

Hier kann der Nutzer Klarnamen für die einzelnen Items vergeben (Spalte „Name“), den jeweiligen Datentyp ändern (Spalte „Datatype“) und ein Metadatum wie z.B. *PatientID*, oder *Import Date* vergeben.

Durch einen Klick auf **Guess Schema** wird versucht, die Spalten automatisch zu füllen. Der Knopf **Clear Table** leert die angezeigte Tabelle.

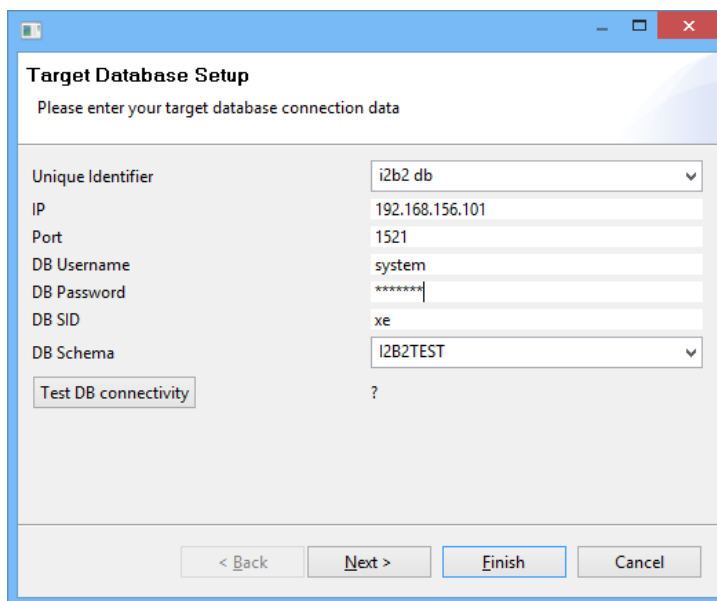
Bei einem Klick auf **Finish** wird der Importvorgang gestartet. Der Importvorgang lässt sich durch einen Klick auf Knopf „Stop“ im Hauptfenster abbrechen. Nach dem Import-Vorgang bekommt der Nutzer eine Meldung, ob der Import erfolgreich war, oder nicht.



## Datenbank Import

Wählt der Nutzer Import DB Table aus dem Kontext-Menü aus, öffnet sich genau wie bei den anderen Importern das erste Fenster, das Angaben zu dem Ziel i2b2-Projekt erfordert.

IP der Datenbank  
Port der Datenbank  
Benutzername für den DB-Login  
Passwort für den DB-Login  
Oracle System Identifier  
Oracle Nutzer des i2b2-Projektes  
Testet die Konfiguration

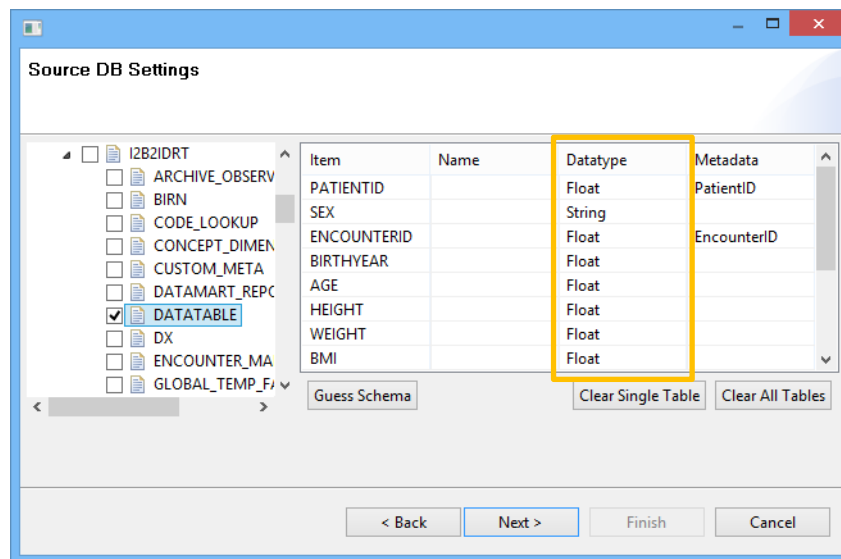


A dialog box titled "Target Database Setup" with a subtitle "Please enter your target database connection data". It contains several input fields and a "Test DB connectivity" button. The fields are:

- Unique Identifier: i2b2 db (dropdown)
- IP: 192.168.156.101
- Port: 1521
- DB Username: system
- DB Password: \*\*\*\*\*
- DB SID: xe
- DB Schema: i2B2TEST (dropdown)

At the bottom, there is a "Test DB connectivity" button with a question mark icon. Below the form are four buttons: "< Back", "Next >", "Finish", and "Cancel".

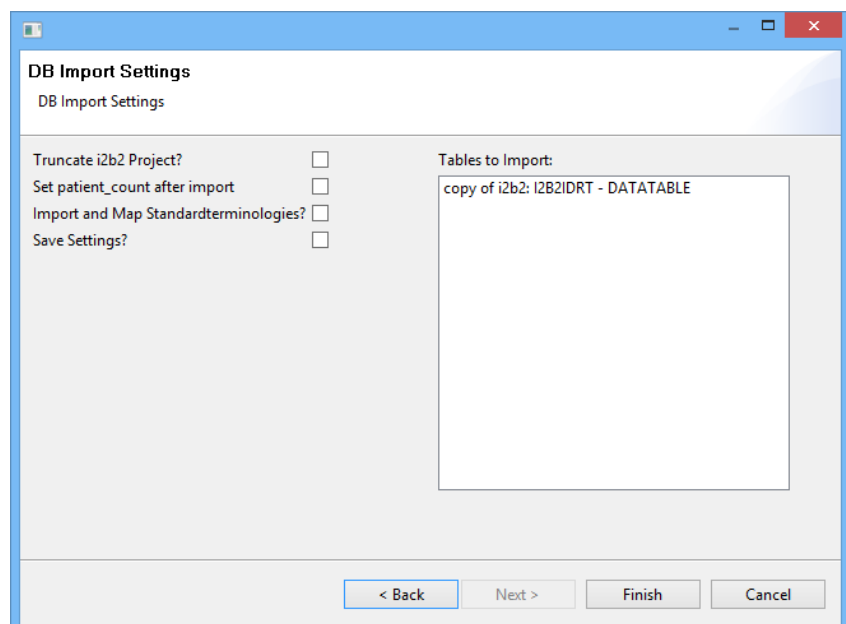
Auf der nächsten Seite ist eine Übersicht der eingetragenen Quell-Server zu sehen. Dort lassen sich Tabellen auswählen und bearbeiten, indem ähnlich dem CSV-Import, Metadaten zu den einzelnen Spalten angegeben werden.



Die Tabellen werden automatisch lokal gespeichert, wodurch ein erneuter Import der Daten vereinfacht wird.

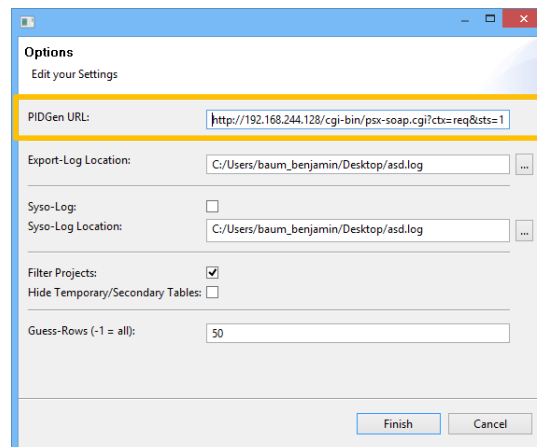
Der Knopf **Clear Single Table** leert die angezeigte Tabelle, wohingegen **Clear All Tables** alle gespeicherten Tabellen entfernt. Wird mindestens eine Tabelle für den Import markiert bekommt der Nutzer auf der nächsten Seite eine Übersicht über die zu importierenden Tabellen und kann das i2b2-Projekt leeren, Standardterminologien importieren und einen Ordner für das Zwischenspeichern der Tabellen festlegen.

Leert das i2b2-Projekt  
Setzt die patient\_count Spalte  
Standardterminologien importieren  
Einstellungen speichern

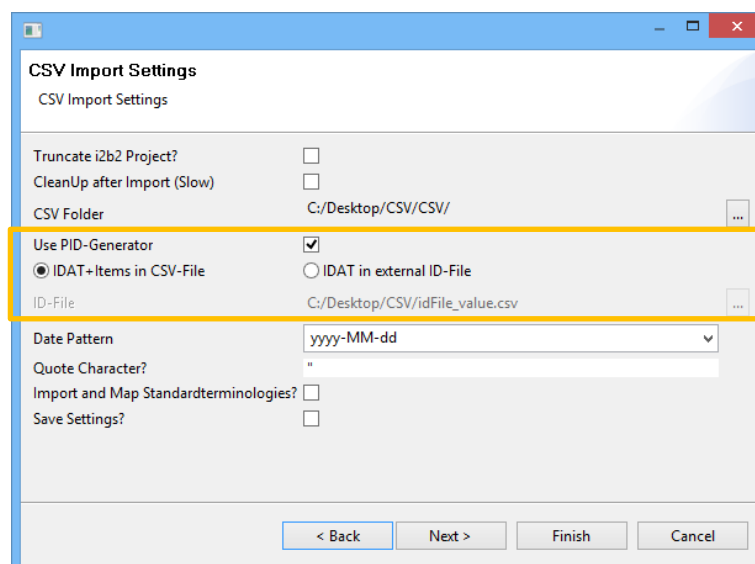


## PID-Generator Anbindung

Um den PID-Generator der TMF verwenden zu können, muss als Erstes die entsprechende URL des PID-Generators im Options-Menü des IDRTImport-Tools eingetragen werden.



Für die Pseudonymisierung der Daten können zwei unterschiedliche Szenarien ausgewählt werden.



### a) IDAT+Items in einer CSV-Datei

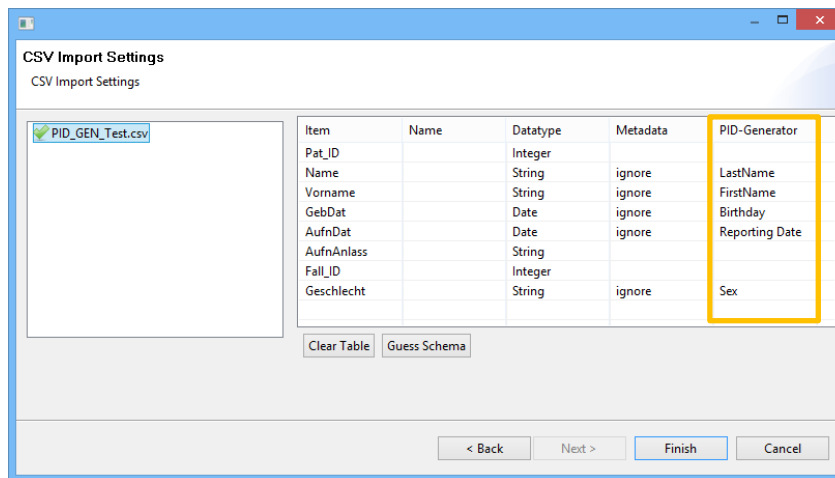
- Die identifizierende Daten der Patienten sind hier zusammen mit den zu importierenden Daten in einer einzelnen CSV-Datei.

Beispiel Ausgangsdatei:

LastName	FirstName	Birthday	Reporting Date	SEX	Encounter ID	Birthyear	AGE	Height	Weight
Müller	Timo	1923-03-12	2011-08-10	f	1000	1971	41	209	50
Berg	Moritz	1911-04-03	2004-11-24	m	1001	1964	48	148	50
Bali	Rolf	1909-05-04	2007-04-13	m	1002	1933	79	148	93
Klein	Moritz	1910-02-02	2007-04-13	m	1003	1963	49	134	75

Die Konfiguration des PID-Generators wird dann auf der nächsten Seite eingestellt. Im Vergleich zum normalen CSV-Import wird eine neue Spalte (PID-Generator) angezeigt, in der

die notwendigen PID-Generator-Felder eingetragen werden müssen. Die Reihenfolge der Spalten in der Ausgangsdatei ist beliebig; das Datumsformat muss unter „Date Pattern“ angegeben werden.



Item	Name	Datatype	Metadata	PID-Generator
Pat_ID		Integer		
Name		String	ignore	LastName
Vorname		String	ignore	FirstName
GebDat		Date	ignore	Birthday
AufnDat		Date	ignore	Reporting Date
AufnAnlass		String		
Fall_ID		Integer		
Geschlecht		String	ignore	Sex

#### b) IDAT in externer ID-File

- Die identifizierenden Daten sind in einer externen CSV-Datei.
- Die externe ID-File und die zu importierende CSV-Datei müssen hier über einen eigenen Identifier verknüpft sein (siehe Beispiel). Dieses externe ID-File muss im IDRT-Import-Tool als PatientID angegeben werden.

Beispiel ID-File:

Unique Identifier	Nachname	Vorname	Geschlecht	Geburtstag	Meldedatum
0	Timo	Müller	f	1923-03-12	2011-08-10
1	Moritz	Berg	m	1911-04-03	2004-11-24
2	Rolf	Bali	m	1909-05-04	2007-04-13
3	Moritz	Klein	m	1910-02-02	2007-04-13

**Die Reihenfolge der Spalten ist fest und muss exakt eingehalten werden! Das Geschlecht kann „m oder f“ (für „male/female“) annehmen und jedes Datum muss das Format „yyyy-MM-dd“ haben.**

Beispiel Import-Datei:

Unique Identifier	Encounter ID	Birthyear	AGE	Height	Weight
0	1000	1971	41	209	50
1	1001	1964	48	148	50
2	1002	1933	79	148	93
3	1003	1963	49	134	75

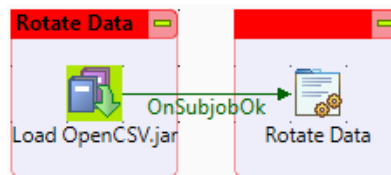


- **Cleanup:** Durchsucht die Datenbank nach vollendetem Import nach unverlinkten Patienten und entfernt diese. Dadurch befinden sich nur Patienten in der Datenbank, zu denen es auch Fakten gibt. Des Weiteren wird die Patient\_Count Spalte in der i2b2-Tabelle geschrieben.

Des Weiteren werden hier die zu importierenden CSV-Dateien geladen. Dafür wird der Ordner in der CV **folderCSV** nach CSV-Dateien durchsucht und es wird geprüft, ob es eine entsprechende Config-Datei gibt. Ist dies der Fall wird mit der Extraktion der Daten begonnen.

## Extraktion

Bei der Extraktion wird zunächst die geladene CSV-Datei um 90° gedreht und in ein Entity-Attribute-Value-Modell (EAV-Modell) gebracht. Hierfür wird der SubJob Rotate Data ausgeführt. Dieser lädt zwei Bibliotheken und führt eine Java-Methode in der Routine IDRTHelper aus.



Gleichzeitig wird die i2b2-Ontologie erstellt. Diese wird in der Datei folderMain/folderOutput/ont.csv (Trennzeichen: Tabulator) gespeichert, wobei **folderMain** ein Pfad beginnend z.B. mit C:/ (Windows) oder /home/ (Linux) sein muss und **folderOutput** einen relativen Namen wie „output“ oder „temp“ tragen muss.

Diese CSV-Datei hat den folgenden Ausgang:

**HLEVEL** - äquiv. zu HLEVEL in i2b2-Tabelle, gibt die Ebene in der Ontologie an.

**Name** - Der angezeigte Name in der Ontologie.

**Path** - Der Pfad zu diesem Namen. Wird in C\_FULLNAME und C\_DIMCODE verwendet.

**DataType** - Der Datentyp des Items.

**Update\_Date** - Füllt die UPDATE\_DATE-Spalte.

**Import\_Date** - Füllt die IMPORT\_DATE-Spalte.

**Download\_Date** - Füllt die DOWNLOAD\_DATE-Spalte.

**PathID** - Ein eindeutiger Bezeichner für das jeweilige Item.

**visual** - Füllt die C\_VISUALATTRIBUTES-Spalte in der Tabelle.

**itemCode** - Falls eine Codeliste mit dem Item verknüpft ist, dann ist hier der eindeutige Bezeichner des Code-Items anzugeben. (Std. leer)

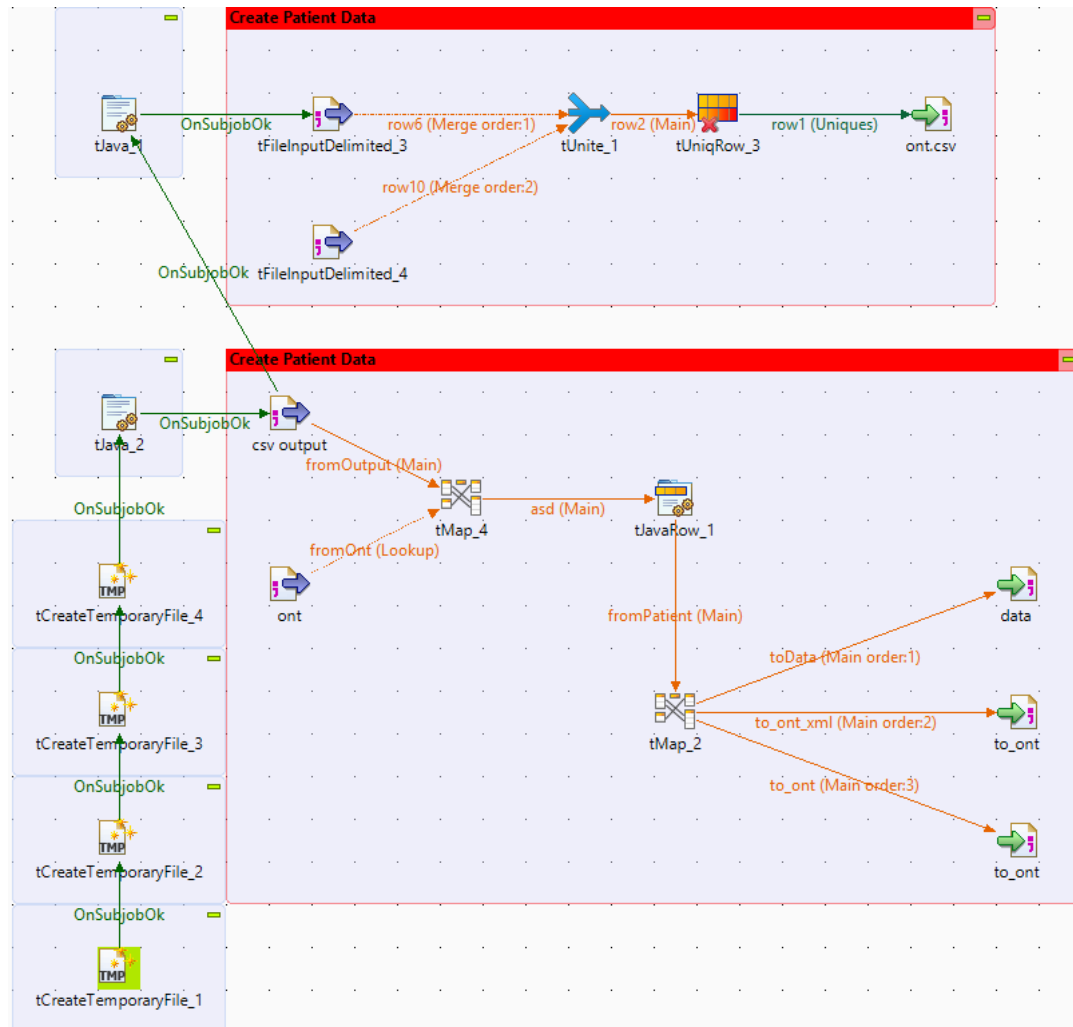
**source** - Die Quelle des Imports. (Std. leer)

**StartDate** - StartDate der Observation\_Fact

Dieses Ontologie-Schema ist generisch und wird auch bei den anderen Export-Jobs verwendet und kann für einen eigenen Extraktor verwendet werden.

Im nächsten Schritt werden die Patienten Daten extrahiert.





Hier werden die untersten Blätter, was die Antworten bzw. Values im EAV-Modell darstellen, mit der Ontologie verknüpft. Des Weiteren entsteht eine weitere Datei, data.csv (Trennzeichen: Tabulator), die im selben Ordner wie die ont.csv liegt. Die Datei hat den folgenden Inhalt:

**itemID** - Eindeutige ID des Items.

**Value** - Der Wert des Items.

**VisitID** - Die zugehörige Visite.

**FormID** - Das zugehörige Formular.

**SubjectKey** - Patienten ID.

**Path** - Der Pfad zum Item aus der Ontologie.

**PathID** - Eindeutiger Bezeichner des Pfades.

**DataType** - Datentyp.

**Update\_Date** - Füllt die UPDATE\_DATE-Spalte.

**Import\_Date** - Füllt die IMPORT\_DATE-Spalte.

**Download\_Date** - Füllt die DOWNLOAD\_DATE-Spalte.

**StudyEventRepeatKey** - Falls eine Studie wiederholt wird (z.B. beim FollowUp), wird hier hochgezählt.

**itemGroupRepeatKey** - Falls eine ItemGroup wiederholt wird (z.B. Patientenidentifikation), wird hier hochgezählt.

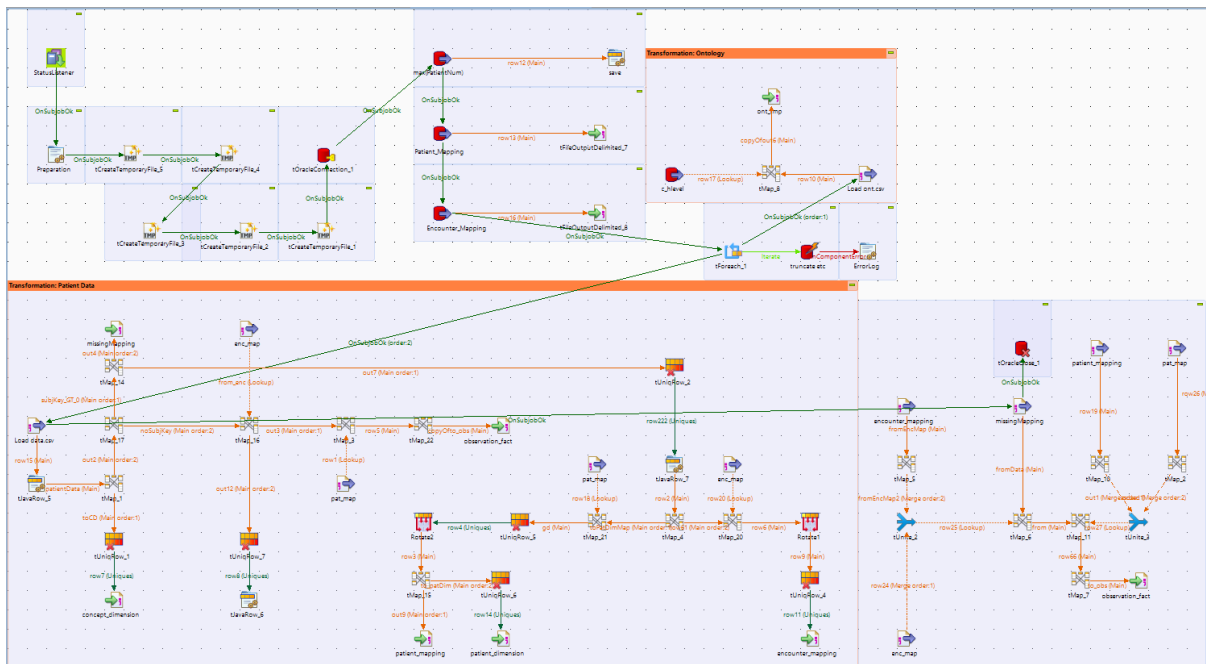
**startDate** - Anfangsdatum des Items.

**source** - Quellsystem des Items.

## Transformation

Der SubJob **Create DB Schema**, der für die Transformation der Daten zuständig ist, benötigt die vorher erstellten Dateien **ont.csv** und **data.csv**. Aus diesen Dateien werden die folgenden CSV-Dateien erstellt, die im Format den i2b2-Tabellen in der Datenbank gleichen:

- i2b2
- observation\_fact
- concept\_dimension
- patient\_mapping
- patient\_dimension
- encounter\_mapping



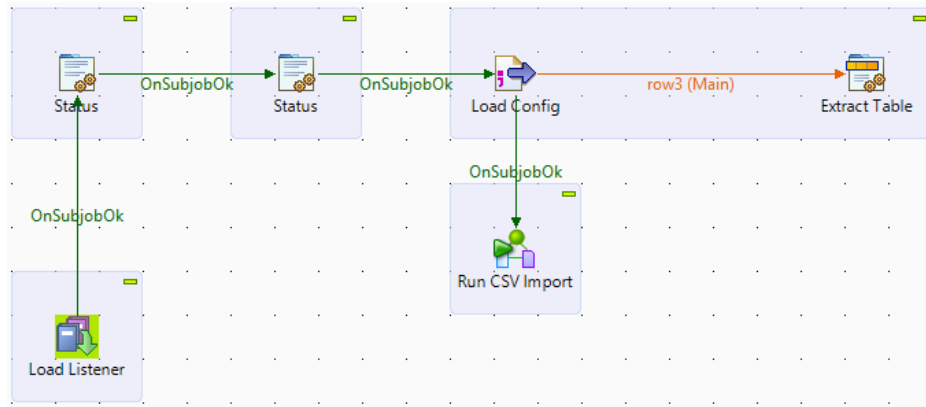
## Load

Im letzten Schritt werden die einzelnen CSV-Dateien aus dem Transformations-Schritt in die i2b2-Datenbank geladen. Dafür werden zunächst



## TOS-Job: DB-Import

Der DB-Import-Job exportiert zunächst alle in einer Konfigurationsdatei angegebenen Tabellen und führt darauf den CSV-Import-Job aus. Die nötigen Vorbereitungen werden anhand eines Beispiels erläutert. Um die Vorbereitungen zu umgehen, kann man den Import einmalig mit dem IDRTImporter durchführen und die automatisch generierten Konfigurationsdateien für den TOS-Job verwenden.



## DB Konfigurationsdateien

Der DB-Import benötigt mehrere Konfigurationsdateien. Die Erste (Trennzeichen: Semikolon) muss die Verbindungsinformationen zu den jeweiligen Tabellen beinhalten, die importiert werden sollen und wird in der CV **exportDBConfig** angegeben.

Dies ist eine Beispielkonfiguration (exportDBConfig.csv):

Server Name	Server IP	Server Port	Server SID	Server Username	Server Password	Server Schema	Server Table
importDB	192.168.1.131	1521	XE	SYSTEM	demouser	I2B2IDRT	Datatable

Diese Konfigurationsdatei wird Zeilenweise abgearbeitet und die Daten der Tabellen werden extrahiert. Da das Schema der jeweiligen Tabelle ungewiss ist, bietet sich die Talend Komponente tOracleInput leider nicht an, und muss durch eigenen Java-Code realisiert werden. Dieser befindet sich in der Routine **ExportDB**.

Zu allen Tabellen muss eine eigene Konfigurationsdatei erstellt werden (vgl. CSV-Import), die den Namen <Server Name>\_<Server Schema>\_<Server Table>.cfg.csv hat. In unserem Beispiel würde die Konfigurationsdatei der zu importierenden Tabelle den Namen importDB\_I2B2IDRT\_Datatable.cfg.csv haben und hätte den aus der Vorbereitung des CSV-Imports bekannten Inhalt,

Spaltenname (Pflicht)	PID	Date1	Date2	Date3	Source	Item1	Item2	Item3	Item4
Datentyp (Pflicht)	Integer	Date	Date	Date	String	Integer	Float	Integer	String
Name (kann leer sein)	PID	UpdateDate	ImportDate	DownloadDate	Sourcesystem	EncounterID	Name1	Name1	Name3
Metainformationen (mind. PatientID oder EncounterID)	PatientID	UpdateDate	ImportDate	DownloadDate	Sourcesystem	EncounterID			

wobei die Zeile Spaltennamen die Namen der Spalten der Datenbanktabelle beinhalten. Diese Konfigurationsdateien müssen im Ordner der CV **csvFolder** liegen und werden beim anschließenden CSV-Import verwendet.

## TOS-Job: ODM-Import

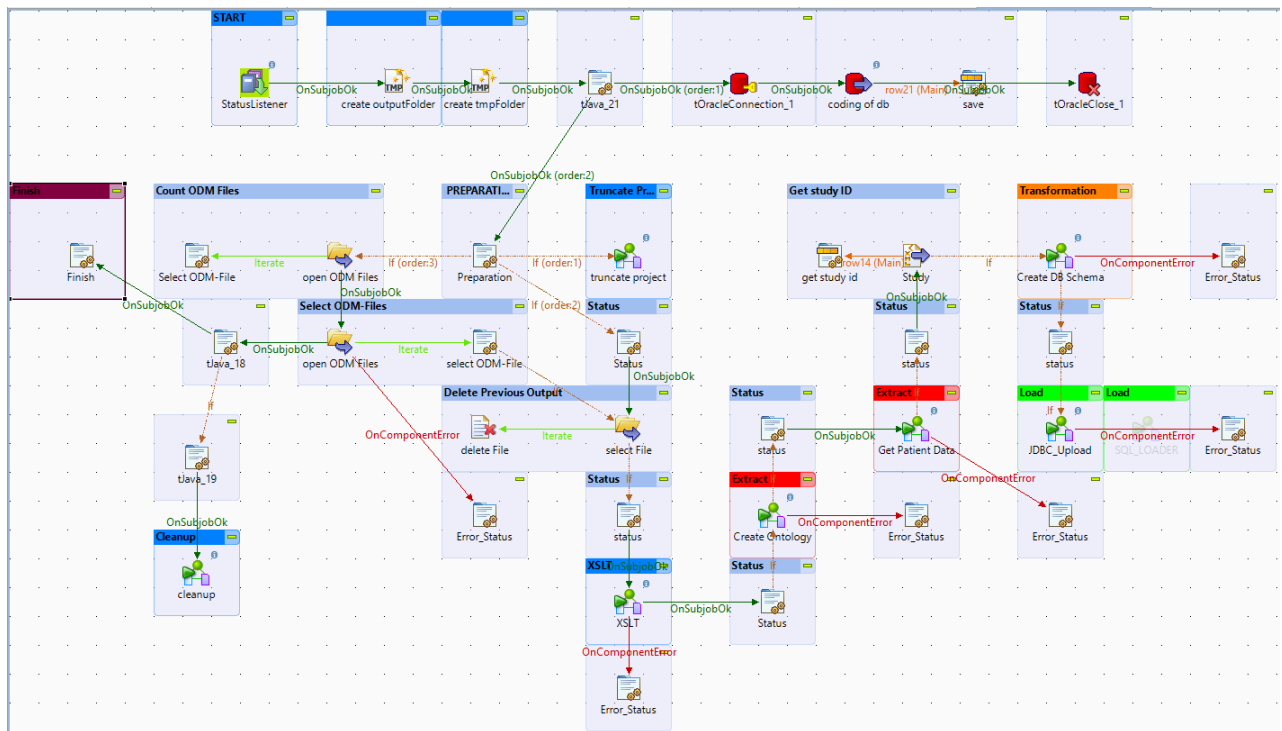
Der ODM-Import Job in Talend Open Studio verläuft ähnlich dem CSV-Import. Der Job ist in vier Bereiche unterteilt, wobei jeder aus einem oder mehreren Unterjobs besteht:

**Blau** – Vor- und Nachbereitung.

**Rot** – Extraktion der Daten aus einer ODM-Datei, Erstellung der Ontologie und der Patientendaten.

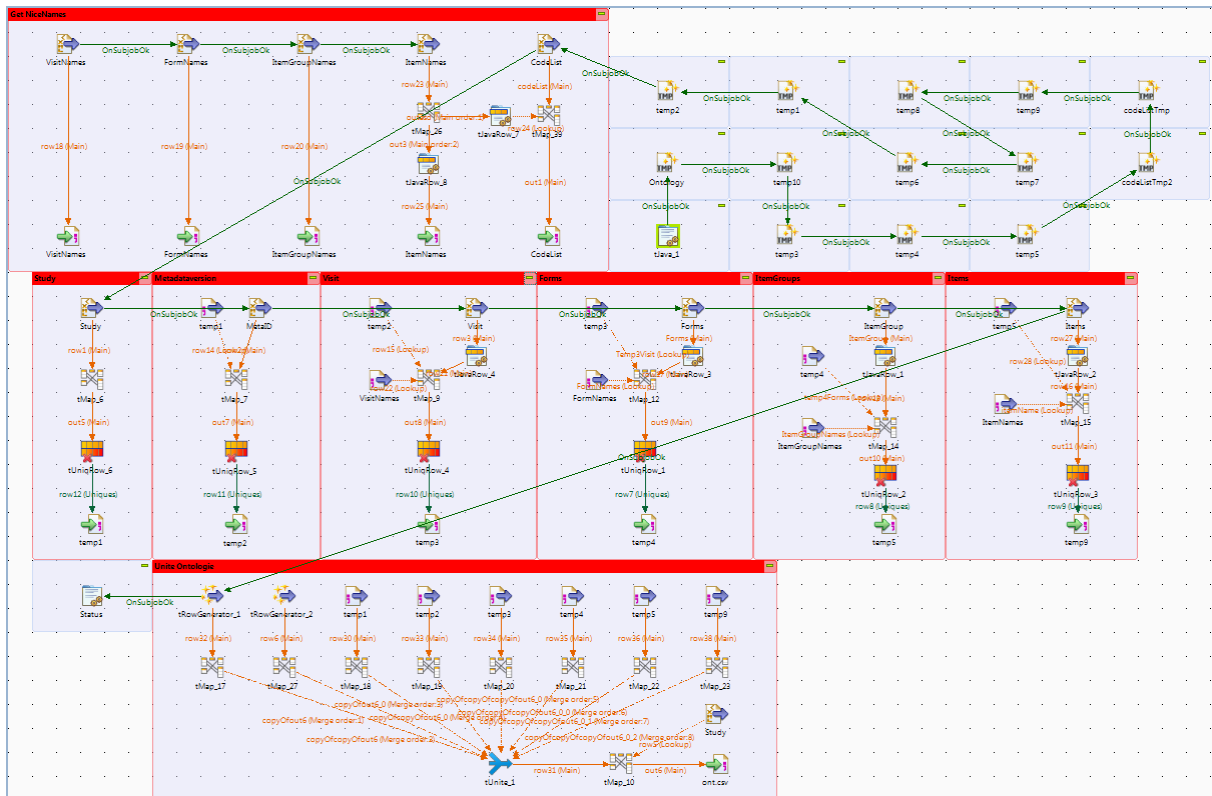
**Orange** – Transformation in das i2b2-Schema.

**Grün** – Laden des fertigen Schemas in die i2b2-Datenbank.



## Extraktion - Ontologie

Der Extraktions-Job, um die Ontologie zu erstellen, lädt die aktuelle ODM-Datei und verfolgt einen Top-Down Ansatz. Das bedeutet, dass die ODM-Datei Schritt für Schritt vom höchsten bis zum niedrigsten Element durchgegangen wird. Dabei werden alle Elemente extrahiert, in temporäre Dateien zwischengespeichert und anschließend in eine CSV-Datei ont.csv vereint.



Die Ontologie wird in der Datei folderMain/folderOutput/ont.csv (Trennzeichen: Tabulator) gespeichert, wobei **folderMain** ein Pfad beginnend z.B. mit C:/ (Windows) oder /home/ (Linux) sein muss und **folderOutput** einen relativen Namen wie „output“ oder „temp“ tragen muss. Diese CSV-Datei hat den folgenden Ausgang:

**HLEVEL** - äquiv. zu HLEVEL in i2b2-Tabelle, gibt die Ebene in der Ontologie an.

**Name** - Der angezeigte Name in der Ontologie.

**Path** - Der Pfad zu diesem Namen. Wird in C\_FULLNAME und C\_DIMCODE verwendet.

**DataType** - Der Datentyp des Items.

**Update\_Date** - Füllt die UPDATE\_DATE-Spalte.

**Import\_Date** - Füllt die IMPORT\_DATE-Spalte.

**Download\_Date** - Füllt die DOWNLOAD\_DATE-Spalte.

**PathID** - Ein eindeutiger Bezeichner für das jeweilige Item.

**visual** - Füllt die C\_VISUALATTRIBUTES-Spalte in der Tabelle.

**itemCode** - Falls eine Codeliste mit dem Item verknüpft ist, dann ist hier der eindeutige Bezeichner des Code-Items anzugeben. (Std. leer)

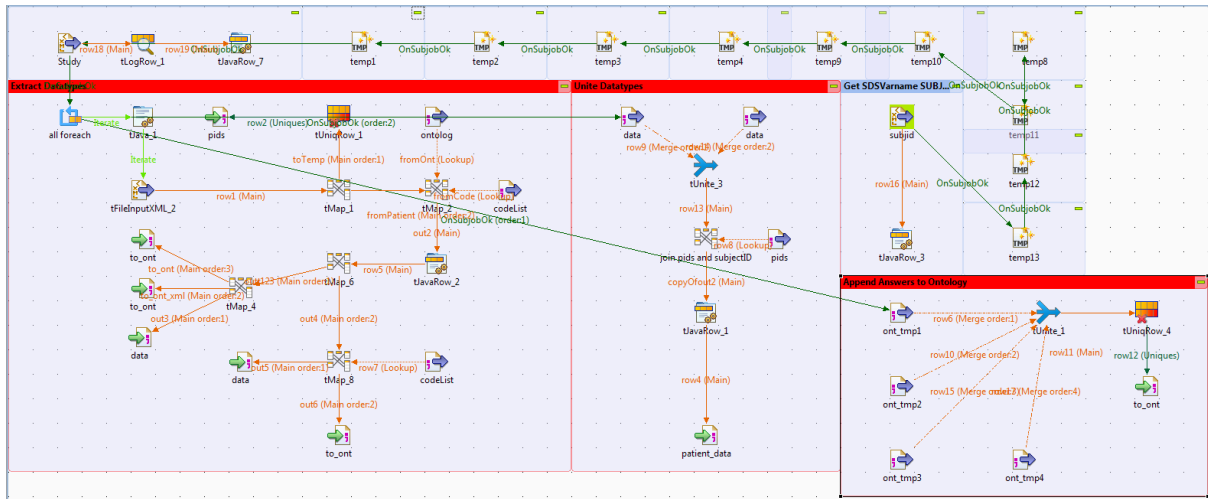
**source** - Die Quelle des Imports. (Std. leer)

**StartDate** - StartDate der Observation\_Fact

## Extraktion – Patientendaten

Als Vorbereitung der Extraktion dient der SubJob XSLT, der zunächst alle Namespaces enternt und durch XSLT-Umwandlungen die ODM-Datei in Kopie derart verändert, dass die nachfolgende Extraktion performanter ist.

Alle Datentyp-Items der neuen ODM-Datei werden in einer For-Schleife extrahiert und in die Zwischendatei data.csv geschrieben. Des Weiteren werden alle Antworten an die richtige Stelle der Ontologie als Blätter angehängt.



Die data.csv, die im selben Ordner wie die ont.csv liegt, hat den folgenden Inhalt:

**itemID** - Eindeutige ID des Items.

**Value** - Der Wert des Items.

**VisitID** - Die zugehörige Visite.

**FormID** - Das zugehörige Formular.

**SubjectKey** - Patienten ID.

**Path** - Der Pfad zum Item aus der Ontologie.

**PathID** - Eindeutiger Bezeichner des Pfades.

**DataType** – Datentyp.

**Update\_Date** - Füllt die UPDATE\_DATE-Spalte.

**Import\_Date** - Füllt die IMPORT\_DATE-Spalte.

**Download\_Date** - Füllt die DOWNLOAD\_DATE-Spalte.

**StudyEventRepeatKey** - Falls eine Studie wiederholt wird (z.B. beim FollowUp), wird hier hochgezählt.

**itemGroupRepeatKey** - Falls eine ItemGroup wiederholt wird (z.B. Patientenidentifikation), wird hier hochgezählt.

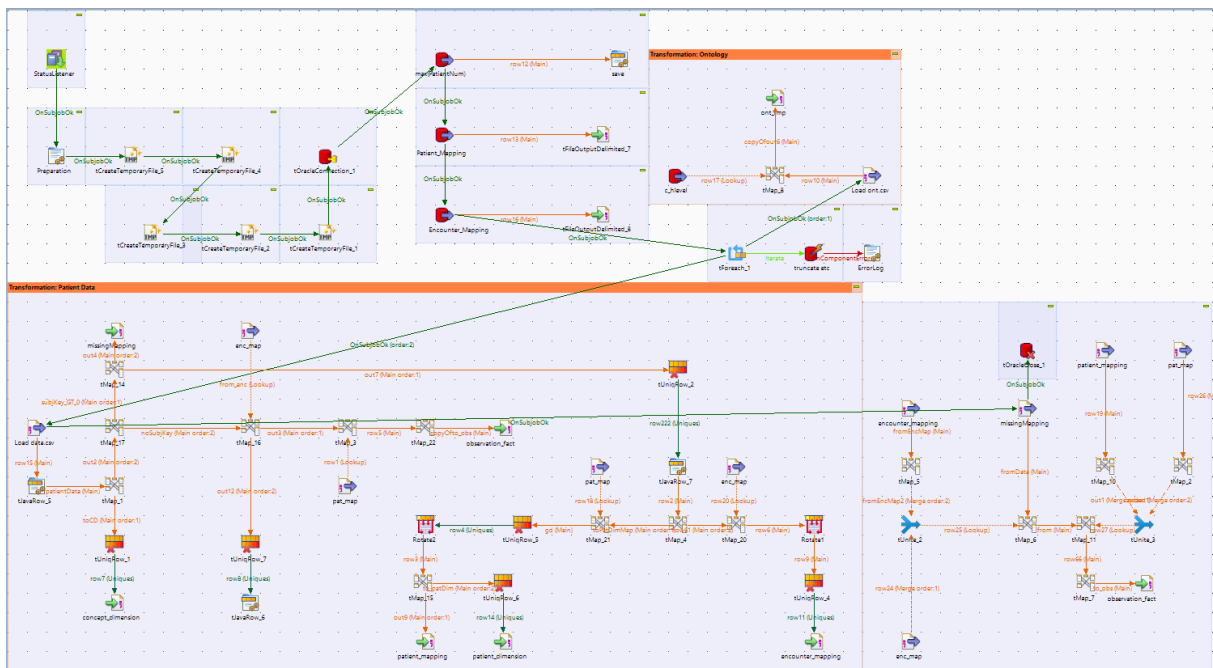
**startDate** - Anfangsdatum des Items.

**source** - Quellsystem des Items.

## Transformation

Der SubJob **Create DB Schema**, der für die Transformation der Daten zuständig ist, benötigt die vorher erstellten Dateien **ont.csv** und **data.csv**. Aus diesen Dateien werden die folgenden CSV-Dateien erstellt, die im Format den i2b2-Tabellen in der Datenbank gleichen:

- i2b2
- observation\_fact
- concept\_dimension
- patient\_mapping
- patient\_dimension
- encounter\_mapping



## Load

Im letzten Schritt werden die einzelnen CSV-Dateien aus dem Transformations-Schritt in die i2b2-Datenbank geladen.



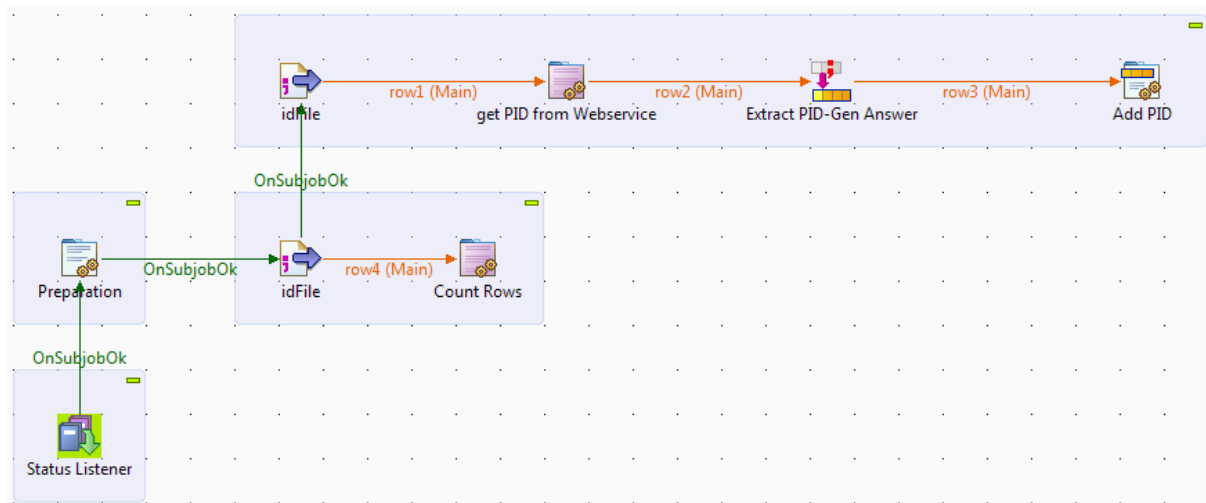
## TOS-Job: PID-Generator Anbindung

Der PID-Generator SubJob kommuniziert über die Webschnittstelle des TMF-PID-Generators mit Diesem. Dort werden Name, Vorname, Geburtstag, Geschlecht und Meldedatum aus einer ID-File übertragen. Als Rückgabe bekommt der Aufruf die neue PID des Patienten oder eine Fehlermeldung.

ID-File.csv (Trennzeichen: Semikolon, keine Überschrift) hat das folgende Schema:

Unique Identifizier	Nachname	Vorname	Geschlecht	Geburtsdatum	Meldedatum
String/Integer	String	String	m/f	yyyy-MM-dd	yyyy-MM-dd

Der „Unique Identifier“ ist das Verknüpfende Element zwischen den identifizierenden- und den medizinischen Daten.





## ***Fehlerbehandlung***

**F:** Der Import schlägt fehl.

**L:** Sollten noch i2b2-User angemeldet sein, kann es passieren, dass bestimmte, für den Import wichtige, Tabellen gesperrt sind. Durch den Menüeintrag Remove Locks können diese gelöst werden (Siehe Überblick).

**F:** Meine ODM-Daten sind sehr groß und das Programm bricht mitten im Import ab.

**L:** Den zugeteilten Java Heap Space in der IDRTImport.ini vergrößern.

**F:** Der CSV-Import funktioniert nicht.

**L:** Sicherstellen, dass die Config-Datei exakt das richtige Format hat und alle Spalten übereinstimmen.