PRORATIO
SMART SOLUTIONS

Kettle

TURNING DATA INTO BUSINESS

ProSAPCONN© & Kettle©

Version 2.2

Quick Start and Reference Guide

2006-01-30

Bischheimer Weg 1
55129 Mainz, Germany

www.proratio.de/prosapconn

Table of contents

1 What is ProSAPCONN & Kettle?

With **ProSAPCONN** you can read data from all tables of your SAP R/3® system.

We provide a special plugin for **Kettle** – an open source data warehouse ETL-tool for **E**xtraction, **T**ransformation, **T**ransportation and **L**oading of data, in short: ETTL.

All this makes data warehouses much easier to build, update and maintain. People working with ProSAPCONN find the extraction of SAP® data very easy and usefull. Not only for data warehousing also for ad hoc analyses as a consultant, administrator or user.

Kettle provides a graphical user interface **Spoon** - a user friendly designer for your transformations, also for your SAP® queries over tables.

After designing your transformations let them run interactive with Spoon or in batch mode with **Pan**.

A **free trial version** is available from www.proratio.de/prosapconn .

If you need any special feature, training or consulting don´t hesitate to contact us – we could do it.

2 Documentation overview

The following manuals are included in your **trial and registered** version:

Manual	**Description**
ProSAPCONN - 1 - Quick Start and Reference	a short reference for beginners and a more detailed reference
ProSAPCONN - 2 – FAQ and Trouble Shooting	all FAQs beginning with PSCFAQ will be found here and the trouble shooting part helps you solving problems

The following manuals describe SAP® customizing. They are available for **registered** users only or upon special request.

Manual	**Description**
ProSAPCONN - A - Customizing for Optimization	You need this manual when using floating points, special numeric values, SAP® release 4.7 or optimizing speed for big rows.
ProSAPCONN - B - Customizing for Security	This is a step by step introduction for customizing security with ProSAPCONN.

3 Installation

3.1 Software

3.1.1 Supported platforms, operating systems

The installation process is described for windows platforms. If you need further assistance for other platforms please let us know. ProSAPCONN & Kettle supports many different platforms.

3.1.2 Java Runtime Environment

You need Java 2 Platform, Standard Edition (J2SE) version 1.4 or higher. You could check if you have already installed it with:
Start / Settings / Control Panel / Add or remove programs
[German: Start / Einstellungen / Systemsteuerung / Software]

If you haven´t installed it yet, please download a version 1.4 or higher from http://java.sun.com/j2se/ and install it.

3.1.3 Kettle

Unzip the Kettle zip-file in a directory of your choice, e.g. in
C:\Program files\Kettle

Try to run Spoon.bat. If it fails the Java Runtime Environment is not installed or see the FAQ-Manual: PSCFAQ0001 - How to set the correct Java Runtime Version?

For a detailed description of the installation consult the Kettle manual.

3.1.4 ProSAPCONN as a Kettle plugin

The ProSAPCONN plugin is delivered with Kettle. If you want to install an update version see PSCFAQ0002 - How to install an updated version of ProSAPCONN without installing Kettle?

3.1.5 SAP® JCo

The SAP® Java Connector (SAP® JCo) is a toolkit that allows a Java application to communicate with any SAP® System. Unfortunately the redistribution of the SAP® JCo is not allowed, so you have to download it from the SAP® Service Marketplace at

http://service.sap.com/connectors SAP® Java Connector / Tools & Services

There you will also find all available distribution packages for the various supported platforms and processors.

If you don´t have a login for the SAP® Service Marketplace ask your SAP® administrator or request the data by SAP® (see PSCFAQ0003 - How to get access to the SAP® Service Marketplace?).

After you have downloaded the latest JCo, e.g. sapjco-ntintel-2.1.6.zip extract this to a temporary folder.

Copy the following files out of the zip archive:

Filename	**Target folder**
sapjco.jar	Copy to the directory plugins\steps\ProSAPCONN below the kettle path: e.g. C:\Program files\Kettle\plugins\steps\ProSAPCONN
sapjcorfc.dll	Copy to the directory libswt\win32 below the kettle path: e.g. C:\Program files\Kettle\libswt\win32 [This is different to the JCo documentation.]
librfc32.dll	Copy to the directory [Windows]\System32

Important note for librfc32.dll:
Please check if you have an old version of librfc32.dll in your [Windows]\System32 folder. If so please update it with the newer librfc32.dll. Sometimes this fails because the DLL is in use by another process. If so, please restart your system and don´t start the SAP® GUI before you replace the old DLL.

In future releases of ProSAPCONN this installation process will be more automated.

3.2 Firewall configuration

JCo uses the native RFC library with the CPI-C protocol based on TCP/IP for its low level network communication. For CPI-C you need special TCP/IP ports going through your firewall.

These ports are 3300 plus your target system number e.g. 3308 for system number 08.

For detailed information on the network configuration see the JCo documentation […]/docs/jco/configuration.html

3.3 Customizing security on your SAP® system (basics)

In order to retrieve meta data information from the SAP® system's data dictionary, you have to grant the access rights (authorization object: S_RFC, ACTVT: 16, FUGR) for the following function groups:

R/3 Release	Function Groups
since 3.1H	RFC1, SG00, SRFC, SUNI, SYST
since 4.0A	RFC1, SDIF, SG00, SRFC, SYST, SYSU, SUNI
since 4.6A	RFC1, SDIF, SG00, SRFC, SYST, SYSU
since 4.6D	RFC1, SDIFRUNTIME, SG00, SRFC, SYST, SYSU

For requesting the data we use RFC_READ_TABLE belonging to function group SDTX or a customized RFC belonging to function group ZPSCSDTX . So you have to grant the access rights (authorization object: S_RFC, ACTVT: 16, FUGR) also to **SDTX** or **ZPSCSDTX**.

The requested table is checked against the table group authorization, defined in the authorization object **S_TABU_DIS**. If you want to have security on table level (and not only on table groups as the default) or need a more detailed description on the above mentioned topics you can request the special **manual "Customizing for Security".** If you´re a registered user you already have this manual.

3.4 Customizing the SAP® System for optimization

You should customize a special RFC (Remote Function Call) for

1. adding security to the **table level** (and not only on table groups) or
2. giving **better performance** on table rows exceeding 512 bytes or
3. using **floating point** values and special numerical values or
4. using release **4.7**

ProSAPCONN is automatically determining the capability of the SAP® system and uses a customized RFC if it is available. If it uses the default RFC_READ_TABLE it runs automatically multiple queries to get the needed data exceeding the 512 bytes boundary. So more overhead is there and if data changes between queries you possibly get wrong data. There are some internal tests getting the right data but we recommend using the customized RFC.

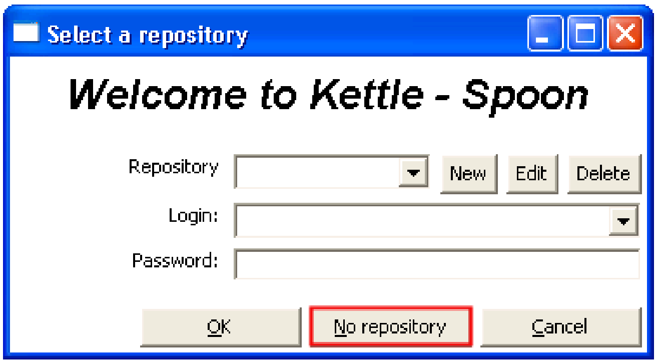
You get a detailed description in the **manual "Customizing for Optimization"** as a registered user or upon special request.

4 Introduction - first steps

4.1 Start Spoon – the graphical user interface of Kettle

- Start Spoon.bat

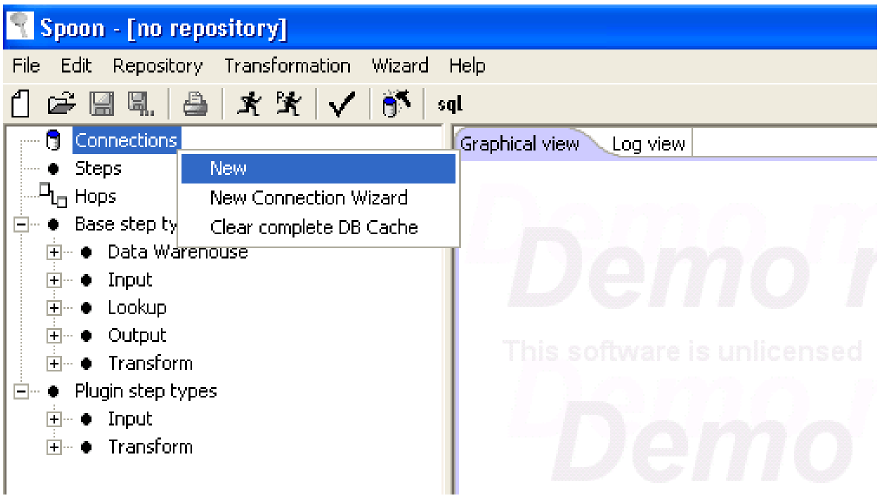
- If you´re asked for “Select a repository” press “No repository” (you can save your work in XML files – later on you can use repositories for saving your transformations in a repository database).



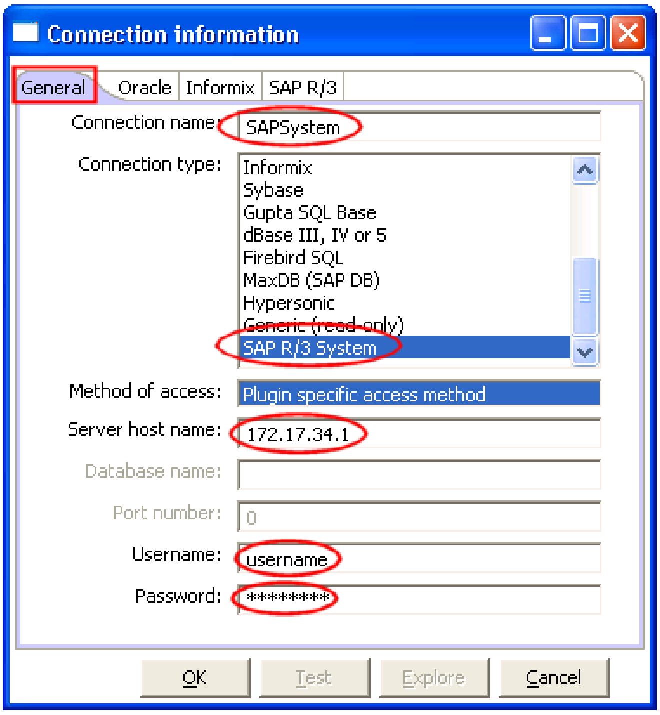
For a detailed description of Spoon consult the Kettle manual (see the PDFs in your Kettle-directory under docs).

4.2 Setting up the connection

- Select “New” in the Connections context menu:



- Enter your corresponding connection information like this:



Connection name: SAPSystem

Connection type: Informix, Sybase, Gupta SQL Base, dBase III, IV or 5, Firebird SQL, MaxDB (SAP DB), Hypersonic, Generic (read-only), SAP R/3 System

Method of access: Plugin specific access method

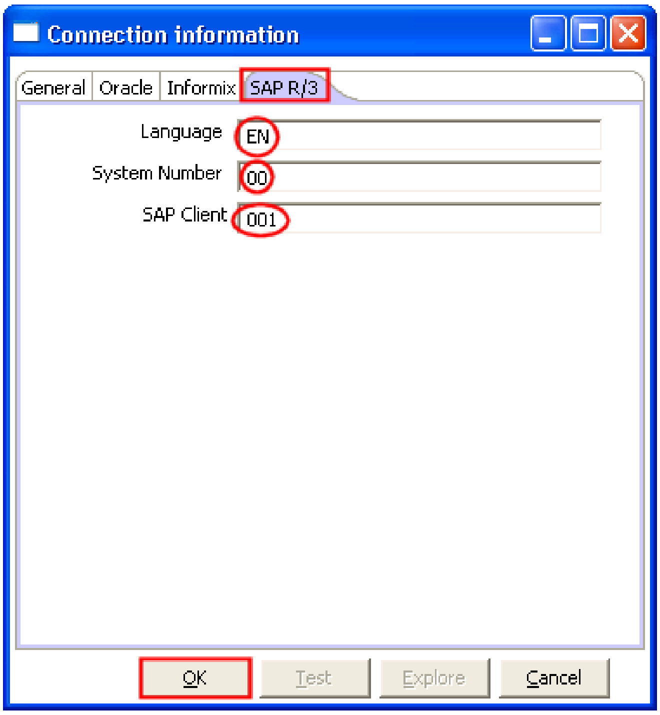
Server host name: 172.17.34.1

Database name:

Port number: 0

Username: username

Password: *********



Language: EN

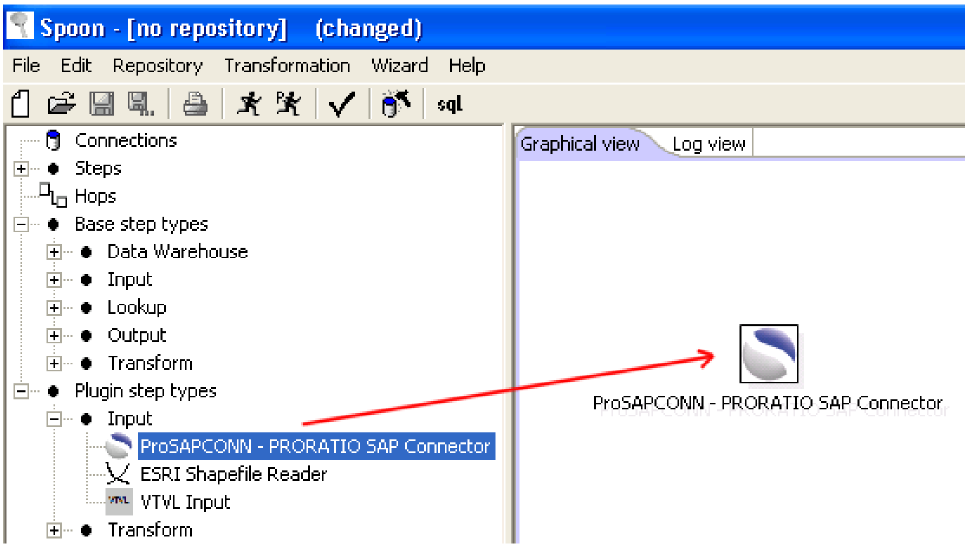
System Number: 00

SAP Client: 001

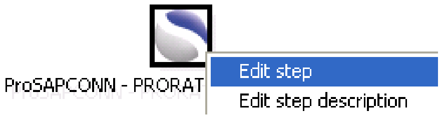
4.3 Entering the ProSAPCONN (trial) license code

The first time you edit a ProSAPCONN – PRORATIO SAP® Connector step you will be asked for your license code:

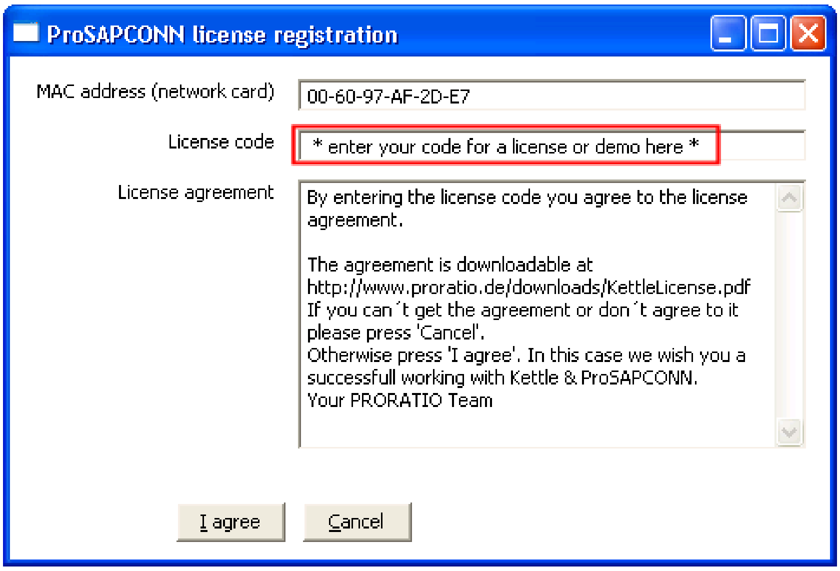
- **Draw** the “ProSAPCONN – PRORATIO SAP® Connector” step from left to right **into the “Graphical view”.**



- Select “Edit step” in the context menu (or double click)



- The dialog ProSAPCONN license registration opens. You need to enter a full licence code or demo licence code here.


ProSAPCONN license registration

MAC address (network card): 00-60-97-AF-2D-E7

License code: * enter your code for a license or demo here *

License agreement:

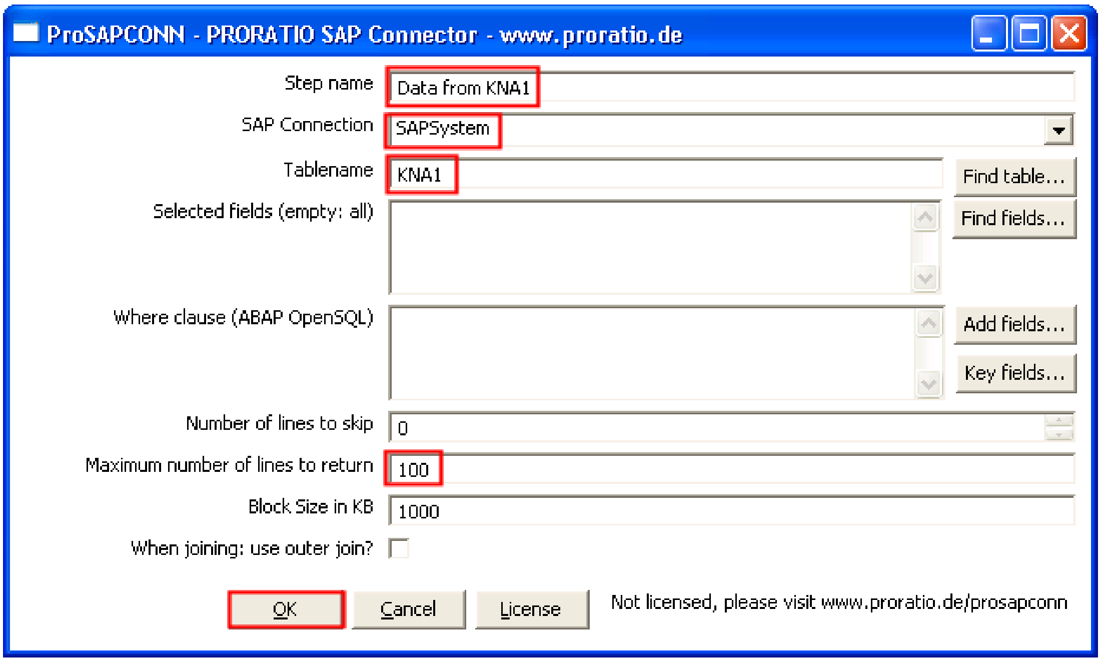
By entering the license code you agree to the license agreement.

The agreement is downloadable at http://www.proratio.de/downloads/KettleLicense.pdf
If you can´t get the agreement or don´t agree to it please press 'Cancel'.
Otherwise press 'I agree'. In this case we wish you a successfull working with Kettle & ProSAPCONN.
Your PRORATIO Team

4.4 Example: Query the customer data table KNA1

This example is a simple query for the customer data table KNA1.

- Enter the following information (we first limit to 100 rows for testing):



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name: Data from KNA1

SAP Connection: SAPSystem

Tablename: KNA1

Selected fields (empty: all):

Where clause (ABAP OpenSQL):

Number of lines to skip: 0

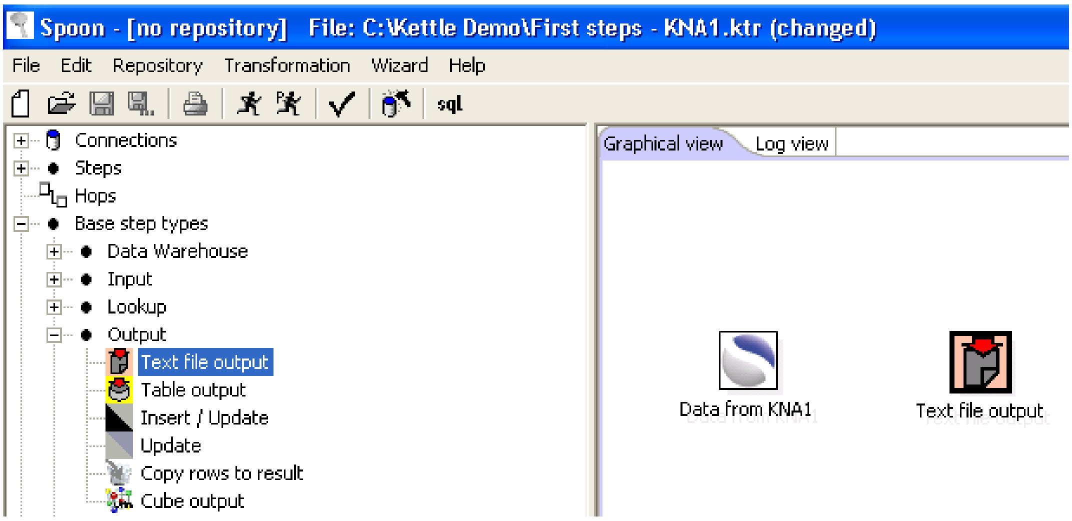
Maximum number of lines to return: 100

Block Size in KB: 1000

When joining: use outer join? □

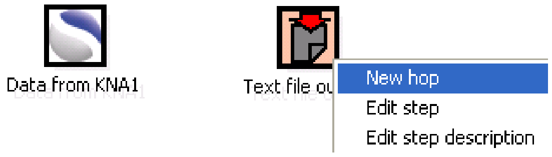
Not licensed, please visit www.proratio.de/prosapconn

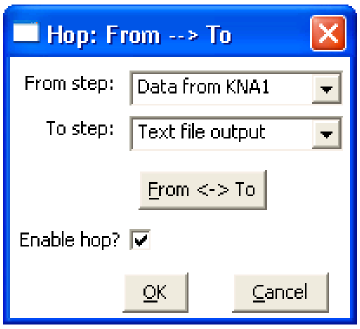
- Draw the "Output / Text file output" step from left to right into the "Graphical view".



Spoon - [no repository] File: C:\Kettle Demo\First steps - KNA1.ktr (changed)

- Now connect these two steps - this connection is called a **hop**. To do this, **select both steps (by holding the Ctrl-key)** in the “Graphical view” and select “New hop” in the context menu:



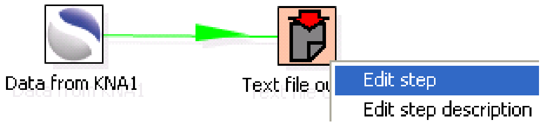


From step: Data from KNA1

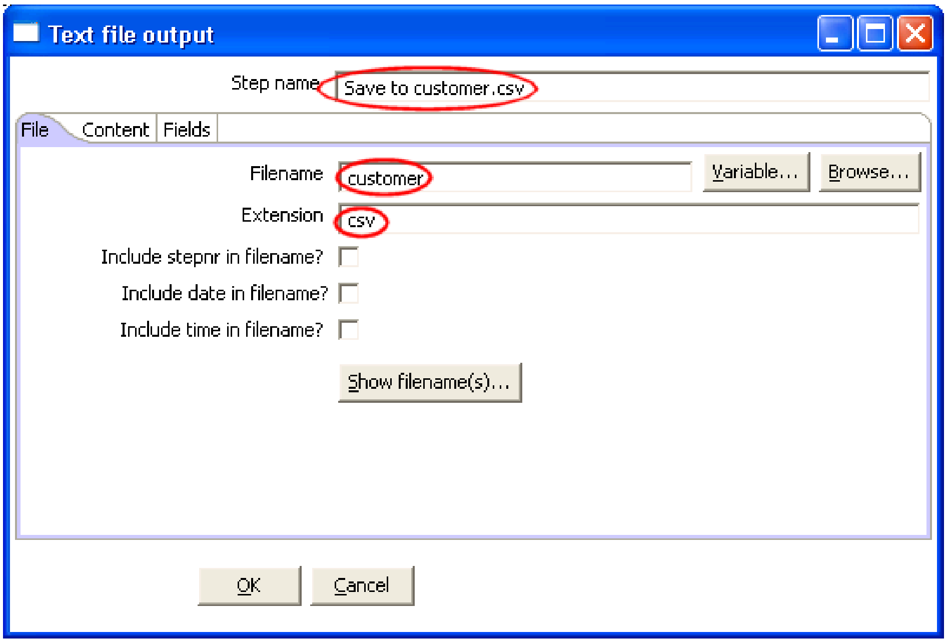
To step: Text file output

Enable hop? ☑

- The hop is created, now edit the step “Text file output”:



- Change the step name to “Save to customer.csv” and enter the filename “customer” and the extension “csv”. You could also use “txt” for the extension but with “csv” you can open this file directly with Excel.



Step name: Save to customer.csv

Filename: customer

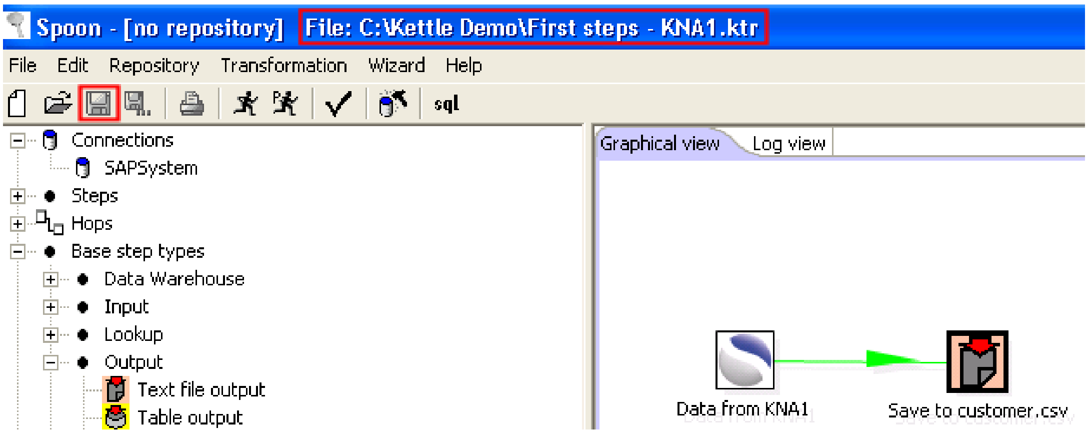
Extension: csv

Include stepnr in filename? □

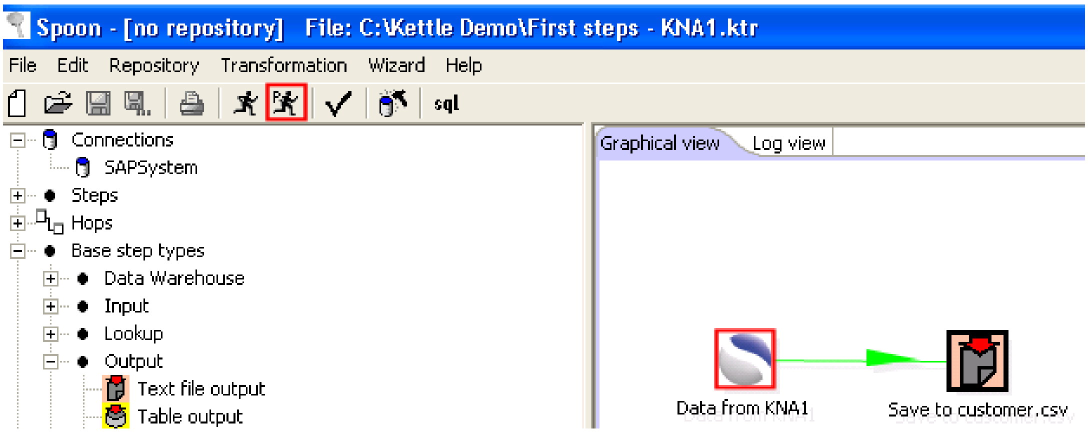
Include date in filename? □

Include time in filename? □

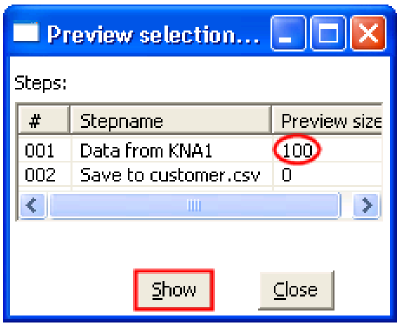
- Please save your transaction first (File / Save or CTRL-S) e.g. as “First steps - KNA1”:



- Now you´re ready to run a preview. Select the “Data from KNA1” step and the icon for running a preview:



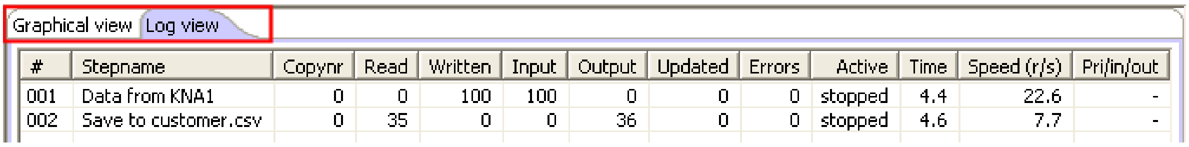
- The default preview size is 100 – press “Show”:



Steps:

#	Stepname	Preview size
001	Data from KNA1	100
002	Save to customer.csv	0

- Now you see the “Log view” and after successful connection the data. If you want to switch back to designing your transformation select the tab “Graphical view”.

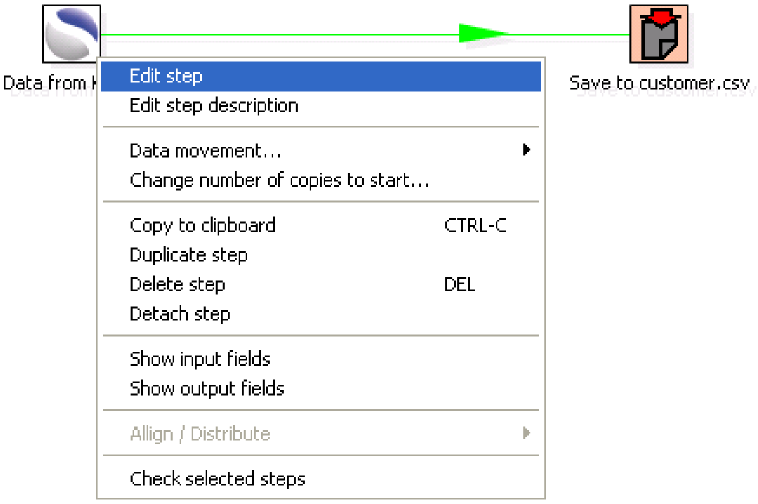


#	Stepname	Copynr	Read	Written	Input	Output	Updated	Errors	Active	Time	Speed (r/s)	Pri/in/out
001	Data from KNA1	0	0	100	100	0	0	0	stopped	4.4	22.6	-
002	Save to customer.csv	0	35	0	0	36	0	0	stopped	4.6	7.7	-

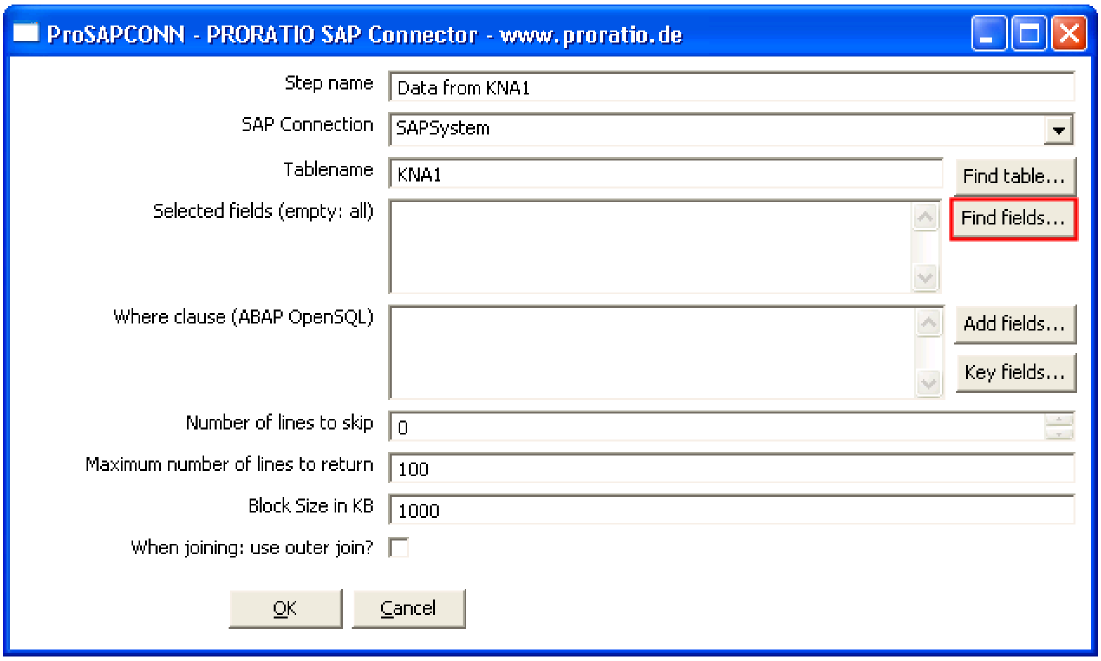
4.5 Example: Selecting fields on customer data table KNA1

This example shows you how to select specific fields on the customer data table KNA1.

- Use the transformation from the previous chapter and edit the “Data from KNA1” step:



- Press “Find fields” and select some:



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name: Data from KNA1

SAP Connection: SAPSystem

Tablename: KNA1

Selected fields (empty: all):

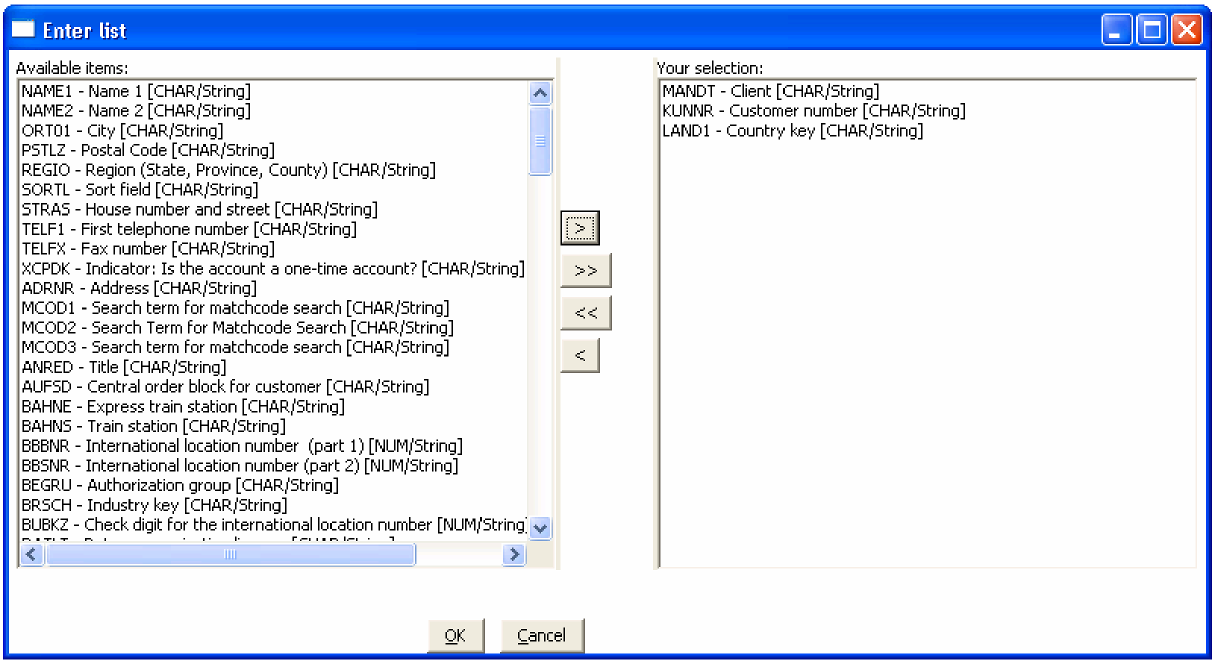
Where clause (ABAP OpenSQL):

Number of lines to skip: 0

Maximum number of lines to return: 100

Block Size in KB: 1000

When joining: use outer join? □



Enter list

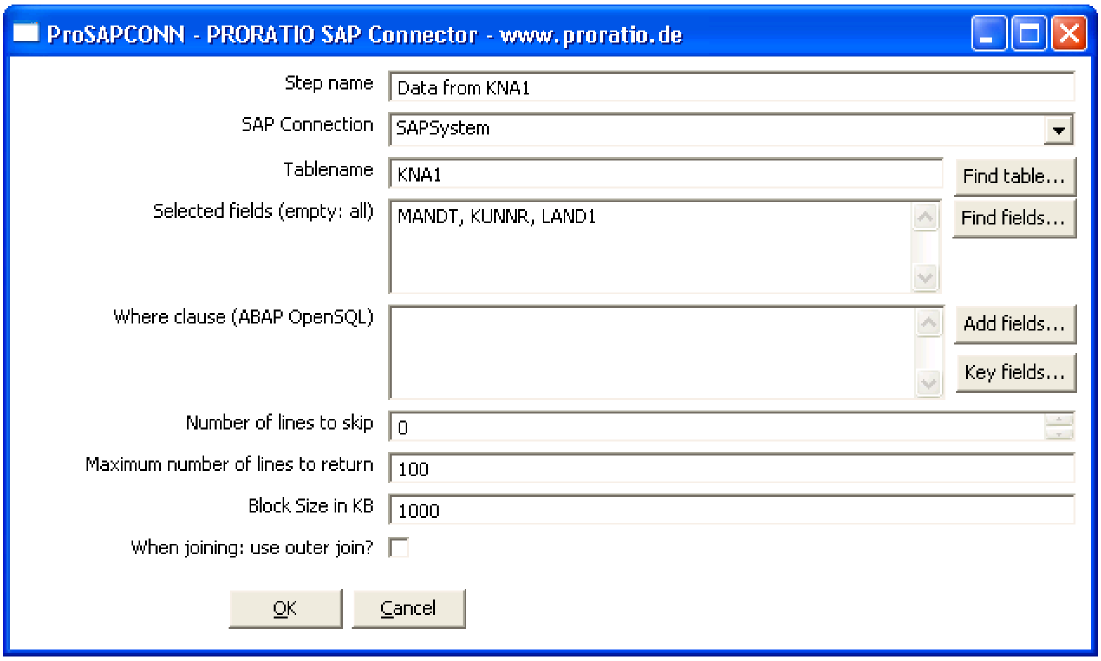
Available items:

NAME1 - Name 1 [CHAR/String]
NAME2 - Name 2 [CHAR/String]
ORT01 - City [CHAR/String]
PSTLZ - Postal Code [CHAR/String]
REGIO - Region (State, Province, County) [CHAR/String]
SORTL - Sort field [CHAR/String]
STRAS - House number and street [CHAR/String]
TELF1 - First telephone number [CHAR/String]
TELFX - Fax number [CHAR/String]
XCPDK - Indicator: Is the account a one-time account? [CHAR/String]
ADRNR - Address [CHAR/String]
MCOD1 - Search term for matchcode search [CHAR/String]
MCOD2 - Search Term for Matchcode Search [CHAR/String]
MCOD3 - Search term for matchcode search [CHAR/String]
ANRED - Title [CHAR/String]
AUFSD - Central order block for customer [CHAR/String]
BAHNE - Express train station [CHAR/String]
BAHNS - Train station [CHAR/String]
BBBNR - International location number (part 1) [NUM/String]
BBSNR - International location number (part 2) [NUM/String]
BEGRU - Authorization group [CHAR/String]
BRSCH - Industry key [CHAR/String]
BUBKZ - Check digit for the international location number [NUM/String]

Your selection:

MANDT - Client [CHAR/String]
KUNNR - Customer number [CHAR/String]
LAND1 - Country key [CHAR/String]

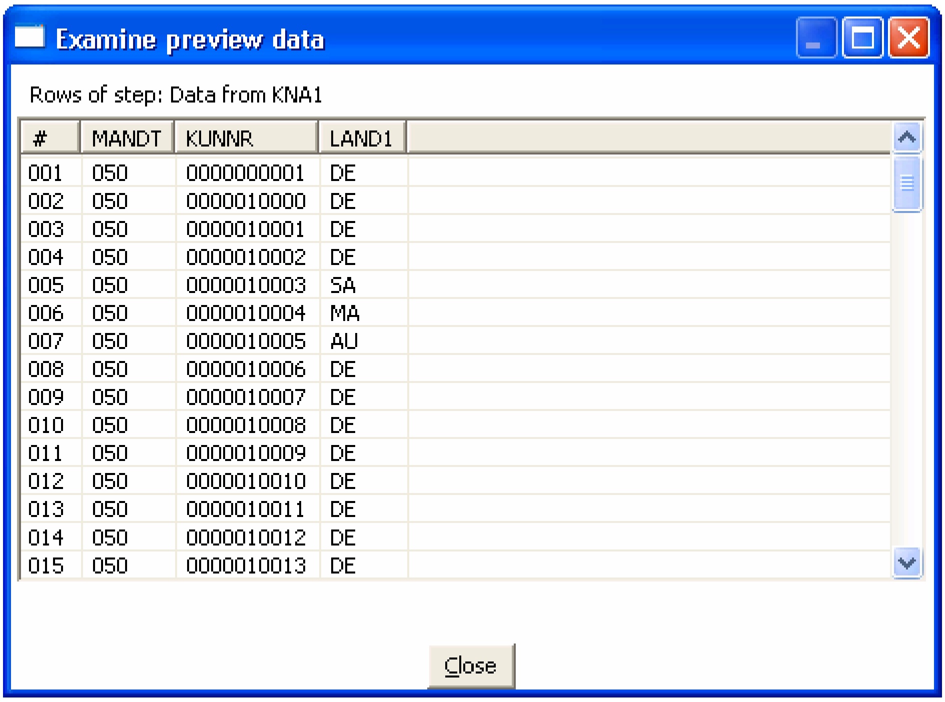
- Now the step looks like this:



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name	Data from KNA1
SAP Connection	SAPSystem
Tablename	KNA1
Selected fields (empty: all)	MANDT, KUNNR, LAND1
Where clause (ABAP OpenSQL)	
Number of lines to skip	0
Maximum number of lines to return	100
Block Size in KB	1000
When joining: use outer join?	□

- You can press OK and test the transformation by running the preview as described before. The preview could look like this:



Examine preview data

Rows of step: Data from KNA1

#	MANDT	KUNNR	LAND1
001	050	0000000001	DE
002	050	0000010000	DE
003	050	0000010001	DE
004	050	0000010002	DE
005	050	0000010003	SA
006	050	0000010004	MA
007	050	0000010005	AU
008	050	0000010006	DE
009	050	0000010007	DE
010	050	0000010008	DE
011	050	0000010009	DE
012	050	0000010010	DE
013	050	0000010011	DE
014	050	0000010012	DE
015	050	0000010013	DE

4.6 Lookup Example: Combining data with customer data table KNA1

Now we want to check which customer is in the european community and combine the data with table T005. We have two possibilities for this:

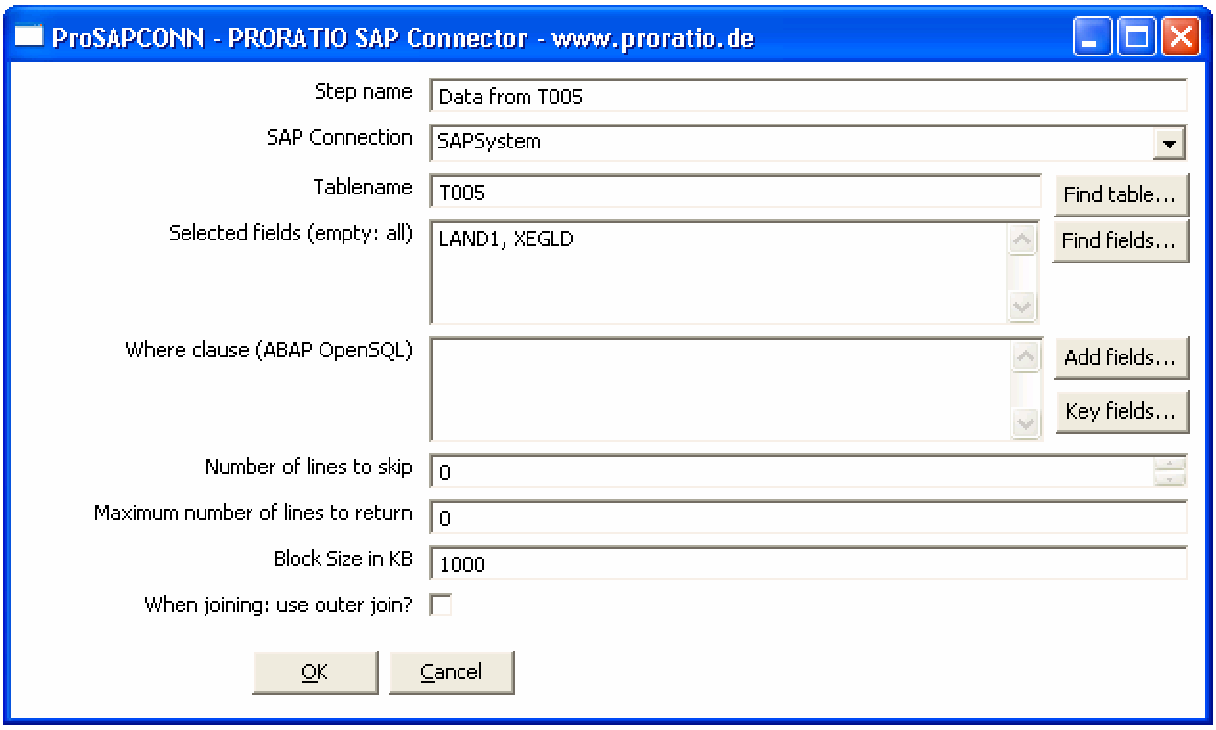
1) Download all country data from the SAP® system and do a lookup for every customer row. This is better when the lookup table (in this case T005) is small and the source data (KNA1) is big.

2) Or run a query against the lookup table T005 for each row in the source data (KNA1). This is better when the lookup table is big and would take a long time to download from the SAP® system.

In our case solution 1) would be better because the lookup table T005 is small. For completeness we show you both solutions in the following.

4.6.1 First possibility: Download T005 and combine it with KNA1

- Create a new step with the following data:



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name: Data from T005

SAP Connection: SAPSystem

Tablename: T005

Selected fields (empty: all): LAND1, XEGLD

Where clause (ABAP OpenSQL):

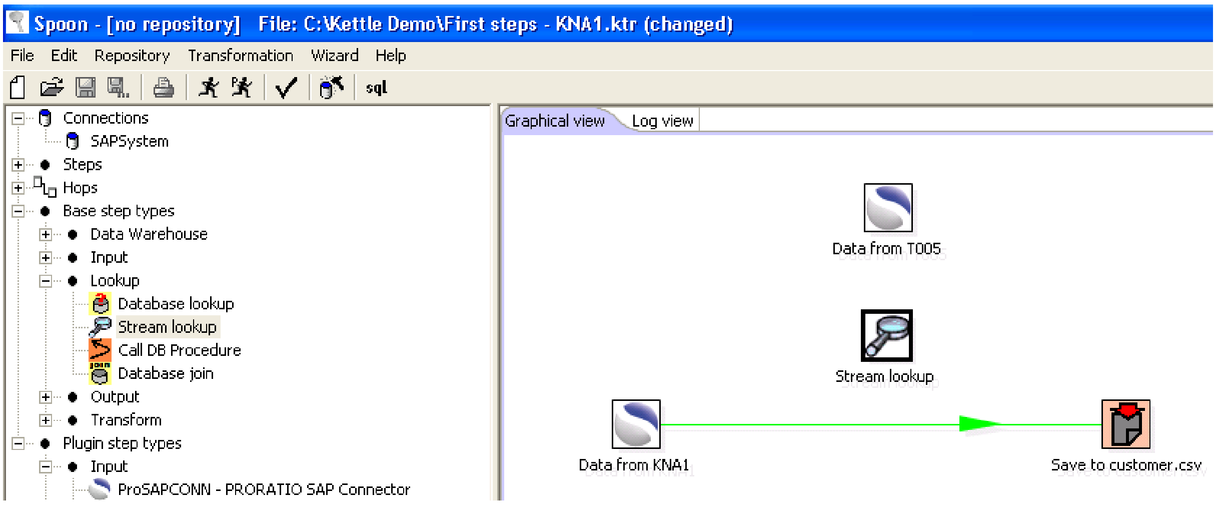
Number of lines to skip: 0

Maximum number of lines to return: 0

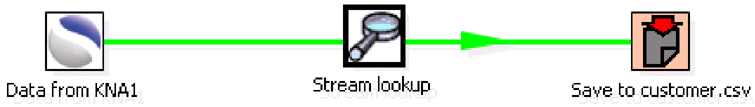
Block Size in KB: 1000

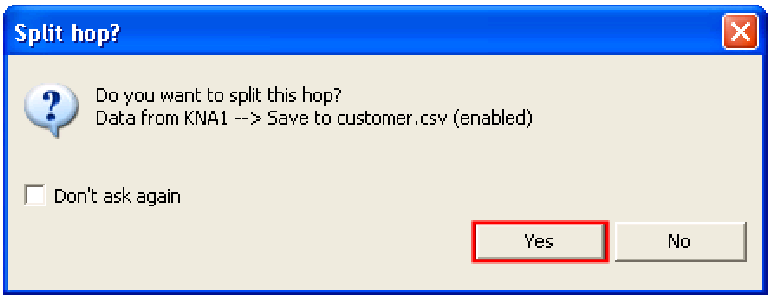
When joining: use outer join? □

- Add a “Lookup / Stream lookup” step:



- Move the “Stream lookup” over the hop between “Data from KNA1” and “Save to customer.csv”, so the arrow will be bold – release the mouse button and answer “Yes” to the following dialog:

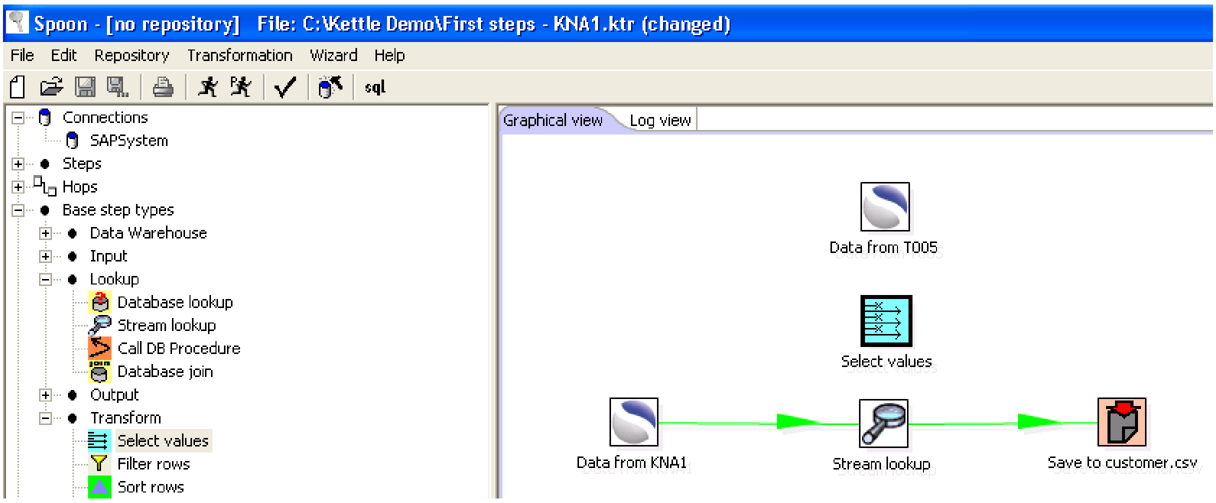




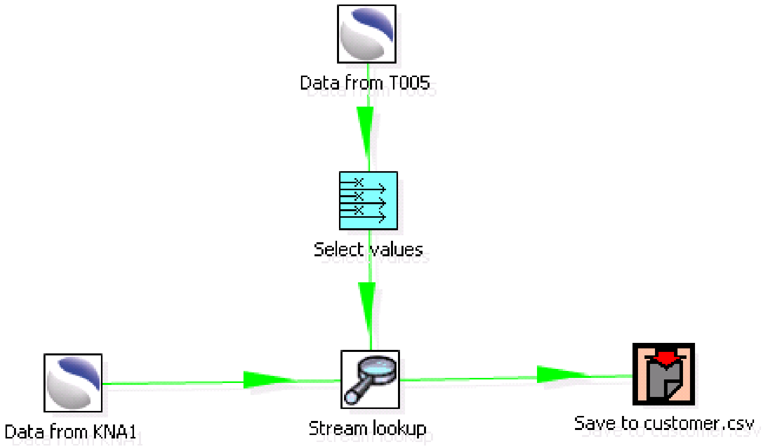
Do you want to split this hop?
Data from KNA1 --> Save to customer.csv (enabled)

□ Don't ask again

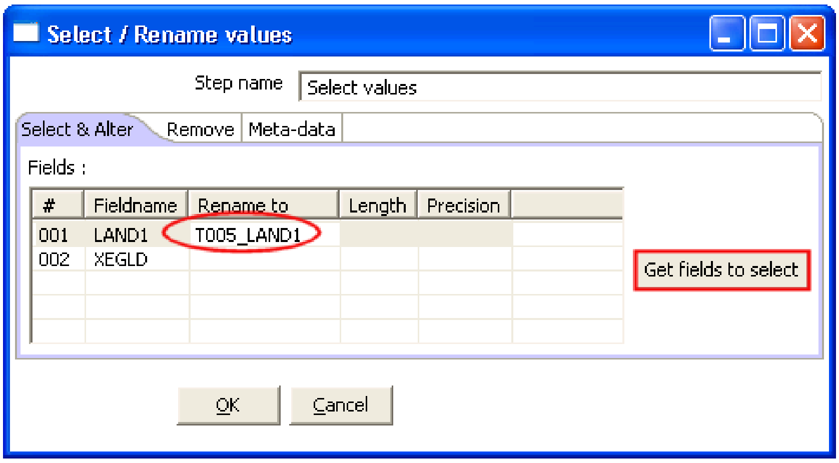
- Add a “Transformation / Select values” step:



- Add hops from “Data from T005” to “Select values” and to “Stream lookup”:



- Edit the “Select values” step: Press the button “Get fields to select” and rename “LAND1” to “T005_LAND1” (this is needed for the “Stream lookup” process when field names in KNA1 and T005 are identical.)

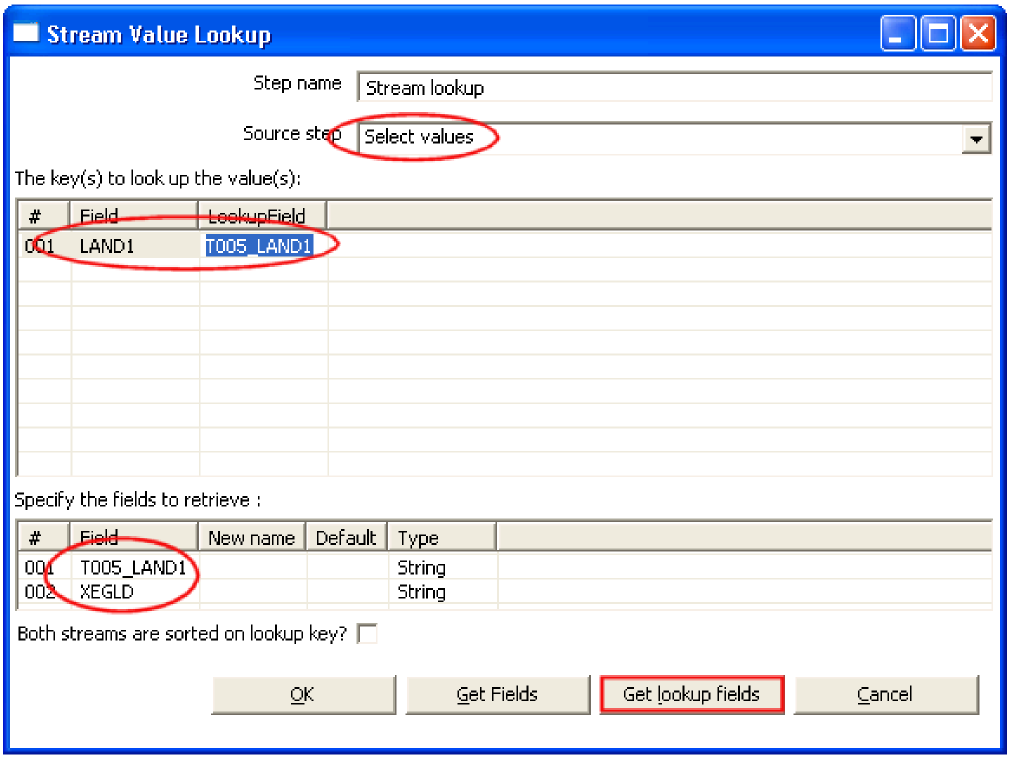


Step name: Select values

Fields :

#	Fieldname	Rename to	Length	Precision
001	LAND1	T005_LAND1		
002	XEGLD			

- Now edit the “Stream lookup” step like this:



Step name: Stream lookup

Source step: Select values

The key(s) to look up the value(s):

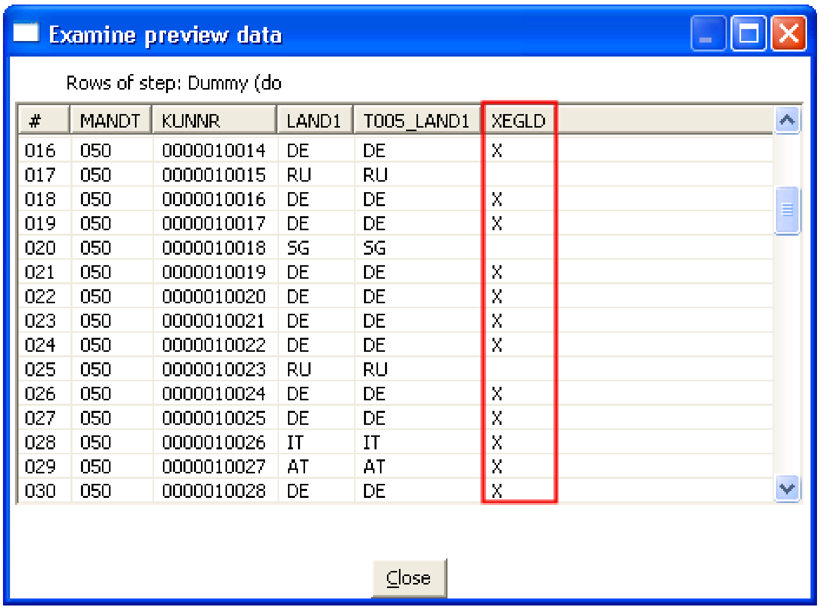
#	Field	LookupField
001	LAND1	T005_LAND1

Specify the fields to retrieve :

#	Field	New name	Default	Type
001	T005_LAND1			String
002	XEGLD			String

Both streams are sorted on lookup key? □

- Now we are ready and could check the result. Select the “Save to customer.csv” and run a preview:



Rows of step: Dummy (do

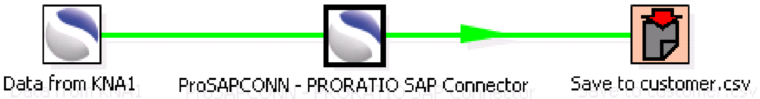
#	MANDT	KUNNR	LAND1	T005_LAND1	XEGLD
016	050	0000010014	DE	DE	X
017	050	0000010015	RU	RU	
018	050	0000010016	DE	DE	X
019	050	0000010017	DE	DE	X
020	050	0000010018	SG	SG	
021	050	0000010019	DE	DE	X
022	050	0000010020	DE	DE	X
023	050	0000010021	DE	DE	X
024	050	0000010022	DE	DE	X
025	050	0000010023	RU	RU	
026	050	0000010024	DE	DE	X
027	050	0000010025	DE	DE	X
028	050	0000010026	IT	IT	X
029	050	0000010027	AT	AT	X
030	050	0000010028	DE	DE	X

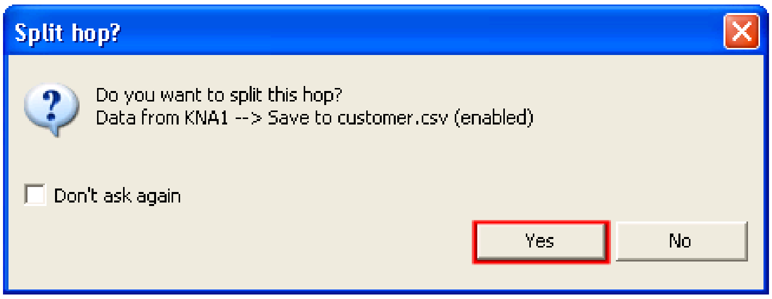
The column XEGLD shows you which customer is in the european community.

4.6.2 Second possibility: Query T005 for each KNA1 row

The second possibility is easier to customize because ProSAPCONN does a little bit helping on the relations. As mentioned above mind that this step could be slower when much rows have to be combined.

- Move a new “ProSAPCONN” step over the hop between “Data from KNA1” and “Save to customer.csv”, so the arrow will be bold – release the mouse button and answer “Yes” to the following dialog:

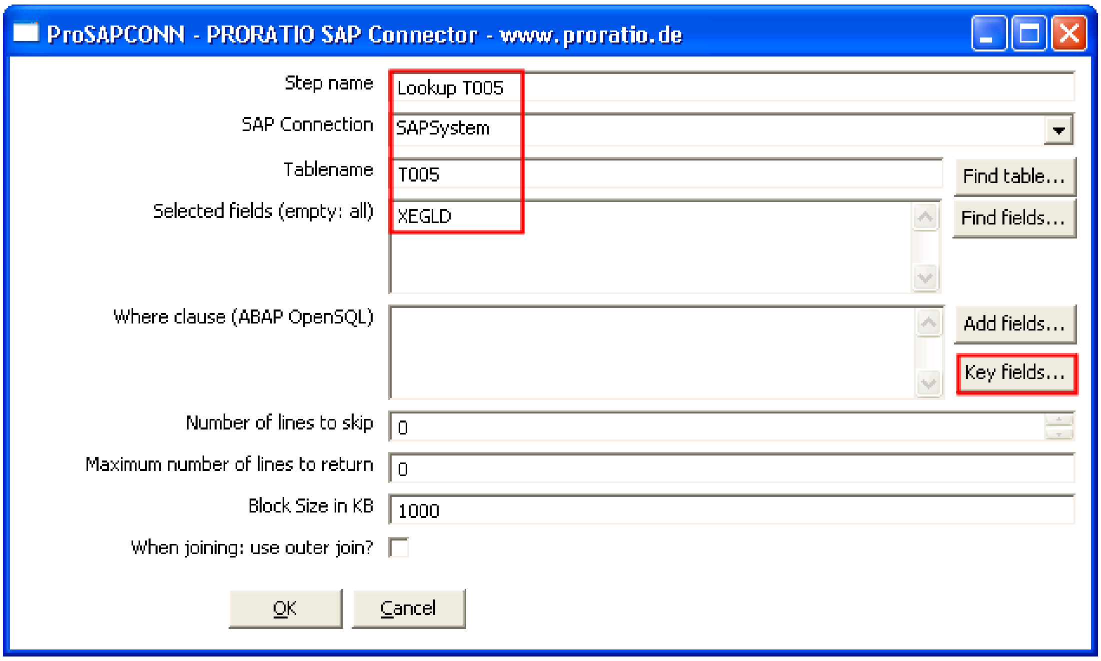




Do you want to split this hop?
Data from KNA1 --> Save to customer.csv (enabled)

□ Don't ask again

- Edit the new step as descibed here and press “Key fields”:



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name: Lookup T005

SAP Connection: SAPSystem

Tablename: T005

Selected fields (empty: all): XEGLD

Where clause (ABAP OpenSQL):

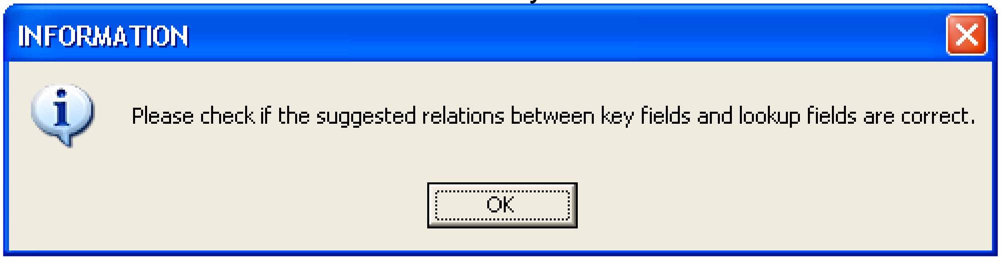
Number of lines to skip: 0

Maximum number of lines to return: 0

Block Size in KB: 1000

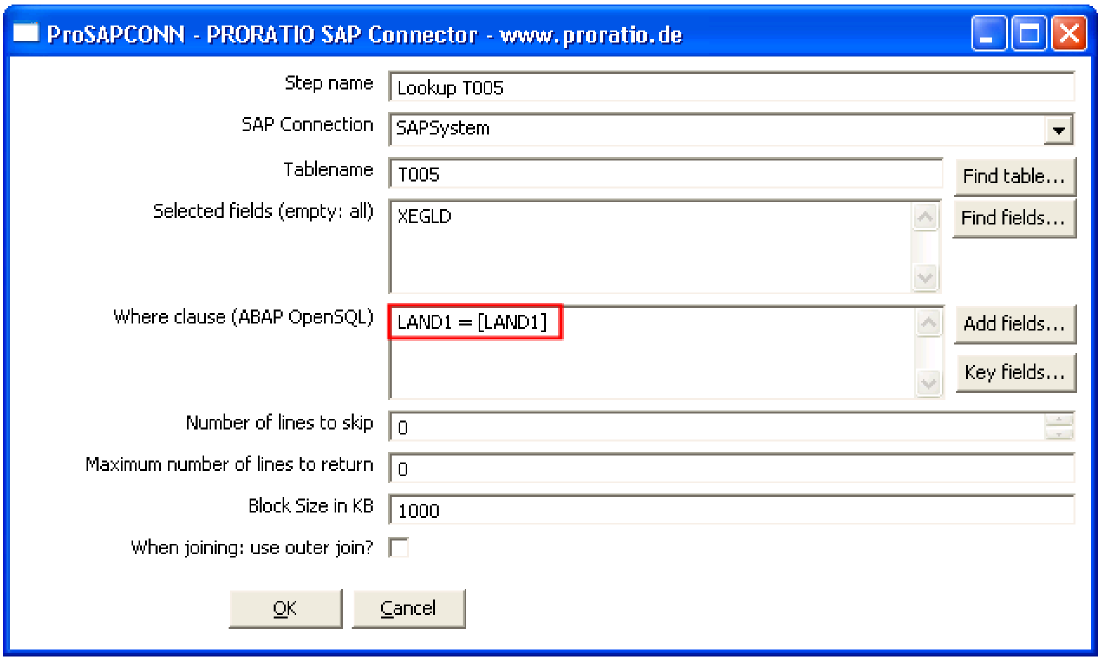
When joining: use outer join? □

Now the relations will be automatically inserted…



INFORMATION

Please check if the suggested relations between key fields and lookup fields are correct.



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name: Lookup T005

SAP Connection: SAPSystem

Tablename: T005

Selected fields (empty: all): XEGLD

Where clause (ABAP OpenSQL): LAND1 = [LAND1]

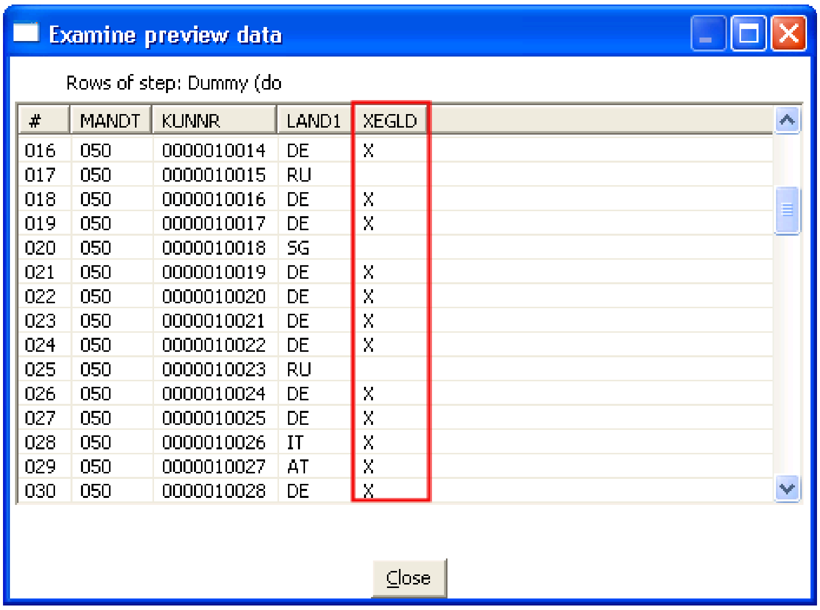
Number of lines to skip: 0

Maximum number of lines to return: 0

Block Size in KB: 1000

When joining: use outer join? □

- That´s right – so let´s run the transformation by selecting “Save to customer.csv” and do the preview:



Rows of step: Dummy (do

#	MANDT	KUNNR	LAND1	XEGLD
016	050	0000010014	DE	X
017	050	0000010015	RU	
018	050	0000010016	DE	X
019	050	0000010017	DE	X
020	050	0000010018	SG	
021	050	0000010019	DE	X
022	050	0000010020	DE	X
023	050	0000010021	DE	X
024	050	0000010022	DE	X
025	050	0000010023	RU	
026	050	0000010024	DE	X
027	050	0000010025	DE	X
028	050	0000010026	IT	X
029	050	0000010027	AT	X
030	050	0000010028	DE	X

As you saw this transformation could be a little bit slower (there is also more network traffic due to more queries so connections via VPN are slower, too).

As a further difference to the first method you see the column T005_LAND1 is no more needed. Now it´s up to you which method to use in the future.

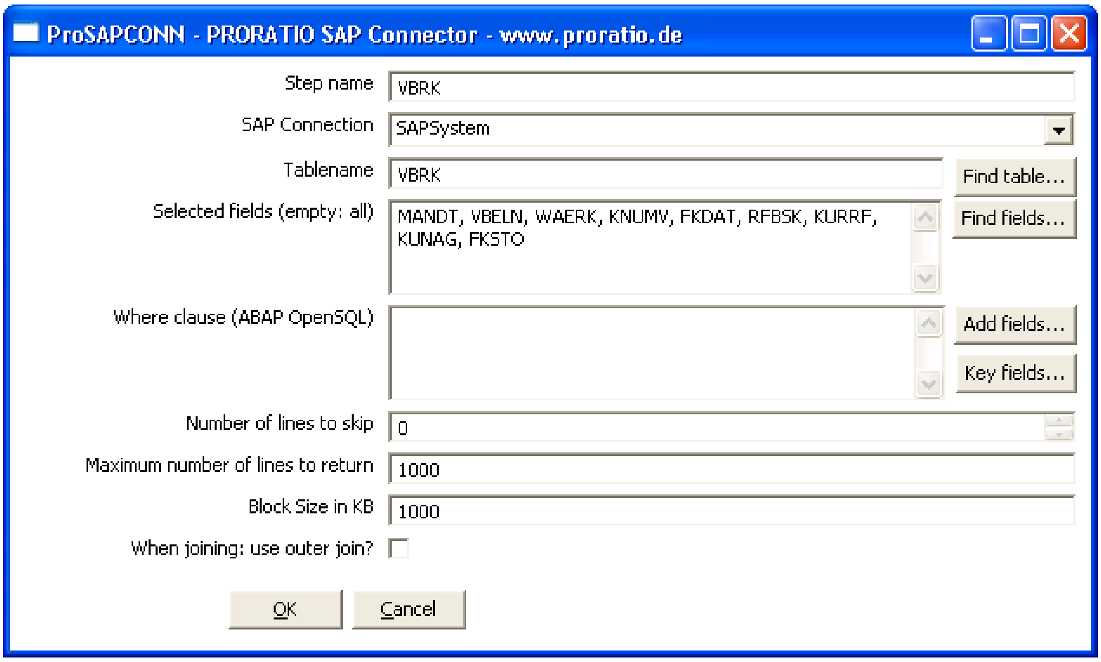
When doing a lookup you could set the “Maximum number of lines to return” to 1. Do it if there is no well-defined n:1 relationship and you want to ensure this. Otherwise it could become a join with returning more data (n:n, see next chapter).

4.7 Join Example: Joining billing header (VBRK) and item data (VBRP)

This chapter is aimed to the more experienced user, therefore not every step is described in detail as before. If you have questions, do not hesitate to contact us.

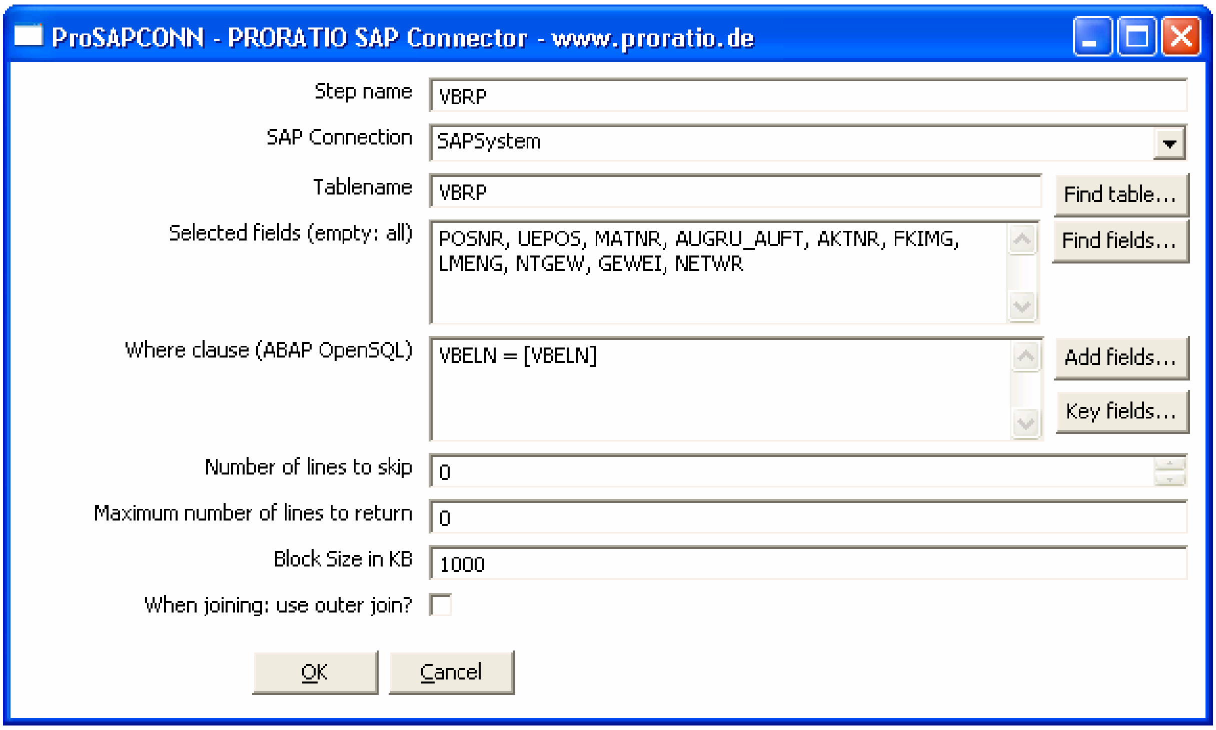
In the ETL process you need to define a start billing number (VBELN) or a start date and e.g. an end date (mostly yesterday). You could integrate this in the where clause (not shown here but the example is available on request).

Start with the billing header (VBRK) – for testing purposes define a maximum of lines:



Step name: VBRK
SAP Connection: SAPSystem
Tablename: VBRK
Selected fields (empty: all): MANDT, VBELN, WAERK, KNUMV, FKDAT, RFBSK, KURRF, KUNAG, FKSTO
Where clause (ABAP OpenSQL):
Number of lines to skip: 0
Maximum number of lines to return: 1000
Block Size in KB: 1000
When joining: use outer join? □

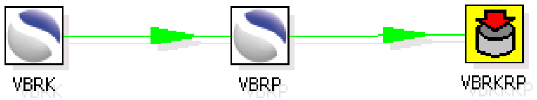
Join the billing items table VBRP:



ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name	VBRP
SAP Connection	SAPSystem
Tablename	VBRP
Selected fields (empty: all)	POSNR, UEPOS, MATNR, AUGRU_AUFT, AKTNR, FKIMG, LMENG, NTGEW, GEWEI, NETWR
Where clause (ABAP OpenSQL)	VBELN = [VBELN]
Number of lines to skip	0
Maximum number of lines to return	0
Block Size in KB	1000
When joining: use outer join?	□

Add a table output step, e.g. VBRKRP, your transformation could look like this:



In the table output VBRKRP click on “SQL” to create the target table on your database:

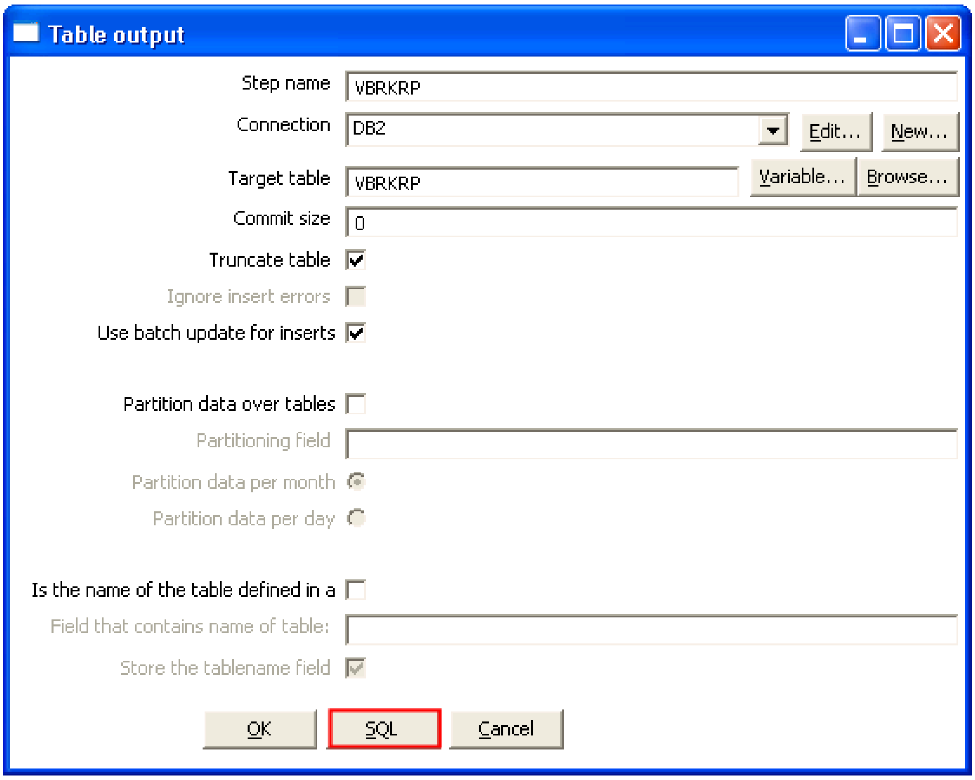


Table output

Step name: VBRKRP

Connection: DB2

Target table: VBRKRP

Commit size: 0

Truncate table ☑

Ignore insert errors □

Use batch update for inserts ☑

Partition data over tables □

Partitioning field:

Partition data per month ●

Partition data per day ○

Is the name of the table defined in a □

Field that contains name of table:

Store the tablename field ☑

Let your transformation run and refine your basic solution as you need it.

If you want to exercise and integrate the conditions, join the table KONV as a next step.

4.8 More performance aspects on joins and lookups

If you have a large amount of data, e.g. at the first initial loading of the data warehouse it could be better to load all data for a given period without joining them with ProSAPCONN. You could load them temporary with Kettle and join them with the Kettle “Lookup step” or you could store the data temporary to a database and use a join at database level.

Performance aspects depend on a lot of factors and are customer specific, mostly because of different data structures and environments. Therefore we could not predict what to do, but with ProSAPCONN you have a wide range of possibilities and a helping hand in selecting the right data.

4.9 Some examples of where clauses (ABAP® OpenSQL)

Here you will find some examples of where clauses in queries running against KNA1.

Mind that **ABAP® OpenSQL always needs a space as a seperator** before and after a fieldname and some other parts of the where part. If you run in an error check the spaces first (e.g. an error “One of the field names in the SELECT clause was not recognized” is caused by a missing space).

Wrong: PSTLZ='55129'
Correct: PSTLZ = '55129'

Combine your selections with AND OR NOT and brackets

e.g. LAND1 = ‘DE’ AND PSTLZ = ‘55129’
e.g. LAND1 = ‘DE’ AND (PSTLZ = ‘55129’ OR PSTLZ = ‘55129’)

Use of BETWEEN, LIKE, IN, IS NULL:

field **BETWEEN** a AND b
e.g. PSTLZ BETWEEN ‘55129’ AND ‘55131’
finds 55129, 55130, 55131

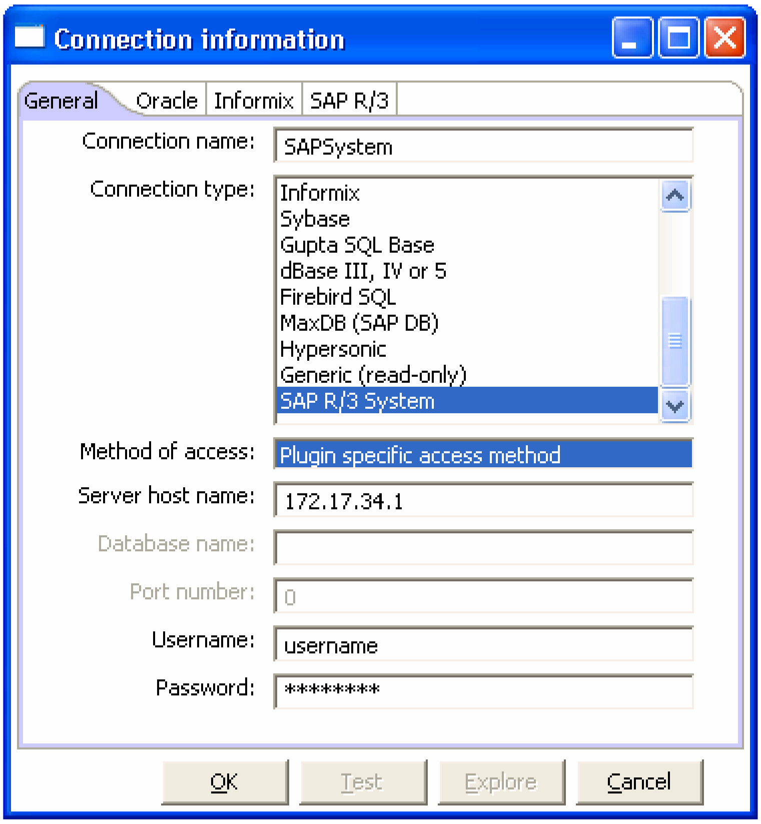
field **LIKE** '%town'
e.g. ORT01 LIKE 'MAIN%'
finds MAINZ, MAINHAUSEN, MAINZ-KASTEL

field **IN** ('a', 'b', 'c')
e.g. PSTLZ IN (‘55129’, ‘55131’)
finds rows with 55129, 55131

A real NULL (nothing) could be checked with IS NULL.
e.g. PSTLZ IS NULL
mostly this doesn´t happen in base SAP® tables

5 Reference

5.1 Connection properties



Connection name: SAPSystem

Connection type:
- Informix
- Sybase
- Gupta SQL Base
- dBase III, IV or 5
- Firebird SQL
- MaxDB (SAP DB)
- Hypersonic
- Generic (read-only)
- SAP R/3 System

Method of access: Plugin specific access method

Server host name: 172.17.34.1

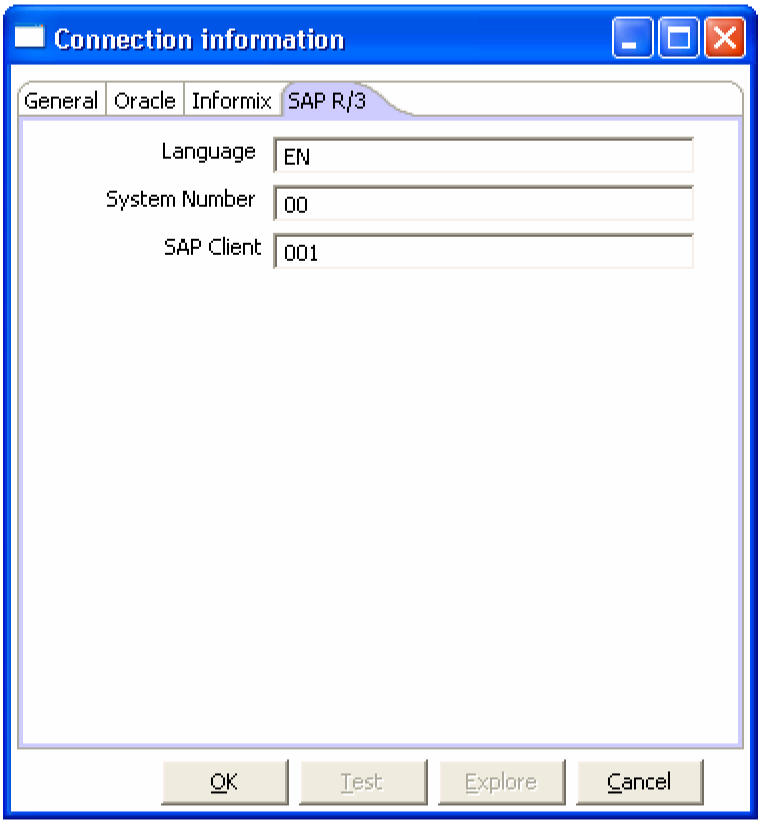
Database name:

Port number: 0

Username: username

Password: ********

Connection name	a unique connection name of your choice
Server host name	the server name or IP-Adress of your SAP® system
Username	valid user name for your SAP® system
Password	password (remind it could be case sensitive – on some systems you have to use uppercase letters even though your GUI logon don´t need it in uppercase)



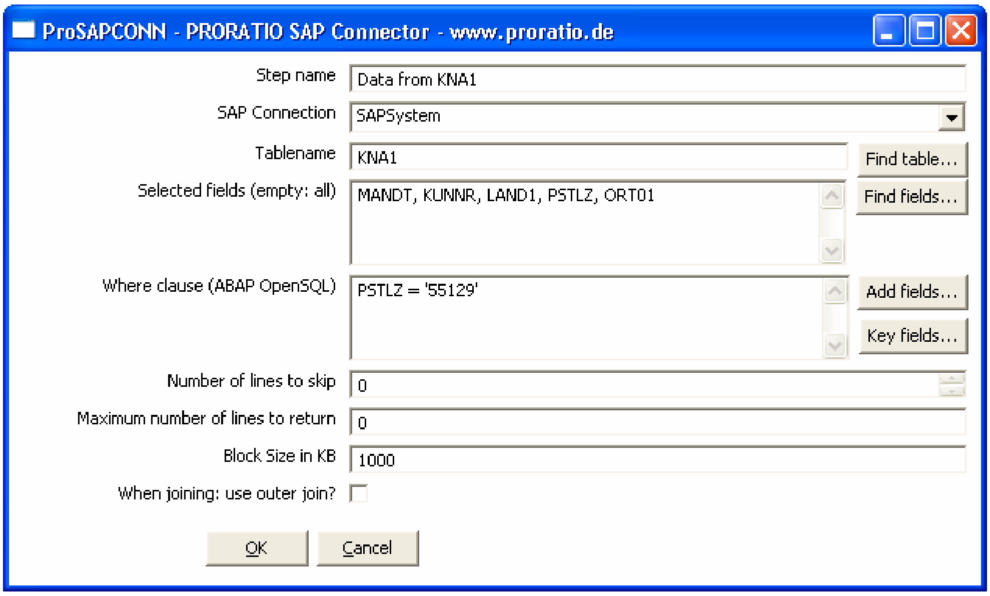
Language: EN

System Number: 00

SAP Client: 001

Language	select your language – mostly needed for error messages or as the default language for table or field descriptions
System Number	system number of your SAP® system
SAP Client	your dedicated SAP® client used internal for all table queries

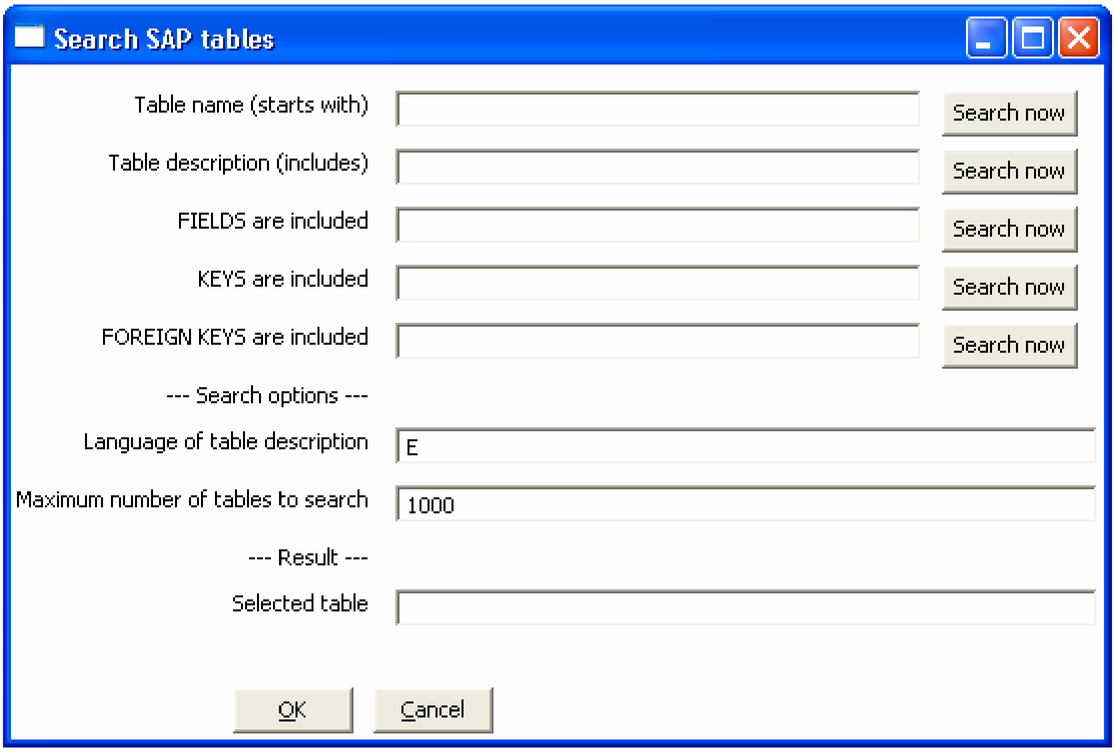
5.2 ProSAPCONN step properties



Step name: Data from KNA1
SAP Connection: SAPSystem
Tablename: KNA1
Selected fields (empty: all): MANDT, KUNNR, LAND1, PSTLZ, ORT01
Where clause (ABAP OpenSQL): PSTLZ = '55129'
Number of lines to skip: 0
Maximum number of lines to return: 0
Block Size in KB: 1000
When joining: use outer join? □

Step name	a unique step name of your choice
SAP Connection	select your defined SAP® connection
Tablename	select the SAP® table or view
Selected fields	enter the fields returned by this step separated by commas – needed spaces are internal inserted automatically
Where clause	enter the where clause in ABAP® OpenSQL syntax – also see chapter “Some examples of where clauses” When there is a preceding step, the join conditions are given here: Fields from the preceding step are enclosed in square brackets, e.g. “VBELN = [VBELN]” or another example “VBELN > [LastVBELN]”
Number of lines to skip	when reading data skip this number of lines
Maximum number of lines to return	when reading data stop after this maximum number of lines (when doing a lookup this could be 1)
Block size in KB	When reading large tables they are loaded in blocks and stored internal in memory. Giving a higher block size reduces the blocking overhead. On the other hand you need more memory and in worst case run out of memory. For getting more memory see the Java runtime options (e.g. set OPT=-Xmx256m).
When joining: use outer join?	When there is a preceding step and the join conditions are given in the where clause: Select if you want an outer or inner join. Outer join: if nothing is found, return at least one source row with the rest filled up with NULL. Inner join: do not return rows if nothing is found.

5.2.1 Tablename: “Find table” button



Search SAP tables

Table name (starts with)

Table description (includes)

FIELDS are included

KEYS are included

FOREIGN KEYS are included

--- Search options ---

Language of table description: E

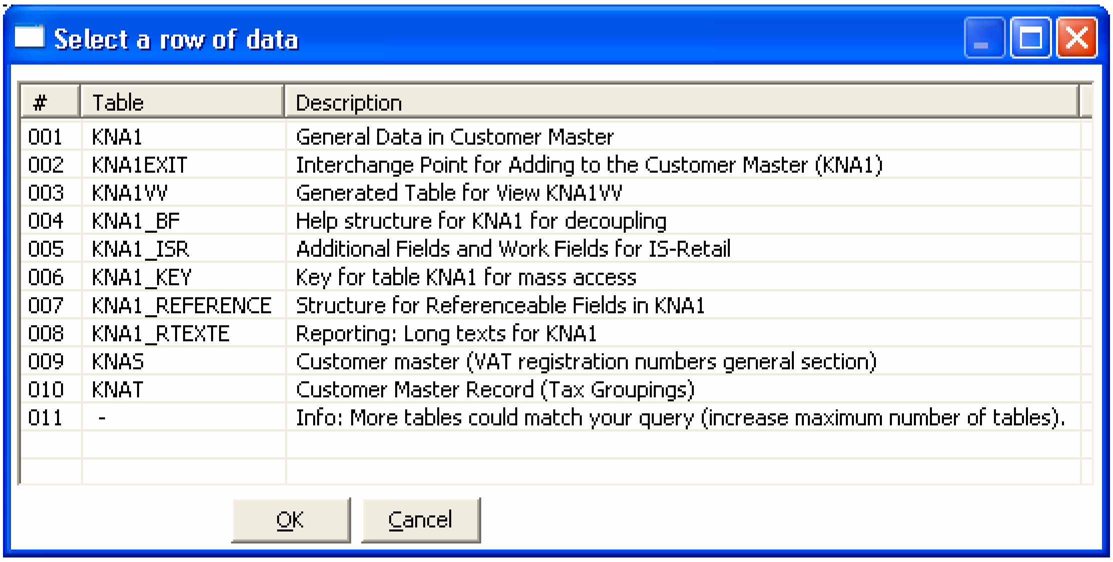
Maximum number of tables to search: 1000

--- Result ---

Selected table

Table name	search for tables beginning with
Table description	search for tables they include this description as a part
FIELDS are included	search for tables by the given fields – all fields separated by commas must be included in the table
KEYS are included	same as FIELDS but for KEYS
FOREIGN KEYS as included	same as FIELDS but for FOREIGN KEYS
Search options: Language	select the language (one letter) for the given descriptions – mind this language must be installed in your system
Search options: Maximum number of tables to search	Especially for searching the FIELDS or (FOREIGN)KEYS combination you can limit the size of tables to search. If ProSAPCONN finds more results a warning in the last line is given: “Info: More tables could match your query (increase maximum number of tables).”
Result: Selected table	This table will be transferred to the ProSAPCONN step as the selected table when pressing “OK”.

Example of finding tables beginning with KNA:

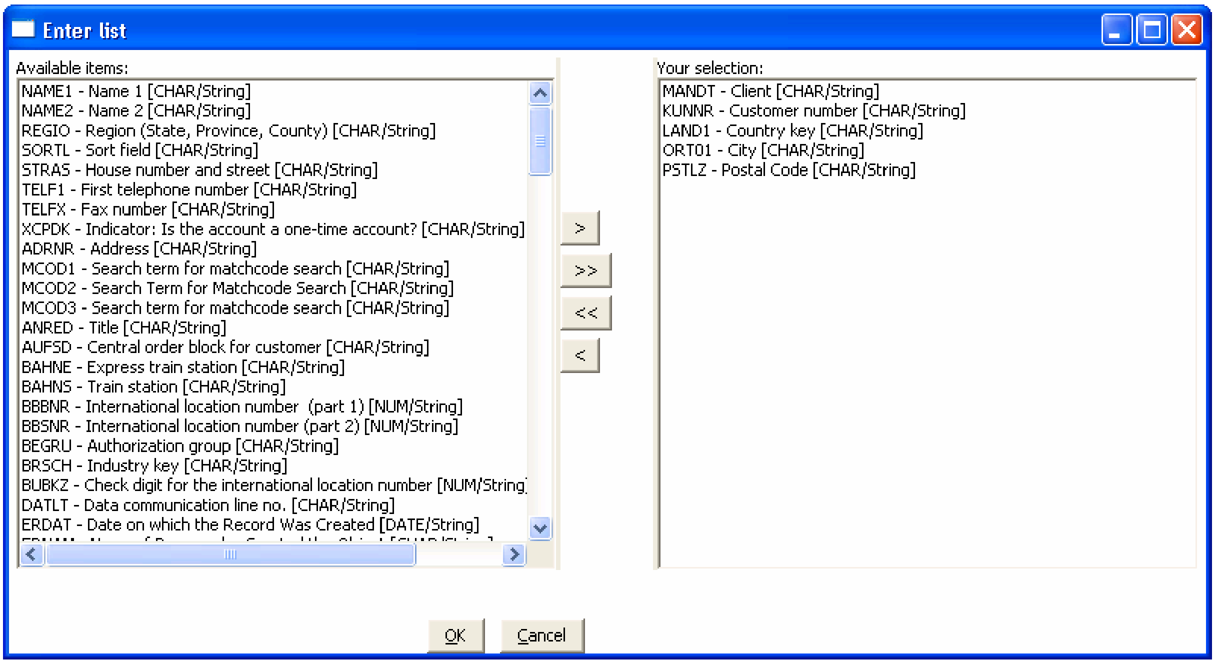


Select a row of data

#	Table	Description
001	KNA1	General Data in Customer Master
002	KNA1EXIT	Interchange Point for Adding to the Customer Master (KNA1)
003	KNA1VV	Generated Table for View KNA1VV
004	KNA1_BF	Help structure for KNA1 for decoupling
005	KNA1_ISR	Additional Fields and Work Fields for IS-Retail
006	KNA1_KEY	Key for table KNA1 for mass access
007	KNA1_REFERENCE	Structure for Referenceable Fields in KNA1
008	KNA1_RTEXTE	Reporting: Long texts for KNA1
009	KNAS	Customer master (VAT registration numbers general section)
010	KNAT	Customer Master Record (Tax Groupings)
011	-	Info: More tables could match your query (increase maximum number of tables).

5.2.2 Selected fields: “Find fields” button

Press this button for selecting the desired fields, example:



Enter list

Available items:

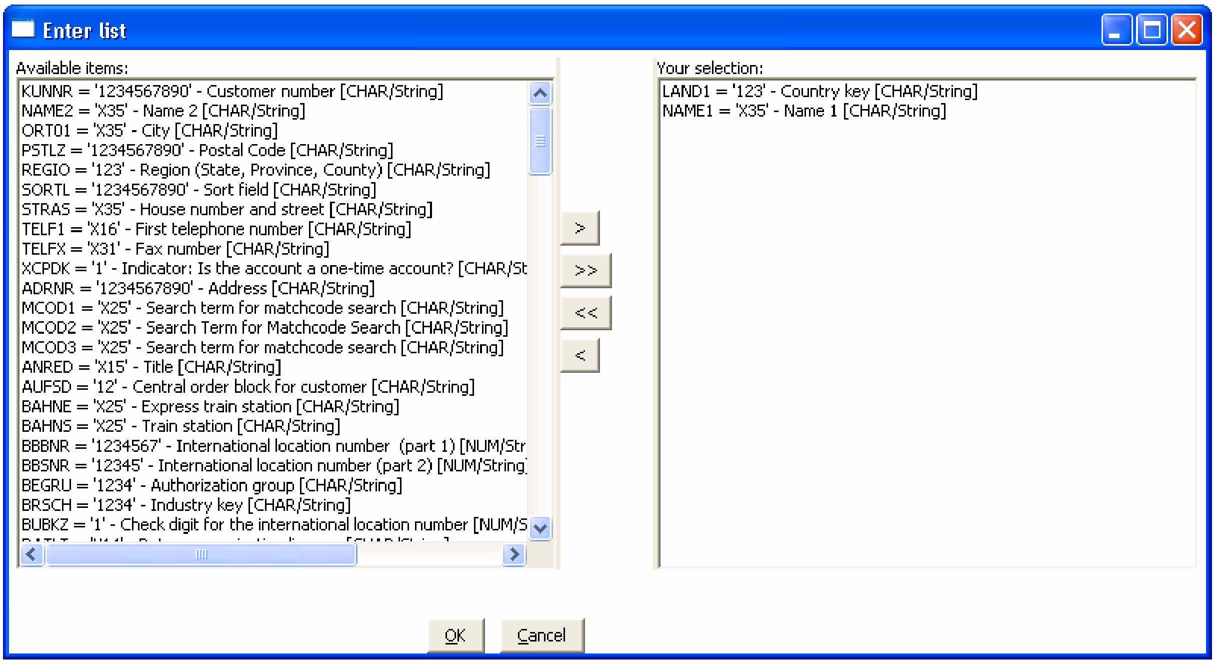
NAME1 - Name 1 [CHAR/String]
NAME2 - Name 2 [CHAR/String]
REGIO - Region (State, Province, County) [CHAR/String]
SORTL - Sort field [CHAR/String]
STRAS - House number and street [CHAR/String]
TELF1 - First telephone number [CHAR/String]
TELFX - Fax number [CHAR/String]
XCPDK - Indicator: Is the account a one-time account? [CHAR/String]
ADRNR - Address [CHAR/String]
MCOD1 - Search term for matchcode search [CHAR/String]
MCOD2 - Search Term for Matchcode Search [CHAR/String]
MCOD3 - Search term for matchcode search [CHAR/String]
ANRED - Title [CHAR/String]
AUFSD - Central order block for customer [CHAR/String]
BAHNE - Express train station [CHAR/String]
BAHNS - Train station [CHAR/String]
BBBNR - International location number (part 1) [NUM/String]
BBSNR - International location number (part 2) [NUM/String]
BEGRU - Authorization group [CHAR/String]
BRSCH - Industry key [CHAR/String]
BUBKZ - Check digit for the international location number [NUM/String]
DATLT - Data communication line no. [CHAR/String]
ERDAT - Date on which the Record Was Created [DATE/String]

Your selection:

MANDT - Client [CHAR/String]
KUNNR - Customer number [CHAR/String]
LAND1 - Country key [CHAR/String]
ORT01 - City [CHAR/String]
PSTLZ - Postal Code [CHAR/String]

5.2.3 Where clause: “Add fields” button

Use the “Add fields” button for selecting where fields. For the convenience the type and length of each field is given: If a field is numeric no quote signs are added. If a field is not numeric quote signs are added and the field length is given as a proposal – e.g. field size for LAND1 is 3, given as LAND1 = '123'. If the field size is bigger than 10 it is prefixed with X and the actual length is given, e.g. NAME1 = 'X35'. More than one field is automatically added with ADD, e.g. LAND1 = '123' AND NAME1 = 'X35'



Available items:

```
KUNNR = '1234567890' - Customer number [CHAR/String]
NAME2 = 'X35' - Name 2 [CHAR/String]
ORT01 = 'X35' - City [CHAR/String]
PSTLZ = '1234567890' - Postal Code [CHAR/String]
REGIO = '123' - Region (State, Province, County) [CHAR/String]
SORTL = '1234567890' - Sort field [CHAR/String]
STRAS = 'X35' - House number and street [CHAR/String]
TELF1 = 'X16' - First telephone number [CHAR/String]
TELFX = 'X31' - Fax number [CHAR/String]
XCPDK = '1' - Indicator: Is the account a one-time account? [CHAR/St
ADRNR = '1234567890' - Address [CHAR/String]
MCOD1 = 'X25' - Search term for matchcode search [CHAR/String]
MCOD2 = 'X25' - Search Term for Matchcode Search [CHAR/String]
MCOD3 = 'X25' - Search term for matchcode search [CHAR/String]
ANRED = 'X15' - Title [CHAR/String]
AUFSD = '12' - Central order block for customer [CHAR/String]
BAHNE = 'X25' - Express train station [CHAR/String]
BAHNS = 'X25' - Train station [CHAR/String]
BBBNR = '1234567' - International location number  (part 1) [NUM/Str
BBSNR = '12345' - International location number (part 2) [NUM/String]
BEGRU = '1234' - Authorization group [CHAR/String]
BRSCH = '1234' - Industry key [CHAR/String]
BUBKZ = '1' - Check digit for the international location number [NUM/S
```

Your selection:

```
LAND1 = '123' - Country key [CHAR/String]
NAME1 = 'X35' - Name 1 [CHAR/String]
```

5.2.4 Where clause: “Key fields” button

The “Key fields” button automatically inserts the needed Key fields except for key field MANDT (which is fixed by the user logon). As an example for table VBRP the key fields are POSNR = '123456' AND VBELN = '1234567890'.

If you use this step as a join/lookup step as described in chapter “Example: Combining data with customer data table KNA1” the relations are automatically determined and inserted – the join/lookup key is given in brackets. e.g. LAND1 = [LAND1]

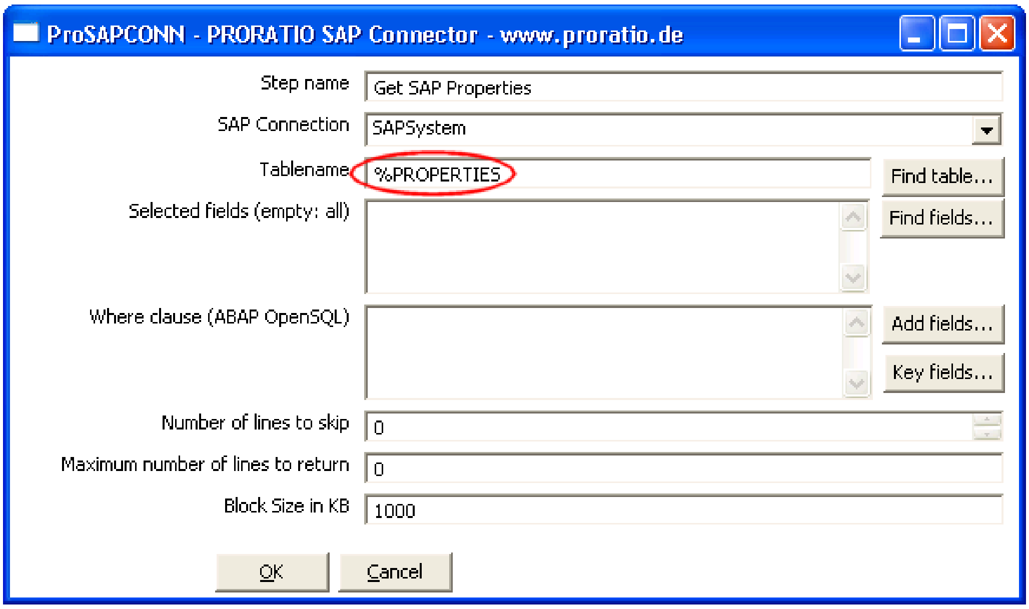
5.3 Conversion of data types

The following table lists the conversion from SAP® data types to Kettle types. The JCo data type is shown for completeness and is used internally. You can check the data types if you right click on a step and choose “Show output fields”. In the columns “Step origin” you will see the field description followed by [JCo data type/Kettle data type]. **For using data types marked with * you should read the “Customizing for Optimization Guide”.**

SAP® data type	JCo data type	Kettle data type
ACCP	DATE	STRING (length 8, YYYYMMDD)
CHAR	CHAR	STRING
CLNT	CHAR	STRING (length 3)
CUKY	CHAR	STRING (length 5)
CURR (*)	BCD	BIGNUMBER (see DEC, needs CUKY)
DATS	DATE	STRING (length 8, YYYYMMDD)
DEC (*)	BCD	BIGNUMBER (maximum length 31)
FLTP (*)	FLTP	NUMBER (significance 16 digits)
INT1	INT1	INTEGER (0 to 255)
INT2	INT2	INTEGER (-32767 to 32767)
INT4	INT	INTEGER (-2177483647 to 2177483647)
LANG	CHAR	STRING (length 1)
LCHAR	CHAR	STRING (not allowed in where clause)
LRAW (*)	BYTE	STRING (Hex values, not allowed in where clause)
NUMC	NUM	STRING (only digits, fixed size,'0' padded)
PREC	CHAR	STRING (length 2)
QUAN	BCD	BIGNUMBER (see DEC, needs UNIT)
RAW (*)	BYTE	STRING (Hex values, not allowed in where clause)
TIMS	TIME	STRING (length 6, HHMMSS)
UNIT	CHAR	STRING (length 2 or 3)
STRING	-	- not used in SAP® database tables -
RAWSTRING	-	- not used in SAP® database tables -
VARC	CHAR	STRING (not allowed in where clause)

5.4 Get Information about ProSAPCONN, JCO and the SAP® Environment

You can use a special table named "%PROPERTIES" in a ProSAPCONN step that retrieves information about ProSAPCONN and the SAP® Environment. This data (e.g. CON_USER or CON_SYSTEMID) can be used in subsequent steps or as a reference. You also see the installed and supported RFC_TABLE_READ on the SAP® system (e.g. ZRFC_READ_TABLE_PSC1024).


ProSAPCONN - PRORATIO SAP Connector - www.proratio.de

Step name: Get SAP Properties
SAP Connection: SAPSystem
Tablename: %PROPERTIES
Selected fields (empty: all):
Where clause (ABAP OpenSQL):
Number of lines to skip: 0
Maximum number of lines to return: 0
Block Size in KB: 1000

This step produces the following rows:

PropertyID	PropertyDescription	PropertyValue (Example)
PSC_KernelVersion	Version of the ProSAPCONN Kernel	2.1.0
PSC_Debug	ProSAPCONN debugging to console	false
RFC_512	Installed and supported RFC_TABLE_READ on the SAP® system	ZRFC_READ_TABLE_PSC512
RFC_1024	Installed and supported RFC_TABLE_READ on the SAP® system	ZRFC_READ_TABLE_PSC1024
JCO_Version	Version of the JCO-library	2.1.6 (2005-06-20)
JCO_jco.middleware.name	Name of the middleware implementation	sapjcorfc
JCO_jco.middleware.version	Version of the middleware	2.1.6 (2005-06-20)

	implementation	
JCO_jco.middleware.libjrfc_version	Version of the sapjcorfc library	2.1.6 (2005-06-20)
JCO_jco.middleware.libjrfc_path	Path to loaded sapjcorfc library	…\Kettle\libswt\win32\sapjcorfc.dll
JCO_jco.middleware.librfc_version	Path to loaded sapjcorfc library	
JCO_jco.middleware.librfc_numversion	Version of the RFC library as numerical string	640.0.81
JCO_jco.middleware.librfc_path	Path to loaded RFC library	
JCO_jco.middleware.librfc_sap_codepage	SAP_CODEPAGE environment variable	
JCO_jco.middleware.librfc_rfc_trace	RFC_TRACE environment variable	
JCO_jco.middleware.librfc_cpic_trace	CPIC_TRACE environment variable	
JCO_jco.middleware.snc_lib	Path to SNC library	SECUDE.dll
JCO_jco.middleware.wait_for_request_time	Time in seconds to wait incessantly for incoming requests	2
JCO_jco.middleware.max_startup_delay	Maximum server startup delay time in seconds	3600
JCO_jco.middleware.allow_start_of_programs	List of programs that are allowed to be started by the RFC library	
JCO_jco.middleware.monitoring	Turns on reporting performance data for each call	
CON_Client	Client	0
CON_CPICConversationID	Low-level CPIC conversion ID for this connection	6862779
CON_Destination	Destination	proratiosap
CON_Host	Host	proratiosap
CON_ISOLanguage	Logon language of the connection, two-byte character string	EN
CON_KernelRelease	Release of the remote SAP system's kernel	46D
CON_Language	Logon language of the connection, one-byte character string	E
CON_OwnBytesPerChar	Number of bytes per character for the currently used local codepage	1
CON_OwnCharset	Java charset equivalent of the local SAP® codepage	ISO8859_1
CON_OwnCodepage	Local codepage in SAP® notation	1100

CON_OwnEncoding	MIME encoding equivalent of the local SAP® codepage	ISO-8859-1
CON_PartnerBytesPerChar	Number of bytes per character for the currently used remote codepage	1
CON_PartnerCharset	Java charset equivalent of the codepage used by the remote system	ISO8859_1
CON_PartnerCodepage	Codepage used by the remote system in SAP® notation	1100
CON_PartnerEncoding	MIME encoding equivalent of the remote SAP® codepage	ISO-8859-1
CON_PartnerHost	Partner host, i.e. the name of the remote host	172.17.34.1
CON_PartnerRelease	Release of the remote SAP® system	46D
CON_PartnerType	Type which specifies the partner of the communication	3
CON_Release	Release of the local SAP® system or transport library	640
CON_RfcRole	Role of the connection	C
CON_SSOTicket	SSO Ticket	
CON_SystemID	System ID	SAPTST
CON_SystemNumber	System number	0
CON_Trace	Trace	false
CON_Type	Type which specifies the local program	E
CON_User	User ID the connection is associated with	USERNAME

6 History

6.1 Changes to ProSAPCONN

Version	Released	Description
2.1.1	2005-09-29	Added support for data types RAW and LRAW.
2.2.0	2006-01-26	1) Added support for inner and outer joins. 2) Changed the default language for table descriptions to 'E' (englisch). 3) Added internal version ID to ProSAPCONN steps when saving transformation (this is to prevent using ProSAPCONN steps with an outdated version not supporting newer properties).

6.2 Documentation changes

Relates to version	since	Document	Description
2.1.1	2005-09-29	Customizing for Optimization Guide	Added support for data types RAW and LRAW, see Chapter "Changes for special numerical values, data types RAW/LRAW"
2.1.1	2005-12-09	Quick Start and Reference	Some minor changes because Kettle is open source now
2.2.0	2006-01-30	Quick Start and Reference	Added support for inner and outer joins, see Chapters "Join Example: Joining billing header (VBRK) and item data (VBRP)", "More performance aspects on joins and lookups" and "ProSAPCONN step properties"