

How To Use Data Services I - Data Quality Made Easy

Applicable Releases:

SAP NetWeaver 7.0

IT Practice:

Business Information Management

IT Scenario:

Enterprise Data Warehousing

Version 1.0

November 2008

© Copyright 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Document Version	Description
1.00	First official release of this guide

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
< Example text >	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

- 1. **Business Scenario**..... 1
- 2. **Background Information**..... 1
 - 2.1 Introduction to Business Objects Data Services 1
 - 2.2 Which product to use for which scenario..... 3
 - 2.3 Data Services within the SAP BI staging process 4
- 3. **Prerequisites** 4
- 4. **Step-by-Step Procedure**..... 6
 - 4.1 Connecting SAP BI with Data Services 7
 - 4.2 Defining the SAP BI system as Datastore in Data Services 10
 - 4.3 Data Service Features 11
 - 4.3.1 Scenario 11
 - 4.3.2 Profiling 11
 - 4.3.3 Domain Value or Plausibility Check 14
 - 4.3.4 Pattern Matching 17
 - 4.3.5 String Matching 19
 - 4.4 Loading data from Data Services into SAP BI..... 21

1. Business Scenario

Data Quality - An ongoing concern.

Poor Data Quality can impact an organization in many ways. It can result in misguided marketing promotions being sent to the wrong address with incorrect information -- including improperly spelled names, title, phone numbers and company. Surveys revealed that up to 75% of wrong business decisions are made due to flawed data. In addition, companies spend up to 30% of their time verifying the accuracy and quality of the data they use to make decisions. Hence, investing in Data Quality does not only improve the quality of decision making, but also lowers significantly Total Cost of Ownership (TCO).

Growing information sources and volumes are accelerating the need for Data Integration and Quality.

Data Quality becomes an increasingly important topic for Enterprise Data Warehousing (EDW). Data for Reporting is retrieved from all types of sources, including SAP and Non-SAP sources. Some of the data, especially from Non-SAP sources, do need various ETL (Extraction Transformation and Loading) processing and Data Quality measures before they can be considered as trusted data.

Business Objects Data Services provides a broad set of tools in the area of ETL and Data Quality. Especially for Data Quality, Data Services goes way beyond the capabilities available in SAP BI. Hence, using it in conjunction with SAP BI does leverage the quality of the data in the enterprise enormously.

This document is one of a series of three How To papers which cover the usage of Business Objects Data Service within the SAP BI staging process. It introduces Data Services and its' tools, and describes some easy to use Data Quality features which provide an instant gain in your implementation. The further papers describe more complex scenarios which use Data Services features like *Address Cleansing*, *Matching* and a Close-Loop scenario for cleansing SAP BI data.

2. Background Information

2.1 Introduction to Business Objects Data Services

Business Objects Data Services provides tools for Data Integration and Data Quality within a single product. Formerly, Data Integrator and Data Quality were two different products, which have been merged in Data Services to a single one, joining forces for a best of breed product in both areas. Data Services establishes an intuitive development user interface (UI), administration environment and single runtime architecture. The architecture of Data Services is depicted in the underneath picture.

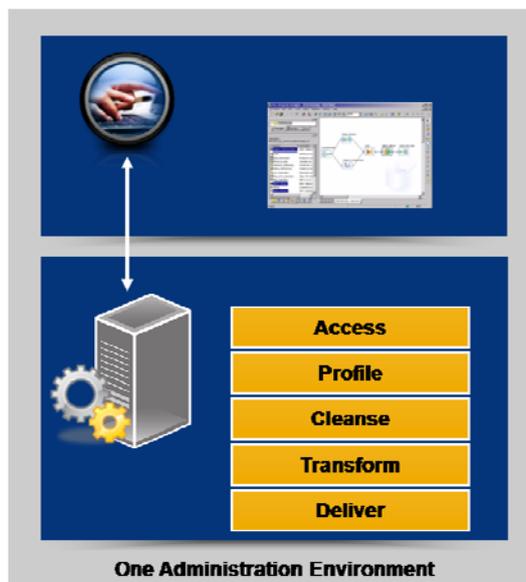


Figure: Data Services XI 3.0 Architecture

For the series of publication we will focus on the Data Quality perspective of Data Services. It comprises tools for Data Assessment (Profiling), Data Cleansing (Addresses and Data Cleansing), Enhancement, Matching and Consolidation and Continuous Monitoring. These tools build an iterative Data Quality circle for an overall approach.

1. **Data Assessment**

Data Assessment is the inspection, measurement, and analysis of data defects to help you understand the data and the impacts of its defects. Most importantly, it will allow you and the stakeholders to use that information to make sound decisions regarding your Data Quality approach. If you don't already have a Data Quality approach, the Data Assessment will provide the information you need to take action and also to prioritize the areas to target your approach.

The Profiling feature in Data Services allows detecting patterns in the data at hand, which columns contain potentially invalid values, and how often certain values occur in particular columns. Profiling can be executed on a single or multiple objects.

2. **Data Cleansing**

This tool parses, cleanses and standardizes data such as names/addresses, emails, phone numbers, Social Security Numbers, and dates into individual components. It manages international data for over 190 countries and reads and writes Unicode data. It improves integrity of data to identify matches and ultimately create a single customer view.

The *Data Cleanse* Transform uses rule-based parsing, identifying and isolating specific parts of mixed data, and standardizes your data based on information stored in the parsing dictionary, business rules defined in the rule file, and expressions defined in a pattern file.

You can use Data Cleanse to assign gender codes and prenames, split records with dual names into individual records, create personalized greetings, and generate standards used in the match process Data Cleanse can also parse and manipulate various forms of international data, as well as operational and product data.

3. **Enhancement**

Completes records with directory information by appending name, address, phone number, or email address. It also provides geo-coding capabilities for geographic and demographic marketing initiatives.

4. Matching and Consolidation

The Matching and Consolidation component of Data Services solution matches and consolidates data elements based on user-defined business rules. Duplicate records can be identified and eliminated. All information of each individual customer or an entire corporation or household is consolidated, hence only the unique records are saved to the database. Since the matching logic is governed by a set of match rules that can be customized to implement a user-centric solution, you are not forced to adopt pre-established rules. This provides a flexibility in determining what is a 'match' and what is 'no match' in the database.

5. Continuous Monitoring

The final factor in a data quality strategy is continuous monitoring. Adhering to the principals of Total Quality Management (TQM), continuous monitoring is measuring, analyzing, and then improving a system in a continuous manner. Continuous monitoring is crucial for the effective use of data, as data will immediately age after capture, and future capture processes can generate errors.

This feature allows to monitor and measuring the activities within Data Services at any time. It can be customized for periodic assessment. Thresholds can be set, and notification can be send on exceeding the defined thresholds. There are also various reports available to analyze and track the Data Quality process.

2.2 Which product to use for which scenario

Understanding the basic principles of Data Services is one side of the coin. Exploring its' potential in the SAP BI environment requires also an assessment in which area and for which process Data Services is most suitable.

The underneath table tries to provide an overview / decision support about which product can be used in a given scenario. It lists scenarios and whether or not SAP BI or Data Services can fulfill the specific requirement

Scenario	SAP NetWeaver BI	Data Services	Comment
Referential Integrity	X	Partially	With Data Services by modeled dataflow
Plausibility Check	Partially	X	In SAP BI by programming
Pattern matching	-	X	e.g. validating phone numbers
Lookups	X	X	In SAP BI mostly programming
BI Master Data attribute derivation	X	-	
Profiling	-	X	
Address cleansing	-	X	
Data Cleansing	-	X	
Matching	-	X	
Extraction of SAP data	X	X	SAP BI superior

Extraction of Non-SAP data	X	X	Data Services superior
Formula support	X	X	
Custom routines and functions	ABAP	X	
Auditing	-	X	No standard approach in SAP BI

2.3 Data Services within the SAP BI staging process

How can Data Services leverage the SAP BI staging process? As described in the previous section, there are many areas where Data Services offers tools not available in SAP BI or are superior to the existing tools in SAP BI. This provides an instant gain in Data Quality measures while lowering the TCO due to lower efforts to implement certain scenarios.

Data Services tools can be incorporated into the SAP BI staging process, especially for Non-SAP data, where the enhanced integration capabilities of Data Services might be needed anyway. The Data Integration capabilities of Data Services allow the sourcing and processing of almost any data source – covering structured and unstructured data as well as dedicated applications.

The procedure for incorporating Data Services features into the SAP BI staging process is the following:

1. Take a set of data (preferable Non-SAP data), and load it in the Data Services engine.
2. Perform Data Quality measures to the data in Data Services.
3. Load the data into SAP BI

3. Prerequisites

- Software
 - Business Objects Data Services XI 3.0

 **Note**

The information provided should also be applicable to prior or future releases of Data Services, though with changes in the realization.

- Hardware
 - No particular hardware needed (apart from the standard specifications for SAP NetWeaver BI and Data Services)

Provide information about:

- Relevant SAP Notes
 - None
- Additional background/starting documentation (also provide a link)

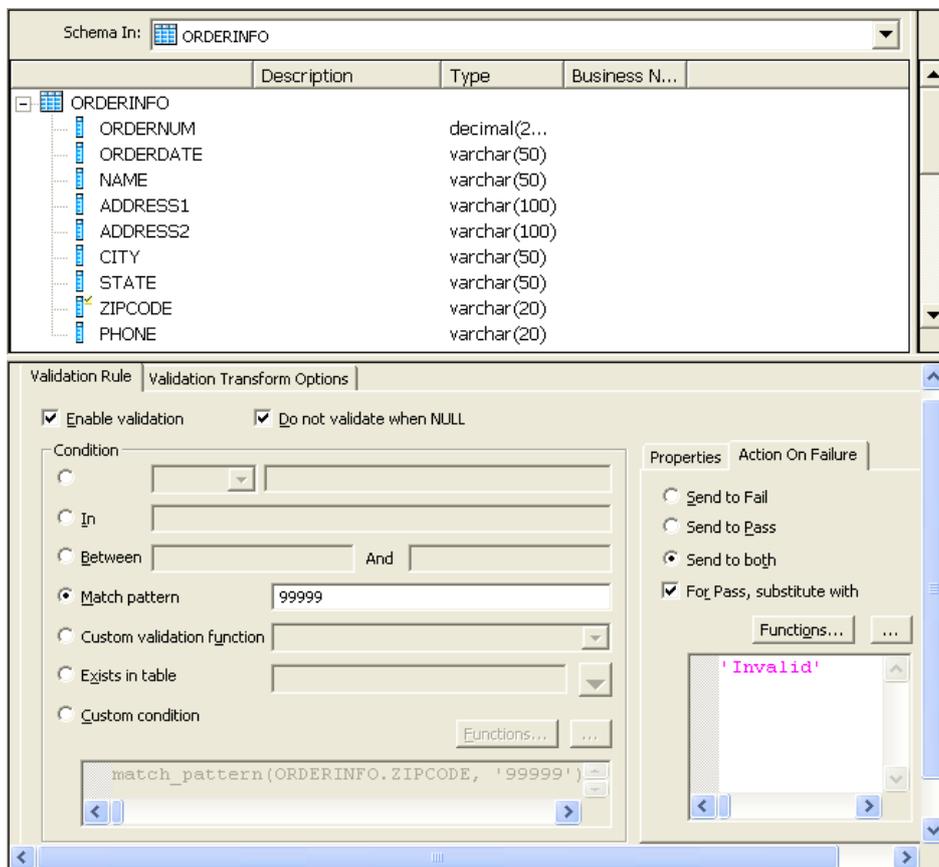
- For SAP BI - help.sap.com (SAP Solutions → SAP NetWeaver → SAP NetWeaver 7.0 → Functional View → SAP NetWeaver by Key Capabilities → Information Integration → Business Intelligence)
- For Data Services - help.sap.com (Business Objects → choose product and release)
 - Data Services Designer Guide
 - Data Services Reference Guide
- Additional information and implementation assistance can be found in the various communities like SDN (SAP NetWeaver BI and Business Objects products) and DIAMOND (Business Objects technical community)
- Required/recommended expertise or prior knowledge
 - SAP BI – Intermediate EDW knowledge
 - Data Services - Basic knowledge

4. Step-by-Step Procedure

This paper is focusing on simple Data Quality measures. Most of them can be achieved with one specific Transform in the Data Services. We are mainly using the *Validation* Transform. For some validation tasks, like string patterns, it might be possible to use the *Query* Transform. In addition, the *Profiling* functionality in Data Services is also introduced.

The *Validation* Transform qualifies a data set based on rules for input schema columns. It allows one validation rule per column and it filters out or replaces data that fails the criteria.

Data output / Result - The *Validation* Transform can output up to two data outputs: *Pass* and *Fail*. Data outputs are based on the validation condition that you specify in the *Validation* Transform. You set the data outputs when you connect the output of the *Validation* Transform with a *Pass* object, a *Fail* object, or both a *Pass and Fail* object in the workspace.



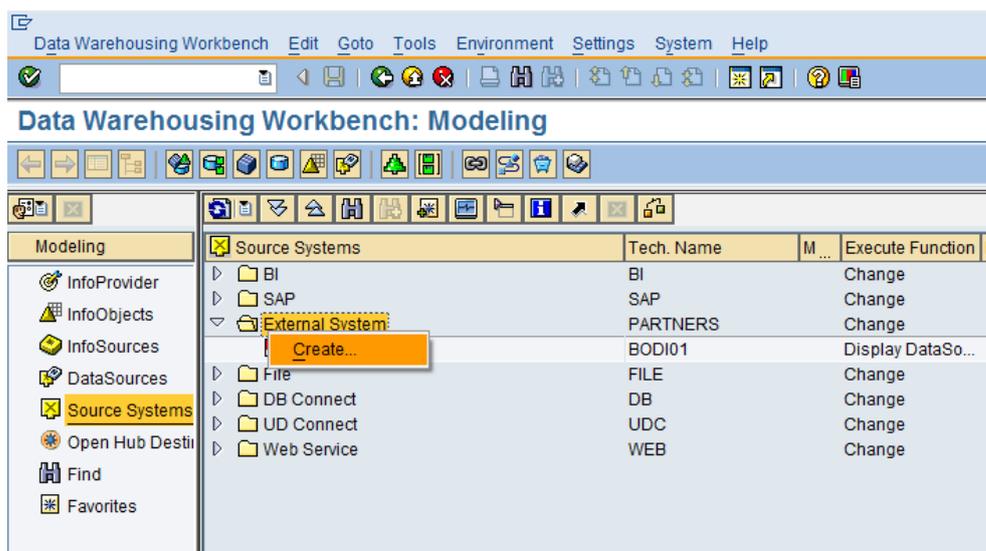
The following main validation conditions can be defined:

Option	Description
Enable Validation	Enable validation. Or deselect to disable validation.
Condition	Set conditions on the validation rule.
In / Between	Allows you to specify a range or set of values for a column.
Match Pattern	Use a pattern like for phone number, e.g. (9).99.99.99.
Customer Validation Functions	Use custom function from a list for validation purposes.
Exists in table	Select to specify that a column's value must exist in another table's (Datastore) column.
Custom Condition	Create more complex expressions by using the the Function selection option or smart editors.

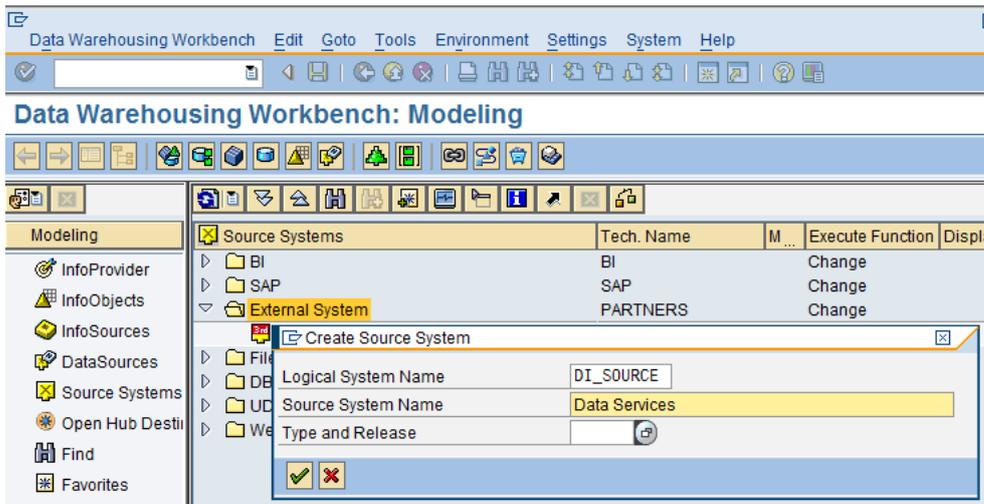
4.1 Connecting SAP BI with Data Services

In order to allow the exchange of data and meta data between SAP BI and Data Services, we have first to establish a connection between both systems. On the SAP BI side this is achieved by creating a 3rd party Source System, whereas in Data Services we have to create new Datastore.

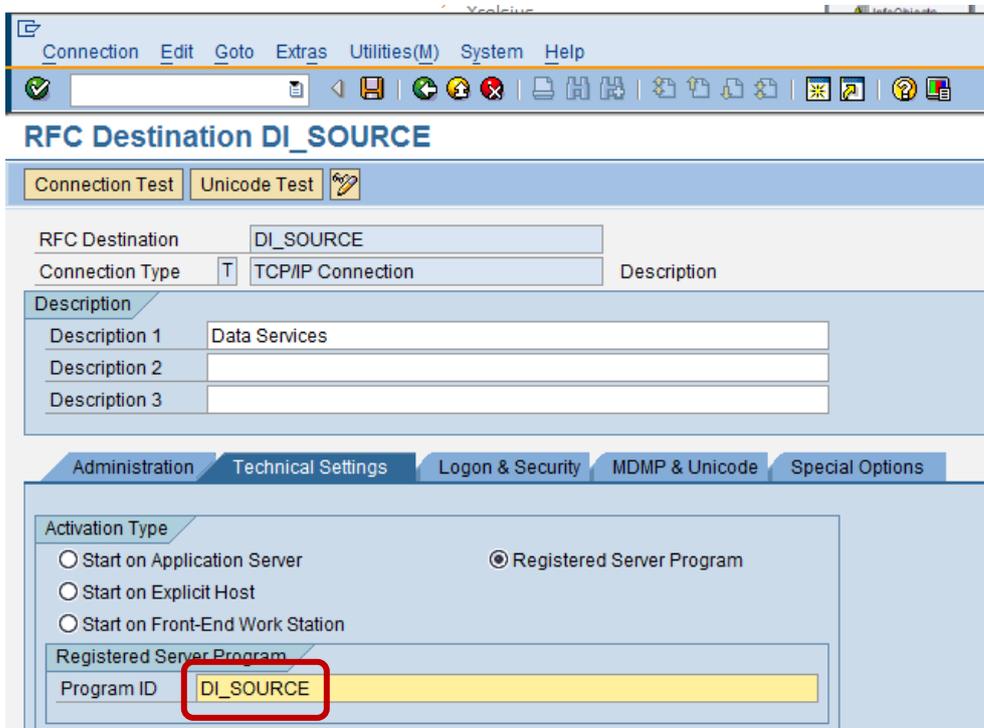
1. Start the Data Warehousing Workbench (RSA1) and select in the navigation area the option *Modeling*. Pick the entry *Source Systems*.
2. Position the cursor on the *External System* folder, and choose *Create* from the context menu.



- Type in a name and description for the Source System.

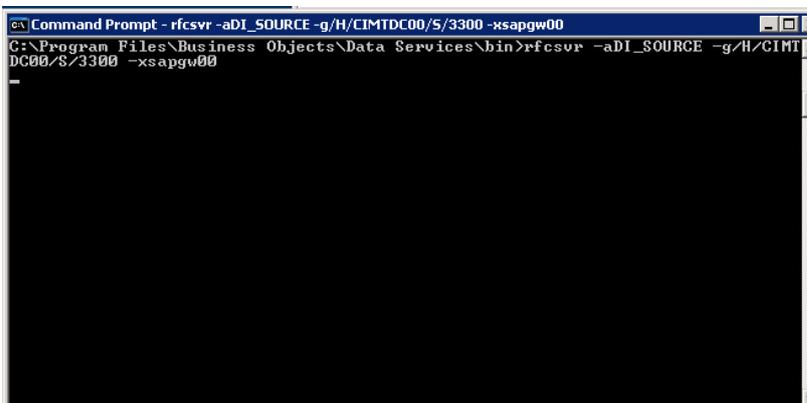


- Data Services will start an RFC Server program and indicates to SAP BI that it is ready to receive RFC calls. To identify itself as the RFC Server representing this SAP BI Source System a keyword is exchanged, in the screen shot below it is "DI_SOURCE". This is the Registered Server Program, the Data Services RFC Server will register itself with at SAP. Therefore, provide the same Program ID that you want to use for the call of the RFC Server on Data Services side.
All other settings for the Source System can remain on the default settings. To complete the definition of the Source System, save it.

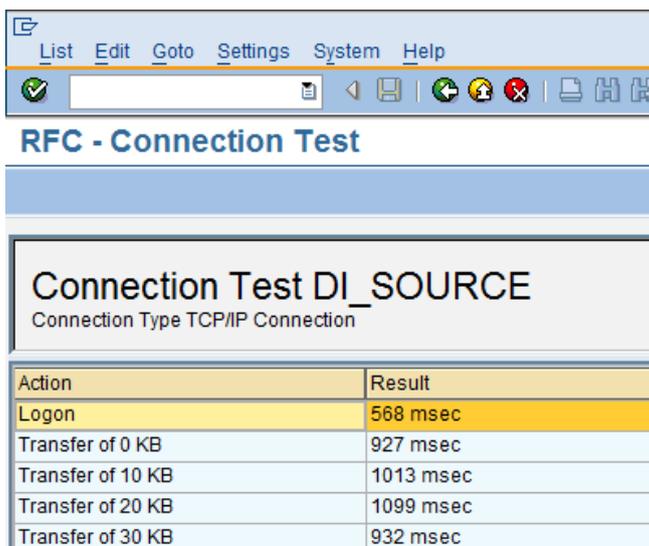


- To test the connection and for the loading of data from Data Services to the SAP BI system, we have first to start the RFC Server on the Data Services side.

- a. Open a MS-DOS Command Prompt window on the client where Data Services is running.
- b. Change to the binary folder of the Data Services program (in our case C:\Program Files\Business Objects\Data Services\bin)
- c. Start the RFC Server program with the following parameters (rfcsvr -aXX -g/H/YY/S/33ZZ -xsapgwZZ). It should not return. If you experience errors, you might want to check the file 'dev_rfc.trc' with the notepad to find a detailed description (a typo is one of the most occurring errors).
 - i. XX = name of the Registered Server Program, in our example DI_SOURCE
 - ii. YY = host name SAP BI message server, in our example cimtdc00
 - iii. ZZ = system number of the SAP BI system, in our example 00 (for details see also the entry for the SAP BI system in the SAP Logon)



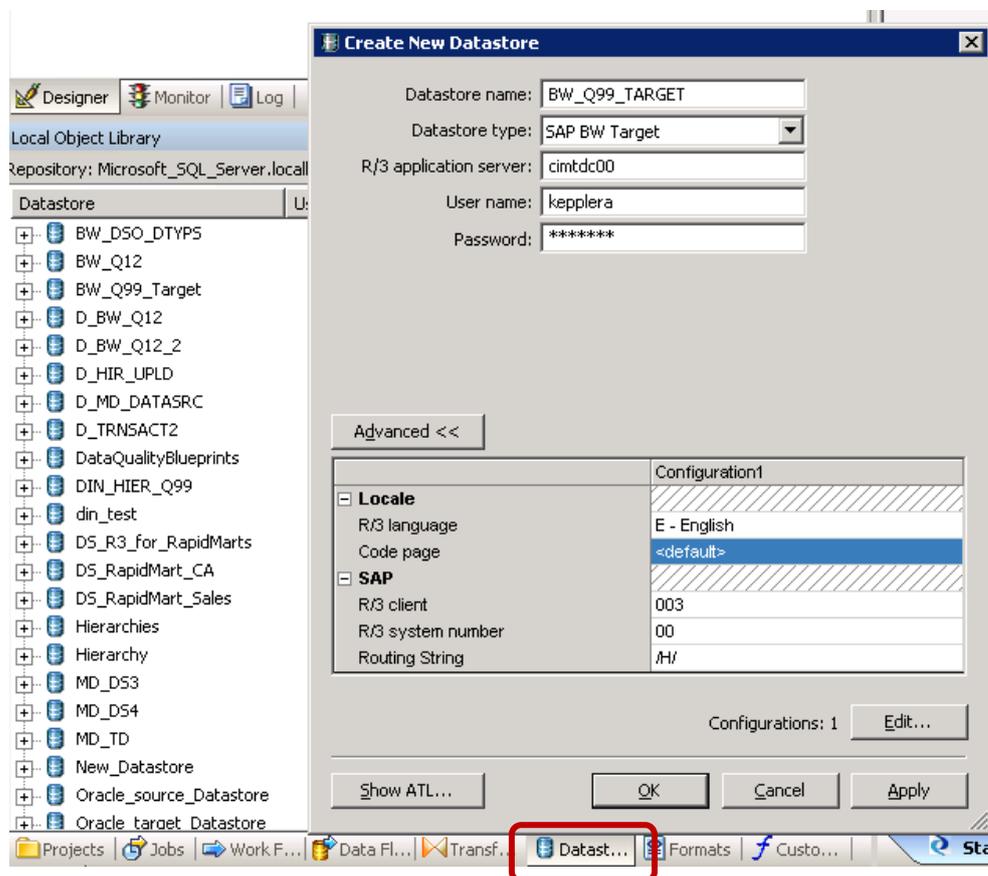
6. After starting the RFC Server, you can test the RFC connection from SAP BI to Data Services. The result should look similar to the underneath picture



4.2 Defining the SAP BI system as Datastore in Data Services

1. Open Data Services
2. Open the tab strip with the Datastores.
3. Position the cursor on the header line, and choose *New* from the context menu.
4. Enter a Datastore Name, application server (as in the previous chapter), a user name and a password for the SAP BI system. Choose also an appropriate Datastore Type (SAP BW Source or SAP BW Target), depending on whether you want to extract data from or load data into a SAP BI system.

Open the *Advanced* dialog, and enter an appropriate client and system number. As Routing string, it was sufficient in your example to use '/H/' (if applicable, you have to specify a router).



5. After clicking OK, you have defined a new Datastore, which you can use in a DataFlow of Data Services (after importing the required structures).

4.3 Data Service Features

4.3.1 Scenario

To show the potential of the Data Services features we are using the following scenario.

We have an EXCEL file with customer details from a Non-SAP system. The contents is sketched in the underneath screenshot.



	A	B	C	D	E	F	G	H	I	J	K
1	KUNNR	NAME1	NAME2	ORT01	PSTLZ	COUN	REG	SORTL	STRAS	TELF1	TELFX
2		2 Wett		Walldorf	69190	DE	8	WETT	Astorstrasse 34		
3		3 Wettler		Walldorf	69190	TK	8	WETT	Astorstrasse 34		
4		110 Auto Klement	Exclusive Automot	München	81737	TK	9	AM	Bert-Brecht-Allee 29	089/93534	089/93530
5		999 Ship-to location #1		Frankfurt	60329	DE	6	SHIP	Kaiserstrasse 59		
6		1000 Becker Berlin		Berlin	13467	DE	11	BECKER	Calvinstrasse 36	030-8853-	030-8853-999
7		1001 Lampen-Markt GmbH		Frankfurt	65936	DE	6	LAMPEN	Auf der Schanz 54	069-46765	069-4676521
8		1002 Omega Soft-Hardware Markt		Nuernberg	90455	DE	9	OMEGA	Gustav-Jung-Strasse 425	09256-454	09256-45485
9		1003 IDES AG Tochtergesellschaft		Mannheim	68219	DE	8		Karlsruher Strasse 20	0621/7700-0	
10		1005 CPG Europa		Offenbach	63067	DE		RFID	Kaiserstrasse		
11		1007 Pharma AG		Frankfurt	60311	DE		RFID	Koenig Strasse		
12		1010 Becker Berlin (Versand)		Berlin	13505	DE	11	BECKER	Beatestrasse 4	030-83732	030-9847658
13		1012 Autohaus Franz GmbH	SAPSOTA Haupth Muenchen		80939	DE	9	AM	Schwarzhauptstrasse 51	089/35467	089/3546722

Since the data does not originate from an SAP system (where the maintenance transaction would provide a high data quality), we need measures to check the data prior to uploading it to the SAP BI system. To guarantee a high quality of the data and a most efficient upload into the SAP BI system, we are using the following features of Data Services to check / improve the Data Quality

- **Profiling**
To analyze the existing data set and define a Data Quality strategy
- **Domain Value or Plausibility Check**
To check the validity of a specific value, e.g. date ranges, certain countries or the instances used to defined the gender of a person.
- **Pattern Matching**
To check columns in the data set on the existence of a specific structure, e.g. phone numbers, dates or times
- **String Matching**
To find the occurrence of a string (with or without wildcards) within a set of records

4.3.2 Profiling

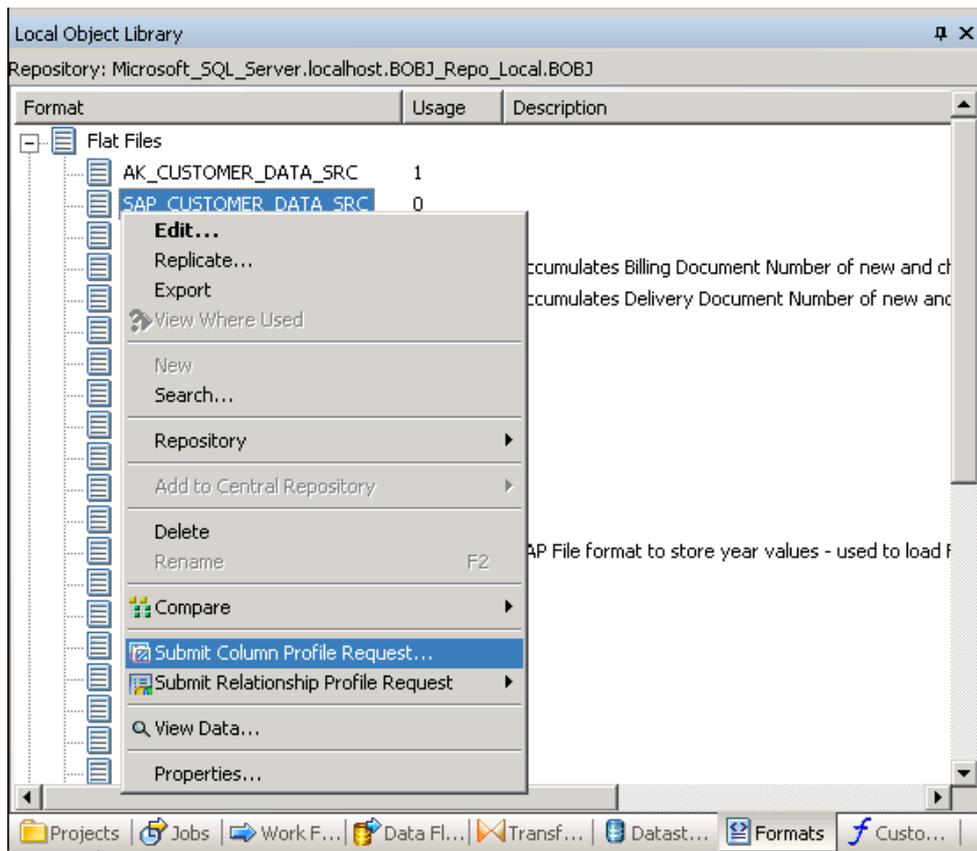
Profiling provides you with a first option to analyze the structure and quality of your data. It assists in defining the required Data Quality measures and shows immediately serious defects in the overall data set.

Profiling can be performed on tables and flat files. Basic Profiling provides information about Min, Min Count, Max, Max Count, Average, Min string length, Max string length, Average string length, Nulls, Nulls%, Blank, Blank %, Zero, and Zero %. Detailed Profiling adds Median, Median String Length, Distincts, Distincts %, Patterns, and Pattern %. Profiling can be executed on a single or multiple objects.

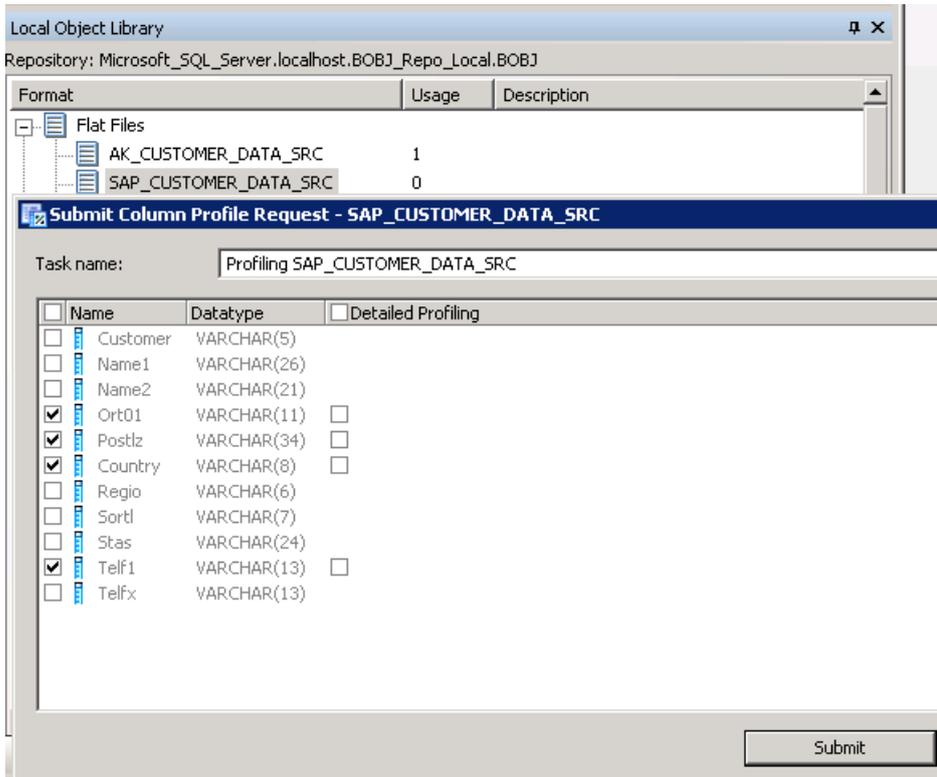
Before you can execute Profiling, you have to log into the Profile Server. You do so, by choosing the menu entry *Tools* → *Profile Server Login*. Provide your user credentials for the Profile Server.

In our example we are profiling a small sample flat file.

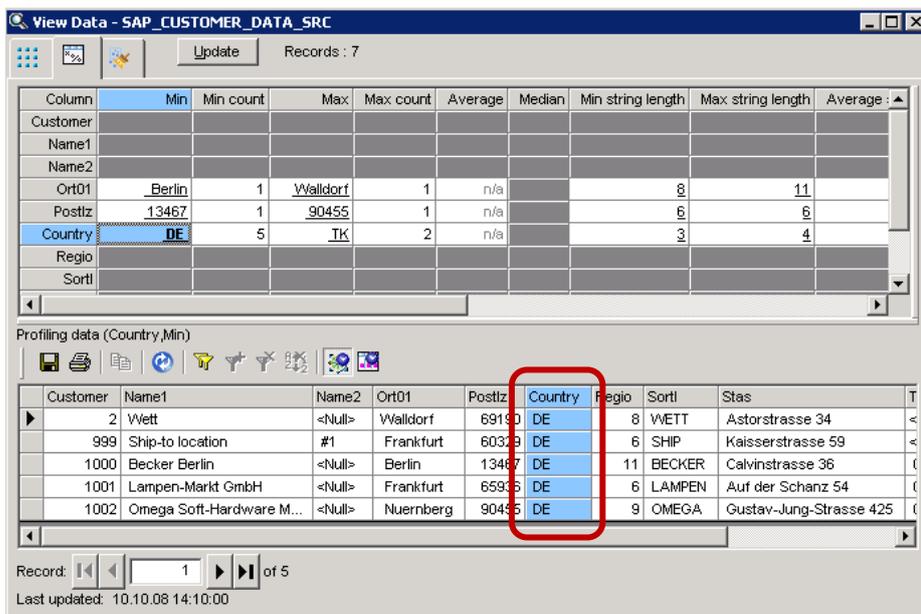
1. Pick the *Formats* tab strip.
2. Position the cursor on a file, and choose *Submit Column Profile Request* from the context menu. Alternatively, the same profiling definition dialog can be called from the *View Data* display. To do so, use the *Update* button on the second tab strip of the pop-up. On the first execution the underneath described profiling definition dialog is called, whereas on any subsequent execution the profiling result is updated.



- Pick the columns you want to analyze, and press the submit button.



- To view the result of the Profiling, use the *View Data* feature, which is available everywhere the flat file is used on the context menu to it. The second tab strip on the pop-up displayed shows the Profiling result. To drill-down to detailed information, position on the underlined values. It shows the respective records containing a specific value. On the screen shot underneath, we have drilled-down on the detail for the Country 'DE'.

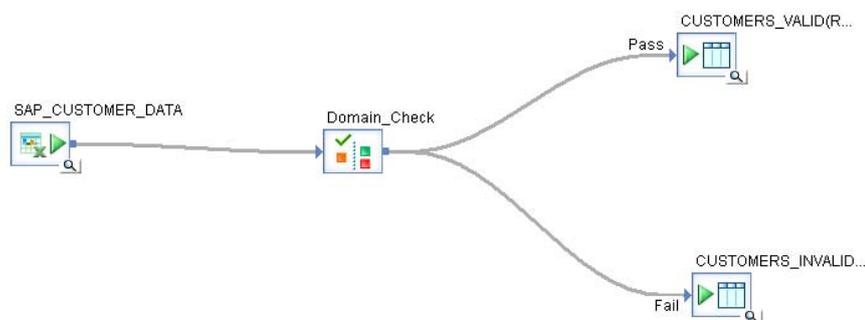


4.3.3 Domain Value or Plausibility Check

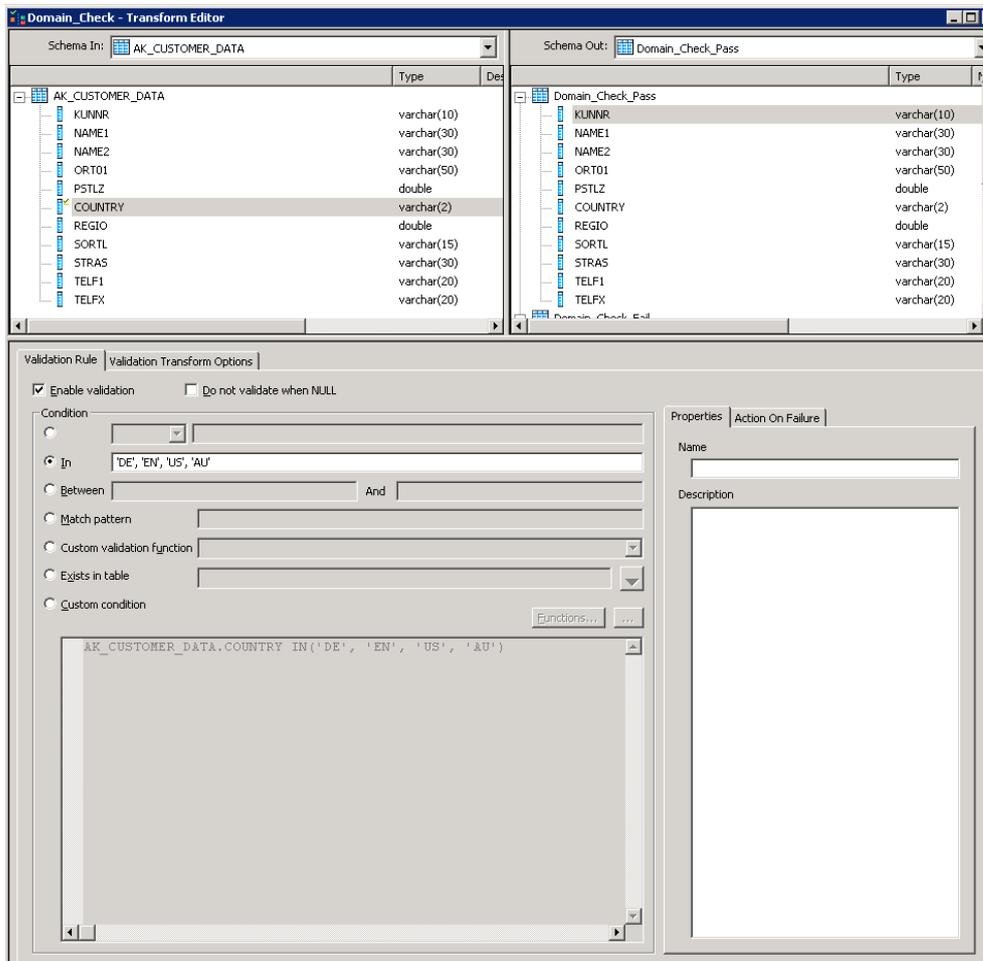
To start with an easy example, we are first checking the values for a particular field / column. The purpose is to filter out values that are not in a given / allowed set of values. In our example, we are checking whether the *Country* field contains only values for USA, Germany, England and Australia. These records should be passed on as valid records for further processing, whereas the invalid records should be written to a separate target to allow for further analysis and correction of the data.

Other alternative checks could be on the gender of a person, the range of postal codes, valid date ranges, etc.

Our sample data set contains 2 records which do not fulfill the above stated condition for the *Country* field. These records should be written to a specific container (template table *CUSTOMERS_INVALID*).

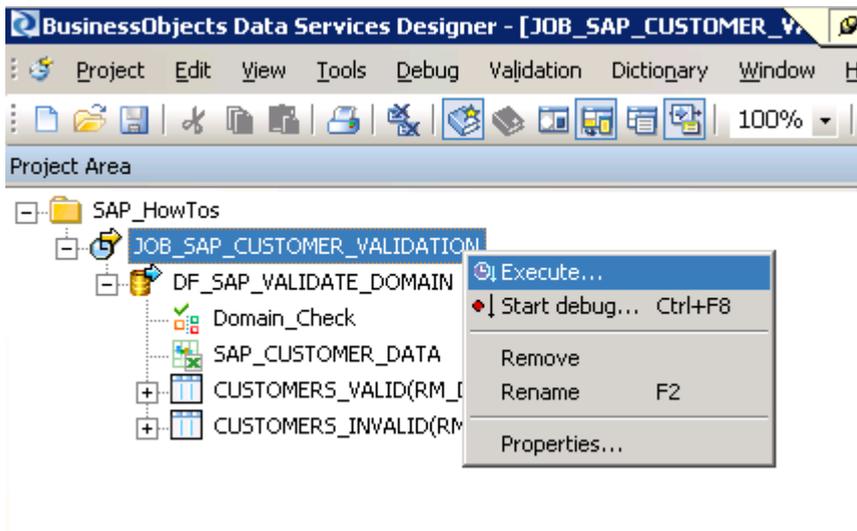


1. Open Data Services and define a Job and a DataFlow.
2. Within the DataFlow drag your data source to the canvas. In our example, we are using a flat file.
3. Choose the *Transform* tab strip, and pick from the *Platform* folder the *Validation* Transform. Drag it to the canvas.
4. Choose the targets for the valid and invalid records. This can be a table, a file or a template table.
5. Connect the data source, the *Validation* Transform and the targets.
6. Double-Click on the *Validation* Transform (in our case titled *Domain_Check*).
7. Mark the *COUNTRY* field, and define the following parameters for the *Validation Rule*.
 - a. Switch on the check button *Enable-validation*
 - b. Choose the Condition *In* and enter the strings 'DE', 'EN', 'US' and 'AU' as valid values for the *COUNTRY* field

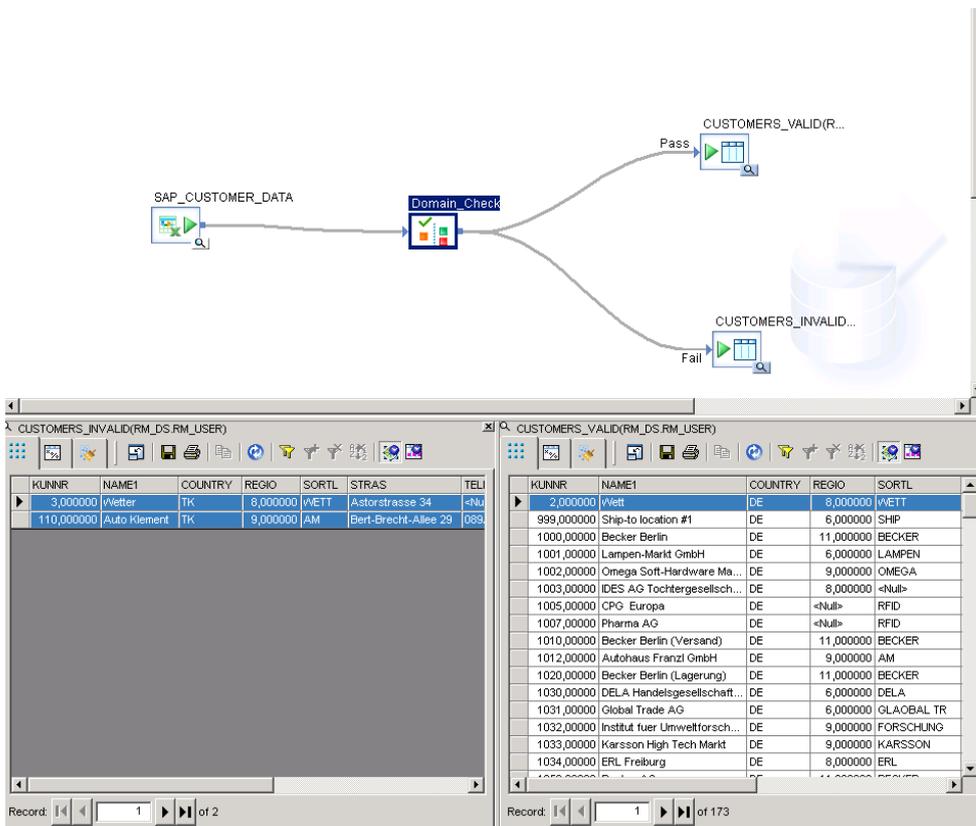


8. Add additional validations or close the window.
9. Validate your DataFlow by using either the menu entry *Validation → Validate → Current View / All objects in view* or use the respective icons ( )

- If your Job and DataFlow is correct, you can save and execute it. Position the cursor on the Job name, and select from the context menu *Execute*.



- After the job has completed successfully, you can check the result of the validation by returning to the DataFlow display. After using the *View Data* option for the valid and invalid targets, we detect that 2 records do not fulfill our condition (left hand side of the screen shot underneath), namely the *Country* field contains the value 'TK'.



4.3.4 Pattern Matching

In order to guarantee that the data has a certain structure (pattern), you can use the pattern matching of the *Validation* Transform. It allows you to check the pattern of phone numbers, dates, times and numbers.

For our customer data example, we are checking the pattern for the provided telephone number. We are only accepting phone number starting with the character '0', and separating the area code from the actual phone number with either a '/' or a '-' character. Hence we define the pattern '0*99[-/]99*' for the field *TELF1*. For a detailed description on how to define a pattern, and all the allowed objects in a pattern, please refer to the reference guide and the function *match_pattern*.

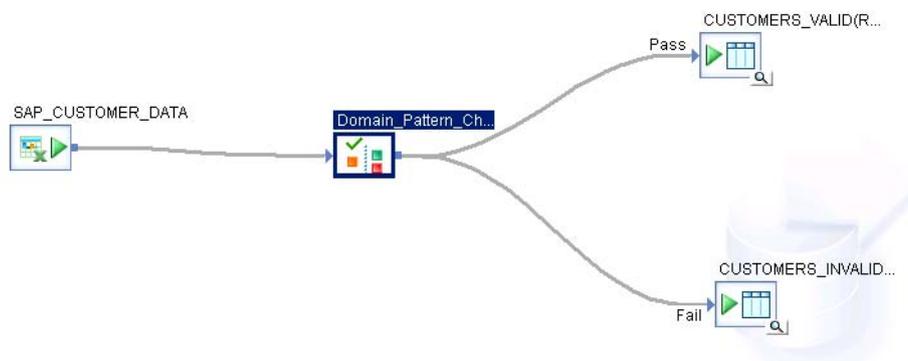


Tip

If you want to realize even more complex Pattern Matching, you can use the function *match_regex* in a custom defined condition.

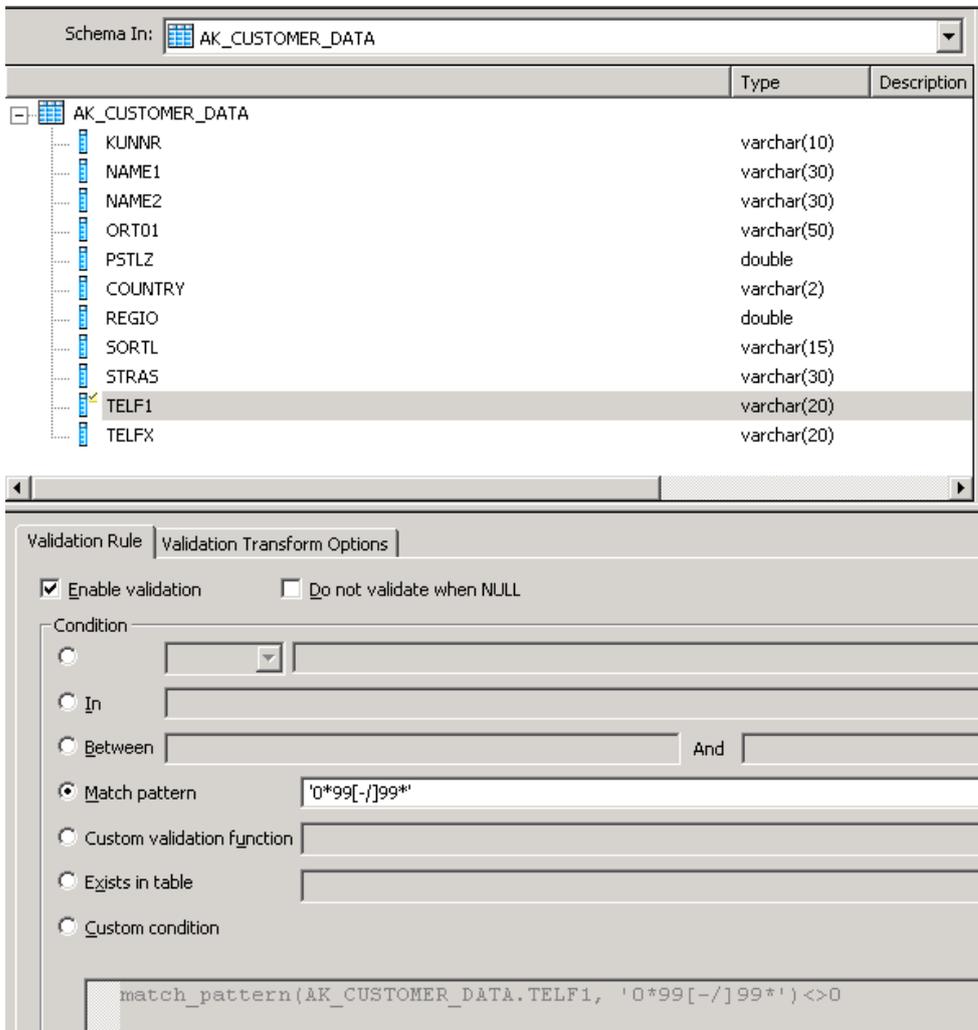
Since we are checking on another field as in the domain check, yet the same data, we are incorporating the pattern matching check in the same *Validation* Transform we are already using for the domain check.

1. Open the workflow, you have created in the previous section, and rename the *Validation* Transform for the domain value check to reflect also the pattern check.



2. Double-Click on the *Validation* Transform (now titled *Domain_Pattern_Check*).

3. Mark the field *TELF1*, and define the following parameters for the *Validation Rule*.
 - a. Switch on the check button *Enable-validation*
 - b. Choose the *Match pattern* and enter the string *'0*99[-/]99*'* as valid pattern for the *TELF1* field



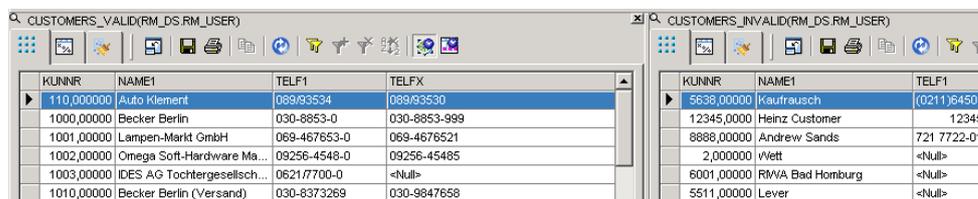
The screenshot shows the SAP Data Services interface. The top part displays a tree view of the schema *AK_CUSTOMER_DATA* with fields: *KUNNR* (varchar(10)), *NAME1* (varchar(30)), *NAME2* (varchar(30)), *ORT01* (varchar(50)), *PSTLZ* (double), *COUNTRY* (varchar(2)), *REGIO* (double), *SORTL* (varchar(15)), *STRAS* (varchar(30)), *TELF1* (varchar(20)), and *TELFX* (varchar(20)). The *TELF1* field is selected.

The bottom part shows the *Validation Rule* configuration window. The *Enable validation* checkbox is checked. The *Match pattern* radio button is selected, and the pattern *'0*99[-/]99*'* is entered in the text field. The *Validation Transform Options* tab is active, and the following SQL expression is shown in the bottom text area:

```
match_pattern(AK_CUSTOMER_DATA.TELF1, '0*99[-/]99*') <> 0
```

4. Add additional validations or close the window.
5. Validate your DataFlow by using either the menu entry *Validation* → *Validate* → *Current View / All objects in view* or use the respective icons ( )
6. If your Job and DataFlow is correct, you can save and execute it. Position the cursor on the Job name, and select from the context menu *Execute*.

- After the job has completed successfully, you can check the result of the validation by returning to the DataFlow display. By using the *View Data* option for the valid and invalid targets, you can display which records satisfy the pattern / structure validated in the Transform. In our example all customers without a phone number or wrong format have been written to the invalid container (right hand side of the underneath screen shot).



KUNNR	NAME1	TELF1	TELF2
110,00000	Auto Klement	089-93534	089-93530
1000,00000	Becker Berlin	030-8853-0	030-8853-999
1001,00000	Lampen-Markt GmbH	069-467653-0	069-4676521
1002,00000	Omega Soft-Hardware Ma...	09256-4548-0	09256-45485
1003,00000	IDES AG Tochtergesellsch...	0621/7700-0	<Null>
1010,00000	Becker Berlin (Versand)	030-8373269	030-9847658

KUNNR	NAME1	TELF1
5638,00000	Kauftrausch	(0211)6450
12345,00000	Heinz Customer	123456
8888,00000	Andrew Sands	721 7722-01
2,000000	vWett	<Null>
6001,00000	R/WA Bad Homburg	<Null>
5511,00000	Lever	<Null>

4.3.5 String Matching

Many objects, like names, cities, products, customers, etc. might occur with different spellings within a data set, though defining only a single object or part of an object. Hence, a field in a record can be spelled in different ways, e.g. 'Business Objects', 'BusinessObjects', 'Business Object Germany', 'Busines Object', etc.. In order to filter all records which contain a certain string, it might be necessary to specify a rule / string with wildcards, either for a single or multiple occurrences of characters, like defining for all records containing the string 'Busi' AND 'Object' as valid, hence defining a search string as `"*Busi*Object"`.

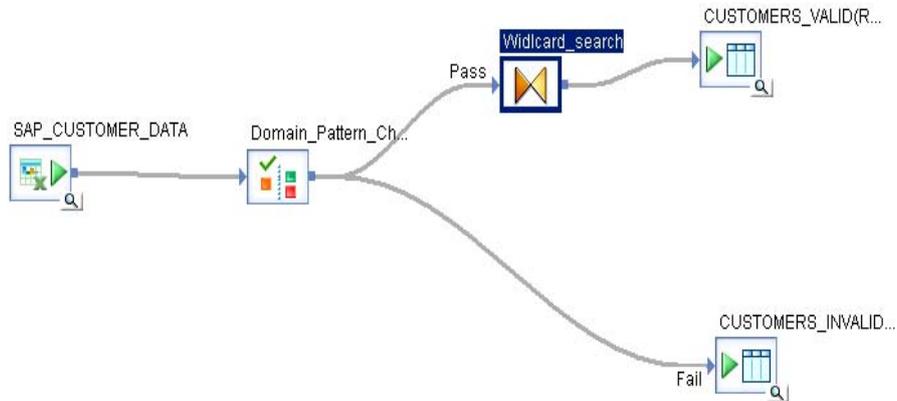
A business scenario could be to find suburbs of the same main city, find all records that are based on the same / similar field value (e.g. subsidiaries belonging to one company), or detect misspellings of the same object.

This check cannot be achieved with the *Validation* Transform, yet with a *Query* Transform. We are using the WHERE clause with the LIKE statement within the *Query* Transform. The available wildcard are the standard SQL wildcards:

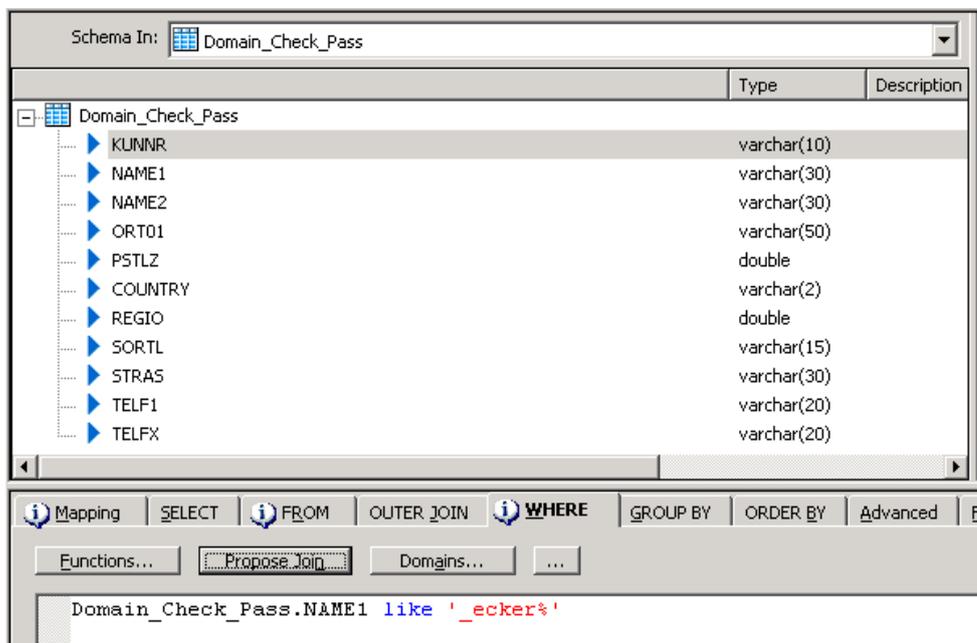
- **'%' character** a substitute for zero or more characters
- **'_' character** substitute for exactly one character
- **'[charlist]'** any single character in charlist
- **'![charlist]'** any single character NOT in charlist

In our example we want to find all records that somehow belong to the company with the name 'Becker' in the field *NAME1*. To cater also for misspellings in respect to upper and lower case, we are using the search string `'_ecker%'` (skipping the check of the first character). If you want to be even more precise, you could use the search string `'[bB]ecker%'`.

- Open the workflow, you have created in the previous section, and add a *Query* Transform after the *Validation* Transform.
- Connect the *Query* Transform for the case of *Passing* the *Validation* Transform.



3. Double-Click on the *Query* Transform.
4. Switch to the *Where* tab strip.
5. Drag the field NAME1 to the canvas, and complete the WHERE statement with the string 'like '_ecker%'".



6. Close the window.
7. Validate your DataFlow by using either the menu entry *Validation* → *Validate* → *Current View / All objects in view* or use the respective icons ()

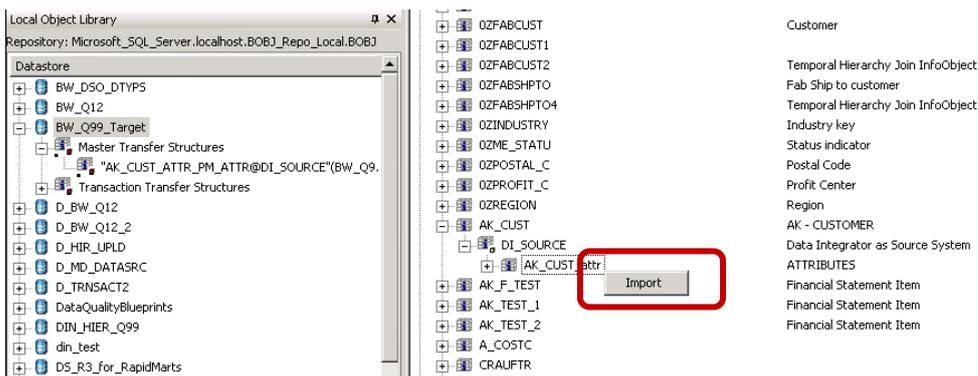
8. If your Job and DataFlow is correct, you can save and execute it. Position the cursor on the Job name, and select from the context menu *Execute*.
9. After the job has completed successfully, you can check the result by returning to the DataFlow display. By using the *View Data* option for the valid and invalid targets, you can display which records match the defined string.

KUNNR	NAME1	COUNTRY	REGIO	SORTL	STRAS	TELF1	TELFX
1000,00000	Becker Berlin	DE	11,000000	BECKER	Calvinstrasse 36	030-8853-0	030-8853-999
1010,00000	Becker Berlin (Versand)	DE	11,000000	BECKER	Beatestrasse 4	030-8373269	030-9847658
1020,00000	Becker Berlin (Lagerung)	DE	11,000000	BECKER	Blankenburgerweg 4	030-9321107	030-9321110
1050,00000	Becker AG	DE	11,000000	BECKER	Industriestrasse 23	030-982699-0	030-923899-800
1320,00000	Becker Koeln	DE	5,000000	BECKER	Wahnheider Strasse 57	0221-5689-100	0221-5689-300
1321,00000	Becker Stuttgart	DE	8,000000	BECKER	Triberger Strasse 42	0711-7100	<Null>

4.4 Loading data from Data Services into SAP BI

Last, but not least, we want to load the data eventually into a SAP BI system. This procedure is very straight forward. Instead of writing the data to a template table and / or flat file, we are including the respective target SAP BI DataSource (after import) in the DataFlow. Since the loading process has to be initiated by the SAP BI system, we have to create a batch file for the execution of the Job in Data Services. This batch file can subsequently be called by an InfoPackage in the SAP BI system.

1. To import the SAP BI structures, open the *Datastore* tab strip in the Object Library. Search for the Datastore that you have created for the SAP BI system as target (in our case *BW_Q99_Target*). Position the cursor on the Datastore name and choose *Open* from the context menu. Depending on the structures you want to use, expand the Master InfoSources or Transaction InfoSources tree. Find your *InfoSource*, and open its subtree. Position the cursor on the DataSource name, and use the option *Import* from the context menu. Afterwards, the DataSource will be available in the Object Library for your SAP BI system.



Note

Alternatively, you can import the structures also directly from the Object Library. Position the cursor on SAP BI system, and choose either the option *Reimport All* or *Import By Name* from the context menu.

2. Open the workflow, you have created in the previous section, and replace the *Query Transform* target for the valid records (in our case the template table *Customers_Valid*) by the SAP BI target. To do so, open the *Datastore* tab strip in the Object Library. Search for the Datastore that you have created for the SAP BI system as target (in our case *BW_Q99_Target*). Drag the

structure to the canvas you want to load the data in the SAP BI system to. Connect the existing *Query Transform* with the new target.



- Adjust the mapping in the *Query Transform* to map the fields to the SAP BI DataSource

Note

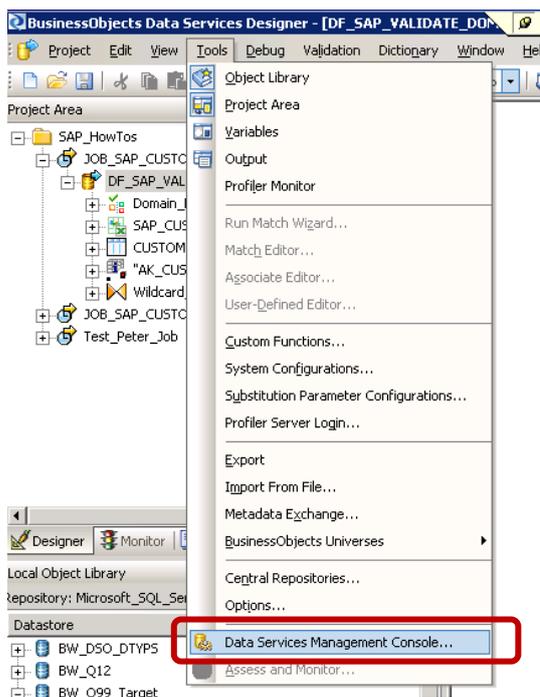
If the mapping in the *Query Transform* is not automatically changed or you cannot figure out it to map to the SAP BI target, remove the *Query Transform*, and recreate it from scratch. This will automatically provide you with the correct output schema

- Validate your DataFlow by using either the menu entry *Validation* → *Validate* → *Current View / All objects in view* or use the respective icons ()
- If your Job and DataFlow is correct, you save it.

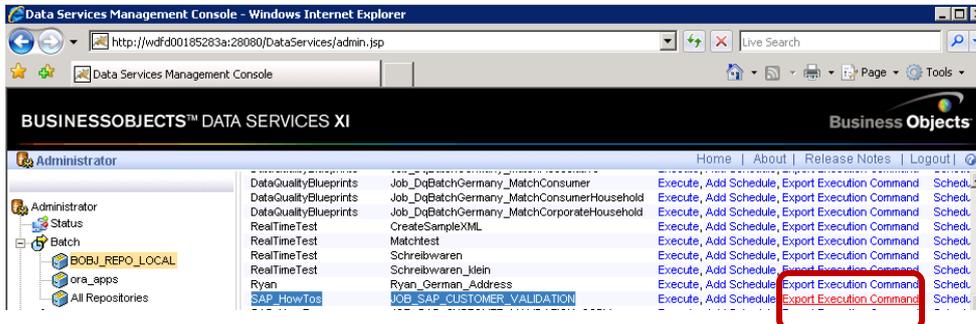
Note

The execution of the Job in Data Services would fail. The job will be triggered by an InfoPackage created in the SAP BI system

- Open the Data Services Management Console by choosing the corresponding menu entry from the *Tools* menu.



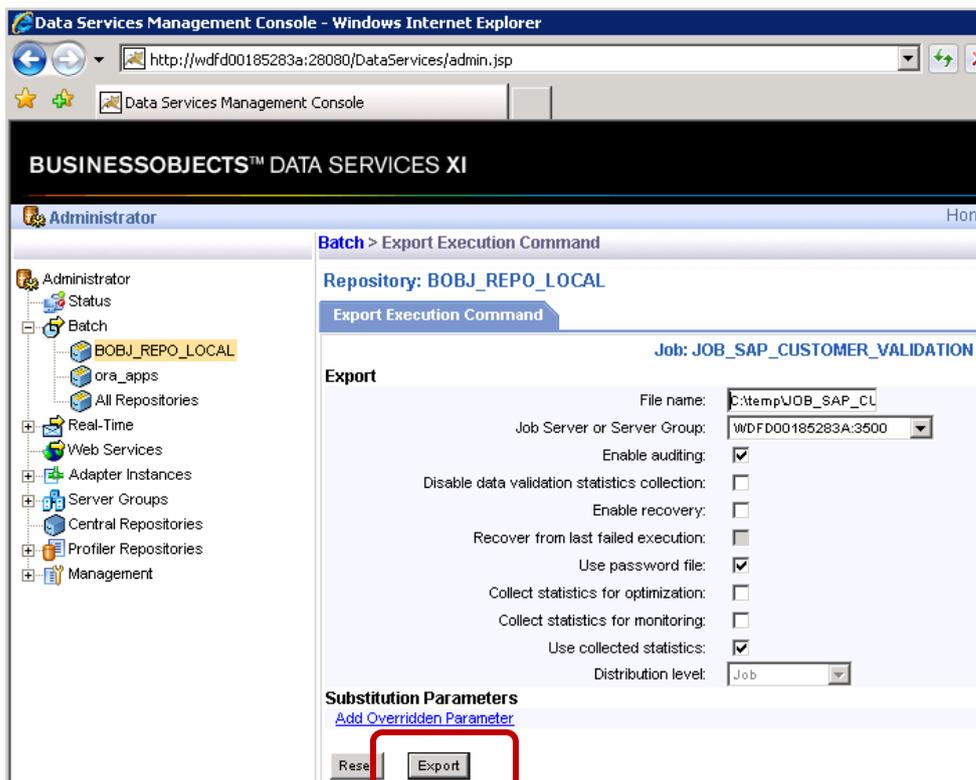
7. Log in with your user credentials. In the Management Console navigate to the Administrator, and open your repository (or *All Repositories*) of the *Batch* folder.
8. Switch to the *Batch Job Configuration* tab strip and find your Job you want to schedule. Choose the option *Export Execution Command*.



9. Provide a File Name for the batch file. Leave the other settings on the default values and press the *Export* button.

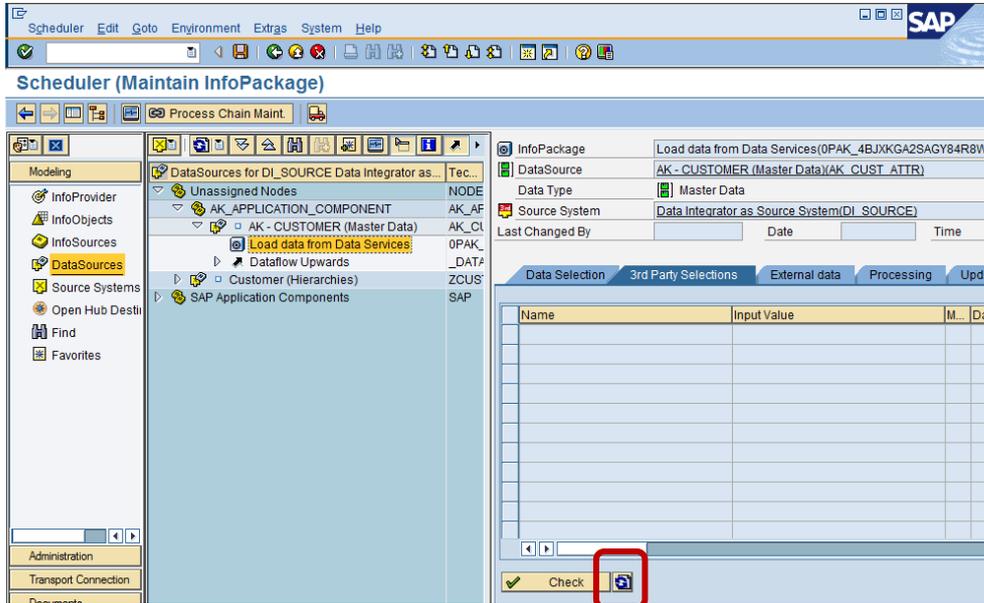
Note

Since the maximum length of the *file name* entry field in the InfoPackage is limited to 44 characters, your file name entered must not exceed 40 characters (44 characters minus 4 characters for the extension *.bat*)

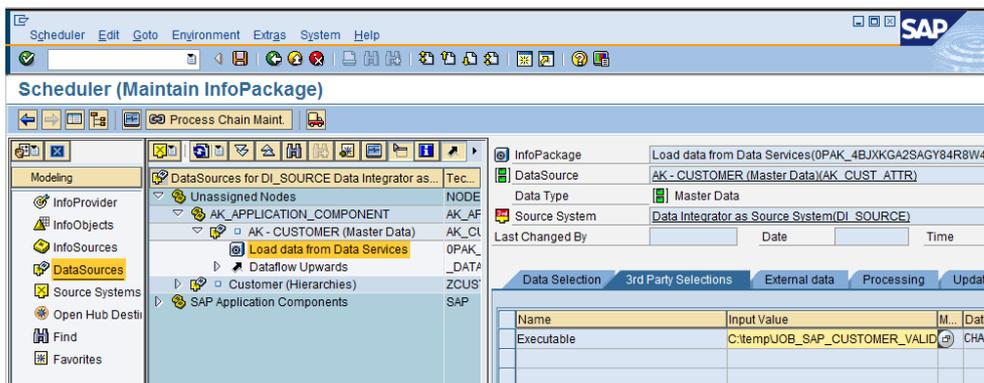


10. Before you switch to the SAP BI system, check that the RFC server is still running. See also section [Connecting SAP BI with Data Services](#).

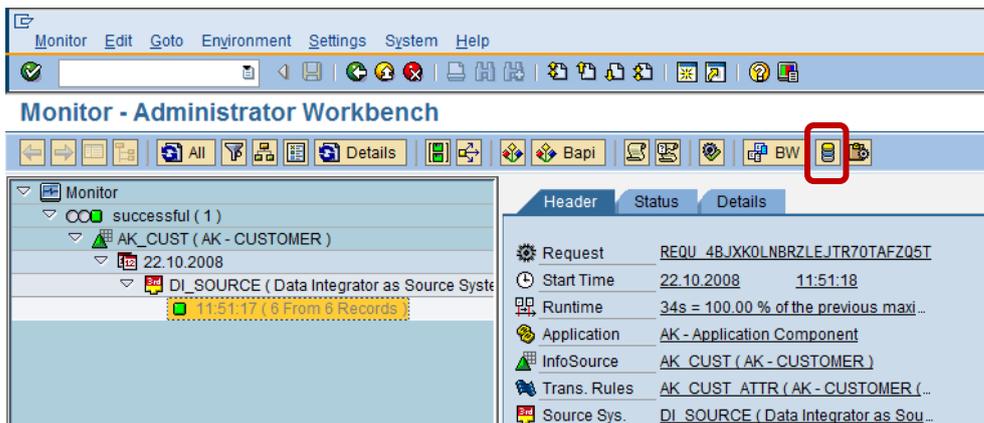
- Switch to the SAP BI system you want to load the data to. Open the *Data Warehousing Workbench* and find your InfoSource / DataSource. Create an InfoPackage for the DataSource. Switch to the *3rd Party Selection* tab strip. Press the *Refresh Sel. Fields* button to display the input fields.



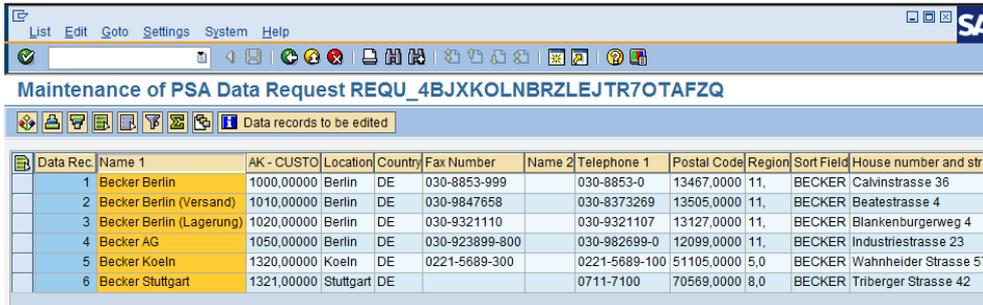
- Enter the File Name that you have provided in the creation of the batch file.



- You can now schedule the load. If you schedule the execution in the future, please check that the RFC server is running at the time of execution. After the execution of the InfoPackage, you can check in the monitor the status of the load.



- To check whether your data has been transferred as expected, you can check the PSA for the load. From the monitor, you can choose the  icon in the monitor to jump to the PSA display. In our example we transferred the records which passed all our Data Quality measures, i.e. all records which contain the customer Name *Becker*, are located in the USA, Australia, Germany and England, and adhere to the pattern we defined for the phone number. In total this are 6 records.



Maintenance of PSA Data Request **REQU_4BJXKOLNBRZLEJTR7OTAFZQ**

Data records to be edited

Data Rec	Name 1	AK - CUSTO	Location	Country	Fax Number	Name 2	Telephone 1	Postal Code	Region	Sort Field	House number and str
1	Becker Berlin	1000,00000	Berlin	DE	030-8853-999		030-8853-0	13467,0000	11,	BECKER	Calvinstrasse 36
2	Becker Berlin (Versand)	1010,00000	Berlin	DE	030-9847658		030-8373269	13505,0000	11,	BECKER	Beatestrasse 4
3	Becker Berlin (Lagerung)	1020,00000	Berlin	DE	030-9321110		030-9321107	13127,0000	11,	BECKER	Blankenburgerweg 4
4	Becker AG	1050,00000	Berlin	DE	030-923899-800		030-982699-0	12099,0000	11,	BECKER	Industriestrasse 23
5	Becker Koeln	1320,00000	Koeln	DE	0221-5689-300		0221-5689-100	51105,0000	5,0	BECKER	Wahnheider Strasse 57
6	Becker Stuttgart	1321,00000	Stuttgart	DE			0711-7100	70569,0000	8,0	BECKER	Triberger Strasse 42

www.sdn.sap.com/irj/sdn/howtoguides