



# **MDM and Data Services Integration Scenarios - Part I: Advanced Consolidation**

**Applicable Releases:**

**SAP NetWeaver Master Data Management 5.5**

**SAP NetWeaver Master Data Management 7.1**

**SAP Business Objects Data Services XI 3.1 and above**

**Topic Area:**

**Enterprise Information Management**

**Capability:**

**Master Data Management**

**Version 1.0**

**April 2010**



**HOW-TO GUIDE**

© Copyright 2010 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
<b>Example text</b>	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.
<b>Example text</b>	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<b>&lt;Example text&gt;</b>	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

<b>1.</b>	<b>Business Scenario</b> .....	<b>1</b>
<b>2.</b>	<b>Background Information</b> .....	<b>3</b>
<b>3.</b>	<b>Prerequisites</b> .....	<b>3</b>
<b>4.</b>	<b>Step-by-Step Procedure</b> .....	<b>4</b>
4.1	Enhance repository structure in MDM .....	4
4.2	Create Slave repository in MDM.....	7
4.3	Create DB Views for Slave repository .....	10
4.4	Create Business Objects Data Store to access MDM Data .....	11
4.5	Create Business Objects Flat File format for import of Legacy Data .....	16
4.6	Importing XML Files delivered by Business Objects into MDM (New records) .....	19
4.7	Importing XML Files delivered by Business Objects into MDM (Updated records) ...	23
4.8	Consolidate Master Data in the MDM Data Manager.....	24

## 1. Business Scenario

SAP NetWeaver Master Data Management (MDM) is a powerful solution to import, maintain and syndicate master data of all types across different systems. Its in-memory technology allows searching for master data within the MDM repository with very short response times. You can identify potential duplicates across systems and manage key mappings as a prerequisite for master data distribution.

For new master data to be stored in the MDM repository – either when creating manually or when connecting new Client systems to an MDM landscape – it is best practice to perform a data cleansing and also a matching in order to find duplicate records. MDM itself supports cleansing and matching, however SAP Business Objects Data Services (DS) provides more comprehensive cleansing and matching capabilities, e.g., standardizing and correcting names, connecting to international address data directories such as US Postal directory, supporting different matching algorithms (heuristic, rules-based, probabilistic, deterministic, empirical, template), etc. Besides this, it provides full ETL functionality to extract data from any kind of source, transform it, and load it to any destination.

We have identified the following scenarios where both solutions can be combined in order to complement each other:

- **Advanced cleansing and consolidation (data migration scenario):**

You would like to connect a new client system to your MDM landscape, hence initially load the master data into MDM. Use the full flexibility of Data Services to extract, transform, and load the data into the MDM repository. Run cleansing, validation, normalization, and de-duplication in Data Services prior to loading the data into MDM. In order to match the new data against the existing in the MDM repository, read data via DB views. After having loaded the data in MDM, syndicate the same to your clients.
- **Real-time cleansing and enrichment on data entry:**

You have chosen MDM as the central repository for your master data. You would like to create new master data in the MDM Data Manager. You can ensure data governance via MDM workflow. On data entry, the data needs to be enriched by calling data quality and enrichment services provided by Data Services. Integration to Data Services is done via the Enrichment Architecture.
- **Periodic cleansing, and consolidation of data in MDM repository:**

You have chosen MDM as the central repository for your master data. You have noticed that some of your data is incomplete. Furthermore, it might be that some data is outdated, e.g., data must be updated with the most recent address information. So, on a regular basis you need to run cleansing, and consolidation jobs to ensure data integrity. The data needs to be exported, processed and re-imported into the MDM repository which is handled via the Enrichment Architecture.
- **Central master data creation with data quality check on data entry:**

You have chosen MDM as the central repository for your master data. Maintenance of your master data is done from within the Enterprise Portal, either via iViews or Web Dynpro components. You can control data governance via SAP NetWeaver Business Process Management (BPM). Integration to MDM is done via Web Services or MDM Java API. In order to prevent duplicate entries from entering the MDM repository, you call data quality services on data entry such as duplicate check, address cleansing, name cleansing, and normalization, etc. Integration to Data Services is done via Web Services.
- **Extracting data for all kinds of sources and data transfer into MDM:**

Use native ETL functionality of Data Services to extract, transform, and load your master data into MDM, see also above.

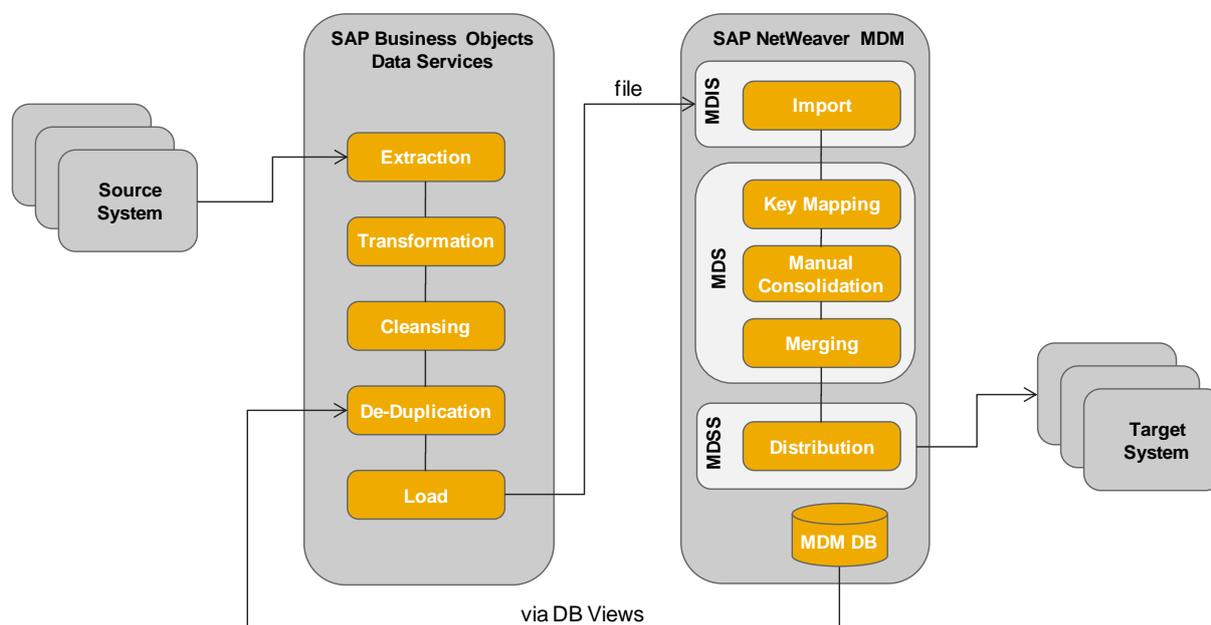
- Master data joined with analytics:  
Run data profiling reports provided by Data Services to get an insight of the quality of your master data in MDM, and visualize the results in dashboards.

This document is the first part of a how-to guide series dealing with the integration between SAP NetWeaver MDM and SAP Business Objects Data Services. The goal of the how-to guides is to provide you with guidelines about how to implement the scenarios above. We will stick to the data migration, the cleansing on data entry (including periodic cleansing), and the central master data management scenarios. Implementation teams can use those findings as a jump start for their own integration projects.

In the second part of the how-to guide series, we cover the on entry and periodic cleansing scenarios. In the third part, we focus on the central master data creation scenario. You can download those how-to guides from the SAP NetWeaver how-to guides area on SAP Developer Network (SDN) <https://www.sdn.sap.com/howtoguides>.

In this document it is explained how you can connect SAP NetWeaver MDM with SAP Business Objects to perform address cleansing and matching for data, and finally to import this data into MDM.

## 2. Background Information



The above picture shows the system landscape required to setup the scenario. Data from the different client systems are loaded into SAP Business Objects Data Services by using different native extractors from SAP Business Objects. Extraction of Data from SAP NetWeaver MDM is done via Database Views. Once all required data are extracted the cleansing and de-duplication process starts in Business Objects Data Services. This is an automatic process without any user interaction. Results of these operations are written into several xml-Files.

There are two XML-Files written

- One for completely new records.
- One for existing records within MDM which are determined as duplicates

Both XML-Files have to be imported into MDM using different Import Maps.

After the import of both files in MDM a manual Consolidation is performed utilizing the information provided by Business Objects.

Afterwards the MDM Syndication Server syndicates the now cleansed records back to the respective Client systems.

## 3. Prerequisites

- SAP NetWeaver MDM 5.5 or SAP NetWeaver MDM 7.1
- SAP Business Objects Data Services 3.1 or higher
- Data Cleansing packages depending on your needs
- SAP DB View generation package (please contact your SAP rep)

## 4. Step-by-Step Procedure

As already explained in the previous section, this is a cross scenario going over several applications. While the cleansing and matching functionality in SAP Business Objects Data Services can get quite complex (depending on data structures to be extracted and cleansing and matching logic) we provide here a description on how to setup MDM and as well how to setup the interface between MDM and Business Objects. We assume that after a successful Business Objects matching run, we would get two XML-Files delivered which have to be imported into MDM. The structure of the XML-Files including some sample data will be explained later.

### 4.1 Enhance repository structure in MDM

It is important to bring the matching scores and related data from Business Objects into MDM, so it is obvious to enhance the MDM data structure in order to capture this information. All enhancements are done in the main table for which you would like to import records.

1. Add the field “BO\_Breakkey” with the properties seen in the screenshot below. Break keys are used in SAP Business Objects Data Services in order to pre-group records before matching. Break keys are calculated using a defined formula and only records with the same break key are matched against each other. This has the advantage that matching utilizing break keys has a much higher performance.

Field Detail	
Name	BO Break Key
Code	BO_BREAK_KEY
Description	BO DS TECHNICAL FIELD
Type	Text
Required	No
Writable Once	No
Matrix	Normal
Multi-Lingual	No
Keyword	None
Display Field	No
Unique Field	No
Sort Index	Normal
Calculated	No
Calculation	
Width	50
Sort Type	Case Insensitive

2. Add the field “Matching Group” with the properties seen in the screenshot below. When several records match with each other, SAP Business Objects Data Services assigns a matching group to these records. So when for instance 5 records match with each other, you can find them in Data Manager by either searching for the same matching group or simply sort all records according to the matching group and you get all records displayed in ascending/descending order and for further processing can mark the records with the same matching group.

Field Detail	
Name	Matching Group
Code	Matching_Group
Description	
Type	Text
Required	No
Writable Once	No
Matrix	Normal
Multi-Lingual	No
Keyword	Stemmer
Display Field	No
Unique Field	No
Sort Index	Normal
Calculated	No
Calculation	
Width	10
Sort Type	Case Insensitive

3. Add the field “Matching Score” with the properties seen in the screenshot below. This field stores the matching score for the record.

Field Detail	
Name	Matching Score
Code	Matching_Score
Description	
Type	Text
Required	No
Writable Once	No
Matrix	Normal
Multi-Lingual	No
Keyword	Stemmer
Display Field	No
Unique Field	No
Sort Index	Normal
Calculated	No
Calculation	
Width	10
Sort Type	Case Insensitive

The matching score is the result of the matching comparison of the current record with a Golden record (In Business Objects DS such a record is called Driver record).

Field Detail	
Name	Golden Record
Code	Golden_Record
Description	
Type	Boolean
Required	No
Writable Once	No
Matrix	Normal
Multi-Lingual	
Keyword	None
Display Field	No
Unique Field	No
Sort Index	Normal
Calculated	No
Calculation	
Search Tab	No
True Value	YES
False Value	NO
Default Value	None

- Add the field "AutoMergeID" with the properties seen in the screenshot below. This field stores a unique identifier for a record. It is a combination of record-id and Remote-system. Reason behind this is that in different remote systems two completely different records have the same record identifier. Combined with the remote-system id we are able to see a truly unique identifier over all systems, e.g., when we are in the comparing records view in the MDM Data Manager.

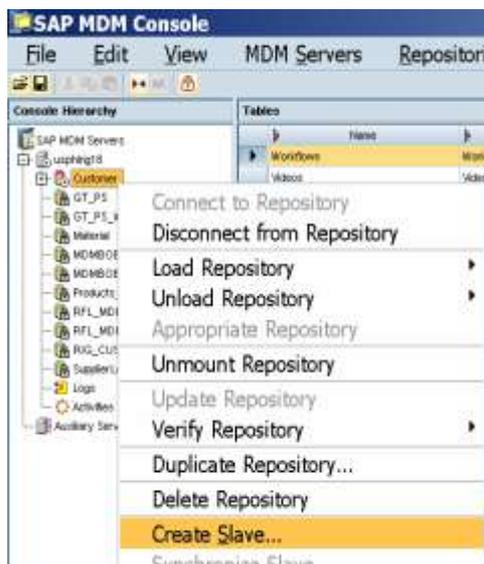
Field Detail	
Name	AutoMergeID
Code	GoldenRecordID
Description	
Type	Text
Required	No
Writable Once	No
Matrix	Normal
Multi-Lingual	No
Keyword	None
Display Field	No
Unique Field	No
Sort Index	Normal
Calculated	No
Calculation	
Width	15
Sort Type	Case Insensitive

- Add the field "status". This is a Boolean field and serves the purpose that each newly imported record is by default inactive. The idea would be that after an approval process this field can be set to true and thus the record would be considered active. It also serves the purpose when some of the new records are duplicates, they would have the status inactive. After a consolidation records can be merged together, the duplicates would be gone and the remaining golden record would get the status "active".

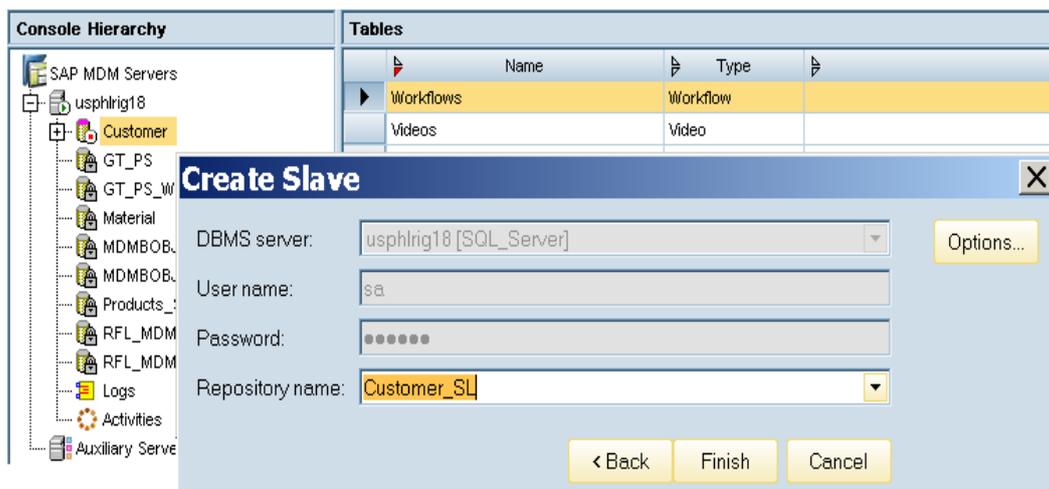
## 4.2 Create Slave repository in MDM

1. Since there is no native extractor available to extract Data from MDM, extraction occurs via DB Views from a MDM Database. As of now, extraction of data is only possible from Slave servers (due to potential locking issues during the extraction in parallel to an ongoing write operation onto the database). Since the Slave Server will be used for data extraction into Business Objects DS, please be aware that slaves need to be synchronized from time to time with their master in order to have up-to-date data. Otherwise it can be that outdated data are used for cleansing and matching in Business Objects DS. Depending on how many changes are performed on the master, the slave needs to be synchronized more or less often.

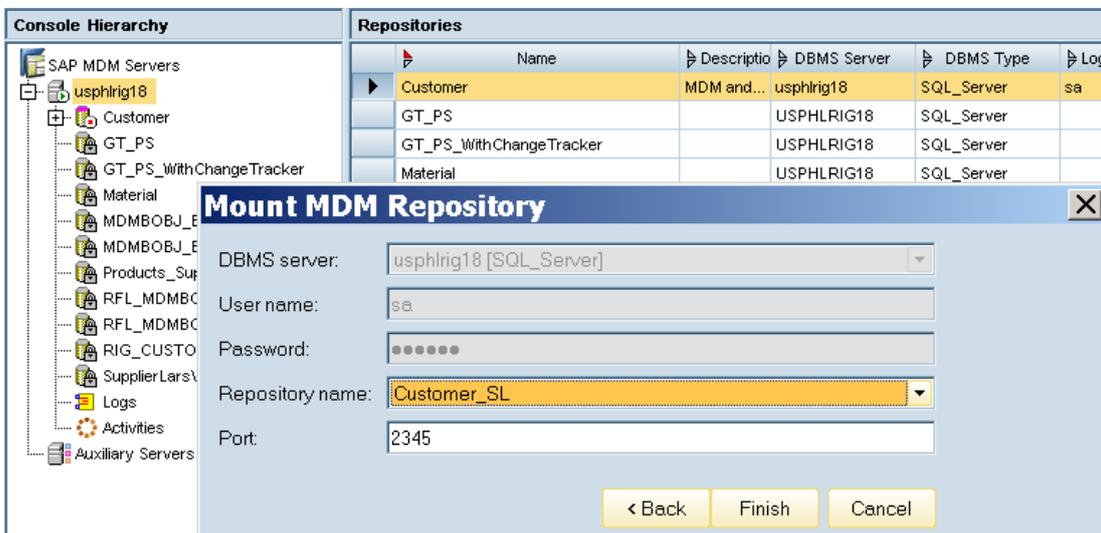
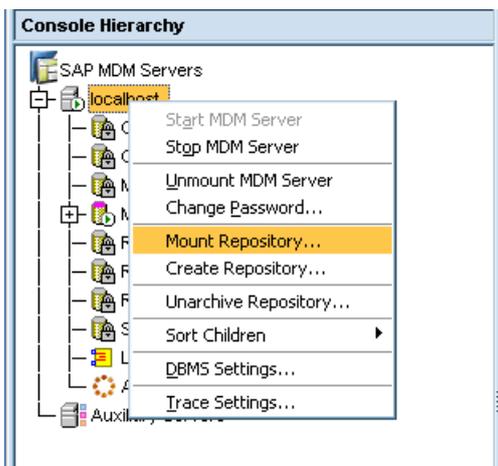
So the first step to perform here is to create a slave server from our repository.



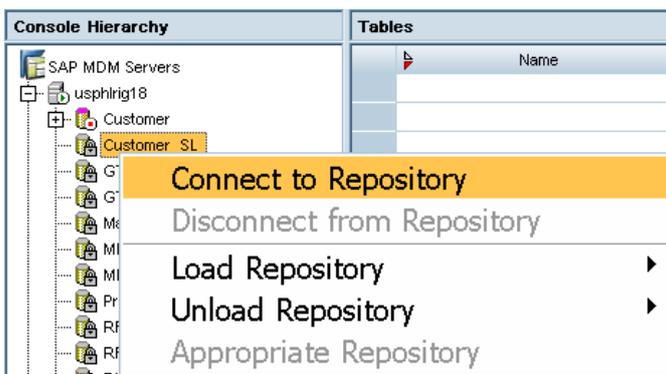
2. Enter the name of the slave repository.



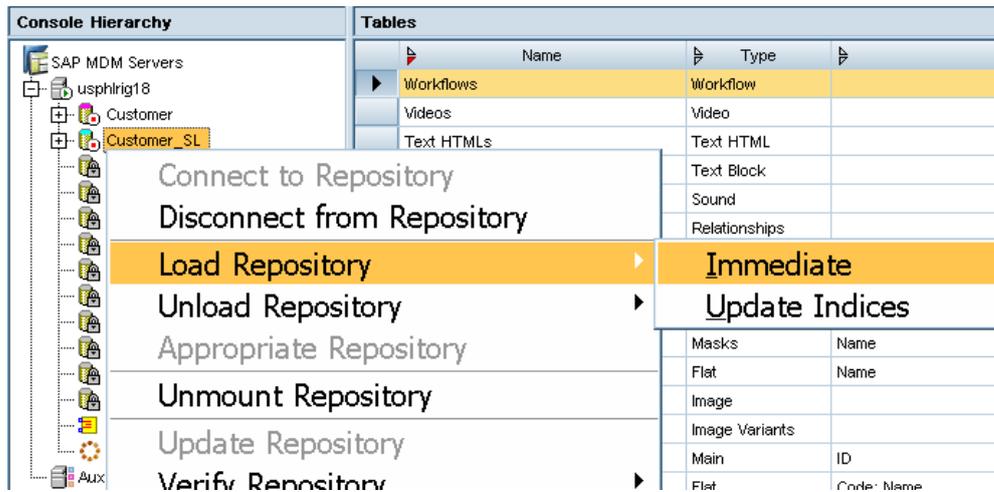
3. Mount the slave repository.



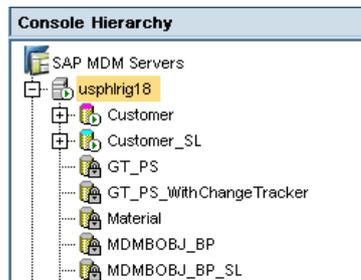
4. Connect to the Slave repository.



5. Load the slave repository.



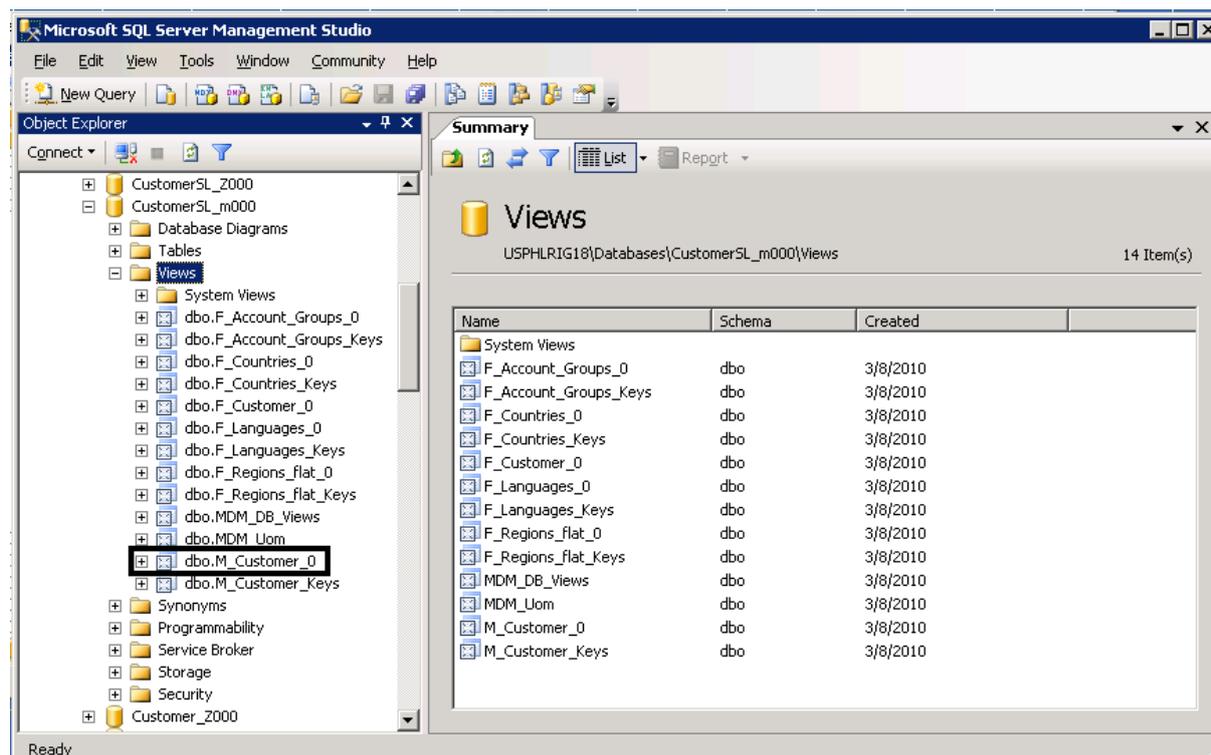
6. After the successful load, your slave repository should be displayed as "Loaded Running".



### 4.3 Create DB Views for Slave repository

For the DB View creation, SAP provides a special tool which does the creation for you. The tool is not available on the SAP Service Marketplace, so please touch base with your SAP representative to get the tool. Together with the tool there comes also a guide on how to use it and as well describing the nomenclature of the generated views. Please consider that as of now the DB View generation allows only the generation of DB Views for Slave repositories.

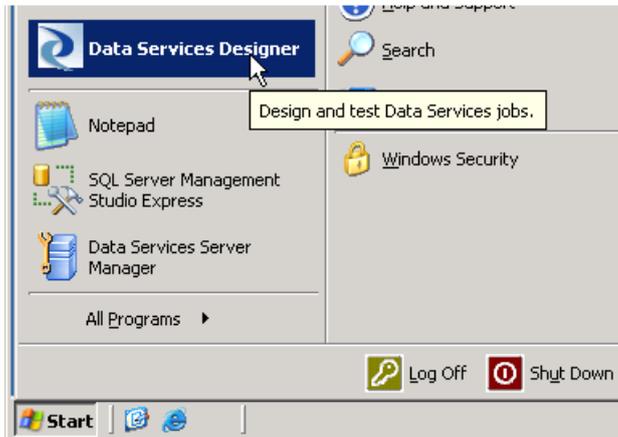
Once the DB views are created, you should see a picture like this when you go into your Database (screenshot here taken from MS SQL-Server).



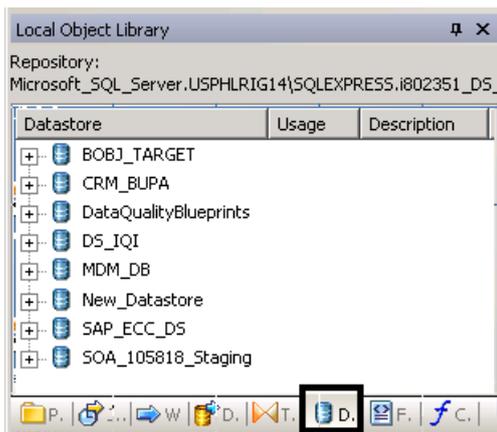
The DB View “dbo.M\_Customer\_0” is the view used to view the main table records of the repository. Later in Business Objects, we will enable access to this DB View.

## 4.4 Create Business Objects Data Store to access MDM Data

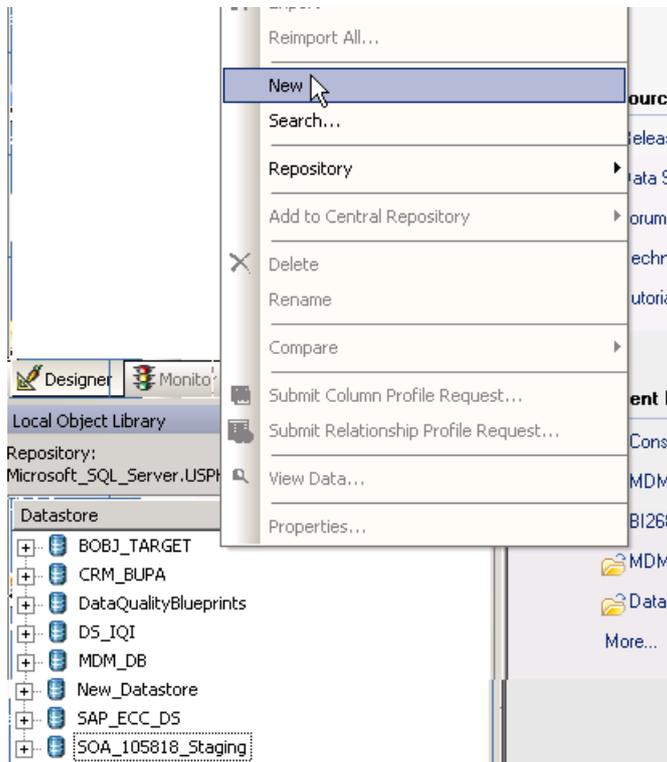
1. Start the Business Objects Data Services Designer.



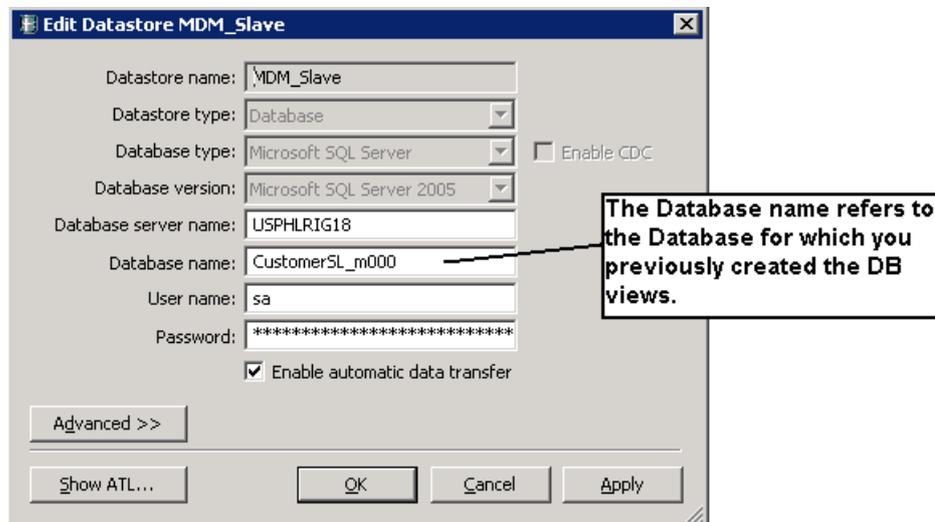
2. In the Designer in the local object library, switch to the "Data Stores" view.



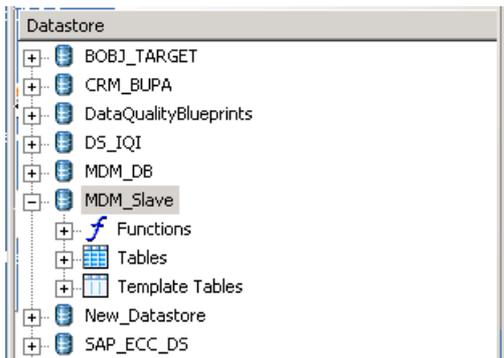
There exists not yet a Data Store for access to the Slave repository, so we decided to create one now. Therefore, place the mouse in the Data Store pane and via right mouse click select “New”.



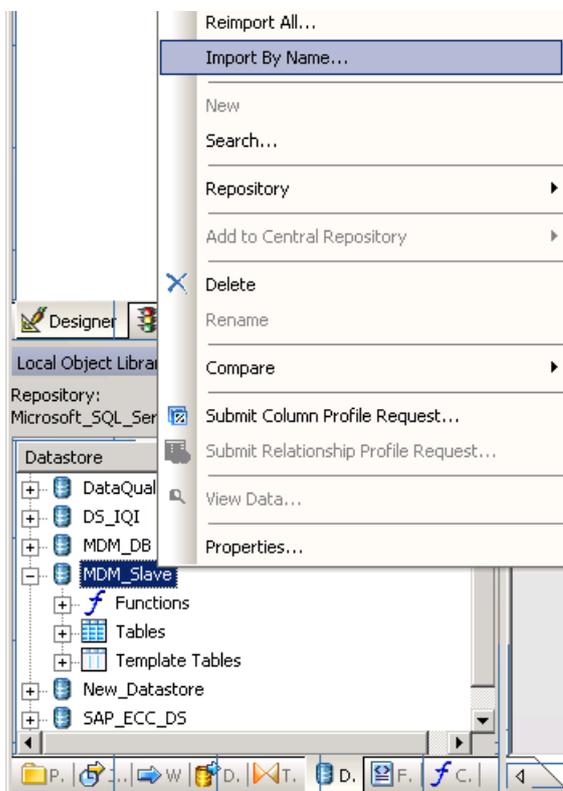
3. Populate the fields for the Data Store to be created. Press the “OK”-button once you’re done.



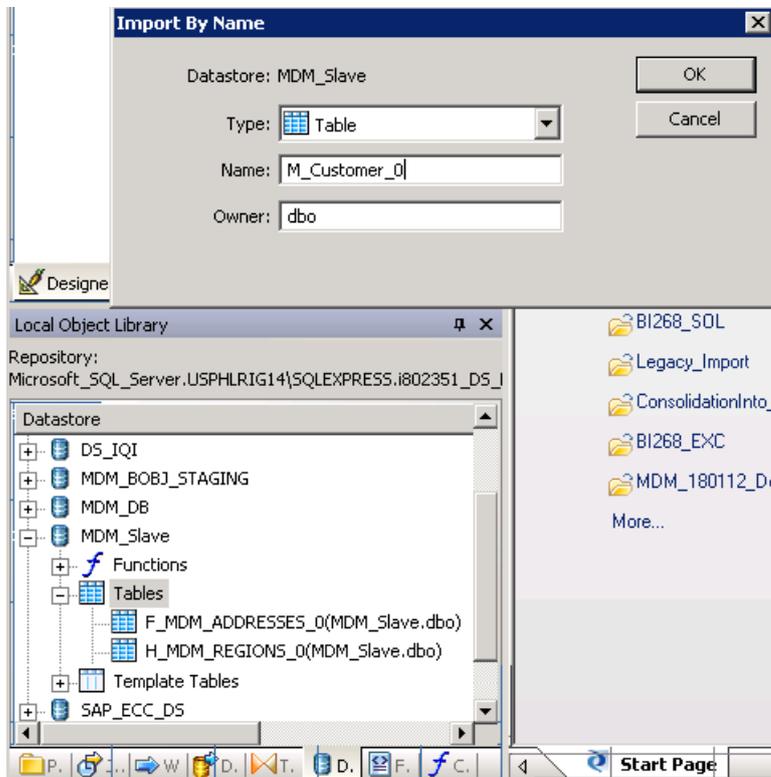
4. Now the Data Store should be visible in the list of Data Stores.



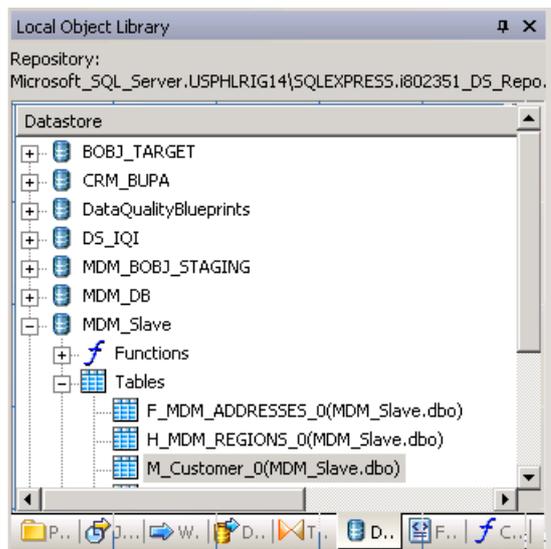
5. Now we need to import the DB View definitions which are of interest for us. Therefore, select the Data Store and via context menu select "Import By Name".



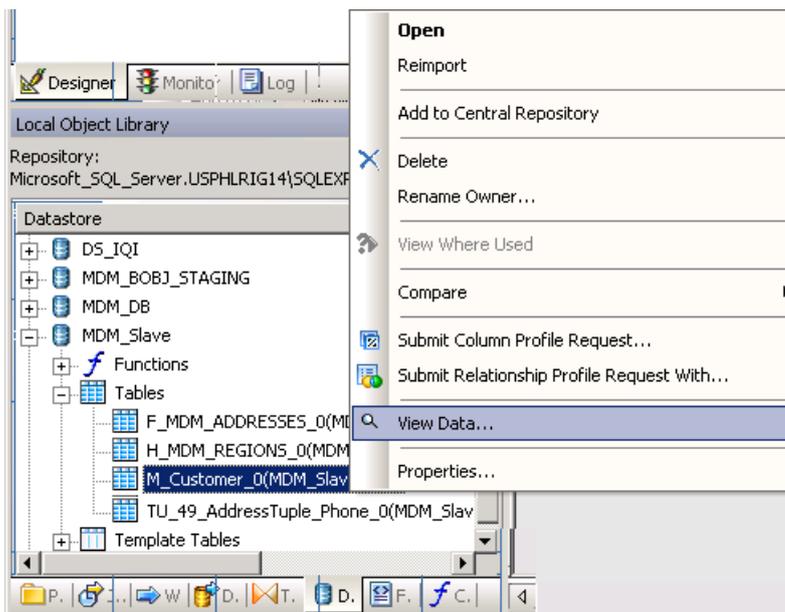
6. Now we need to import the DB View definitions which are of interest for us. Therefore, in the popup select “Table” and enter the name of the DB View which we would like to import (in our case “M\_Customer\_0”). This view represents the main table of the repository. In the field “Owner” enter the name as it is displayed in your Database system (context sensitivity)



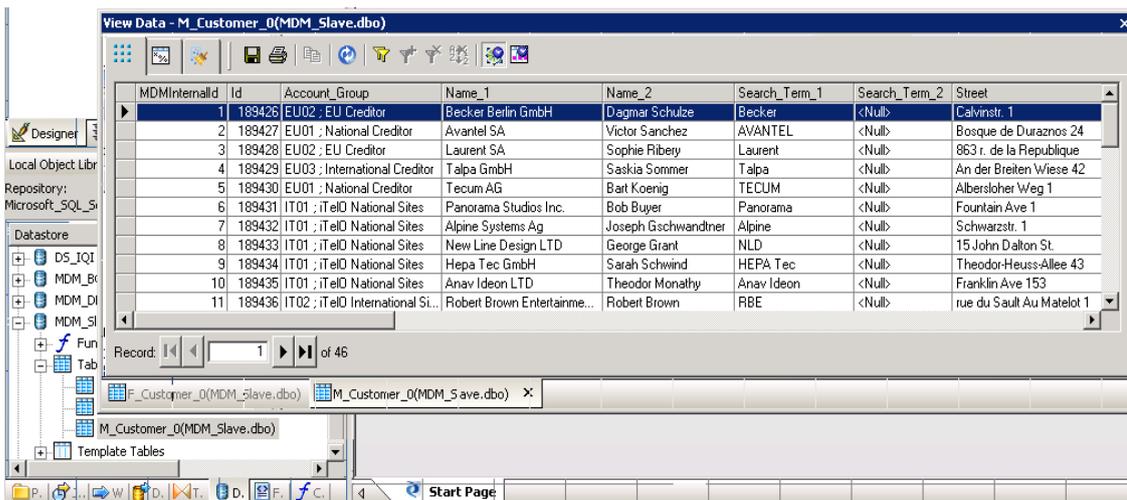
7. When you expand now the “Tables” icon under the *MDM\_Slave* Data Store, you can see all tables which you have imported.



8. Mark the table we just previously imported and via context menu select "View Data".

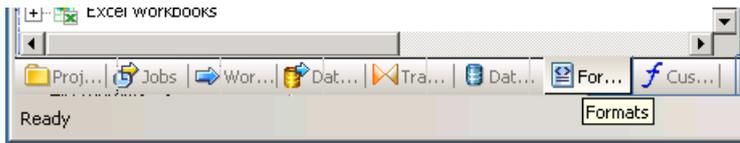


9. Now you can see the data of the main table of the Slave repository. The representation is pretty much the same as you would view the data in the MDM Data Manager.

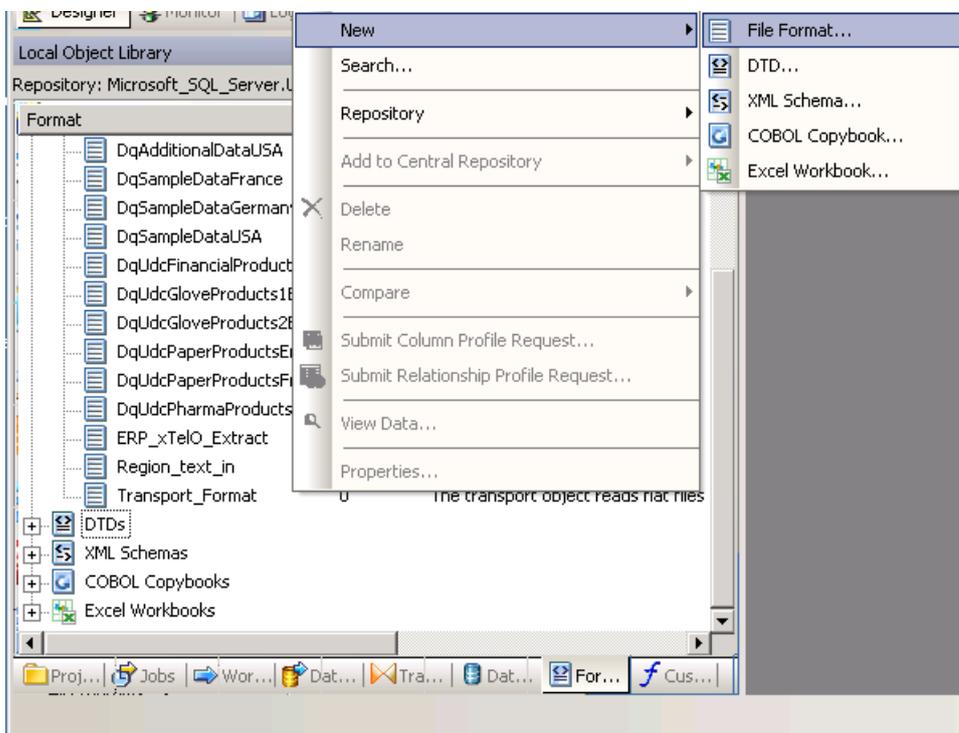


## 4.5 Create Business Objects Flat File format for import of Legacy Data

1. We assume that the data from the legacy system are delivered in text format, each column delimited by a tab. In order to read this file into Business Objects Data Services, we have to create a file format. Therefore, in the Data Services Designer in the Local Objects toolbar select the "Formats tab."

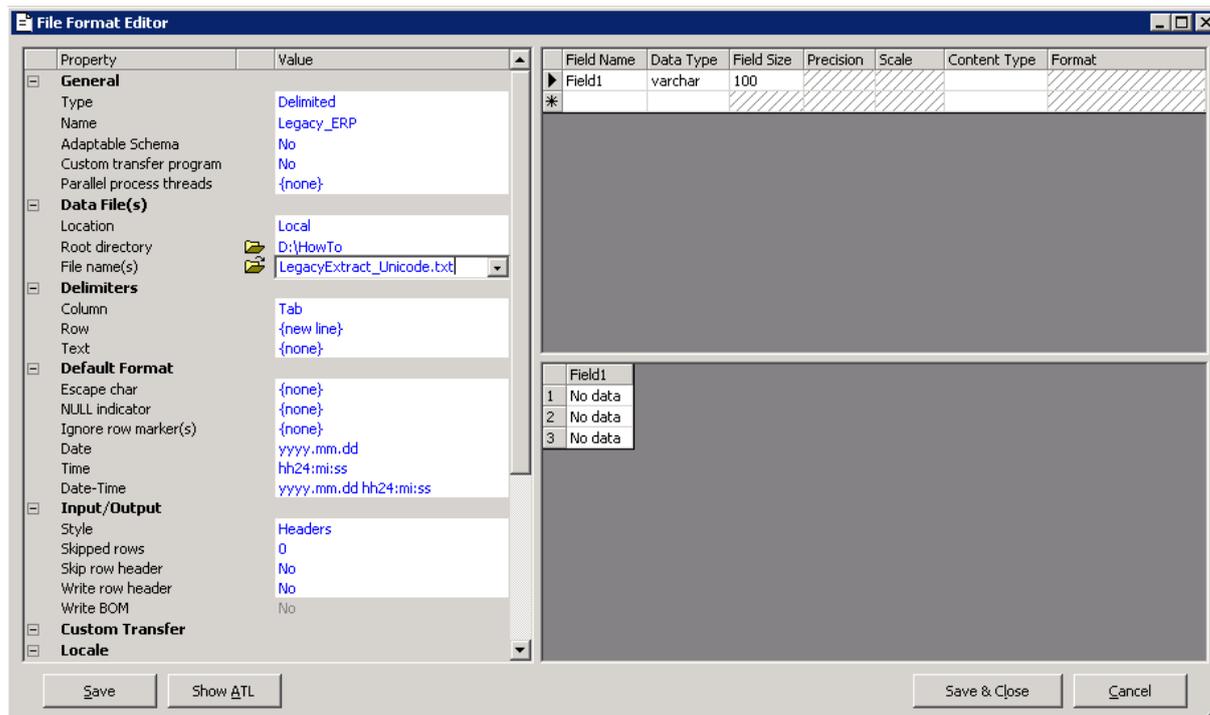


2. Via the context menu, select "New" → "File Format".



3. Populate in the dialog the necessary information to access and read the data from the file. In our situation, these are the following fields:

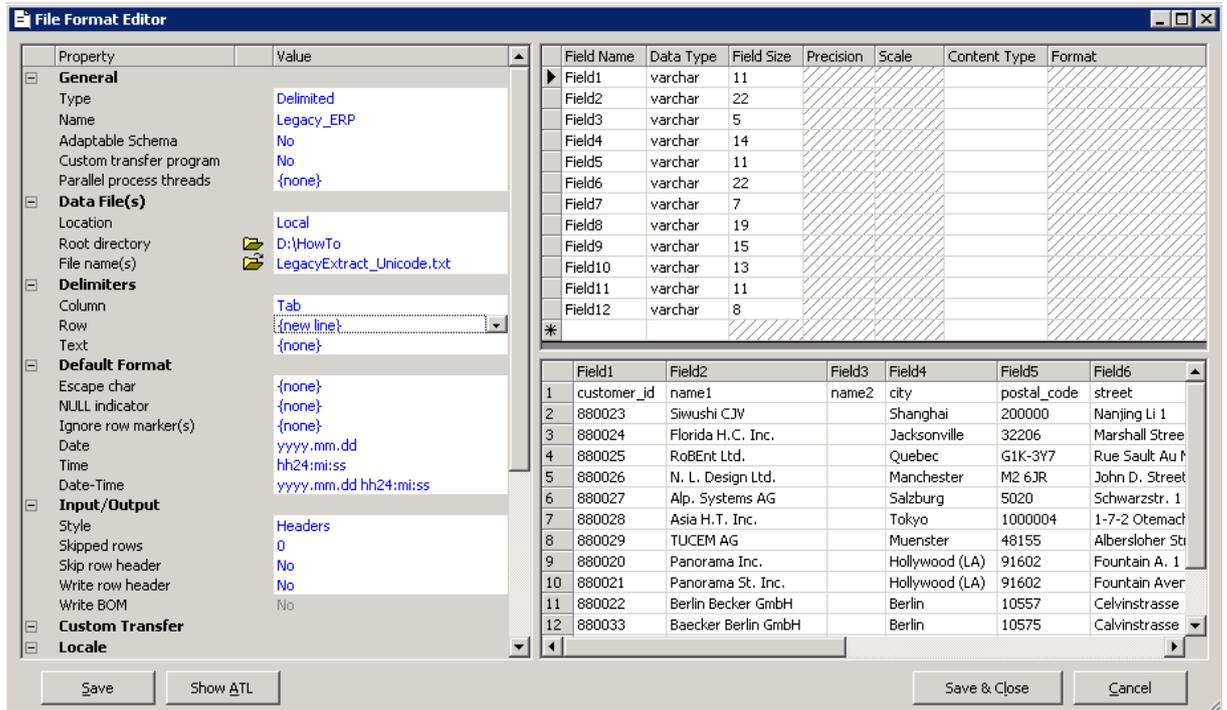
- Name                      Name of the format – we call it “Legacy\_ERP”
- Root directory         Directory where the file is stored
- File name(s)          Name of the File
- Column                 Tab (means the columns in the file are separated via a tab)



4. Once you select return after populating the dialog, the systems asks you whether the current schema definition should be overwritten with the schema definition in the source file. Select then yes, because that is what we finally would like to achieve.



5. Now the structure of the source file including the records is read and displayed in the dialog.



6. Select "Save & Close". The newly created format appears now in the list of file formats.



## 4.6 Importing XML Files delivered by Business Objects into MDM (New records)

As already described in the introduction, two XML files are written by the Business Objects job.

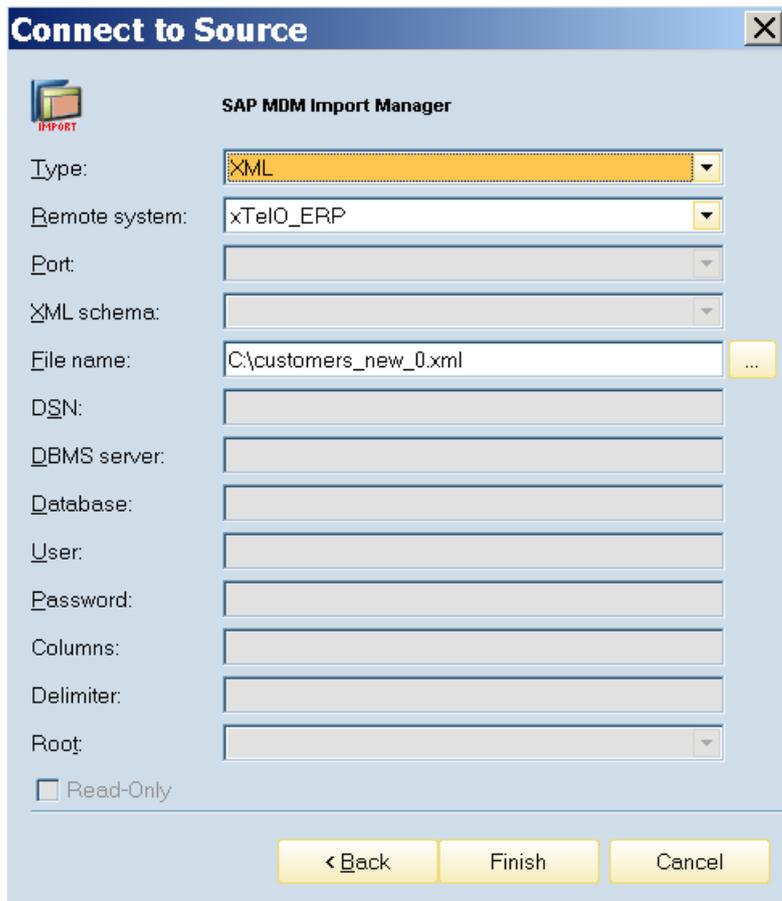
- One for completely new records in cleansed form. It should look like this:

```
- <CUSTOMER_ELEMENT>
  <CUSTOMER_NUMBER>880033</CUSTOMER_NUMBER>
  .....
  .....
  <MATCHING_SCORE />
  <MATCHING_GROUP />
  <GOLDEN_RECORD />
  <BO_BREAKKEY>1CDE</BO_BREAKKEY>
  <AUTOMERGE_ID>XTELO,880022</AUTOMERGE_ID>
</CUSTOMER_ELEMENT>
```

- One for existing records within MDM which are determined as duplicates. It should look like this:

```
- <CUSTOMER_ELEMENT>
  <MDM_ID>189430</MDM_ID>
  <MATCHING_SCORE />
  <MATCHING_GROUP>5</MATCHING_GROUP>
  <GOLDEN_RECORD>1</GOLDEN_RECORD>
  <AUTOMERGE_ID>ITELO,2154</AUTOMERGE_ID>
</CUSTOMER_ELEMENT>
```

1. First we import completely new records into the repository. Therefore, start the MDM Import Manager, choose the respective Remote system and file to read in. The format is XML.



2. Viewing records in the “Source preview Tab”. In this concrete example you can see that even among the new records there are some potential duplicates. You can see this because the columns “Matching Group” and “Matching Score” are actually populated.

Source Preview									
Partition Field/Value		Map Fields/Values		Match Records		Import Status			
Tables	Records								
CUSTOMERS_NEW	ACCOUNT_GROU	AUTOMERGE_ID	BO_BREAKKEY	CITY	CITY_POS	CUSTOMER_NUM	GOLDEN_RECORI	MATCHING_GROUP	MATCHING_SCORE
CUSTOMER_ELEMENT	EU04	XTELO,880023	ZNCN	Shanghai	200085	880023			
	EU04	XTELO,880024	1CUS	Newtown Square	19073	880024			
	EU04	XTELO,880020	9FUS	N Hollywood	91602	880020	0	2	90
	EU04	XTELO,880021	9FUS	N Hollywood	91602	880021	0	2	93
	EU04	XTELO,880022	1CDE	Berlin	10557	880022	0	1	97
	EU04	XTELO,880033	1CDE	Berlin	10575	880033	1	1	
	EU04	XTELO,880034	6DDE	Waldorf	69190	880034			

- Now we continue to the “Map Fields\Values”- Tab to perform the necessary field mappings. On this tab, map all fields which are of interest to you. In addition make sure that the following source fields are mapped.

**AUTOMERGE\_ID**



**BO\_BREAKKEY**



**GOLDEN\_RECORD**



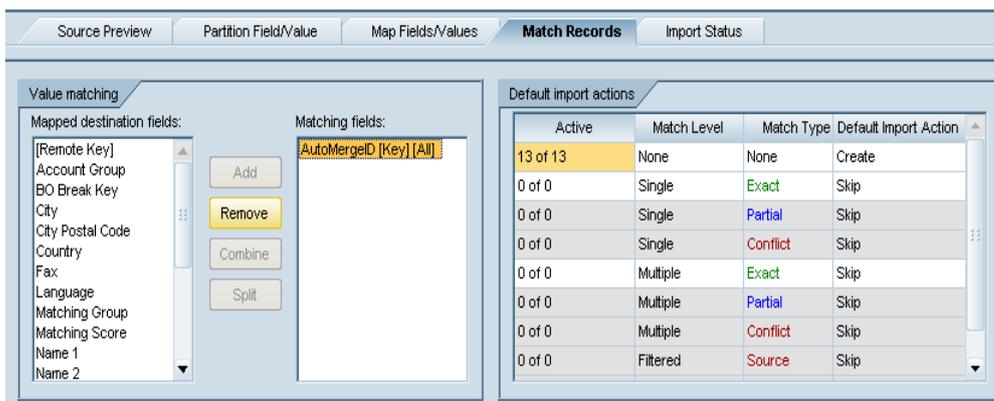
**MATCHING\_GROUP**



MATCHING\_SCORE



- Once you are done with the field and value mapping, switch to the "Match Records" tab and select the field "AUTOMERGE\_ID" as matching relevant field and import the records.



## 4.7 Importing XML Files delivered by Business Objects into MDM (Updated records)

The matching results for the existing records in the repository need to be transferred as well into MDM. Our file contains only matching records which have a match either among the already existing records in MDM or which have matches with the previously (newly) imported records. Therefore the structure of that file is very simple, it consists only of the identifying information for a record and a part for the matching results.

```
- <CUSTOMER_ELEMENT>
  <MDM_ID>189430</MDM_ID>
  <MATCHING_SCORE />
  <MATCHING_GROUP>5</MATCHING_GROUP>
  <GOLDEN_RECORD>1</GOLDEN_RECORD>
  <AUTOMERGE_ID>ITELO,2154</AUTOMERGE_ID>
</CUSTOMER_ELEMENT>
```

1. Viewing records in the “Source preview Tab”. In our example below, we can see that both records are “Golden Records” or “Driver Records”. The field Golden Record has a value of “1” (TRUE) while the field “MATCHING\_SCORE” is empty.

Source Preview		Partition Field/Value	Map Fields/Values	Match Records	Import Status
<b>Tables</b>	<b>Records</b>				
CUSTOMERS_UPDATE	AUTOMERGE_ID	GOLDEN_RECORD	MATCHING_GROUP	MATCHING_SCORE	MDM_ID
CUSTOMER_ELEMENT	ITELO,301185	1	2		189431
	ITELO,301200	1	1		189442

2. Continue to the “Map Fields/Values”- Tab to perform the necessary field mappings. We map the following fields

AUTOMERGE\_ID

GOLDEN\_RECORD

MATCHING\_SCORE

MATCHING\_GROUP

3. Once you are done with the field and value mapping, switch to the “Match Records” Tab and select the field “AUTOMERGE\_ID” as matching relevant field and import the records.

Value matching		Default import actions			
Mapped destination fields:	Matching fields:	Active	Match Level	Match Type	Default Import Action
Matching Group	AutoMergeID [Key] [All]	0 of 0	None	None	Skip
Matching Score		2 of 2	Single	Exact	Update (All Mapped Fields)
		0 of 0	Single	Partial	Skip
		0 of 0	Single	Conflict	Skip
		0 of 0	Multiple	Exact	Skip
		0 of 0	Multiple	Partial	Skip
		0 of 0	Multiple	Conflict	Skip
		0 of 0	Filtered	Source	Skip

## 4.8 Consolidate Master Data in the MDM Data Manager

In this step you will use the MDM Data Manager to manually merge the customer records. Remember, new and updated records including their matching relevant info has been imported previously. This matching relevant info (MATCHING GROUP, MATCHING SCORE) is now used for further consolidation.

1. Click on the **Matching Group** column header to sort the record list by the matching group and to display only those have a matching group assigned. For all those records, the SAP Business Objects Data Services matching engine has found potential duplicates in the migration file as well as in the existing data in the MDM repository.

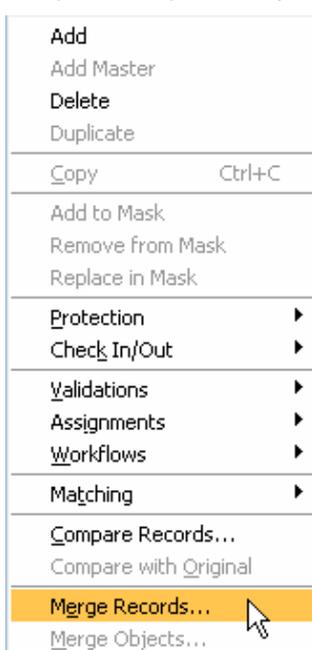


Name 2	Matching Group
Dagmar Schulze	1
	SAP
	SAP

2. Let us have a closer look at the records belonging to the matching group 1. Select both Becker Berlin companies (**Matching Group = 1**) with the help of the control key.

Records											
ID	Name 1	Search Term 1	Street	Matching Score	Matching Group	Golden Record	City	Postal Co			
189494	Tucem AG	TECUM	Albersloher Str. 1					48155			
189496	SAP AG	SAP	Dietmar-Hopp-Str. 20					69190			
189487	Berlin Becker GmbH	Becker	Calvinstr. 1	97	1	NO		10557			
189495	Becker Berlin GmbH	Becker	Calvinstr. 1		1	YES		10557			
189431	Panorama Studios Inc.	Panorama	Fountain Ave 1		2	YES		91602			
189485	Panorama Inc.	Panorama	Fountain A 1	90	2	NO		91602			
189486	Panorama St. Inc.	Panorama	Fountain Avenue 1	93	2	NO		91602			

3. Right click on the selected records and choose **Merge Records...** from the context menu to compare and potentially merge them.



4. In the Merge records screen, both records are displayed in a comparison view. Select the values of the Golden Record, marked with Golden Record flag = Yes. Those values will be defined as values of the „new“/combined record.

	[Merged Record]	189495	189487
Account Group		EU04, Intercompany Creditor	EU04, Intercompany Creditor
Name 1		Becker Berlin GmbH	Berlin Becker GmbH
Name 2			
Search Term 1		Becker	Becker
Search Term 2			
Street		Calvinstr. 1	Calvinstr. 1
Matching Score			97
Matching Group		1	1
Golden Record		YES	NO
City Postal Code		10557	10557
City		Berlin	Berlin
PO Box			
PO Box Postal Code			
Region		DE, Germany, Berlin	DE, Germany, Berlin
Country		DE, Germany	DE, Germany
VAT Registration Number			
Tax Number			
Telefon		+49 30 88530	+49 30 88530
Fax			
DUNS Number			
Language		German	German
ImportID		189495	189487
BO Street Name			
BO Street Number			
BO Break Key		1CDE	1CDE
AutoMergeID		XTELO,880033	XTELO,880022

5. Right click on the column header of the **Golden Record** and choose **Set All** from the context menu to set all values of this specific record.

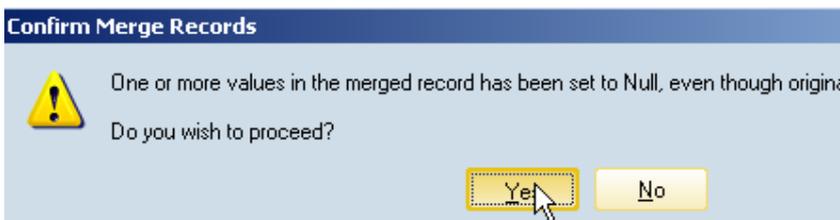
	[Merged Record]	189
Account Group		EU04, Intercom
Name 1		Becker Berlin
Name 2		
Search Term 1		Becker
Search Term 2		
Street		Calvinstr. 1
Matching Score		
Matching Group		1
Golden Record		YES

- Hide
- Unhide
- Sort
- Width
- Set All (Equal)
- Set All (Equal or NULL)
- Set All**
- Set with Non-NULLs Only
- Set NULLs Only
- Clear All

6. Click on **OK**.

	[Merged Record]	189495	189487
Account Group	EU04, Intercompany C...	EU04, Intercompany C...	EU04, Intercompany C...
Name 1	Becker Berlin GmbH	Becker Berlin GmbH	Berlin Becker GmbH
Name 2			
Search Term 1	Becker	Becker	Becker
Search Term 2			
Street	Calvinstr. 1	Calvinstr. 1	Calvinstr. 1
Matching Score			97
Matching Group	1	1	1
Golden Record	YES	YES	NO
City Postal Code	10557	10557	10557
City	Berlin	Berlin	Berlin
PO Box			
PO Box Postal Code			
Region	DE, Germany, Berlin	DE, Germany, Berlin	DE, Germany, Berlin
Country	DE, Germany	DE, Germany	DE, Germany
VAT Registration Number			
Tax Number			
Telefon	+49 30 88530	+49 30 88530	+49 30 88530
Fax			
DUNS Number			
Language	German	German	German
ImportID	189495	189495	189487
BO Street Name			
BO Street Number			
BO Break Key	1CDE	1CDE	1CDE
AutoMergeID	XTELO,880033	XTELO,880033	XTELO,880022

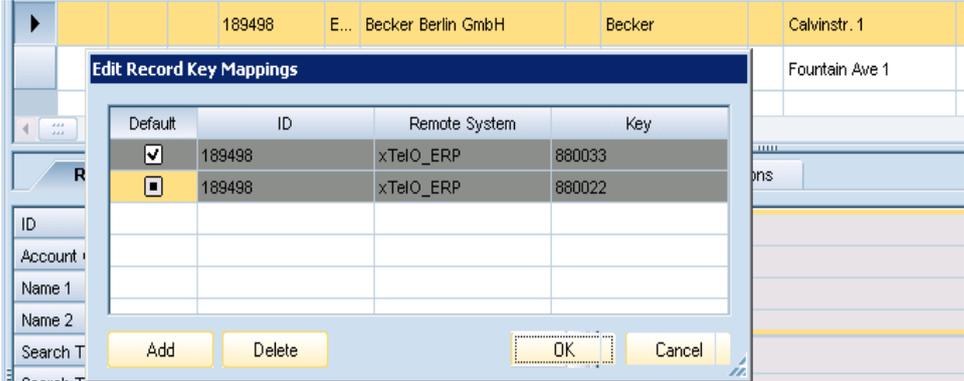
7. A pop-up appears. Click on **Yes** to proceed and to confirm the merging of the two records.



8. Right click on the merged record (automatically selected; at the bottom) and choose **Edit Key Mappings** from the context menu.

	189429	EU03, Internation	bH	Saskia Sommer
	189428	EU02, EU Credit	A	Sophie Ribery
	189427	EU01, National C	A	Victor Sanchez
	189484	EU02, EU Credit	Becker Berlin GmbH	Dagmar Schulze

9. The key mapping information of the merged record contains two entries now.



Click on **OK** to close the key mapping dialog box.

[www.sdn.sap.com/irj/sdn/howtoguides](http://www.sdn.sap.com/irj/sdn/howtoguides)