



# HOW TO GUIDE

SAP MDM and SAP Business Objects  
DS Integration

SAP NetWeaver®  
MDM 5.5 SP06

SAP Business  
Objects XI3.0

Document Version 1.02



SAP AG  
Dietmar-Hopp-Allee 16  
69190 Walldorf  
Germany  
T +49/18 05/34 34 24  
F +49/18 05/34 34 20  
[www.sap.com](http://www.sap.com)

© Copyright 2007 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, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, 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<sup>®</sup>, 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.

#### **Disclaimer**

Some components of this product are based on Java<sup>™</sup>. 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<sup>™</sup> 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.

#### **Documentation on SAP Service Marketplace**

You can find this documentation at  
<https://service.sap.com/installMDM>

## Typographic Conventions

## Icons

Type Style	Represents	Icon	Meaning
<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.		Caution
	Cross-references to other documentation.		Example
<b>Example text</b>	Emphasized words or phrases in body text, graphic titles, and table titles.		Note
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.		Recommendation
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.		Syntax
<b>Example text</b>	Exact user entry. 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.		

# Contents

<b>SAP NETWEAVER MDM AND SAP BUSINESSOBJECTS DS INTEGRATION GUIDE .....</b>	<b>1</b>
<b>1 INTRODUCTION.....</b>	<b>1</b>
<b>2 MOTIVATIONS FOR INTEGRATION.....</b>	<b>4</b>
<b>3 INSTALLATION.....</b>	<b>5</b>
3.1 SAP NetWeaver® MDM Installation.....	5
3.2 SAP Business Objects Data Services Installation.....	5
<b>4 OVERVIEW.....</b>	<b>6</b>
4.1 Design Time Integration.....	6
4.2 Runtime Integration .....	6
<b>5 INBOUND SCENARIO.....</b>	<b>7</b>
5.1 Design Time Integration.....	7
5.1.1 Creating an XML Schema File.....	7
5.1.2 Creating Data Services Mapping .....	7
5.1.3 Creating MDM Mapping.....	8
5.1.4 Creating Format Transformation in Data Services.....	8
5.2 Runtime Integration .....	9
5.2.1 Defining an MDM Port.....	10
5.2.2 Creating a Data Services Job .....	10
<b>6 OUTBOUND SCENARIO.....</b>	<b>12</b>
6.1 Design Time Integration.....	12
6.1.1 Creating an XML Schema File.....	12
6.1.2 Creating Data Services Mapping .....	12
6.1.3 Creating MDM Mapping.....	13
6.1.4 Creating Format Transformation in Data Services.....	13
6.2 Runtime Integration .....	14
6.2.1 Defining an MDM Port.....	14
6.2.2 External Method.....	15
6.2.3 Internal Method .....	16
<b>7 MONITORING.....</b>	<b>18</b>
7.1 Monitoring Data Services.....	18
7.2 Monitoring the MDM Syndication Server .....	18
7.3 Monitoring the MDM Import Server .....	18
<b>8 ENABLING AN INCREMENTAL UPDATE.....</b>	<b>20</b>
8.1 Delta Load of DataFlow .....	21
8.2 Delta import.....	22
8.3 Delete rows .....	22
<b>9 APPENDIX.....</b>	<b>23</b>
9.1 Defining a Data Services Job as a Web Service.....	23
9.2 Syndication Command Line Code Sample .....	23
9.2.1 Compilation.....	23

9.2.2	Installation.....	23
9.2.3	Execution.....	24

# SAP NetWeaver MDM and SAP BusinessObjects DS Integration Guide

## 1 Introduction

Integrating SAP NetWeaver MDM and SAP BusinessObjects Data Services (DS) with its ETL capabilities can enrich the MDM solution for any enterprise. Using the highly scalable data integration solution provided by SAP Business Objects, together with SAP NetWeaver MDM, an enterprise can access, integrate, transform, and move any type of master data, from multiple sources to any target system in real time.

This guide describes how to perform ETL integration between SAP NetWeaver MDM and SAP BusinessObjects Data Services, with emphasis on:

- Integration scenarios
- Design time integration
- Runtime integration
- Monitoring



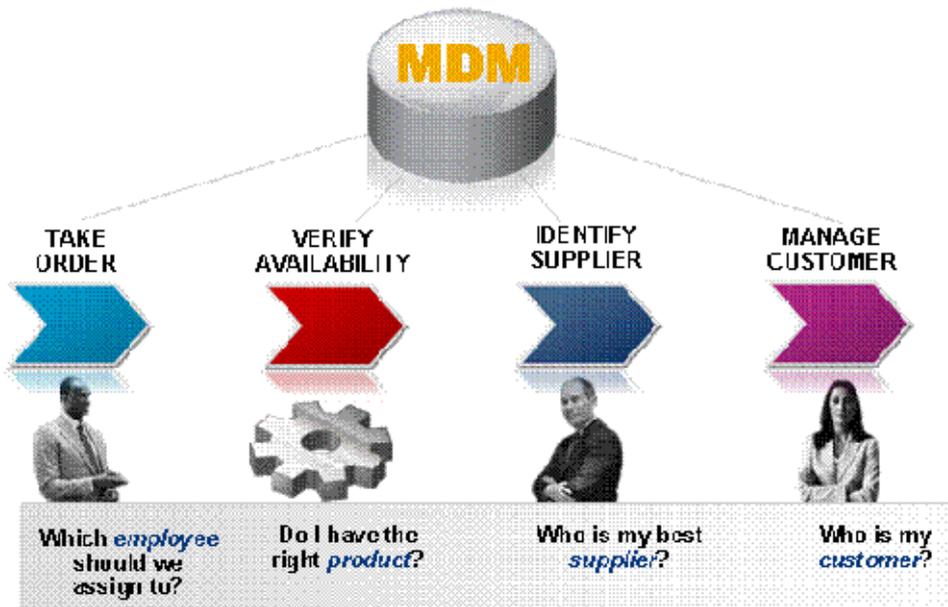
Code samples are provided at: [SAP MDM and SAP Business Objects DS Integration Source Files](#)

### SAP NetWeaver MDM

**SAP NetWeaver MDM** plays a critical role in enterprises by providing a truly harmonized hub of consistent, accurate, timely master data that is accessible across the enterprise.

MDM enables organizations to consolidate and harmonize their master data (for example, customers, employees, products, and suppliers) within heterogeneous IT landscapes. With SAP NetWeaver MDM, organizations can vastly reduce data maintenance costs, ensure cross-system data consistency, accelerate the execution of business processes, and greatly improve decision-making. In this context, SAP NetWeaver MDM is indispensable within an enterprise service-oriented architecture (enterprise SOA).

SAP NetWeaver MDM ensures that questions about master data as specified below are answered so that business processes can run smoothly and business decisions are well informed.



## SAP BusinessObjects Data Services

SAP BusinessObjects Data Services is a product that combines the following two SAP Business Objects products:

- **Data Integration**

Obtains a complete view of your business with tools that allow you to access any type of data (structured and unstructured) from virtually any source; integrate data through extraction, transformation and loading (ETL); and deliver reliable data.

This guide describes how to perform integration between *Data Integration* and SAP NetWeaver MDM.

- **Data Quality**

Improves information accuracy by monitoring, analyzing and reporting on information across your organization

Integration of *Data Quality* capabilities is covered in the *MDM Enrichment Architecture Guide* which can be accessed at <http://service.sap.com/installMDM> → Operations

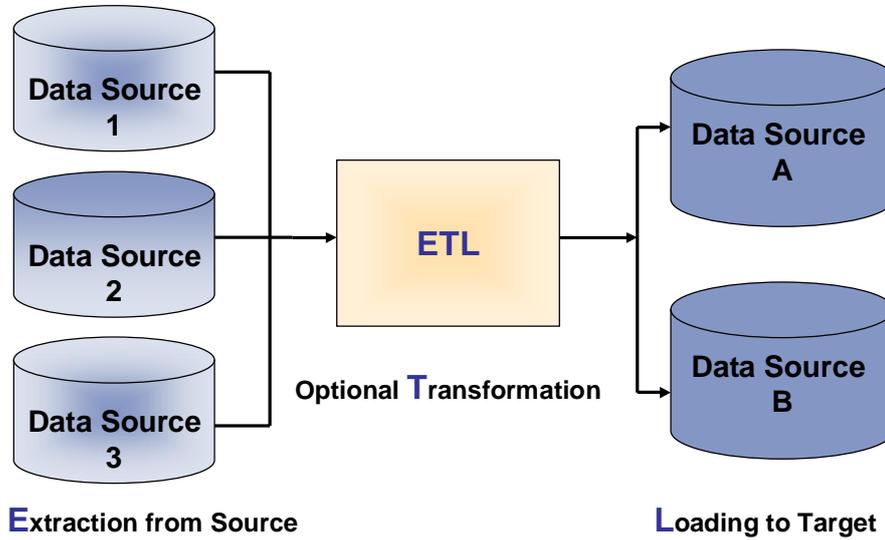
Applications often need to process information originating from various data sources, including XML documents, relational databases, and various legacy systems. Data integration is the process of combining data residing at these different sources and providing the user with a unified view of the data in real time so they can realize its full value to the business. These capabilities are provided by SAP Business Objects' standalone ETL tool.

ETL is short for *extract, transform, load*, and these three functionalities are combined into one tool to pull data out of data sources (source), optionally transform it, and place it into target data sources.

- Extract – the process of reading data from a data source.
- Transform – the process of converting the extracted data from its previous form into the form it needs to be so that it can be placed into another data source (target).

Transformation occurs by using rules or lookup tables or by combining the data with other data.

- Load – the process of writing the data into the target data source



Information from the source databases is extracted, transformed and then loaded into the target data sources

## 2 Motivations for Integration

SAP NetWeaver MDM has ETL capabilities. Import and Syndicator tools enable customers to load and extract data to and from MDM using files and relational databases as a source/target. Transformation capabilities are available both in the Import and Syndicator tools. SAP NetWeaver MDM introduces the port mechanism of the MDM Import Server and MDM Syndication Server for batch processing of Import and Syndication jobs respectively. Furthermore, SAP NetWeaver MDM enables activation of a Syndicator batch job from its workflow.

SAP BusinessObjects Data Services is an ETL tool which has built-in transformation capabilities. Custom transformations can also easily be created. It supports various data sources including files, XML, relational databases, applications, web services and mainframes and enables complex ETL maps (data flows) that can combine several sources and/or several targets in one map. It also has a batch workflow capability that enables creation of complex ETL jobs.

SAP NetWeaver MDM customers can benefit by integrating with SAP BusinessObjects Data Services in the following areas:

- Enriching the set of data source types that can be used for Import and Syndication since SAP BusinessObjects Data Services supports various data sources.
- Enriching the set of available transformations that can be used in the process.
- Easing the Import and Syndication process by moving some bottleneck transformations from the MDM Import Server/MDM Syndication Server to Data Services and therefore improving the overall performance.
- Enabling more complex ETL jobs that require several data sources.  
For example, extracting/receiving a complete set of data from a data source that does not support *Change Data Capture* (CDC) and importing only the delta.

### Use Cases

- Integrating with non-SAP systems such as Oracle Applications, Siebel, PeopleSoft and more.

SAP BusinessObjects Data Services provides predefined content and packaged data integration for enterprise applications such as Oracle, Siebel, PeopleSoft, etc. in the form of Rapid Marts

For more information on *Rapid Marts*, see BusinessObjects Rapid Marts at [http://www.businessobjects.com/product/catalog/rapid\\_marts/](http://www.businessobjects.com/product/catalog/rapid_marts/)

- Initial load scenario

When starting a new MDM implementation or integrating a new remote system to existing MDM implementations, a massive data load into MDM is carried out. The ETL capabilities of SAP BusinessObjects Data Services can be used to benefit this process.

- Enabling an incremental update

When receiving a complete set of data within a file, a Data Services Job is used to generate an import file that contains only the updated records.

For example, if a manufacturer gives a full list of its products to all distributors, and only part of the product data in that list is updated, then Data Services can be used to generate an import file that contains only the updated records that are relevant to this distributor (according to any business rule).

## 3 Installation

In the following installation guides, you can find comprehensive information about the installation and upgrade of SAP Master Data Management and the SAP BusinessObjects Data Services. We recommend that you use the documents available here. The guides are updated regularly.

### 3.1 SAP NetWeaver® MDM Installation

#### MDM Installation Guide

As well as the information provided in this guide, you can find the most recent information about the technical implementation of SAP MDM 5.5 and the latest installation and configuration guides on the SAP Service Marketplace at: [service.sap.com/installMDM](http://service.sap.com/installMDM).

### 3.2 SAP Business Objects Data Services Installation

#### Data Services Installation Guide

This guide provides an overview of the SAP BusinessObjects Data Services platform and architecture. It also describes how to upgrade and install Data Services. This guide is available at <http://help.sap.com> → *Business Objects* → *Data Services Installation Guide*

## 4 Overview

There are two possible scenarios for integrating SAP NetWeaver MDM and SAP BusinessObjects Data Services:

- Inbound Scenario  
Data Services is integrated with the MDM Import Manager and MDM Import Server
- Outbound Scenario  
Data Services is integrated with the MDM Syndicator and MDM Syndicator Server

Both scenarios include two components – design time integration and runtime integration.



### 4.1 Design Time Integration

During design time integration, the customer creates mapping in both MDM (Import Manager/Syndicator) and the Data Services Designer. These two maps are complementary; one of the maps creates an output file which the other utilizes as an input. XML files are recommended for the intermediate file because MDM data structure is not flat in nature. Both sides (MDM and Data Services), share the same file format.

In most cases, the Data Services map contains the structural mapping while the Import/Syndication map contains value mapping and update strategy rules. The intermediate file format should be generated from the repository or created manually as an XML schema (XSD) file.

There are some cases in which the mapping is simple and where it is recommended to perform format transformation in Data Services (source to XML), and to do all the mapping through Import/Syndication. In these cases the XML schema (XSD) file is generated using the Data Services designer capability of creating XSD.

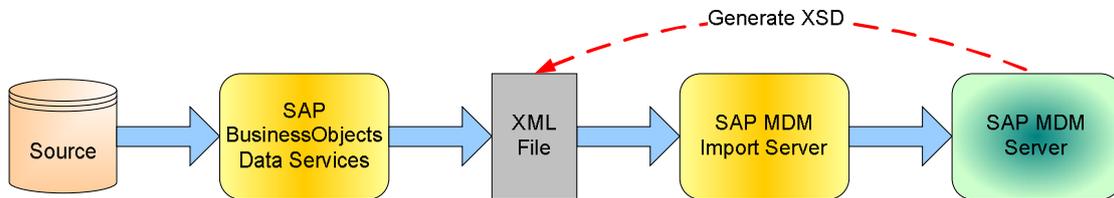
### 4.2 Runtime Integration

The runtime integration merges the MDM port mechanism and the Data Services batch workflow capabilities. In the inbound scenario, Data Services starts the process by extracting data from a source(s) and then writing to a file. When successfully completed, the file is moved to the relevant port and is processed by the Import Server.

During runtime integration, the customer can monitor the complete process – in the inbound scenarios from the source system to MDM, and vice versa in the outbound scenarios.

## 5 Inbound Scenario

This chapter describes how to perform integration of Data Services and the MDM Import Manager/Server for an inbound scenario.



### Inbound Scenario Overview

## 5.1 Design Time Integration

Design time integration includes three steps:

1. Creating an XML schema file that represents an MDM schema or subset of the schema
2. Creating Data Services mapping into an XML schema file
3. Creating MDM mapping from the same XML schema file

### 5.1.1 Creating an XML Schema File

Creating an XML Schema (XSD) as an intermediate file format can be accomplished by using existing SAP consulting utilities or by creating an XSD file from an existing XML file that represents the relevant MDM table.

### 5.1.2 Creating Data Services Mapping

Data Services Mapping is carried out via the *SAP BusinessObjects Data Services Designer* application as follows:

1. Create a new project – from the menu, choose *Project → New → Project*.
2. Create a batch job – in the project area, right-click the project and choose *New Batch Job*.
3. Create a workflow – open the job, choose *Workflow* from the palette window (on the right), and create it in the job.
4. Create a data flow – open the workflow, choose *Dataflow* from the palette window (on the right), and create it in the workflow.
5. Create a source (can be a datastore or file format).
  - To create a datastore, from the menu, choose *Project → New → Datastore* (according to the source data type).
  - To create a file format, from the menu, choose *Project → New → File Format*.
6. Create an XML Schema format:
  - a. From the menu, choose *Project → New → XML Schema*. An XML Schema format dialog opens.

- b. Populate the following fields and click OK.

Input field Name	Value
<i>Format name</i>	Name of the format
<i>File name/URL</i>	Browse for the XSD file
<i>Root element name</i>	XSD root element name

- c. Add the newly created XML schema as a target to the data flow.
7. Map source to target – perform mapping between source and target using Data Services transformations.
- For more information on how to create Data Services mapping, see the *Data Integrator Designer Guide* which is available at <http://help.sap.com> → *Business Objects*.
8. Create an XML sample file – add a condition in the *Query Transform Where* tab to get the correct sample data and run the job.

### 5.1.3 Creating MDM Mapping

MDM Mapping is carried out via the SAP MDM Import Manager application as follows:

1. Define the (XSD) XML schema using the MDM Console:
  - a. In the MDM Console, expand *Admin* in the repository tree view and choose *XML schemas*.
  - b. Right-click in the XML schemas window and choose *Add XML schema*.
  - c. In the file name field, browse for the full path of the XML schema file created in Data Services.

For more information, see the section on *XML Schemas Table* in the *MDM Console Reference Guide* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

2. Create the import map:
  - a. Open the Import Manager using the XML schema file and with the relevant remote system.
  - b. Create mapping in the Import Manager from the XML schema file as a source to the MDM repository.

For more information, see the *MDM Import Manager Reference Guide* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

### 5.1.4 Creating Format Transformation in Data Services

When mapping is uncomplicated, we recommend that you perform format transformation in Data Services (source to XML) and do all the mapping in *Import*. In this case the intermediate file format will not be generated from the MDM repository, but will be generated according to the source structure.

Follow these steps to create *Data Services Format Transformation* and to generate XSD:

1. Create a new project – from the menu, choose *Project* → *New* → *Project*.

2. Create a batch job – in the project area, right-click the project and choose *New Batch Job*.
3. Create a workflow – open the job, choose *Workflow* from the palette window (on the right), and create it in the job.
4. Create a data flow – open the workflow, choose *Dataflow* from the palette window (on the right), and create it in the workflow.
5. Create a source (can be datastore or file format).
  - To create a datastore, from the menu, choose *Project* → *New* → *Datastore* (according to the source data type).
  - To create a file format, from the menu, choose *Project* → *New* → *File Format*.
6. Map the source datastore to an XML template as follows:
  - a. Open the data flow, choose the datastore, and apply it in the data flow as a source.
  - b. Open the datastore (double-click), and set the appropriate attributes (if relevant).
  - c. Choose *Query Transform* from the palette window (on the right), and apply it in the data flow.
  - d. Connect the datastore to the *Query Transform*.
  - e. Open the *Query Transform* (double-click), and drag the source tree to the target.
  - f. Add an XML template from the palette window (on the right), to the data flow.
  - g. Connect the *Query Transform* to the XML template.
  - h. Open the XML template (double-click), and set the appropriate attributes.
7. Generate an XML schema – in the XML template, right-click on the target top level and choose *Generate XML Schema*.
8. Create an XML sample file – add a condition in the *Query Transform Where* tab to get the correct sample data and run the job.

## 5.2 Runtime Integration

At runtime, the Data Services job starts a Data Services workflow. It first executes the Data Services data flow. Upon success, it moves the output file into the appropriate directory (MDM port). The MDM Import Server performs an import from this file.

Runtime integration includes two steps:

1. Creating an MDM port for the relevant map and remote system
2. Extending the Data Services workflow to copy the output file into the appropriate directory (MDM port). The Data Services workflow performs this operation only if the Data Services data flow was executed successfully.

## 5.2.1 Defining an MDM Port

1. In the MDM Console, expand *Admin* in the repository tree view and choose *Ports*.
2. Right-click in the *Ports* window and choose *Add Port*.
3. Enter the following input for the relevant parameters:

Parameter	Input
<i>Name</i>	Name of port
<i>Code</i>	Code name of port
<i>Type</i>	Inbound
<i>Remote System</i>	Remote system name
<i>Map</i>	Map name
<i>Format</i>	XML schema
<i>XML Schema</i>	Name of XML schema
<i>Processing type</i>	Automatic

For more information on how to create an MDM port, see the *MDM Console Reference Guide* → *Ports Table* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

## 5.2.2 Creating a Data Services Job

This section describes how to extend the Data Services workflow that we have already created into an integrative workflow.

For more information, see [Creating Data Services Mapping](#)

After successful completion of the Data Services data flow, the output file can be copied or moved into the MDM port which we created above.

For more information, see [Defining an MDM Port](#)



### Inbound Data Services Workflow

1. Open the Data Services workflow and surround the *Data Flow* with both a *Try* and *Catch* workflow step.
2. Open the *Catch* step and make sure that it is set to *catch all exceptions*.
3. Add a script to the *Catch* step.
4. Open the script and add what actions take place upon failure. At the end, insert a `raise_exception()` function which in case of exception, fails the entire workflow and stops execution.
5. Add a script after the *Catch* step which stipulates that upon success, the file will be moved or copied to the appropriate folder according to the MDM port.

Examples:

Copy File : `exec('cmd', 'copy "C:\OrigDir\File.out" "C:\NewDir\File.out", 8 ) ;`

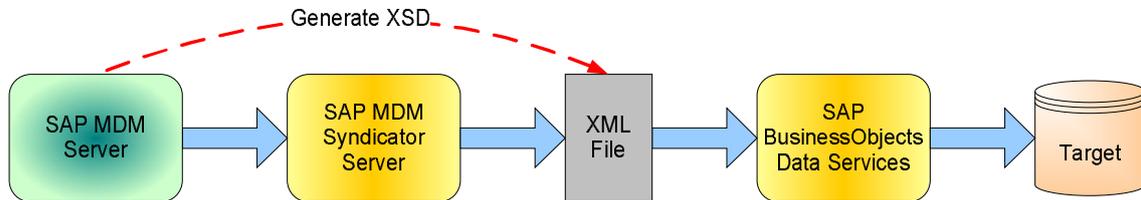
Move File: `exec('cmd', 'move "C:\OrigDir\File.out" "C:\NewDir\File.out", 8 ) ;`



We recommend that you copy the file into the directory with a different extension and rename it, or copy and move it to a different local folder.

## 6 Outbound Scenario

This chapter describes how to perform integration of Data Services and the MDM Syndicator Server for an outbound scenario.



### Outbound scenario overview

## 6.1 Design Time Integration

Design time integration includes three steps:

1. Creating an XML schema file
2. Creating Data Services mapping from the XML schema file
3. Creating MDM mapping from the same XML schema file

### 6.1.1 Creating an XML Schema File

Creating an XML Schema (XSD) as an intermediate file format can be accomplished by using existing SAP consulting utilities or by creating an XSD file from an existing XML file that represents the relevant MDM table.

### 6.1.2 Creating Data Services Mapping

Data Services Mapping is carried out via the SAP BusinessObjects Data Services Designer application as follows:

1. Create a new project – from the menu, choose *Project* → *New* → *Project*.
2. Create a batch job – in the project area, right-click the project and choose *New Batch Job*.
3. Create a workflow – open the job, choose *Workflow* from the palette window (on the right), and create it in the job.
4. Create a data flow – open the workflow, choose *Dataflow* from the palette window (on the right), and create it in the workflow.
5. Create a target (can be a datastore or file format).
  - To create a datastore from the menu, choose *Project* → *New* → *Datastore* (according to the target data type).
  - To create a file format from the menu, choose *Project* → *New* → *File Format*.
6. Create an XML Schema format:
  - a. From the menu, choose *Project* → *New* → *XML Schema*. An XML Schema format dialog opens.

- b. Populate the following fields and click OK.

Input field Name	Value
<i>Format name</i>	Name of the format
<i>File name/URL</i>	Browse for the XSD file
<i>Root element name</i>	XSD root element name

- c. Add the newly created XML schema as a source to the data flow.

7. Map source to target – perform mapping between source and target using Data Services transformations.

For more information on how to create Data Services mapping, see the *Data Integrator Designer Guide* which is available at <http://help.sap.com> → *Business Objects*.

### 6.1.3 Creating MDM Mapping

MDM Mapping is carried out via the SAP MDM Syndicator application as follows:

1. Define the (XSD) XML schema using the MDM Console:
  - a. In the MDM Console, expand *Admin* in the repository tree view and choose *XML schemas*.
  - b. Right-click in the XML schemas window and choose *Add XML schema*.
  - c. In the file name field, browse for the full path of the XML schema file created in Data Services.

For more information, see the *MDM Console Reference Guide* → *XML Schemas Table* on the SAP Service Marketplace at <https://websmp104.sap-ag.de/installmdm> → *Reference Documentation*.

2. Create the syndicator map:
  - a. Open the Syndicator using the XML schema file and with the relevant remote system.
  - b. Create mapping in the Syndicator using the XML schema file as a target.

For more information, see the *MDM Syndicator Reference Guide* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

### 6.1.4 Creating Format Transformation in Data Services

When mapping is uncomplicated, you can perform format transformation in Data Services (source to XML) and do all the mapping in the Syndicator. In this case the intermediate file format will not be generated from the MDM repository, but will be generated according to the target structure.

Follow these steps to create *Data Services Format Transformation* and to generate XSD:

1. Create a new project – from the menu, choose *Project* → *New* → *Project*.
2. Create a batch job – in the project area, right-click the project and choose *New Batch Job*.
3. Create a workflow – open the job, choose *Workflow* from the palette window (on the right), and create it in the job.
4. Create a data flow – open the workflow, choose *Dataflow* from the palette window (on the right), and create it in the workflow.

5. Create a target (can be a datastore or file format).

To create a datastore, from the menu, choose *Project* → *New* → *Datastore* (according to the target data type)

To create a file format, choose *Project* → *New* → *File Format*

6. Map the target datastore to an XML schema as follows:

- a. Open the data flow, choose the datastore and apply it in the data flow as a target.
- b. Open the datastore (double-click), and set the appropriate attributes (if relevant).
- c. Generate an XML schema file out of the target datastore – in the datastore, right-click on the target top level and choose *Generate XML Schema*.
- d. Create the XML schema – from the menu, choose *Project* → *New* → *XML Schema*. Use the XML schema file that was generated in above step (c).
- e. Add the newly created XML schema as a source to the data flow.
- f. Open the XML schema (double-click), and set the appropriate attributes.
- g. Choose *Query Transform* from the palette window (on the right), and apply it in the data flow.
- h. Connect the source to the *Query Transform*.
- i. Open the *Query Transform* (double-click), and drag the source tree to the target.
- j. Connect the *Query Transform* to the target datastore.

## 6.2 Runtime Integration

At runtime there are two methods to initiate an MDM syndication command:

- Externally from Business Process Management (BPM)
- Internally from an MDM workflow

In both cases an MDM port should be defined.

### 6.2.1 Defining an MDM Port

1. In the MDM Console expand *Admin* in the repository tree view and choose *Ports*.
2. Right-click in the *Ports* window and choose *Add Port*.
3. Enter the following input for the relevant parameters:

Parameter	Input
<i>Name</i>	Name of port
<i>Code</i>	Code name of port
<i>Type</i>	Outbound
<i>Remote System</i>	Remote system name
<i>Map</i>	Map name
<i>Format</i>	XML schema

Parameter	Input
<i>XML Schema</i>	Name of XML schema
<i>Processing Type *</i>	Automatic/Manual

\* Set the *Processing Type* parameter according to the desired syndication method.

For more information on how to create an MDM port, see the *MDM Console Reference Guide* → *Ports Table* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

## 6.2.2 External Method

In the external method (BPM), a syndication command line or web service is used. Both are based on the Java API Syndication command and can be easily created.

A sample of the syndication command line code is shown below.

The process of initiating a syndication command using the *External* method is as follows:

1. Create an MDM Port.
2. Create a Data Services Job.

### 6.2.2.1 Defining an MDM Port

For more information, see [Defining an MDM Port](#)

### 6.2.2.2 Creating a Data Services Job

This section describes how to extend the Data Services workflow that we have already created into an integrative workflow.

For more information, see [Creating Data Services Mapping](#)

The workflow initiates syndication using a syndication command line. After successful completion, it starts the *Data Services Dataflow*. If the syndication fails, it raises an error and exits.



### Outbound External Data Services Workflow

1. Open the Data Services workflow and add a script.
2. Connect the script to the Data Flow.
3. Open the script and add to it an `exec` function that will execute the syndication command line.

For more information, see [Syndication Command Line Code Sample](#)

```
exec('cmd.exe', 'java -cp "C:\\Program Files\\SAP MDM\\SyndicateCmd\\SyndicateCmd.jar"
SyndicateCMD ... ', 1);
```



**SyndicateCMD ...** stands for complete syndication command

For more information, see [Execution](#)

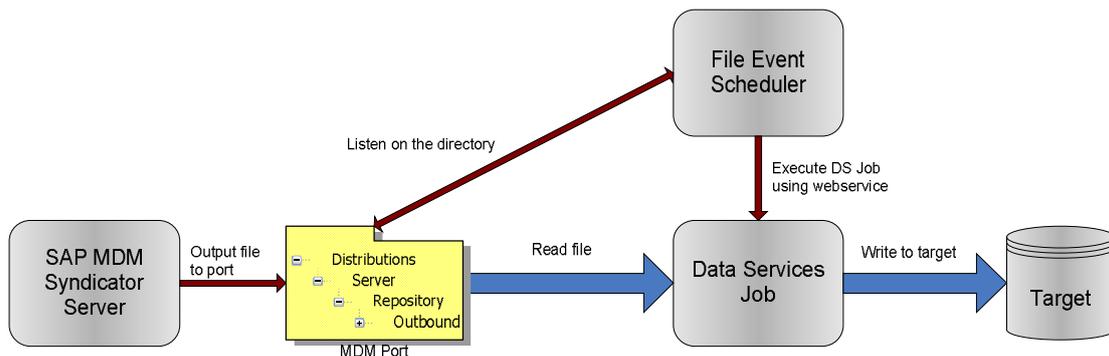


The syndication command line JAR file and the appropriate MDM Java API should be installed in the Data Services server machine. In the above example the syndication command line JAR file was installed in directory, *C:\Program Files\SAP MDM\SyndicateCmd*. Make sure that the MDM Java API JAR files are in the CLASSPATH environment variable or add them to Java – *cp* parameter.

### 6.2.3 Internal Method

In the internal method (internal MDM workflow), the syndication is initiated using an MDM workflow. A file trigger event mechanism listens to the appropriate directory (syndication port).

When a syndication output file is created, it activates the *Data Services Job* that processes the syndication output file. A trigger event mechanism is obtained either by using a third party external scheduler or by implementing such a mechanism using Data Services.



#### Internal Workflow

The process of initiating a syndication command using the *Internal* method is as follows:

1. Create an MDM Port.
2. Create a Data Services Job.
3. Create a File Event Scheduler Job.

#### 6.2.3.1 Defining an MDM Port

For more information, see [Defining an MDM Port](#)

#### 6.2.3.2 Creating a Data Services Job

Wrap the Data Services mapping with a batch job.

For more information, see [Creating Data Services Mapping](#)

### 6.2.3.3 Creating a File Event Scheduler Job

There are two ways to create a file event scheduler job:

- Using a third party external scheduler that will listen to the relevant syndication directory and activate the *Data Services Job* using its web service interface

For more information, see [Defining a Data Services Job as a Web Service](#)

- Implementing a file trigger event mechanism in Data Services by creating a Data Services workflow that contains a `wait_for_file` function.

## 7 Monitoring

Monitoring in an integration environment is complex. The administrator needs to monitor in each tool separately and integrate the results to get a complete picture.

This chapter describes briefly how to monitor in each tool.

For more details, consult the relevant product manual which is referenced under each subsection below.

### 7.1 Monitoring Data Services

Using the *Administrator*, you can monitor job execution of any batch job in a connected repository. The *Batch Job Status* page lists each batch job execution. Use this list to view the overall status of each execution and to access more detailed statistics and log files.

To view overall status of executed jobs, select *batch* → *repository* in the Administrator Console. Information about job status, execution monitoring, trace files, and error logs is provided.

For more information, see the *Data Integrator Management Console: Administrator Guide* → *Batch Jobs* → *Monitoring jobs*

### 7.2 Monitoring the MDM Syndication Server

The MDM Syndication Server logs the progress and the results of each syndication job it undertakes. These log entries detail each step of the syndication process and can be monitored from the repository's *Reports* table in the MDM Console. Selecting a syndication report in the *Reports* table displays the report in the *Report Detail* pane.

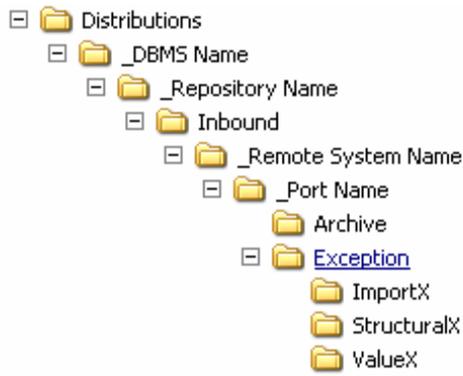
For more information, see the *MDM Syndicator Reference Guide* → *MDM Syndication Server (MDSS)* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

### 7.3 Monitoring the MDM Import Server

The MDM Import Server logs the progress and the results of each import job it undertakes. These log entries detail each step of the import process and can be monitored from the repository's *Reports* table in the MDM Console. Selecting an import report in the *Reports* table displays the report in the *Report Detail* pane.

The MDM Import Server indicates several exception types: *Structural Exceptions*, *Value and Import Exceptions*, and *Port Blocking*.

Exception files are placed in folders according to their type (*ImportX*, *StructuralX*, or *ValueX*). All exception folders belong to the fixed port directory structure.



### Exception Folder Directory Structure

A log file in the parent *Exception* folder points to exception files set aside during a particular import. The *StructuralX* folder contains the original source files placed in the port's *Ready* folder. The *ImportX* and *ValueX* folders contain the virtual extended records where an import or value exception was found.

- *Structural Exceptions*

Structural exceptions prevent the MDM Import Server from processing any import records in an import file. When a structural exception is found, the MDM Import Server moves the offending import file from the port's *Ready* folder to the port's *StructuralX* folder. No importing occurs and the port is blocked until the problem is resolved.

More information: *Port Blocking* (below)

- *Value and Import Exceptions*

Value and import exceptions prevent the MDM Import Server from importing a specific record. When such exceptions are encountered, an exception file for the offending record is copied to the *ValueX* or *ImportX* folder, respectively. The MDM Import Server then continues processing the import file and all other, non-exceptional records are imported as normal.

- *Port Blocking*

Any port containing a structural exception is blocked by default from any further import activity. When the structural exception is fixed, the import unblocks the port and processing resumes as normal. To turn off automatic port blocking, set the port's block on the *Structural Exceptions* property to **No** in the *Port Detail* pane on the MDM Console.

For more information, see the MDM Import Manager Reference Guide → *MDM Import Server (MDIS)* on the SAP Service Marketplace at <http://service.sap.com/installMDM> → *Reference Documentation*.

## 8 Enabling an Incremental Update

The fundamental principle of an incremental update is to periodically take a snapshot of the data in the source system and then load only the changes from the previous snapshot into the target.

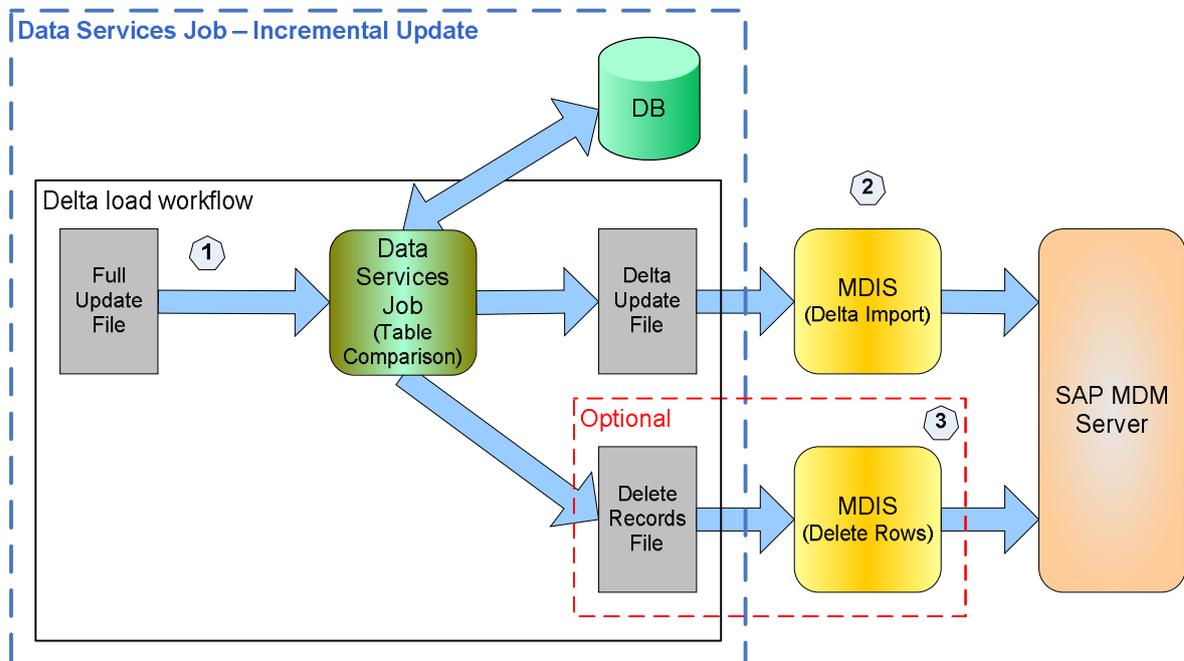
Incremental update is one of the use cases of ETL in general and SAP BusinessObjects Data Services in particular, in conjunction with the MDM Import Server. Some source systems can only produce a fully refreshed file and are incapable of producing a delta update file to only reflect the changes.

This method is known as the snapshot differential method which keeps the previous snapshot (*Full Update File*) in a relational table. Comparison of the records from the current *Full Update File* and the previous snapshot is carried out and according to the type of change, the record keys are matched to detect the type of change. Records that are inserted or updated are moved to the *Delta Update File* while the deleted records are moved to the *Delete Records File*.

A use case for a *Delta Update* file would be:

A customer receives a file consisting of a complete list of objects which he needs to import into MDM. For a large number of records, if the update rate is frequent, this import operation is a burden on the MDM server. To ease the load, the customer should import only the delta updated objects and not the complete list.

An example of such a use case could be an MDM implementation of a distributor who each day receives from each of his suppliers, a file with a list of all their products. This list may consist of hundreds of thousands or even millions of products. The updated rows are only about 5% of the entire number of rows in the file. It would therefore be more efficient to reduce the input file to reflect only the updated rows (delta update).



### Table Comparison Capture

The figure above shows an example of a *Data Services Job* of an incremental update with a *Table Comparison Capture*.

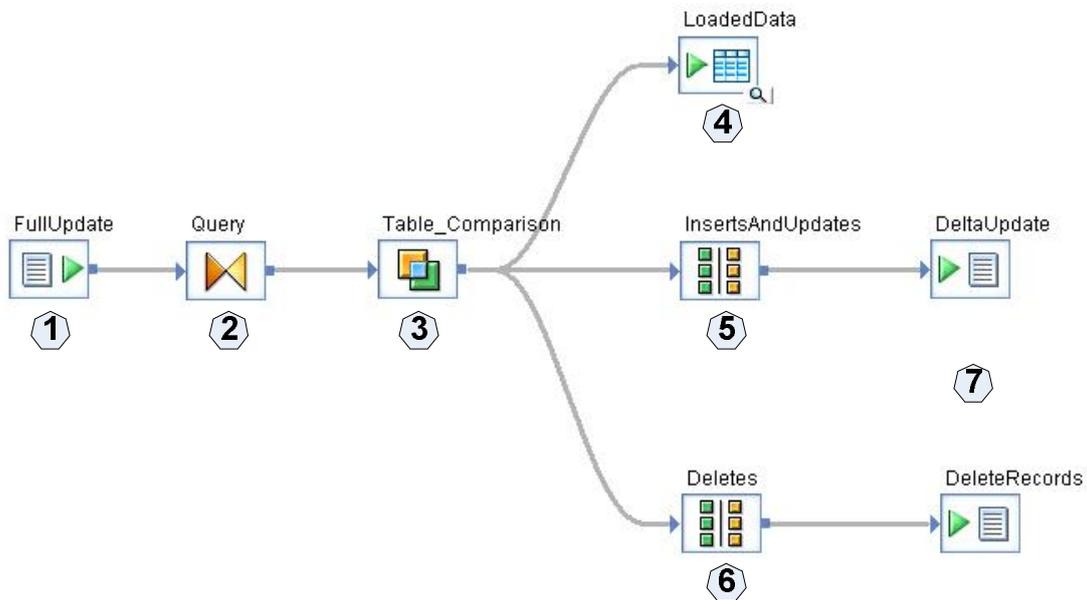
To enable an incremental update, there are three steps to carry out:

1. Delta load of *DataFlow*
2. Delta import
3. Delete rows (optional).

## 8.1 Delta Load of DataFlow

Compares the input file to the *LoadedData* table which is a relational table that contains the data from the previous load, and generates delta update files. These files should contain inserts and updates, and optionally delete record files containing keys of rows to delete. Apart from creating the delta update and deleting the record files, it updates the *LoadedData* table.

For performance sake, make sure that the *LoadedData* table is as limited as possible and contains only the key fields and the comparable data. In some cases the comparable data could be simply a time stamp of the last update and in extreme cases it could be the whole table.



### Delta Load DataFlow for Table Comparison Capture

Steps to create *Delta Load DataFlow*:

1. Create a new file source for the *FullUpdate* file.
2. Create a *Query* step and map it to the source.
3. Create a *Table Comparison* step as follows:
  - a. In the *Table name* field, choose the *LoadedData* table from RDBMS.
  - b. Select the checkbox *Detect deleted rows from comparison table*.
  - c. Select *Row by row* as the comparison method.
  - d. In the *Input primary key columns* box, add the table primary key.
  - e. Leave the *Compare columns* box empty.

4. Create a target for the *LoadedData* table to update it with the new data.
5. Select *InsertsAndUpdates* and create a map operation step with the following settings:

Input row type	Output row type
normal	discard
update	normal
insert	normal
delete	discard

6. Optionally, select *Deletes* and create a map operation step with the following settings:

Input row type	Output row type
normal	discard
update	discard
insert	discard
delete	normal

7. Create a file target for the *DeltaUpdate* and optionally for *DeleteRecords*.

## 8.2 Delta import

Import the job that takes the delta update files and imports them into MDM.

## 8.3 Delete rows

Import the job that takes the delete record files and deletes them from the MDM rows that are included in this file.

## 9 Appendix

### 9.1 Defining a Data Services Job as a Web Service

SAP BusinessObjects Data Services enables creation of web services from *Data Services Jobs*. It is a simple procedure that you can perform from the Data Services Management Console.

Follow these steps to create a web service from a *Data Services Job*:

1. Open the Data Services Management Console and login.
2. Choose *Administrator*.
3. In the tree view on the left, choose *Web Services*.
4. Choose the web services configuration tab.
5. In the dropdown list, select *Add Batch Job* and click *Apply*.
6. Choose a repository and select one or more batch jobs.
7. Click *Add* to create the web service(s).

To use these web services follow this URL to get the WSDL:

`http://<host>:<port>/DataServices/servlet/webservices?ver=2.0&wsdlxml`

Default port number is 28080.

For more information, see the *Data Integrator Management Console: Administrator Guide* → *Support for Web Services* → *Server functionality*.

### 9.2 Syndication Command Line Code Sample

The following is a java application code sample for a syndication command line.

This sample code implements a synchronized *Command* line for syndication. It was tested in MDM 5.5 SP6 patch 2.

The sample code contains three classes, the main class is `SyndicateCMD`, and there are two support classes called `SyndicationListener` and `SyndicateArguments`.

The main class gets repository credentials, port name, remote system, output file prefix, and time out. It initiates a syndication job which is assigned to the given port and remote system. It registers for notification in the MDM server and waits until the syndication job is finished. When the syndication has finished (successfully or unsuccessfully), the command line returns with a status code 0 for success, and a different value for fail.

#### 9.2.1 Compilation

1. Take the three Java files and put them in a package called *Syndicate*.
2. Compile them and include the MDM Java API of the relevant version you are using.
3. Create a JAR file called *Syndicate.jar*.

#### 9.2.2 Installation

1. Install the MDM Java API of the relevant version.
2. Copy the JAR file *Syndicate.jar* that was created in the previous section.

### 9.2.3 Execution

Prerequisites for execution:

- MDS, MDIS and JAVA API are installed and share the same version.
- The MDM repository is loaded.
- The syndication server is running.
- The syndication port is set up and is ready for syndication.
- Both MDM Java API JAR files and *Syndicate.jar* are in the class path.

**Command line usage:**

```
Syndicate MDMServer RepositoryName UserName RemoteSystem Port  
FilePrefix LanguageName TimeOut Password
```

The table below provides an explanation of each of the parameters above:

MDMServer	MDM server name or IP address
RepositoryName	Repository name
UserName	MDM repository user name
RemoteSystem	Remote system name
Port	MDM syndication port name (associated with the remote system)
FilePrefix	Syndication output file prefix
LanguageName	Language connection name. For example, "English [US]"
TimeOut	Time out for the command (in seconds) to wait for the syndication to end
Password	MDM password (optional parameter if the password is empty)

To view code samples, see: [SAP MDM and SAP Business Objects DS Integration Source Files](#)