



Welcome to your RKT Live Expert Session

SAP NetWeaver MDM 5.5 Java API

Andreas Seifried
SAP NetWeaver Product Management

Please note that we are recording this session!

Your Instructor

Andreas Seifried
SAP NetWeaver MDM Product Manager

**This Live Expert Session is
no replacement for the
RKT MDM 5.5 Java API workshop !**

Required:

- **Basic Java Programming Skills**
- **SAP MDM 5.5 Product Knowledge**

Recommended:

- **RADME03: MDM 5.5 Java API**
- **Experiences with SAP NetWeaver Developer Studio**



➔ Getting Started with MDM Java API

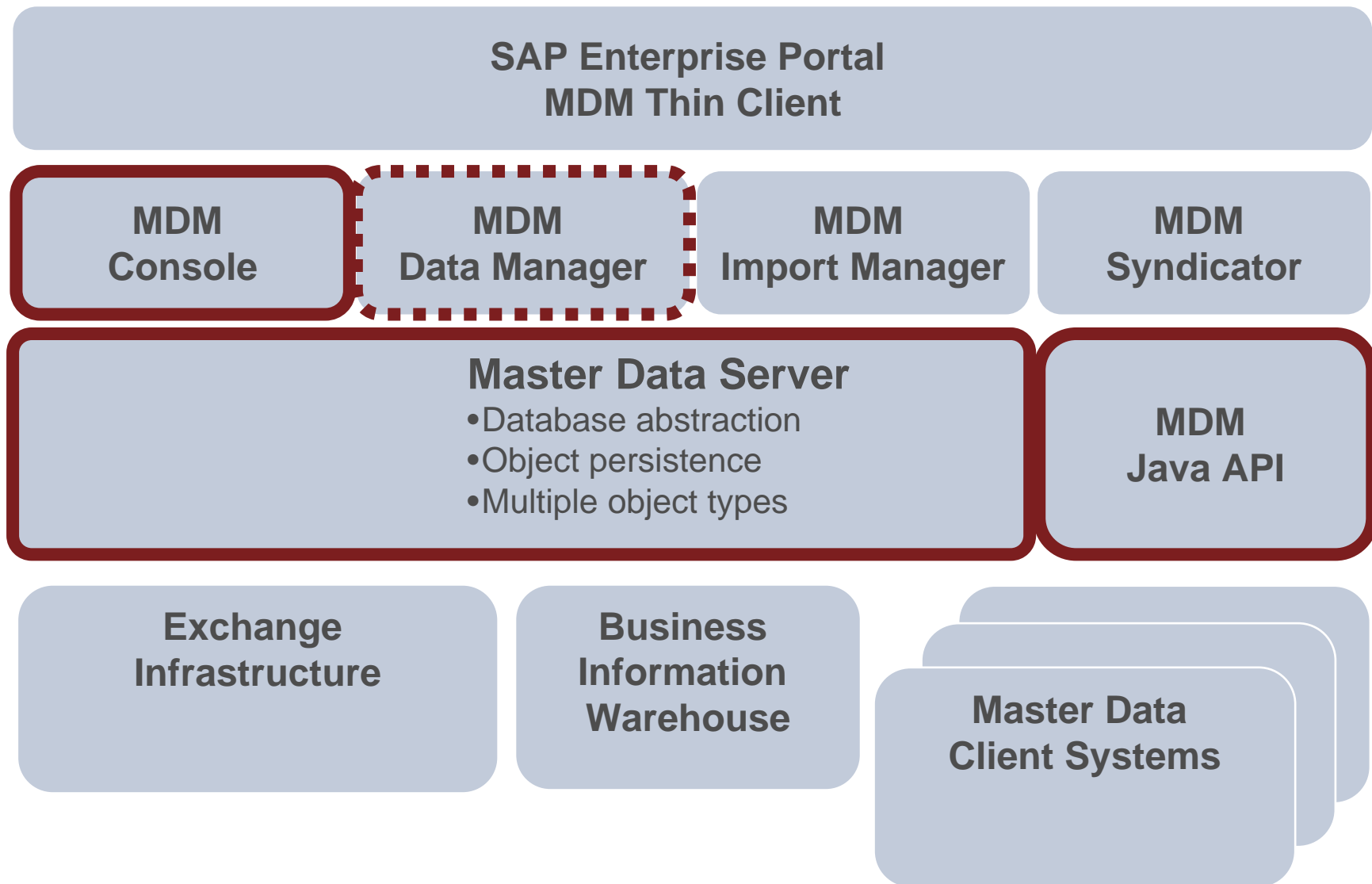
Changes in the Java API for SP02

MDM Java Connector

Demonstration:
Integrating MDM Client &
J2EE Application

Q & A

Overview of Landscape Components



Database Management System

Microsoft SQL Server (or other DB)
MSDE sufficient for non-prod usage

MDM Server

MDM Console

MDM Client

MDM Java API

Single JAR

J2EE Library

SAP NetWeaver Developer Studio

Optionally

■ **Further MDM Components**

■ **WebAS 6.40 for J2EE Applications**

Everything is available on the SAP Service Marketplace

→ <http://service.sap.com/swdc>

Installations

Patches

→ <http://service.sap.com/instguides>

MDM 5.5 Java API Library Reference Guide

Reference Guide in PDF Format

service.sap.com/instguides

MDM 5.5 Java API Java Doc

Java Doc in the API Archive

included in installation archive

RKT Material on Service Marketplace

Presentation & Exercises

service.sap.com/rkt-netweaver

How To use the MDM Java Connector in SAP Web AS

How To Guide in PDF Format

service.sap.com/netweaver

MDM Java API is pure Java, no DLLs / native libraries

Installation consists of:

For J2SE Applications

- Include MDM4J.JAR in classpath

For J2EE Applications

- Deploy com.sap.mdm.tech.mdm4j.sda with SDM
- Reference library in J2EE application

→ *Allows updating the library independent of J2EE application*

The API Library version must match the MDM Server version !



Getting Started with MDM Java API



Changes in the Java API for SP02

MDM Java Connector

Demonstration:
Integrating MDM Client &
J2EE Application

Q & A

Deprecated Methods

GetAgencies

has been replaced by **GetClientSystems**

```
public ClientSystemArray GetClientSystems()
```

```
public class ClientSystem
```

Method Summary

```
int GetID()
```

```
java.lang.String GetName()
```

```
void SetID(int id)
```

```
void SetName(java.lang.String name)
```

New Methods

CatalogData and **CatalogCache** got new methods to support internationalization

GetCodeRegion

Returns the coding region used by the API.

SetCodeRegion

This method sets the coding region used by the API.

Programs making use of this method are able to connect to any language in the repository without having to distinguish names (like field names) for each login language.

GetLoginRegion

Returns the data region use by the API.

New Methods

CMFieldInfo and **DMTableInfo** got new methods to support internationalization

GetCode

Returns the field name use for coding

New Methods

CMFieldInfo got new methods

CMFieldInfo.IsModifyOnce

Returns whether the field is set modify-once

CMFieldInfo.IsMultiLingual

Returns whether the field is set multi-lingual

Name Changes for Tables and Fields

Data Groups

The table “DataGroup” has been changed to “Data Groups”

Data Locations

The table “OrigLocation” has been changed to “Data Locations”

Image Formats

The table “Formats” has been changed to “Image Formats”

Image Rotations

The table “Rotations” has been changed to “Image Rotations”

Name Changes for Tables and Fields

Image Crops

The table “Croppings” has been changed to “Image Crops”.

Data Groups.Name

The field “DataGroupField” from table “Data Groups” has been changed to “Name”.

Data Locations.Name

The field “OrigLocationField” from table “Data Locations” has been changed to “Name”.

DataObject

The field “DataObject” has been changed to “Data ID”.



Getting Started with MDM Java API

Changes in the Java API for SP02



MDM Java Connector

Demonstration:
Integrating MDM Client &
J2EE Application

Q & A

Preferred way to access MDM Server for J2EE applications

MDM Java Connector offers

- **Connection Pooling**
- **API Usage in EJBs** (CatalogData is not serializable)

Standards Based: **Java Connector Architecture (JCA) 1.0 &
SAP Connector Framework**

Used by SAP MDM Portal iViews

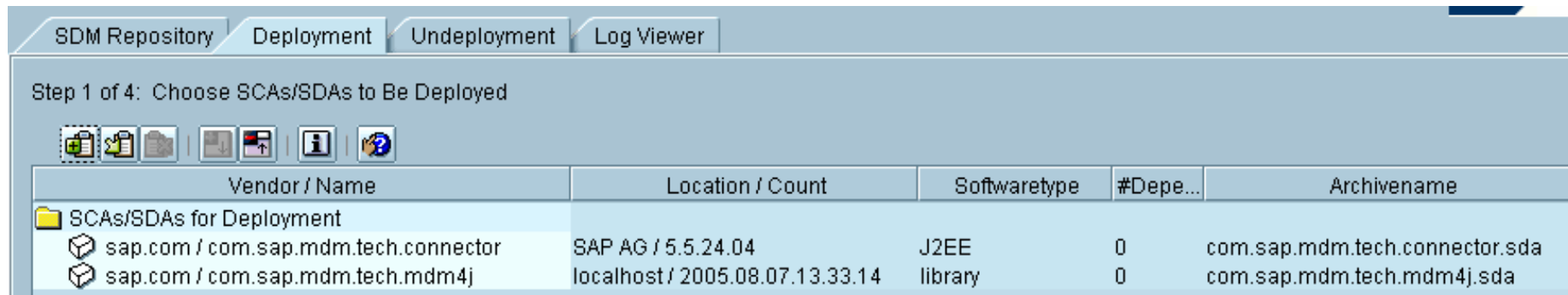
Further information on Connector Framework in SAP Library

<http://help.sap.com>

→ Portal → Portal Developer Guide → Connector Framework

Installation of MDM Java Connector

- Available as **SDA** `com.sap.mdm.tech.connector.sda`
- Requires **MDM Java API Lib** `com.sap.mdm.tech.connector.sda`
- Simply deploy with **SDM**

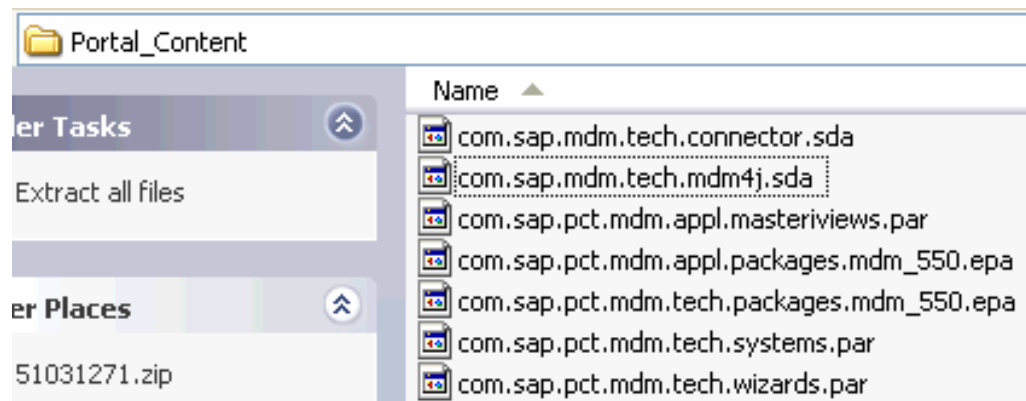


SDM Repository | Deployment | Undeployment | Log Viewer

Step 1 of 4: Choose SCAs/SDAs to Be Deployed

Vendor / Name	Location / Count	Softwaretype	#Depe...	Archivename
SCAs/SDAs for Deployment				
sap.com / com.sap.mdm.tech.connector	SAP AG / 5.5.24.04	J2EE	0	com.sap.mdm.tech.connector.sda
sap.com / com.sap.mdm.tech.mdm4j	localhost / 2005.08.07.13.33.14	library	0	com.sap.mdm.tech.mdm4j.sda

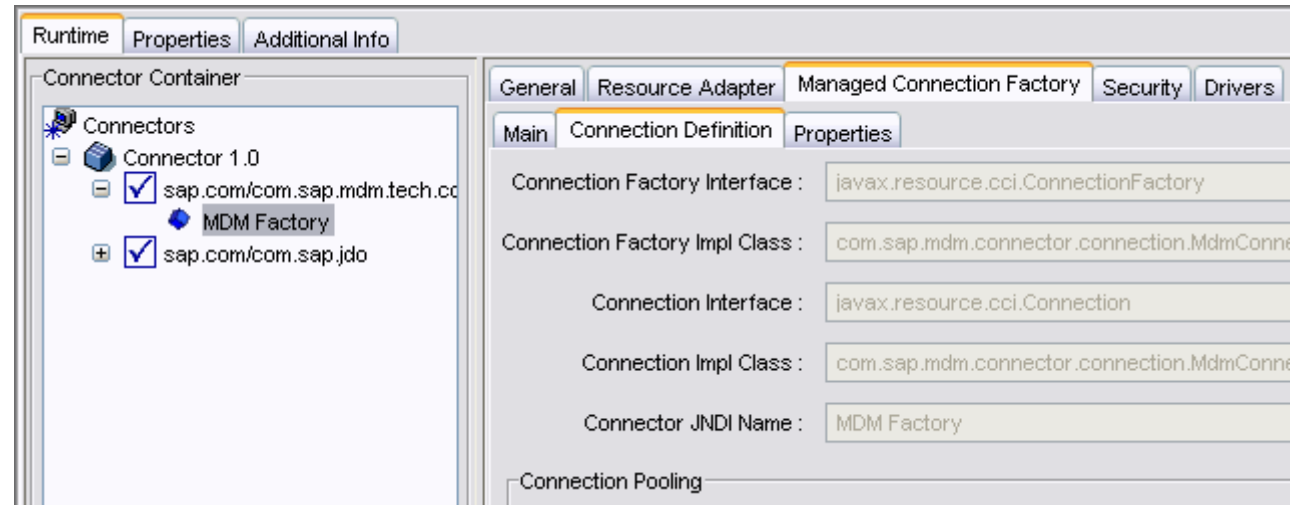
The SDAs are part of the
SAP MDM 5.5 SP02
installation CD



Administration of MDM Java Connector

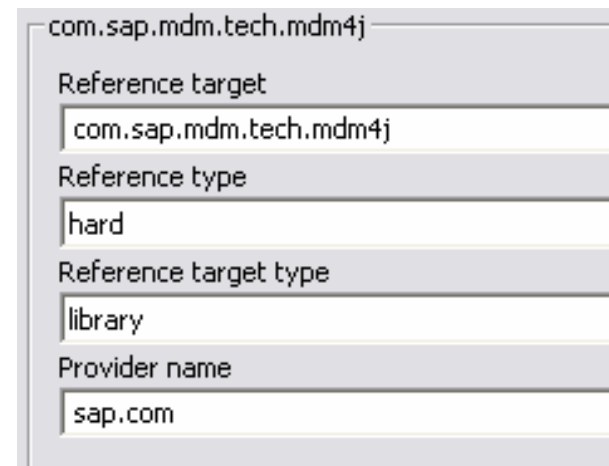
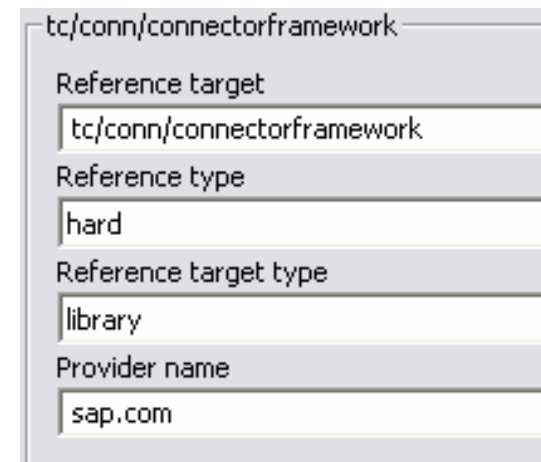
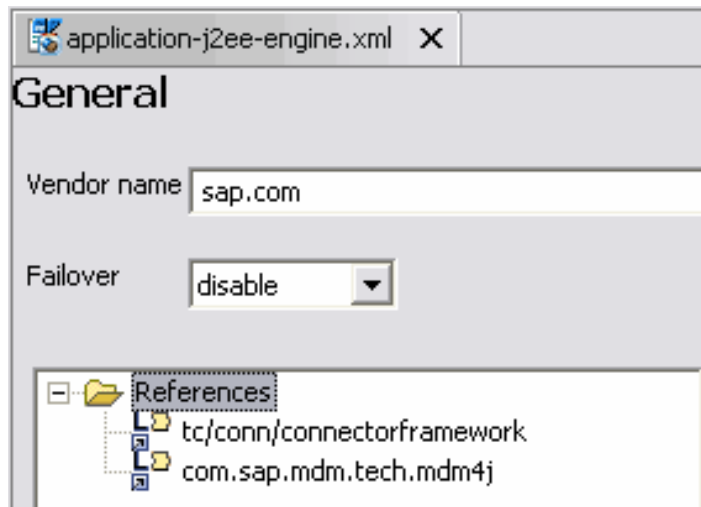
Via J2EE Administrator

- J2EE Admin
- Server
- Services
- Connector Container

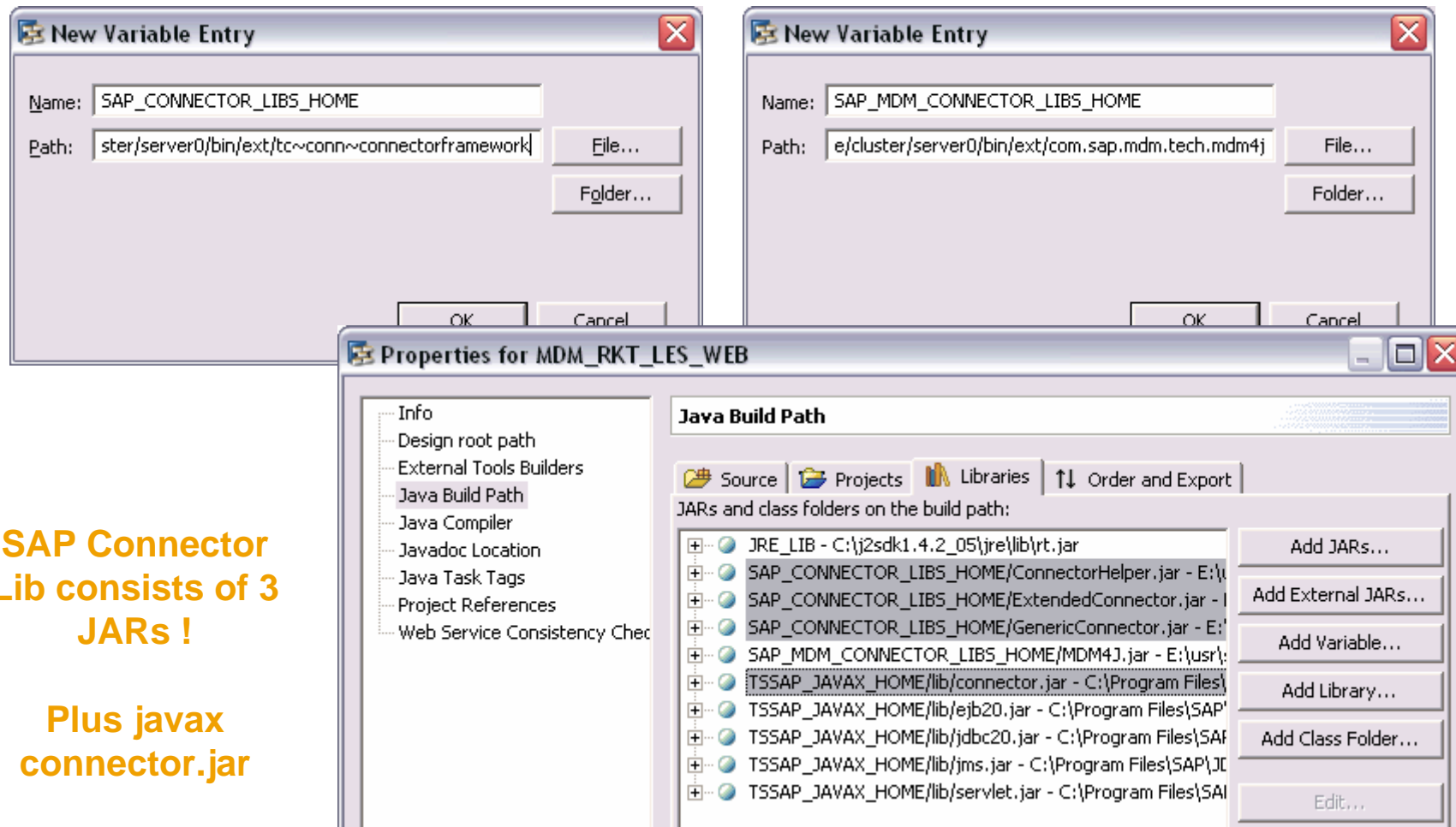


Expiration:	<input checked="" type="checkbox"/>		
Connection Lifetime:	<input type="text" value="60"/>	Maximum Connections:	<input type="text" value="40"/>
Cleanup Thread:	<input type="text" value="100"/>	Maximum Time to Wait for Connection:	<input type="text" value="60"/>

Your J2EE applications needs to reference the MDM Java Connector and the SAP Connector Framework in application-j2ee-engine.xml



For referencing the classes during development & compilation of java code, add also references to these libraries in the project.



SAP Connector Lib consists of 3 JARs !

Plus javax connector.jar

In your Java coding you need to import packages for JNDI and Connector classes as well as the MDM Java API packages you need

```
import javax.naming.*;
```

```
import com.sapportals.connector.connection.*
```

```
import a2i.common.*;
```

```
import a2i.core.*;
```

```
(...)
```

1. Get Connection Factory via JNDI Lookup

```
IConnectionFactory connectionFactory =  
(IConnectionFactory)ctx.lookup("deployedAdapters/MDM  
Factory/shareable/MDM Factory");
```

2. Get Connection Spec

```
IConnectionSpec spec = connectionFactory.getConnectionSpec();
```

3. Set Connection Information

```
spec.setPropertyValue("Server", "P120773");  
(...)
```

4. Obtain a Connection

```
connection = connectionFactory.getConnectionEx(spec);
```

5. Retrieve Native MDM Connector Interface

```
INative nativeInterface = connection.retrieveNative();
```

6. Retrieve Native MDM Connection (instance of CatalogData)

```
catalog = (CatalogData)  
nativeInterface.getNative(CatalogData.class.getName());
```


Having completed the work, close the connection with
`connection.close()`

Do not call `CatalogData.Logout()` as this is done by the connector internally (connection pooling) !

After `connection.close()` has been called the obtained `CatalogData` object must not be used any longer !

The logins to a MDM repository are pooled by the MDM Java connector among all connections, that were obtained with the same property settings of a `IConnectionSpec` object.

```
spec.setPropertyValue("Server", "P120773");  
(...)  
connection = connectionFactory.getConnectionEx(spec);
```



Getting Started with MDM Java API

Changes in the Java API for SP02

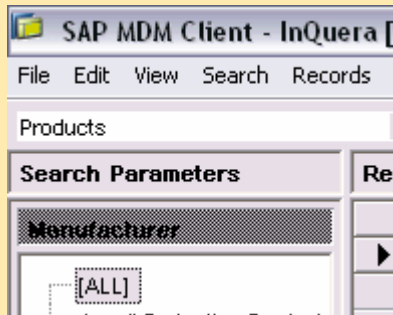
MDM Java Connector



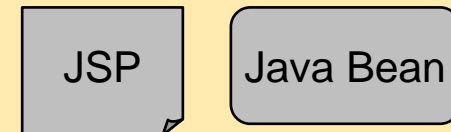
Demonstration:
Integrating MDM Client &
J2EE Application

Q & A

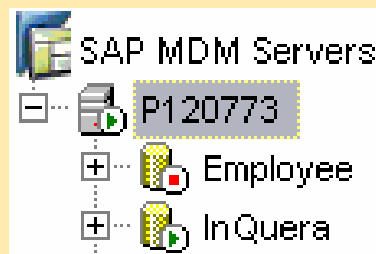
Example: Integration of MDM Client & J2EE Application



**MDM Client
Web Tab**



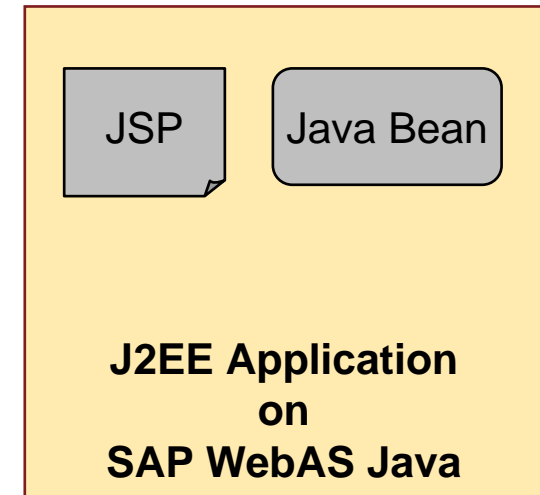
**J2EE Application
on
SAP WebAS Java**



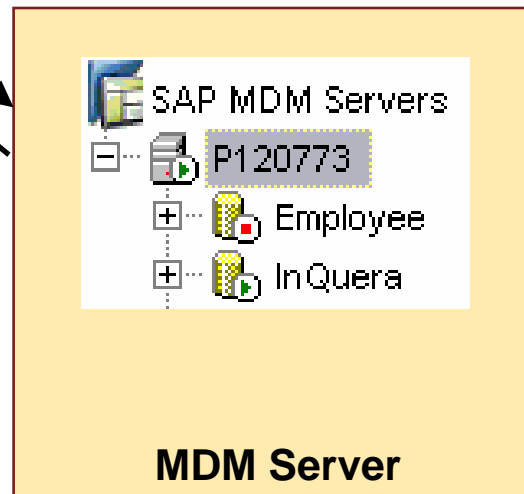
MDM Server

Example: Integration of MDM Client & J2EE Application

1



User is connected to MDM Server and selects records in MDM Client

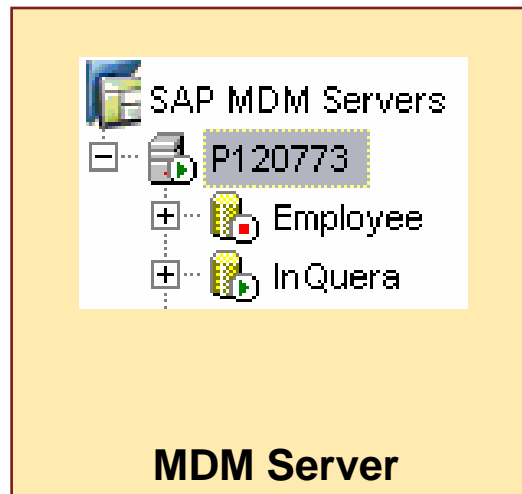
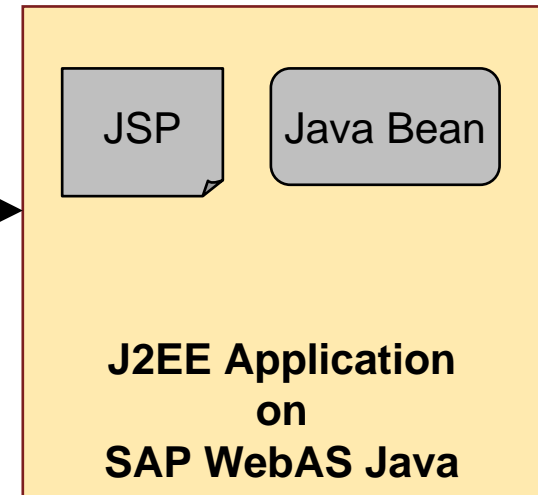


Example: Integration of MDM Client & J2EE Application

2

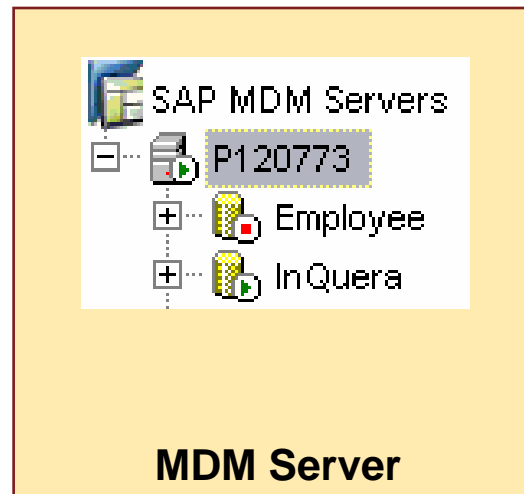
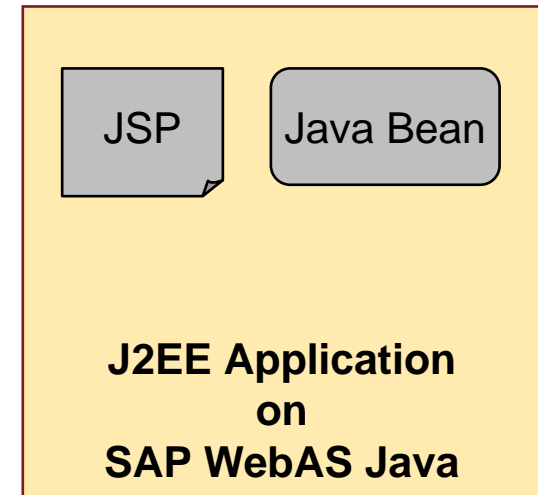


- User switches to Web Tab to access J2EE application.
- If required a new J2EE session is instantiated.
- Java Bean is initialized with search selection.



Example: Integration of MDM Client & J2EE Application

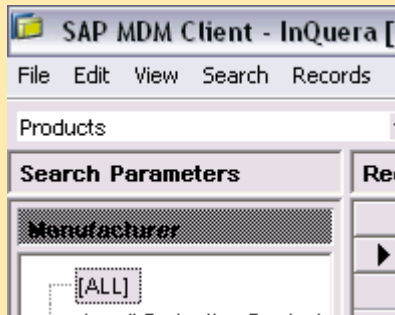
3



- Bean requests connection from connector pool
- MDM Connector creates new connection if none is available
- Bean retrieves further information from MDM Server.

Example: Integration of MDM Client & J2EE Application

4



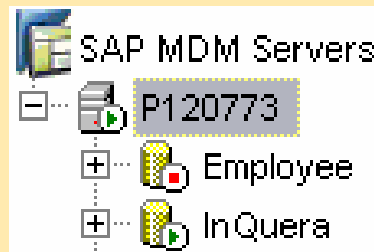
MDM Client

- JSP generates HTML page and returns information to MDM Client
- The user may select subsequent actions on the selected records
- further processing as before ...

JSP

Java Bean

**J2EE Application
on
SAP WebAS Java**



MDM Server

- **Administrator defines available URLs in the Console**
- **User can choose one of these to be displayed in Client**

5 parameters may be added to the URLs:

<t> all modes, replaced with the current table id

<r> Record/Hierarchy, replaced with a semi-colon delimited string of selected record ids, prepended with the count

<f:n> Record/Hierarchy, replaced with the appropriate field values of the selected records from field with id n. (e.g. <f:3>)

<c> Taxonomy, replaced with the selected category id

<a> Taxonomy, replaced with a semi-colon delimited string of selected attribute ids, prepended with the count

Some Points on the Example Application ...

**No per session (or request !) CatalogData object instantiated -
CatalogData instance requested from MDM Connector pool**

→ Expensive Login() calls avoided

**Multiple concurrent J2EE user sessions share a common
pool of MDM Server Logins**

→ Reduced session and login load on MDM Server side

Pooling is handled by SAP Connector Framework

→ No custom coding to support pooling in J2EE applications

“Shared” Authentication

- Currently anybody can use the J2EE application
- Authorizations of fixed system user applies

Enhancement: Authentication per user session

- Prompt user for name and password, when session is started
- Check credentials with `CatalogData.Authenticate()`
- Store “isAuthenticated” in master bean

Use the J2EE Admin Log Viewer. You may see Exceptions like this:

Exception

com.sap.engine.services.connector.exceptions.BaseResourceException: **Cannot get connection for 60 seconds. Possible reasons:** 1) Connections are cached within SystemThread(can be any server service or any code invoked within SystemThread in the SAP J2EE Engine), 2) **The pool size of adapter "MDM Factory" is not enough according to the current load of the system** or 3) The specified time to wait for connection is not enough according to the pool size and current load of the system. In case 1) the solution is to check for cached connections using the Connector Service list-conns command, in case 2) to increase the size of the pool and in case 3) to increase the time to wait for connection property. In case of application thread, there is an automatic mechanism which detects unclosed connections and unfinished transactions.

at

com.sap.engine.services.connector.jca.ConnectionHashSet.match(ConnectionHashSet.java:272)

...



Getting Started with MDM Java API

Changes in the Java API for SP02

MDM Java Connector

Demonstration:
Integrating MDM Client &
J2EE Application

➔ Q & A

Thank you !

Additional information:

<http://intranet.sap.com/rkt-netweaver>

<http://service.sap.com/rkt-netweaver>

-> SAP MDM 5.5

- 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, and Informix are trademarks or registered trademarks of IBM Corporation 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.
-
- The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.
 - This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.
 - SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided 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 have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.
 - The statutory liability for personal injury and defective products is not affected. 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