

Using Java to Access Your CA IDMS™ Databases and Applications

Dave Ross
CA Technologies

IUA/CA IDMS™ Technical Conference
May 16-20, 2016



Abstract

- Many users are focused on developing new applications in Java, but still have a large investment in CA IDMS databases and applications. This session shows how you can use JDBC and SQL to leverage your CA IDMS databases and business logic from your Java applications. It includes an overview of Java programming concepts and JDBC for those new to Java, as well as more advanced topics for more experienced users.

Agenda

- OOPS Concepts
- Java Programming Elements
- JDBC Overview
- Sample JDBC Application
- Advanced JDBC Features

OOPS Concepts

Object Oriented Programming

- Objects
- Messages
- Classes
- Inheritance

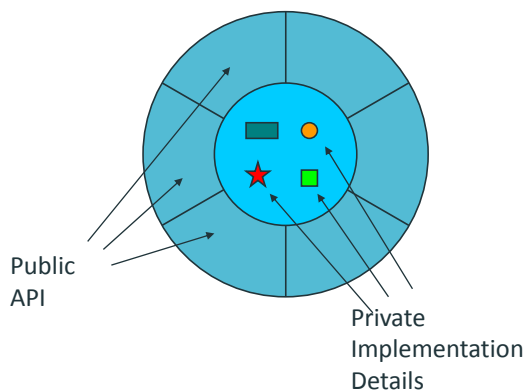
5 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Object

- Software bundle
 - Variables
 - Methods
- State
- Behavior



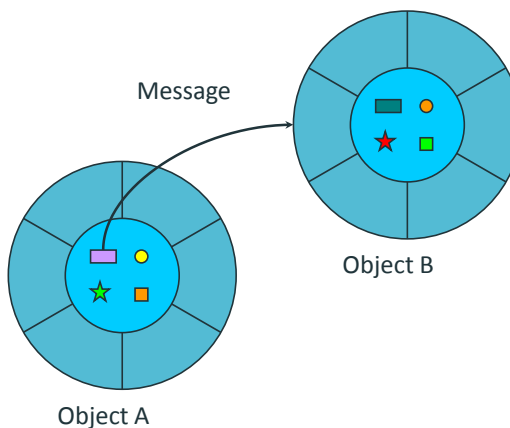
6 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Message

- How objects interact
- Components
 - Object address
 - Method name
 - Parameters



7

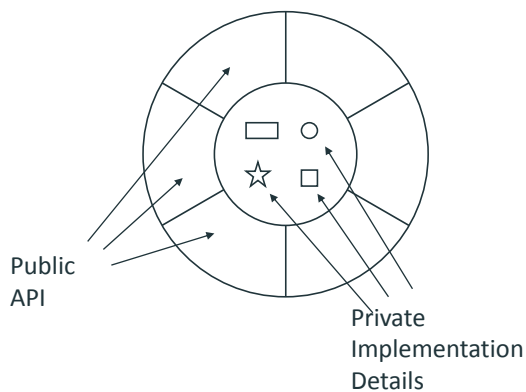
IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Class

- Object “blueprint”
- Members
 - Class
 - Instance



8

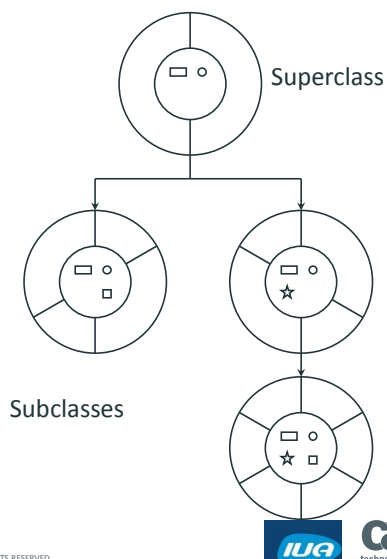
IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Inheritance

- **Superclass**
 - Common behavior
 - Reusable
- **Subclass**
 - Specialized behavior
 - Override methods
 - Hide variables



Java Programming Elements

Java

- Object oriented programming language
- Software platform
 - Java Virtual Machine
 - Java Application Programming Interface

Java Language Elements

- class
- interface
- package
- exception
- import

Class

- Basic programming unit
- Members
 - Variables
 - Methods

Interface

- More than a header
- Named “protocol”
 - Defines abstract methods
 - Declares constants
- Classes implement interfaces
- Reference type

Package

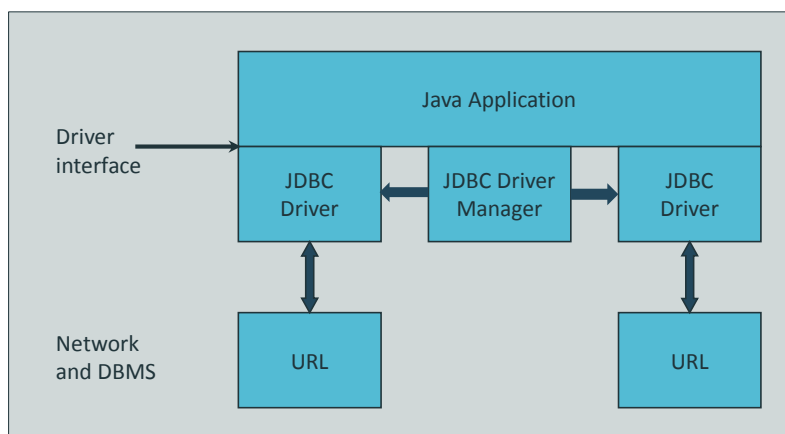
- Like a library
- Related
 - Classes
 - Interfaces
 - Exceptions
- #import into Java program

JDBC Overview

JDBC Concepts

- “Java Data Base Connectivity”
- Call Level Interface
- Object Oriented
- Interoperability
- Any Java platform

JDBC Architecture



JDBC Objects

- Packages
 - java.sql
 - Javax.sql
- Interfaces
- Classes
- Exceptions

Commonly Used JDBC Interfaces

- Driver
- Connection
- Statement
- ResultSet
- PreparedStatement
- CallableStatement
- DatabaseMetaData
- ResultSetMetaData

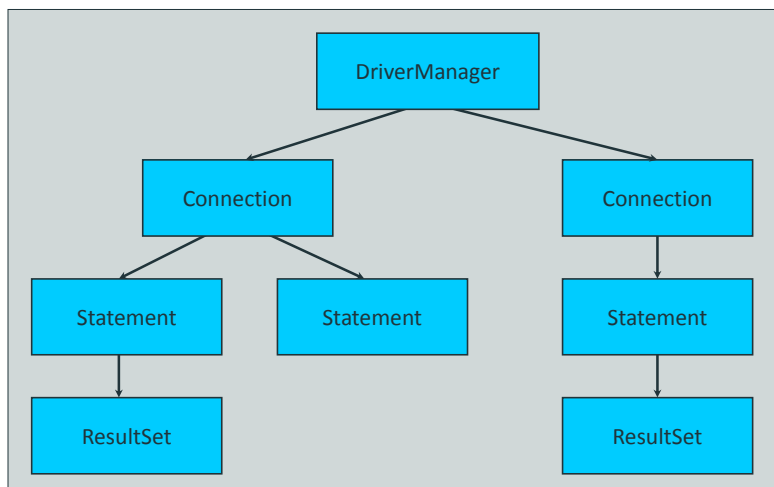
JDBC Classes

- DriverManager
- DriverPropertyInfo
- Types
- Date
- Time
- Timestamp

JDBC Exceptions

- SQLException
- SQLWarning
- DataTruncation

JDBC Object Relationships



Sample JDBC Application

JDBC Query Application

- Simplified version of IdmsExample
 - Included with CA IDMS Server
 - Can execute any SQL statement
 - Java version of “BCF”
- Connect to a database
- Execute a query
- Retrieve results
- Handle errors

Housekeeping

```
import java.io.*;           // standard output
import java.util.*;         // properties
import java.sql.*;          // basic JDBC classes

class IuaExample
{
    public static void main (String[] args)
    {
        // register driver
        try
        {
            Class.forName("ca.idms.jdbc.IdmsJdbcDriver");
        }
    }
}
```

Get Connected

```
// set database name and location
String url = "jdbc:ids://host:3709/dictname";

// set sign on information
Properties info = new Properties();
info.put("userid", "password");

// get a connection to the database
Connection conn =
    DriverManager.getConnection(url, info);
```

Here's the Meat

```
// get a statement object to...
Statement stmt = conn.createStatement();

// execute the query and get a result set
ResultSet rs = stmt.executeQuery(
    "SELECT EMPFNAME, EMPLNAME FROM DEMO.EMPL");

// get and display the result set columns
while (rs.next())
    System.out.println(
        rs.getString(1) + " " + rs.getString(2));
```

The Punting Game

```
// disconnect
conn.close();
}
catch (ClassNotFoundException e)
{
    System.out.println("No driver: " + e);
}
catch (SQLException e)
{
    System.out.println("SQL Error: " + e);
}
}
```

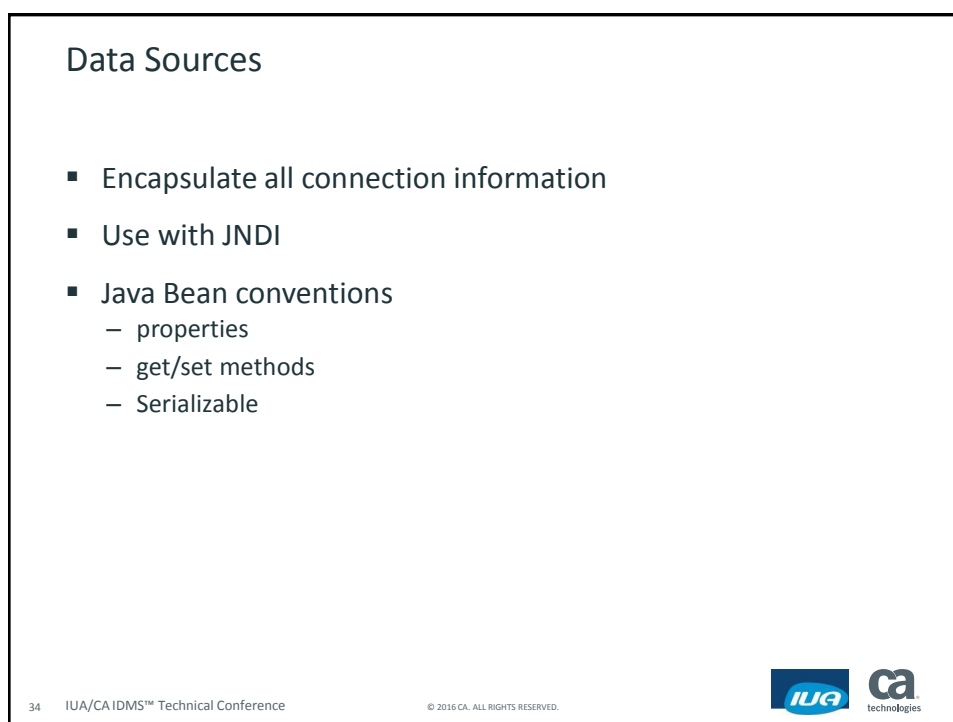
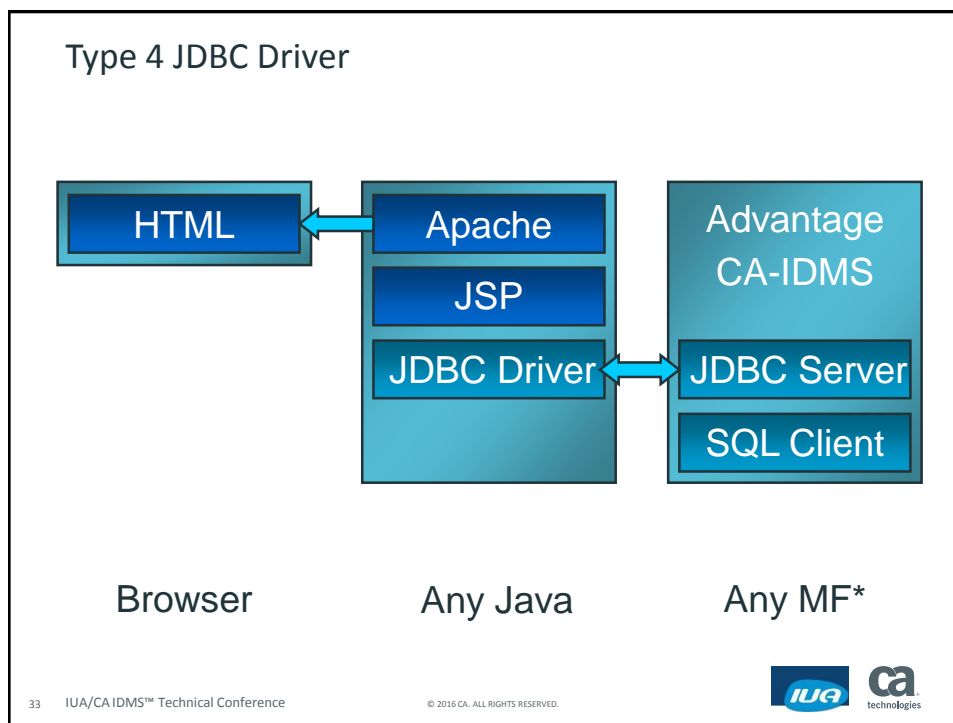
Advanced JDBC Features

CA IDMS JDBC Supported Features

- Type 4 JDBC Driver
- Data Sources
- Procedures and Returned Result Sets
- Batch and Positioned Updates
- Enhanced Result Sets

Type 4 JDBC Driver

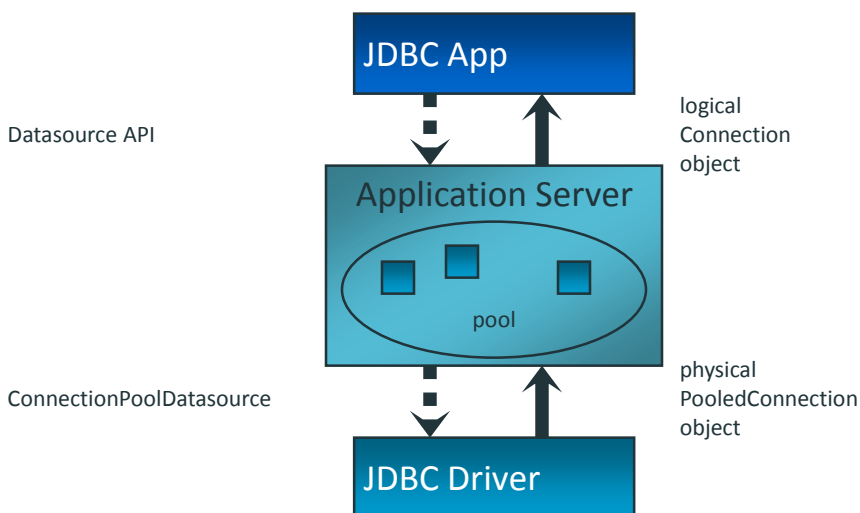
- Direct connection from JDBC Driver to CV
 - Uses TCP/IP line driver
 - CAICCI not needed
- JDBC Server in CV
 - Generic listener task
- Transparent to JDBC driver
 - Just identified by URL

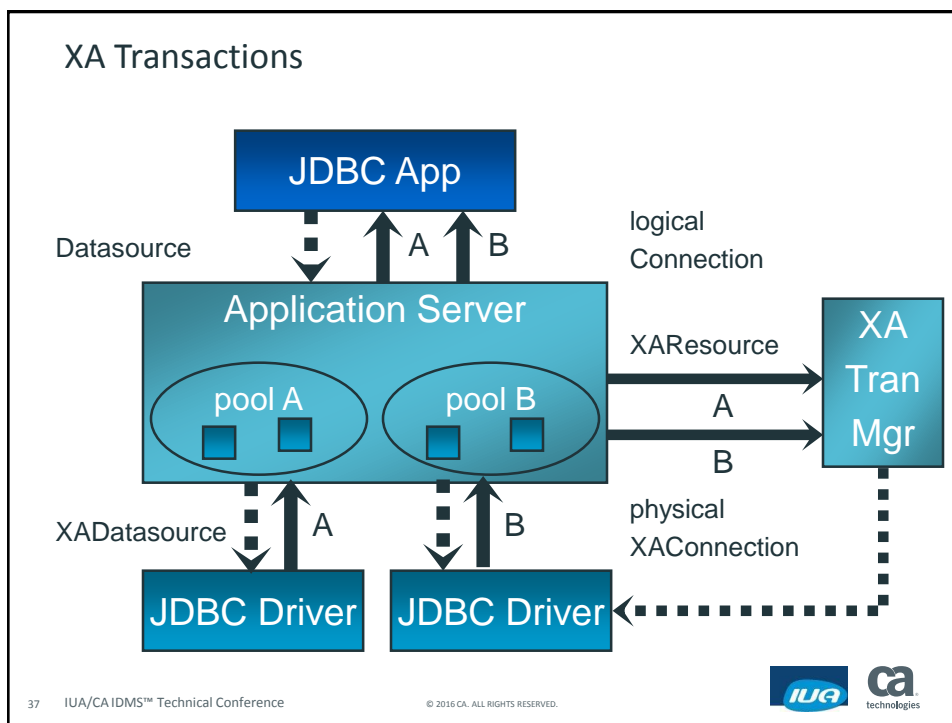


IDMS DataSource Classes

- IdmsDataSource
- IdmsConnectionPoolDataSource
- IdmsXADataSource
- Typically implemented by application server
 - Except when using distributed transactions

Connection Pooling





Called Procedures

- IdmsCallableStatement class
- Escape syntax
 - {call procedurename (?) }
- DatabaseMetaData methods
 - `getProcedures ()`
 - `getProcedureColumns ()`
- CallableStatement methods
 - Output parameters

38 IUA/CA IDMS™ Technical Conference © 2016 CA. ALL RIGHTS RESERVED.

JDBC named parameters

- Bind parameters to parameter markers using names instead of ordinal indexes:
 - Use with SQL CALL statement
 - Names as specified by CREATE PROCEDURE
- Simplifies use of procedures that have many parameters, particularly when defaults are acceptable
- Named parameters can be specified in any order
- Unused parameters can be omitted
- Supports both input and output parameters

JDBC Named Parameter Binding

```
CallableStatement cstmt =
    conn.prepareStatement("CALL GETEMPLOYEE(?, ?, ?)");

cstmt.setString("EmpID", "B503-8907-15");
cstmt.registerOutParameter("Name",
    Types.VARCHAR);
cstmt.registerOutParameter("Age",
    Types.INTEGER);

cstmt.execute();

System.out.println("Employee: " +
    cstmt.getString("Name") + " is " +
    cstmt.getInt("Age") + " old.");
```

CA IDMS Returned Result Sets in Procedure

- Procedures can return result sets

```
CREATE PROCEDURE RSPROC  
    (CHAR SOMEPARM, ...)  
    DYNAMIC RESULT SETS 5
```

- Cursors left open after procedure exits are RETURNED

```
ALLOCATE CURSNAME CURSOR WITH RETURN
```

CA IDMS Returned Result Sets in Calling Program

- Calling program gets result sets
- Allocate RECEIVED cursor for result sets

```
CALL RSPROC (...)  
ALLOCATE RCURNAME FOR PROCEDURE  
    SPECIFIC PROCEDURE RSPROC
```

- Step through RETURNED result sets

```
Loop until SQLSTATE = '0100D'  
    Loop until SQLSTATE = '02000'  
        FETCH RCURNAME  
    CLOSE RCURNAME
```

JDBC Multiple Result Sets

- JDBC driver can have multiple RECEIVED cursors
 - `KEEP_CURRENT_RESULT`
 - `CLOSE_CURRENT_RESULT`
 - `CLOSE_ALL_RESULTS`
- Allocate cursor and step through result sets

```
rc = statement.executeQuery("CALL RSPROC");
while (rc != false) {
    resultSet = statement.getResultSet();
    while (rc != false) {
        resultSet.fetch();
    }
    rc = statement.getMoreResults(CLOSE_CURRENT...);
}
```

Positioned Updates

- Statement methods:
 - `setCursorName("CURSOR1")`
 - `getCursorName()`
- SQL statements
 - `SELECT ... FOR UPDATE`
 - `UPDATE WHERE CURRENT OF CURSOR1`
- Disables BULK FETCH

Batched Updates

- Statement methods
 - `addBatch()`
 - `executeBatch()`
 - `cancelBatch()`
- Required for J2EE compliance
- Driver caches most SQL commands
- Direct support for INSERT BULK

Enhanced Result Sets

- Scrollable
- Updateable
- Typical ResultSet methods:
 - `getRow()`
 - `updateRow()`
- Driver caches fetched rows
- Uses “optimistic concurrency” for updating

JDBC Result Sets

- ResultSet type attribute
 - `TYPE_FORWARD_ONLY`
 - `TYPE_SCROLL_INSENSITIVE`
 - `TYPE_SCROLL_SENSITIVE`
- ResultSet concurrency attribute
 - `CONCUR_READ_ONLY`
 - `CONCUR_UPDATABLE`
- ResultSet methods
 - Position cursor
 - Update row

CA IDMS Scrollable Result Sets

- Caches rows as fetched from CA IDMS
- ResutSet type attribute
 - `TYPE_FORWARD_ONLY`
 - `TYPE_SCROLL_INSENSITIVE`
- ResultSet concurrency attribute
 - `CONCUR_READ_ONLY`
- ResultSet methods
 - Position cursor

JDBC Row Sets

- Extend ResultSet interface
- Reference Implementation
 - CachedRowSet
 - Included in JRE
- Extends CA IDMS result set implementation
 - **TYPE_SCROLL_SENSITIVE**
 - **CONCUR_UPDATABLE**
 - Update methods

Summary

- OOPS Concepts
- Java Programming Elements
- JDBC Overview
- Sample JDBC Application
- Advanced JDBC Features

FOR INFORMATION PURPOSES ONLY Terms of this Presentation

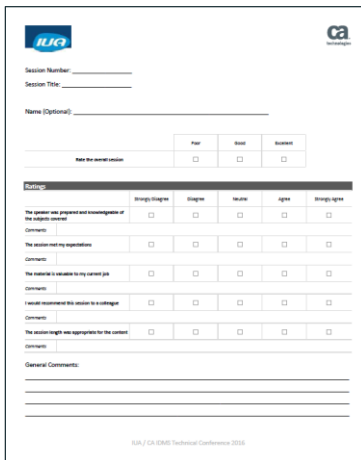
This presentation was based on current information and resource allocations as of May 2016 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Certain information in this presentation may outline CA's general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this presentation "as is" without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages. CA confidential and proprietary. No unauthorized copying or distribution permitted.

Questions and Answers

Please Complete a Session Evaluation Form

- The number for this session is **A08**
- After completing your session evaluation form, place it in the envelope at the front of the room



The form is titled "Session Evaluation Form" and includes the IUA and CA Technologies logos. It contains fields for "Session Number", "Session Title", and "Name (Optional)". Below these is a table for "Rate the session/lecture" with columns for "Poor", "Good", and "Excellent", each with a checkbox. The "Rating" section features a table with five columns: "Strongly Dislike", "Dislike", "Neutral", "Like", and "Strongly Like". There are four rows of evaluation questions, each with a "Comments" field and checkboxes for each rating column. The questions are: "The speaker was prepared and knowledgeable of the subject matter", "The speaker met my expectations", "The material is relevant to my current job", "I would recommend this session to a colleague", and "The session length was appropriate for the content". A "General Comments" section with three lines for text is at the bottom. The footer reads "IUA / CA IDMS Technical Conference 2016".