Guidelines: Writing Java applications with CA Datacom and CA Datacom Server

CTC 18 – App Dev – Thursday, April 28 2016

Lynette Elwell

Abstract

- This session provides an overview of the steps and best practices used to code and execute JDBC applications to access CA Datacom tables with CA Datacom Server.

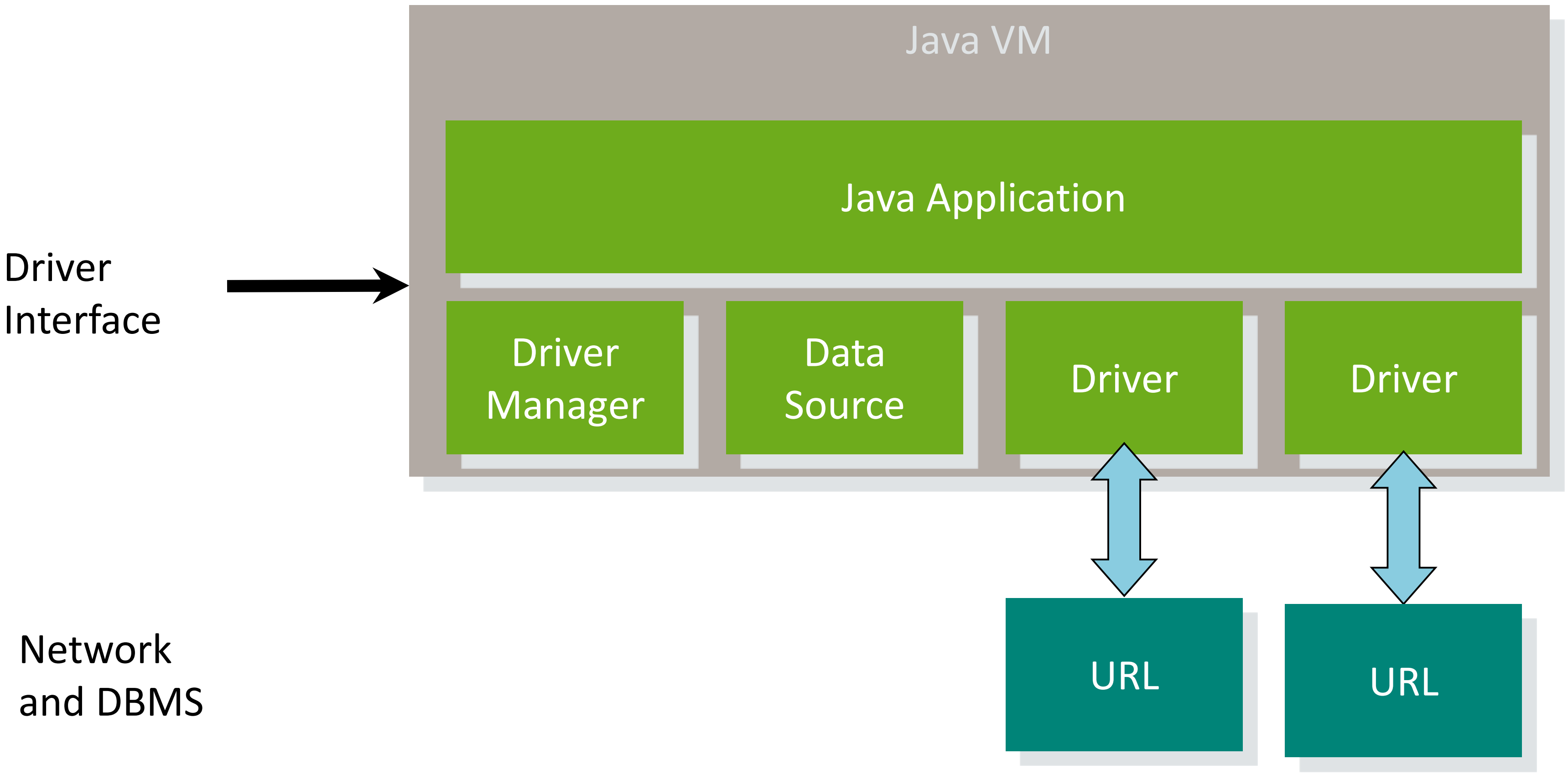
Agenda

- JDBC Concepts
- JDBC Architecture

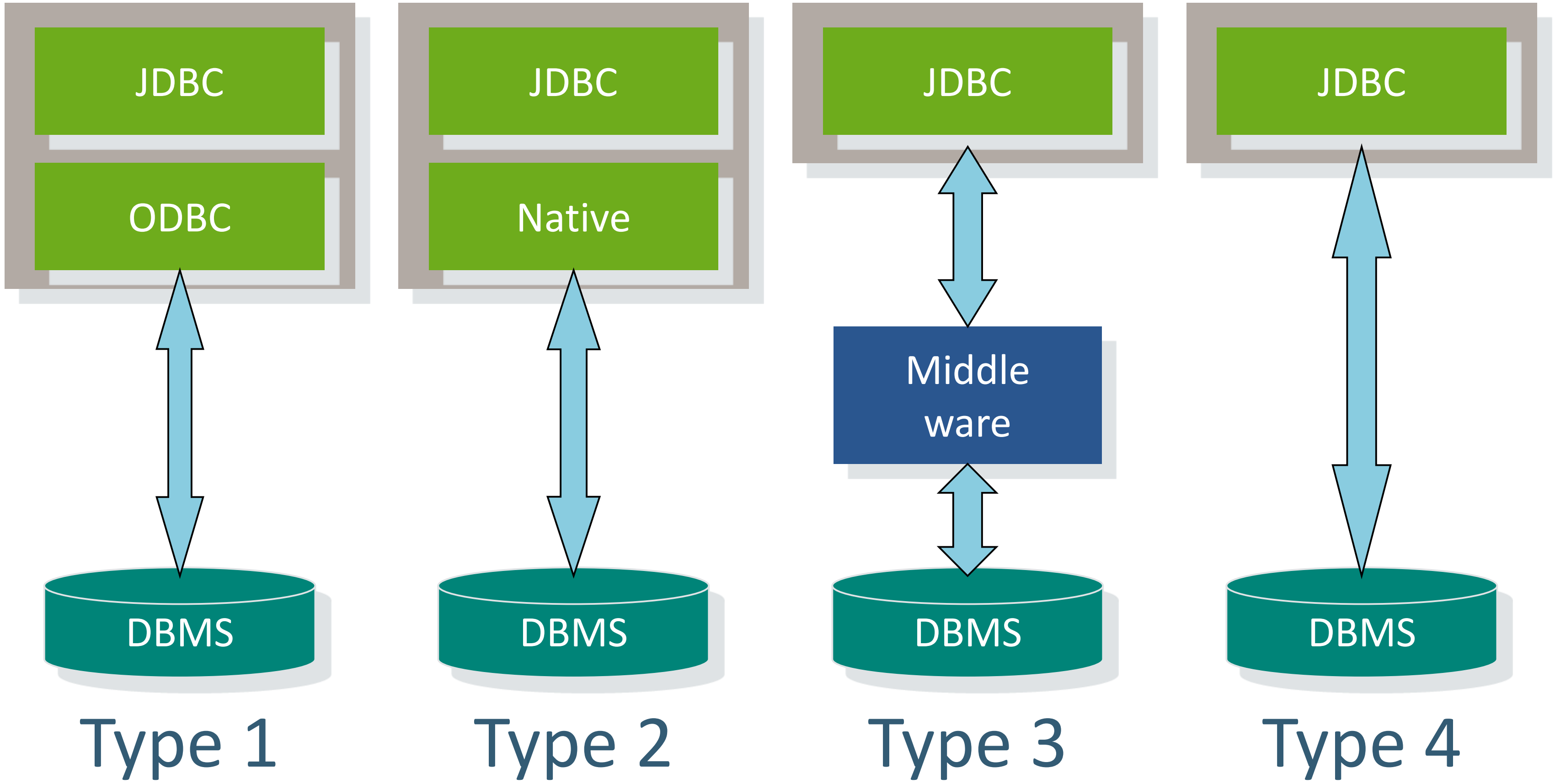
JDBC – Java Database Connectivity

- Industry standard API for database-independent connectivity
- Object oriented
- Interoperability
- “Write Once, Run Anywhere” capabilities of Java for access to enterprise data
- Runs on any platform that supports Java

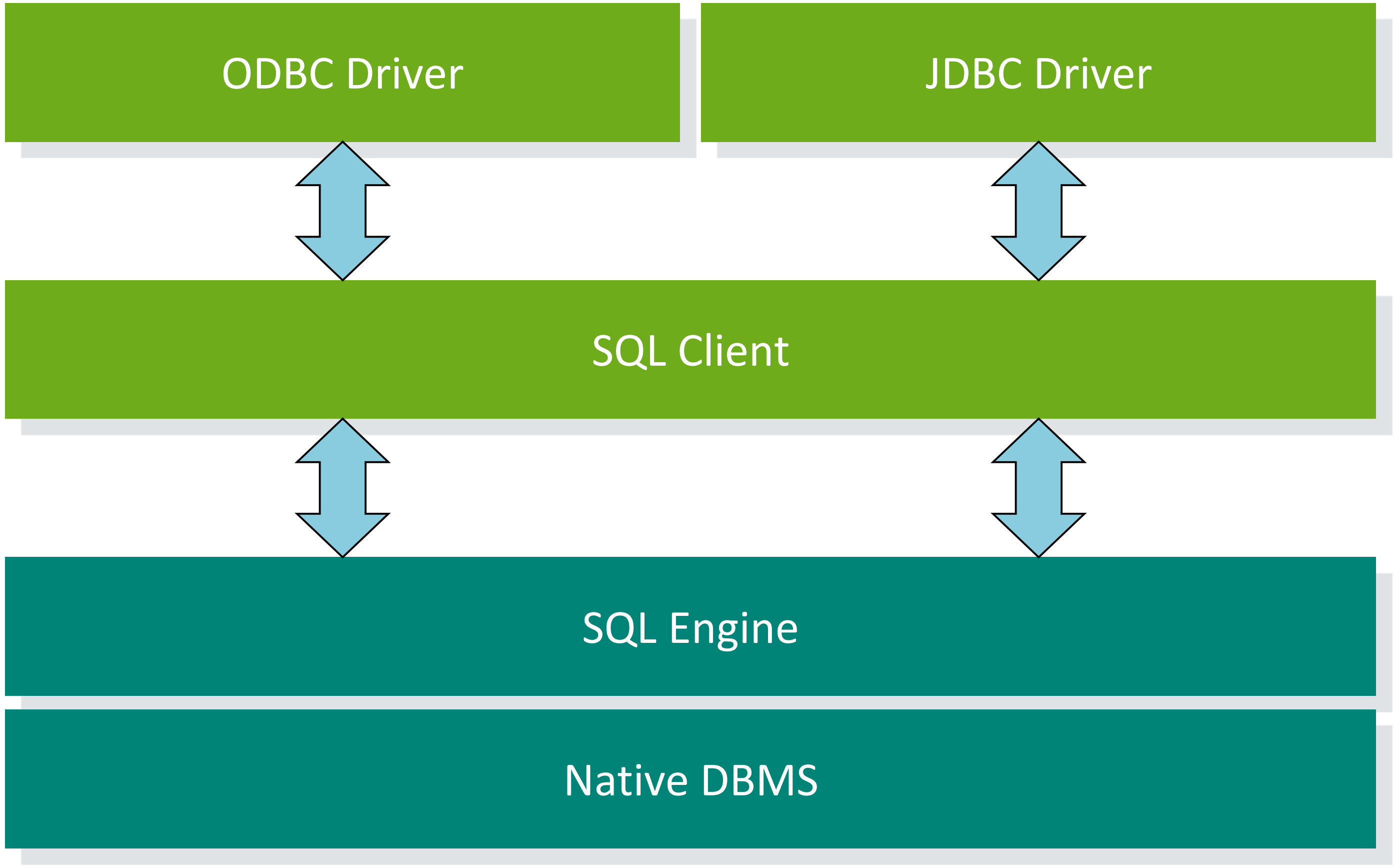
JDBC Architecture



JDBC Driver Types



CA Datacom Server Basic Architecture

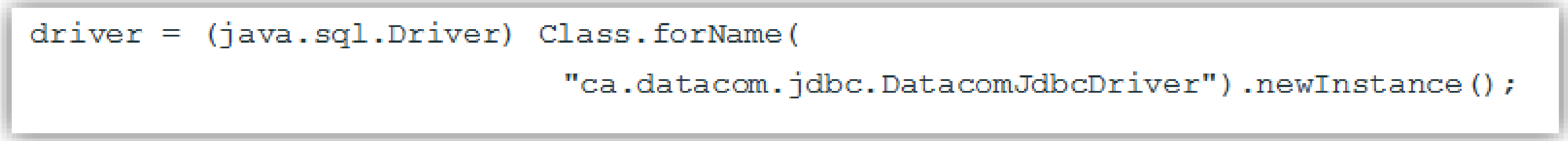


JDBC Driver – cadcjdbc.jar

- JDBC 4.0 compliant
- “Universal” driver
  - Types 2, 3, 4
- All client platforms that support Java
  - Windows
  - z/OS USS
  - Linux(x86, z/Series)
  - Other Unix

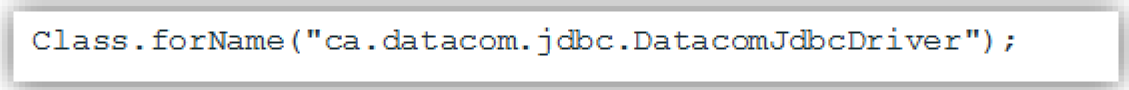
JDBC Driver – cadcjdbc.jar

- Driver Name: ca.datacom.jdbc.DatacomJdbcDriver
- Auto-Loading of JDBC Driver – JDBC 4.0
  - No need to invoke Class.forName()



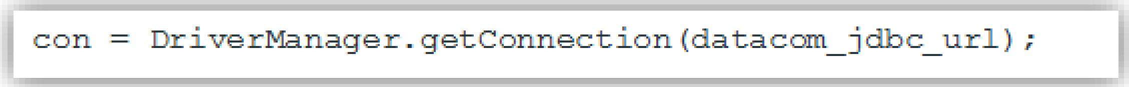
```
driver = (java.sql.Driver) Class.forName(
                                  "ca.datacom.jdbc.DatacomJdbcDriver").newInstance();
```

  - … or …



```
Class.forName("ca.datacom.jdbc.DatacomJdbcDriver");
```

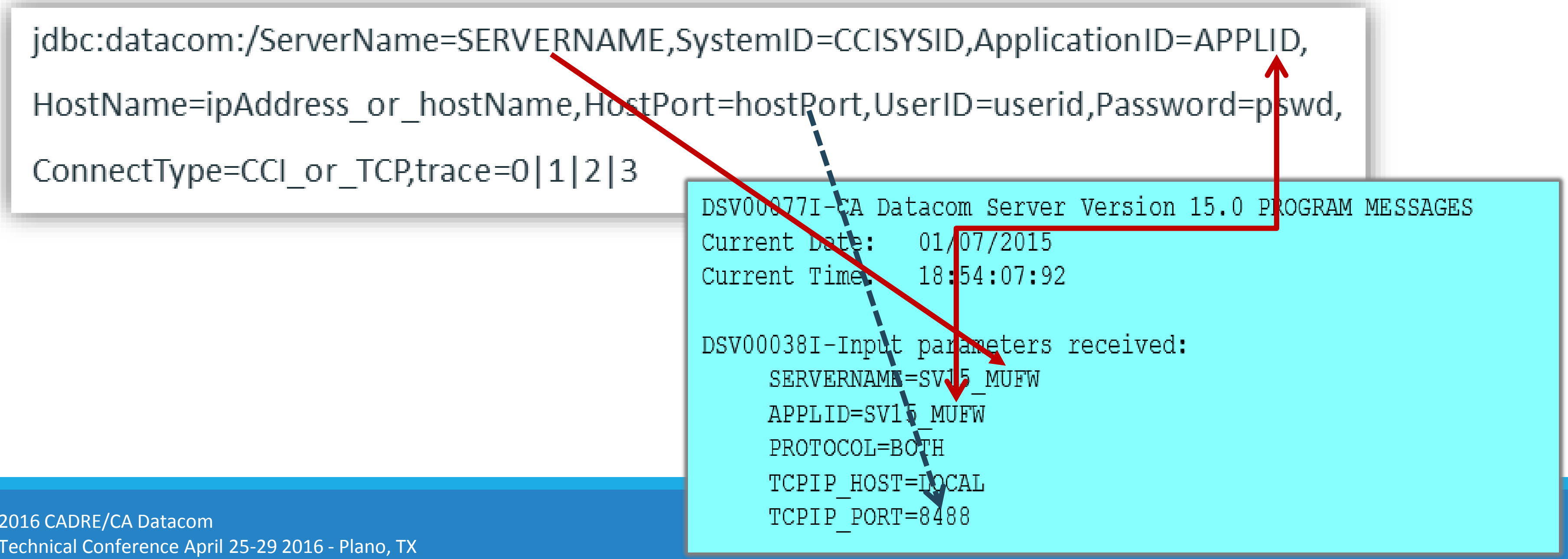
  - Now, simply call getConnection()



```
con = DriverManager.getConnection(datacom_jdbc_url);
```

JDBC Type 2 Connection

- Requires CA Datacom Server native interface
  - cadcdb32 – 32bit JVMs
  - cadcdb64 – 64bit JVMs
- Windows and z/OS USS, only – not valid on Linux platforms

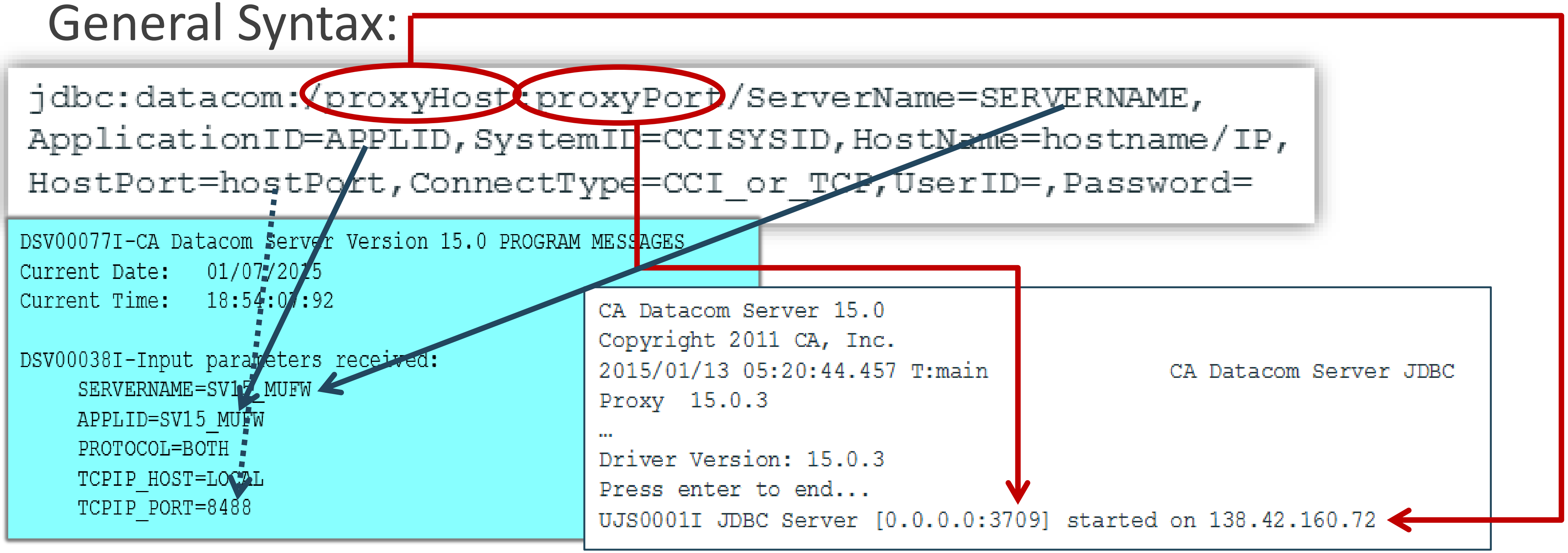


```
jdbc:datacom:/ServerName=SERVERNAME,SystemID=CCISYSID,ApplicationID=APPLID,
HostName=ipAddress_or_hostName,HostPort=hostPort,UserID=userid,Password=pswd,
ConnectType=CCI_or_TCP,trace=0|1|2|3
```

```
DSV00077I-CA Datacom Server Version 15.0 PROGRAM MESSAGES
Current Date:   01/07/2015
Current Time:   18:54:07:92

DSV00038I-Input parameters received:
     SERVERNAME=SV15_MUFW
     APPLID=SV15_MUFW
     PROTOCOL=BOTH
     TCPIP_HOST=LOCAL
     TCPIP_PORT=8488
```

JDBC Type 3 Connection

- Type 3 – Requires a JDBC Proxy on Windows or USS
- Linux Platforms – x86 or z/Series
- General Syntax:



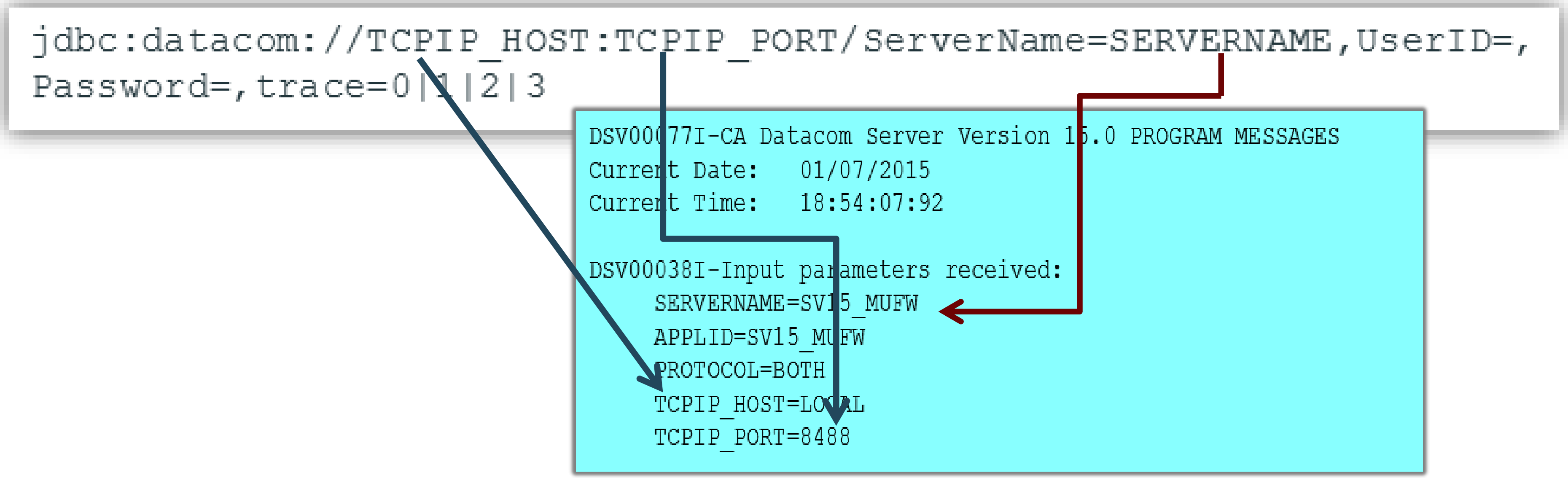
```
jdbc:datacom://proxyHost:proxyPort/ServerName=SERVERNAME,
ApplicationID=APPLID,SystemID=CCISYSID,HostName=hostname/IP,
HostPort=hostPort,ConnectType=CCI_or_TCP,UserID=,Password=
```

```
DSV00077I-CA Datacom Server Version 15.0 PROGRAM MESSAGES
Current Date:   01/07/2015
Current Time:   18:54:07:92

DSV00038I-Input parameters received:
     SERVERNAME=SV15_MUFW
     APPLID=SV15_MUFW
     PROTOCOL=BOTH
     TCPIP_HOST=LOCAL
     TCPIP_PORT=8488
```

```
CA Datacom Server 15.0
Copyright 2011 CA, Inc.
2015/01/13 05:20:44.457 T:main                    CA Datacom Server JDBC
Proxy  15.0.3
…
Driver Version: 15.0.3
Press enter to end...
UJS0001I JDBC Server [0.0.0.0:3709] started on 138.42.160.72
```

JDBC Type 4 Connection

- Type 4 – Requires TCP/IP - not valid with PROTOCOL=CCI
- Pure Java - Supported on all Java platforms
- Direct Connection to CA Datacom Server Mainframe region

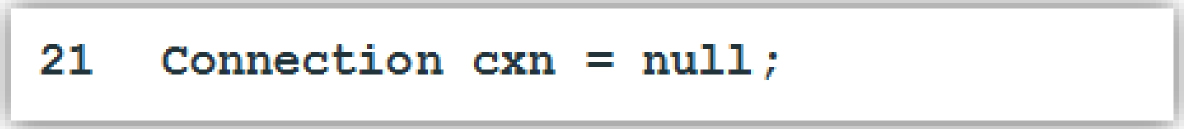


```
jdbc:datacom://TCPIP_HOST:TCPIP_PORT/ServerName=SERVERNAME,UserID=,
Password=,trace=0|1|2|3
```

```
DSV00077I-CA Datacom Server Version 15.0 PROGRAM MESSAGES
Current Date:    01/07/2015
Current Time:    18:54:07:92

DSV00038I-Input parameters received:
     SERVERNAME=SV15_MUFW
     APPLID=SV15_MUFW
     PROTOCOL=BOTH
     TCPIP_HOST=LOCAL
     TCPIP_PORT=8488
```

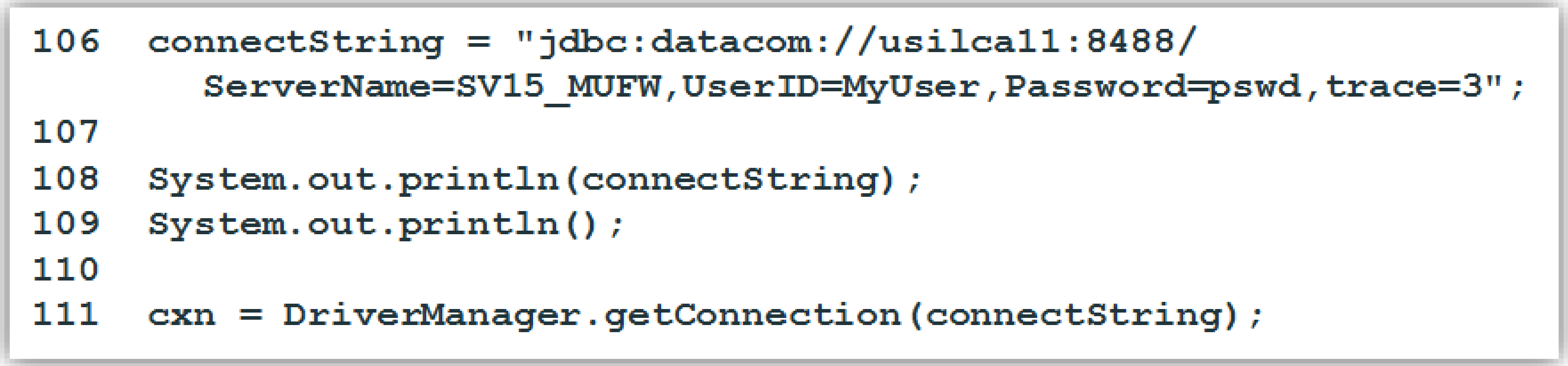
JDBC_Example.java – Establish a Connection

- Instantiate a connection object



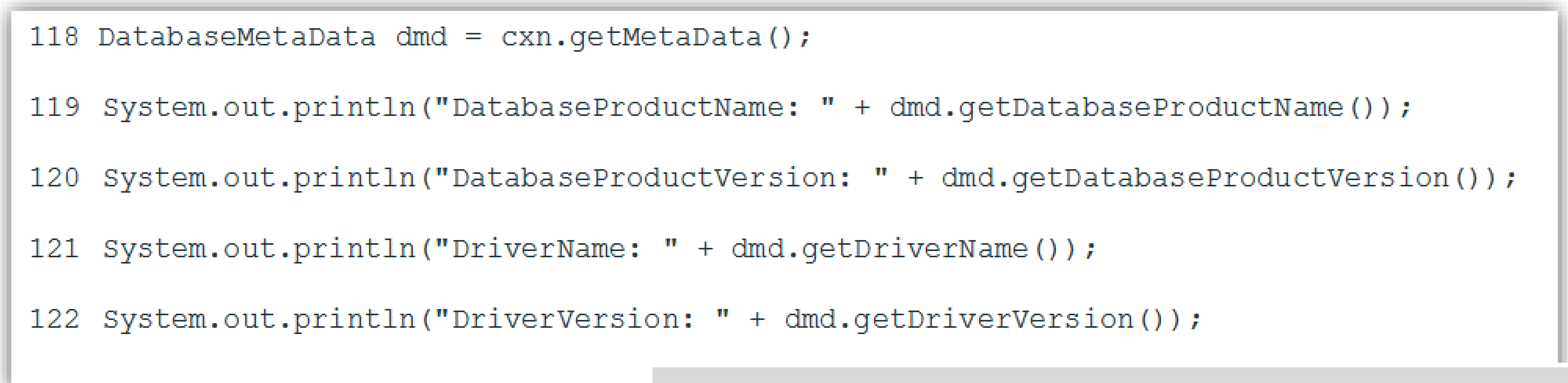
```
Connection cxn = null;
```

- Get a connection

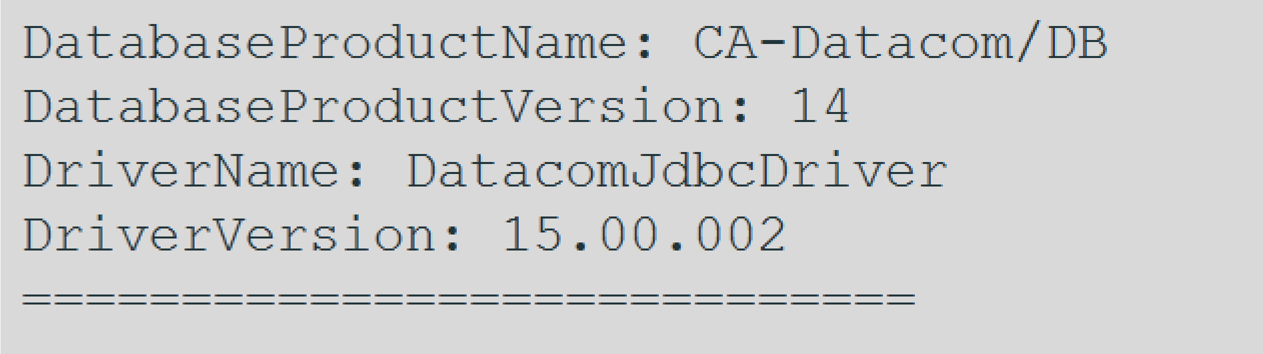


```
connectString = "jdbc:datacom://usilca11:8488/
    ServerName=SV15_MUFW,UserID=MyUser,Password=pswd,trace=3";

System.out.println(connectString);
System.out.println();

cxn = DriverManager.getConnection(connectString);
```

JDBC_Example.java
Data Base Metadata



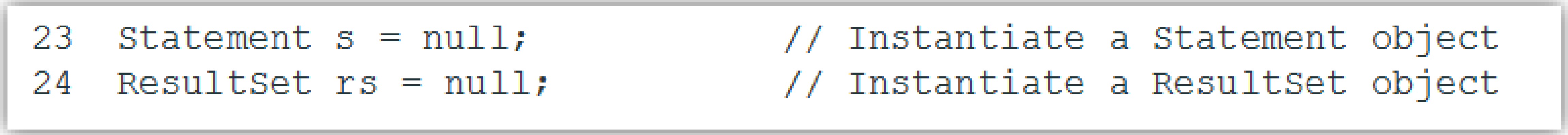
```java
DatabaseMetaData dmd = cxn.getMetaData();

System.out.println("DatabaseProductName: " + dmd.getDatabaseProductName());

System.out.println("DatabaseProductVersion: " + dmd.getDatabaseProductVersion());

System.out.println("DriverName: " + dmd.getDriverName());

System.out.println("DriverVersion: " + dmd.getDriverVersion());
```



```
DatabaseProductName: CA-Datacom/DB
DatabaseProductVersion: 14
DriverName: DatacomJdbcDriver
DriverVersion: 15.00.002
==============================
```

JDBC_Example.java – Execute a simple Statement and retrieve the results

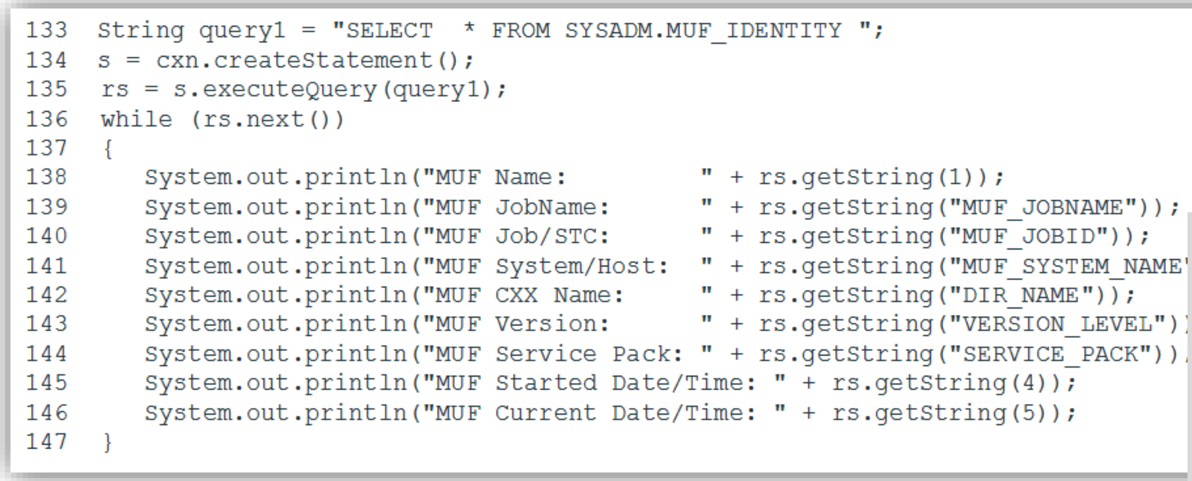
- Instantiate a Statement object and a ResultSet object



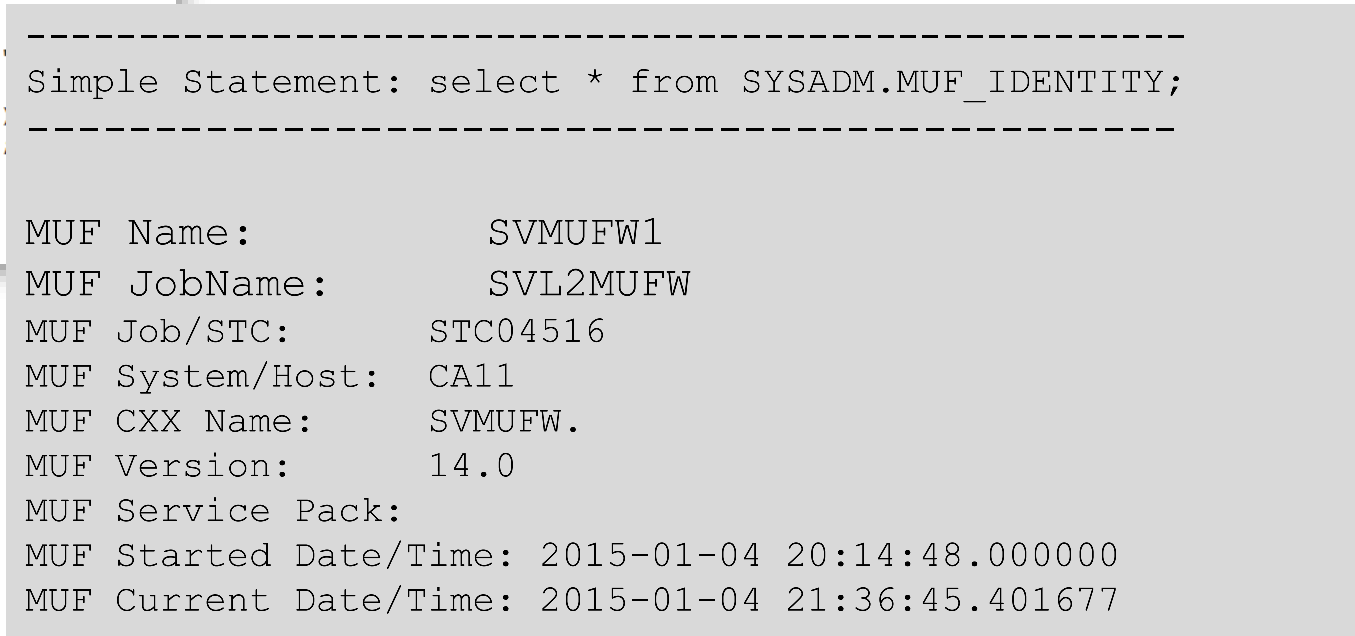
```
Statement s = null;                 // Instantiate a Statement object
ResultSet rs = null;                // Instantiate a ResultSet object
```

JDBC_Example.java – Execute a simple Statement and retrieve the results

- createStatement(), executeQuery(), next()



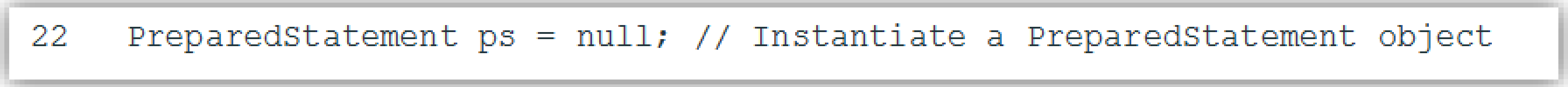
```
String query1 = "SELECT  * FROM SYSADM.MUF_IDENTITY ";
s = cxn.createStatement();
rs = s.executeQuery(query1);
while (rs.next())
{
    System.out.println("MUF Name:         " + rs.getString(1));
    System.out.println("MUF JobName:      " + rs.getString("MUF_JOBNAME"));
    System.out.println("MUF Job/STC:      " + rs.getString("MUF_JOBID"));
    System.out.println("MUF System/Host:  " + rs.getString("MUF_SYSTEM_NAME"
    System.out.println("MUF CXX Name:     " + rs.getString("DIR_NAME"));
    System.out.println("MUF Version:      " + rs.getString("VERSION_LEVEL"))
    System.out.println("MUF Service Pack: " + rs.getString("SERVICE_PACK"))
    System.out.println("MUF Started Date/Time: " + rs.getString(4));
    System.out.println("MUF Current Date/Time: " + rs.getString(5));
}
```



```
-----------------------------------------------------
Simple Statement: select * from SYSADM.MUF_IDENTITY;
----------------------------------------------------


MUF Name:         SVMUFW1
MUF JobName:      SVL2MUFW
MUF Job/STC:      STC04516
MUF System/Host:  CA11
MUF CXX Name:     SVMUFW.
MUF Version:      14.0
MUF Service Pack:
MUF Started Date/Time: 2015-01-04 20:14:48.000000
MUF Current Date/Time: 2015-01-04 21:36:45.401677
```

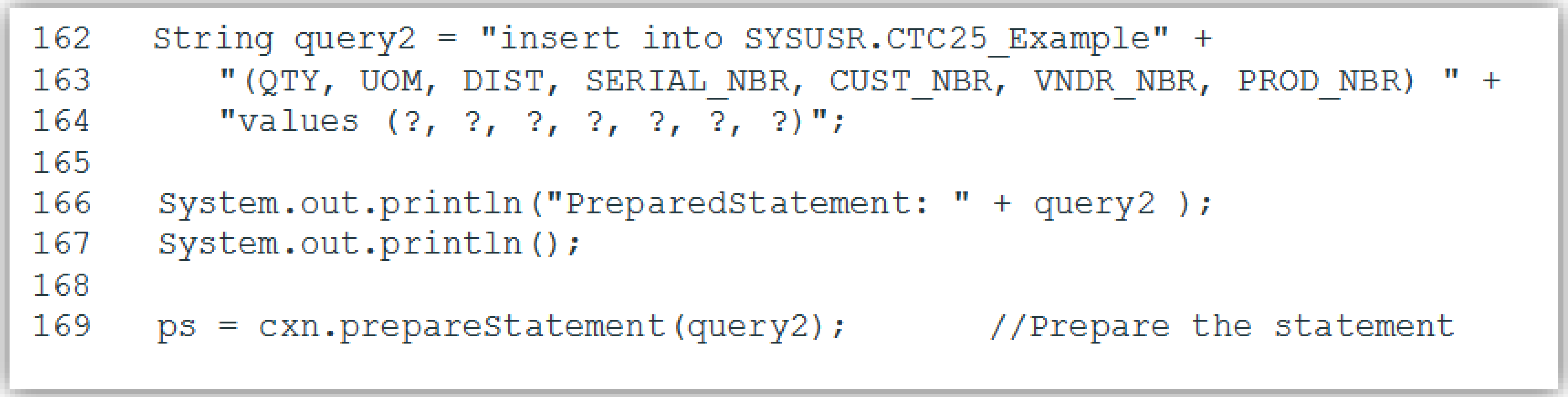
JDBC_Example.java – PreparedStatement with SQL Parameter Markers

- Instantiate a PreparedStatement object



```java
PreparedStatement ps = null; // Instantiate a PreparedStatement object
```

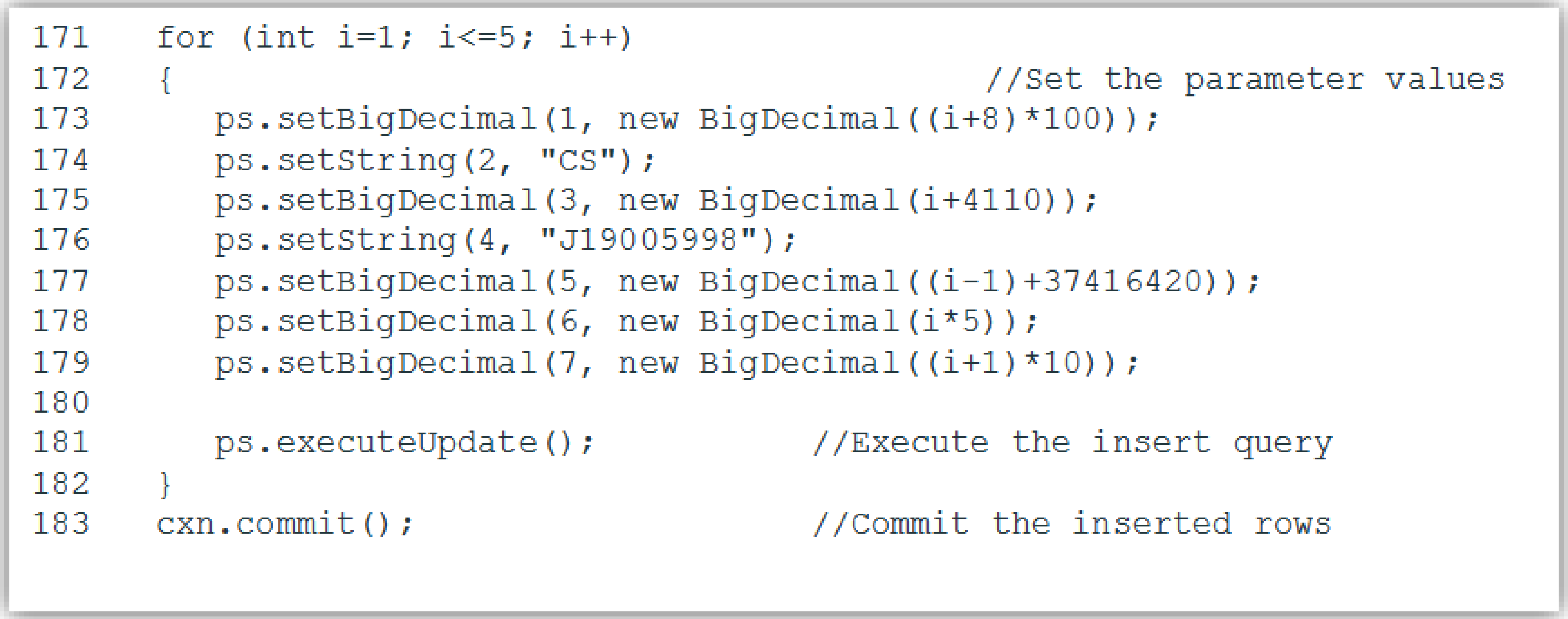
- prepareStatement()



```java
String query2 = "insert into SYSUSR.CTC25_Example" +
   "(QTY, UOM, DIST, SERIAL_NBR, CUST_NBR, VNDR_NBR, PROD_NBR) " +
   "values (?, ?, ?, ?, ?, ?, ?)";

System.out.println("PreparedStatement: " + query2 );
System.out.println();

ps = cxn.prepareStatement(query2);       //Prepare the statement
```

JDBC_Example.java – Set Parameter Values and Execute PreparedStatement

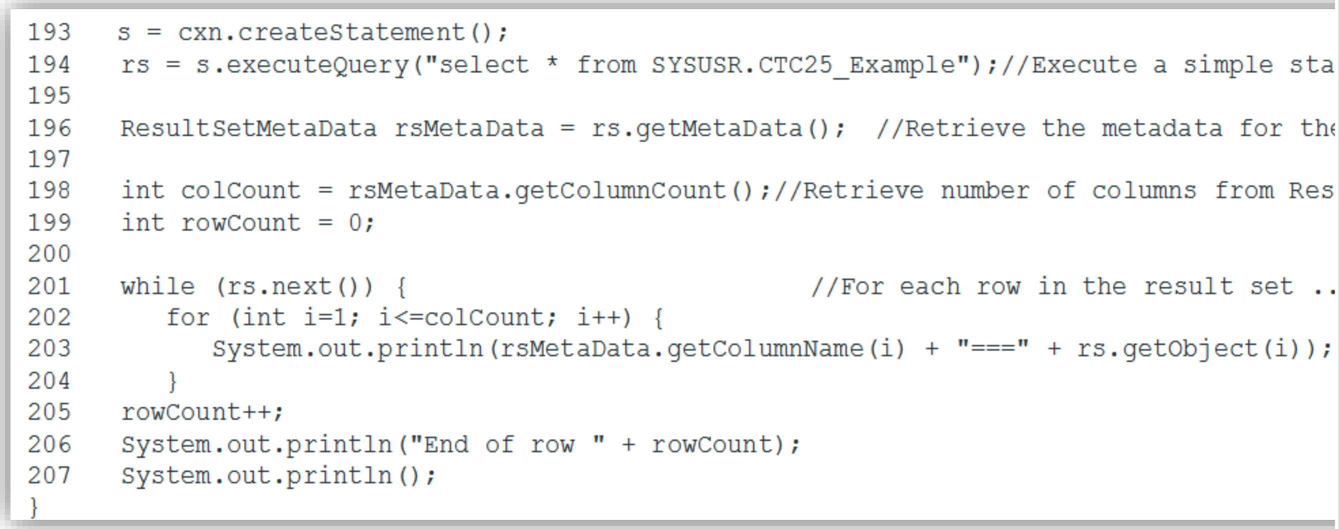
- Setter methods – setBigDecimal(), setString()
- executeUpdate()



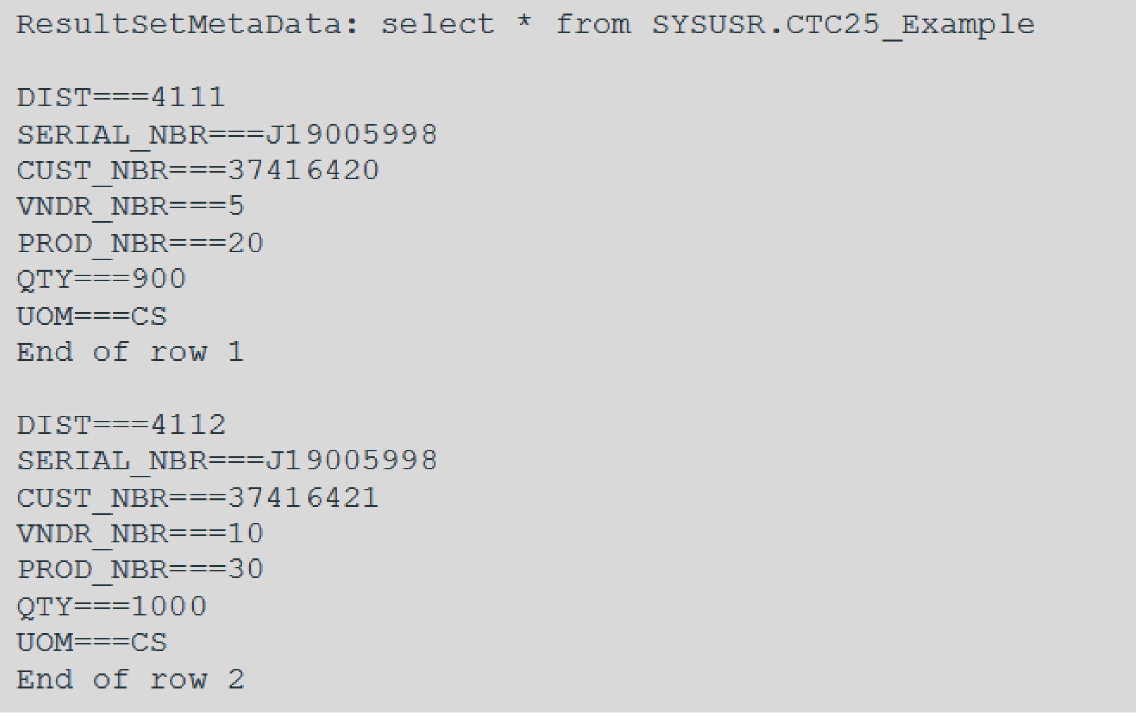
```
for (int i=1; i<=5; i++)
{                                        //Set the parameter values
   ps.setBigDecimal(1, new BigDecimal((i+8)*100));
   ps.setString(2, "CS");
   ps.setBigDecimal(3, new BigDecimal(i+4110));
   ps.setString(4, "J19005998");
   ps.setBigDecimal(5, new BigDecimal((i-1)+37416420));
   ps.setBigDecimal(6, new BigDecimal(i*5));
   ps.setBigDecimal(7, new BigDecimal((i+1)*10));

   ps.executeUpdate();          //Execute the insert query
}
cxn.commit();                   //Commit the inserted rows
```

JDBC_Example.java – Result Set Metadata

- ResultSet.getMetaData()
- ResultSetMetaData.getColumnCount(), getColumnName() and getObject()



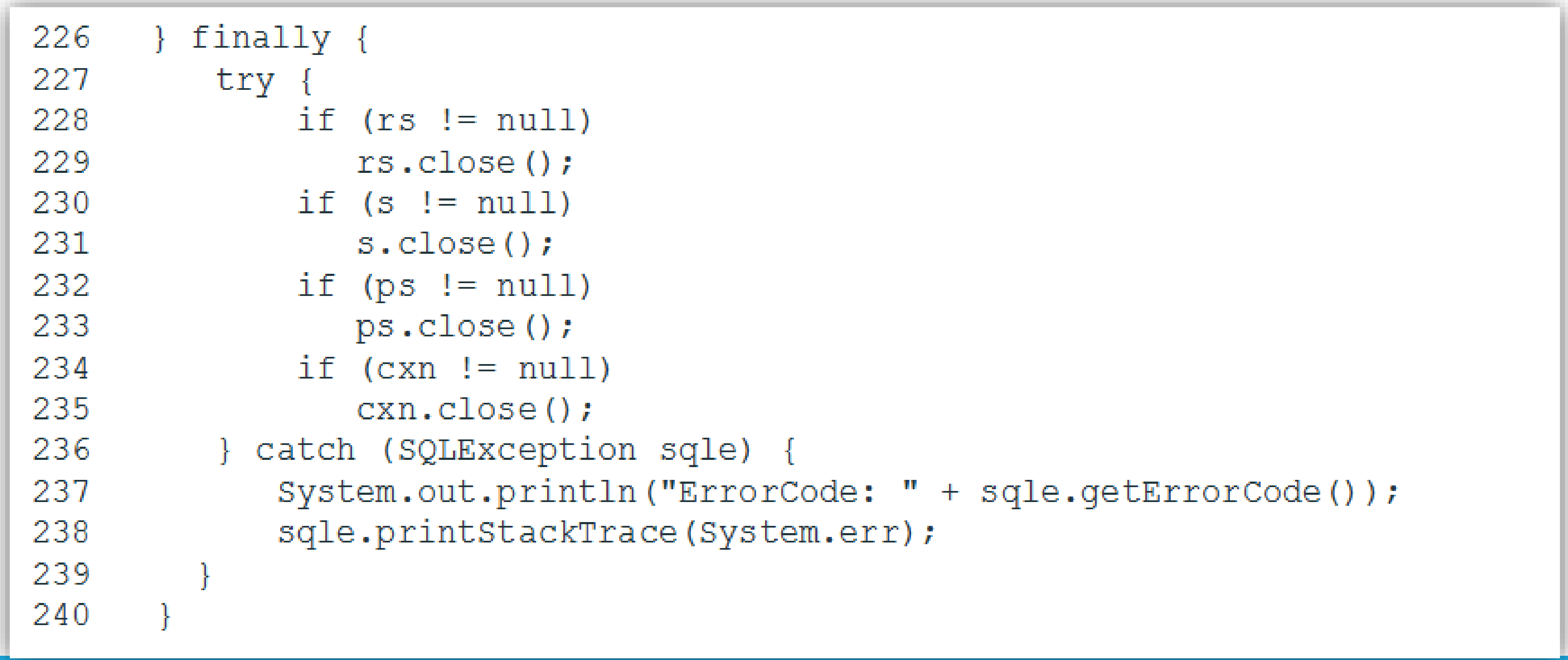
```
      s = cxn.createStatement();
      rs = s.executeQuery("select * from SYSUSR.CTC25_Example");//Execute a simple sta

      ResultSetMetaData rsMetaData = rs.getMetaData();  //Retrieve the metadata for the

      int colCount = rsMetaData.getColumnCount();//Retrieve number of columns from Res
      int rowCount = 0;

      while (rs.next()) {                           //For each row in the result set ..
         for (int i=1; i<=colCount; i++) {
            System.out.println(rsMetaData.getColumnName(i) + "===" + rs.getObject(i));
         }
      rowCount++;
      System.out.println("End of row " + rowCount);
      System.out.println();
}
```



```
ResultSetMetaData: select * from SYSUSR.CTC25_Example

DIST===4111
SERIAL_NBR===J19005998
CUST_NBR===37416420
VNDR_NBR===5
PROD_NBR===20
QTY===900
UOM===CS
End of row 1

DIST===4112
SERIAL_NBR===J19005998
CUST_NBR===37416421
VNDR_NBR===10
PROD_NBR===30
QTY===1000
UOM===CS
End of row 2
```

JDBC_Example – Closing Objects

- ResultSet.close(), Statement.close(), PreparedStatement.close(), Connection.close()



```
} finally {
   try {
       if (rs != null)
          rs.close();
       if (s != null)
          s.close();
       if (ps != null)
          ps.close();
       if (cxn != null)
          cxn.close();
   } catch (SQLException sqle) {
      System.out.println("ErrorCode: " + sqle.getErrorCode());
      sqle.printStackTrace(System.err);
  }
}
```

References

- Java™ Platform, Standard Edition 7 API Specification
  - http://docs.oracle.com/javase/7/docs/api/

- CA Datacom Tools – 15.0 – Working with Datacom Server
  - http://docops.ca.com/ca-datacom-tools/15-0/en/working-with-datacom-server

Summary

- Industry standard JDBC implementation
- Utilizes CA Datacom SQL “under the covers”
- Provides open systems access to CA Datacom data

FOR INFORMATION PURPOSES ONLY

Terms of this Presentation

This presentation was based on current information and resource allocations as of April 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?