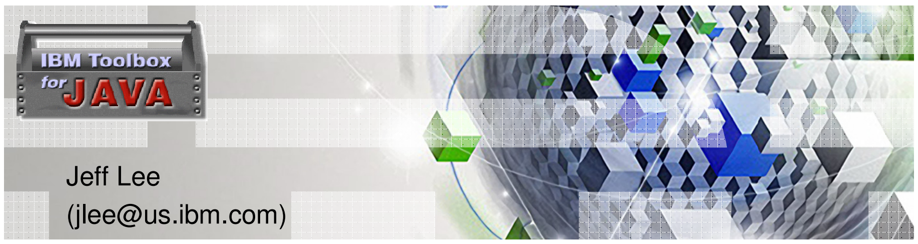
Session Number: 403552

IBM Toolbox for Java™: Advanced



Jeff Lee
(jlee@us.ibm.com)

Power your planet.

IBM Toolbox for Java™: Advanced

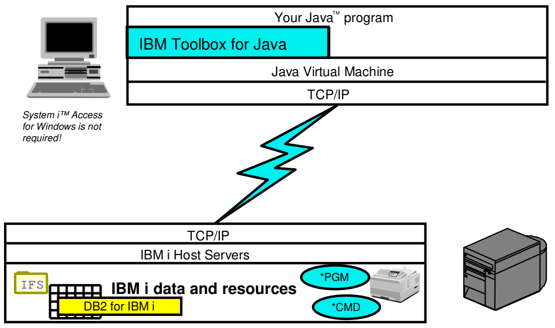
Table of Contents

- Introduction
- Using the Toolbox on IBM i
- Component list
- JTOpen (open source)
- The AS400 object
- Connection pooling
- Command call and program call
- Program Call Markup Language (PCML)
- Data queues
- User spaces
- JDBC (SQL)
- Record-level database access (DDM)
- HTML and Servlet classes
- System Debugger and Debug Manager
- JarMaker
- What's new since V6R1
- References

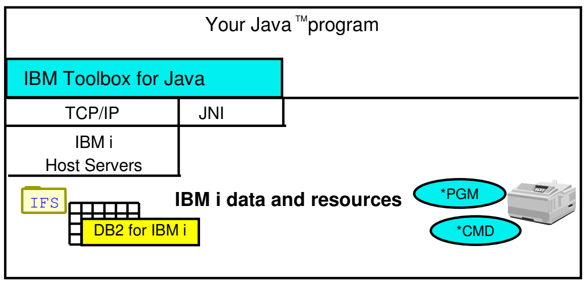
Power your planet.

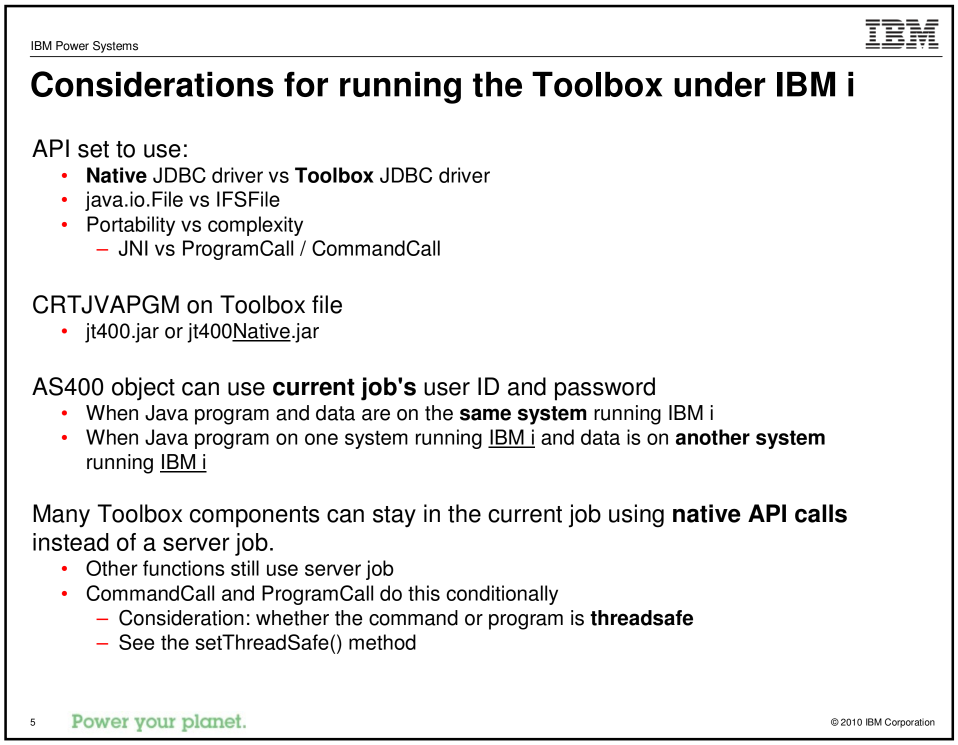
IBM Toolbox for Java™: Advanced

- The Toolbox/JTOpen is a set of Java classes and utilities which provide access to IBM i® data and resources
- Useful in client/server environments - any Java client!
  - Java **client application**
  - Java **applet** (in browser)
  - Java **servlet** - communicating with the IBM i from another web server



- Toolbox runs **optimized on IBM i** - makes direct API calls using JNI
  - Your application code is the same - the Toolbox selects its own implementation based on whether it is running on the IBM i or not
- Useful in server environments - *any* Java server
  - Server to a client application
  - Server application
  - Java servlet - running on IBM i





Considerations for running the Toolbox under IBM i

API set to use:

- **Native** JDBC driver vs **Toolbox** JDBC driver
- java.io.File vs IFSFile
- Portability vs complexity
  - JNI vs ProgramCall / CommandCall

CRTJVAPGM on Toolbox file

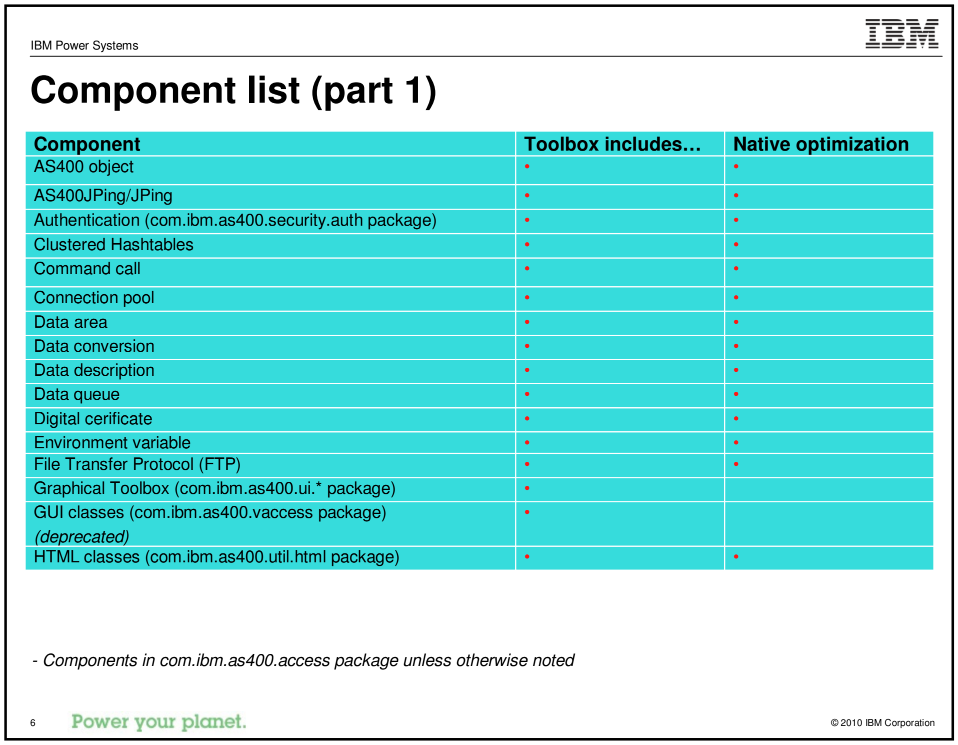
- jt400.jar or jt400Native.jar

AS400 object can use **current job's** user ID and password

- When Java program and data are on the **same system** running IBM i
- When Java program on one system running IBM i and data is on **another system** running IBM i

Many Toolbox components can stay in the current job using **native API calls** instead of a server job.

- Other functions still use server job
- CommandCall and ProgramCall do this conditionally
  - Consideration: whether the command or program is **threadsafe**
  - See the setThreadSafe() method



Component list (part 1)

Component	Toolbox includes…	Native optimization
AS400 object	•	•
AS400JPing/JPing	•	•
Authentication (com.ibm.as400.security.auth package)	•	•
Clustered Hashtables	•	•
Command call	•	•
Connection pool	•	•
Data area	•	•
Data conversion	•	•
Data description	•	•
Data queue	•	•
Digital cerificate	•	•
Environment variable	•	•
File Transfer Protocol (FTP)	•	•
Graphical Toolbox (com.ibm.as400.ui.* package)	•	
GUI classes (com.ibm.as400.vaccess package) *(deprecated)*	•	
HTML classes (com.ibm.as400.util.html package)	•	•

- Components in com.ibm.as400.access package unless otherwise noted

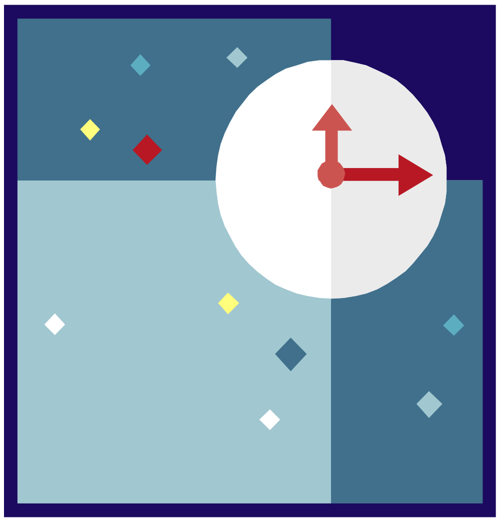
Component list (part 2)

Component	Toolbox includes…	Native optimization
Integrated file system (IFS)	•	• or use java.io.File
JarMaker	•	
Java application call	•	•
Java program information	•	•
JDBC	•	• or use native JDBC driver
Job and job log	•	•
Message file	•	•
Message queue	•	•
Micro Edition classes (com.ibm.as400.micro package)	•	•
NetServer	•	•
Permissions	•	•
Print (e.g. spooled files, printers)	•	•
Product license	•	•
Product, ProductList	•	•
Program call	•	•
Program Call Markup Language (PCML & XPCML) (com.ibm.as400.data package)	•	•
PTF, PTFCoverLetter	•	•

- Components in com.ibm.as400.access package unless otherwise noted

Component	Toolbox includes…	Native optimization
Proxy server	•	•
Record-level database access	•	•
Record Format Markup Language (RFML) (com.ibm.as400.data package)	•	•
Resource framework (com.ibm.as400.resource package) *(deprecated)*	•	•
Save File	•	•
Secure Sockets Layer (SSL)	•	•
Service program call	•	•
Servlet classes (com.ibm.as400.util.servlet package)	•	•
System Debugger (tes.jar)	•	
System pool	•	•
System status	•	•
System value	•	•
Toolbox installer *(deprecated)*	•	
Users and groups	•	•
User Space	•	•
Validation list	•	•

- Components in com.ibm.as400.access package unless otherwise noted



JTOpen (Open Source)

All of the primary Toolbox packages are open source

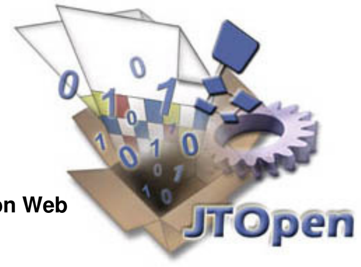
http://sourceforge.net/projects/jt400

- Part of IBM's open source development community
- Use source as a **debug tool**
- Submit new function under the **IBM Public License (IPL)**
- Modify source for your use
- Submit problem reports and bug fixes

Latest version is JTOpen 6.7

Two **delivery modes** of the Toolbox:

- Licensed program (5722-JC1 or 5761-JC1)
  - –Supported by IBM
  - –Fixes delivered by **PTFs**
- Open source version
  - –Supported by IBM and open source community
  - –New releases are available as **free downloads on Web**
  - –New functions and fixes available here first

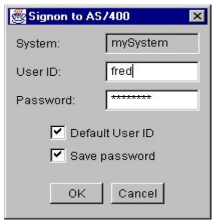


The AS400 object

Represents a **connection** to the IBM i

Encapsulates security/identity

- Password caching available
- Establish a default User ID
- Sign-on GUI if UserID/password not supplied by application
- Change password GUI when appropriate
- Provides **Secure Sockets Layer (SSL)** communication
  - Encryption and server authentication


Signon to AS/400

System: mySystem

User ID: fred

Password: *********

☑ Default User ID

☑ Save password

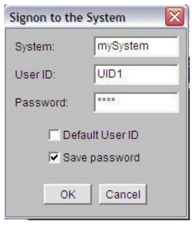
Most Toolbox classes use the AS400 object

When running on IBM i, Toolbox can exploit **current job's** user ID and password

- Use default constructor or specify "*CURRENT"

  new AS400();

  new AS400("localhost", "*CURRENT", "*CURRENT");


Signon to the System

System: mySystem

User ID: UID1

Password: ****

☐ Default User ID

☑ Save password

Represents a **connection** to the IBM i

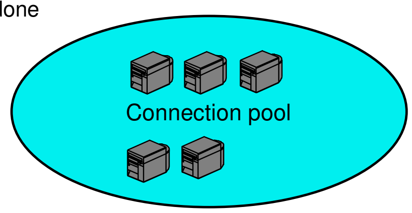
- Single vs multiple *identities*
- Single vs multiple *connections*
- Implicit vs explicit connection

```
AS400 sys   = new AS400();                   // if on client, will prompt for system, uid, pwd
AS400 sys2 = new AS400("mySystem");        // if on client, will prompt for uid, pwd
AS400 sys3 = new AS400("mySystem", "uid1", "pwd1");
AS400 sys4 = new AS400("mySystem", "uid2", "pwd2");

CommandCall cc = new CommandCall(sys);         // cc and cc2 will share a connection
CommandCall cc2 = new CommandCall(sys);
CommandCall cc3 = new CommandCall(sys3);     // cc3 and cc4 tasks will go against
CommandCall cc4 = new CommandCall(sys4);     //   different profiles
```

Connection pooling

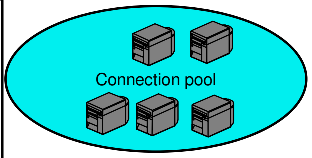
Connection pooling can **improve performance**

- Each new connection to the server can be an expensive operation
- Pooling means **reusing** AS400 objects - i.e. keeping the connection open for later
- Saves frequent disconnects and reconnects
- Common scenario: **servlets**
  - Without pooling: Create a new AS400 object for each invocation of the servlet
  - With pooling: Grab a preconnected AS400 object from the pool for each invocation of the servlet, return it when done
- Connections will be added as needed



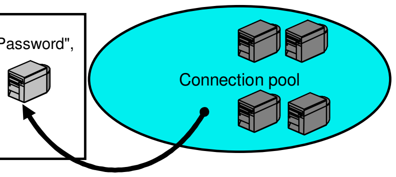
Set up the connection pool

```
AS400ConnectionPool pool = new AS400ConnectionPool();
pool.setMaxConnections(128);

// Preconnect 5 connections to the AS400.COMMAND service.
pool.fill("myAS400", "myUserID", "myPassword", AS400.COMMAND, 5);
```



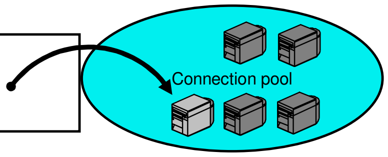
Use a connection from the pool

```
AS400 connection = pool.getConnection("myAS400", "myUserID", "myPassword",
AS400.COMMAND);

CommandCall cmd = new CommandCall(connection);
cmd.run("CRTLIB FRED");
```



Return it to the pool when done

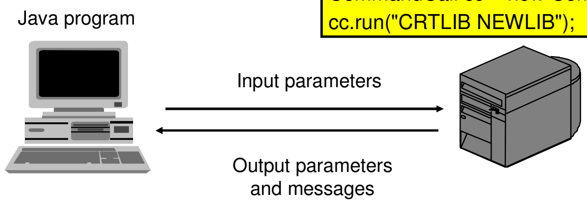
```
pool.returnConnectionToPool(connection);
```



Command call and program call

Make use of legacy code and system APIs

```
AS400 system = new AS400();
CommandCall cc = new CommandCall(system);
cc.run("CRTLIB NEWLIB");
```



```
AS400 system = new AS400();
ProgramParameter[] parmList = new ProgramParameter[n];
parmList[0] = new ProgramParameter(data);
...
ProgramCall pc = new ProgramCall(system,
                        "/QSYS.LIB/MYLIB.LIB/MYPGM.PGM", parmList);
pc.run();
```

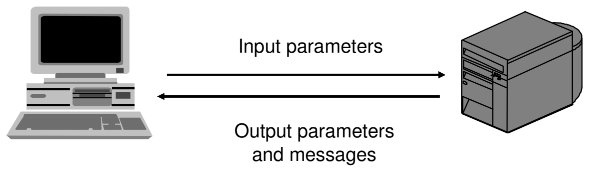
Program Call Markup Language (PCML)

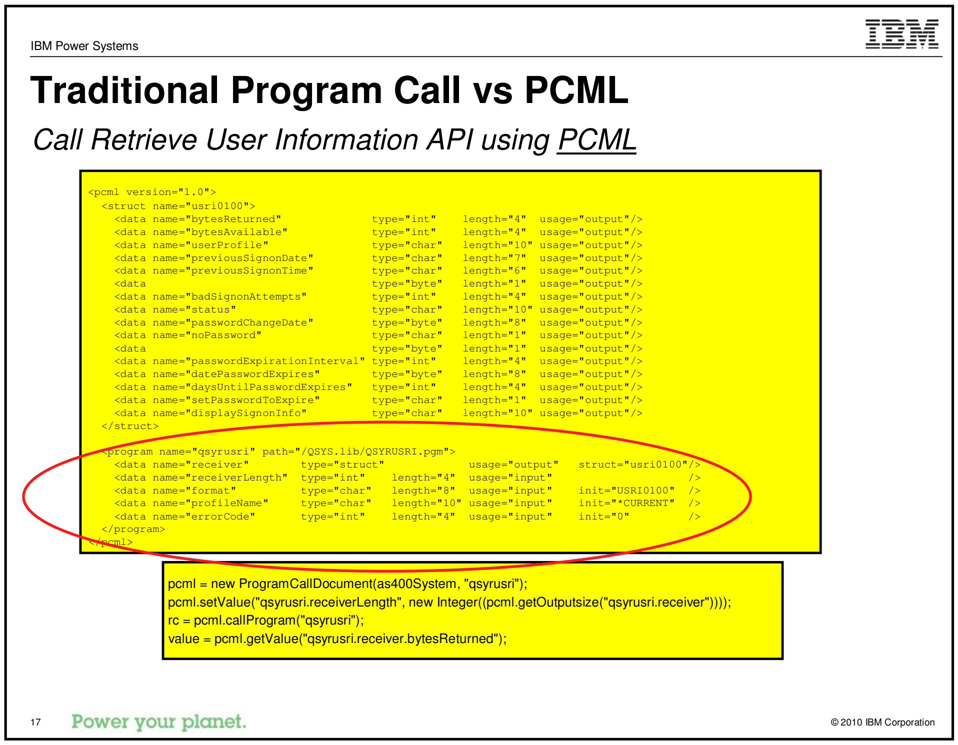
Describe program calls using XML

Parameter handling in traditional Toolbox **ProgramCall** can be **tedious**

PCML:

- **Simplifies** data description and conversion
- Iterative development **without recompile**



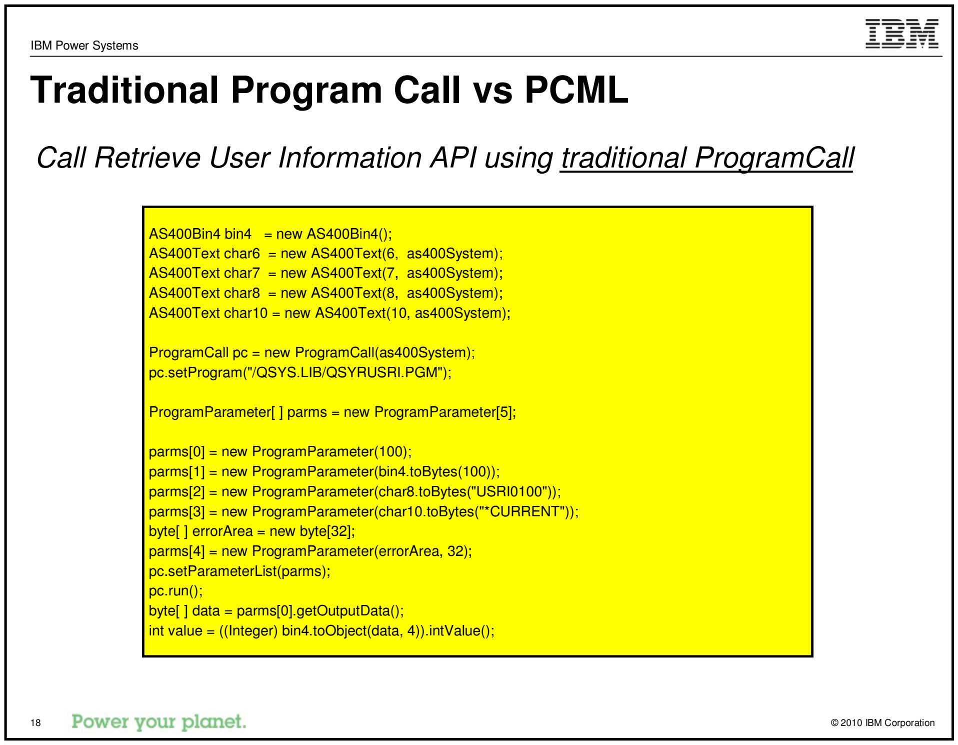


Traditional Program Call vs PCML

Call Retrieve User Information API using PCML

```
<pcml version="1.0">
  <struct name="usri0100">
    <data name="bytesReturned"              type="int"    length="4"  usage="output"/>
    <data name="bytesAvailable"             type="int"    length="4"  usage="output"/>
    <data name="userProfile"                type="char"   length="10" usage="output"/>
    <data name="previousSignonDate"         type="char"   length="7"  usage="output"/>
    <data name="previousSignonTime"         type="char"   length="6"  usage="output"/>
    <data                                   type="byte"   length="1"  usage="output"/>
    <data name="badSignonAttempts"          type="int"    length="4"  usage="output"/>
    <data name="status"                     type="char"   length="10" usage="output"/>
    <data name="passwordChangeDate"         type="byte"   length="8"  usage="output"/>
    <data name="noPassword"                 type="char"   length="1"  usage="output"/>
    <data                                   type="byte"   length="1"  usage="output"/>
    <data name="passwordExpirationInterval" type="int"    length="4"  usage="output"/>
    <data name="datePasswordExpires"        type="byte"   length="8"  usage="output"/>
    <data name="daysUntilPasswordExpires"   type="int"    length="4"  usage="output"/>
    <data name="setPasswordToExpire"        type="char"   length="1"  usage="output"/>
    <data name="displaySignonInfo"          type="char"   length="10" usage="output"/>
  </struct>

  <program name="qsyrusri" path="/QSYS.lib/QSYRUSRI.pgm">
    <data name="receiver"       type="struct"             usage="output"   struct="usri0100"/>
    <data name="receiverLength" type="int"    length="4"  usage="input"                     />
    <data name="format"         type="char"   length="8"  usage="input"    init="USRI0100"  />
    <data name="profileName"    type="char"   length="10" usage="input"    init="*CURRENT"  />
    <data name="errorCode"      type="int"    length="4"  usage="input"    init="0"         />
  </program>
</pcml>
```

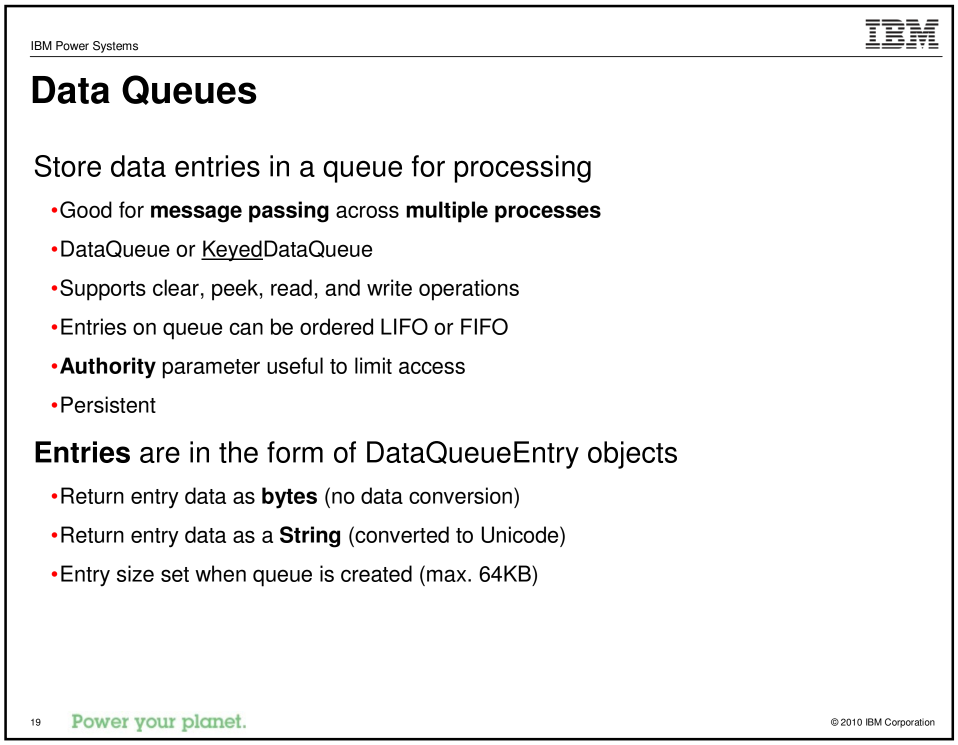
```
pcml = new ProgramCallDocument(as400System, "qsyrusri");
pcml.setValue("qsyrusri.receiverLength", new Integer((pcml.getOutputsize("qsyrusri.receiver"))));
rc = pcml.callProgram("qsyrusri");
value = pcml.getValue("qsyrusri.receiver.bytesReturned");
```



Traditional Program Call vs PCML

Call Retrieve User Information API using traditional ProgramCall

```
AS400Bin4 bin4   = new AS400Bin4();
AS400Text char6  = new AS400Text(6,  as400System);
AS400Text char7  = new AS400Text(7,  as400System);
AS400Text char8  = new AS400Text(8,  as400System);
AS400Text char10 = new AS400Text(10, as400System);

ProgramCall pc = new ProgramCall(as400System);
pc.setProgram("/QSYS.LIB/QSYRUSRI.PGM");

ProgramParameter[ ] parms = new ProgramParameter[5];

parms[0] = new ProgramParameter(100);
parms[1] = new ProgramParameter(bin4.toBytes(100));
parms[2] = new ProgramParameter(char8.toBytes("USRI0100"));
parms[3] = new ProgramParameter(char10.toBytes("*CURRENT"));
byte[ ] errorArea = new byte[32];
parms[4] = new ProgramParameter(errorArea, 32);
pc.setParameterList(parms);
pc.run();
byte[ ] data = parms[0].getOutputData();
int value = ((Integer) bin4.toObject(data, 4)).intValue();
```



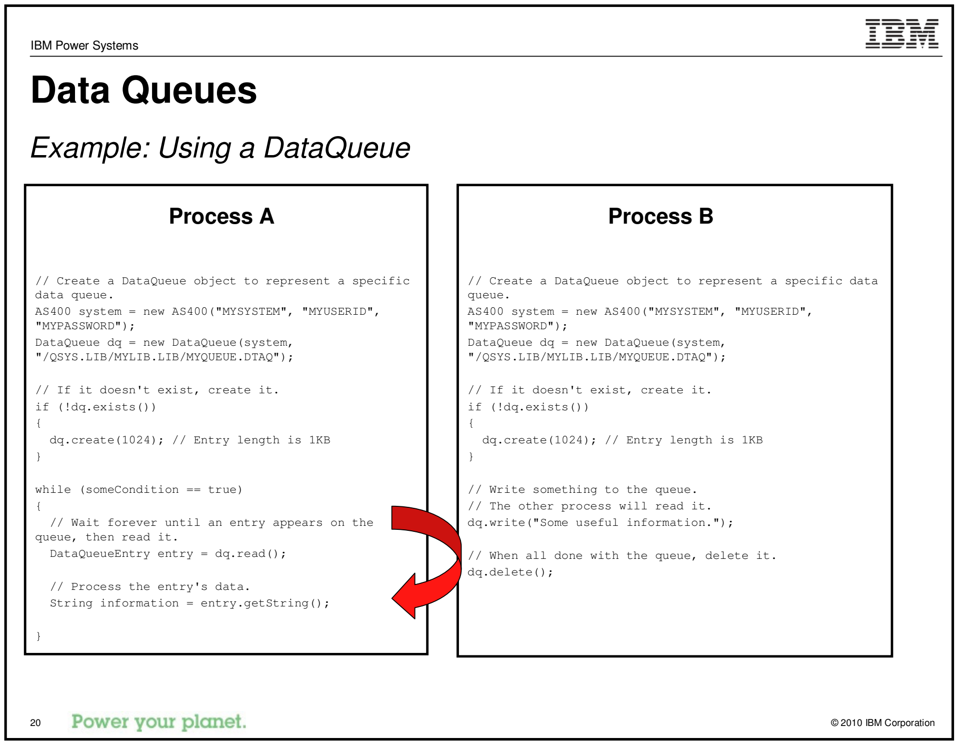
Data Queues

Store data entries in a queue for processing

- Good for **message passing** across **multiple processes**
- DataQueue or <u>Keyed</u>DataQueue
- Supports clear, peek, read, and write operations
- Entries on queue can be ordered LIFO or FIFO
- **Authority** parameter useful to limit access
- Persistent

Entries are in the form of DataQueueEntry objects

- Return entry data as **bytes** (no data conversion)
- Return entry data as a **String** (converted to Unicode)
- Entry size set when queue is created (max. 64KB)



Data Queues

Example: Using a DataQueue

Process A

```
// Create a DataQueue object to represent a specific
data queue.
AS400 system = new AS400("MYSYSTEM", "MYUSERID",
"MYPASSWORD");
DataQueue dq = new DataQueue(system,
"/QSYS.LIB/MYLIB.LIB/MYQUEUE.DTAQ");

// If it doesn't exist, create it.
if (!dq.exists())
{
  dq.create(1024); // Entry length is 1KB
}

while (someCondition == true)
{
  // Wait forever until an entry appears on the
queue, then read it.
  DataQueueEntry entry = dq.read();

  // Process the entry's data.
  String information = entry.getString();

}
```

Process B

```
// Create a DataQueue object to represent a specific data
queue.
AS400 system = new AS400("MYSYSTEM", "MYUSERID",
"MYPASSWORD");
DataQueue dq = new DataQueue(system,
"/QSYS.LIB/MYLIB.LIB/MYQUEUE.DTAQ");

// If it doesn't exist, create it.
if (!dq.exists())
{
  dq.create(1024); // Entry length is 1KB
}

// Write something to the queue.
// The other process will read it.
dq.write("Some useful information.");

// When all done with the queue, delete it.
dq.delete();
```

User Spaces

Store data in an indexed memory "space"

- Good for **sharing** common data across multiple processes
- Supports read and write operations
- Specify **offset** to index inside the user space
- Set initial value and length properties
- Max. length is just under 16MB
- **Authority** parameter useful to limit access
- Persistent

*Some IBM i APIs return output data in a **user space** instead of in a ProgramCall **output parameter***

RFML (Record Format Markup Language)

Very similar to PCML (Program Call Markup Language)

While PCML is designed only for Program Parameters,
RFML is useful for **parsing/composing**:
- Data queue entries
- User spaces
- Physical file records
- Data buffers

Specify record formats using **XML**; get/set field values

Segregate the **data layout** from the **program logic**

RFML vs. FieldDescription

Example: Composing a customer record

Using RFML:

```
import com.ibm.as400.data.RecordFormatDocument;

RecordFormatDocument rfmlDoc =
                        new RecordFormatDocument("customer");

( In a separate file named "customer.rfml": )

<rfml version="4.0" ccsid="37">
  <recordformat name="cusrec">
    <data name="cusnum" type="int"   length="2" precision="16"/>
    <data name="lstnam" type="char"  length="8"/>
    <data name="baldue" type="zoned" length="6" precision="2"/>
  </recordformat>
</rfml>
```

Without RFML:

```
import com.ibm.as400.access.AS400Text;
import com.ibm.as400.access.AS400UnsignedBin2;
import com.ibm.as400.access.AS400ZonedDecimal;
import com.ibm.as400.access.BinaryFieldDescription;
import com.ibm.as400.access.CharacterFieldDescription;
import com.ibm.as400.access.RecordFormat;
import com.ibm.as400.access.ZonedDecimalFieldDescription;

RecordFormat recFmt = new RecordFormat("cusrec");

AS400UnsignedBin2 conv1 = new AS400UnsignedBin2();
BinaryFieldDescription desc1 = new BinaryFieldDescription(conv1, "cusnum");
recFmt.addFieldDescription(desc1);

AS400Text conv2 = new AS400Text(8, 37);
CharacterFieldDescription desc2 = new CharacterFieldDescription(conv2,
"lstnam");
recFmt.addFieldDescription(desc2);

AS400ZonedDecimal conv3 = new AS400ZonedDecimal(6, 2);
ZonedDecimalFieldDescription desc3 = new
ZonedDecimalFieldDescription(conv3, "baldue");
recFmt.addFieldDescription(desc3);
```

JDBC

The Java standard for database access

Write Java programs in terms of standard JDBC interfaces, then plug in ***any*** JDBC driver - to work with ***any*** database!

- Java gives you platform independence; JDBC gives you database independence

java.sql package in Java Developers Kit

SQL is used extensively

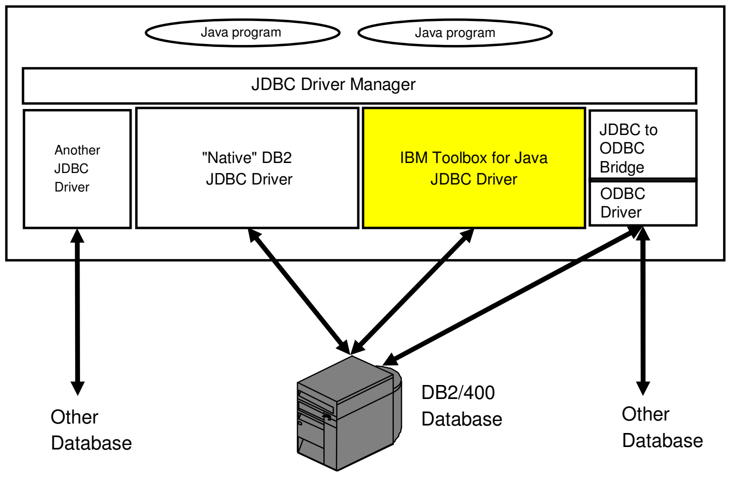
- Based on X/Open SQL Call Level Interface

Also supports:

- Database definitions, manipulations, and queries
- Stored procedures
- Catalog methods
- Transactions (commit, rollback, isolation levels, distributed)

JDBC

The Java standard for SQL database access



JDBC

Registering a JDBC driver

A JDBC driver must be **registered** with the DriverManager:

- Most JDBC drivers will register themselves when they are **loaded**:
  - Class.forName("*JDBC.driver.class.name*"); // this is the preferred method
- You can also register JDBC drivers **explicitly**:
  - DriverManager.**registerDriver**(new *JDBC.driver.class.name*());
- The DriverManager can now dispatch requests to the registered JDBC driver

```
// Register using a Java property
java   -Djdbc.drivers=com.ibm.as400.access.AS400JDBCDriver   myProgram

// Register by writing Java code
java.sql.DriverManager.registerDriver(new com.ibm.as400.access.AS400JDBCDriver());
java.sql.DriverManager.registerDriver(new com.ibm.db2.jdbc.app.DB2Driver());
```

JDBC

Connecting to a database

- Use the DriverManager to **connect** to a database
  - Connection connection = DriverManager.**getConnection**("jdbc:*your-database's-URL*");
- Userid and password are optional
- The DriverManager will dispatch the connection request to the *appropriate* JDBC driver
- Some drivers recognize additional **connection properties**

```
Properties connProps = new Properties();
connProps.put("cursor hold", "0");
connProps.put("date format", "iso");

Connection c  = DriverManager.getConnection("jdbc:as400://mySystem", connProps);
```

IBM i JDBC driver choices

IBM Toolbox for Java JDBC driver

com.ibm.as400.access.AS400JDBCDriver

- Communicates with the **database host server** using TCP/IP **sockets**
- Provides extended dynamic performance optimizations
- Great for:
  - Client/server applications
  - Applets
  - Servlets where the Web server and data are **not on the same IBM i** system

"Native" DB2 JDBC driver

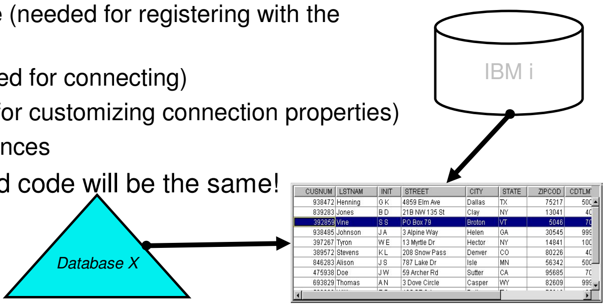
com.ibm.db2.jdbc.app.DB2Driver

- Communicates with the database using **direct CLI calls**
- Great for:
  - Server applications
  - Servlets where the **Web server and data** are **on the same IBM i** system
- Toolbox JDBC driver can switch to use the DB2 driver
  - Use the JDBC property "**driver=native**" on the connection URL

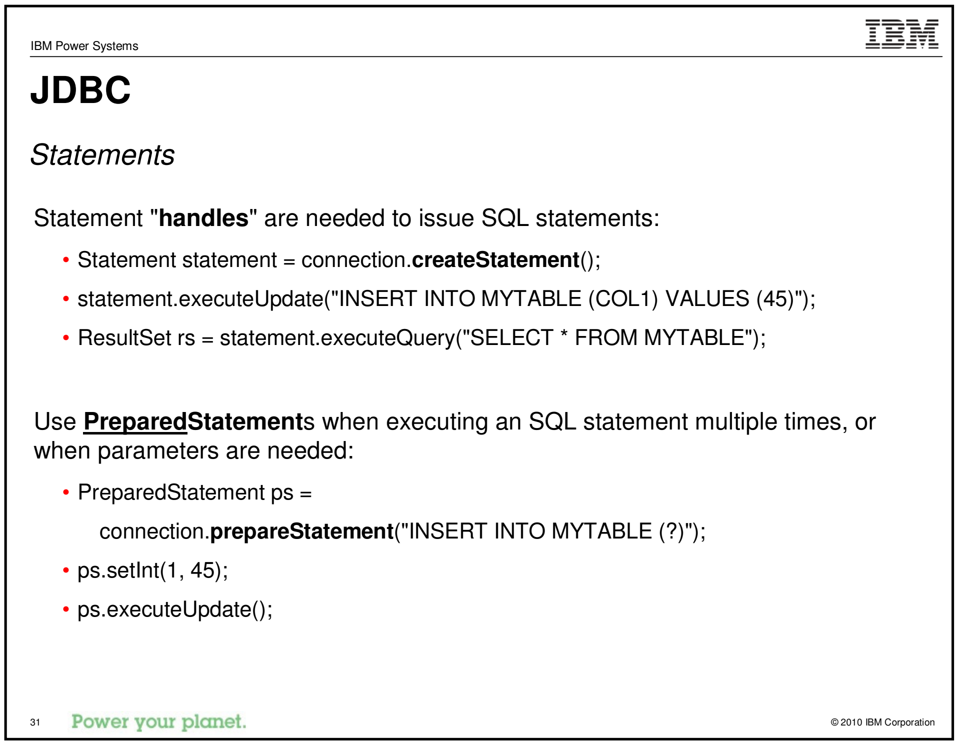
JDBC

Code your program to be configurable

- **Don't hardcode** a JDBC driver
  - Allow your end users to plug in other JDBC drivers
  - Now your program works with *any* database!
- Differences between JDBC drivers:
  - Driver **class name** (needed for registering with the DriverManager)
  - **URL** syntax (needed for connecting)
  - **Properties** (used for customizing connection properties)
  - Subtle **SQL** differences
- Most of the logic and code will be the same!



CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT
938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	500
839283	Jones	B D	21B NW 135 St	Clay	NY	13041	40
392859	Vine	S S	PO Box 79	Broton	VT	5046	70
938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	999
397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	100
389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	40
846283	Alison	J S	787 Lake Dr	Isle	MN	56342	500
475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	70
693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	999



JDBC

Statements

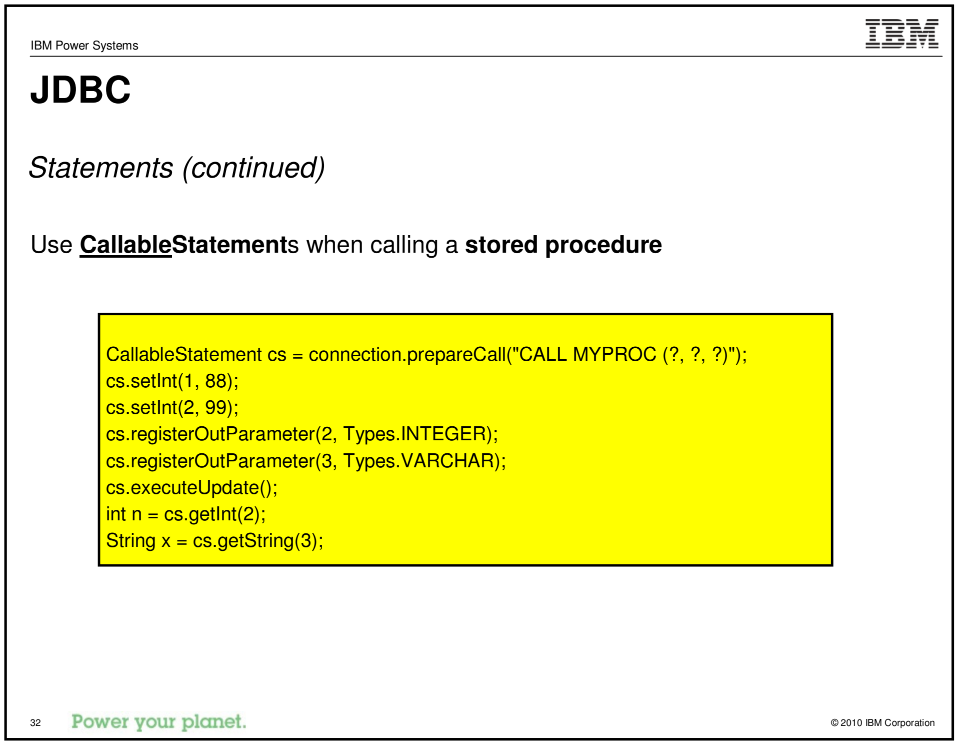
Statement "**handles**" are needed to issue SQL statements:

- Statement statement = connection.**createStatement**();
- statement.executeUpdate("INSERT INTO MYTABLE (COL1) VALUES (45)");
- ResultSet rs = statement.executeQuery("SELECT * FROM MYTABLE");

Use **Prepared****Statement**s when executing an SQL statement multiple times, or when parameters are needed:

- PreparedStatement ps =

  connection.**prepareStatement**("INSERT INTO MYTABLE (?)");
- ps.setInt(1, 45);
- ps.executeUpdate();



JDBC

Statements (continued)

Use **Callable****Statement**s when calling a **stored procedure**

```
CallableStatement cs = connection.prepareCall("CALL MYPROC (?, ?, ?)");
cs.setInt(1, 88);
cs.setInt(2, 99);
cs.registerOutParameter(2, Types.INTEGER);
cs.registerOutParameter(3, Types.VARCHAR);
cs.executeUpdate();
int n = cs.getInt(2);
String x = cs.getString(3);
```

JDBC

ResultSets

ResultSets contain the **result data from a query**

- **ResultSet** rs = statement.**executeQuery**("SELECT * FROM MYTABLE");
- String value = rs.getString("COLUMNA");

ResultSetMetaData objects describe the **columns** in a ResultSet

- **ResultSetMetaData** rsmd = rs.**getMetaData**();
- String columnName = rsmd.getColumnName(1);
- int displaySize = rsmd.getColumnDisplaySize(1);

JDBC

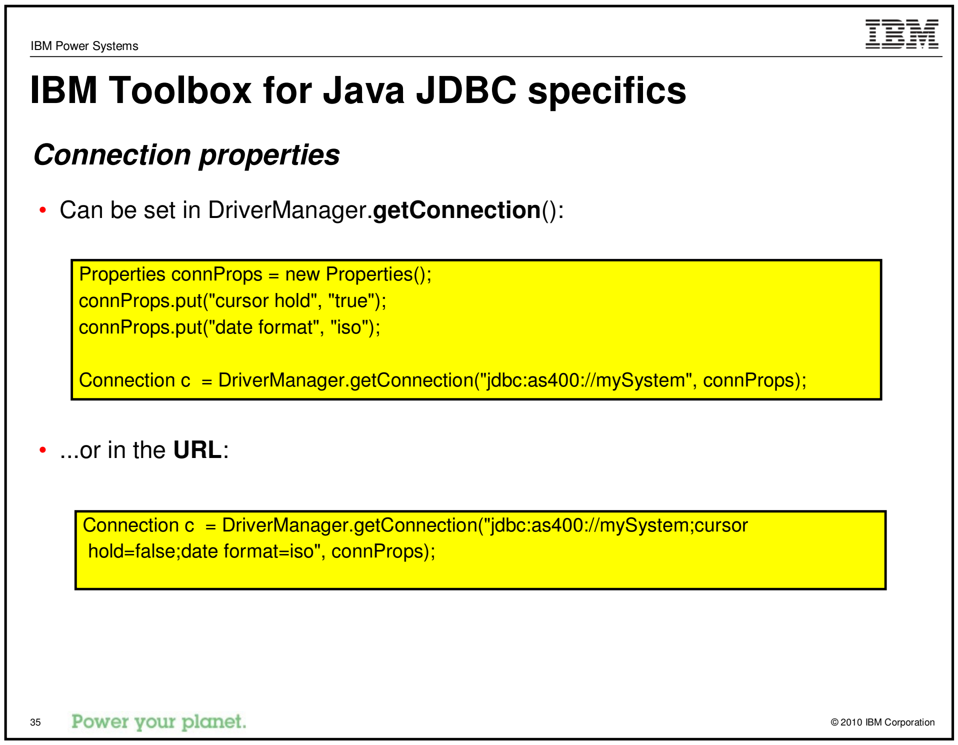
What else is there?

DatabaseMetaData
- Information about tables, columns, procedures, ...

SQLExceptions and SQLWarnings
- Used for error handling

JDBC 3.0
- Savepoints
- Parameter meta data
- BLOB and CLOB methods
- Independent auxiliary storage pools (IASPs)

JDBC 4.0 ★
- SQL XML data type
- Enhanced exception management – new exception subclasses
- Wrapper pattern support
- Client info support
- ***Requires Java 6.0 JVM (JDK 1.6)*** ★
- ***Not downward-compatible to prior JDKs*** ★



IBM Toolbox for Java JDBC specifics

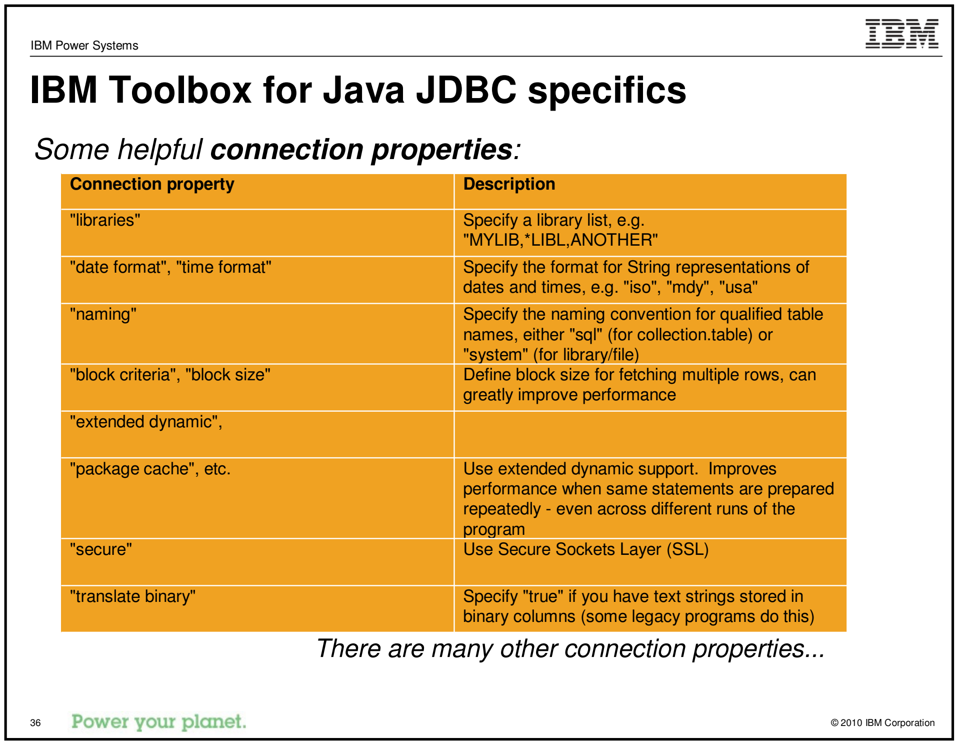
Connection properties

- Can be set in DriverManager.**getConnection**():

```
Properties connProps = new Properties();
connProps.put("cursor hold", "true");
connProps.put("date format", "iso");

Connection c  = DriverManager.getConnection("jdbc:as400://mySystem", connProps);
```

- ...or in the **URL**:

```
Connection c  = DriverManager.getConnection("jdbc:as400://mySystem;cursor
 hold=false;date format=iso", connProps);
```

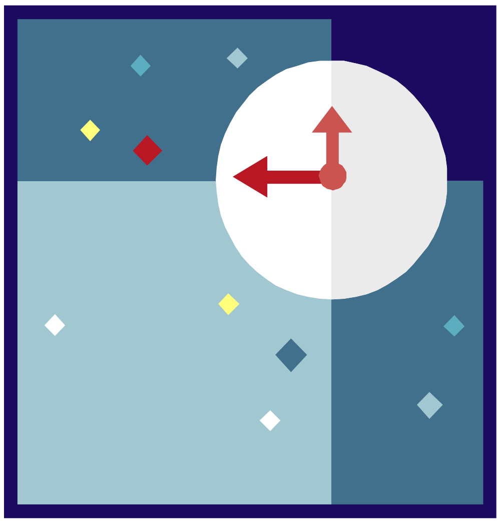


IBM Toolbox for Java JDBC specifics

Some helpful ***connection properties***:

Connection property	**Description**
"libraries"	Specify a library list, e.g. "MYLIB,*LIBL,ANOTHER"
"date format", "time format"	Specify the format for String representations of dates and times, e.g. "iso", "mdy", "usa"
"naming"	Specify the naming convention for qualified table names, either "sql" (for collection.table) or "system" (for library/file)
"block criteria", "block size"	Define block size for fetching multiple rows, can greatly improve performance
"extended dynamic",	
"package cache", etc.	Use extended dynamic support. Improves performance when same statements are prepared repeatedly - even across different runs of the program
"secure"	Use Secure Sockets Layer (SSL)
"translate binary"	Specify "true" if you have text strings stored in binary columns (some legacy programs do this)

There are many other connection properties...



Record-level database access

Fast access to IBM i database files

Provides access to **physical** and **logical files**:

- Access records **sequentially**, or by **record number** or **key**
- Support for **locking**
- Support for **transactions** (commit and rollback)

Options for **describing** the **Record Format**:

- The programmer can write the code
- The Toolbox can retrieve the record format at **development-time** and output Java source code
- The Toolbox can retrieve the record format at **run-time**

Fast!

When running on IBM i, direct API calls are made instead of using the host server (these are known as *"**native optimizations**"*)

Record-level database access

Fast access to IBM i database files

```
QSYSObjectPathName fileName = new QSYSObjectPathName("QIWS", "QCUSTCDT", "FILE");

SequentialFile file = new SequentialFile(as400, fileName.getPath());

file.setRecordFormat();      // Loads the record format directly from the server.

file.open();

Record data = file.readNext();

while (data != null)
{
  System.out.print((String)data.getField("INIT") + "  ");
  System.out.print((String)data.getField("LSTNAM") + "  ");
  System.out.println((BigDecimal)data.getField("BALDUE"));
  data = file.readNext();
}
```

Record-level database access

Fast access to IBM i database files

Performance tips

- Avoid retrieving the record format multiple times. Retrieve it once and save a reference to it, or hard code the record format.
- Blocking factor means record caching. Experiment with different sizes or specify zero and let the Toolbox determine the blocking factor.
- Blocking factor is valid only when the file is opened for READ_ONLY or WRITE_ONLY access.
- Opening keyed files is slower than opening sequential files. Use sequential files unless you need to specifically search by key.

HTML and Servlet classes

*Web components create **tables** and **forms***

Provides **access** to database files:

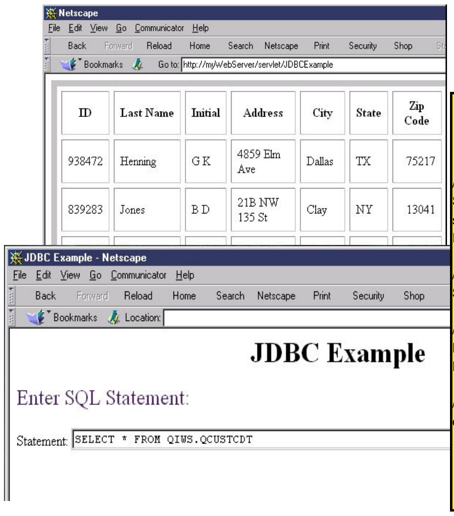
- Access database file with Record Level Access or SQL via JDBC
- Includes Meta Data

Provides classes to **display data**:

- Display data in tables or forms
- Toolbox provides converters that will produce HTML tables or forms based on the row data

```
HTMLTableConverter converter = new HTMLTableConverter();

ResultSet resultSet = statement.getResultSet();
SQLResultSetRowData rowdata = new SQLResultSetRowData(resultSet);

String[ ] html = converter.convert(rowdata);
out.println(html[0]);
```

Web components create tables and forms


Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Go to: http://myWebServer/servlet/JDBCExample

ID	Last Name	Initial	Address	City	State	Zip Code
938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217
839283	Jones	B D	21B NW 135 St	Clay	NY	13041

JDBC Example - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location:

JDBC Example

Enter SQL Statement:

Statement: SELECT * FROM QIWS.QCUSTCDT

- Classes for generating HTML output
- Useful for servlets, report generating, etc.

```
// Execute an SQL statement and get the result set.
Statement statement = connection.createStatement();
statement.execute("SELECT * FROM QIWS.QCUSTCDT");
ResultSet resultSet = statement.getResultSet();

// Create the SQLResultSetRowData object and initialize to the result set.
SQLResultSetRowData rowData = new SQLResultSetRowData(resultSet);

// Create an HTML converter object and convert the rowData to HTML.
HTMLTableConverter conv = new HTMLTableConverter();
HTMLTable[ ] html = conv.convertToTables(rowData);

// Display the HTML table generated by the converter.
out.println(html[0]);
```

HTML and Servlet classes

Web components create tree hierarchy

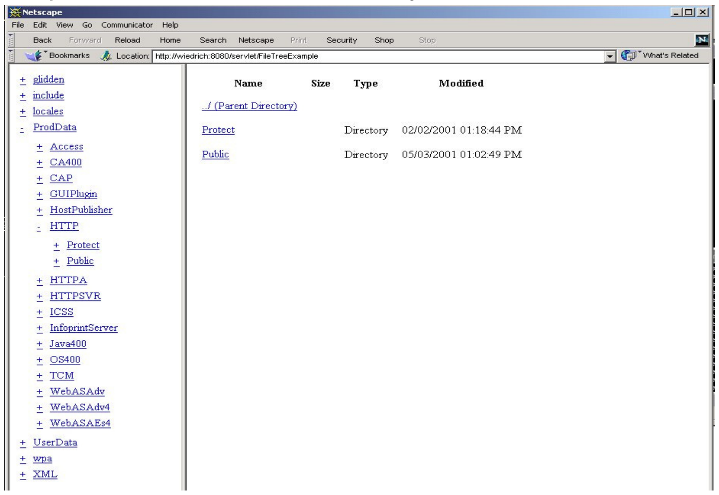
Provides classes to display the **Integrated File System**:

- Display contents of the Integrated File System
- Toolbox provides classes to create and display a customized and **traversable** tree

```
HTMLTree tree = new HTMLTree(HTTPrequest)

IFSJavaFile root = new IFSJavaFile(systemObject, "/QIBM");

DirFilter filter = new DirFilter();

File[] dirList = root.listFiles(filter);

for (int i=0; i<dirList.length; i++)
    {
    FileTreeElement node = new FileTreeElement(dirList[i]);
    tree.addElement(node);
    }
```

HTML and Servlet classes

Web components create tree hierarchy

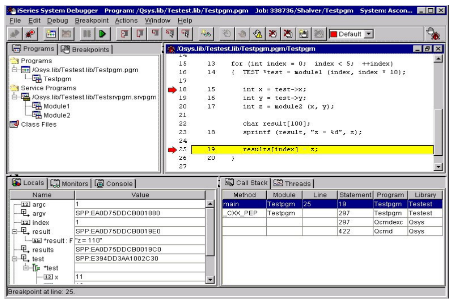

Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location:
What's Related

http://wiedrich:8080/servlet/FileTreeExample

- + glidden
- + include
- + locales
- - ProdData
  - + Access
  - + CA400
  - + CAP
  - + GUIPlugin
  - + HostPublisher
  - - HTTP
    - + Protect
    - + Public
  - + HTTPA
  - + HTTPSVR
  - + ICSS
  - + InfoprintServer
  - + Java400
  - + OS400
  - + TCM
  - + WebASAdv
  - + WebASAdv4
  - + WebASAEs4
- + UserData
- + wpa
- + XML

Name	Size	Type	Modified
../ (Parent Directory)			
Protect		Directory	02/02/2001 01:18:44 PM
Public		Directory	05/03/2001 01:02:49 PM

System Debugger and Debug Manager



iSeries System Debugger   Program: /Qsys.lib/Testest.lib/Testpgm.pgm   Job: 338736/Shalver/Testpgm   System: Ascon...

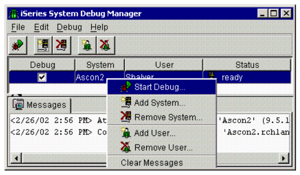
File Edit Debug Breakpoint Actions Window Help
Default
Programs
Breakpoints
Programs
/Qsys.lib/Testest.lib/Testpgm.pgm
Testpgm
Service Programs
/Qsys.lib/Testest.lib/Testsrvpgm.srvpgm
Module1
Module2
Class Files
/Qsys.lib/Testest.lib/Testpgm.pgm/Testpgm
Locals
Monitors
Console
Call Stack
Threads

```
14
15    13    for (int index = 0;  index < 5;  ++index)
16    14    {  TEST *test = module1 (index, index * 10);
17
18    15       int x = test->x;
19    16       int y = test->y;
20    17       int z = module2 (x, y);
21
22             char result[100];
23    18       sprintf (result, "z = %d", z);
24
25    19       results[index] = z;
26    20    }
27
```

Name	Value
argc	1
argv	SPP:EA0D75DDCB001880
index	1
result	SPP:EA0D75DDCB0019E0
*result : F	"z = 110"
results	SPP:EA0D75DDCB0019C0
test	SPP:E394DD3AA1002C30
*test	
x	11

Method	Module	Line	Statement	Program	Library
main	Testpgm	25	19	Testpgm	Testest
_CXX_PEP	Testpgm		297	Testpgm	Testest
			297	Qcmdexc	Qsys
			422	Qcmd	Qsys

Breakpoint at line: 25.



iSeries System Debug Manager

Debug	System	User	Status
✓	Ascon2	Shalver	ready

```
<2/26/02 2:56 PM> At                'Ascon2' (9.5.1
<2/26/02 2:56 PM> Co                 'Ascon2.rchlan
```

- Supports **all ILE languages**: C, C++, RPG, Java, COBOL, CL
- Point and click **breakpoint** manipulation in source code
- Automatic **variable evaluation** with mouse and **local variable display**
- Program **call stack** and **thread** display

- Requires JDK1.3 and **tes.jar**, jt400.jar, and jhall.jar

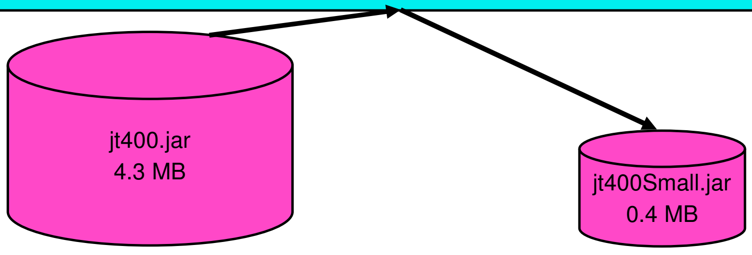
- Invoke with following: `java` **`utilities.DebugMgr`** *or* `java` **`utilities.Debug`** `-s system -u user`

JarMaker

Reduce jar file sizes

- The latest Toolbox jar file (jt400.jar) is approximately 4.3 MB.
- A given program typically only needs a small portion of the code (e.g. only CommandCall or only JDBC).
- ToolboxJarMaker "distills" jt400.jar down to only the code you need.
- JarMaker also works on jar files other than jt400.jar.

```
java utilities.ToolboxJarMaker -source jt400.jar -destination jt400Small.jar -component
                    CommandCall -ccsid 37 -noProxy -excludeSomeDependencies
```



What's new since V6R1

- **"JC1" LPP eliminated as of IBM i 7.1**
  - Integrated into SS1 (product ID 5770-SS1) Option 3
  - Same JAR files available in same IFS directories as in the past
- **New classes added**
  - Package `com.ibm.as400.`**`access`**
    - AS400JDBCArray, AS400JDBCArrayResultSet
    - ErrorCodeParameter
    - ObjectLockListEntry
    - UserObjectsOwnedList, UserObjectsOwnedListEntry
  - Package `com.ibm.as400.`**`security.auth`**
    - ProfileTokenProvider
    - DefaultProfileTokenProvider

What's new since V6R1 - continued

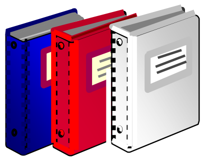
- **Significantly enhanced classes**
  - Package `com.ibm.as400.`**`access`**
    - Many of the JDBC classes
    - CommandCall and ProgramCall - new thread-safety behavior, new methods ★
    - IFSFile, IFSJavaFile - new methods ★
    - AS400 - new system properties, new methods
    - AS400ConnectionPool
    - Trace ★
    - DataArea
    - SpooledFile
- For complete details, refer to the IBM i Information Center
  - http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/topic/rzahh/page1.htm

Top 5 Good Things About the Toolbox

1. It’s free, no strings attached.
2. Fully supported by IBM Service.
   – User forum on Web is monitored daily by IBM developers.
3. Lets any Java app, anywhere on your LAN,
   – Access and exploit your IBM i resources.
4. Thoroughly documented on the Web.
5. In use by IBM and customers since V4R2 (1998).
   – Used under-the-covers in many other IBM products.

That's it!

References



Where can I get more information?

www.ibm.com/systems/i/software/toolbox

- Toolbox for Java: News, downloads, FAQs, articles, COMMON labs

sourceforge.net/projects/jt400

- JTOpen - open source, bug reporting, feature requests

www.ibm.com/systems/support/i/forums

- IBM i Technical Forums - including IBM Toolbox for Java/JTOpen Forum

IBM Toolbox for Java Programmers Guide

- Shipped with the IBM Toolbox for Java
- Contains overview, full API documentation (javadoc), and code examples
- Available in the IBM i Information Center
  - http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/topic/rzahh/page1.htm

Questions

Please fill in the following information on your evaluation sheet:

Session Title: **IBM Toolbox for Java: Advanced**

Session ID: **407180**

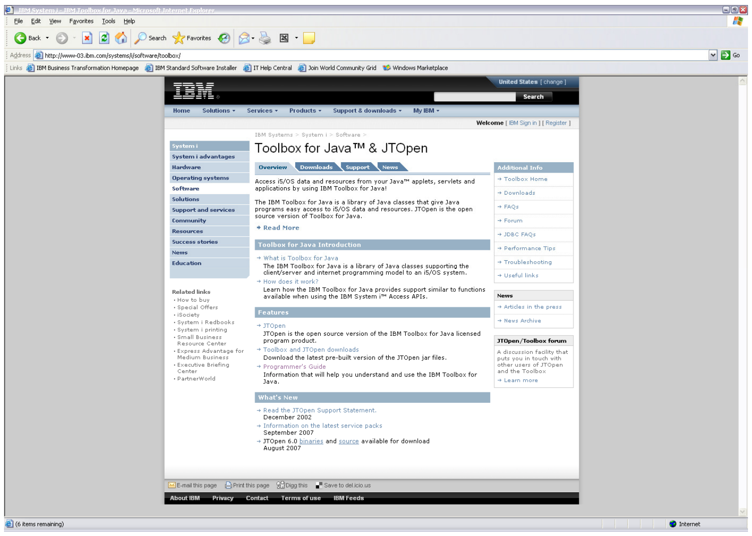
Agenda Key: **35SB**

Speaker: **Jeff Lee**

Please fill in evaluation sheets and place in the bag

Backup Slides

IBM Toolbox for Java home page



IBM Systems > System i > Software >

Toolbox for Java™ & JTOpen

Overview | Downloads | Support | News

Access i5/OS data and resources from your Java™ applets, servlets and applications by using IBM Toolbox for Java!

The IBM Toolbox for Java is a library of Java classes that give Java programs easy access to i5/OS data and resources. JTOpen is the open source version of Toolbox for Java.

→ Read More

Toolbox for Java Introduction

→ What is Toolbox for Java
The IBM Toolbox for Java is a library of Java classes supporting the client/server and internet programming model to an i5/OS system.

→ How does it work?
Learn how the IBM Toolbox for Java provides support similar to functions available when using the IBM System i™ Access APIs.

Features

→ JTOpen
JTOpen is the open source version of the IBM Toolbox for Java licensed program product.

→ Toolbox and JTOpen downloads
Download the latest pre-built version of the JTOpen jar files.

→ Programmer's Guide
Information that will help you understand and use the IBM Toolbox for Java.

What's New

→ Read the JTOpen Support Statement.
December 2002

→ Information on the latest service packs
September 2007

→ JTOpen 6.0 binaries and source available for download
August 2007

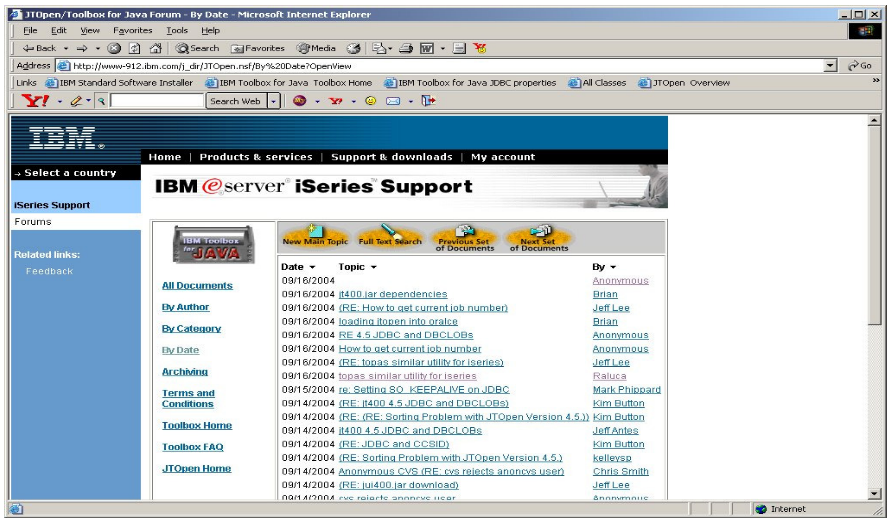
Additional Info
- Toolbox Home
- Downloads
- FAQs
- Forum
- JDBC FAQs
- Performance Tips
- Troubleshooting
- Useful links

News
- Articles in the press
- News Archive

JTOpen/Toolbox forum
A discussion facility that puts you in touch with other users of JTOpen and the Toolbox
→ Learn more

Related links
- How to buy
- Special Offers
- iSociety
- System i Redbooks
- System i printing
- Small Business Resource Center
- Express Advantage for Medium Business
- Executive Briefing Center
- PartnerWorld

IBM Toolbox for Java/JTOpen Forum



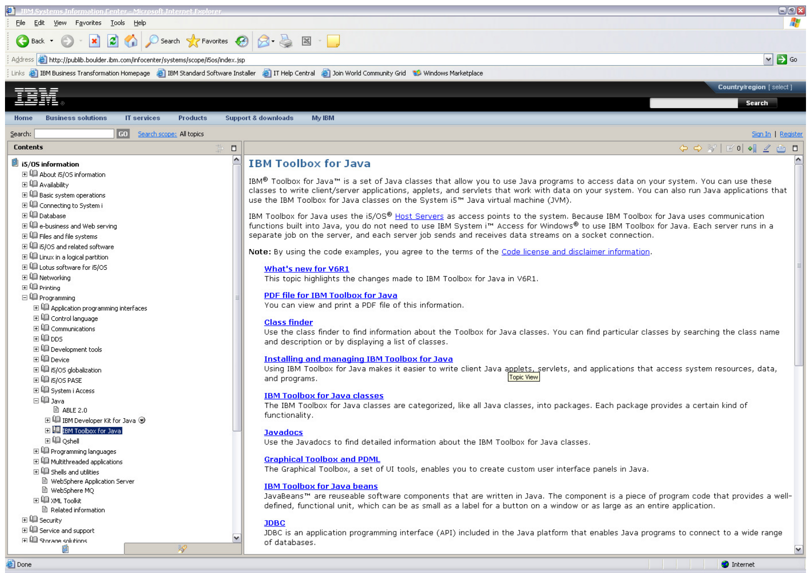
JTOpen/Toolbox for Java Forum - By Date - Microsoft Internet Explorer

Address: http://www-912.ibm.com/j_dir/JTOpen.nsf/By%20Date?OpenView

IBM eserver iSeries Support

Date	Topic	By
09/16/2004		Anonymous
09/16/2004	jt400.jar dependencies	Brian
09/16/2004	(RE: How to get current job number)	Jeff Lee
09/16/2004	loading jtopen into oralce	Brian
09/16/2004	RE 4.5 JDBC and DBCLOBs	Anonymous
09/16/2004	How to get current job number	Anonymous
09/16/2004	(RE: topas similar utility for iseries)	Jeff Lee
09/16/2004	topas similar utility for iseries	Raluca
09/15/2004	re: Setting SO_KEEPALIVE on JDBC	Mark Phippard
09/14/2004	(RE: jt400 4.5 JDBC and DBCLOBs)	Kim Button
09/14/2004	(RE: (RE: Sorting Problem with JTOpen Version 4.5.))	Kim Button
09/14/2004	jt400 4.5 JDBC and DBCLOBs	Jeff Antes
09/14/2004	(RE: JDBC and CCSID)	Kim Button
09/14/2004	(RE: Sorting Problem with JTOpen Version 4.5.)	kelleysp
09/14/2004	Anonymous CVS (RE: cvs rejects anoncvs user)	Chris Smith
09/14/2004	(RE: jui400.jar download)	Jeff Lee
09/14/2004	cvs rejects anoncvs user	Anonymous

Toolbox Programmer's Guide



Contents

i5/OS information
- About i5/OS information
- Availability
- Basic system operations
- Connecting to System i
- Database
- e-business and Web serving
- Files and file systems
- i5/OS and related software
- Linux in a logical partition
- Lotus software for i5/OS
- Networking
- Printing
- Programming
  - Application programming interfaces
  - Control language
  - Communications
  - DDS
  - Development tools
  - Device
  - i5/OS globalization
  - i5/OS PASE
  - System i Access
  - Java
    - ABLE 2.0
    - IBM Developer Kit for Java
    - IBM Toolbox for Java
    - Qshell
  - Programming languages
  - Multithreaded applications
  - Shells and utilities
  - WebSphere Application Server
  - WebSphere MQ
  - XML Toolkit
  - Related information
- Security
- Service and support

IBM Toolbox for Java

IBM® Toolbox for Java™ is a set of Java classes that allow you to use Java programs to access data on your system. You can use these classes to write client/server applications, applets, and servlets that work with data on your system. You can also run Java applications that use the IBM Toolbox for Java classes on the System i5™ Java virtual machine (JVM).

IBM Toolbox for Java uses the i5/OS® Host Servers as access points to the system. Because IBM Toolbox for Java uses communication functions built into Java, you do not need to use IBM System i™ Access for Windows® to use IBM Toolbox for Java. Each server runs in a separate job on the server, and each server job sends and receives data streams on a socket connection.

Note: By using the code examples, you agree to the terms of the Code license and disclaimer information.

What's new for V6R1
This topic highlights the changes made to IBM Toolbox for Java in V6R1.

PDF file for IBM Toolbox for Java
You can view and print a PDF file of this information.

Class finder
Use the class finder to find information about the Toolbox for Java classes. You can find particular classes by searching the class name and description or by displaying a list of classes.

Installing and managing IBM Toolbox for Java
Using IBM Toolbox for Java makes it easier to write client Java applets, servlets, and applications that access system resources, data, and programs.

IBM Toolbox for Java classes
The IBM Toolbox for Java classes are categorized, like all Java classes, into packages. Each package provides a certain kind of functionality.

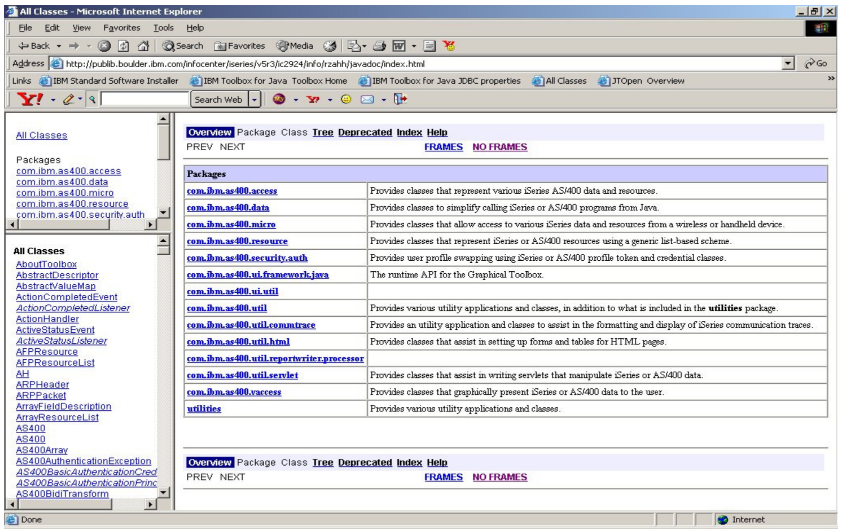
Javadocs
Use the Javadocs to find detailed information about the IBM Toolbox for Java classes.

Graphical Toolbox and PDML
The Graphical Toolbox, a set of UI tools, enables you to create custom user interface panels in Java.

IBM Toolbox for Java beans
JavaBeans™ are reuseable software components that are written in Java. The component is a piece of program code that provides a well-defined, functional unit, which can be as small as a label for a button on a window or as large as an entire application.

JDBC
JDBC is an application programming interface (API) included in the Java platform that enables Java programs to connect to a wide range of databases.

Javadoc



All Classes - Microsoft Internet Explorer

Address http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/rzahh/javadoc/index.html

All Classes

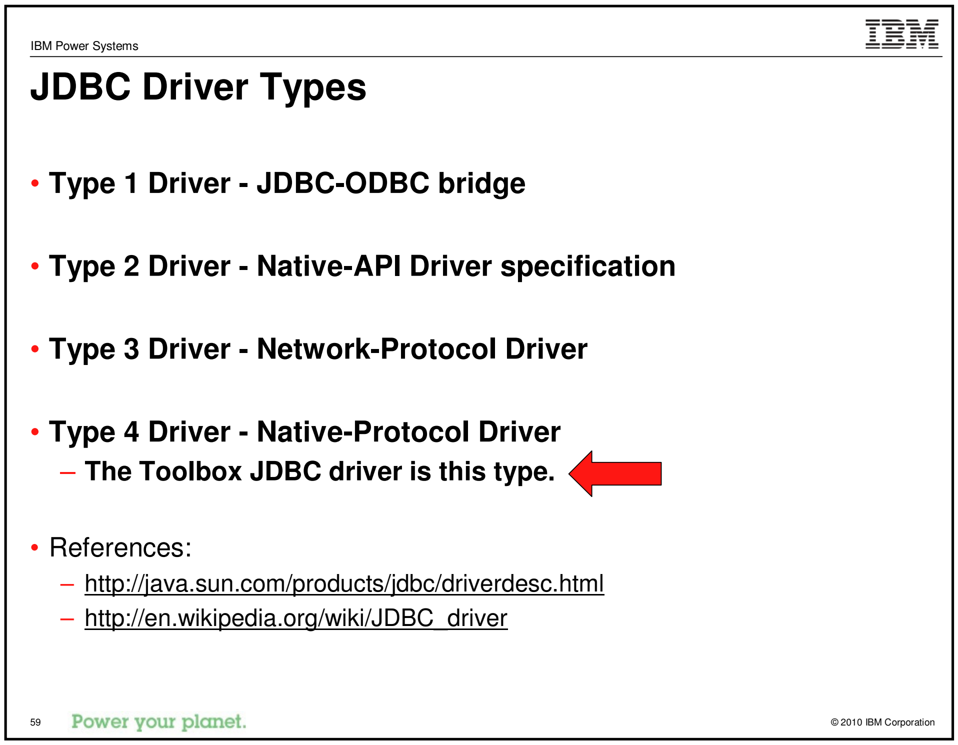
Packages
com.ibm.as400.access
com.ibm.as400.data
com.ibm.as400.micro
com.ibm.as400.resource
com.ibm.as400.security.auth

All Classes
AboutToolbox
AbstractDescriptor
AbstractValueMap
ActionCompletedEvent
ActionCompletedListener
ActionHandler
ActiveStatusEvent
ActiveStatusListener
AFPResource
AFPResourceList
AH
ARPHeader
ARPPacket
ArrayFieldDescription
ArrayResourceList
AS400
AS400
AS400Array
AS400AuthenticationException
AS400BasicAuthenticationCred
AS400BasicAuthenticationPrinc
AS400BidiTransform

Overview Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES

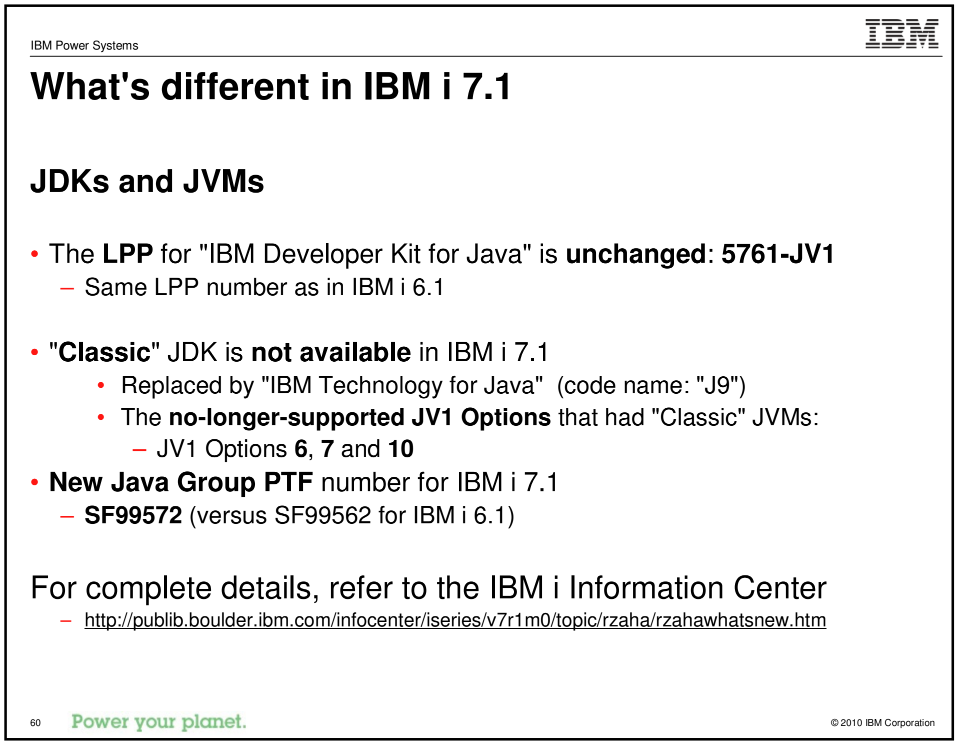
Packages	
com.ibm.as400.access	Provides classes that represent various iSeries AS/400 data and resources.
com.ibm.as400.data	Provides classes to simplify calling iSeries or AS/400 programs from Java.
com.ibm.as400.micro	Provides classes that allow access to various iSeries data and resources from a wireless or handheld device.
com.ibm.as400.resource	Provides classes that represent iSeries or AS/400 resources using a generic list-based scheme.
com.ibm.as400.security.auth	Provides user profile swapping using iSeries or AS/400 profile token and credential classes.
com.ibm.as400.ui.framework.java	The runtime API for the Graphical Toolbox.
com.ibm.as400.ui.util	
com.ibm.as400.util	Provides various utility applications and classes, in addition to what is included in the **utilities** package.
com.ibm.as400.util.commtrace	Provides an utility application and classes to assist in the formatting and display of iSeries communication traces.
com.ibm.as400.util.html	Provides classes that assist in setting up forms and tables for HTML pages.
com.ibm.as400.util.reportwriter.processor	
com.ibm.as400.util.servlet	Provides classes that assist in writing servlets that manipulate iSeries or AS/400 data.
com.ibm.as400.vaccess	Provides classes that graphically present iSeries or AS/400 data to the user.
utilities	Provides various utility applications and classes.

Overview Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES



JDBC Driver Types

- **Type 1 Driver - JDBC-ODBC bridge**
- **Type 2 Driver - Native-API Driver specification**
- **Type 3 Driver - Network-Protocol Driver**
- **Type 4 Driver - Native-Protocol Driver**
  - **The Toolbox JDBC driver is this type.**
- References:
  - http://java.sun.com/products/jdbc/driverdesc.html
  - http://en.wikipedia.org/wiki/JDBC_driver



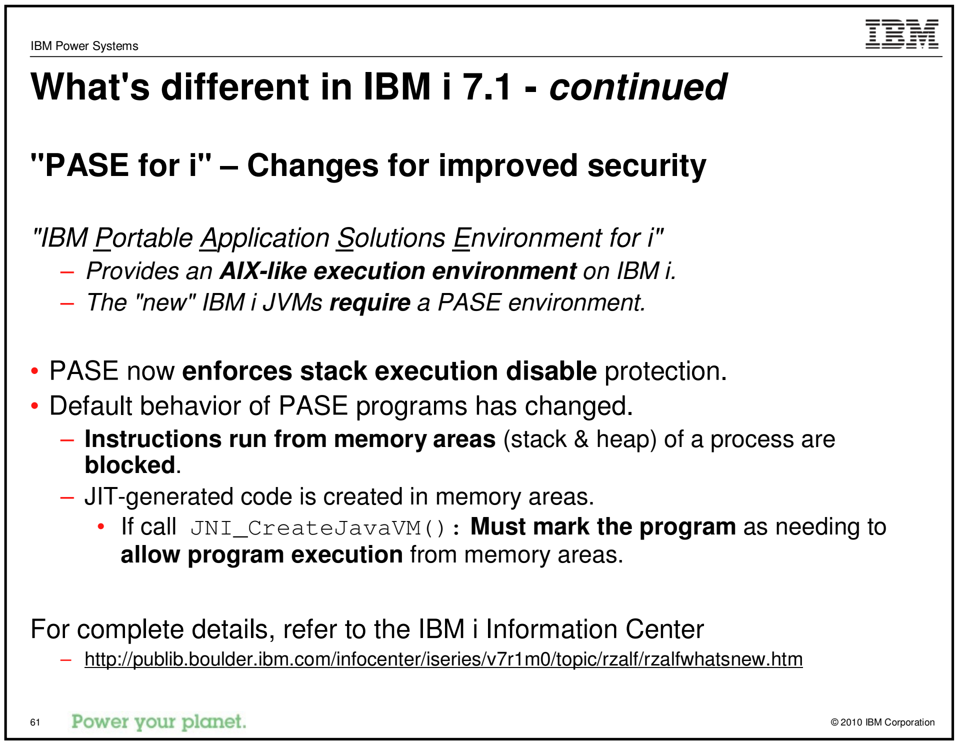
What's different in IBM i 7.1

JDKs and JVMs

- The **LPP** for "IBM Developer Kit for Java" is **unchanged**: **5761-JV1**
  - Same LPP number as in IBM i 6.1
- "**Classic**" JDK is **not available** in IBM i 7.1
  - Replaced by "IBM Technology for Java" (code name: "J9")
  - The **no-longer-supported JV1 Options** that had "Classic" JVMs:
    - JV1 Options **6**, **7** and **10**
- **New Java Group PTF** number for IBM i 7.1
  - **SF99572** (versus SF99562 for IBM i 6.1)

For complete details, refer to the IBM i Information Center

- http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/topic/rzaha/rzahawhatsnew.htm



IBM Power Systems

What's different in IBM i 7.1 - *continued*

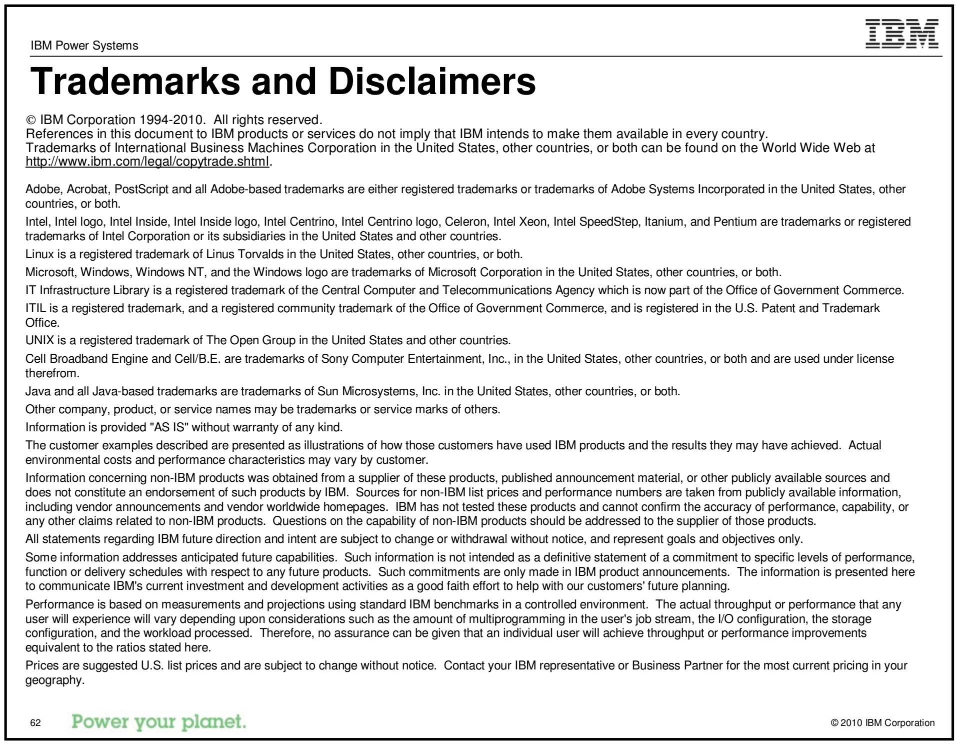
"PASE for i" – Changes for improved security

"IBM Portable Application Solutions Environment for i"
- *Provides an **AIX-like execution environment** on IBM i.*
- *The "new" IBM i JVMs **require** a PASE environment.*

- PASE now **enforces stack execution disable** protection.
- Default behavior of PASE programs has changed.
  - **Instructions run from memory areas** (stack & heap) of a process are **blocked**.
  - JIT-generated code is created in memory areas.
    - If call `JNI_CreateJavaVM()`**: Must mark the program** as needing to **allow program execution** from memory areas.

For complete details, refer to the IBM i Information Center
- http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/topic/rzalf/rzalfwhatsnew.htm

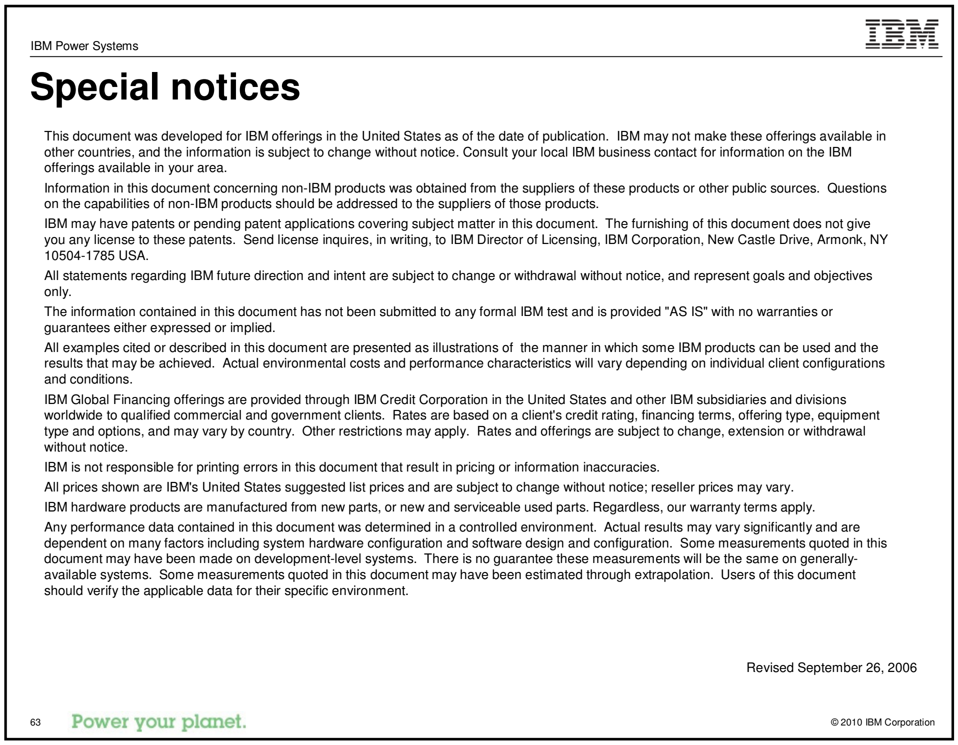
61 Power your planet. © 2010 IBM Corporation



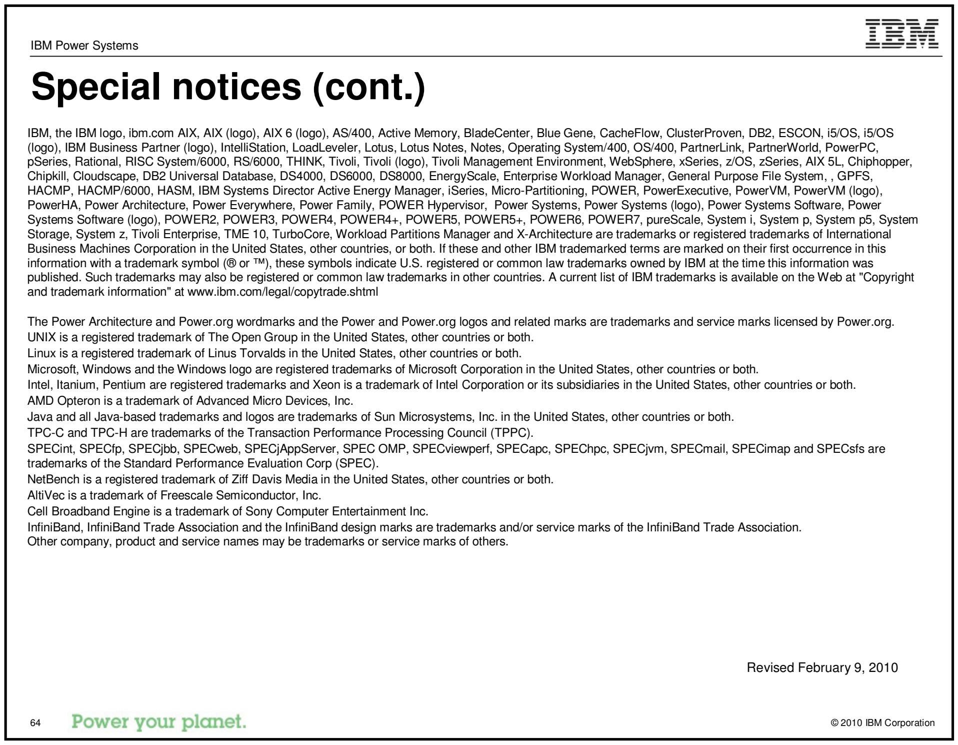
IBM Power Systems

Trademarks and Disclaimers

62 Power your planet. © 2010 IBM Corporation



Special notices



Special notices (cont.)