

Agenda Key: 52CC
Session Number: 407180

IBM Toolbox for Java™: Advanced

Jeff Lee



© Copyright IBM Corporation, 2009. All Rights Reserved.
This publication may refer to products that are not currently available in your country. IBM makes no commitment to make available any products referred to herein.

Java related sessions



Basic Java

Mon 9:30 - 22MD A Java Introduction to Object-Oriented Programming (OOP)

Mon 3:30 - 26MJ Debugging the New Java ***New Session***

Tues 8:00 - 31CD Java 101: Basic Syntax and Structure

Tues 11:00 - 33LA LAB: Introduction to Java ***LAB***

Web 9:30 - 42CD The Future of Java on IBM i

Java Toolbox

Tues 2:00 - 35CB Introducing the IBM Toolbox for Java

Wed 8:00 - 41LA LAB: IBM Toolbox for Java ***LAB***

Thur 9:30 - 52CC IBM Toolbox for Java: Advanced

Advanced Java related topics

Mon 11:00 - 23MH Introduction to XML Processing with Java

Tues 3:30 - 36MG Java Application Performance Analysis and Tuning on IBM i

Wed 2:00 - 45CD Using the JVM Tools Interface (JVMTI)

Thur 12:30 - 54CB Multi-Threaded Programming Using Java

Thur 2:00 - 55MH Java Stored Procedures and Java User-Defined Functions



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
- References

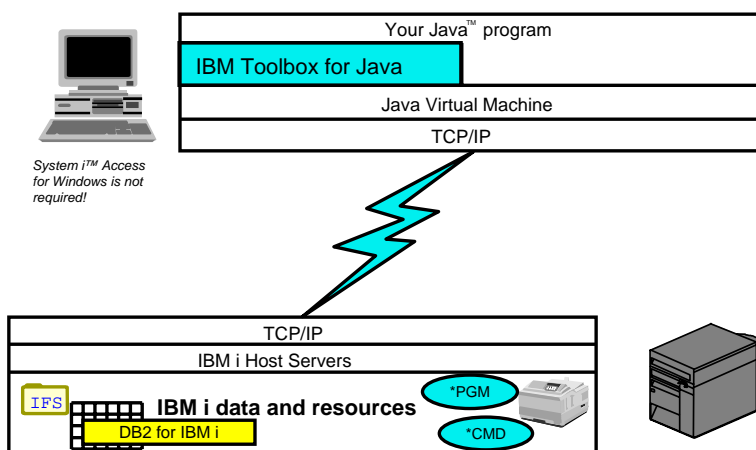
THE NEW POWER EQUATION

3

© 2009 IBM Corporation

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

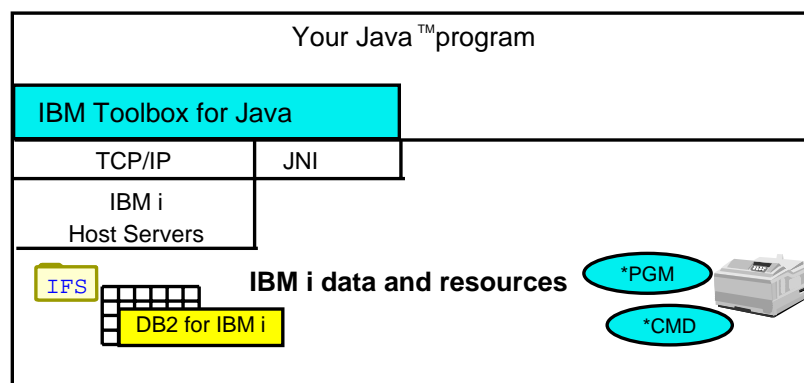


THE NEW POWER EQUATION

4

© 2009 IBM Corporation

- 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



THE NEW POWER EQUATION

5

© 2009 IBM Corporation

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
 - based on whether the command or program is threadsafe
 - see the setThreadSafe() method

THE NEW POWER EQUATION

6

© 2009 IBM Corporation

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 certificate	•	•
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

THE NEW POWER EQUATION

7

© 2009 IBM Corporation

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

THE NEW POWER EQUATION

8

© 2009 IBM Corporation

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

THE NEW POWER EQUATION

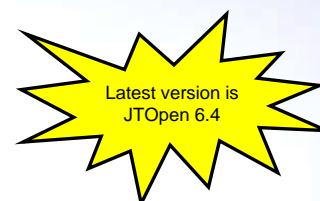
9

© 2009 IBM Corporation

JTOpen (Open Source)

All of the primary Toolbox packages are open source!
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



Two versions 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 NEW POWER EQUATION

10

© 2009 IBM Corporation

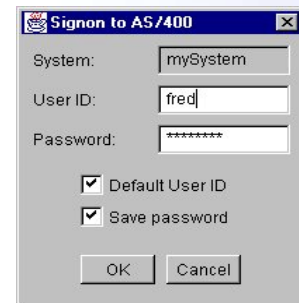
The AS400 object

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



Most Toolbox classes use the AS400 object

THE NEW POWER EQUATION

11

© 2009 IBM Corporation

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

- Use default constructor or *CURRENT
- ```
new AS400();
new AS400("localhost", "*CURRENT", "*CURRENT");
```

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
```

*THE NEW POWER EQUATION*

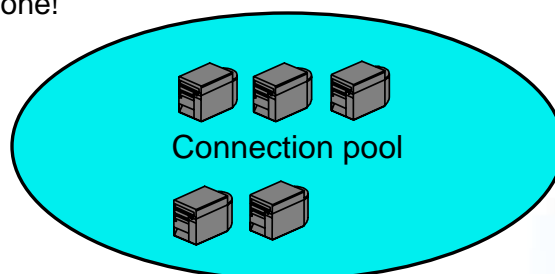
12

© 2009 IBM Corporation

# 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



*THE NEW POWER EQUATION*

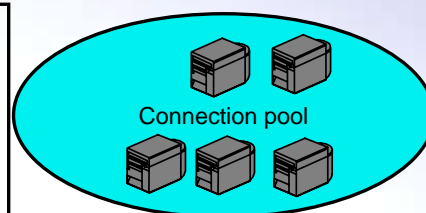
13

© 2009 IBM Corporation

## Set up the connection pool

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

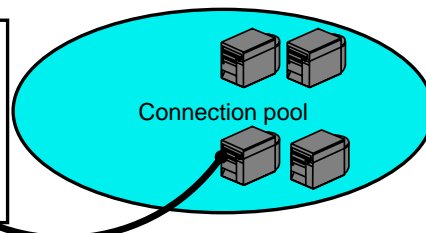
// Preconnect 5 connections to the AS400.COMMAND service.
pool.fill("myAS400", "myUserID", "myPassword", AS400.COMMAND, 5);
```



## Use a connection from the pool

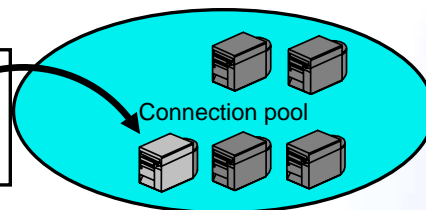
```
AS400 connection = pool.getConnection("myAS400", "myUserID", "myPassword",
AS400.COMMAND);

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



## Return it to the pool when done

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



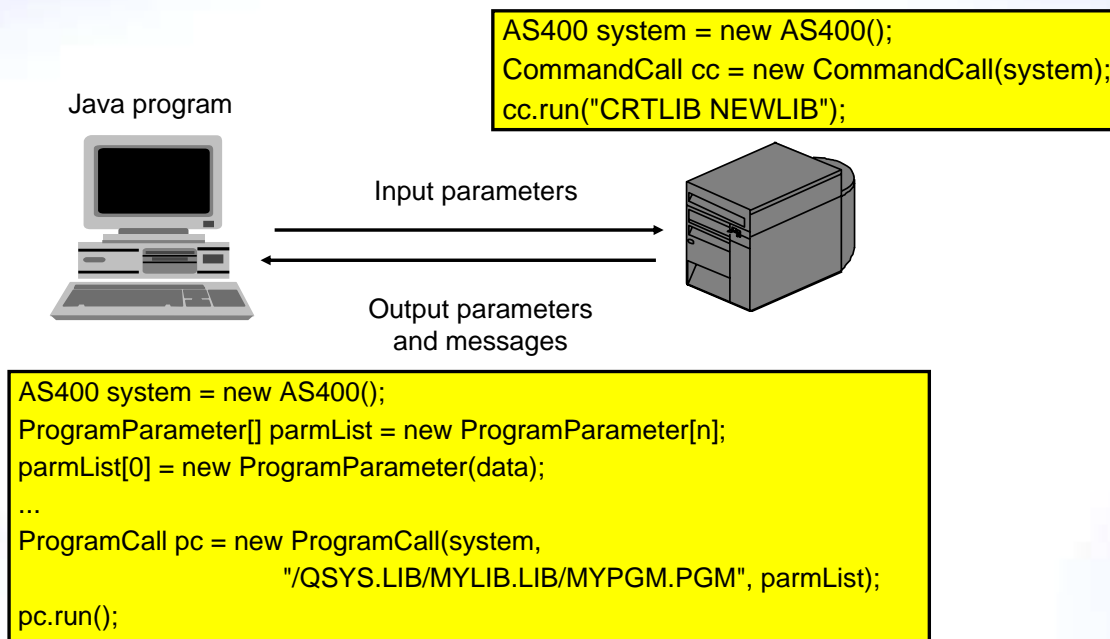
*THE NEW POWER EQUATION*

14

© 2009 IBM Corporation

# Command call and program call

*Make use of legacy code and system APIs*



*THE NEW POWER EQUATION*

15

© 2009 IBM Corporation

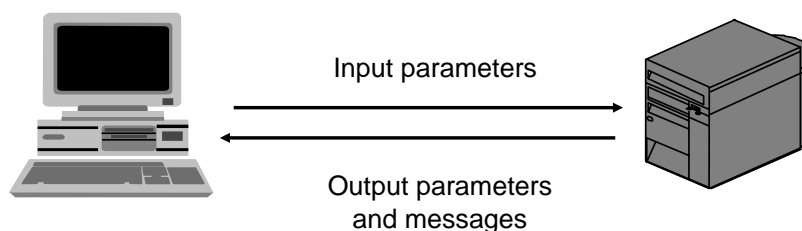
# Program Call Markup Language (PCML)

*Describe program calls using XML*

Parameter handling in traditional Toolbox ProgramCall can be tedious.

PCML:

- Simplifies data description and conversion
- Iterative development without recompile



*THE NEW POWER EQUATION*

16

© 2009 IBM Corporation



# 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 name="badSignonAttempts" type="byte" length="1" usage="output"/>
 <data name="status" type="int" length="4" usage="output"/>
 <data name="passwordChangeDate" type="char" length="10" usage="output"/>
 <data name="noPassword" 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");
```

*THE NEW POWER EQUATION*

17

© 2009 IBM Corporation

# 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();
```

*THE NEW POWER EQUATION*

18

© 2009 IBM Corporation

# Data Queues

## Store data entries in a queue for processing

- Good for message passing across multiple processes
- DataQueue or KeyedDataQueue
- 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)

*THE NEW POWER EQUATION*

19

© 2009 IBM Corporation

# 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();
```

*THE NEW POWER EQUATION*

20

© 2009 IBM Corporation

# 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*

*THE NEW POWER EQUATION*

21

© 2009 IBM Corporation

# 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

*THE NEW POWER EQUATION*

22

© 2009 IBM Corporation

# 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);
```

*THE NEW POWER EQUATION*

23

© 2009 IBM Corporation

# 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)

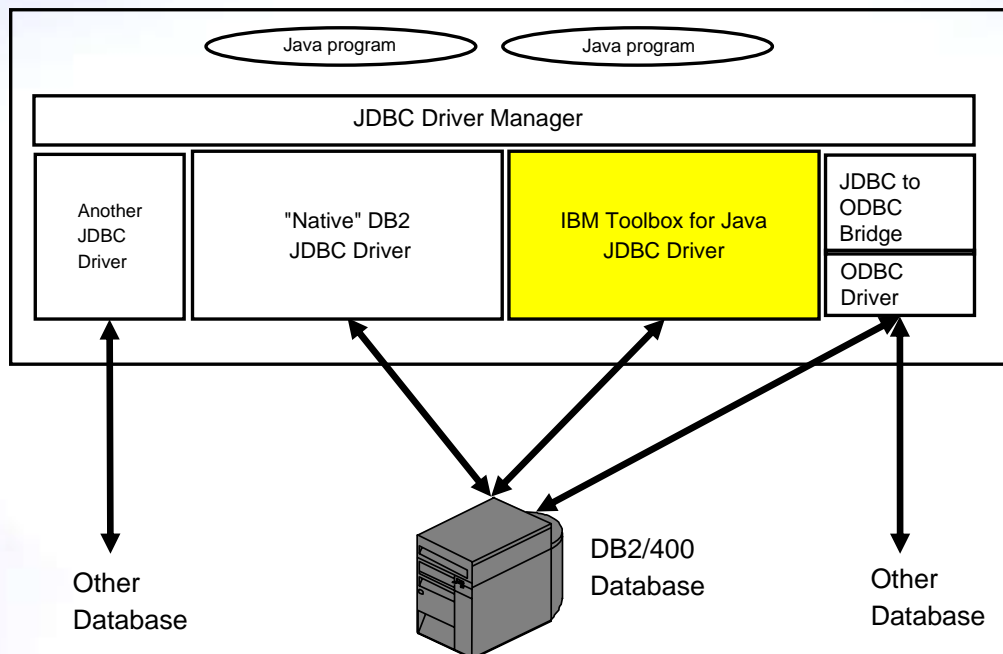
*THE NEW POWER EQUATION*

24

© 2009 IBM Corporation

# JDBC

*The Java standard for SQL database access*



*THE NEW POWER EQUATION*

25

© 2009 IBM Corporation

# 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());
```

*THE NEW POWER EQUATION*

26

© 2009 IBM Corporation

# 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);
```

*THE NEW POWER EQUATION*

27

© 2009 IBM Corporation

## 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

*THE NEW POWER EQUATION*

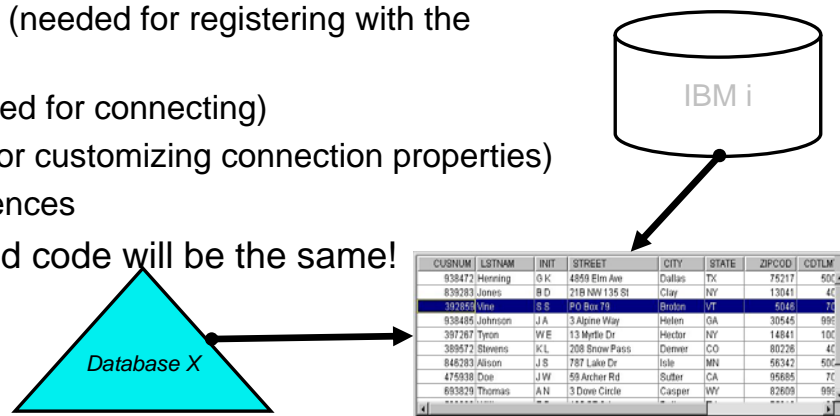
28

© 2009 IBM Corporation

# 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!



*THE NEW POWER EQUATION*

29

© 2009 IBM Corporation

# 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 PreparedStatement 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();

*THE NEW POWER EQUATION*

30

© 2009 IBM Corporation

# JDBC

## Statements (continued)

Use CallableStatements 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);
```

*THE NEW POWER EQUATION*

31

© 2009 IBM Corporation

# 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);`

*THE NEW POWER EQUATION*

32

© 2009 IBM Corporation



# 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

*THE NEW POWER EQUATION*

33

© 2009 IBM Corporation

# 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);
```

*THE NEW POWER EQUATION*

34

© 2009 IBM Corporation

# 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...*

*THE NEW POWER EQUATION*

35

© 2009 IBM Corporation

## Record-level database access

*Fast access to IBM i database files*

Provides access to physical and logical files:

- Access records sequentially, 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



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

*THE NEW POWER EQUATION*

36

© 2009 IBM Corporation

# 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();
}
```

*THE NEW POWER EQUATION*

37

© 2009 IBM Corporation

# 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.

*THE NEW POWER EQUATION*

38

© 2009 IBM Corporation

# 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]);
```

*THE NEW POWER EQUATION*

39

© 2009 IBM Corporation

## Web components create tables and forms

The screenshot shows two browser windows. The top window, titled 'Netscape', displays a table with the following data:

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

The bottom window, titled 'JDBC Example - Netscape', shows a form with the title 'JDBC Example'. It contains a label 'Enter SQL Statement:' and a text input field with the text '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]);
```

*THE NEW POWER EQUATION*

40

© 2009 IBM Corporation

# 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);
}
```

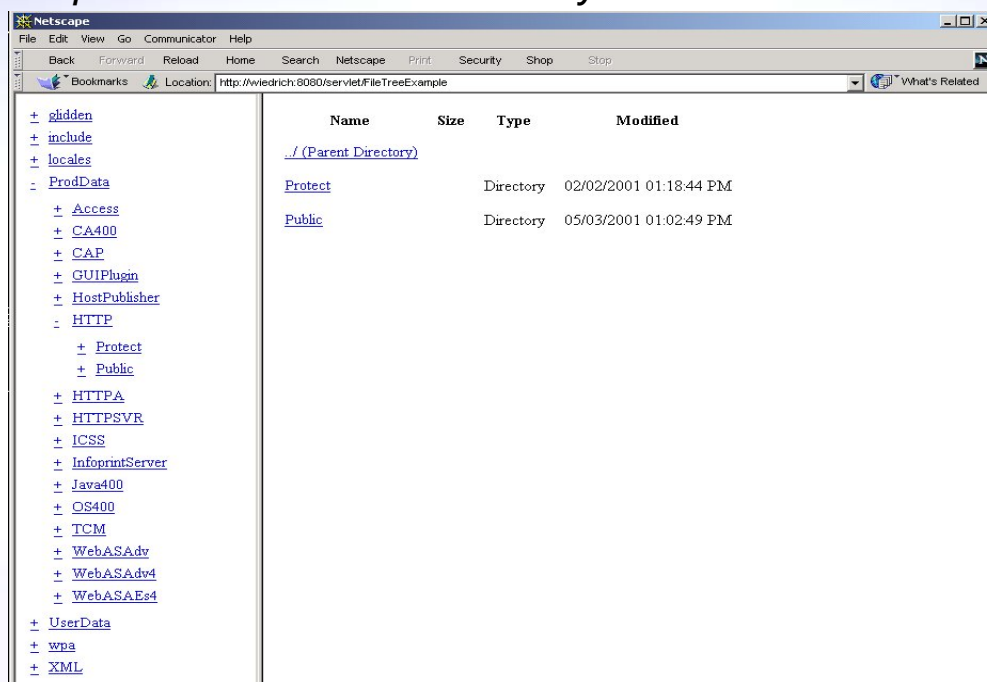
*THE NEW POWER EQUATION*

41

© 2009 IBM Corporation

# HTML and Servlet classes

*Web components create tree hierarchy*



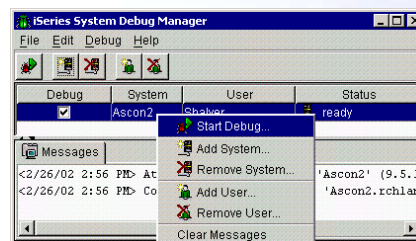
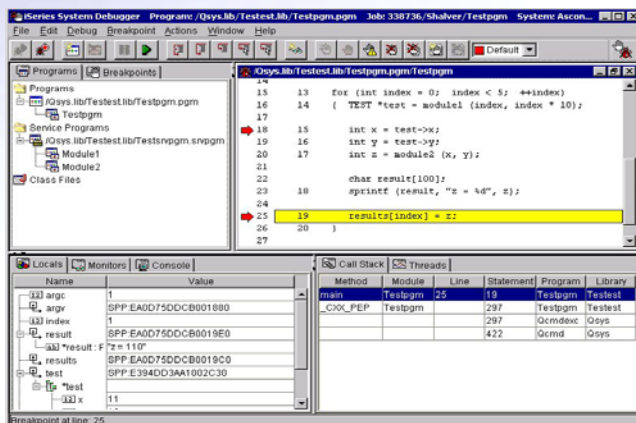
Name	Size	Type	Modified
<a href="#">../ (Parent Directory)</a>			
<a href="#">Protect</a>		Directory	02/02/2001 01:18:44 PM
<a href="#">Public</a>		Directory	05/03/2001 01:02:49 PM

*THE NEW POWER EQUATION*

42

© 2009 IBM Corporation

# System Debugger and Debug Manager



- 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 the following: `java utilities.DebugMgr` or `java utilities.Debug -s system -u user`

*THE NEW POWER EQUATION*

43

© 2009 IBM Corporation

## JarMaker

### Reduce jar file sizes

- The V6R1 Toolbox jar file (jt400.jar) is approximately 4.2 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
```



*THE NEW POWER EQUATION*

44

© 2009 IBM Corporation

## 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.

*THE NEW POWER EQUATION*

45

© 2009 IBM Corporation

## That's it!

*THE NEW POWER EQUATION*

46

© 2009 IBM Corporation

## References

*Where can I get more information?*

[www.ibm.com/systems/i/software/toolbox](http://www.ibm.com/systems/i/software/toolbox)

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

[sourceforge.net/projects/jt400](http://sourceforge.net/projects/jt400)

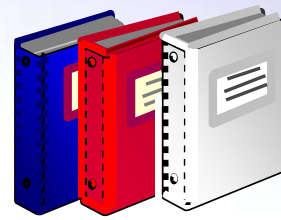
- JTOpen - open source, bug reporting, feature requests

[www.ibm.com/systems/support/i/forums](http://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
  - [publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzahh/page1.htm](http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzahh/page1.htm)



*THE NEW POWER EQUATION*

47

© 2009 IBM Corporation

## Questions

*THE NEW POWER EQUATION*

48

© 2009 IBM Corporation



# Java related sessions



## Basic Java

Mon 9:30 - 22MD A Java Introduction to Object-Oriented Programming (OOP)

Mon 3:30 - 26MJ Debugging the New Java \*\*\*New Session\*\*\*

Tues 8:00 - 31CD Java 101: Basic Syntax and Structure

Tues 11:00 - 33LA LAB: Introduction to Java \*\*\*LAB\*\*\*

Web 9:30 - 42CD The Future of Java on IBM i

## Java Toolbox

Tues 2:00 - 35CB Introducing the IBM Toolbox for Java

Wed 8:00 - 41LA LAB: IBM Toolbox for Java \*\*\*LAB\*\*\*

**Thur 9:30 - 52CC** IBM Toolbox for Java: Advanced



## Advanced Java related topics

Mon 11:00 - 23MH Introduction to XML Processing with Java

Tues 3:30 - 36MG Java Application Performance Analysis and Tuning on IBM i

Wed 2:00 - 45CD Using the JVM Tools Interface (JVMTI)

Thur 12:30 - 54CB Multi-Threaded Programming Using Java

Thur 2:00 - 55MH Java Stored Procedures and Java User-Defined Functions

*THE NEW POWER EQUATION*

49

© 2009 IBM Corporation

# Trademarks and Disclaimers



© IBM Corporation 1994-2009. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Trademarks of International Business Machines Corporation in the United States, other countries, or both can be found on the World Wide Web at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine and Cell/B.E. are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and are used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

The customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Prices are suggested U.S. list prices and are subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

*THE NEW POWER EQUATION*

50

© 2009 IBM Corporation