

SAP Java Connector Programming (JCo)

Thomas G. Schuessler
Founder and President, www.ARAsoft.de



THE BEST-RUN BUSINESSES RUN SAP



Learning Objectives

As a result of this workshop, you will be able to:

- Understand the BAPIs and other RFC-enabled Function Modules.
- Understand the SAP Java Connector (JCo) architecture.
- Write client programs with JCo.



BAPI Overview

SAP Java Connector (JCo) Overview

Building an sRFC Client Application with JCo

Exercise

Summary

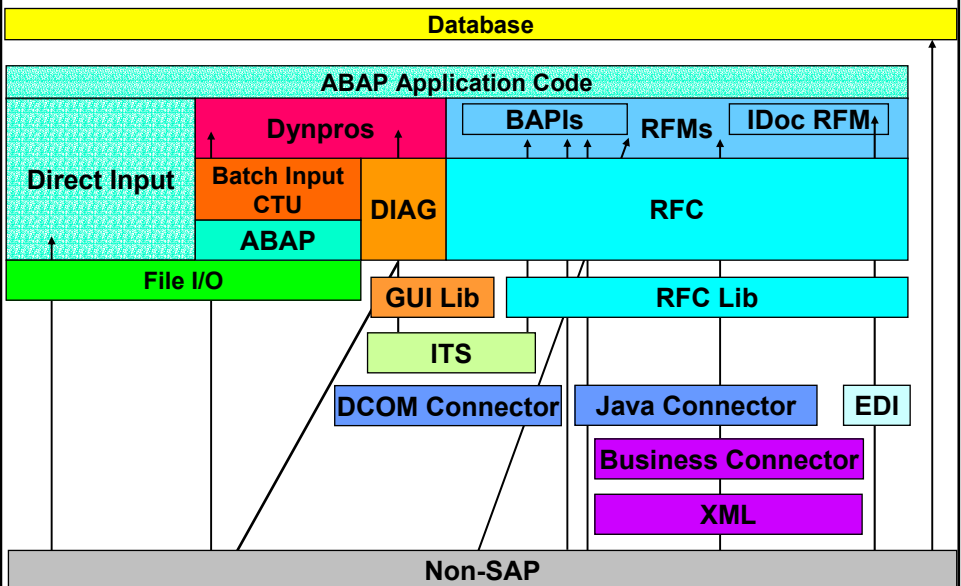
BAPI Overview

BAPIs and Other Interfaces

- BAPIs (Business Application Programming Interfaces) and IDocs are the two most important SAP interface technologies.
- BAPIs are normally called synchronously by an external client application.
- BAPIs are RFC-enabled Function Modules (RFMs) that are defined as methods in the Business Object Repository (BOR).
- IDocs are sent from and to SAP asynchronously.



A Simplified (!) Depiction of Some Inbound SAP Interfaces



RFM Object Model I

de.arasoft.sap.bapi.objectfactory.RfmParameter

```
+String getDescription()
+int getInternalLength()
+String getName()
+String getScalarCheckTable()
+String getScalarConversionExit()
+String getScalarDataTypeABAP()
+String getScalarDataTypeDDO()
+int getScalarDecimals()
+String getScalarDefaultDefinition()
+String getScalarDefaultValue()
+Field getScalarField()
+String getScalarFieldName()
+int getScalarLength()
+int getScalarOutputLength()
+Structure getStructure()
+String getStructureName()
+boolean isBasedOnDictionaryReference()
+boolean isExport()
+boolean isImport()
+boolean isMandatory()
+boolean isOptional()
+boolean isScalar()
+boolean isScalarFixedValuesListDefined()
+boolean isScalarHelpValuesSupported()
+boolean isScalarMixedCaseSupported()
+boolean isStructure()
+boolean isTable()
```

de.arasoft.sap.bapi.objectfactory.Rfm

```
+String getDescription()
+String getGroupName()
+String getName()
+RfmException getRfmException(String name)
+RfmExceptions getRfmExceptions()
+RfmParameter getRfmParameter(String name)
+RfmParameters getRfmParameters()
```

BOCollection

de.arasoft.sap.bapi.objectfactory.RfmParameters

```
+RfmParameters getExportParameters()
+RfmParameters getImportParameters()
+RfmParameters getMandatoryParameters()
+RfmParameters getOptionalParameters()
+RfmParameter getRfmParameter(int index)
+RfmParameter getRfmParameter(String name)
+RfmParameters getScalarParameters()
+RfmParameters getStructureParameters()
+RfmParameters getTableParameters()
```

BOCollection

de.arasoft.sap.bapi.objectfactory.RfmExceptions

```
+RfmException getRfmException(int index)
+RfmException getRfmException(String name)
```

de.arasoft.sap.bapi.objectfactory.RfmException

```
+String getDescription()
+String getName()
```



RFM Object Model II

de.arasoft.sap.bapi.objectfactory.Field

```
+String getCheckTable()
+String getConversionExit()
+String getDataElementName()
+String getDataElementABAP()
+String getDataElementDDO()
+int getDecimals()
+String getDescription()
+String getDomainName()
+FieldTexts getFieldTexts()
+int getIndex()
+int getInternalLength()
+int getLength()
+String getName()
+int getOffset()
+int getOutputLength()
+String getParameterId()
+String getReferenceField()
+String getReferenceTable()
+boolean isFixedValuesListDefined()
+boolean isHelpValuesSupported()
+boolean isKeyField()
+boolean isMixedCaseSupported()
+boolean isSearchHelpSupported()
+boolean isSignedNumber()
+boolean isWritingChangeDocuments()
```

de.arasoft.sap.bapi.objectfactory.FieldTexts

```
+String getColumnHeading()
+String getDescription()
+String getLabelLong()
+String getLabelMedium()
+String getLabelShort()
+String getText()
```



Importing Parameters

- Mandatory or optional
- Scalar or structure

Exporting Parameters

- Always optional
- Scalar or structure

Table Parameters

- Mandatory or optional
- ByRef in RFC

Exceptions

BAPI Explorer (BAPI)

Function Builder (SE37)

- Display
- Change
- Create
- Test

Data Dictionary (SE11)

SAP Java Connector (JCo) Overview



JCo Overview I

- Supports Remote Function Call (RFC) completely
 - ◆ Client and Server
 - ◆ sRFC, tRFC, qRFC
- BAPIs, other RFC-enabled Function Modules (RFMs), and IDocs are completely supported.
- Supports SAP 3.1H and higher
- JCo 2.x requires JDK 1.3 or later
 - ◆ JCo 1.x still supports JDK 1.1
- Supports desktop and (Web) Application Server applications
- High-performance, JNI-based implementation
- Download from <http://service.sap.com/connectors>
- Used by the SAP Business Connector, SAP Enterprise Portal, and in many customer and partner products and projects.



- **Multi-platform**
 - ◆ **Windows NT and 2000, Linux (Intel), zLinux (IBM S/390 and zSeries), Solaris, IBM AIX, HP-UX, OSF1 (Alpha), IBM OS/400**
- **Complete and correct code page handling**
- **Easy to install and deploy**

JCo 2.x Installation and Deployment (Windows)

Required files in WINNT\system32:

- **librfc32.dll** (at least 6.20)

Required files in Java classpath:

- **sapjco.jar**

Required JNI DLL (in the same directory as sapjco.jar)

- **sapjcorfc.dll**

Building an sRFC Client Application with JCo

Import Statements

One import statement for JCO

■ Sample code:

```
import com.sap.mw.jco.*;
```


An object of type JCO.Client represents a session with SAP.

Instantiated via a call to JCO.createClient(...) or reserved in a client pool.

Direct connection (application server) or load balancing group can be used.

JCO.Client createClient I

JCO.createClient(...) for a specific application server

`JCO.Client JCO.createClient(...)`

■ Sample code:

```
JCO.Client mConnection = null;
```

```
mConnection =
```

```
JCO.createClient("400",           // SAP client
                 "arasoft",        // userid
                 "*****",         // password
                 "EN",             // language
                 "host",           // application server host name
                 "00");            // SAP system number
```

■ Language can be null; system default language used.

JCO.createClient(...) with a properties file

`JCO.Client JCO.createClient(Properties properties)`

■ Sample code:

```
JCO.Client mConnection = null;  
OrderedProperties logonProperties =  
    OrderedProperties.load("/logon.properties");  
mConnection =  
    JCO.createClient(logonProperties);
```

■ logon.properties:

```
jco.client.client=400  
jco.client.user=arasoft  
jco.client.passwd=*****  
jco.client.ashost=host  
jco.client.sysnr=00  
jco.client.lang=EN
```

JCO.Client Methods

Connecting to SAP

```
void connect()
```

Disconnecting from SAP

```
void disconnect()
```

Executing an RFM

```
void execute(JCO.Function function)
```


Represents SAP metadata.

Created via a call to the constructor.

```
JCO.Repository (String name, JCO.Client client)
```

```
JCO.Repository (String name, String pool_name)
```

The first parameter is an arbitrary name.

■ **Sample code:**

```
JCO.Repository mRepository;
```

```
    mRepository = new JCO.Repository("TechEd", mConnection);
```

JCO.Repository Methods

Create a function template

```
IFunctionTemplate getFunctionTemplate(String function_name)
```

Returns null if the RFM does not exist.

A function template represents the metadata for an RFM, but has no data; a function object has parameter data. This is similar to the class/object relationship in Java.

Create a JCO.Function object.

`JCO.Function` `getFunction()`

■ Sample code:

```
JCO.Repository mRepository;  
mRepository = new JCO.Repository("TechEd", mConnection);  
public JCO.Function createFunction(String name) {  
    try {  
        return  
            mRepository.  
                getFunctionTemplate(name.toUpperCase()).getFunction();  
    }  
    catch (Exception ex) {}  
    return null;  
}
```

JCO.Function

Represents an RFM.

Contains all metadata required for using the function.

■ Sample code:

```
JCO.Function function =  
    createFunction("BAPI_COMPANYCODE_GETLIST");  
if (function != null) {  
    mConnection.execute(function);  
}
```


Accessing the parameters

```
JCO.ParameterList getExportParameterList()
```

```
JCO.ParameterList getImportParameterList()
```

```
JCO.ParameterList getTableParameterList()
```

- All three methods return null if no parameters of the specific type exist for the function.

Mapping Between ABAP And Java Data Types

ABAP

Java

b	1-byte integer	int	JCO.TYPE_INT1
s	2-byte integer	int	JCO.TYPE_INT2
l	4-byte integer	int	JCO.TYPE_INT
C	Character	String	JCO.TYPE_CHAR
N	Numerical Character	String	JCO.TYPE_NUM
P	Binary Coded Decimal	BigDecimal	JCO.TYPE_BCD
D	Date	Date	JCO.TYPE_DATE
T	Time	Date	JCO.TYPE_TIME
F	Float	double	JCO.TYPE_FLOAT
X	Raw data	byte[]	JCO.TYPE_BYTE
g	String (variable-length)	String	JCO.TYPE_STRING
y	Raw data (variable-length)	byte[]	JCO.TYPE_XSTRING

Accessing generic metadata

```
int getFieldCount()
String getName(int index)
boolean isExport(String name/int index)
boolean isImport(String name/int index)
boolean isOptional(String name/int index)
boolean isStructure(String name/int index)
boolean isTable(String name/int index)
```

Accessing metadata for scalars

```
int getLength(String name/int index)
int getType(String name/int index)
    ■ Cf. mapping table.
int getDecimals(String name/int index)
String getDefault(String name/int index)
```

Accessing scalars (read)

```
BigDecimal getBigDecimal(String name/int index)
byte[] getByteArray(String name/int index)
Date getDate(String name/int index)
double getDouble(String name/int index)
int getInt(String name/int index)
String getString(String name/int index)

Object getValue(String name/int index)
```


Accessing scalars (read; convenience methods)

```
BigInteger getBigInteger(String name/int index)
InputStream getBinaryStream(String name/int index)
char getChar(String name/int index)
Reader getCharacterStream(String name/int index)
short getShort(String name/int index)
long getLong(String name/int index)
Date getTime(String name/int index)
```

Accessing scalars (write)

```
void setValue(..., String name/int index)
```

Supported data types

- byte[]
- char
- int
- long
- double
- short
- java.lang.Object
- java.lang.String

Sample code:

```
function.getImportParameterList().setValue("0001",
"COMPANYCODEID");
```


Accessing structures/tables

```
JCO.Structure getStructure(String name/int index)
```

```
JCO.Table getTable(String name/int index)
```

■ Sample code:

```
JCO.Structure returnStructure =  
    function.getExportParameterList()  
        .getStructure("RETURN");
```

JCO.Structure I

Accessing fields in the structure

```
BigDecimal getBigDecimal(String name/int index)
```

```
byte[] getByteArray(String name/int index)
```

```
Date getDate(String name/int index)
```

```
double getDouble(String name/int index)
```

```
int getInt(String name/int index)
```

```
String getString(String name/int index)
```

```
Object getValue(String name/int index)
```

```
void setValue(..., String name/int index)
```


Accessing structure metadata

```
int getFieldCount()  
  
public final boolean hasField(String field_name)  
  
String getName()
```

- Name of the SAP Data Dictionary structure

Accessing Fields In A Structure

Sample code:

```
JCO.Structure returnStructure =  
  
    function.getExportParameterList().getStructure("RETURN");  
  
if (! (returnStructure.getString("TYPE").equals("") ||  
        returnStructure.getString("TYPE").equals("S"))) {  
  
    System.out.println(returnStructure.getString("MESSAGE"));  
  
}
```


JCO.Table I

Based on the recordset paradigm.

Supports all methods found in JCO.Structure.

Navigating a table

```
int getNumRows()
```

```
void setRow(int pos)
```

```
int getRow()
```

- Current position

```
void firstRow()
```

```
void lastRow()
```

```
boolean nextRow()
```

- False if at end of table

```
boolean previousRow()
```

- False if at beginning of table



JCO.Table II

Manipulating a table

```
void appendRow()
```

- Current row is the new row.

```
void appendRow(int num_rows)
```

- Current row is the first new row.

```
void deleteAllRows()
```

```
void deleteRow()
```

- Deletes the current row.

- ◆ if (current_row < last_row) current_row = current_row;
else current_row--;

```
void deleteRow(int pos)
```

- ◆ if (current_row < last_row) current_row = current_row;
else current_row--;

```
void insertRow(int pos)
```

- Current row is the new row.



■ Sample code I:

```
JCO.Table codes =  
    function.getTableParameterList().getTable("COMPANYCODE_LIST");  
for (int i = 0; i < codes.getNumRows(); i++) {  
    codes.setRow(i);  
    System.out.println(codes.getString("COMP_CODE") + '\t' +  
        codes.getString("COMP_NAME"));  
}
```

■ Sample code II:

```
JCO.Table codes =  
    function.getTableParameterList().getTable("COMPANYCODE_LIST");  
if (codes.getNumRows() > 0) {  
    do {  
        System.out.println(codes.getString("COMP_CODE") + '\t' +  
            codes.getString("COMP_NAME"));  
    } while (codes.nextRow());  
}
```

Adding A New Table Row

■ Sample code:

```
JCO.Table codes =  
    function.getTableParameterList()  
        .getTable("COMPANYCODE_LIST");  
codes.appendRow();  
codes.setValue("XXXX", "COMP_CODE");  
codes.setValue("Does not exist", "COMP_NAME");
```


Sample code:

```
table.appendRows(10);  
  
do {  
    table.setValue("Value1","FIELD1");  
  
    // ...  
} while (table.nextRow());
```

- This is much faster than individual `appendRow()` calls.

Exception Handling

JCO throws three types of exceptions:

- JCO.Exception
- JCO.ConversionException
- JCO.AbapException

Generic exception

```
int getGroup()
```

■ e.g.

```
JCO_ERROR_COMMUNICATION
```

```
JCO_ERROR_LOGON_FAILURE
```

```
JCO_ERROR_SYSTEM_FAILURE
```

```
String getKey()
```

```
String getMessage()
```

JCO.ConversionException

Extends JCO.Exception.

Thrown if a type conversion fails.

Extends JCO.Exception.

Thrown if a BAPI/RFM throws an ABAP exception.

String getKey()

- Contains the ABAP Exception code.

Exception Handling Sample Code

```
JCO.Function function =
    this.createFunction("DDIF_FIELDINFO_GET");
try {
    function.getImportParameterList()
        .setValue("MARA", "TABNAME");
    mConnection.execute(function);
}
catch (JCO.AbapException ex) {
    if (ex.getKey().equalsIgnoreCase("NOT_FOUND")) {
        System.out.println
            ("Dictionary structure/table not found.");
        System.exit(1);
    }
    else {
        System.out.println(ex.getMessage());
        System.exit(1);
    }
}
catch (JCO.Exception ex) {
    // Handle the exception
}
catch (Exception ex) {
    // Handle the exception
}
```


Exercise

Summary

Sometimes a concept is baffling not because it is profound but because it is wrong.

Edward O. Wilson: Consilience, p. 277

Without detailed understanding, confidence cannot be attained.

Richard P. Feynman: "What Do You Care What Other People Think?", p. 227

one of the best times to act confident is when you are totally in the dark

Robert Asprin: M.Y.T.H. Inc. in Action, p. 98

It's not gambling to play against someone who's no good. It's common sense.

Terry Pratchett: Witches Abroad, p. 99

JCo workshops and custom development.

Mailing list for JCo news (Tutorial etc.)

ARAsoft JCo Extension Library
(Evaluation copy on request).



<http://www.arasoft.de>

+49-6222-91171 (phone)

+49-6222-91176 (fax)

+49-6227-748727 (SAP)

tgs@arasoft.de

thomas.schuessler@sap.com



Further Information

→ SAPnet:

service.sap.com/connectors

SAP Customer Services Network: www.sap.com/services/

→ Consulting Contact

Roy Wood, VP SAP NetWeaver Consulting Practice (r.wood@sap.com)

→ Related SAP Education Training Opportunities

<http://www.sap.com/usa/education/>

BIT526: JCo Programming

→ Related Workshops/Lectures at SAP TechEd 2003

OSI301 **Advanced Java Connector (JCo) Programming**, Lecture

OSI252 **The SAP .NET Connector: Architecture and Functionality**, Hands-on



Q&A



Questionnaire

Workshop Name:

Form:

Stoff:

Teilnehmer

☐ Kunde

☐ Interessent

☐ Berater

☐ _____

Abteilung

☐ Geschäftsleitung

☐ IT

☐ Produktion

☐ Produktion/Logistik

☐ Einkauf

☐ Vertrieb

Please complete your session evaluation and
drop it in the box on your way out.

Thank You !

The SAP TechEd '03 Basel Team

■ Integration and Certification Centers – worldwide -

ICC Walldorf, Germany
Phone +49 6227 – 767600
icc@sap.com

ICC Palo Alto, California US
Phone +1 650 - 849 2661
icc-americas@sap.com

ICC Bangalore, India
Phone +91 80 8418155 - 300
icc-apa@sap.com

Since 1995
about **800 vendors**
have successfully certified
more than
1,400 interface products!

<http://www.sap.com/partner/software/directory>

ICC: Integration & Certification Center



■ About SAP Integration and Certification Centers

Mission

- Leveraging the variety of integration opportunities SAP NetWeaver and mySAP Business Suite offer.
- Supporting third-party software vendors before, during, and after intergration certifications to achieve high quality integration.
- Facilitate interactions between customers & third-party vendors to achieve higher customer satisfaction.

Contact us:

- Web: <http://www.sap.com/icc>
- Email: icc-americas@sap.com
- Address:

ICC Palo Alto
SAP Labs, LLC.
3475 Deer Creek Road
Palo Alto, CA 94304

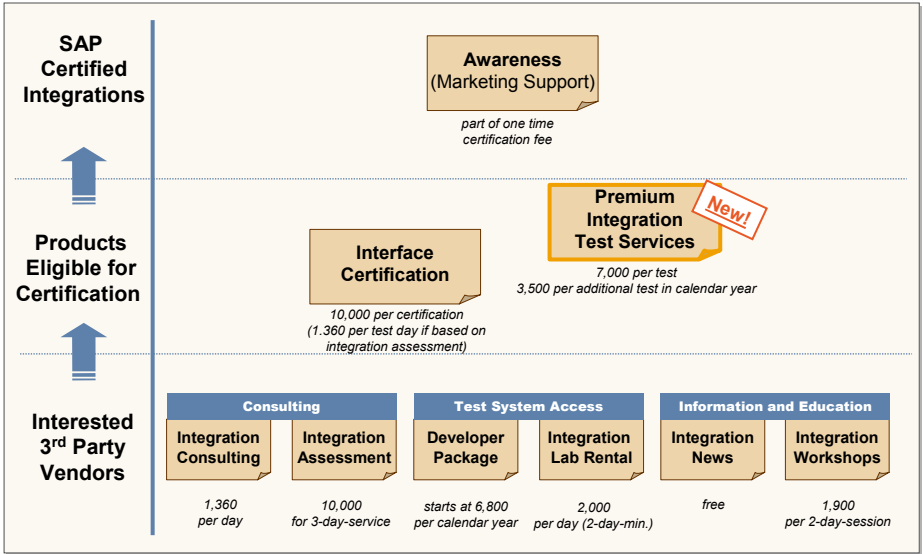
Since 1995 about **800 vendors**
have received more than
1300 certifications
on interface products.

http://www.sap.com/partners/software/directory/directory_nav.asp



Integration Services for Third-Party Vendors

Overview including current ICC fees* charged to third-party vendors



* Fees are subject to change at ICCs own discretion. Currency for Euro members is charged in Euro, all other countries are charged in US Dollars. Prices may vary depending on regional differences.

© SAP AG 2003, TechEd Basel, OSI251, Thomas G. Schuessler, www.arasoft.de / 53

THE BEST-RUN BUSINESSES RUN SAP



SAP Developer Package (DPA): Highlights

Target Group:	Any 3 rd party vendor with a suitable test scenario => does not have to be a partner!
General use:	Integration testing (no demo, no training etc.) => test & demo license still attractive to partners
Access:	Remote access (ISDN, VPN, ISP)* => no software "shipment" besides SAP connectors
Maintenance:	Software & hardware hosted through ICCs => SAP maintenance support for one location only
Duration:	annual basis

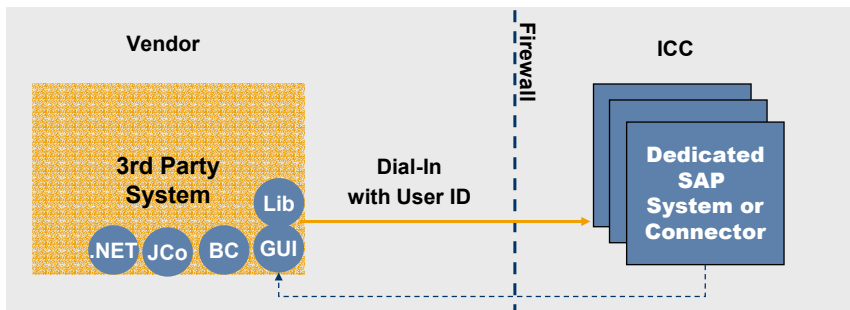
* for testing via direct access see SAP Integration Lab Rental

© SAP AG 2003, TechEd Basel, OSI251, Thomas G. Schuessler, www.arasoft.de / 54

THE BEST-RUN BUSINESSES RUN SAP



Integration Testing via Developer Package



SAP Services available:

IDES Systems	SAP R/3 Releases 3.1I / 4.0B / 4.5B / 4.6C (also Double-Byte Character), (max. 3 in basic kit) SAP R/3 Enterprise 4.7 (new) ; SAP Business Intelligence 3.0A (new) , SAP CRM 3.0 (new) , SAP Enterprise Buyer Professional 3.0 (new) , SAP Web Application Server 6.20 (new) , SAP Retail 4.6C (upon request)
Connectors	SAP Business Connector 4.6 , SAP Java Connector 2.0 , SAP .NET Conn.
Other	S-User for Service Marketplace (SAP Notes, etc.) , CD-ROM kit (SAP Gui, documentation, libraries etc.) , at extra charge OSS write access

Copyright 2003 SAP AG. All Rights Reserved

- No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
- Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA and/or other countries.
- ORACLE® is a registered trademark of ORACLE Corporation.
- UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.
- Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® is a registered trademark of Sun Microsystems, Inc.
- JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies.