IBM Software Group

Solving mainframe Data-centric customer problems using WS II Classic Federation

Raj Datta
WebSphere IIS System z Solutions Architect
IBM Software Group

Agenda

- Mainframe data access from web app
  - The basics – WebSphere II Classic Federation
  - Real world problem and solution
- ETL of Mainframe data
- SOA for Mainframe data

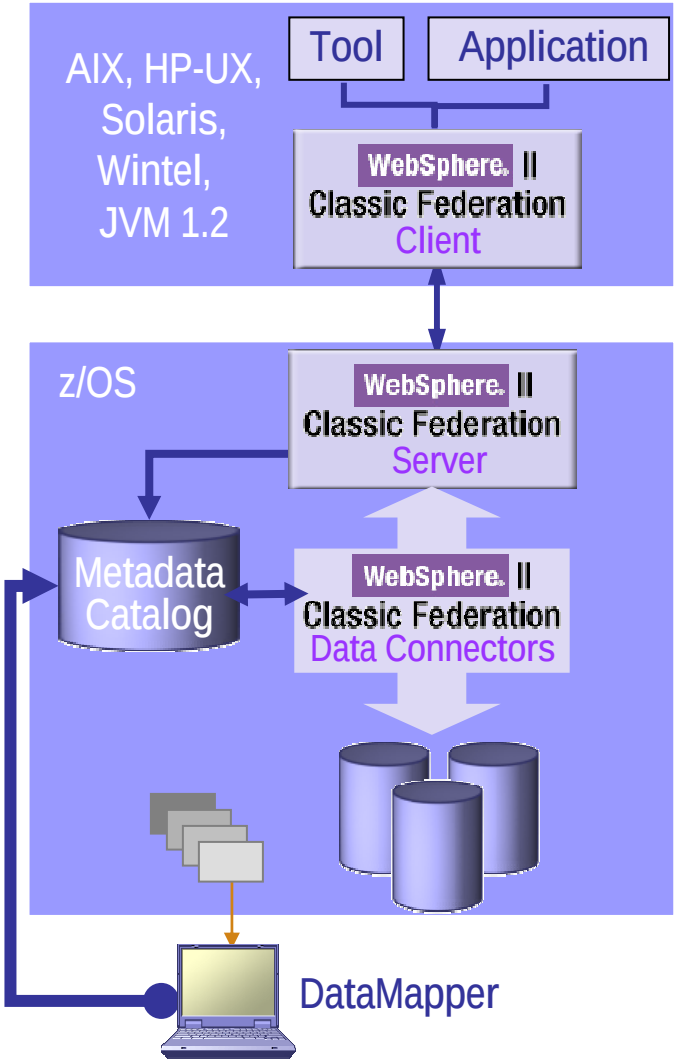
SQL Federation for the Mainframe

Integrate Mainframe Data Assets

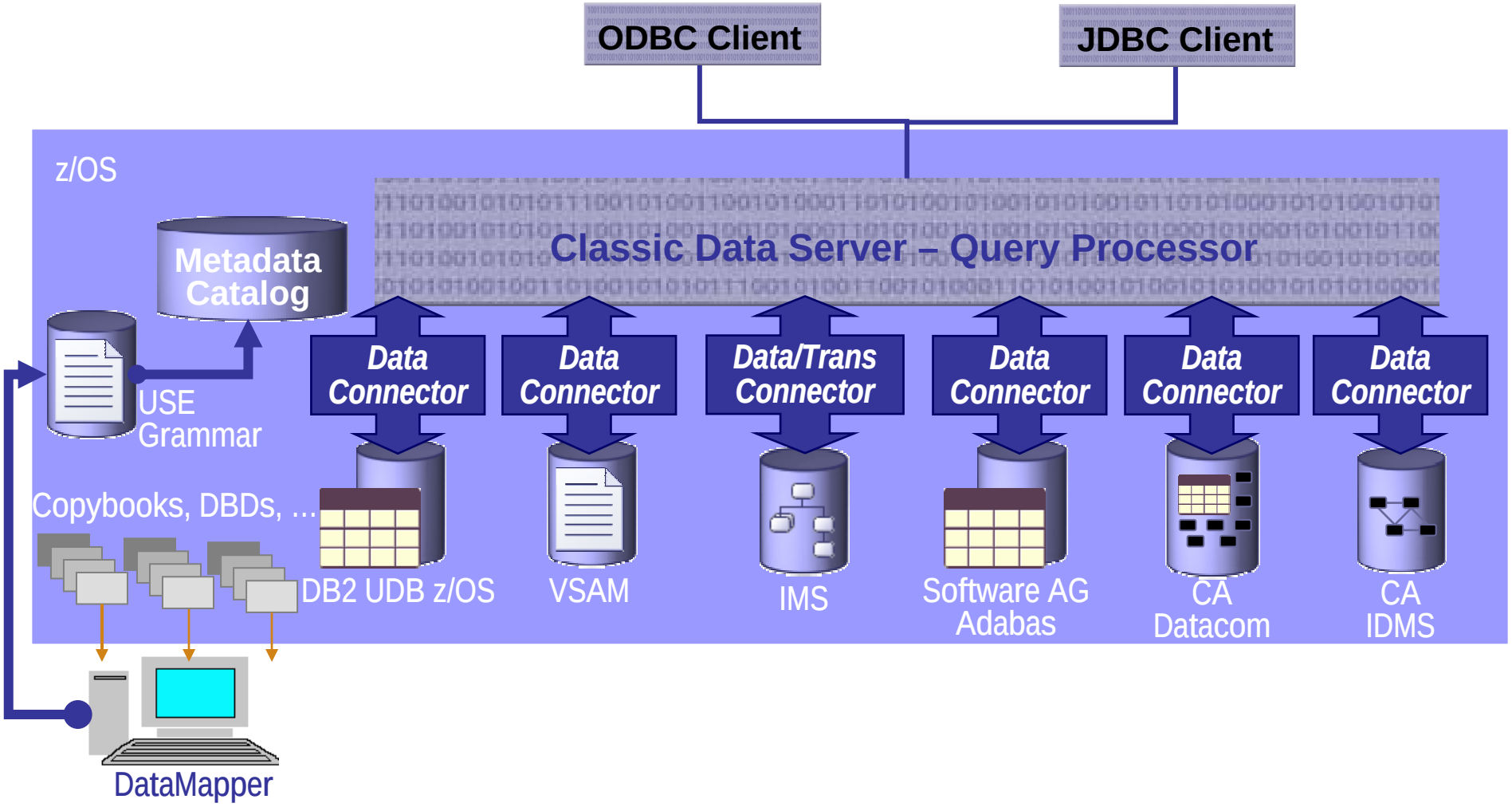
- SQL-based read and write access to mainframe data sources
  - Standard ODBC and JDBC
- Multi-threaded with native drivers for scalable performance
- Metadata-driven for easy configuration and maintenance
  - No mainframe programming required
  - Fast installation & configuration
  - Easy maintenance
- Works with existing
  - Mainframe infrastructure
  - Application infrastructure
  - Tools infrastructure

WebSphere II Classic Federation Implementation

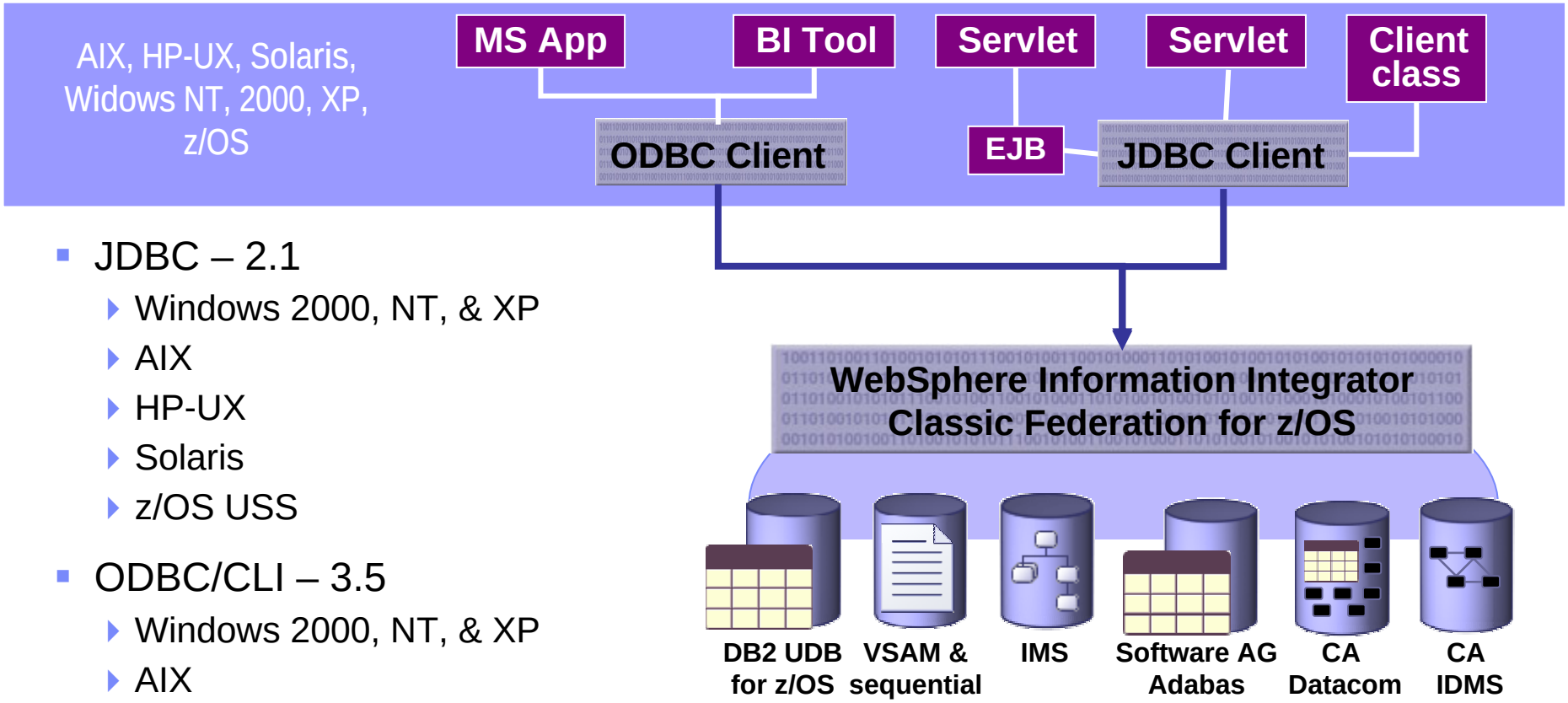
- Create relational description of mainframe data sources by mapping the physical data definitions to logical tables and views
- Mainframe Server and components act as a relational database engine
- JDBC and/or ODBC drivers provide standardized interface for tools and applications



Classic Federation Architecture – A Relational Data Store

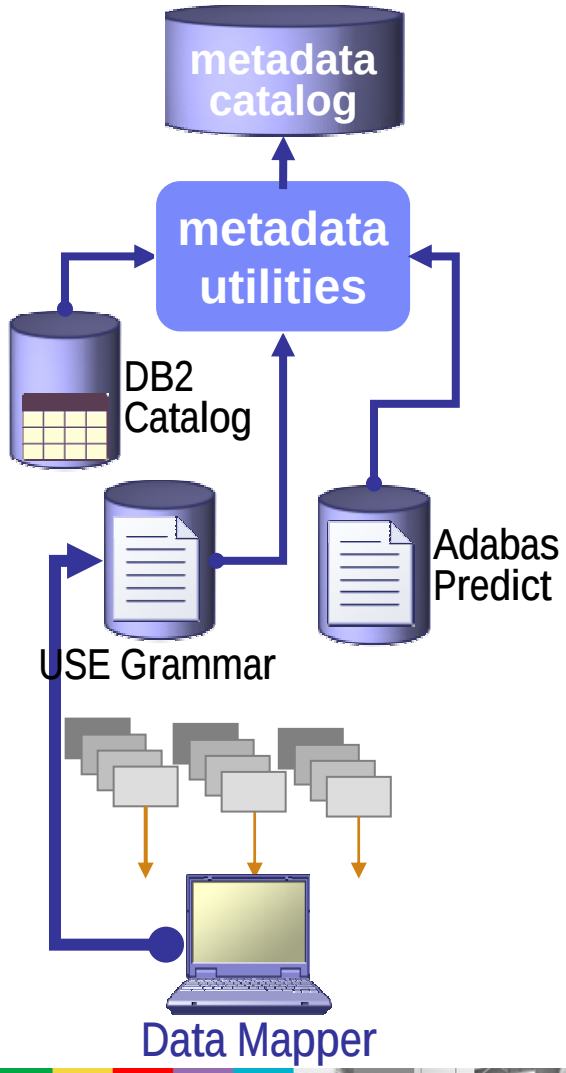


Classic Federation – Standard Clients



- JDBC – 2.1
  - Windows 2000, NT, & XP
  - AIX
  - HP-UX
  - Solaris
  - z/OS USS
- ODBC/CLI – 3.5
  - Windows 2000, NT, & XP
  - AIX
  - HP-UX
  - Solaris

Metadata Management



- Metadata defines business-oriented relational mappings
  - Import existing copybooks, IDMS schemas, IMS DBDs, etc.
  - Generate logical relational reference table definitions
  - GUI to customize logical tables to business requirements
- Simulated RDBMS catalog and more
  - RDBMS-like catalog support: systables, syscolumns, etc
  - Query-able tables for non-relational metadata
- Some metadata-driven features
  - Automatic translation of legacy data types
  - Handles legacy constructs like recurring data and redefines
  - Complex tables can span segments, records, etc.
  - Metadata-driven filtering using WHERE clauses
  - Enhances security via schema mapping, views, & DB2-like security
- Metadata Utilities
  - Create and update metadata catalog entries
  - Verify metadata against physical (e.g. VSAM index checks)
- DataMapper
  - Metadata customization and visual administration

IBM Software Group

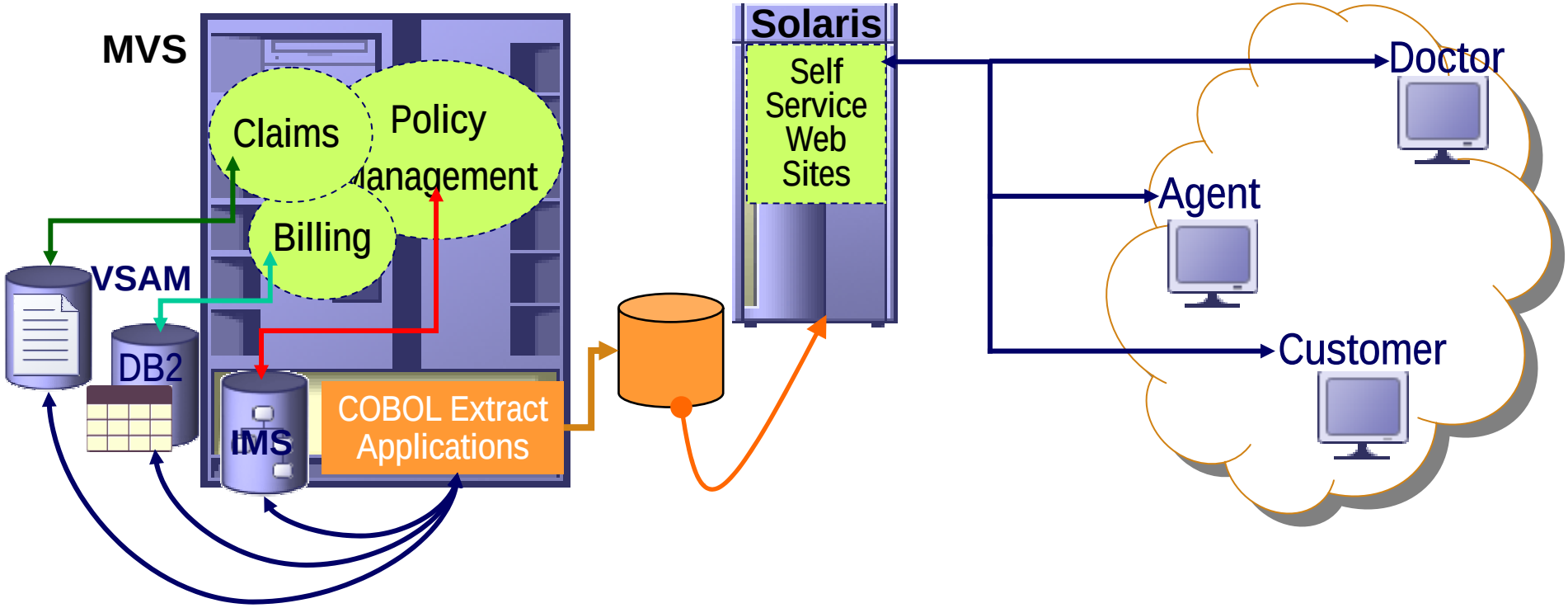
Mainframe data access from J2EE web application

Raj Datta
WebSphere IIS System z Solutions Architect
IBM Software Group

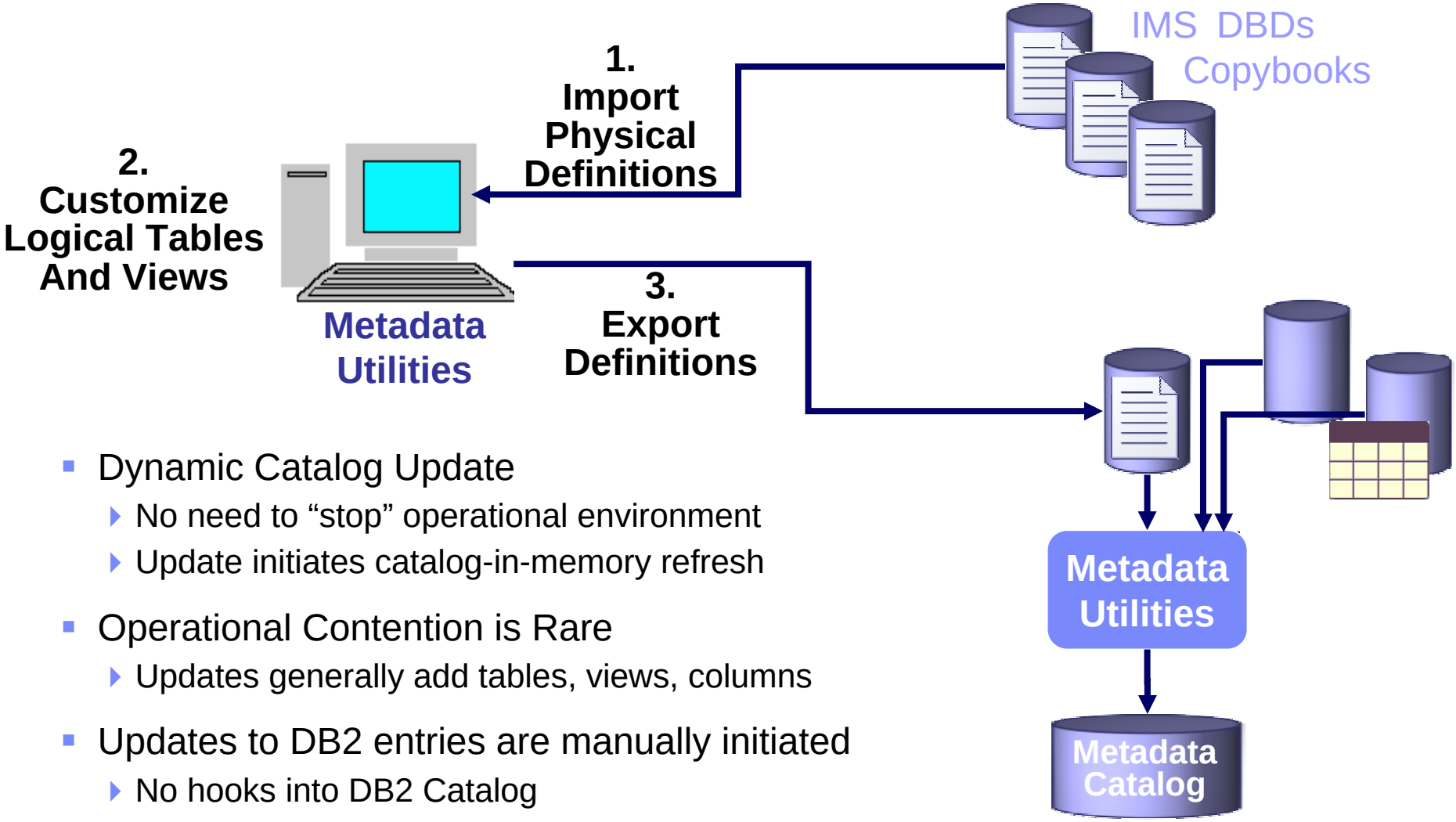
Traditional self-service environments

Insurance Carrier – enhance service

- Solution a: copy data to non-mainframe environments
  - Estimated cost $2M
  - Data refreshed every 30 hours or so
    - Data latency has real revenue impact
  - Solution b: integrate the IMS transactions
  - Estimated cost 10,000 man-hours per application



Metadata Management Workflow

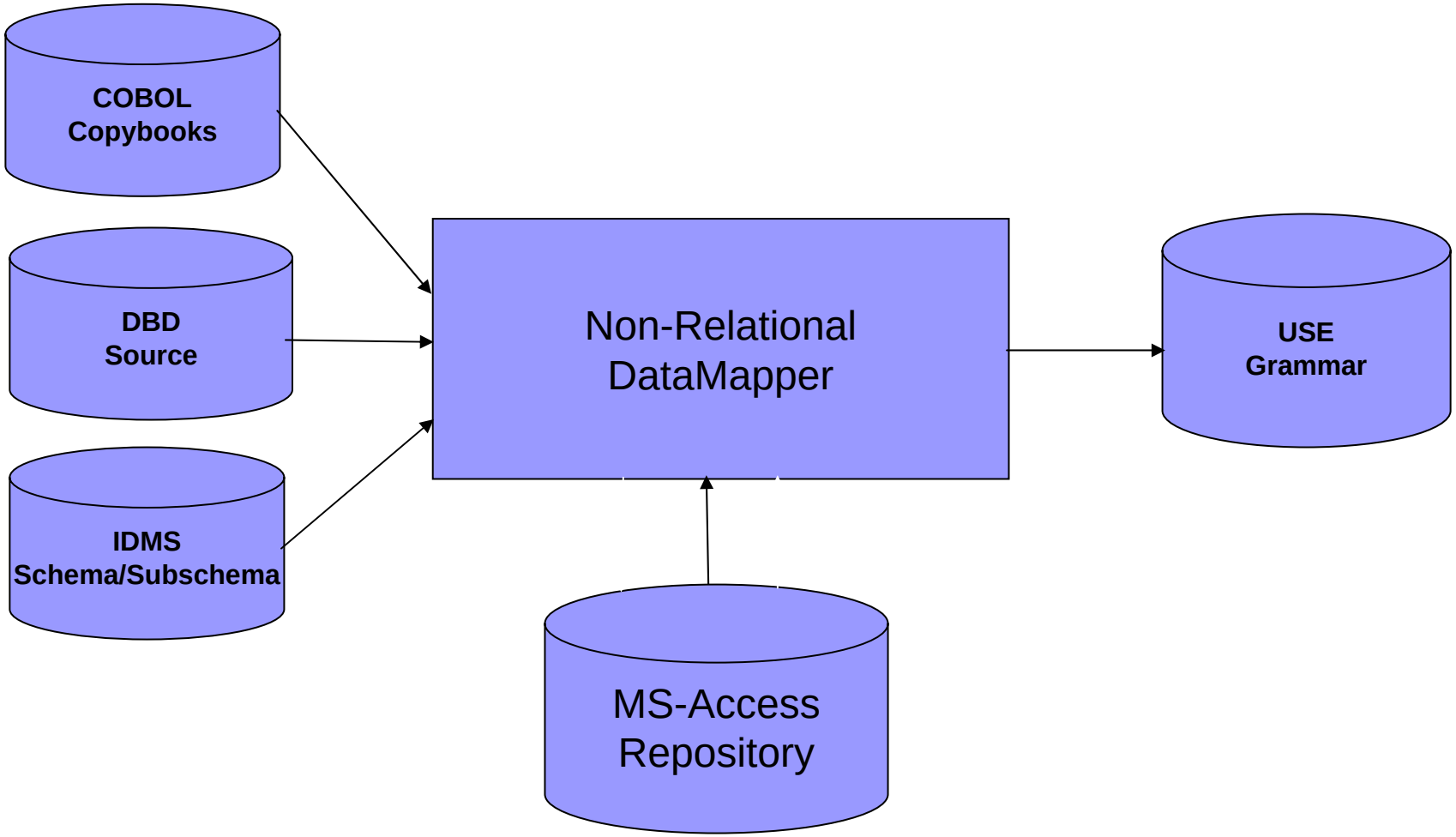


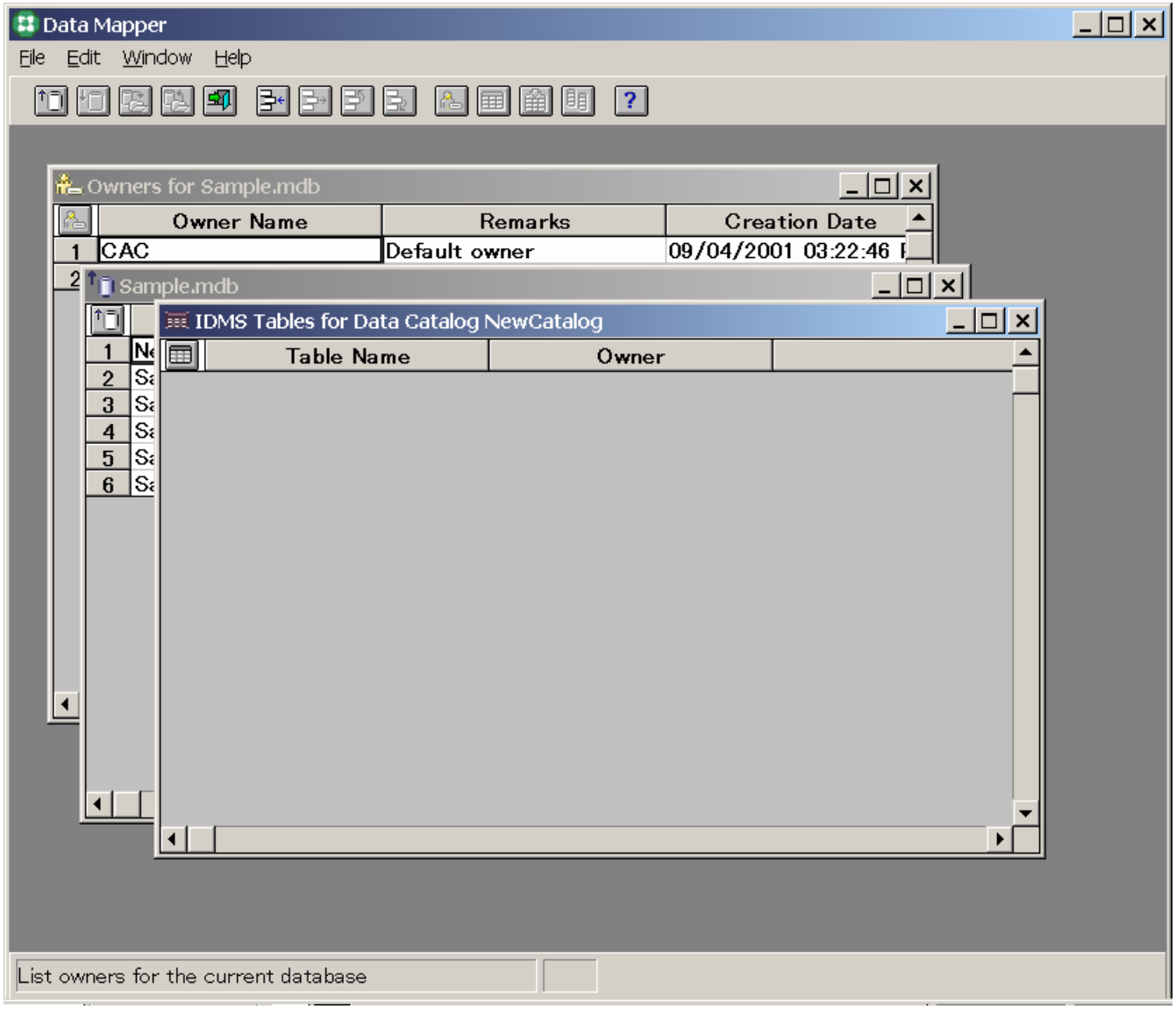
- Dynamic Catalog Update
  - No need to “stop” operational environment
  - Update initiates catalog-in-memory refresh
- Operational Contention is Rare
  - Updates generally add tables, views, columns
- Updates to DB2 entries are manually initiated
  - No hooks into DB2 Catalog

General Information

- The DataMapper is the primary tool for creating Logical Tables:
  - A Logical Table is a relational description of a non-relational database or file.
  - A Logical Table can also be thought of as a virtual table. They are materialized on the fly from the underlying database or file system.
  - Logical Tables are generally prefixed with their DBMS-type (e.g., an IMS Logical Table).
  - A DBMS is also referred to as a data source (e.g., an IDMS) which should not be confused with a CLI, JDBC or ODBC data source which can be used to access any type of Logical Table.
  - Logical Tables have attributes that are associated with all type of Logical Tables as well as DBMS-specific attributes and behaviors.
- The DataMapper is a Windows application
- The DataMapper relies heavily on the use of a Toolbar

Tool Overview




Data Mapper
File Edit Window Help
Owners for Sample.mdb
Owner Name
Remarks
Creation Date
1
CAC
Default owner
09/04/2001 03:22:46
Sample.mdb
IDMS Tables for Data Catalog NewCatalog
Table Name
Owner
List owners for the current database

Mapping Process Overview

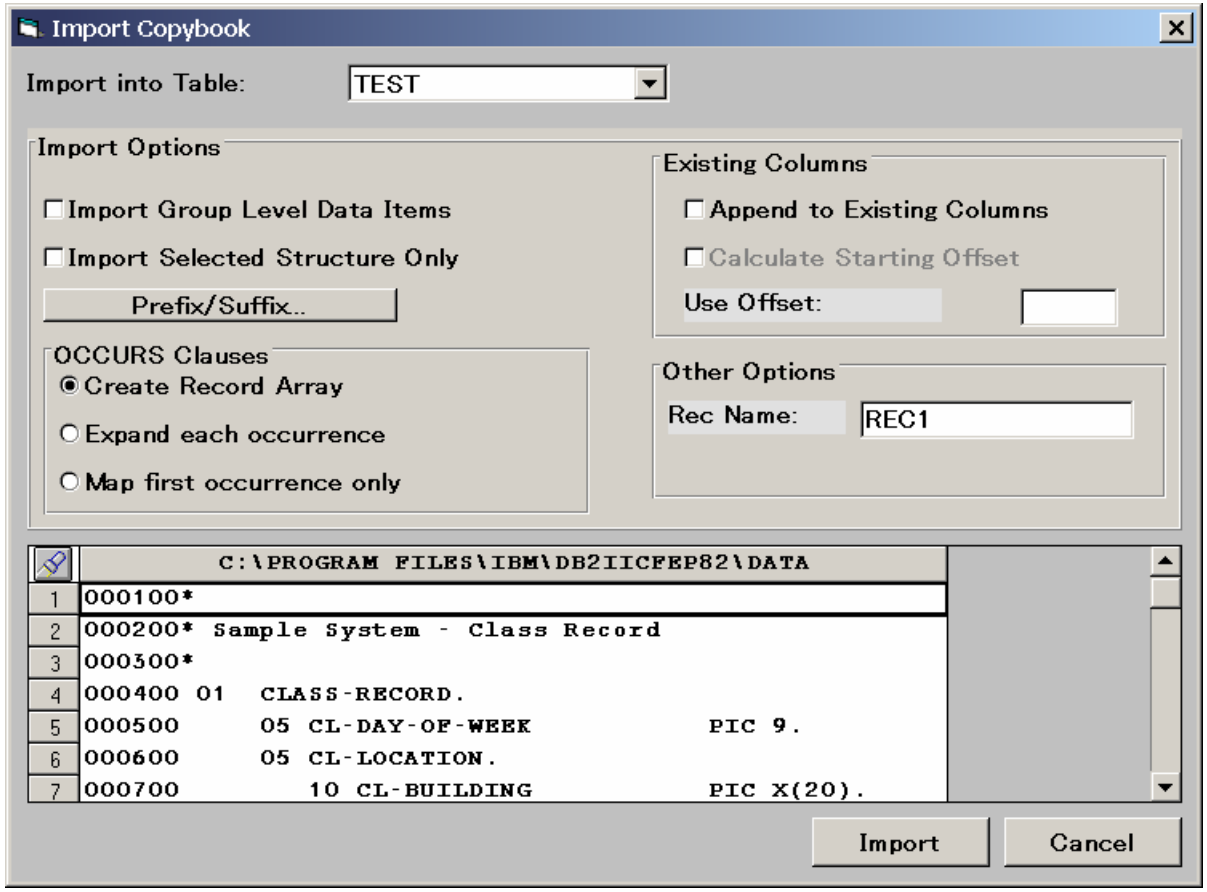
- Discovery and collection
- Mapping – using the DataMapper to create Logical Tables
- Generating the USE grammar and transferring it to Z/OS.
- Loading the System Catalogs

Mapping Process Discovery and Collection

- Identify Target Database/File
- Identify Source Defintion(s)
  - IMS - Logical/Physical DBD(s)
  - IDMS – Schemas/Subschemas
- Identify COBOL Copybooks
- Find out where the source lives so that it can be brought down to the Workstation where the DataMapper is installed.
- Discuss with the DBA what data is available and the keys/indexes that are available to access the data.
- Discuss with the business user/client developers what information is required and how it needs to be presented.
- Generally will want to create some initial discovery mappings and issue queries to determine what is really in the database, general performance aspects and data “quirks”
- Create new Logical Tables to meet individual business needs – a Logical Table should generally represent the data needed for a particular query or class of queries.

Mapping Process Using the DataMapper

- Launch Data Mapper
- Select/Create Repository
- Select/Create Data Catalog
- Optionally, define Owner (s)
- Load Source Definitions
  - Use Built-in FTP capabilities to download or perform manually
- Create Table
- Import COBOL copy book (s) to create the Tables Columns
  - Use built-in FTP capabilities to download or perform manually
- Review/tailor Column Definitions
- Define Index(es)
- Identify Keys
- Generate USE Grammar


Import Copybook
Import into Table:
TEST
Import Options
Import Group Level Data Items
Import Selected Structure Only
Prefix/Suffix...
OCCURS Clauses
Create Record Array
Expand each occurrence
Map first occurrence only
Existing Columns
Append to Existing Columns
Calculate Starting Offset
Use Offset:
Other Options
Rec Name:
REC1
C:\PROGRAM FILES\IBM\DB2IICFEP82\DATA
1 000100*
2 000200* Sample System - Class Record
3 000300*
4 000400 01 CLASS-RECORD.
5 000500 05 CL-DAY-OF-WEEK PIC 9.
6 000600 05 CL-LOCATION.
7 000700 10 CL-BUILDING PIC X(20).
Import
Cancel

Importing COBOL Copybooks Guidelines

- Review the contents of the copybook
- Look for a complex object:
  - Redefinitions
  - OCCURS clauses
  - Multiple OCCURS clauses
- Generally, want to create separate tables for each:
  - Redefinition
  - OCCURS group – with “key” and non-repeating fields
- Do not use default import settings for a complex object
- Consider using a reference table when you encounter a complex object

Columns Overview

- Columns are automatically created based on the data items contained in the COBOL copybook.
- SQL data type are assigned to each Column based on the PICTURE clause associated with each data item.
- Relative offset mapping start at zero:
  - Record/Segment
  - Record Array
- Remarks are not stored in System Catalog.
- The Create/Update Column dialog box is DBMS-specific but contains common elements/functions.
- Common functions:
  - SQL data type support
  - Native data type support
  - NULL specifications
- Techniques for dealing with unsupported data types.

Columns SQL Data Types

- Commonly used data types:
  - CHAR
  - DECIMAL
  - SMALLINT
  - INTEGER
- Exotic data types:
  - FLOAT
  - VARCHAR
  - LONG VARCHAR
  - GRAPHIC
  - VARGRAPHIC
  - LONG VARGRAPHIC

Columns SQL Data Types

Name	COBOL PICTURE CLAUSE
Character	PIC X(n).
Packed Decimal	PIC S9(n)V9(n) COMP-3.
Unsigned Packed Decimal	PIC 9(n)V9(n) COMP-3.
Zoned Decimal	PIC S9(n).
Unsigned Zoned Decimal	PIC 9(n).
Half Word	PIC S9(4) COMP.
Unsigned Half Word	PIC 9(4) COMP.
Full Word	PIC S9(8) COMP.
Unsigned Full Word	PIX 9(8) COMP.
Double Word	COMP-2.
Variable Length Character	STUCTURE. LENGTH PIC S9(4) COMP. DATA PIC X(n).

Mapping Process Generating USE Grammar

- Select Data Catalog Window
- Select Generate
  - File->Generate USE Statements…
- Identify file name and
  - Save to disk or,
  - Use built-in FTP support to transmit file to Z/OS
- Review generated grammar

Mapping Processing Loading the System Catalogs

- Once the USE grammar is on Z/OS you run the Meta Data Utility to load the Logical Table definitions into the System Catalogs.
- The Meta Data Utility:
  - Performs final Table and Column validation
  - Obtains additional DBMS-specific information
  - Populates the System Catalog

Mapping Processing Loading the System Catalogs (con't)

```
1: EE - TN3270 Plus
Host  Edit  View  Setup  Macros  Internet  Help

  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       RAJORSH.JCL(CACMETAU) - 01.03                    Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ********************************* Top of Data ********************************
000001 //RAJORSHM   JOB  ('RAJORSH',376),'RAJ DATTA',TIME=60,
000002 //                CLASS=A,NOTIFY=&SYSUID,
000003 //                MSGCLASS=H,MSGLEVEL=(1,1)
000004 //METAUTL PROC CAC='RAJORSH',        INSTALLED HIGH LEVEL QUALIFIER
000005 //                IMS='IMSVS.IMS7',    IMS HIGH LEVEL QUALIFIER
000006 //                RGN=OM,            REGION SIZE
000007 //                GRAMMAR=USETEST  META DATA GRAMMAR MEMBER NAME
000008 //METAU    EXEC PGM=CACMETA,REGION=&RGN
000009 //STEPLIB  DD   DISP=SHR,DSN=&CAC..V8R2M00.SCACLOAD
000010 //CTRANS   DD   DISP=SHR,DSN=&CAC..V8R2M00.SCACSASC
000011 //CACCAT   DD   DISP=SHR,DSN=RAJORSH.LINEAR.CATALOG
000012 //CACINDX  DD   DISP=SHR,DSN=RAJORSH.LINEAR.CATINDX
000013 //DBDLIB   DD   DISP=SHR,DSN=&CAC..V8R2M00.SCACLOAD
000014 //         DD   DISP=SHR,DSN=&IMS..DBDLIB
000015 //CICSUID  DD   DISP=SHR,DSN=RAJORSH.JCL(CACCICSU)
000016 //SYSTERM  DD   SYSOUT=*
000017 //SYSPRINT DD   SYSOUT=*
000018 //SYSIN    DD   DISP=SHR,DSN=RAJORSH.JCL(&GRAMMAR)
000019 //         PEND
000020 //METAUTL EXEC METAUTL
****** ******************************** Bottom of Data ******************************
TB *                    1:EE                    13,09   00:00.160 07:15 04/11/06
Connected to stlabee.svl.ibm.com port 23                 00:00.160 13,09 IBM-3278-4-E
```

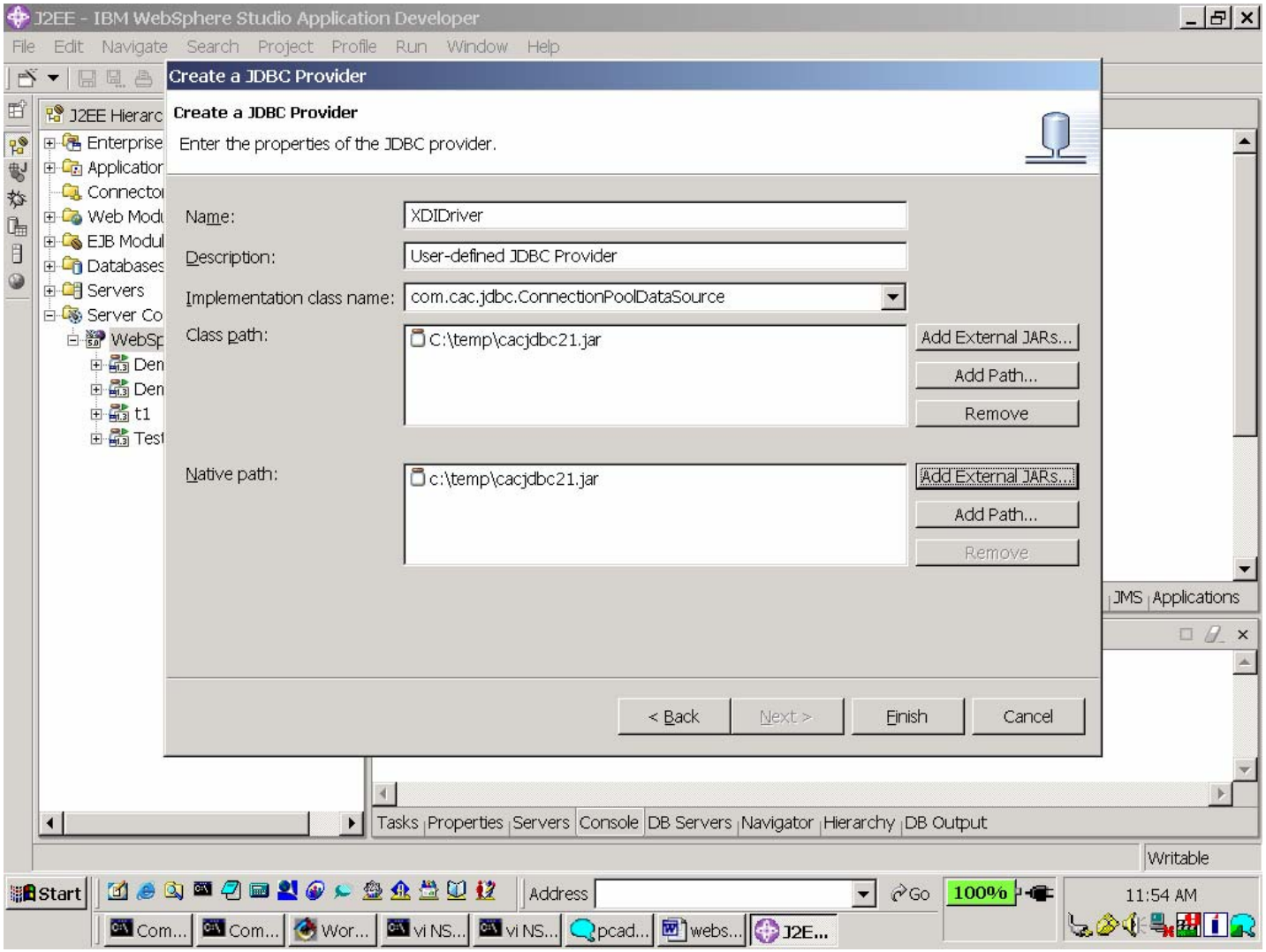
Configure Classic Federation Data Server

```
1: EE - TN3270 Plus
Host  Edit  View  Setup  Macros  Internet  Help

File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT        RAJORSH.JCL(CACDSCF) - 01.11                    Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ********************************* Top of Data ********************************
000001 SERVICE INFO ENTRY = CACCNTL CNTL 0 1 1 100 4 5M 5M NO_DATA
000002 SERVICE INFO ENTRY = CACLOG LOG 1 1 1 100 1 5M 5M DISPLAY
000003 SERVICE INFO ENTRY = CACQP RAJEE 2 9 10 10 4 5M 5M CACQPCF
000004 SERVICE INFO ENTRY = CACQP WCA004DS 2 9 10 10 4 5M 5M CACQPCF
000005 SERVICE INFO ENTRY = CACDRA IMS  2 1 1 50 4 5M 5M 01,RAJORS,DFHSAM04
000006 SERVICE INFO ENTRY = CACVSMS VSAMSRV 2 1 1 50 4 5M 5M CLOSE_ON_IDLE
000007 SERVICE INFO ENTRY = CACINIT TCPIP 2 1 1 100 4 5M 5M ¥
000008  TCP/P39D/8095
000009 MESSAGE POOL SIZE = 16777216
000010 NL = US ENGLISH
000011 NL CAT = DD:ENGCAT
000012 STATIC CATALOGS = 0
****** ********************************* Bottom of Data *****************************
TB ·                    1:EE                    05,15   00:00.060  08:11  04/27/06
Connected to stlabee.svl.ibm.com port 23                 00:00.060 05,15 IBM-3278-4-E
```

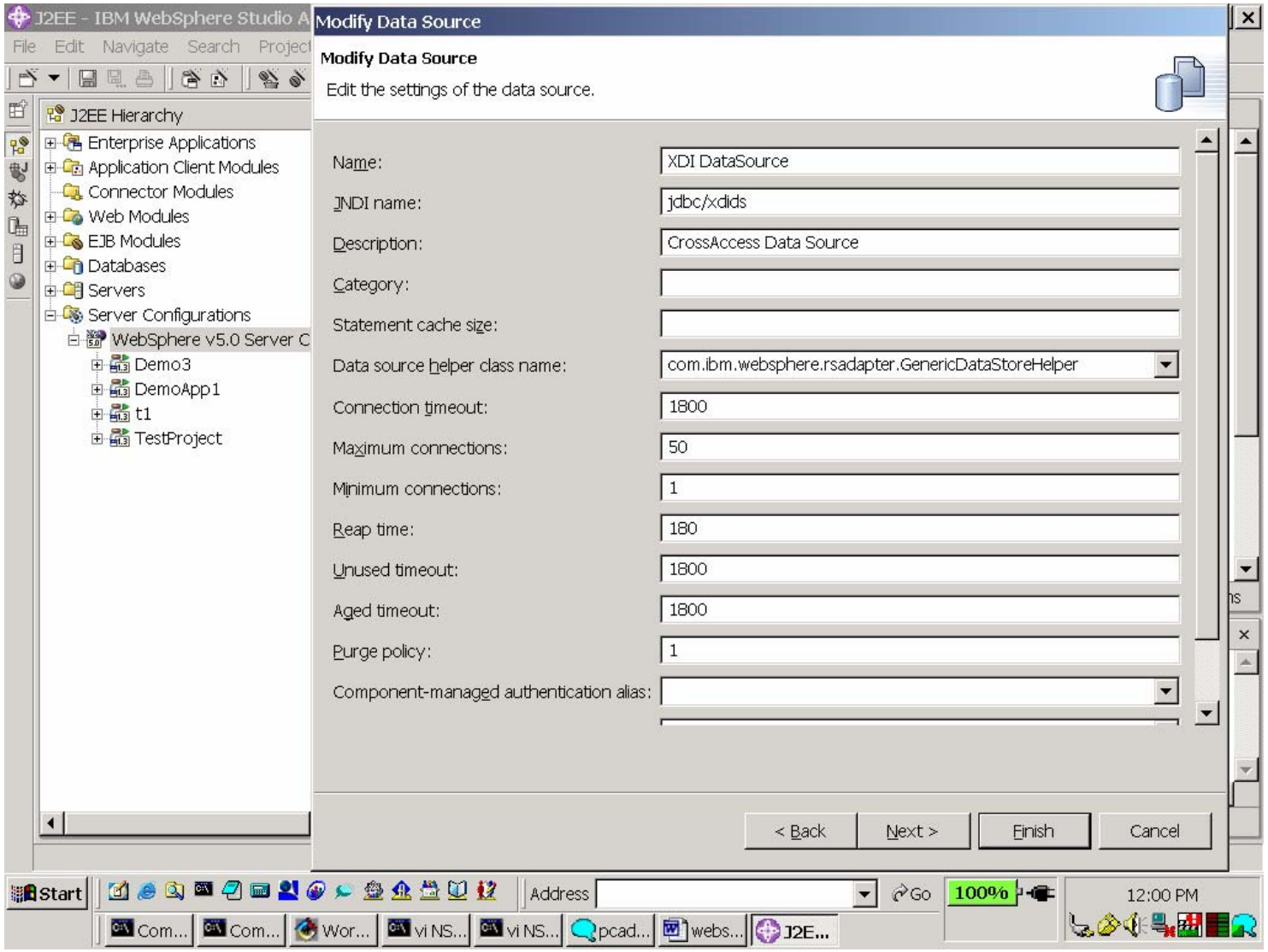
Start Classic Federation Data Server

```
1: EE - TN3270 Plus
Host  Edit  View  Setup  Macros  Internet  Help

Display  Filter  View  Print  Options  Help
SDSF OUTPUT DISPLAY RAJORSHD JOB00293  DSID     2 LINE 0        COLUMNS 02- 81
COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA ***********************************
                    J E S 2   J O B   L O G   --   S Y S T E M   S Y E E   --   N O D E

21.01.09 JOB00293  ---- WEDNESDAY, 26 APR 2006 ----
21.01.09 JOB00293  IRR0101  USERID RAJORSH  IS ASSIGNED TO THIS JOB.
21.01.09 JOB00293  ICH70001I RAJORSH  LAST ACCESS AT 20:59:15 ON WEDNESDAY, APRI
21.01.09 JOB00293  $HASP373 RAJORSHD STARTED - INIT 7    - CLASS A - SYS SYEE
21.01.10 JOB00293  CAC00105I LOG V8.2 03102005: STARTED
21.01.10 JOB00293  CAC00100I CONTROLLER: LOGGING STARTED
21.01.11 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.11 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.11 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.11 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.12 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.12 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.12 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.12 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.12 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.12 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.13 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.13 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.13 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.13 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.13 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.13 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.14 JOB00293  CAC00105I QUERY PROCESSOR V8.2 03102005: STARTED
21.01.14 JOB00293  CAC00102I CONTROLLER: STARTED CACQP
21.01.14 JOB00293  CAC00105I DRA V8.2 03102005: STARTED
21.01.14 JOB00293  CAC00102I CONTROLLER: STARTED CACDRA
21.01.14 JOB00293  CAC00105I VSAM SERVICE V8.2 03102005: STARTED
21.01.14 JOB00293  CAC00102I CONTROLLER: STARTED CACVSMS
21.01.25 JOB00293  CAC00105I CONNECTION HANDLER V8.2 03102005: STARTED
21.01.25 JOB00293  CAC00102I CONTROLLER: STARTED CACINIT
21.01.25 JOB00293  CAC00103I DATA SERVER: V8.2 03102005 READY
        1 //RAJORSHD  JOB ('RAJORSH',376),'RAJ DATTA',TIME=60,
          //                CLASS=A,NOTIFY=&SYSUID,
          //                MSGCLASS=H,MSGLEVEL=(1,1)
          //*****************************************************************
          IEFC653I SUBSTITUTION JCL - ('RAJORSH',376),'RAJ DATTA',TIME=60,CLASS=
TB ·                1:EE                                 04,21   00:00.020 08:07 04/27/06
Connected to stlabee.svl.ibm.com port 23                 00:00.020 04,21 IBM-3278-4-E
```

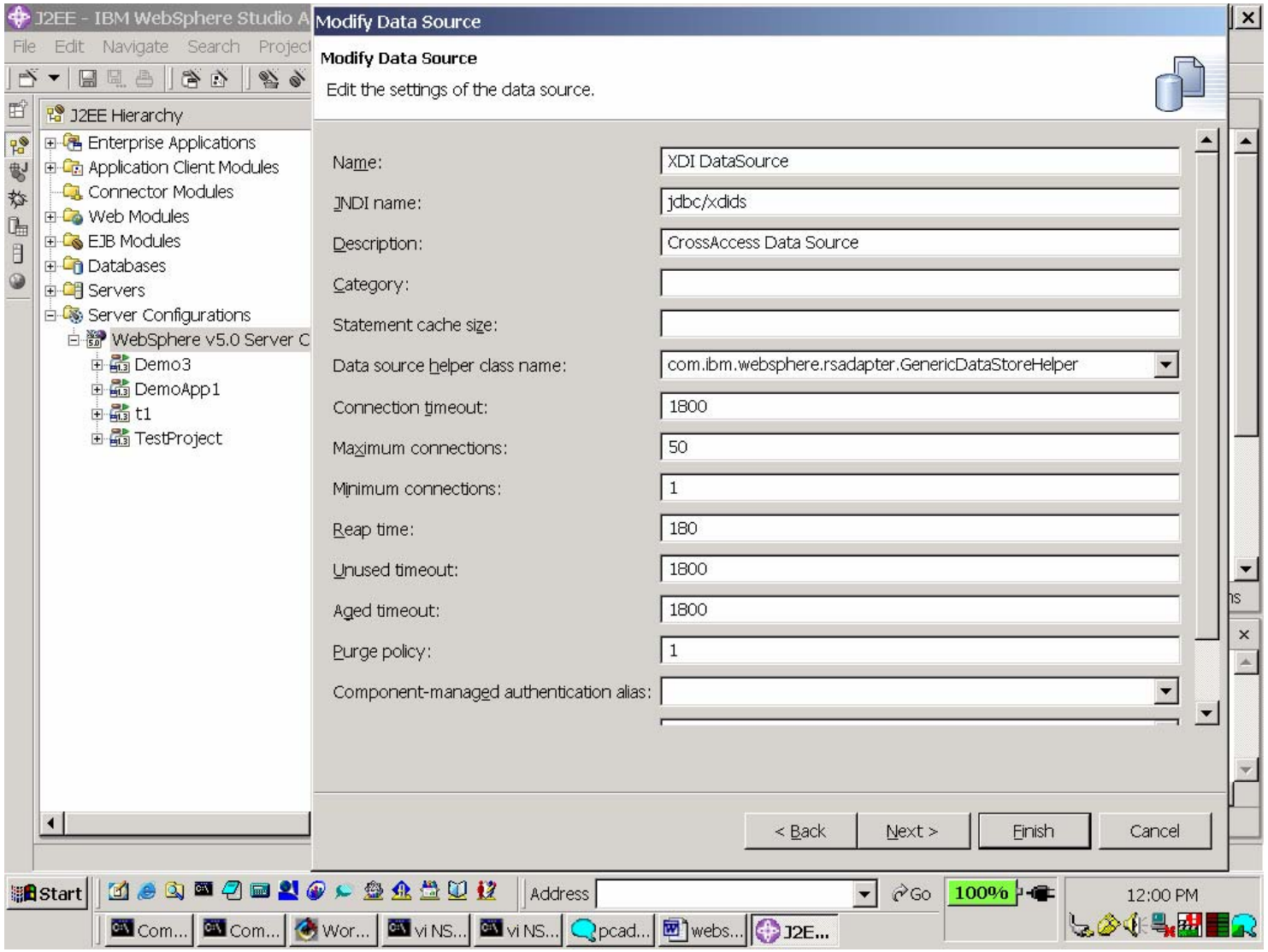
Configuring a JDBC Provider in WebSphere Studio



Configuring a JDBC DataSource in WebSphere Studi



Configuring a JDBC DataSource in WebSphere Studio



Configuring a JDBC DataSource in WebSphere Studio (con't)

three properties essential to the working of WSIICF DataSource:

databaseName - type java.lang.String, corresponds to the SERVICE name.

port - type java.lang.String, corresponds to port number, the initiator is listening on.

serverName - type java.lang.String, corresponds to the IP Address or the Host Name.

Configuring a JDBC DataSource in WebSphere Studio (con't)

J2EE - IBM WebSphere Studio Application Developer

File Edit Navigate Search Project Profile Run Window Help

J2EE Hierarchy

- Enterprise Applications
- Application Client Modules
- Connector Modules
- Web Modules
- EJB Modules
- Databases
- Servers
- Server Configurations
  - WebSphere v5.0 Server Configur.
    - Demo3
    - DemoApp1
    - t1
    - TestProject

*WebSphere v5.0 Server Configuration

JDBC provider list:

Name	Implementation class name
XDI	com.cac.jdbc.ConnectionPoolDataSource
XDIDriver	com.cac.jdbc.ConnectionPoolDataSource

Add... Edit... Remove

Data source defined in the JDBC provider selected above:

Name	JNDI Name	Type
Data source 1	jdbc/ds1	V5
Data source 2	jdbc/ds2	V5
Data source 3	jdbc/xdids3	V5

Add... Edit... Remove

Resource properties defined in the data source selected above:

Name	Value	Type
databaseName	WCA004DS	java.lang.String
port	8095	java.lang.String
serverName	p39d	java.lang.String

Add... Edit... Remove

Configuration | Paths | Environment | Web | Data source | Ports | Variables | Trace | Security | EJB | J2C | JMS | Applications

Console

Tasks | Properties | Servers | Console | DB Servers | Navigator | Hierarchy | DB Output

Writable

Start | Address | Go | 100% | 12:17 PM

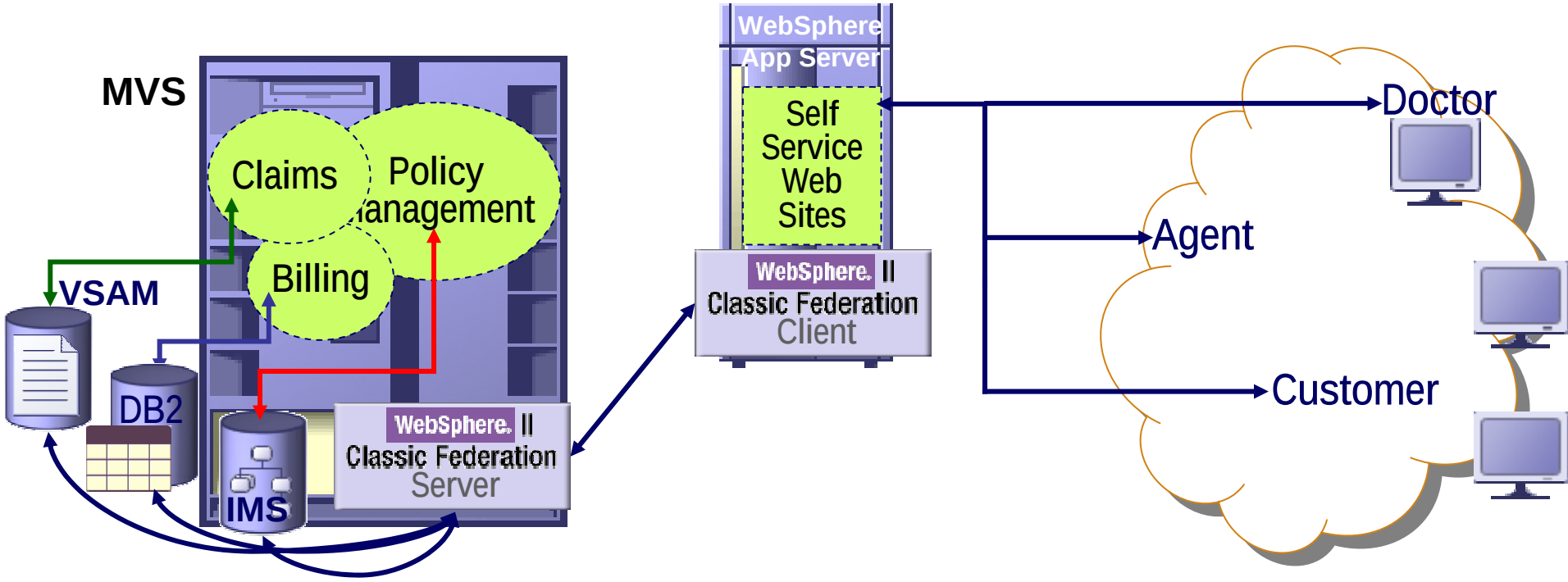
Co... Co... Wo... vi N... vi N... pca... wsa... J2... Anil...

Sample Java bean code

```
// Start Bean Method
public void test1()
        {
                DataSource ds1 = null;
                Connection cx1 = null;
                UserTransaction userTran = null;
                try
                {
                        InitialContext initCtx = new InitialContext();
                        userTran = (UserTransaction)initCtx.lookup(
                  "java:comp/UserTransaction");
                  ds1 = (DataSource) initCtx.lookup("java:comp/env/jdbc/xdids");

                cx1 = ds1.getConnection();
                PreparedStatement st1 =
cx2.prepareStatement(“select partno, descript from sys.stokstat where partcod=?);
                st2.setString(1,“77");
                ResultSet rs = st2.executeQuery();
                st2.close();
                }
                catch ( Exception e )
                {
                        System.out.println("Exception e " + e);
                  }
        }
// End Bean Method
```

IBM solution - empower self-service

- Provide up-to-the-minute policy, claims and accounting information
  - Connect interactive voice response (IVR) system to IMS, VSAM & DB2
    - $250K versus $2M
  - Connect operational data with self-service Web sites
    - 200 man-hours versus 10,000



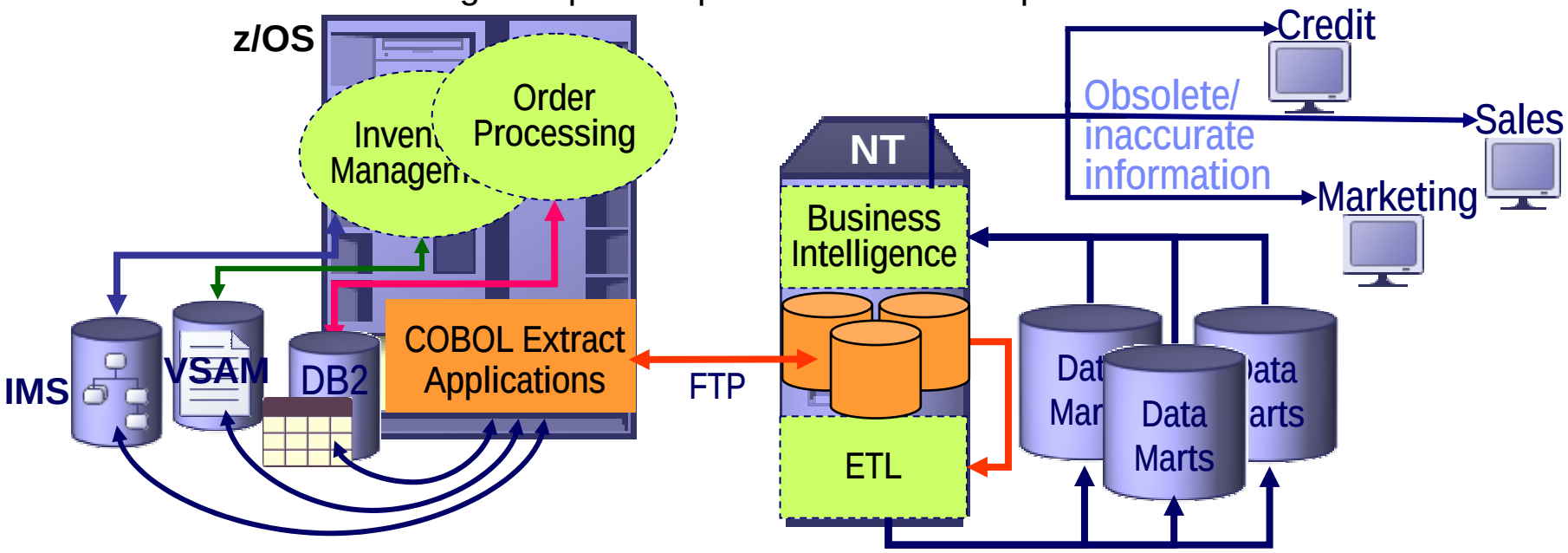
IBM Software Group

Real-time ETL of Mainframe data

Raj Datta
WebSphere IIS System z Solutions Architect
IBM Software Group

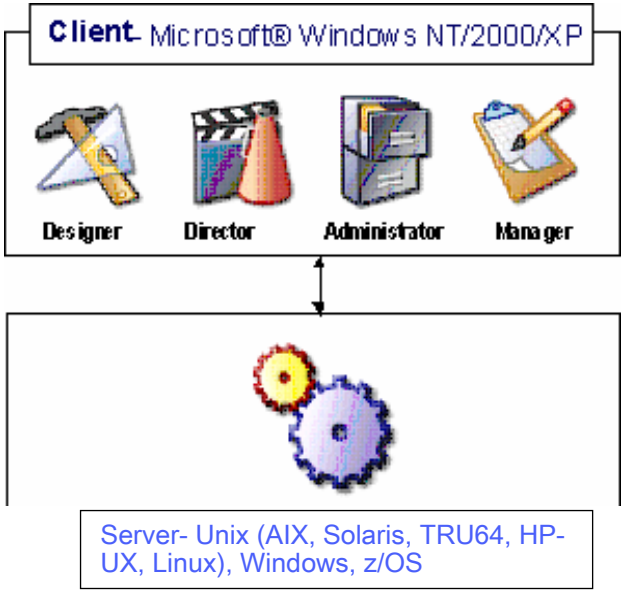
Traditional delivery of data to business intelligence

- Disjoint Process
  - Build and maintain mainframe “extract” process
  - Build and maintain distributed data transform & load
  - Data latency has a real revenue impact
- Management challenges lead to increasing costs
  - Multiple skill sets required:
    Mainframe programming & data warehouse design/build
  - Coordinating multiple components and development teams



DataStage Enterprise Edition Components

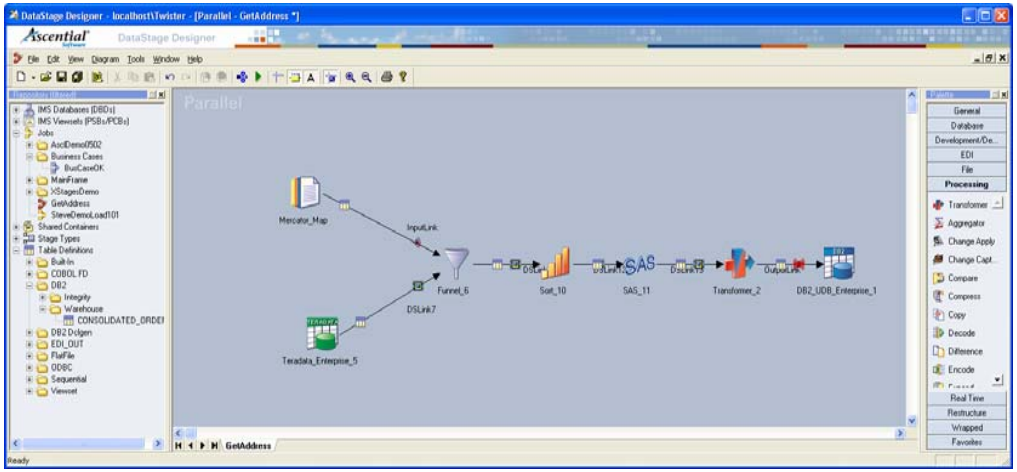
- Designer
  - A design interface used to create DataStage applications (known as jobs)
- Manager
  - Used to view and edit the contents of the DataStage Repository
- Administrator
  - Used to perform administration tasks such as setting up DataStage users, creating and moving projects, and setting up purging criteria
- Director
  - Used to validate, schedule, run, and monitor DataStage jobs
- MetaStage
  - Used for managing enterprise meta data across design, integration, and BI tools



Client/Server Development Environment

Designer

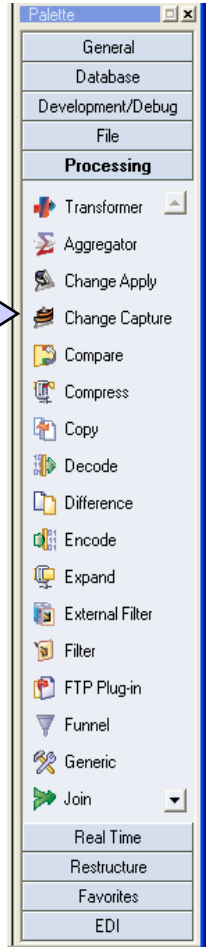
- Complete development environment
  - Graphical, drag and drop metaphor
  - Develop sequentially, deploy in parallel
  - Component-based architecture
  - Does not force design methodology
  - Get started quickly
  - Reuse capabilities

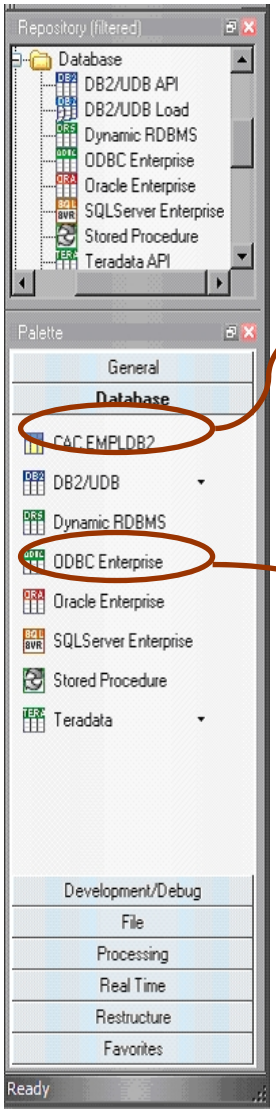


Designer: Components Available

- Over 50 pre-built components available including
  - Files
  - Database
  - Lookup
  - Sort, Aggregation, Transformer
  - Join, Merge
  - Filter, Funnel, Switch, Modify
  - SAS
  - Remove Duplicates
  - Restructure stages

Sample of Stages Available





Bookmark frequently accessed Data sources

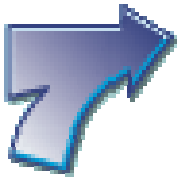
Use to Connect to WS IICF

Examples of Pre-Built Scalable Stages

ODBC stage: extend DataStage's capabilities to communicate with external data sources.

Sort stage: used to perform more complex sort operations in parallel.

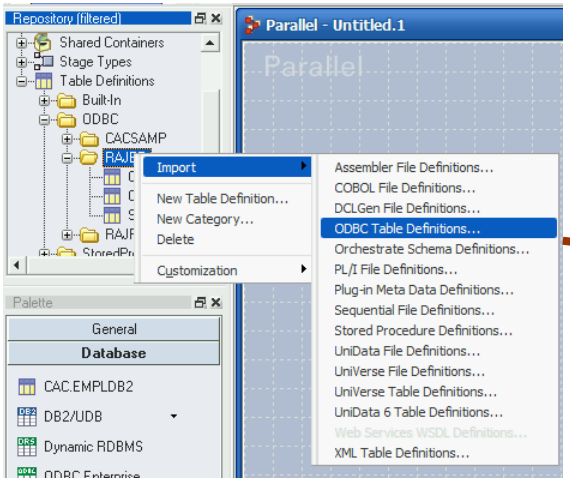
Transformer stage: performs any conversions required on an input data set, and then passes the data to another active stage or a stage that writes data to a target database or file.

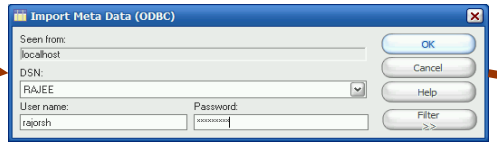


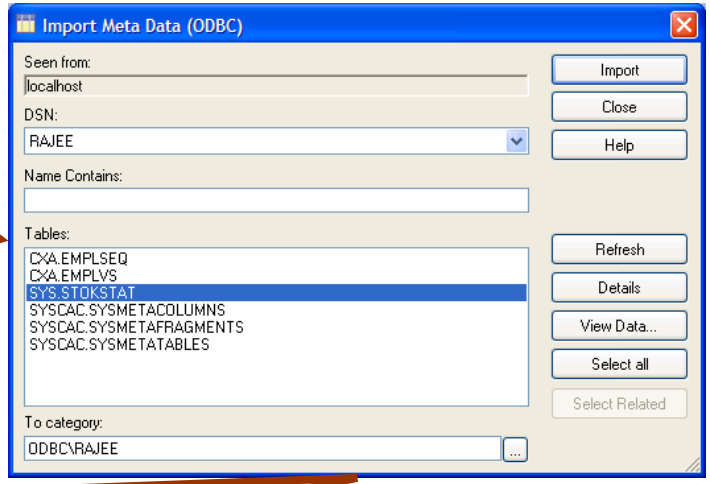
Merge stage: combines a sorted master data set with one or more sorted update data sets.

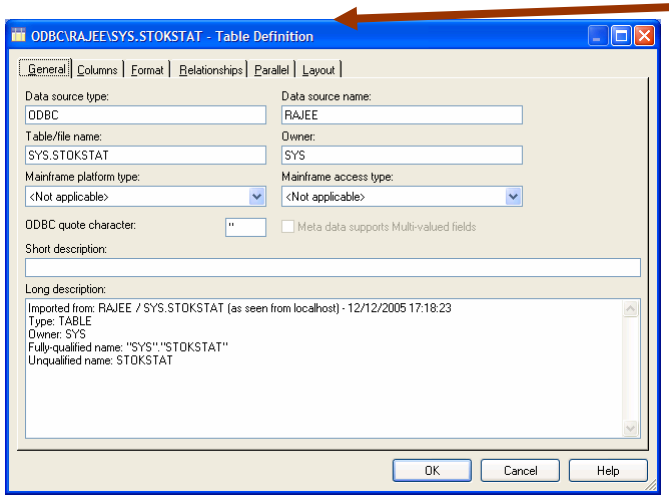
Join stage: performs join operations on two or more data sets input to the stage and then outputs the resulting data set. The stage can perform one of four join operations: inner; left outer; right outer; and full outer.

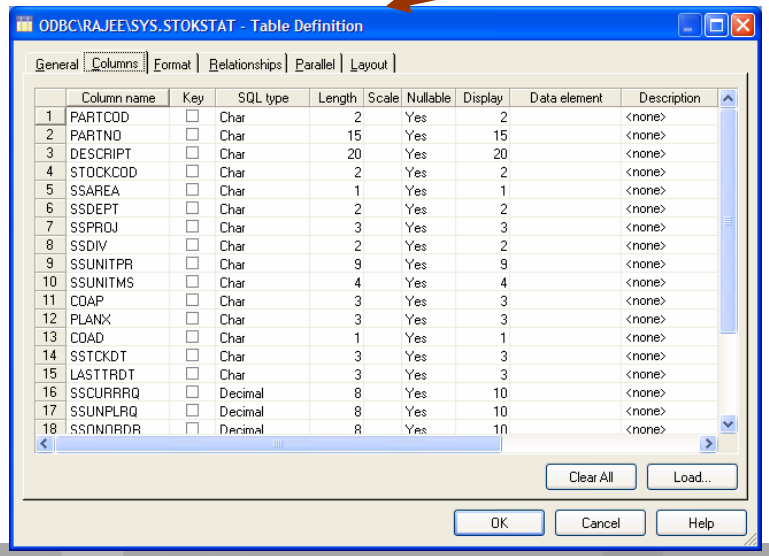
Importing IICF tables through ODBC







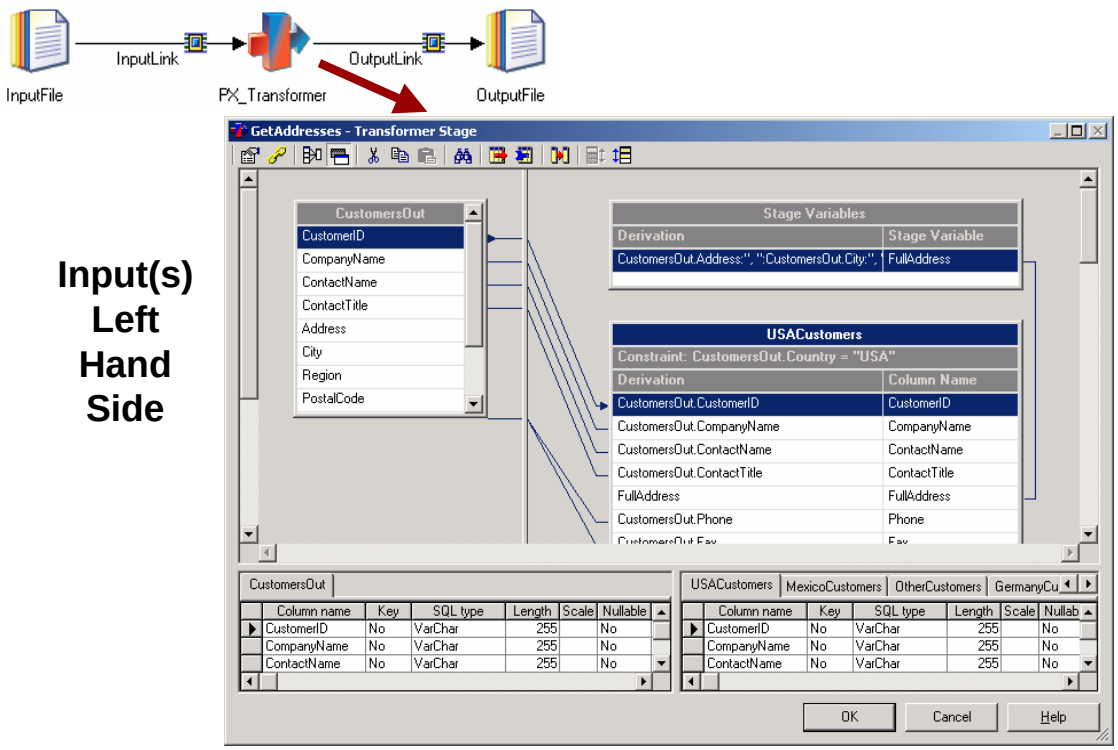




	Column name	Key	SQL type	Length	Scale	Nullable	Display	Data element	Description
1	PARTCOD	☐	Char	2		Yes	2		<none>
2	PARTNO	☐	Char	15		Yes	15		<none>
3	DESCRIPT	☐	Char	20		Yes	20		<none>
4	STOCKCOD	☐	Char	2		Yes	2		<none>
5	SSAREA	☐	Char	1		Yes	1		<none>
6	SSDEPT	☐	Char	2		Yes	2		<none>
7	SSPROJ	☐	Char	3		Yes	3		<none>
8	SSDIV	☐	Char	2		Yes	2		<none>
9	SSUNITPR	☐	Char	9		Yes	9		<none>
10	SSUNITMS	☐	Char	4		Yes	4		<none>
11	COAP	☐	Char	3		Yes	3		<none>
12	PLANX	☐	Char	3		Yes	3		<none>
13	COAD	☐	Char	1		Yes	1		<none>
14	SSTCKDT	☐	Char	3		Yes	3		<none>
15	LASTTRDT	☐	Char	3		Yes	3		<none>
16	SSCURRRQ	☐	Decimal	8		Yes	10		<none>
17	SSUNPLRQ	☐	Decimal	8		Yes	10		<none>
18	SSONORDR	☐	Decimal	8		Yes	10		<none>

Transformer

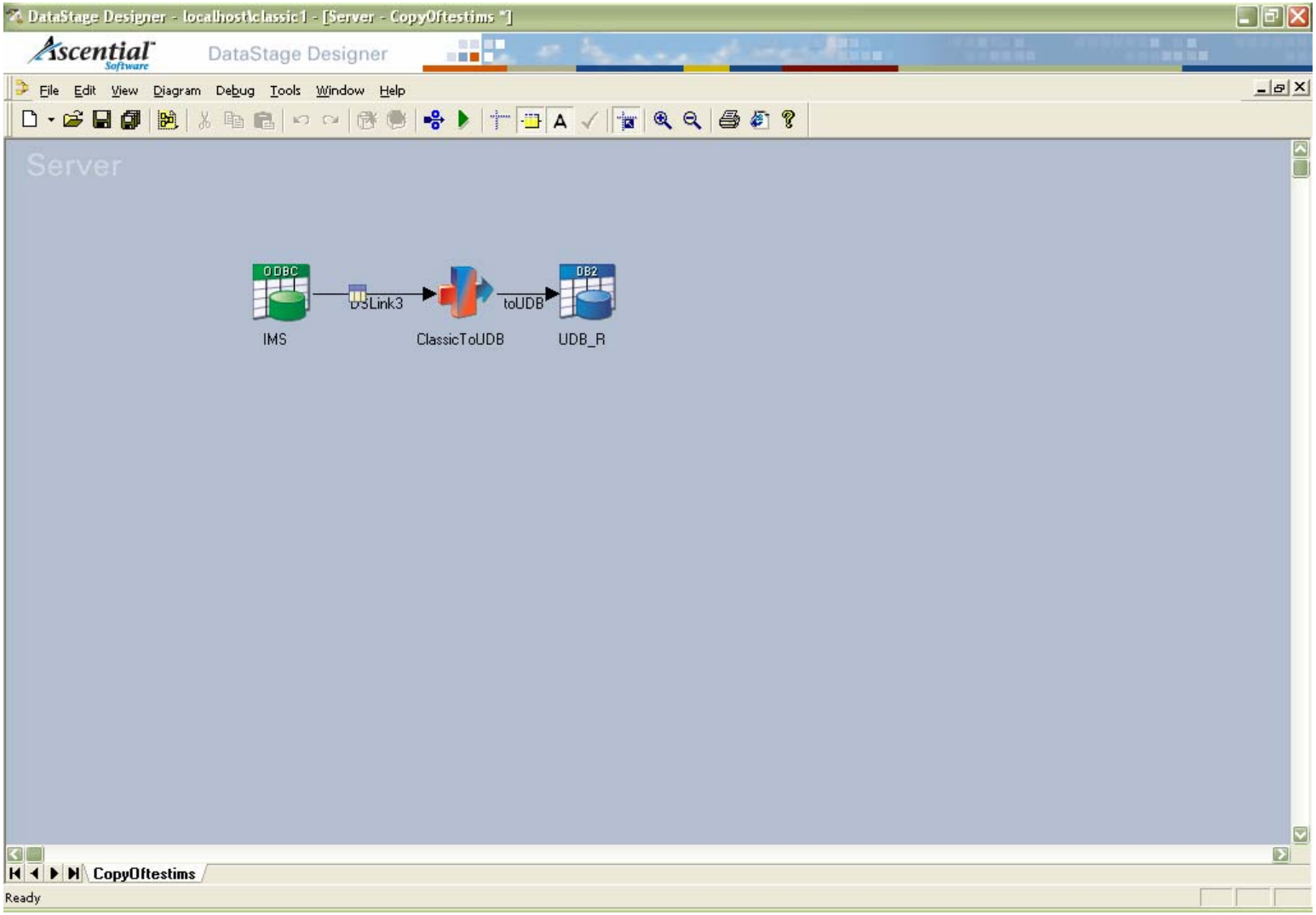
The Transformer stage provides a one-stop location for you to easily create simple or complex transformations.



Input(s) Left Hand Side

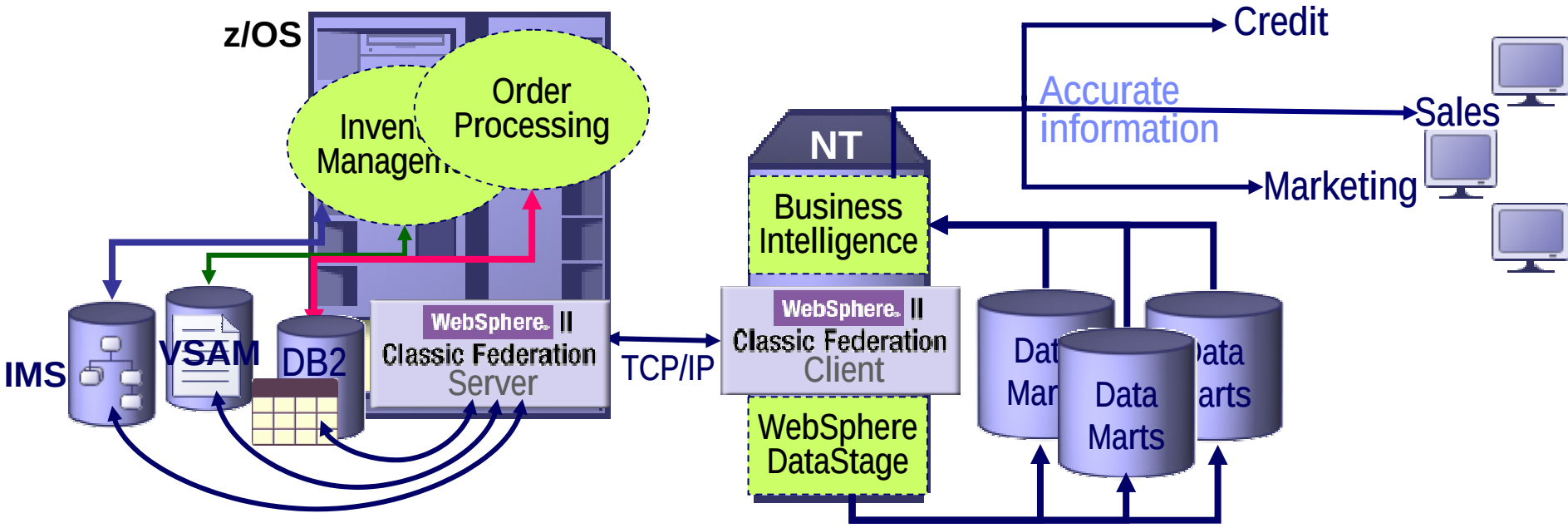
Input(s) Right Hand Side

DataStage job extracting IMS data to UDB



IBM solution -- feed operational data to ETL via SQL

- Dynamically connect data warehouse tool with mainframe data
  - No dependence on mainframe development
    - Dramatically simplified management: One team owns it all
  - One consistent process leverages “power” of ETL tooling
    - Development time “cut in half”
  - Empowers additional uses
    - Dynamic query by business intelligence tools extends the warehouse

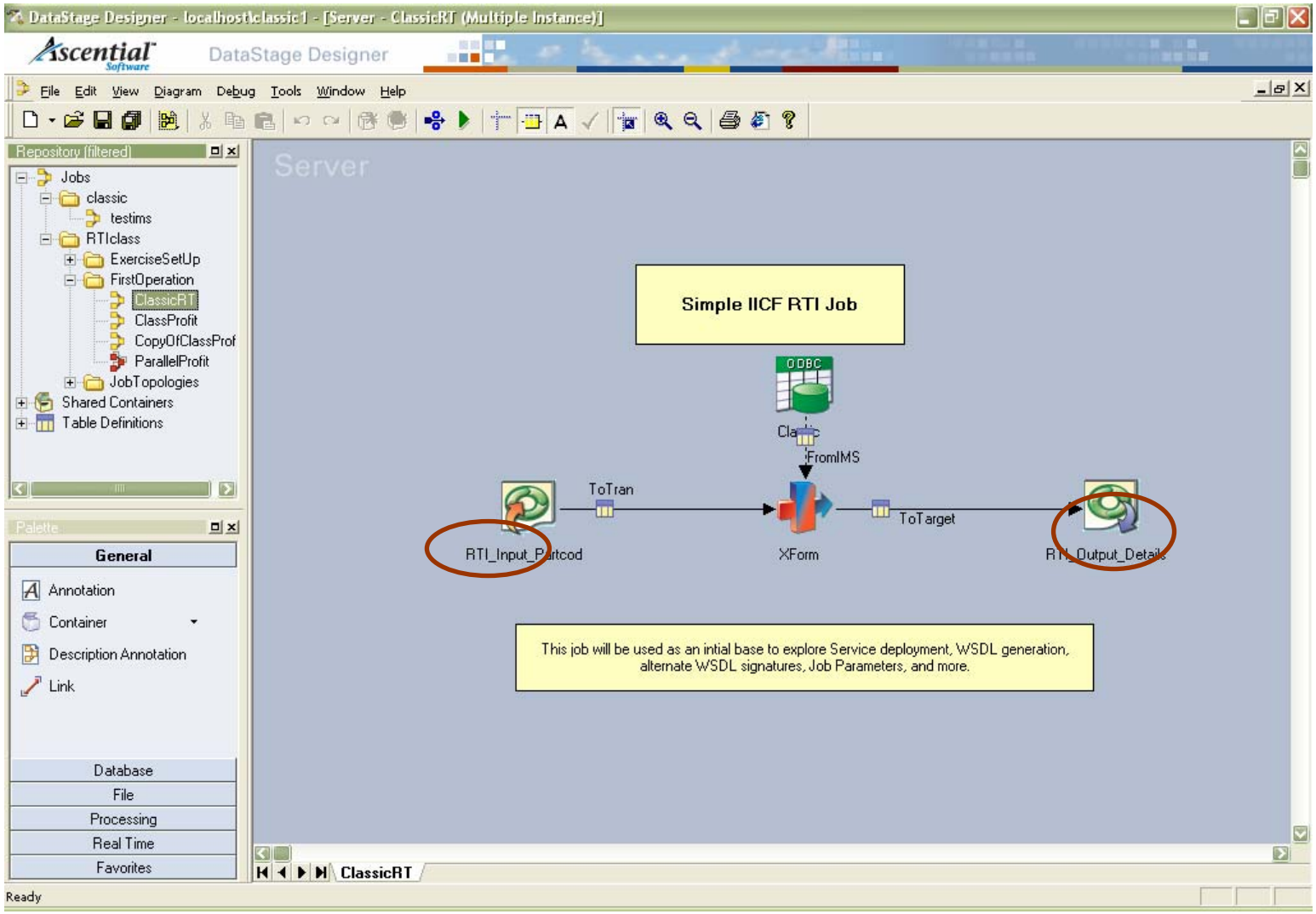


IBM Software Group

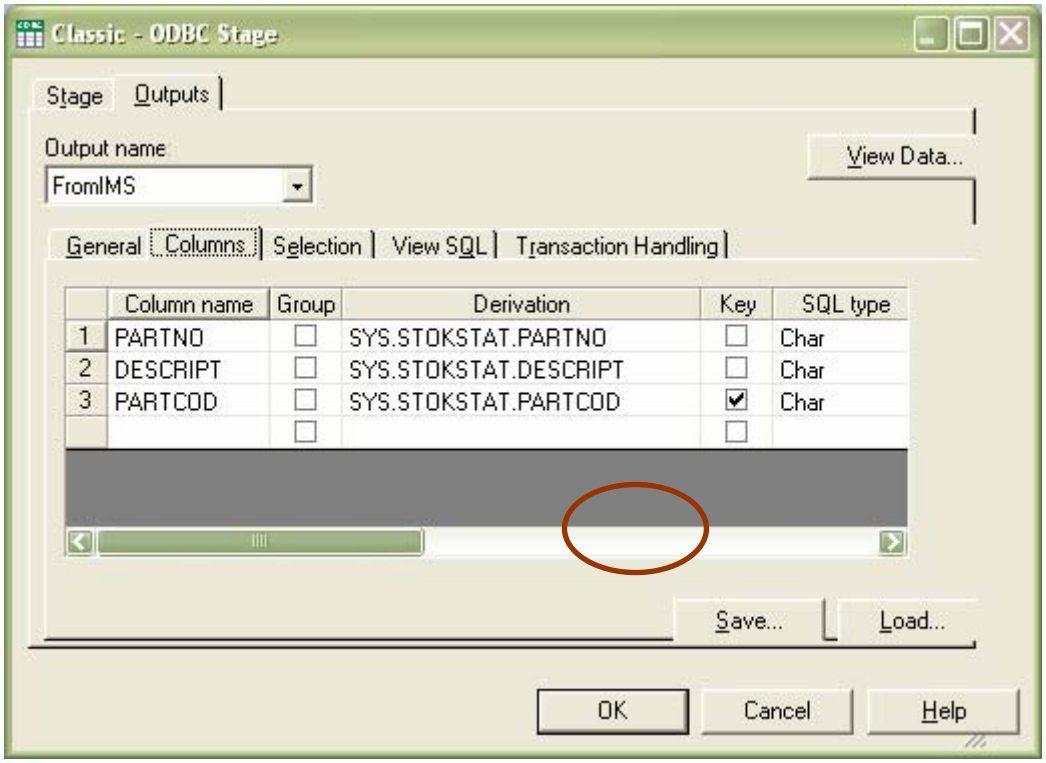
SOA for Mainframe data

Raj Datta
WebSphere IIS System z Solutions Architect
IBM Software Group

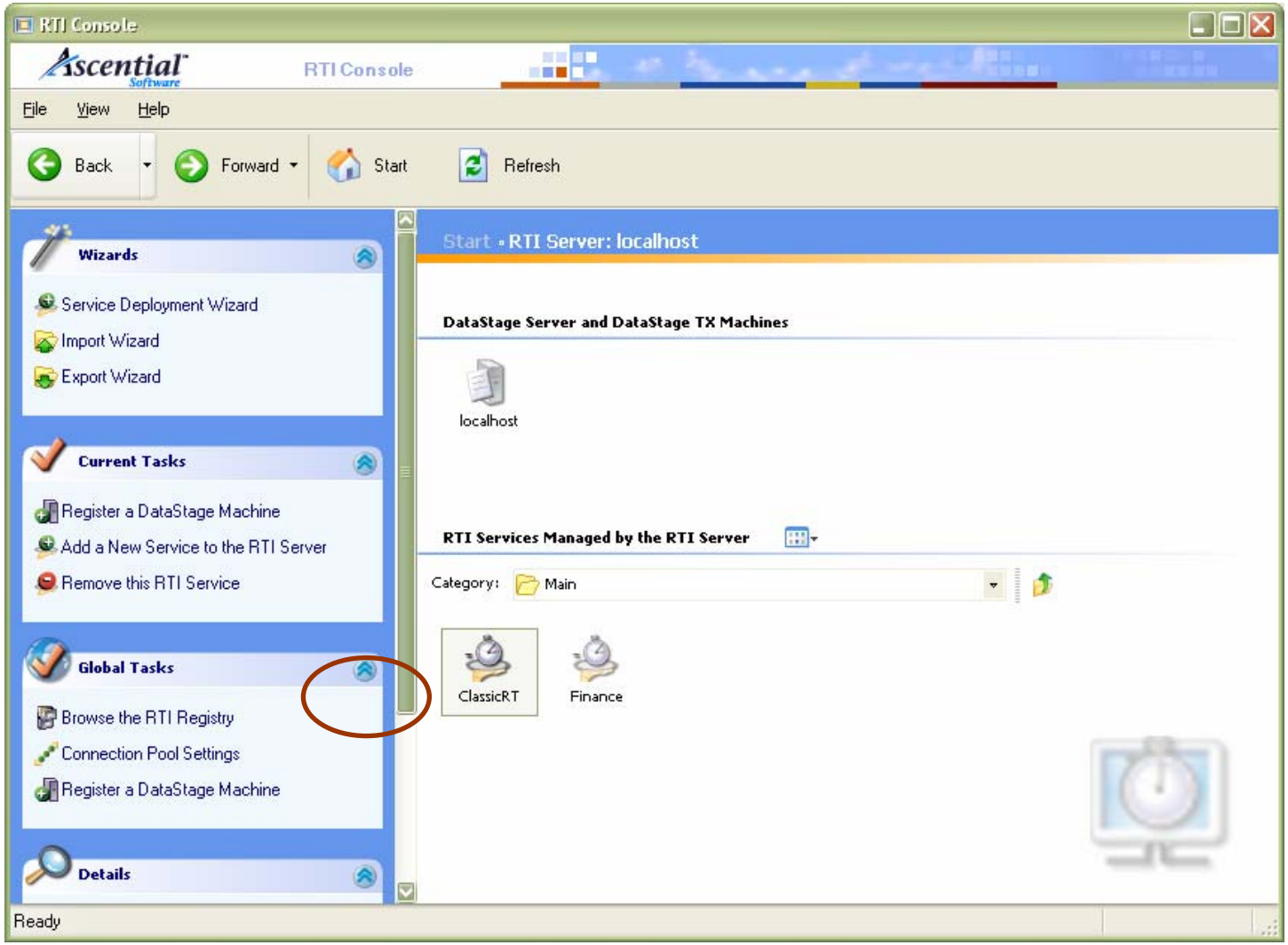
Create DataStage RTI job with RT Input/ Output



Use RT Input as criterion in SQL



Deploy DataStage RTI job as a service



Use a Web services test tool to invoke service

Mindreef SOAPscope - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:7103/tide/soapscope

Mindreef SOAPscope | Home | Workspace | Message | WSDL | Settings | About | Help

SOAPscope may not be configured correctly for your proxy. Click here for help...

Name	URL	Date
ClassicRT	http://127.0.0.1:9080/rti/wsdl/ClassicRT.wsdl	4/27
Finance	http://127.0.0.1:9080/rti/wsdl/Finance.wsdl	4/26

View | Properties | **Invoke** | Compare | Coverage | Analyze | WS-I Test

Choose | **Populate >** | Edit/Preview | Results | Go To Message

Send | Preview/Edit

Destination

HTTP Authentication

Ctrl+CLICK to include or exclude sibling elements

Message Envelope

[
]
ClassicRT
(
xs:string doc:partcod = 77
)

Documentation

service *Created On: 2006-04-27 05:50:27*
Modified On: 2006-04-27 05:50:27
operation *Demonstrates SOA for IMS data using WS II Classic Federation and WS DataStage RTI*

Send | Preview/Edit

Done

Voila! SOA for mainframe data!

Mindreef SOAPscope - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:7103/tide/soapscope

Mindreef SOAPscope | Home | Workspace | Message | WSDL | Settings | About | Help

SOAPscope may not be configured correctly for your proxy. Click here for help...

Name	URL	Date
ClassicRT	http://127.0.0.1:9080/rti/wsdl/ClassicRT.wsdl	4/27
Finance	http://127.0.0.1:9080/rti/wsdl/Finance.wsdl	4/26

View | Properties | Invoke | Compare | Coverage | Analyze | WS-I Test

Choose | Populate | Edit/Preview | Results > | Go To Message

Request — Pseudocode — HTTP Headers

```
ClassicRT
{
   partcod = 77
}
```

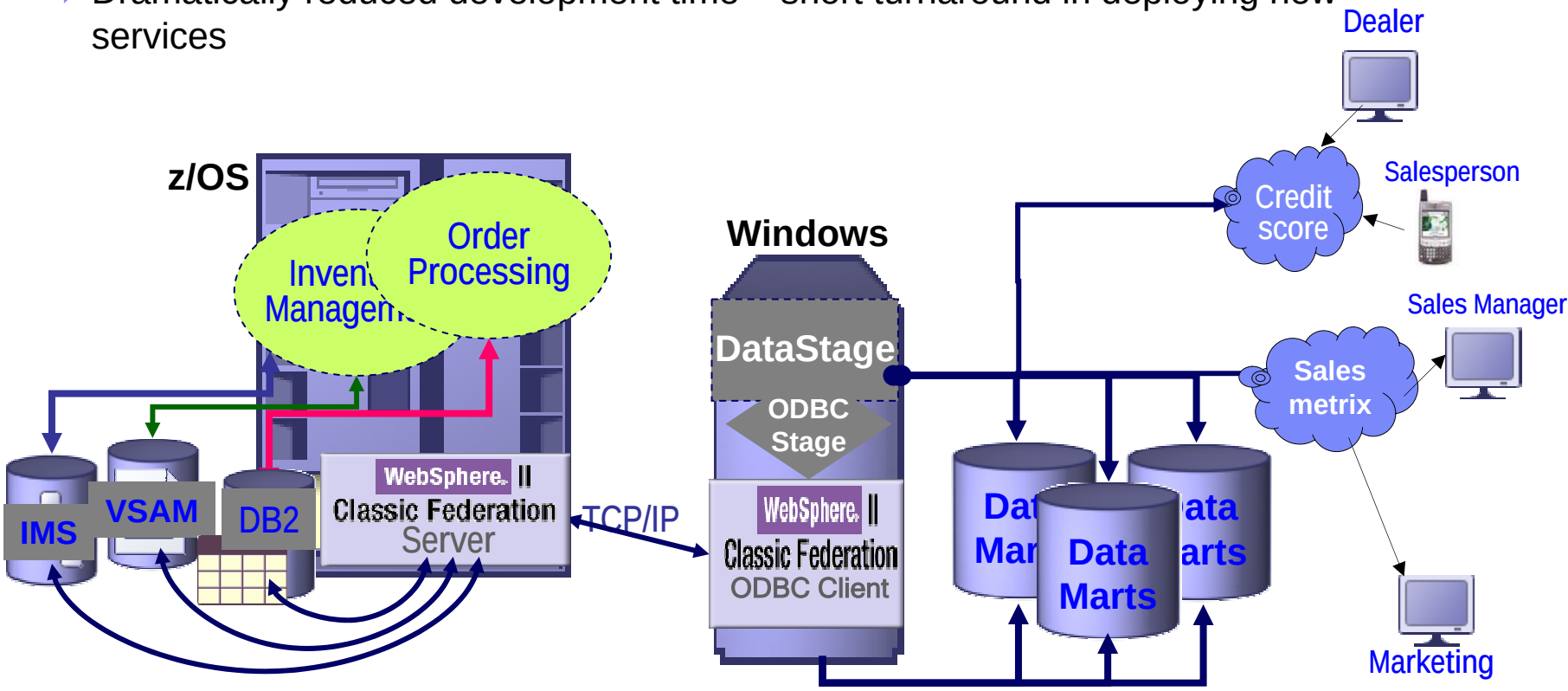
Response — Pseudocode — HTTP Headers

```
return
{
   partno = RAJDATTA1234567,
   descript = NEW YEAR 2002
}
```

Done

IBM solution – Web Services for Mainframe Data

- Dynamically connect web services tool with mainframe data
  - No dependence on mainframe development
  - One consistent process leverages WS tooling
  - Dramatically reduced development time – short turnaround in deploying new services



Classic Integration Scenarios

- Because the Classic Federation product appears as a relational database, supporting standard clients, and publishing in relational format, they can be used with a number of products.
- Our focus is IBM and Business Partner products:
  - IBM WebSphere Portal Server – JDBC
  - IBM WebSphere Studio - JDBC
  - IBM WebSphere Information Integrator – ODBC
  - IBM WebSphere Business Integrator – JDBC
  - IBM WebSphere Business Integrator Message Broker – ODBC
  - IBM WebSphere Data Stage – ODBC
  - Business Objects Data Integrator - ODBC
  - Business Objects Crystal Reports - ODBC
  - Sun SeeBeyond ICAN - JDBC
- But customers have used them with several other products: BEA Weblogic – JDBC, Microsoft Access, MSQuery, Visual .Net – ODBC, Cognos Impromtu
- As well as traditional applications

Usage scenarios

- e-Business
  - Deliver mainframe data to
    - Self-service portals (real-time account details)
    - e-commerce solutions (real-time inventory)
    - Employee portals (real-time claims detail)
  - Web developers become productive with no mainframe skills
  - Eliminates data latency business issues caused by copied data
- Business intelligence
  - Integrates seamlessly with
    - Reporting and analytical tools, e.g. Business Objects
    - Portals, e.g. WebSphere Portal
    - ETL, e.g. Ascential DataStage
- Scenarios
  - Empower self-service environments with key operational data - IVRs, Web sites, Portals, etc.
  - Feed operational data to business intelligence initiatives - Source for data marts, data warehouses, operational data stores
  - Direct, real-time mission critical information - Inventory-value, account-balance, available-credit, etc.

Value to the Business

- Extend the value of existing mainframe investments
  - Instant integration of mainframe assets into current business initiatives
  - Non-disruptive to existing applications and data environment
  - Reduces or eliminates redundant data and its costs
- Fits seamlessly into existing IT infrastructure out-of-the-box
  - Leverages SQL capabilities of modern tools
  - Works with mainframe infrastructure: security, accounting, monitoring, workload mgmt.
  - Reduces dependence on scarce mainframe skills
- Accelerate time-to-value of enterprise integration projects
  - No mainframe programming required
  - Transactional speed and enterprise scale
  - Easy to configure & maintain using its metadata-driven approach

IBM Software Group

Thank you

Raj Datta
WebSphere IIS System Z Solutions Architect
IBM Software Group