mainframe

IBM

Information Management

IBM Inter-University Programming Contest 2011 Training

Chapter 1a: Contest Briefing

Software Group, IBM

Agenda

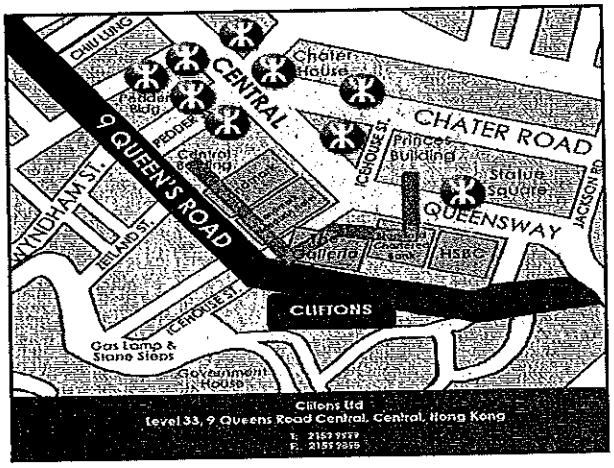
- Contest Rundown
- Overview of the contest technical scope
- Description on the contest PC setup and contest workflow
- Overview of the contest questions
- How to answer the questions

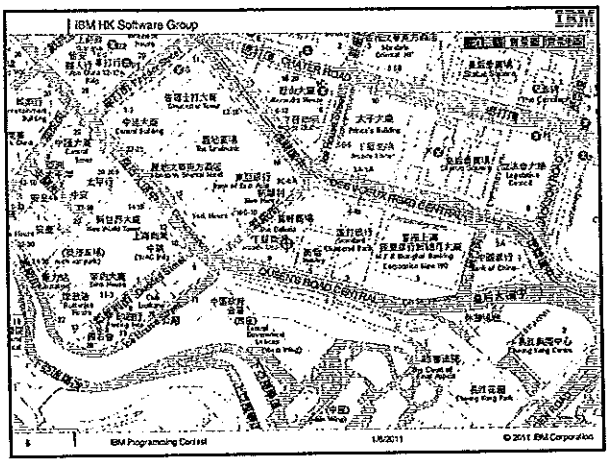
Contest Rundown

Date : Jan 29, 2011 (Saturday)

Time : 2:00 p.m. - 5:30 p.m.
(registration starts at 1:30 p.m.)

Venue : Cliftons Central Training Centre, Level 33, 9 Queen's Road Central, Hong Kong





IBM HK Software Group

Contest Rundown

1:30 - 2:00pm	Registration
2:00 - 2:15pm	Welcome Speech
2:15 – 4:15pm	Competition takes place
4:15 – 4:45pm	Marking
4:45 – 5:30pm	Award Presentation

6 | IBM Programming Contest | 1/6/2011 | © 2011 IBM Corporation

IBM HK Software Group

Prize

Champions	•HK$30,000[#] Cash Award •Champion Trophy •Three-month Summer Internship Program at IBM[@]
First runner ups	•HK$15,000[#] Cash Award •1st runner up Trophy
Second runner ups	•HK$ 9,000[#] Cash Award •2nd runner up Trophy
All Contestants	•IBM Trendy Backpack •IBM Participation Certificate
The Champion University[*]	•Champion Trophy

#The Cash Award is for the whole team.
* The one study with the most points of all that is sent.
@ Winners must be Hong Kong residents or have a "No Objection Letter" (NOL) issued by Immigration Department.

7 | IBM Programming Contest | 1/6/2011 | © 2011 IBM Corporation

IBM HK Software Group

Contest Rules (1)

- The contest has a duration of 2 hours.
- One PC and one terminal will be provided for each team.
- A set of questions will be given to each team to be completed within 2 hours.
- The contest is an open book contest. Contestants could bring in their reference books for the contest.
- Internet Access NOT allowed.
- Inter-Team Communication NOT allowed.
- ONLY Intra-Team Communication allowed.
- All the questions and related schema are provided to the students at the beginning of the contest. Question sheet is provided in hardcopy form.
- Once a team completes a question, the result is submitted to the judge. A time is recorded as the submission time. Then the judge will evaluate the submitted answer. In case the validation fails, the team will be notified and feedback will be provided to the team.

8 | IBM Programming Contest | 1/6/2011 | © 2011 IBM Corporation

IBM HK Software Group

Contest Rules (2)

- **Each team can have THREE chances to submit the answer of each question: (For programming questions only)**
  - Each time the team submits an answer, the judge records the submission time and runs a stored procedure to check the answer. If the answer is correct, they will be awarded points for the question. If the answer is incorrect, the contest system will send the answer back to the team and the team can then re-submit again up to three times.
  - Full points will be provided if the answer is correct.
  - Partial points might be given on programming questions.
  - For each question, the highest point obtained will be recorded as its final point.
- **After the contest is over, the team which gets the most points wins. If several teams get the same number of points, the one with the earliest last submission time wins.**
- **The decisions of the judges will be final.**
- **IBM reserves the right to publish the photographs and/or names of all or any prize winners in any of its promotional materials and/or advertisements as it sees fit.**

IBM HK Software Group

Technical Scope

- **Overview on IBM middleware**
- **DB2 basic operations on distributed platform and mainframe platform. For example:**
  - How to submit DDL and DML commands
  - How to interpret error messages
  - How to save results
- **Overview on graphical interfaces**
- **Overview on web programming using Java, DB2, WebSphere Application Server, Rational Application Developer and WebSphere Portal Server**
- **Overview on Tivoli Netcool Omnibus**

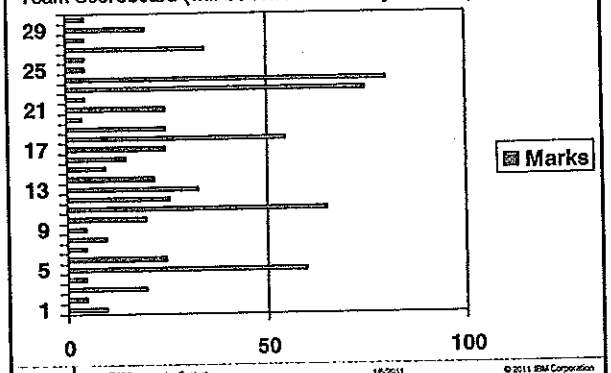
IBM HK Software Group

Contest PC Setup

- **Networked student PC and Instructor PC**
- **CPU : Intel Core 2 Quad Q9400 2.66GHz**
- **Ram : 4GB (2GB effective in VMware)**
- **Hard disk : 500GB HD**
- **One set of keyboard, mouse and a 17" TFT flat screen**

IBM HK Software Group

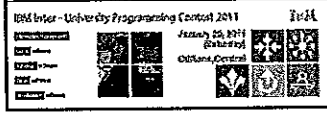
Team Scoreboard (will be refreshed every 15 mins)



Announcement

- **Announcement will be published to the projector and by voice.**

Question Overview



Two Types of Questions

- **Section 1 : Multiple Choices**
  - Choices of A, B, C and D
  - Can only submit all the answers ONCE
- **Section 2 : Programming Questions**
  - Need to perform programming work
  - Provide URL as the answer
  - Judge will check the correctness of the answer by running your program
  - Can submit answers up to three times

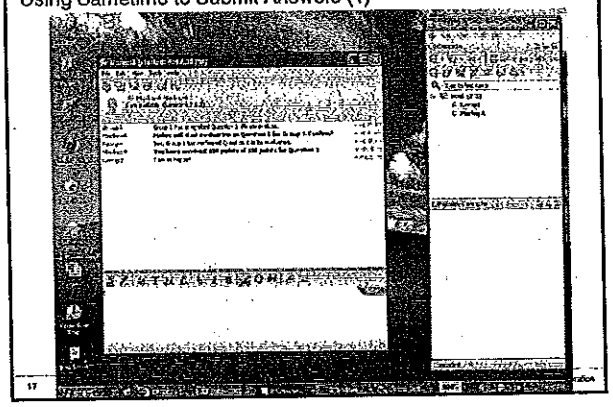
Distribution of Points

Question Number	Difficulty (* to *****)	Point
1.1	*	10
1.2	*	10
1.3	*	10
1.4	*	10
1.5	*	10
1.6	**	20
1.7	**	20
1.8	**	20
1.9	**	20
1.10	**	20
Section 1 To		150
2.1	***	80
2.2	*****	200
2.3	***	80
2.4	****	130
2.5	***	80
2.6	***	80
2.7	*****	200
Section 2 Total		850
Grand Total		1000

Sample Points

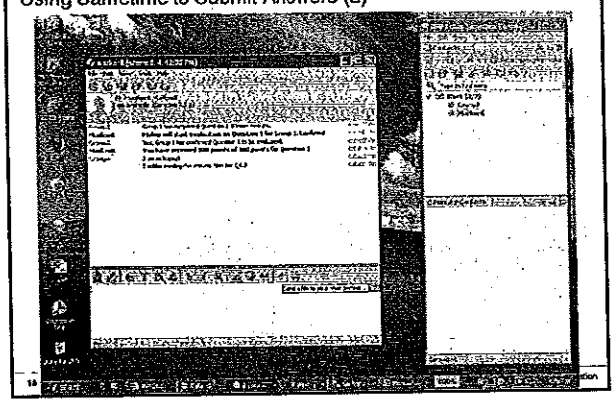
IBM HK Software Group

Using Sametime to Submit Answers (1)



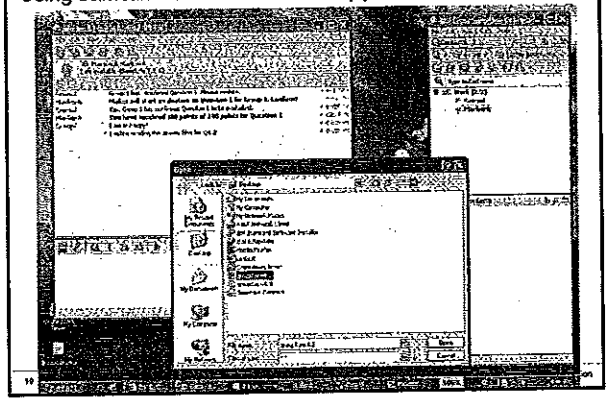
IBM HK Software Group

Using Sametime to Submit Answers (2)



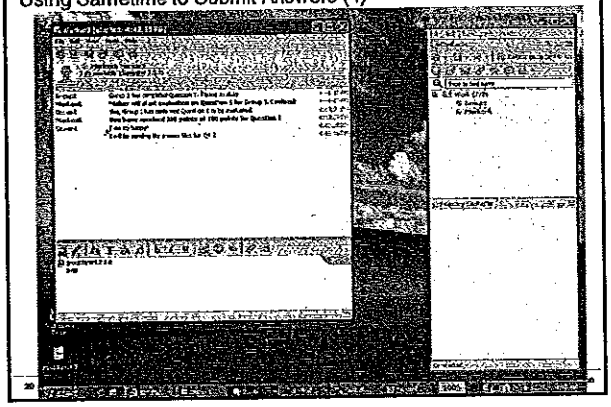
IBM HK Software Group

Using Sametime to Submit Answers (3)

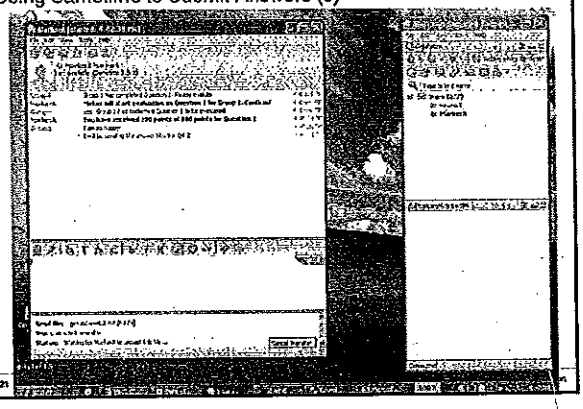


IBM HK Software Group

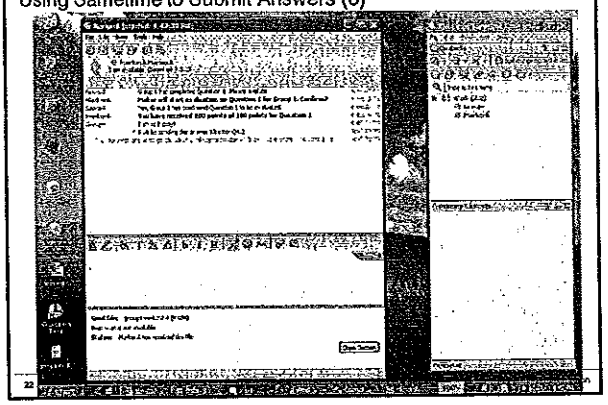
Using Sametime to Submit Answers (4)



Using Sametime to Submit Answers (5)



Using Sametime to Submit Answers (6)



Agenda

- 9:15am - 10:30am: Introduction, contest environment walkthrough and DB2 with LAB
- 10:30am - 11:30am: Tivoli with LAB
- 11:30am - 12:30pm: WAS with LAB
- 12:30pm - 1:30pm: Lunch
- 1:30pm - 2:30pm: Portal with LAB
- 2:30pm - 3: 30pm: Mainframe with LAB
- 3:30pm - 4:30pm: RAD with LAB
- 4:30pm - 5:30pm: Cognos with LAB

Contest PC Environment

- You have a chance to walk through the PC environment by running the lab today

Ready to Go?

IBM

Information Management

IBM Inter-University Programming Contest 2011 Training

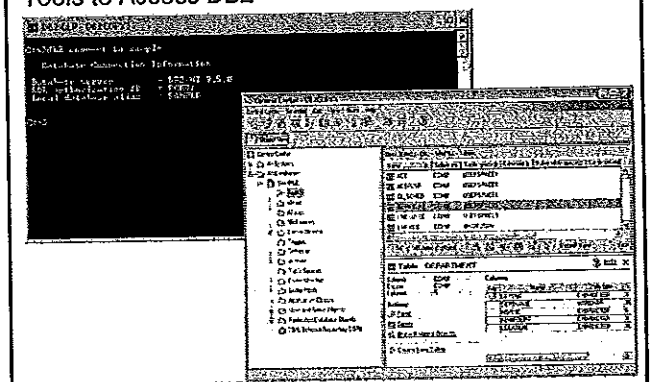
Chapter 1b: DB2 Operation Overview

Software Group, IBM

Chapter Outline

- Tools to Access DB2
- DB2 Instances and Databases
- Bufferpool
- Tablespace
- Schema
- Data Type
- Table
- Insert Statement
- Update Statement
- Delete Statement
- Select Statement
- Select Statement Options

Tools to Access DB2



Command Line Access

- DB2 Command Line Processor (CLP)
  - click on the DB2 Command Line Processor icon or enter db2 at a command prompt
  - an interactive input prompt appears:
    - db2 =>
  - no need to prefix commands with 'db2'
    - e.g. db2 => connect to sample
  - to end the interactive mode, enter quit
  - to disconnect from the database and terminate the DB2 backend process (db2bp), enter terminate
  - to execute OS commands, enter !<OS command>
- DB2 Command Line Window
  - click on the DB2 Command Window icon or from MS-DOS prompt, enter db2cmd
    - The command db2cmd is valid for Windows only
  - invoke the interpreter by prefacing commands and SQL with 'db2'
    - e.g. db2 connect to sample
    - e.g. db2 "select * from employee"
    - e.g. db2 -tvf createtab.db2
  - to end command line mode and terminate the DB2 backend process (db2bp), enter db2 terminate
  - all OS commands can be issued from the DB2 Command Window

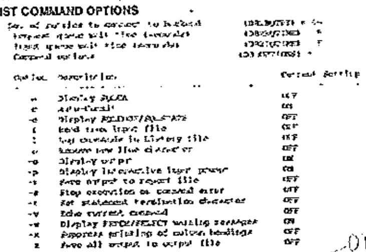
Getting Help

- Obtain syntax and information for DB2 commands from the command line
  - db2 "?" list of all DB2 commands
  - db2 "? <db2-command>" get syntax help for a specific command
  - db2 "? sqlnnnn" get message and brief description of a specific SQLCODE
  - db2 "? db2nnnn" get message and brief description of a DB2 error code
- For example:
  - db2 "? catalog tcpip"

```
CATALOG [ADMIN] TCPIP NODE node-name REMOTE hostname
   [SERVER service-name] [SECURITY {SOCKS}]
   [REMOTE_INSTANCE instance-name] [SYSTEM system-name]
   [OSTYPE os-type] [WITH "comment string"]
```

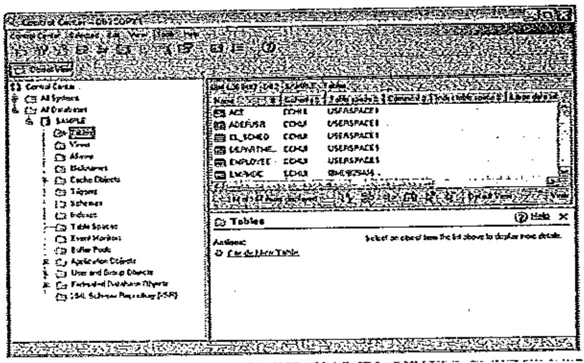

Command Line Options

- To list default command options
  - LIST COMMAND OPTIONS

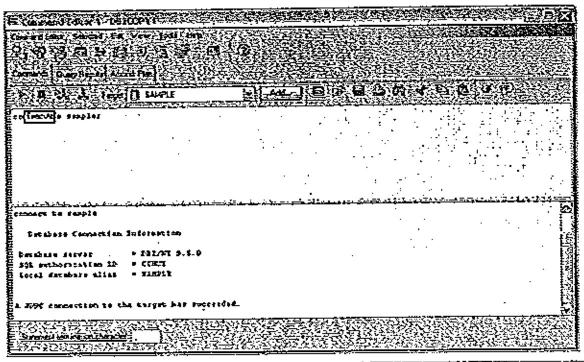


- To change default command options.
  - For example, to turn auto-commit off: set db2options=+c
- To change command option for the current session
  - For example, to save output to file: UPDATE COMMAND OPTIONS USING z ON output.txt

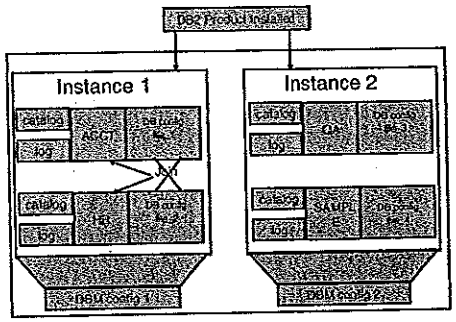
Control Center



Command Center



DB2 Instances and Databases



DB2 Instances

- A DB2 instance is a logical context in which DB2 commands and functions are executed
- An instance manages access to database files
- More than one instance can be defined on a server machine
- Each instance is independent of the others
- An instance is created on install by default
  - db2inst1 (UNIX)
  - DB2 (Windows)
- To start an instance
  - db2start
- Starts a remote instance
  - db2start REMOTE <instance name>
- Start the instance in quiesced mode for administration purposes
  - db2start ADMIN MODE
- Terminate all database connections and stop an instance
  - db2stop force

Managing DB2 Instances

- To create additional instances
  - db2icrt -u <fenced user ID> <instance name>
    - must specify fenced user ID for UNIX platforms
  - db2icrt <instance name>
    - for Windows platforms
- Terminate all database connections and drop an instance
  - db2idrop -f <instance name>
- To list existing instances defined in a server
  - db2ilist
- To update a DB2 instance for access to functions associated with installation or removal of certain product options or fix patches
  - db2iupdt <instance name>
- To migrates an existing instance, for UNIX platforms only
  - db2imigr <instance name>
- To create, drop, update, or migrate an instance, root or administrative access is required

Create a Database

- Simplest format

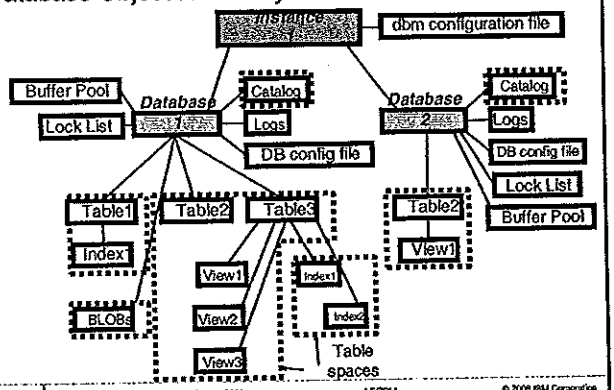
```
CREATE DATABASE database-name
```

- More options:

```
CREATE DATABASE database-name
  [AUTOMATIC STORAGE [NO | YES]]
  [ON drive[,drive]...][DBPATH ON drive]]
  [ALIAS database-alias]
  [USING CODESET codeset TERRITORY territory]
  [PAGESIZE integer [K]]
  [CATALOG TABLESPACE tblspace-defn]
  [USER TABLESPACE tblspace-defn]
  [TEMPORARY TABLESPACE tblspace-defn]
  [AUTOCONFIGURE [USING config-keyword value [{,config-keyword value}...]]
```

Information Management | IBM

Database Object Hierarchy



Information Management | IBM

Bufferpool

- Used to buffer data in memory to reduce the number of I/O operations to the physical database
- Keep often requested data/index pages in memory
- Keep infrequently accessed tables (e.g. random access into very large table) out of main memory
- Most data manipulation takes place in buffer pools, except for large objects and long field data
- IBMDEFAULTBP is the default bufferpool created with every database

Information Management | IBM

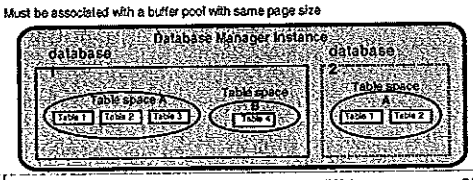
Bufferpool Management

- Memory for the buffer pool is allocated when the database is activated or when the first application connects to the database
- Command to create buffer pool
  - CREATE BUFFERPOOL bpname IMMEDIATE SIZE sz PAGESIZE pgsz
  - CREATE BUFFERPOOL bpname DEFERRED SIZE sz PAGESIZE pgsz
- IMMEDIATE
  - Bufferpool be created immediately
- DEFERRED
  - Bufferpool be created when the database is reactivated
- Information stored in SYSIBM.SYSBUFFERPOOLS
- Alter size of a bufferpool
  - ALTER BUFFERPOOL bpname IMMEDIATE SIZE sz
  - ALTER BUFFERPOOL bpname DEFERRED SIZE sz
- Memory is allocated as soon as if the memory is available.
- If the memory is not available, the changed occurs when the database is reactivated
- If the size of the bufferpool is decreased, memory is deallocated at commit time

Information Management | IBM

Tablespaces

- Table space is a logical grouping of tables and other objects in a database
- All database objects are stored within table spaces
- Two types of storage:
  - System Managed Space (SMS)
  - Database Managed Space (DMS)
- A table space is composed of one or more containers
- Data allocated by extents within containers
- Table spaces are either 4K, 8K, 16K or 32K pages
  - 4K is default size
  - Must be associated with a buffer pool with same page size



user account → database name → a.t1

Default Tablespaces in a Database

- With a simple CREATE DATABASE command:
  - CREATE DATABASE sample
- Three SMS table spaces are created automatically in default locations:
  - SYSCATSPACE - system catalog tables
  - USERSPACE1 - default user data
  - TEMPSPACE1 - temporary data
- Can change table space storage type and explicitly specify the locations of the containers:

```
CREATE DATABASE sample
  CATALOG TABLESPACE
    MANAGED BY SYSTEM
    USING ( 'c:\catdir1' )
  USER TABLESPACE
    MANAGED BY DATABASE
    USING ( FILE 'c:\db2files\usertbsp1 100, FILE 'c:\db2files\usertbsp2 100 )
  TEMP TABLESPACE
    MANAGED BY SYSTEM
    USING ( 'c:\tempspace' ) ;
```

usrspace1

tempspace1

Schema

- A schema is a collection of database objects such as tables, views, indexes, or triggers. It provides a logical classification of database objects
- How is a Schema name used?
  - Fully-qualified object name: "schemaname.tablename"
  - Can have multiple tables with the same name, but different schema names
    - eyerman.staff != jones.staff
- Reserved schema names
  - SYSCAT, SYSIBM, SYSSTAT, SYSFUN
  - Avoid schema names beginning with SYS
- If database object does not specify a schema name, table qualified with current authorization ID
- Alternate schema names can be specified using
  - SET CURRENT SCHEMA or SET CURRENT SQLID command
  - CREATE ALIAS <aliasname> FOR <tab/view name>
  - CREATE VIEW

System Schemas

- Created with every database and placed into the SYSCATSPACE table space
- SYSIBM
  - Base catalogs
  - Access not recommended
- SYSCAT
  - SELECT authority GRANTed to PUBLIC
  - Catalog Read-only Views
  - Recommended way to obtain catalog information
- SYSSTAT
  - Updateable Catalog Views - Influence the Optimizer
- SYSFUN
  - User-Defined Functions

System Catalog Tables and Views

Schema: Table = SYSIBM View = SYSCAT

TABLE	VIEW	DESCRIPTION
SYSDBAUTH	DBAUTH	Authorities on database
SYSCHECKS	CHECKS	Check constraints
SYSCOLUMNS	COLUMNS	Column definitions
SYSCOLCHECKS	COLCHECKS	Columns referenced by check constraints
SYSCOLDIST	COLDIST	Detailed columns statistics
SYSKEYCOLUSE	KEYCOLUSE	Columns used in keys
SYSCONSTDEP	CONSTDEP	Constraint dependencies
SYSDATATYPES	DATATYPES	Datatype definitions (built-in & UDT)
SYSEVENTMONITORS	EVENTMONITORS	Event Monitor Definitions
SYSEVENTS	EVENTS	Events currently monitored
SYSFUNCPARMS	FUNCPARMS	Definitions of Parameters/Results of UDFs
SYSFUNCTIONS	FUNCTIONS	UDF definitions
SYSINDEXAUTH	INDEXAUTH	Index privileges
SYSINDEXES	INDEXES	Index definitions

Information Management

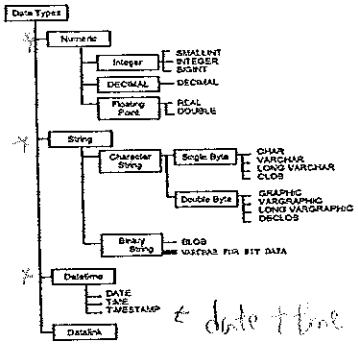
System Catalog Tables and Views

Schema: Table = SYSIBM View = SYSCAT

TABLE	VIEW	DESCRIPTION
SYSPACKAGEAUT	PACKAGEAUTH	Authorities on packages
SYSPACKAGEDEP	PACKAGEDEP	Package dependencies
SYSPACKAGES	PACKAGES	Package definitions
SYSREFERENCES	REFERENCES	Referential constraints definitions
SYSSTATEMENTS	STATEMENTS	Details of package SQL Statements
SYSTABAUTH	TABAUTH	Table Authorities
SYSTABCONST	TABCONST	Table constraint definitions
SYSTABLES	TABLES	Table definitions
SYSTABLESPACES	TABLESPACES	Table Space Definitions
SYSTRIGDEP	TRIGDEPEVENT	Trigger dependencies
SYSTRIGGERS	TRIGGERS	Definitions of triggers
SYSVIEWDEP	VIEWDEP	View dependencies
SYSVIEWS	VIEWS	View definitions

Information Management

Data Types



Information Management

CREATE TABLE Command

- You must have SYSADM or DBADM authority or CREATETAB privilege on the database
- Example:

```
connect to sddb;
create table artists
( artno      SMALLINT NOT NULL,
  name       VARCHAR(50) WITH DEFAULT 'abc',
  classification  CHAR(1) NOT NULL,
  bio        CLOB(100K) LOGGED,
  article    DATALINK LINKTYPE URL FILE
             LINK CONTROL MODE DB2OPTIONS,
  picture    BLOB(2M) NOT LOGGED COMPACT )
  INDEX IN indbsp
  LONG IN longtbsp
  IN datatbsp;
```

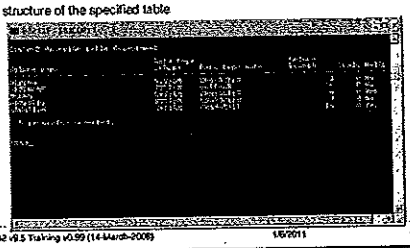

Information Management

Table

- If a table is created without the IN clause, it will be placed:
  - In the IBMDEFAULTGROUP table space (if it exists and if the page size is sufficient)
  - In a user created table space which is of the smallest pagesize that is sufficient for the table
  - Then it will go in USERSPACE1 (if it exists and has a sufficient page size)
- Use IN, INDEX IN, and LONG IN clauses to specify the tablespaces to stored related information
- To duplicate a table:
  - CREATE TABLE tab1new LIKE tab1;
- No constraints, triggers, or indexes copied, and data not copied
- May specify table or view
- To define a table using SQL statement
  - CREATE TABLE t1new
    AS (SELECT c1, c8, c10 FROM t1)
    DEFINITION ONLY;
- Column attributes of defined table based upon referenced table
- Data not populated

Other Useful Table Commands

- LIST TABLES
  - List tables for the current user
- LIST TABLES FOR ALL
  - List all tables defined in the database
- LIST TABLES FOR SCHEMA <schema>
  - List tables for the specified schema
- DESCRIBE TABLE <tablename>
  - Show the structure of the specified table



Insert Statement

- You need to have appropriate table or view privilege
- Can insert one or more rows at a time

```
INSERT INTO department (deptno, deptname, admrdept)
  VALUES ( 'E31', 'ARCHITECTURE', 'E01' ),
         ( 'E32', 'INFRASTRUCTURE', 'E02')
```

- You can use subselect to determine values

```
INSERT INTO department (deptno, deptname)
  SELECT deptno, deptname FROM sales_depts
```

- May also be done to a subset of columns provided:
  - A column not specified accepts NULLs or
  - Columns defined WITH DEFAULT
- Large amounts of data? Look at using LOAD instead

Update Statement

- UPDATES come in 3 main varieties :
  - Full table

```
UPDATE employee SET salary = NULL
UPDATE employee SET (dept) = (
        SELECT deptname FROM department WHERE deptno = 1)
```

  - Searched with a **WHERE** clause

```
UPDATE employee SET salary = salary * 1.10 WHERE title = 'DBA'
```

  - Positioned using a **CURSOR** in a program

```
EXEC SQL DECLARE C1 CURSOR FOR
      SELECT * FROM EMPLOYEE FOR UPDATE OF JOB;
EXEC SQL OPEN C1;
EXEC SQL FETCH C1 INTO ... ;
if ( strcmp (change, "YES") == 0 )
  EXEC SQL UPDATE EMPLOYEE
       SET JOB = :newjob
       WHERE CURRENT OF C1;
EXEC SQL CLOSE C1;
end if
```


Delete Statement

- DELETES can apply to single or multiple rows
  - You can also use subselect to determine values
- Two forms of statement :
  - Searched DELETE form is used to delete one or more rows
    - Optionally determined by a search condition
  - Positioned DELETE form is used to delete exactly one row
    - Determined by the current position of a cursor
- Deleted rows not removed from table
  - Space is marked as unused
  - Reclaim space using the REORG command
- Example:

```
DELETE FROM DEPARTMENT WHERE DEPTNO = 'D11'
```

Select Statement

- SELECT is the QUERY element of the language
- Queries come in many styles:
  - Retrieving all table data
  - Limiting Columns or Rows retrieved
  - Cartesian Product
  - Inner Join
  - Set operators
  - Common table expressions
  - Other options:
    - derived columns
    - sub-query etc.
    - sorting
    - using functions
    - grouping values

ORDER BY Clauses

- ORDER BY clause
  - The ORDER BY clause specifies an ordering of the rows of the result table
  - If a single sort specification (one sort-key with associated direction) is identified, the rows are ordered by the values of that sort specification
  - If more than one sort specification is identified, the rows are ordered by the values of the first identified sort specification, then by the values of the second identified sort specification, and so on
  - Example:

```
SELECT c1, c2 FROM t1 ORDER BY c1;
```


GROUP BY, HAVING Clauses

- GROUP BY ... <grouping expression>
  - A GROUP BY clause contains a grouping expression, it specifies an intermediate result table that consists of a grouping of the rows of R. R is the result of the previous clause of the subselect.

```
SELECT workdept, MAX ( salary ) FROM employee GROUP BY workdept ;
```

- HAVING clause
  - The HAVING clause specifies an intermediate result table that consists of the groups of R for which the search-condition is true. R is the result of the previous clause of the subselect.

```
SELECT workdept, MAX ( salary ) FROM employee GROUP BY workdept
  HAVING MAX ( salary ) < ( SELECT avg ( salary FROM employee ) ;
```

- Obtain distinct value from a table

```
SELECT DISTINCT ( position ) FROM employee ;
```

- Use of LIKE

```
SELECT DISTINCT WORKDEPT FROM employee WHERE WORKDEPT LIKE 'A%';
```

- Use of EXISTS

```
SELECT empno, salary FROM employee
  WHERE NOT EXISTS ( SELECT empno FROM manager ) ;
```

- Use of IN, NOT IN clauses

```
SELECT empno, salary FROM employee
  WHERE department IN ( 'A00', 'B00', 'C00' ) ;
```

Chapter 1: Contest Environment Walkthrough

Objectives

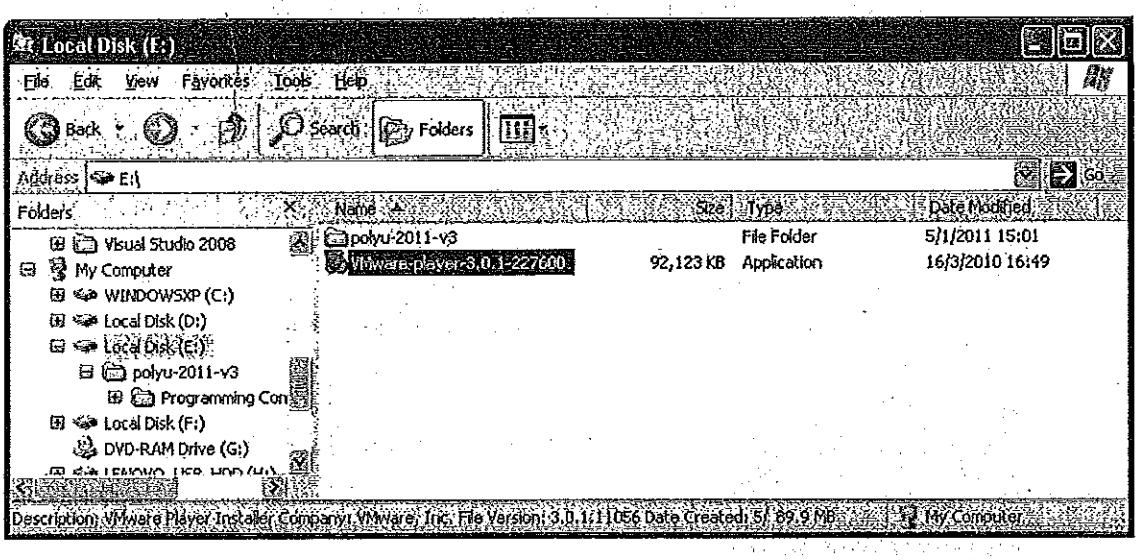
In this exercise, we will learn:

- How to configure your environment
- How to boot up the programming environment using VMware
- How to log in and start up necessary services and system software
- Examine the sample application which will be used in the programming contest
- How to perform programming using the GUI environment
- How to use DB2 in command line mode and GUI mode

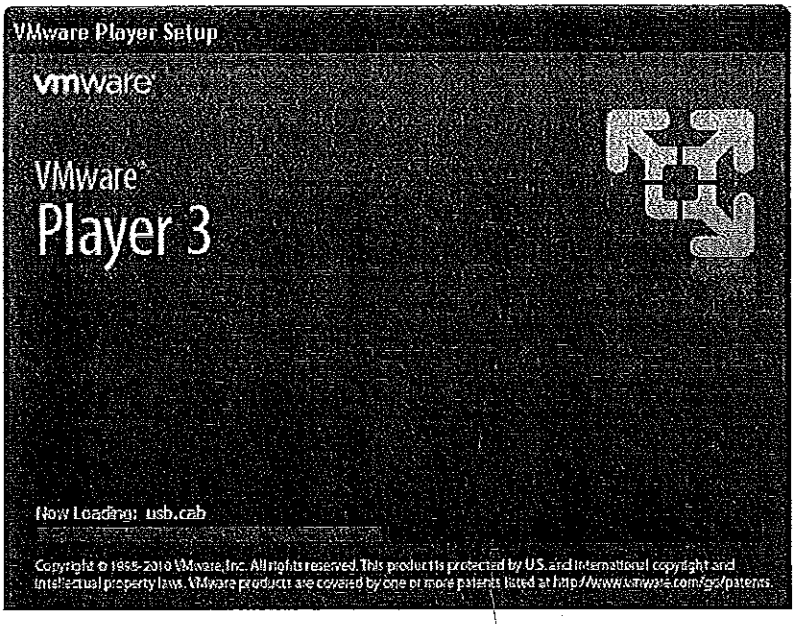
Exercises

1. Log into the PC using the username and password provided by instructor.

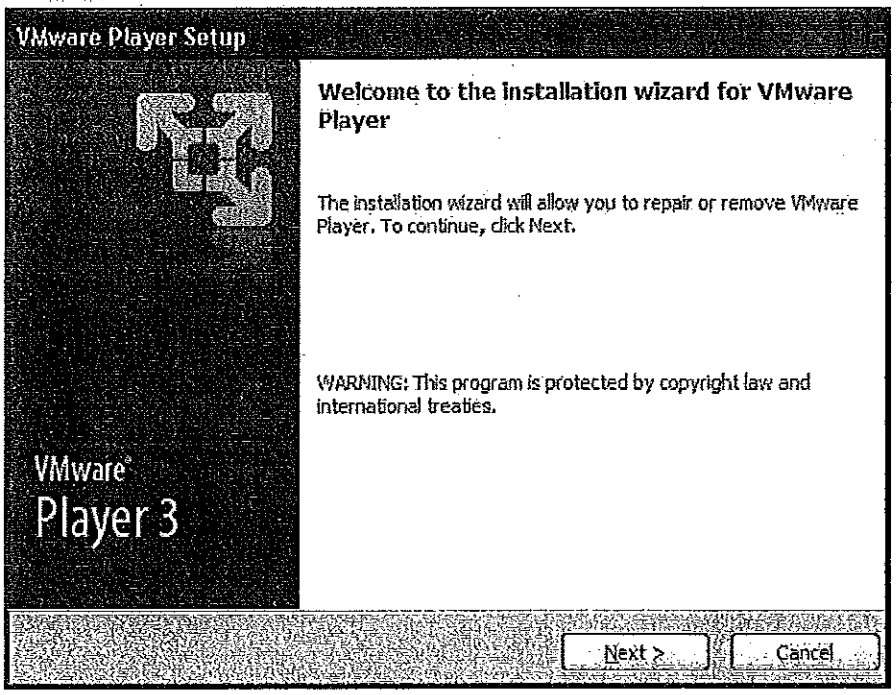
2. Open an Explorer and then navigate to E:\



3. Select and double click the program VMware-player-3.0.1-227600 to install the VMware player. The following screen is displayed:

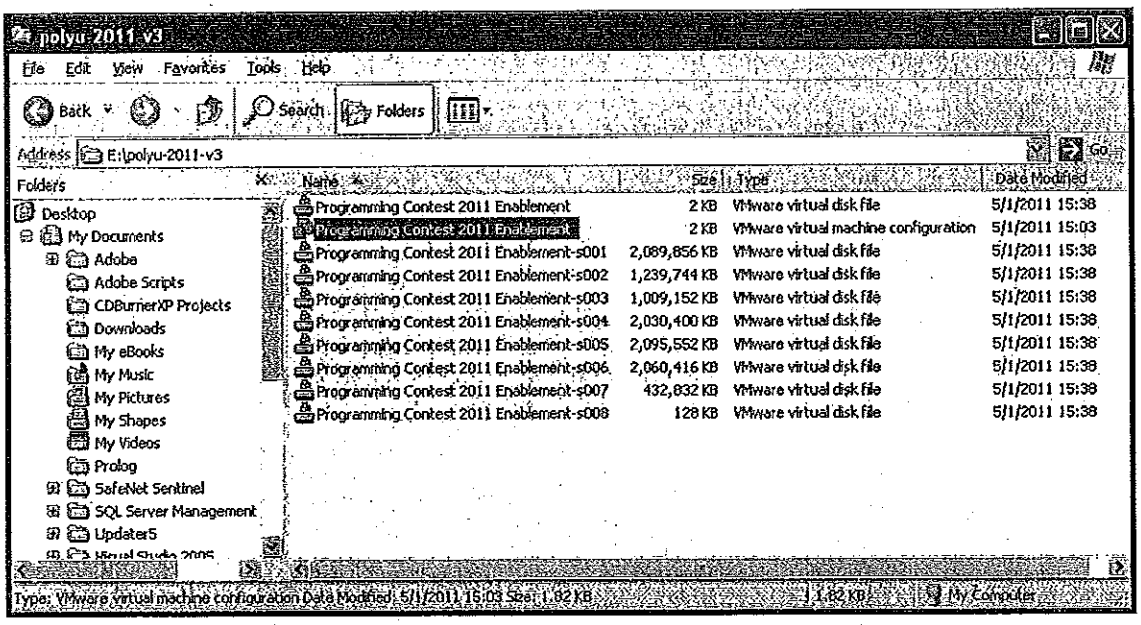


4. After a while the following screen is displayed:



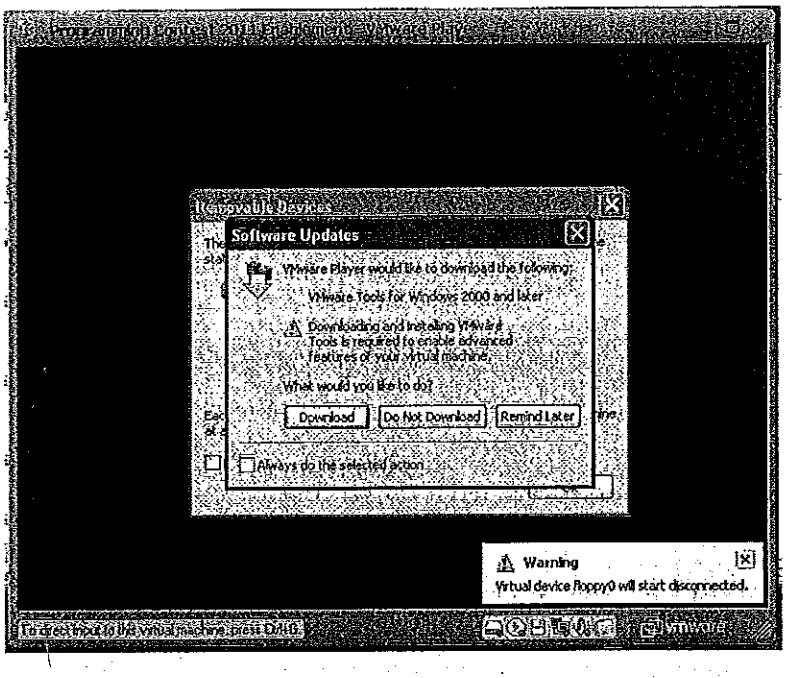
Keep pressing Next and adopting the default choice until VMware player is installed successfully.

5. Using the Explorer, navigate to the folder E:\polyu-2011-v3 and the double click the VMware virtual machine configuration file with name "Programming Contest 2011 Enablement" and then boot up the VMware.

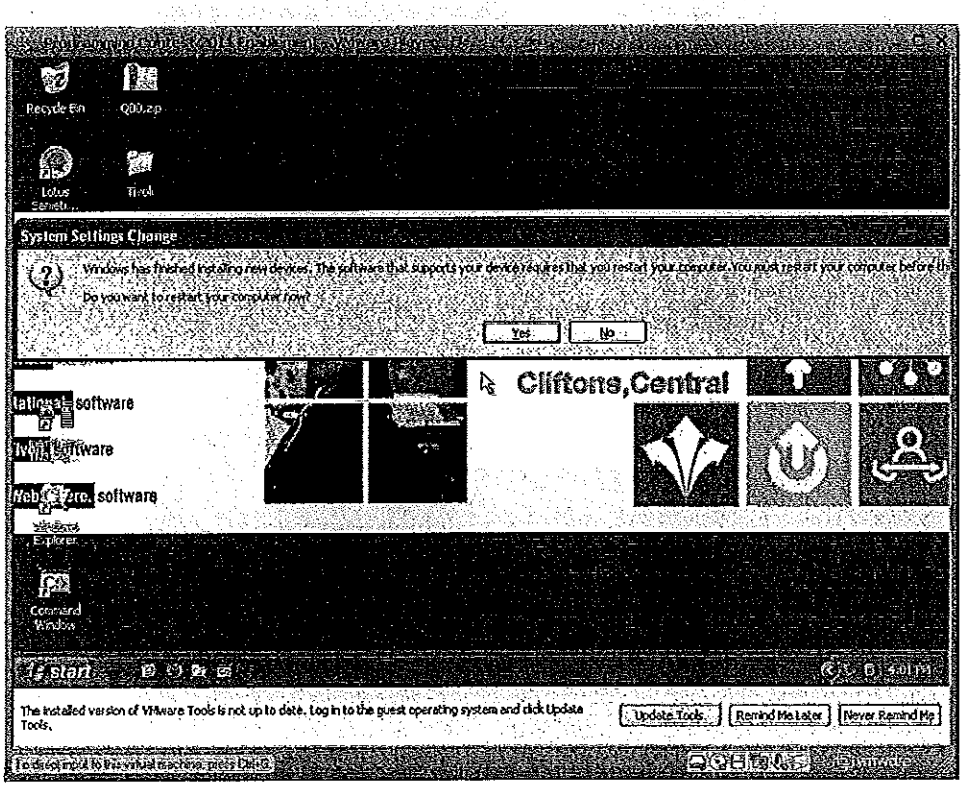


6. The VMware is now being booted up. You might see a pop up window telling you floppy0 is not available. Please "No" and then proceed:

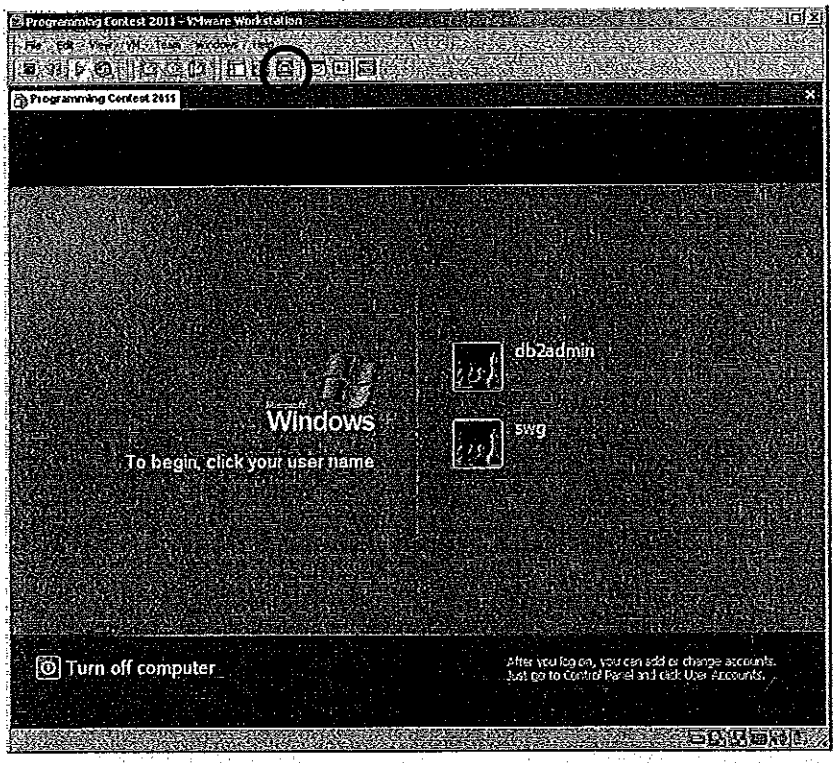
7. During boot up, you might see the following screen asking you whether you want to download the VMware Tools update. Press the button "Do Not Download" and proceed:



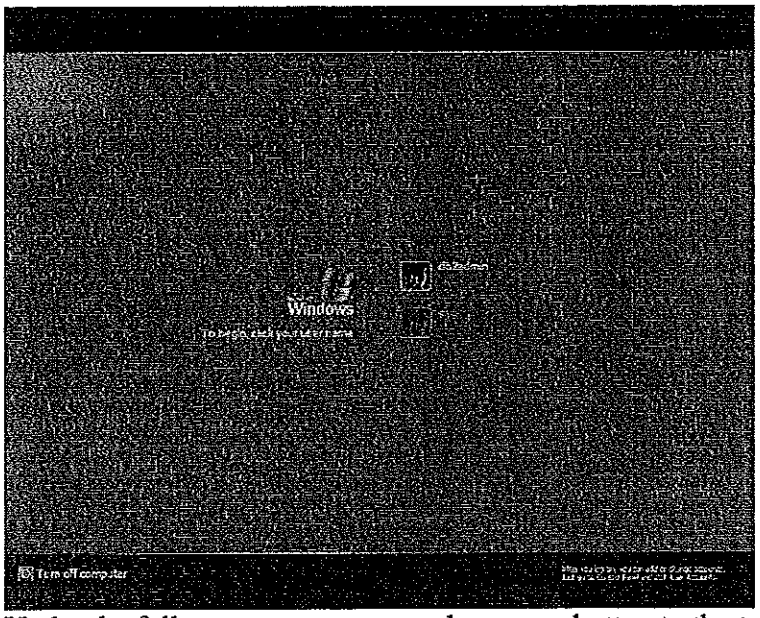
8. Lastly, you might see a pop up window asking you to restart the VMware, press the button "No" and then proceed:



9. When the Windows XP is booted up, the following login screen appears:

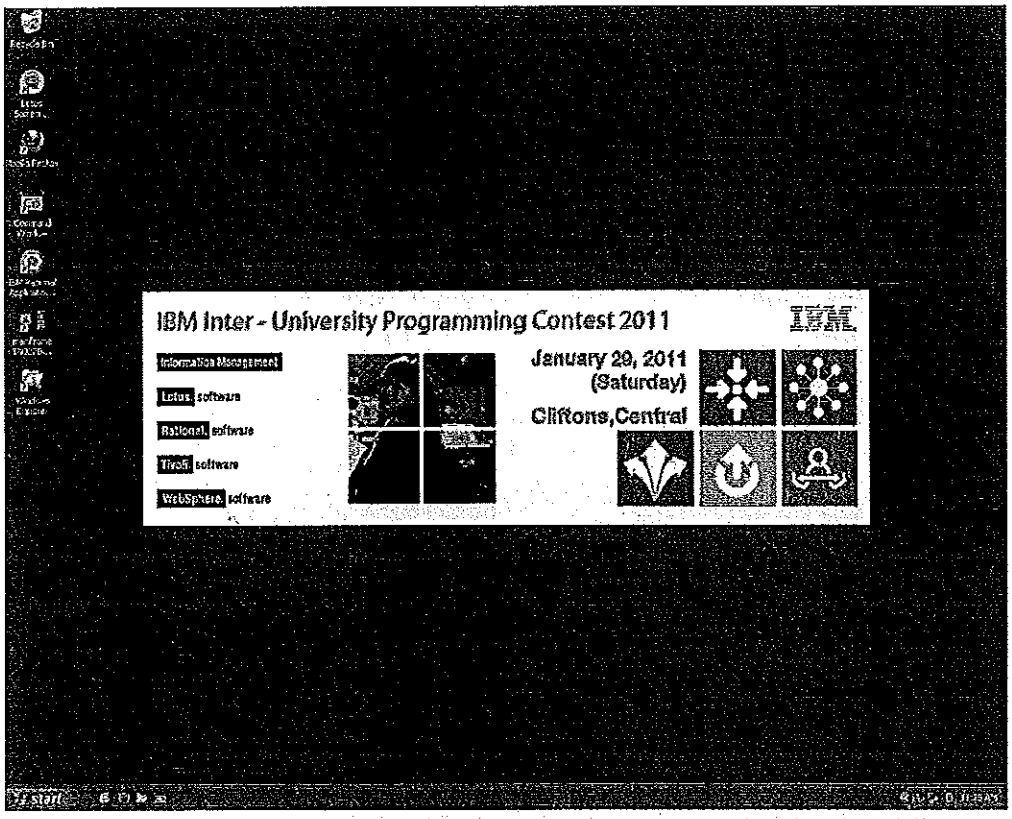


It is possible to click the full screen button (the icon within the black circle) to switch the booted system to full screen. This is shown as the following screen:

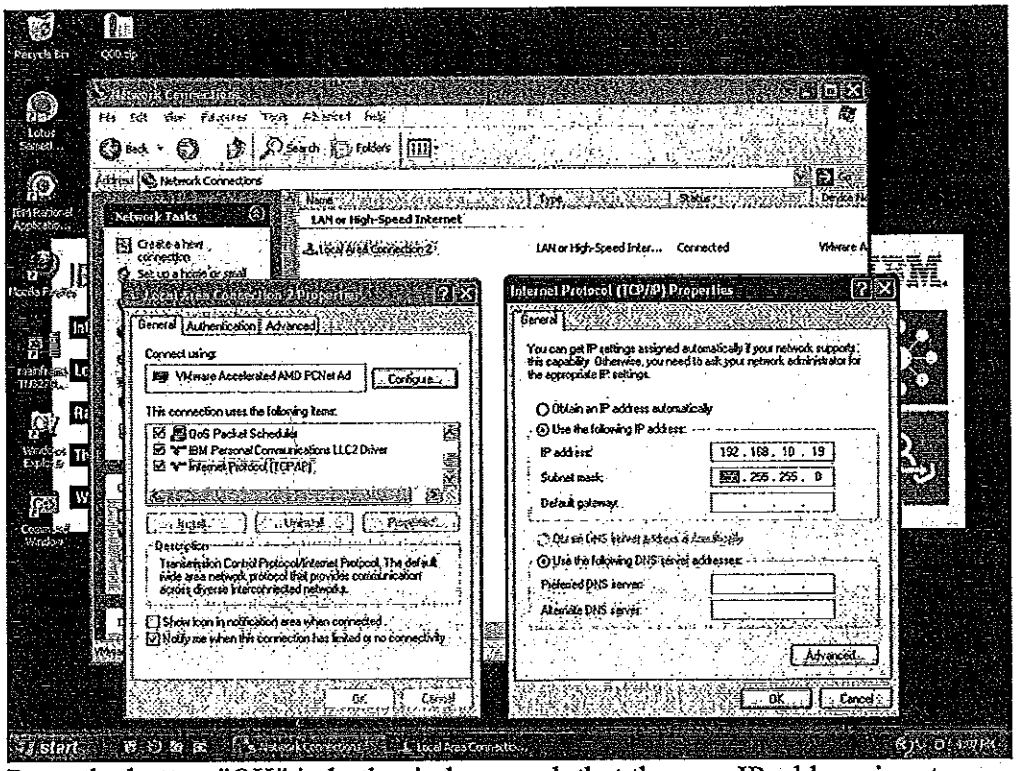


Under the full screen, you can put the mouse button to the top of the screen to activate the VMware manual bar, and then you can go back to windowed mode by pressing the corresponding buttons in the manual bar.

10. Click "db2admin" and enter the password "password" to log into the programming contest environment. The following screen is displayed:

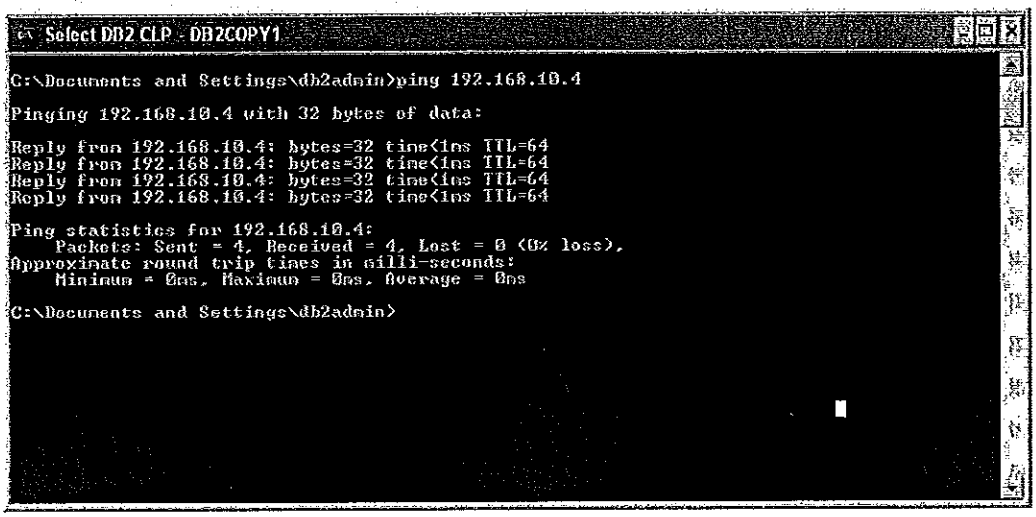


11. Open Control Panel, select Network Connections, and then bring up the LAN properties window. Then select "Internet Protocol (TCP/IP)" then press the button "Properties". The window "Internet Protocol (TCP/IP) Properties" is displayed. Select "Use the following IP address" and then enter the IP address provided when you were enrolled into the class. Use the Subnet mask "255.255.255.0".



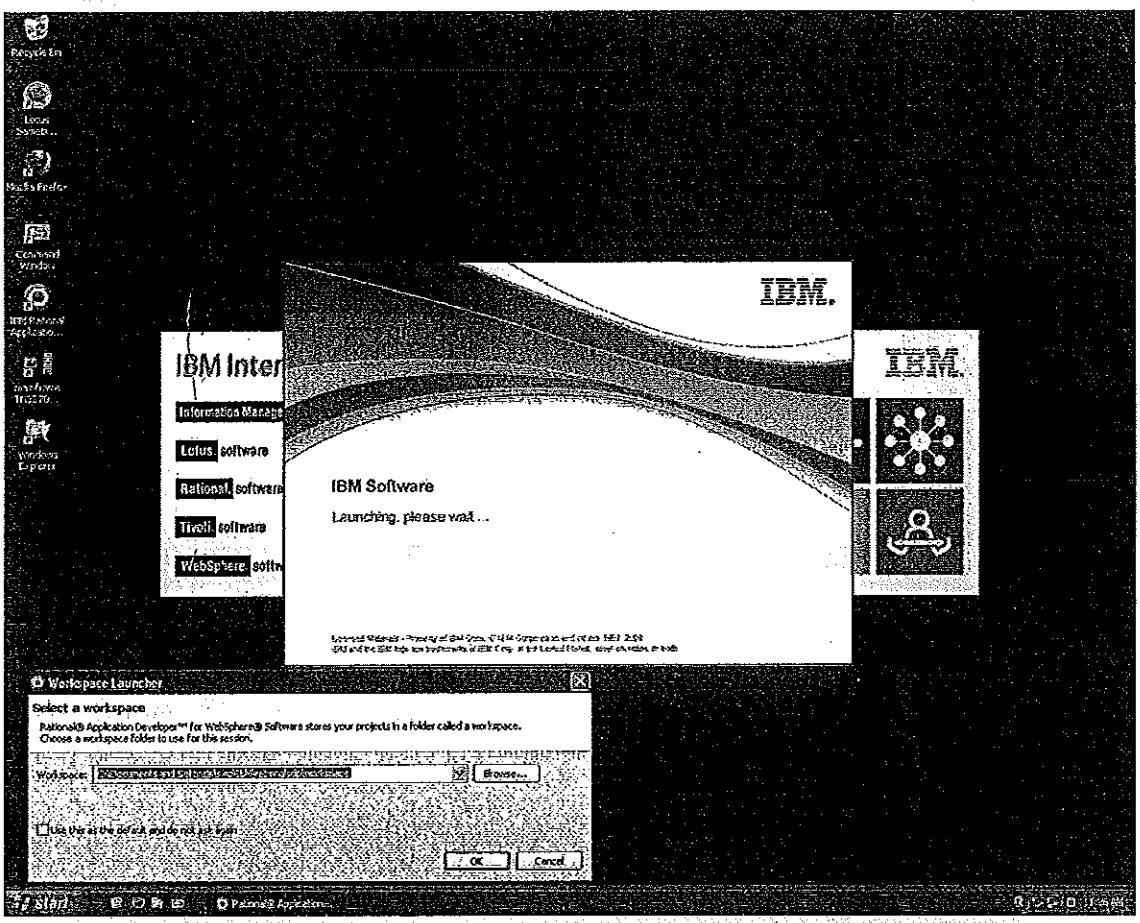
Press the button "OK" in both windows such that the new IP address is set up effectively.

12. You can bring up a command window and the run the command "ping 192.168.10.4" to verify the IP address is set up properly.



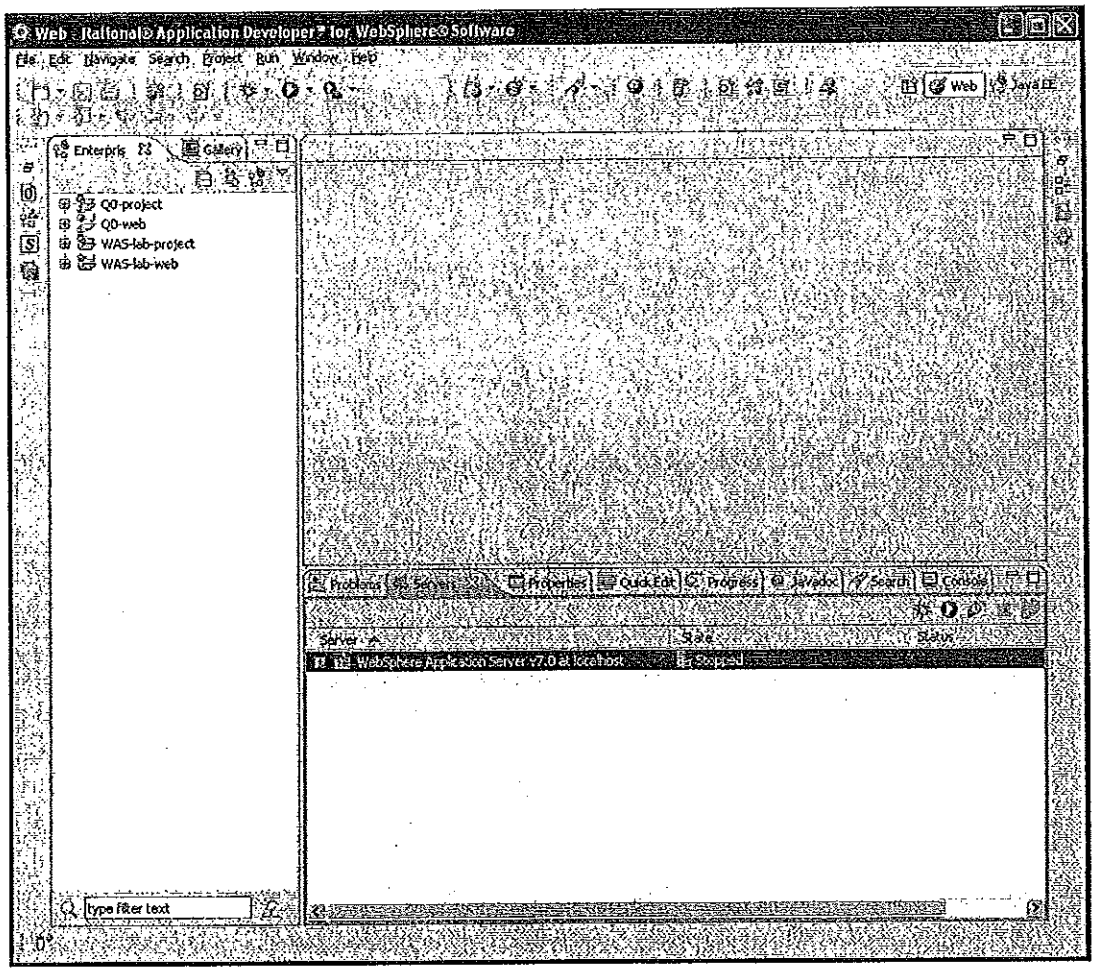
13. The next task we should perform is to can start the RAD (i.e., Rational Application Developer) to look at a J2EE application which will be used during the programming contest. RAD is a GUI development environment which is used to perform J2EE programming. The resultant programs will be published to WAS (WebSphere Application Server) so that it can be executed in the web environment.

    Double clicked the icon "IBM Rational Application Developer" to start RAD. A banner will appear and a Workspace Launcher window will appear which allows you to choose the base location of the workspace:



    Click the "OK" button to adopt the default value which is "C:\Documents and Settings\swg\IBM\rationalsdp\workspace".

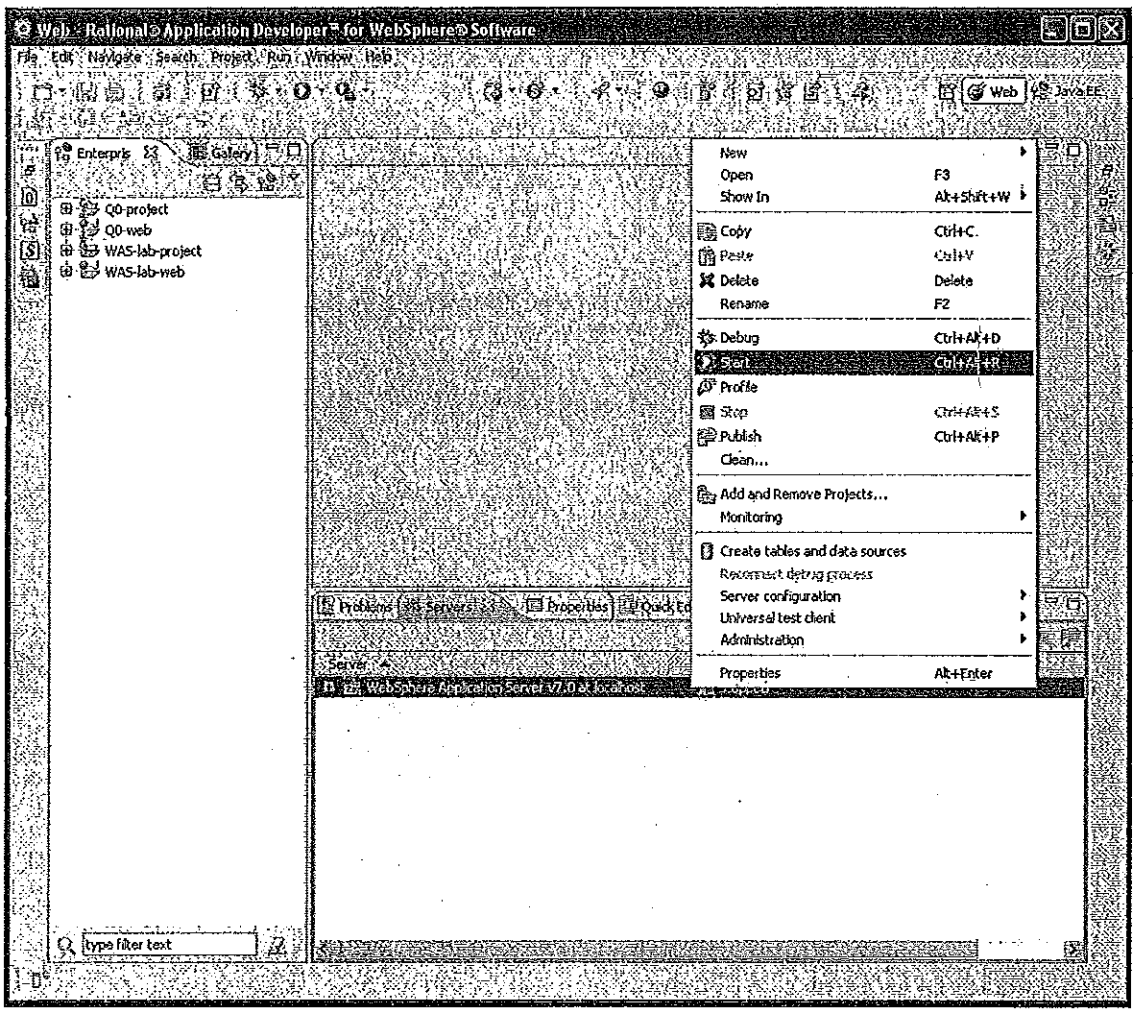
14. After the workspace is loaded, the following window appears:



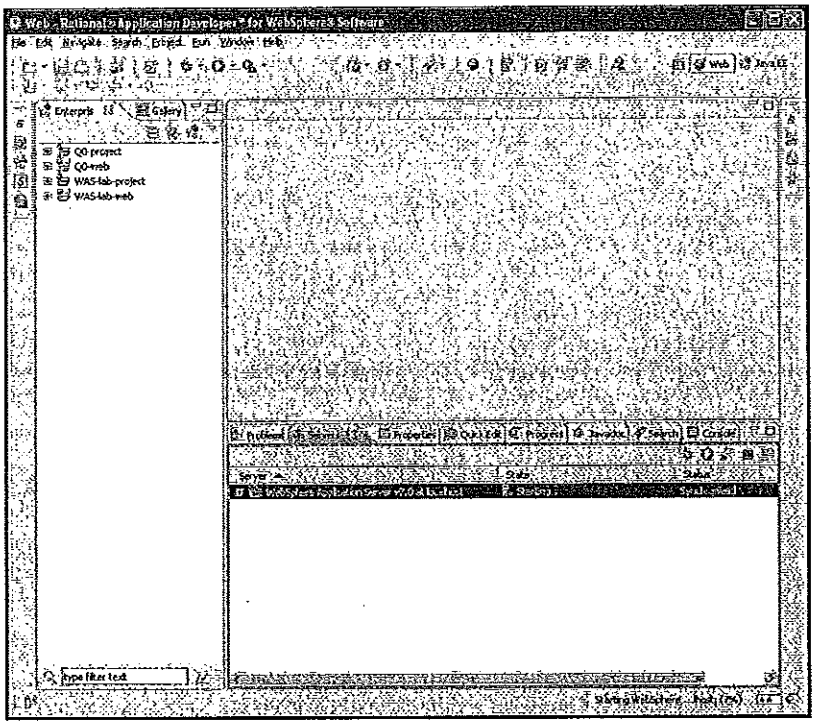
On the left side is the "Project Explorer" folder which contains a list of all available projects. In the above window, there are four available projects (i.e., Q0-project, Q0-web, WAS-lab-project and WAS-lab-web). Q0-project is an application which produces a course report, and its program codes are embedded in the project Q0-web. WAS-lab-project is an application which performs a distributed transaction, and its program codes are embedded in the project WAS-lab-web.

15. Before we start to review the program, we should start WAS (i.e., WebSphere Application Server) such that the program can be run via a web browser (i.e., Internet Explorer in our case).

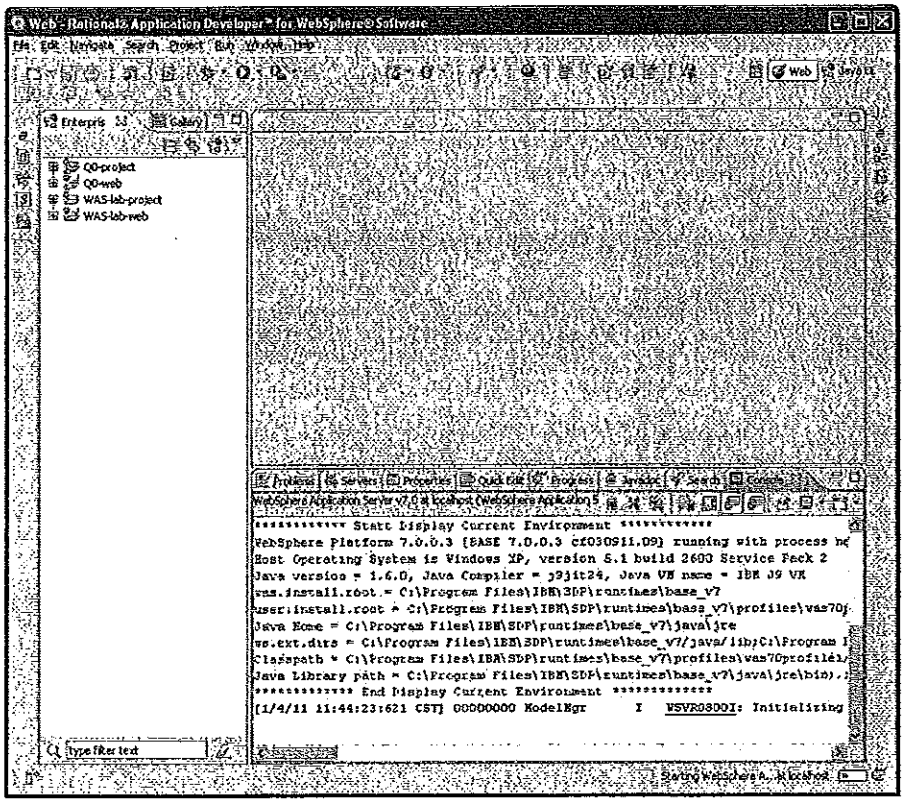
    Click on the 'Server' tab in the middle-bottom view. You see the 'WebSphere Application Server v7.0 at localhost' is stopped at the moment. First click on the server, then right click to show the command manual, then choose 'Start' to start WAS:



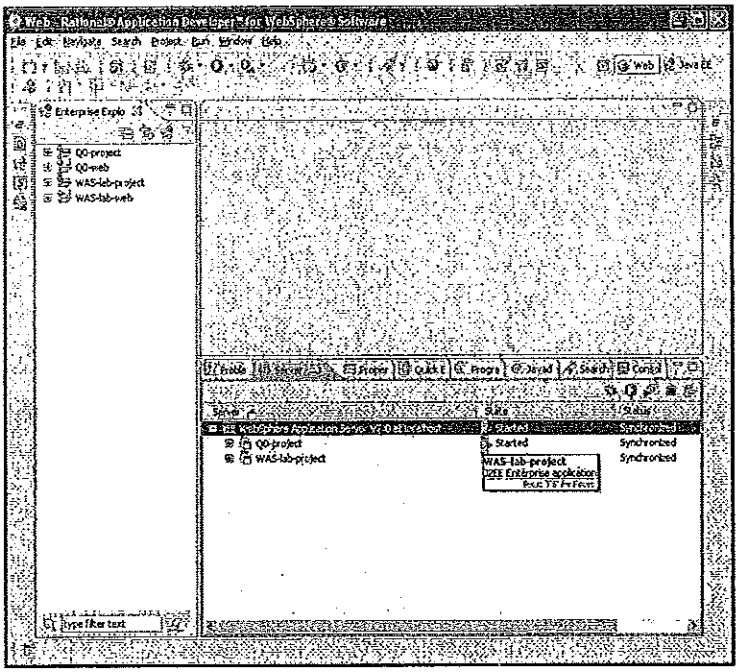
    Then you will see the state of the server changing from 'Stopped' to 'Starting':



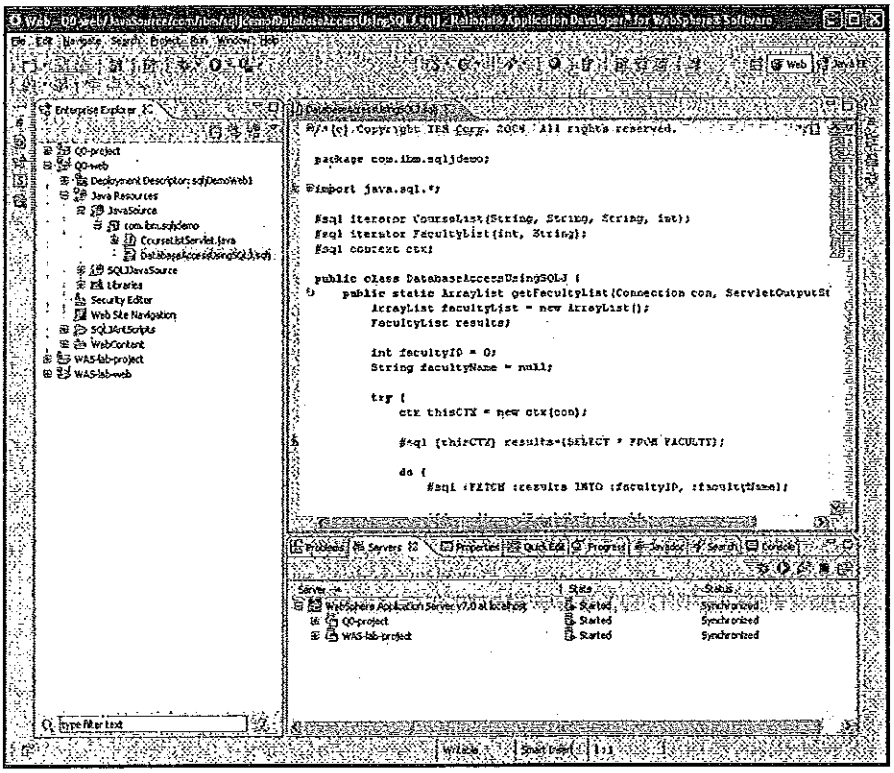
If you click on the tab 'Console', you might be able to see the system messages generated during startup:



After a while, WAS is booted up and the state of the server is changed to 'Started':

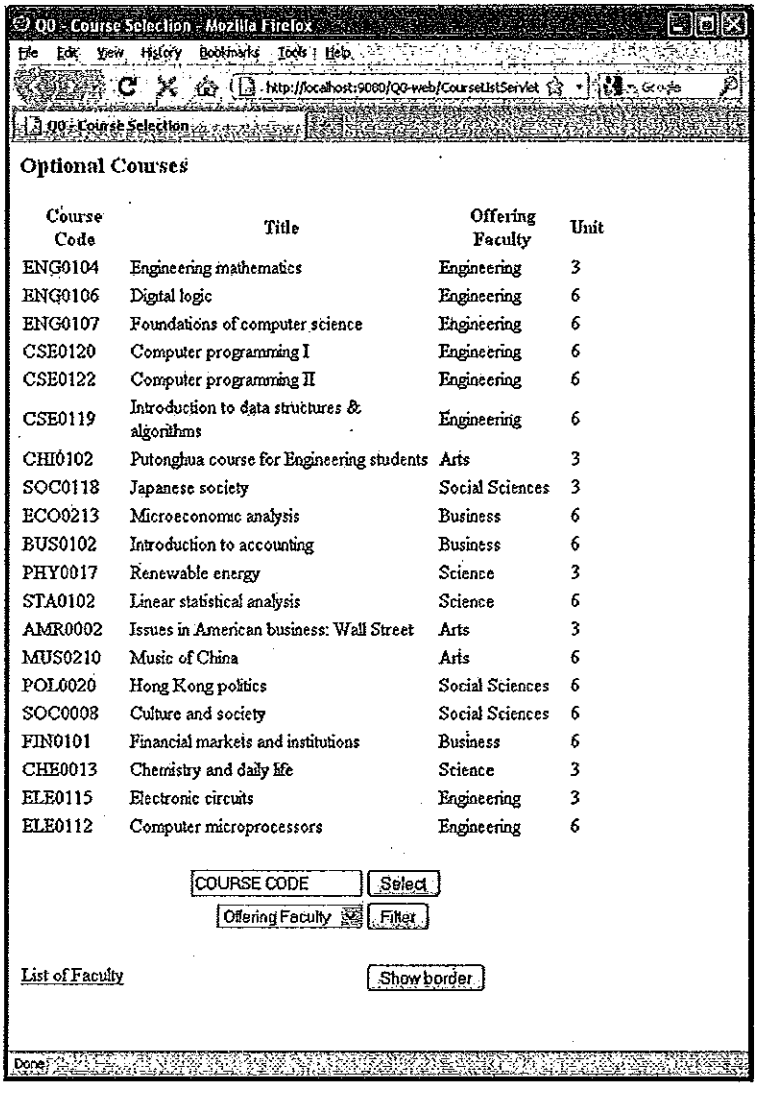


16. Now, we can continue to explore the project. Click the "+" signs of Q0-project and Q0-web to view the content of the projects:

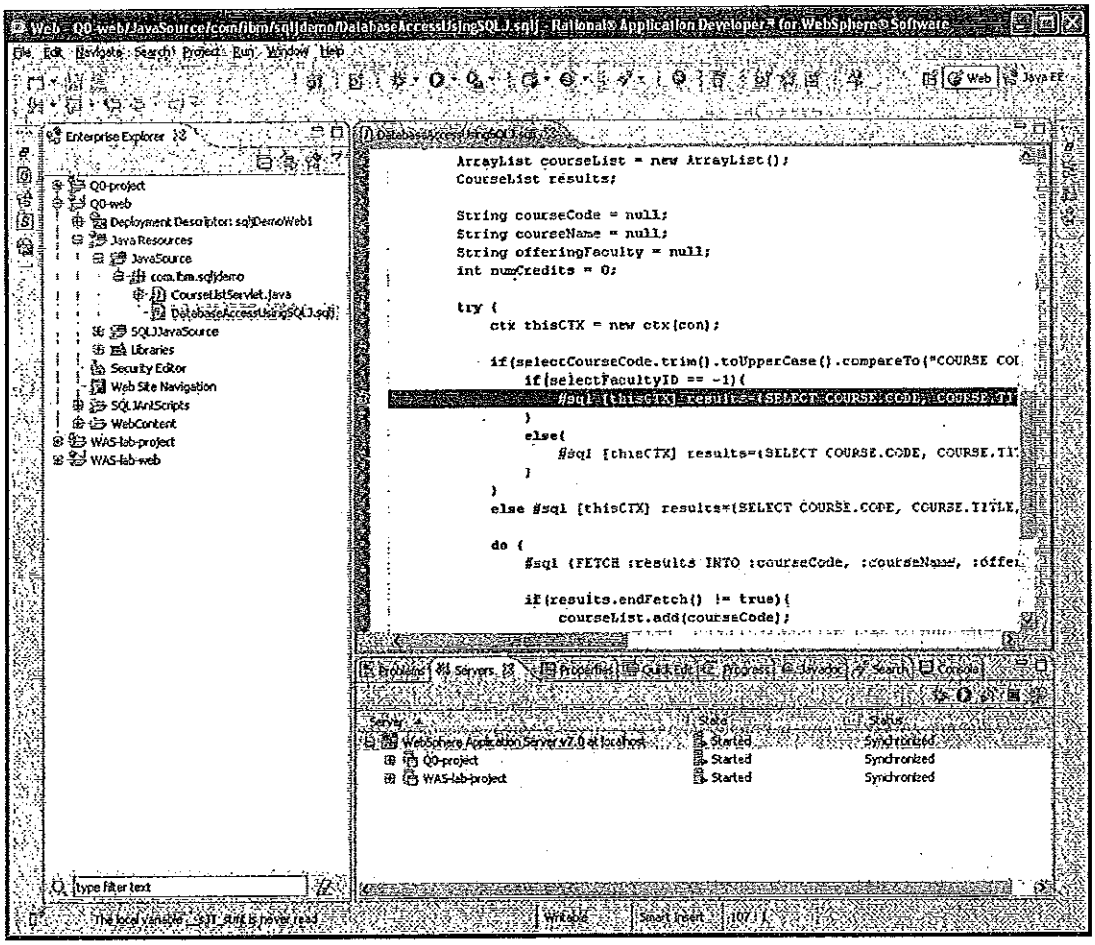


Further expand Q0-web by clicking the "+" signs of "Java Resources", "JavaSource", and "com.ibm.sqljdemo" until you see the 2 programs of the application (i.e., CourseListServlet.java and DatabaseAccessUsingSQLJ.sqlj). Double click "DatabaseAccessUsingSQLJ.sqlj" to edit this program. A new sub-window is displayed on the top-right portion of the window. Please refer to the above screen.

17. Now, let's see the output of this program. Open a Mozilla Firefox browser, and then select 'Bookmarks -> Q0 – Course Selection' to run the program Q0. The following screen is generated:

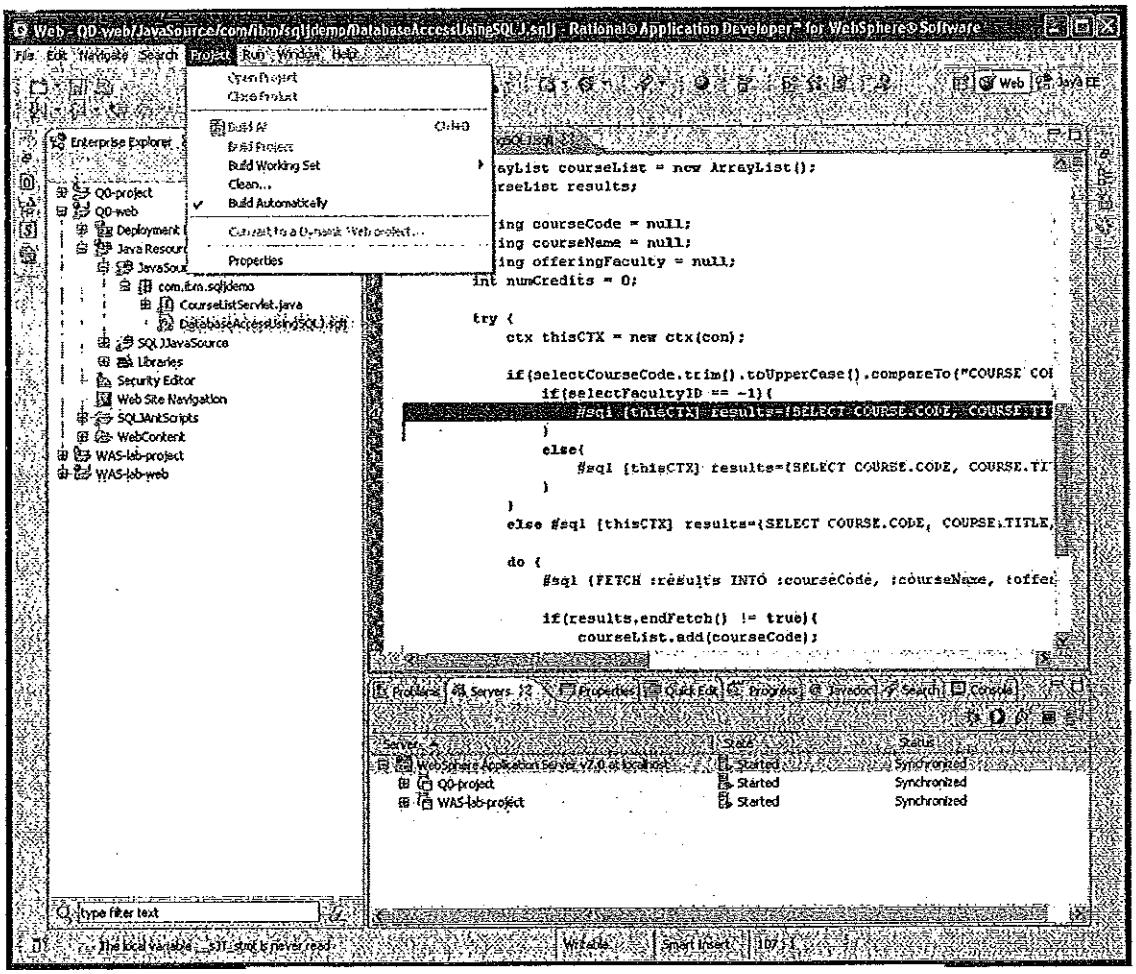


18. Now, we modify the program Q0 to reduce the lines of output. Scroll down the edit window until you reach line 107:



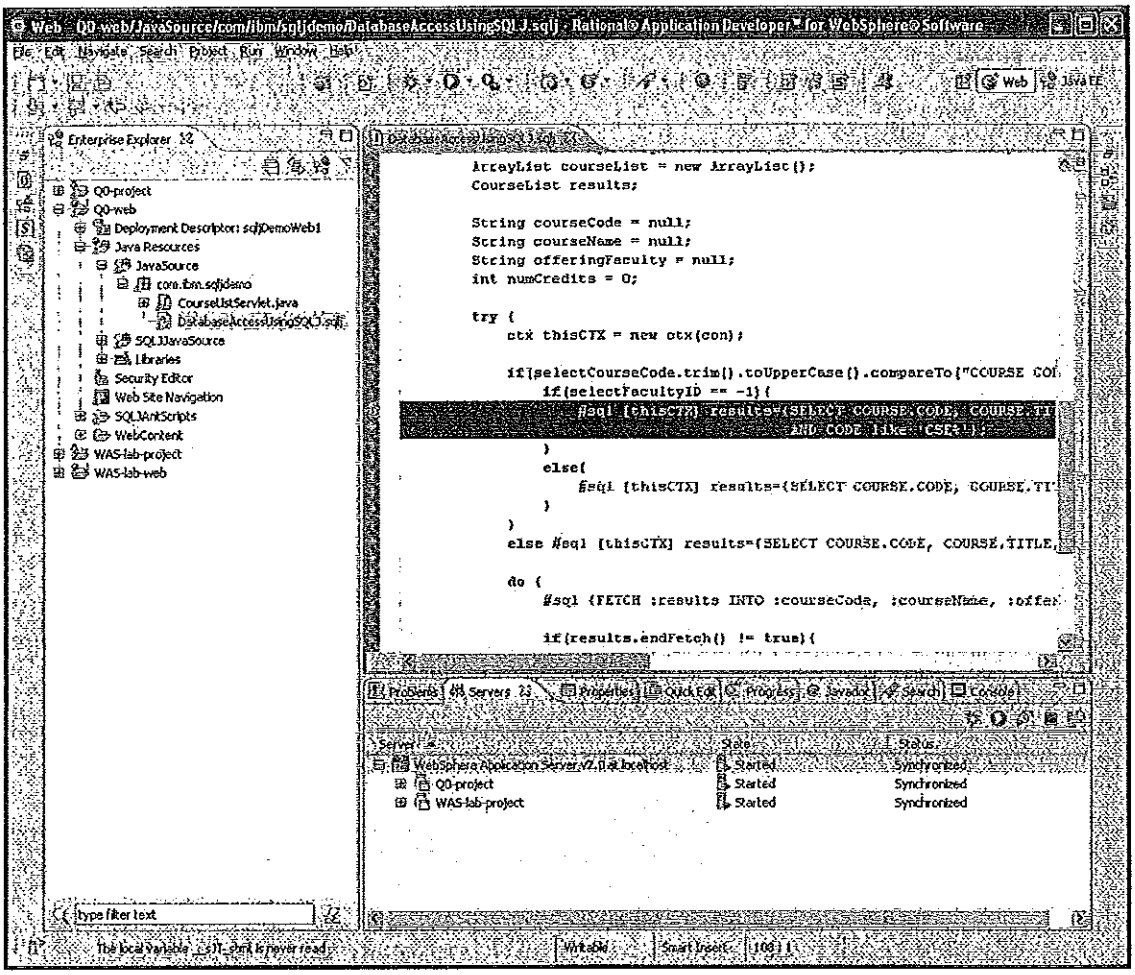
The above window shows the SQL statement "SELECT course.code, course.title, faculty.name, course.unit FROM COURSE, FACULTY where course.faculty_id = faculty.id" which obtains the list of available courses. This statement generates the list of courses and the results are displayed in the Internet Explorer.

19. Let's make some changes to the application. First of all, click the "Project" item under the manual. It can be seen that there is a tick beside the item "Build Automatically":



This means that after you modify something in the program and issue the save command, the project will be rebuilt automatically and the change will be deployed to WAS automatically.

20. Go to line 107 of the edit screen for the program "DatabaseAccessUsingSQLJ.sqlj". Modify the SQL statement by adding a where-clause "where code like 'CSE%'" at the end of the SQL statement:

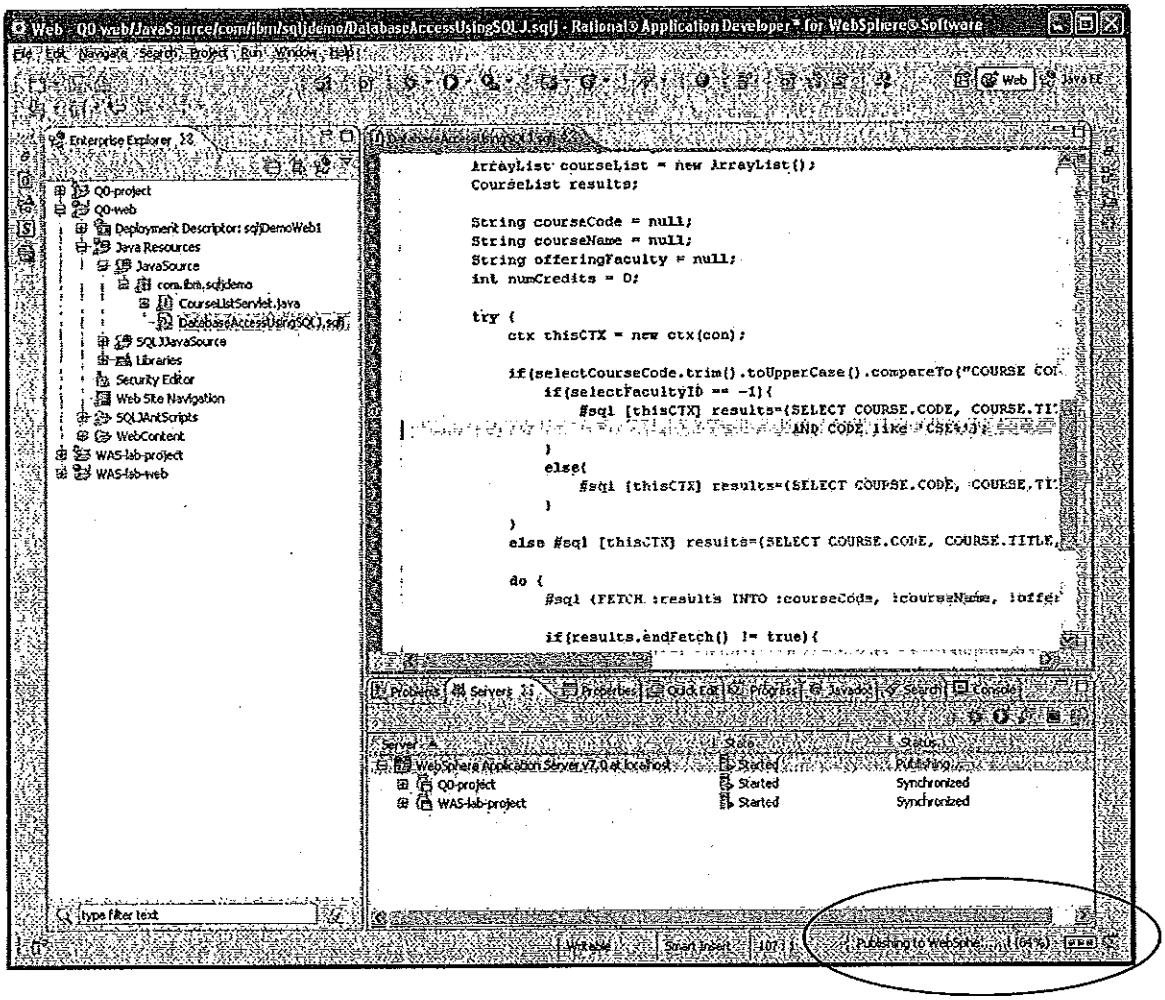


This will restrict the result of the SQL statement by only returning the department names starting with 'A' only. The character '%' is the wildcard character which matches any strings.

21. Press "Control-S" to save the edited content. At the same time, please note that in the bottom-right corner of the screen the following message is displayed

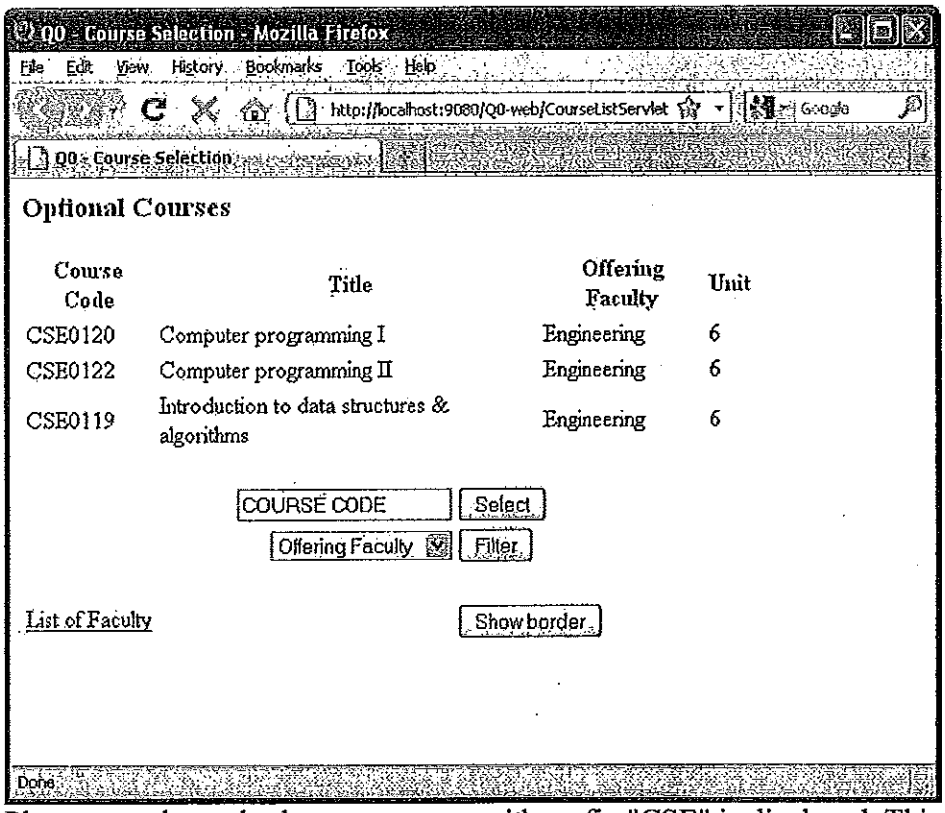
    Building workspace (0%)

    Please refer to the following window for an illustration. Please take attention to the circle on the bottom-right corner which shows the message:



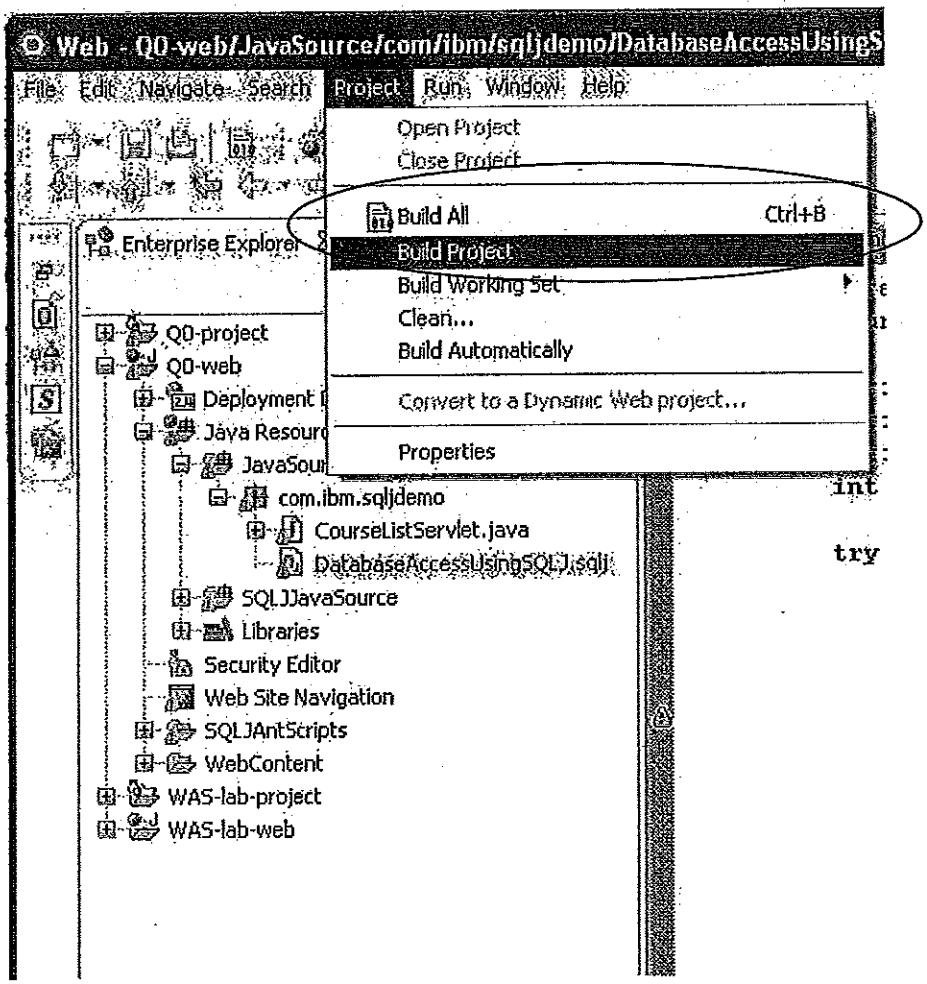
    After a while, the project rebuilding process is completed and the message will disappear.

22. Open the Internet Explorer window, and run the application again (or you can reuse the originally opened Internet Explorer and press the reload button). The following screen appears:



Please note that only the course name with prefix "CSE" is displayed. This indicates the new SQL statement is effective.

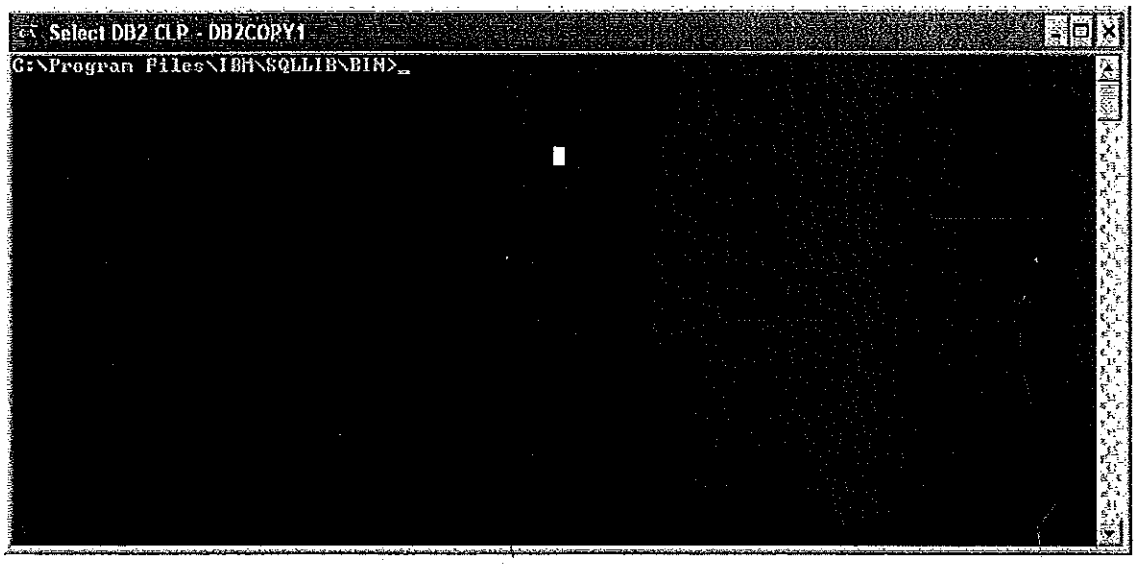
23. Since there is a tick beside the menu item "Build Automatically", the project will be rebuilt and redeployed automatically for every change to the project. If you want to perform multiple changes to the project before you rebuild the project. You can turn off the tick of the menu item "Build Automatically". In this case, you have to select either "Build All" or "Build Project" to manually start the project rebuild process. This is shown in the black circle in the following diagram:



Please also note that when you change a program, there will be a symbol "!" beside the affected objects under the project explorer in the left side of RAD. Please refer to the blue circle in the above diagram for an illustration.

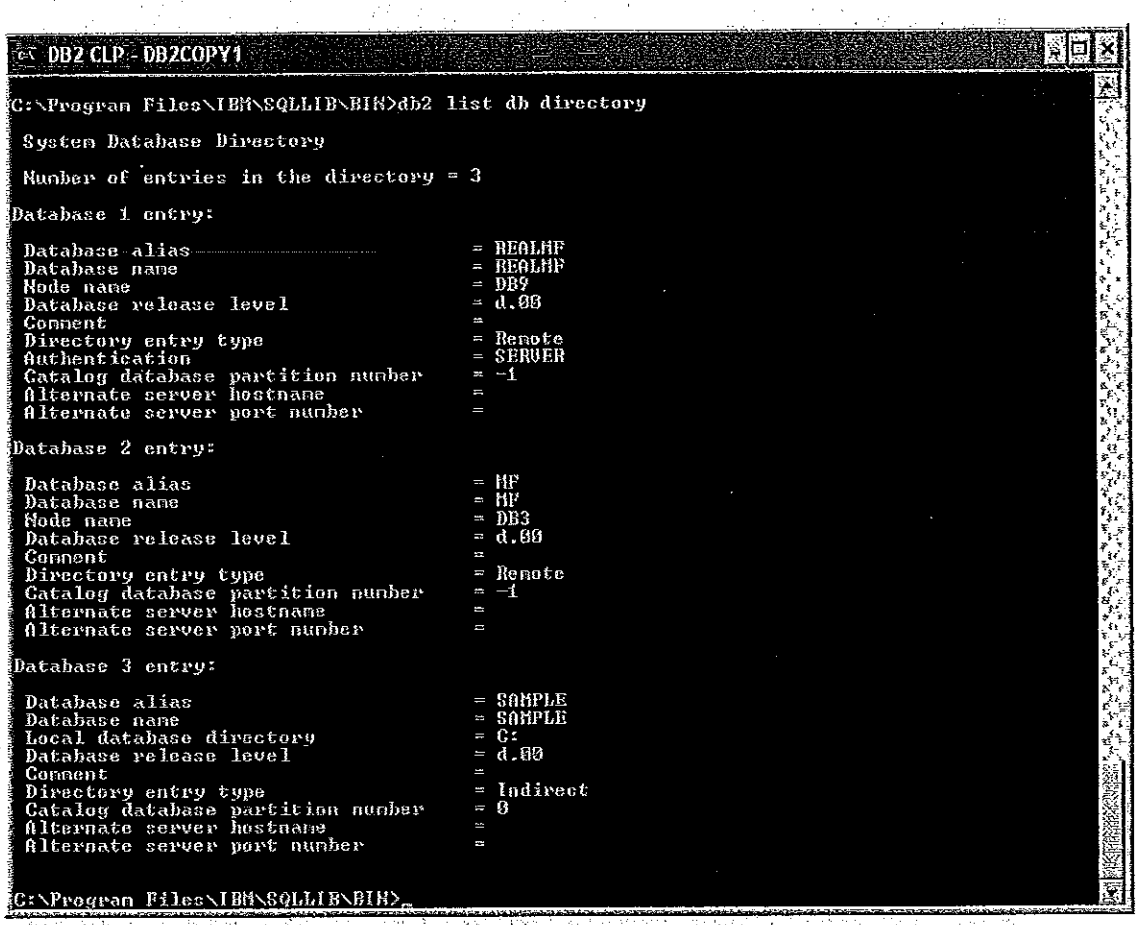
24. To review the DB2 database environment, open a DB2 Command Window by clicking the following Windows menu button: Start -> Programs -> IBM DB2 -> DB2COPY1 (default) -> Command Line Tools -> Command Window.

25. The following DB2 command window appears:



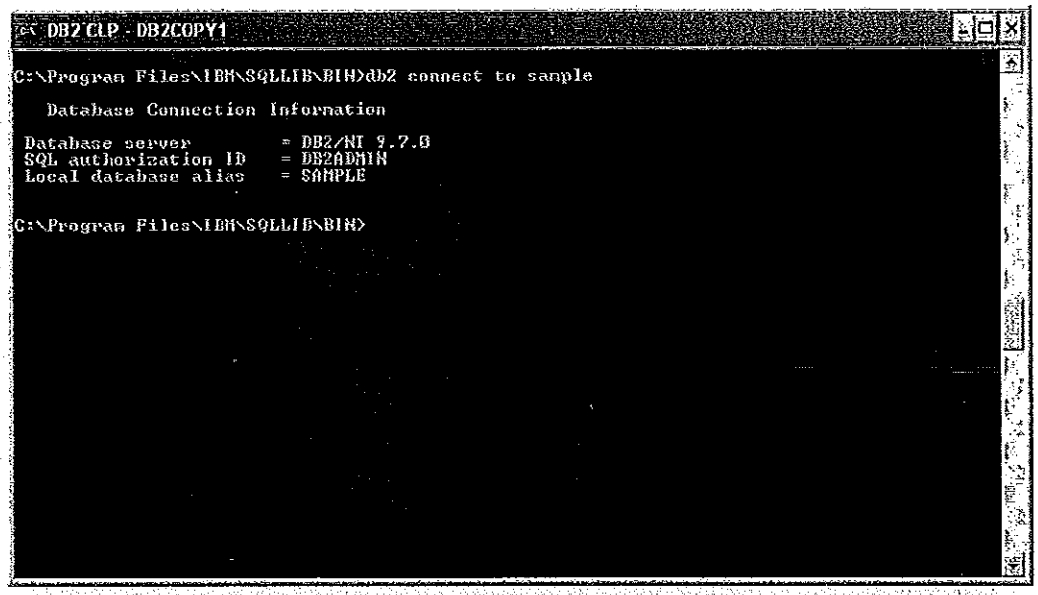
Please note that all DB2 SQL statements and commands must be entered to the DB2 command windows before they can be interpreted by DB2 properly (i.e., normal Windows command window does not work).

26. Execute the command "db2 list db directory" in the DB2 command window and you will see there are three databases (i.e., SAMPLE, MF and REALMF) defined in the system:

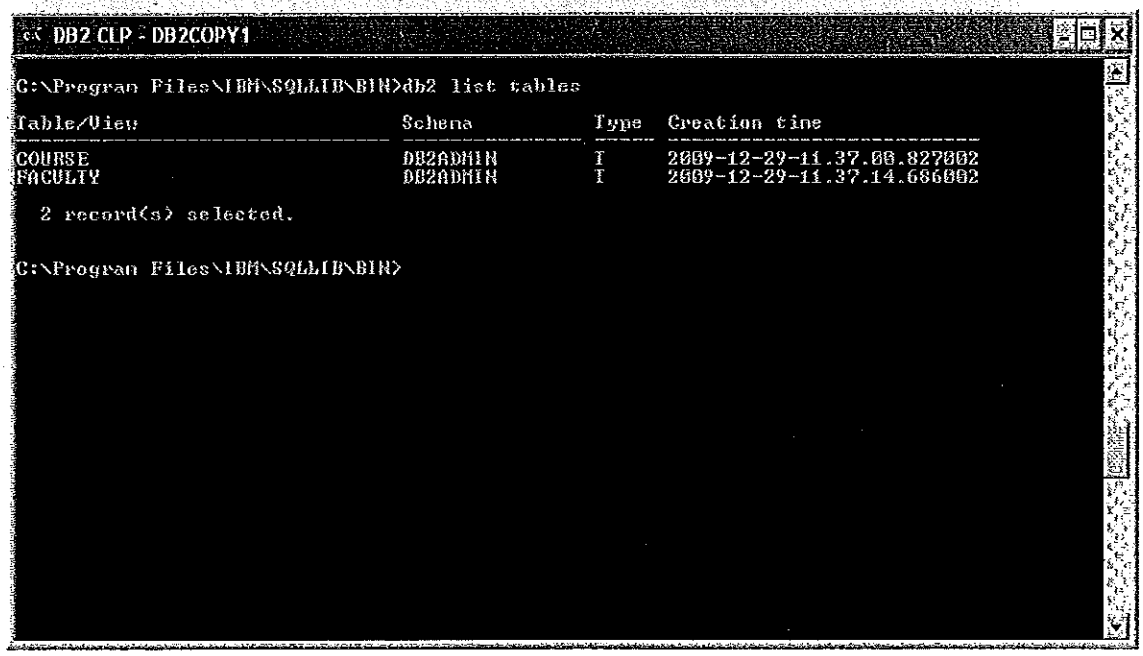


The database SAMPLE is a database defined in the primary instance (DB2) in the local PC. The database MF is a database defined in the secondary instance (DB3) in the local PC. The database REALMF is a database defined in a mainframe server in a remote server. We will access MF and REALMF in the later chapters.

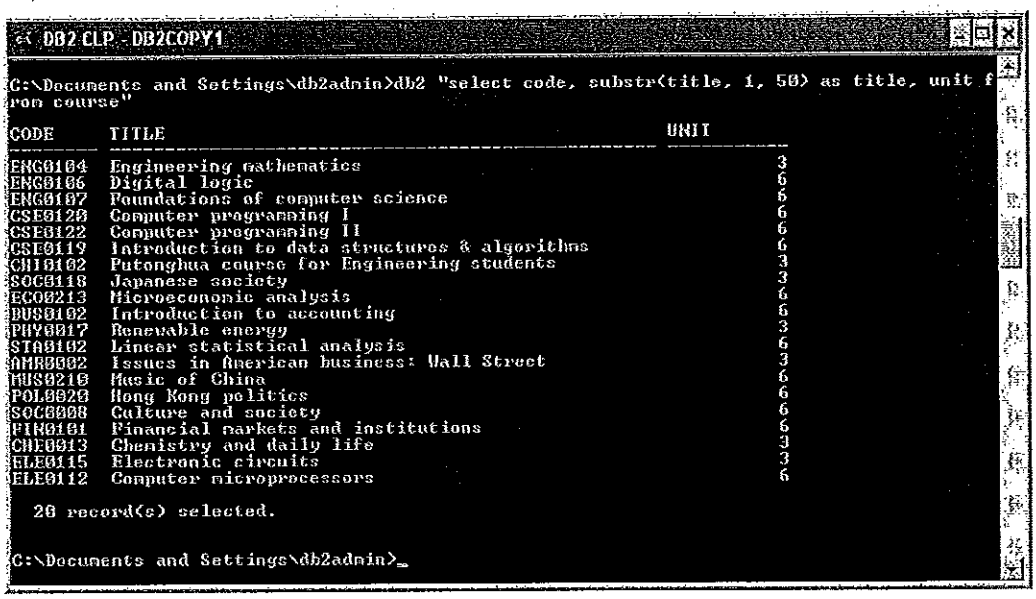
27. Enter the command "db2 connect to sample" to connect to the SAMPLE database. The Q0-project application is configured to query the SAMPLE database:



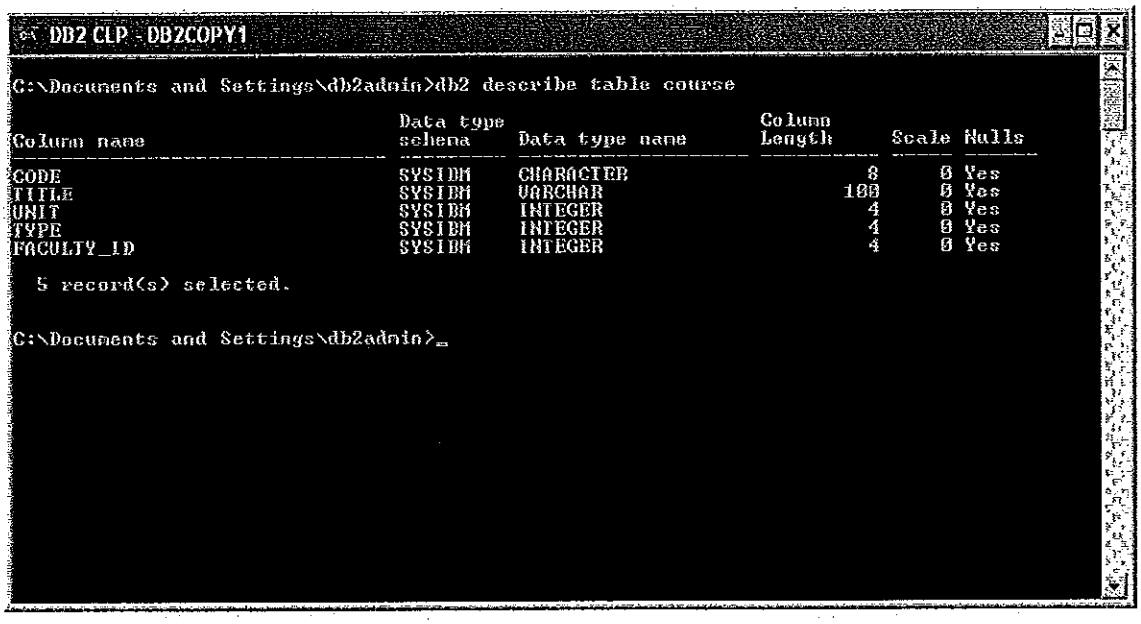
28. Enter the command "db2 list tables" to display the list of major database objects created by the user "db2admin":



29. Enter the command db2 "SELECT code, substr(title, 1, 50) as title, unit from course" which is a slightly modified version of that in the program "DatabaseAccessUsingSQLJ.sqlj". The only change is the addition of the substring function which trims the TITLE column to fit in the window. Refer to the following screen for the results:

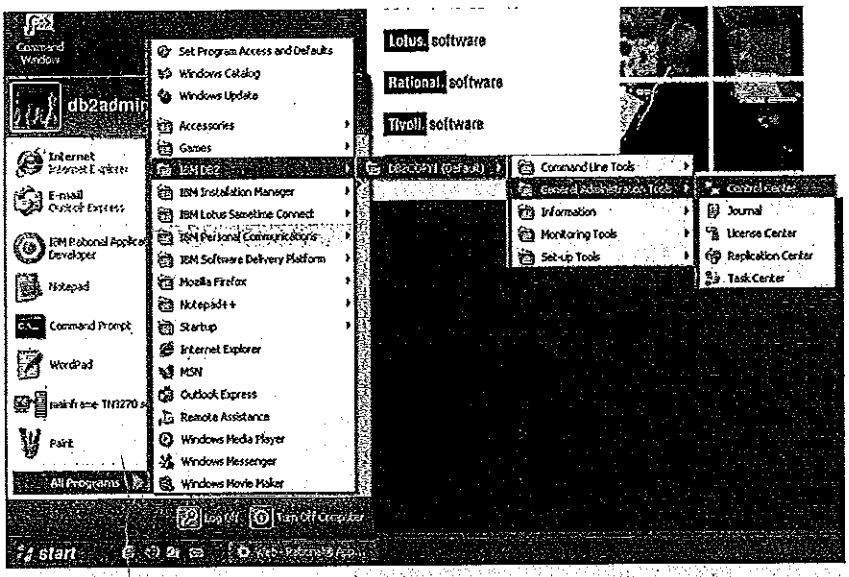


30. Finally, enter the following command "db2 describe table course" to display the column definition of the table COURSE:

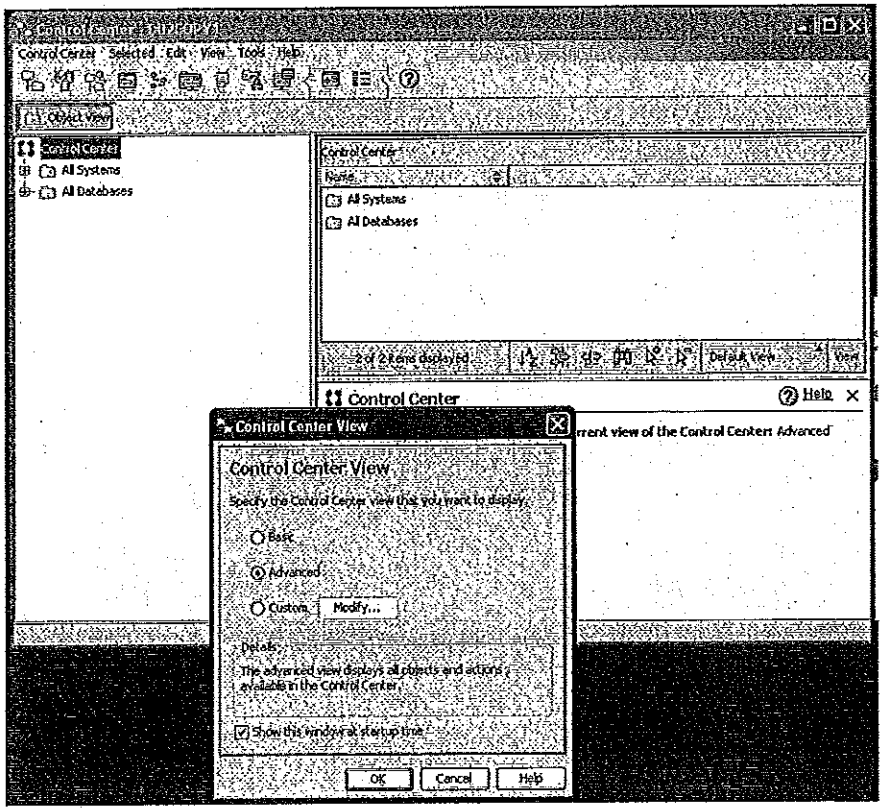


31. In general you can run SQL statements by putting each statement after the command "db2". You need to double quote the command in case it contains wildcard characters.

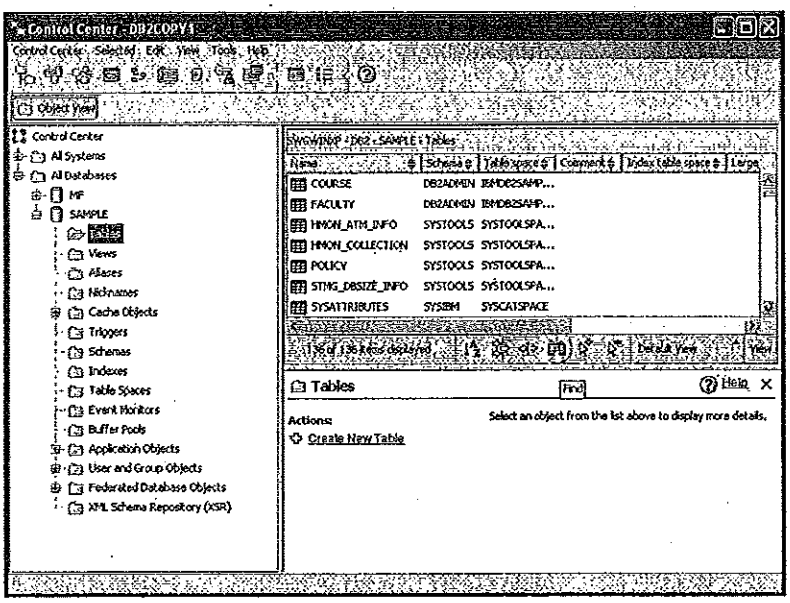
32. Now we will try to use the graphical interface of DB2. Run the DB2 control center from "Start"->"All Programs"->"IBM DB2"->"DB2COPY1 (Default)"->"General Administration Tools"->"Control Center":



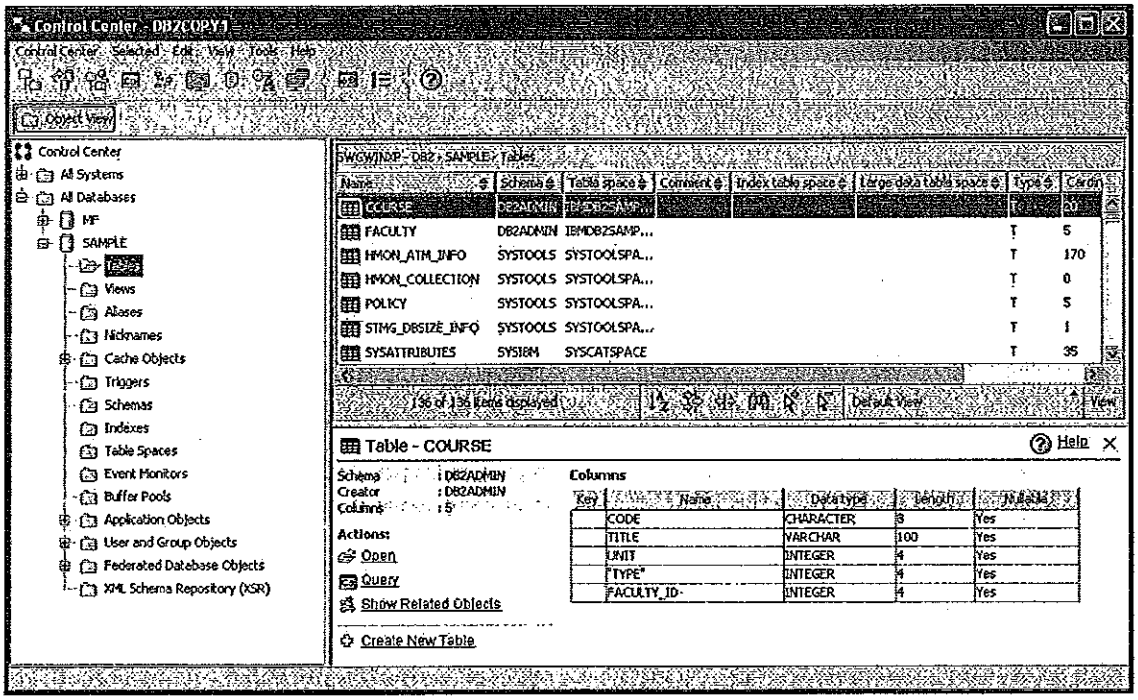
33. The DB2 control center is displayed. Press "OK" on the Control Center View to accept the default view:



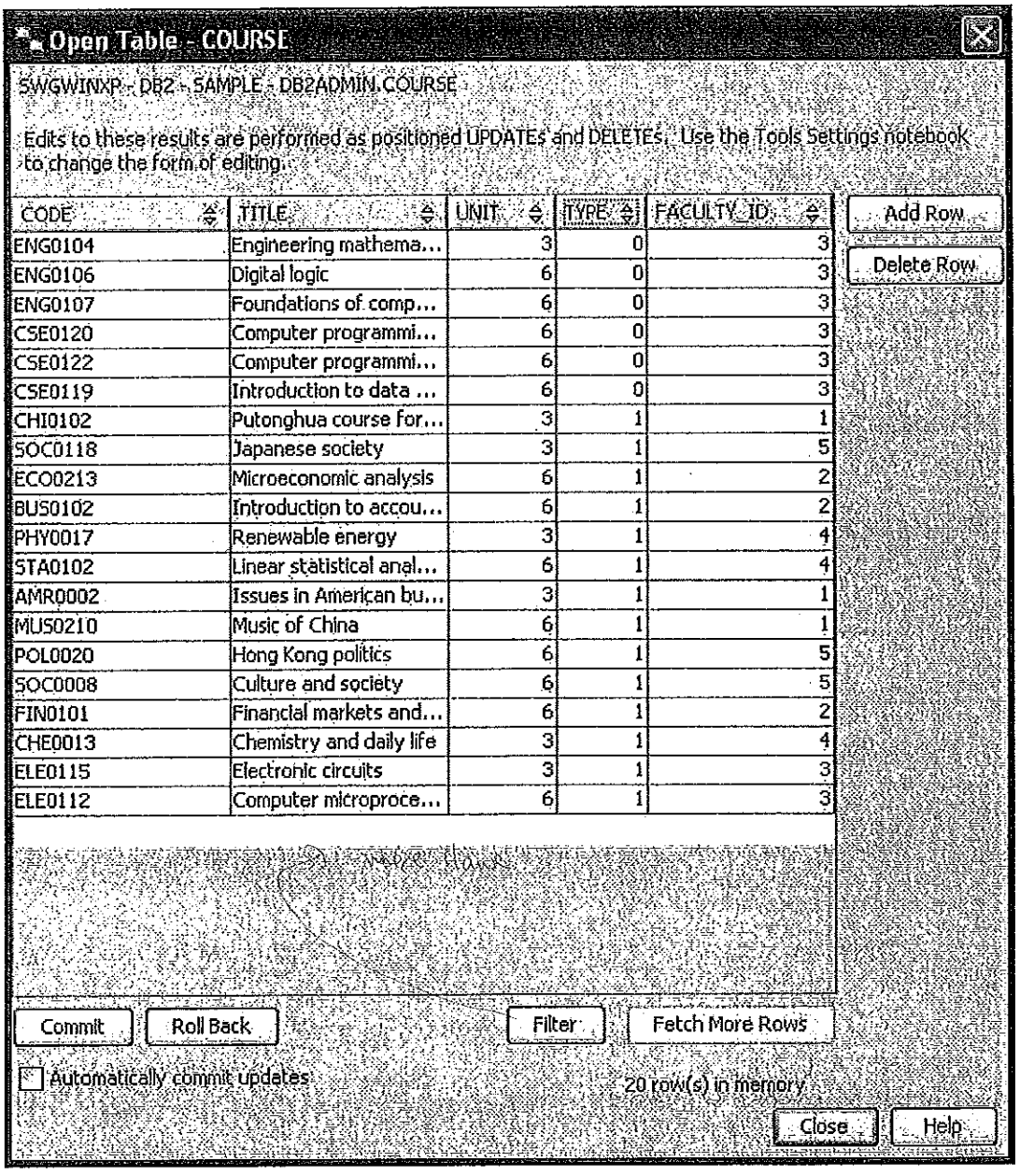
34. In the Control Center screen, navigate to "All Databases"->"SAMPLE"->"Tables" on the left pane and then the list of tables will be displayed on the top right pane:



35. Click on the table COURSE and the field definition of the table will be displayed on the lower right pane:



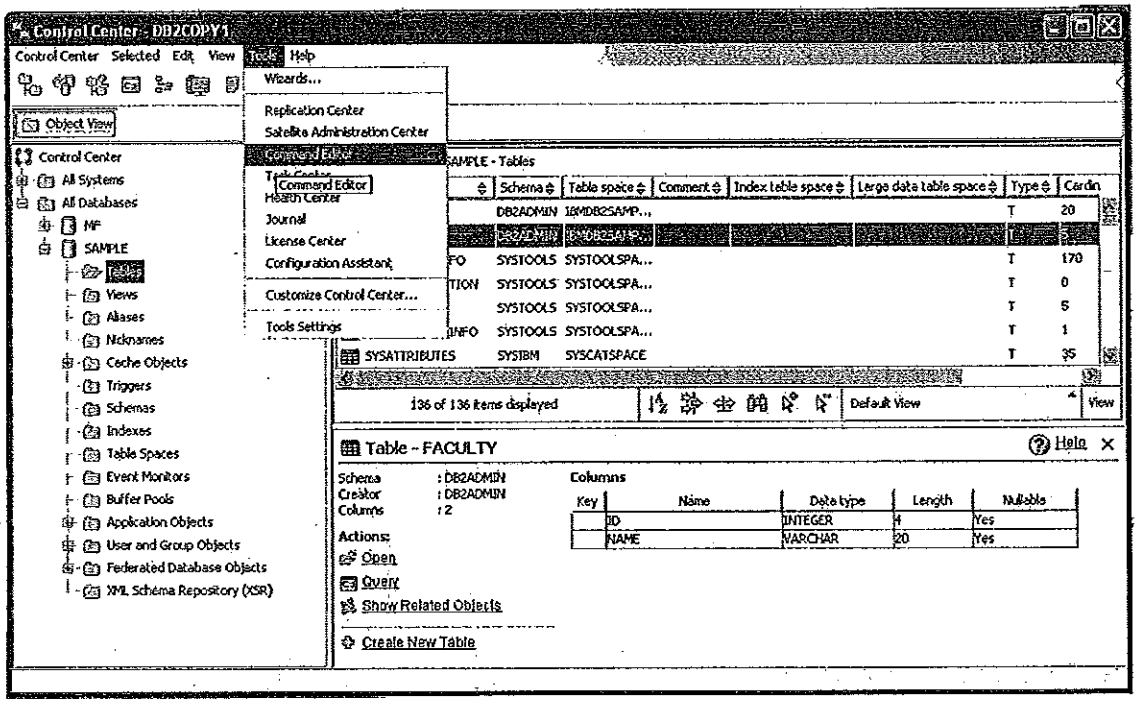
36. Double click on the table COURSE and then a new window is popped up to display the content of the table:



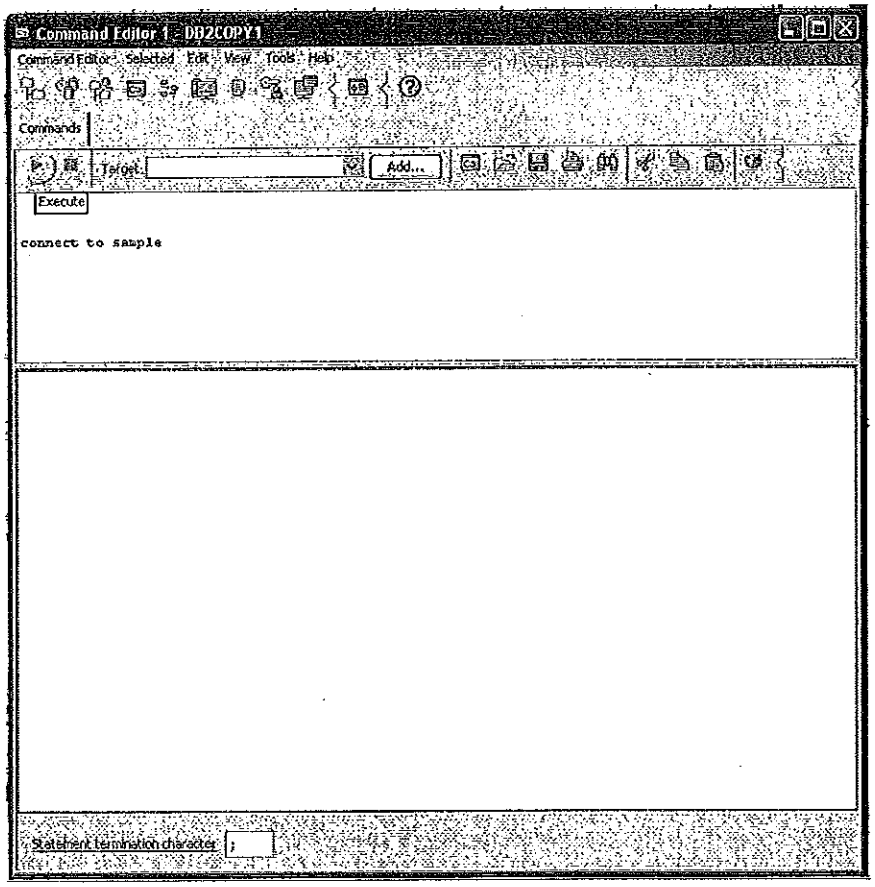
Spend some time on reviewing the functions of this window. Press the button "Close" to close the window.

37. Spend some more time to understand the other objects (e.g., indexes) in the database and functions of Control Center.

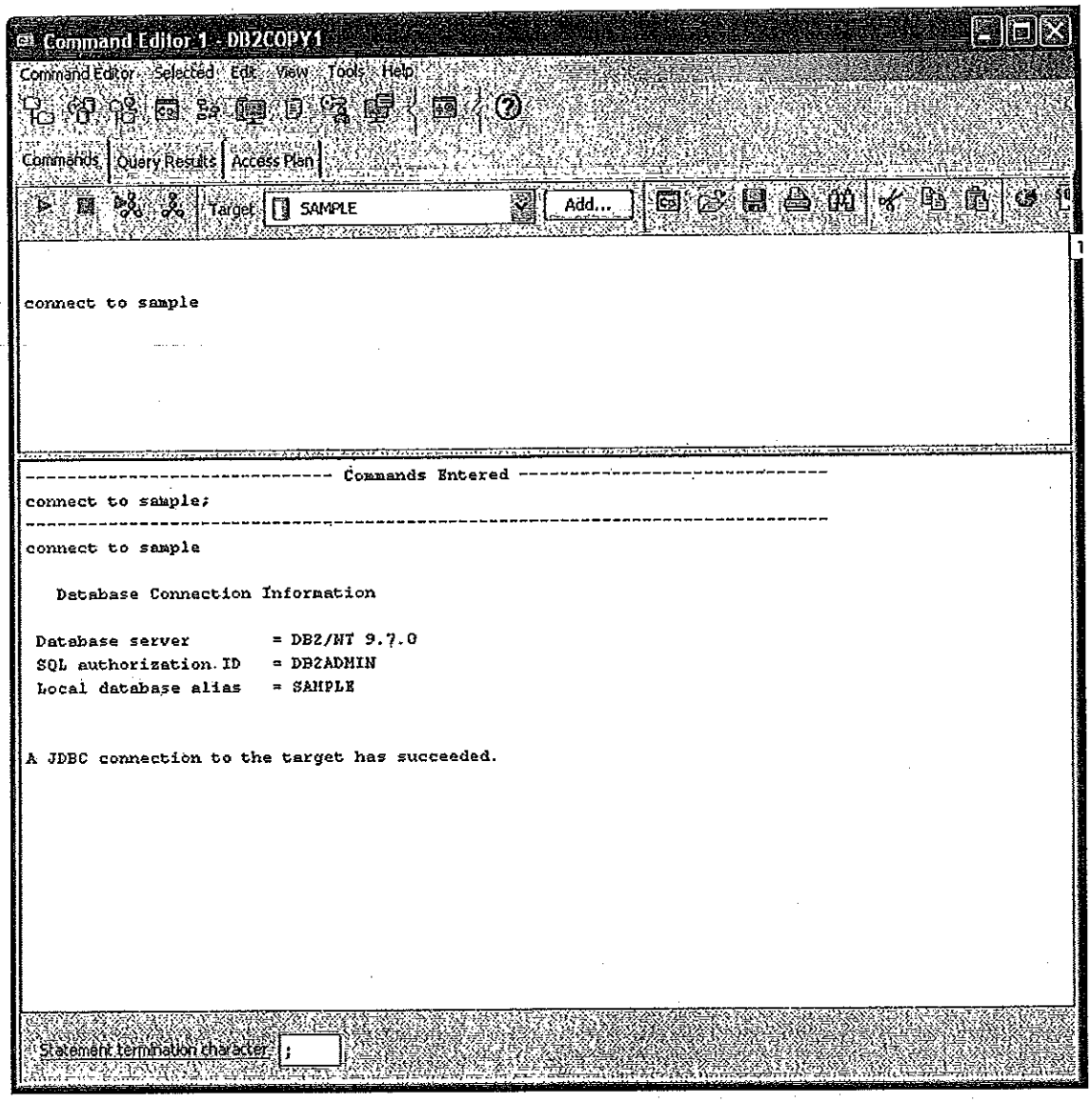
38. Then we will review the DB2 Command Editor. Execute the Command Editor through "Tools"->"Command Editor" from the manual bar of the Control Center:



39. The Command Editor screen is displayed. You can type the commands in the top pane and then the result will be displayed in the bottom pane. Type the command "connect to sample" in the top pane and then press the run button (i.e., the green triangle button) just above the top pane:



40. The Command Editor will then log into the database SAMPLE:



You can run SQL statements just like what you did in the command window. But you don't need to prefix each command by "db2".

Now please try the Command Editor out by entering more commands and see how it responds.

41. This is the end of this exercise.

IBM Inter-University
Programming Contest 2011 Training

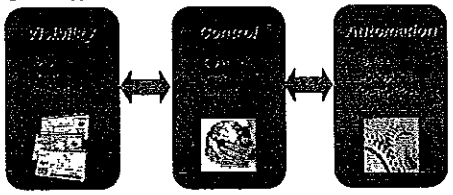
Chapter 2: Tivoli Netcool/OMNIbus Overview

Software Group, IBM

Tivoli software

IBM Service Management (ISM)

An Integrated Approach to Getting Business Results



Only IBM delivers integrated visibility across Business & IT Audiences.

e.g. Contextual UIs, Compliance, Security, Service, & Domain Dashboards

Only IBM delivers integrated control across Business & IT Assets.

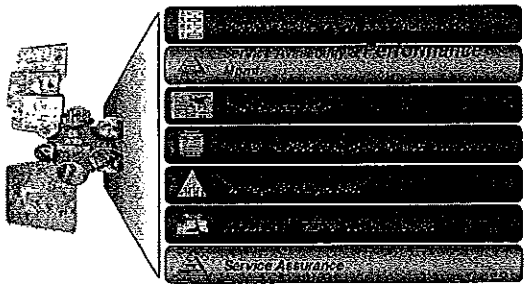
e.g. ESM, IT Asset Mgmt, Change & Config, Access & Identity Mgmt, Data Mgmt

Only IBM delivers integrated automation across Business & IT Operations.

e.g. Enterprise Ops, Service provider Ops, IT Ops, Security Ops, Storage Ops...

ISM - Overcoming the Barriers

'Operations'-specific focal points based on real-world pains



Analyst Accolades

Gartner

***LEADER** in Gartner's Magic Quadrants for Event Correlation & Analysis, Security Information, SOA Governance & J2EE Application Management.*

FORRESTER

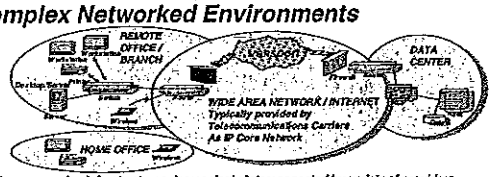
*"IBM is a **LEADER** in the business service management (BSM) space, based on IBM's wealth of experience in delivering effective system management solutions." – Forrester Wave*

*IBM is a **LEADER** in the IT Application Management Software Market – Successful application management solutions must include an end to end view of all components, consolidate reports in a single pane of glass and must support a form of process improvement.*

IDC

*IBM is **#1** in the Performance & Availability Management Market in 2006, with a 19.3% market share.*

What is Event Management? – Extended into Complex Networked Environments



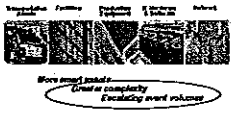
- *Large complex infrastructures, increasingly interconnected by variety of providers*
- *Ever increasing dependency on higher bandwidth / broadband for new "triple play" services (e.g., Voice, Video, and Data)*
- *Maintaining high Quality of Service (QoS) is key for multiple services sharing the same network*
- *Increased range and reach of networks leads to reliance on skilled operators when solving problems*
- *Determining the actual cause of certain problems requires understanding events across IT and Network disciplines*

The goal of End-to-End Event Management is to consolidate events, to focus on "root causes" (and identify related "symptoms"), and – where possible – to automate the resolution in response to any problem in the entire environment reducing MTTR (Mean time to Repair) and maintaining QoS within the network.

Tivoli Netcool/OMNIbus

Market Leading Event Consolidation, Correlation & Automation

Meeting the challenges of Dynamic Infrastructure Management



- **Maximize Service Availability.** Leverage hundreds of out-of-the-box integrations, with included domain intelligent event reduction rules, to monitor end-to-end infrastructure status and health.
- **Reduce Operational Costs.** Consolidate NOCs, tools and management sources into a single pane-of-glass and integrated management infrastructure.
- **Improve Staff Productivity.** Utilize normalization, de-duplication, aggregation, correlation capabilities, as well as time, device, and service based event reductions.

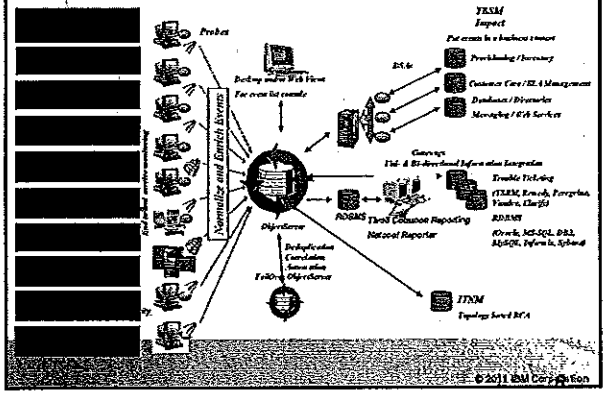
  Seamless UI navigation based on common Tivoli GUI architecture delivers task-based workflows that represent common operations used to perform operations and service management actions
- **Minimize human intervention.** Exchange information between peer systems and automate maintenance actions and procedures.
- **Increase Confidence.** Leverage proven availability and reliability, with huge event throughput capability, trusted system redundancy, failover and security.

IBM Tivoli Netcool OMNIbus

Tivoli software

Event Sources

Tivoli Netcool/OMNIbus Solution



Netcool/OMNIbus Integrations (250+)

Netcool Knowledge Library SNMP support (>175 MIBs), including:

Vendor Alliances (~20) on OPAL:

- Alcatel
- Avaya
- Siemens
- Ericsson
- Tellabs
- Marconi
- Lucent
- Nortel
- Huawei
- Fujitsu
- Ciena
- Cisco
- Juniper
- Checkpoint
- Crossbeam
- McAfee
- SAP
- Vitria
- Verisign

New and updated versions are developed and deployed on a quarterly basis.

IBM Tivoli Netcool OMNIbus Probe

Tivoli software

What are Probes?

- Probes are software components used to collect event information and send it to the ObjectServer.
- Probes enable the ObjectServer to be independent of the systems or devices being monitored.
  - Over 300 types of probe
  - Both generic and vendor-specific probes exist
- Probes can modify and enrich event information
  - Use additional information from lookup tables
  - Perform arithmetic functions
- Probes are resilient
  - Reliable TCP connection
  - Store and forward functionality (SAF)
  - Automatic fail-over to alternate ObjectServer

What are Probes? (continued)

- The ObjectServer obtains most of its event data via the probes
  - Event data may be also obtained via Automations, Tools, Gateways or SQL
- Every probe comprises a binary, a rules file and a properties file:
  - *Probe binary* retrieves and tokenises event streams, and sends the event to the ObjectServer.
  - Interpreted *rules file* used to populate the field values within alerts.status
  - *Properties files* determine the run-time settings of the probe.

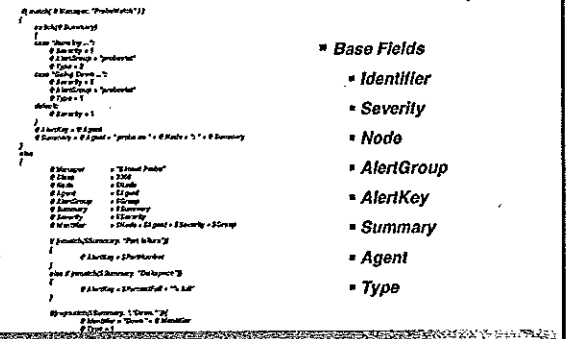
Netcool Omnibus Probes

- *Collection*
  - *Source-specific code for integrating with 3rd party systems and event synchronisation*
    - *Standard protocols: SNMP, Syslog, ODBC, 3GPP...*
    - *Custom message formats: Socket, Logfile*
    - *Proprietary/API: Many Vendors*
- *Normalisation*
  - *Common code for rules-based event processing:*
    - *Normalisation*
    - *Enrichment*
    - *Human Readability*
    - *Pre-classification for ObjectServer Correlation and Netcool processing*

Types of Probes

- Device
- Log file
- Database
- API
- CORBA
- Miscellaneous
- Security

Netcool Omnibus Probes – Rules File Example



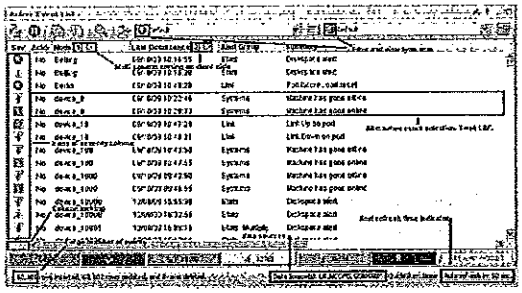
- *Base Fields*
  - *Identifier*
  - *Severity*
  - *Node*
  - *AlertGroup*
  - *AlertKey*
  - *Summary*
  - *Agent*
  - *Type*

IBM Tivoli Netcool/OMNIbus Web GUI

Tivoli software

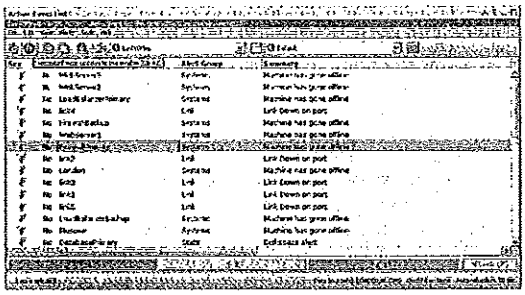
IBM Software Group | Tivoli software

AEL Features



IBM Software Group | Tivoli software

Active Event List – 'Tivoli Enterprise Console' Style Look and Feel

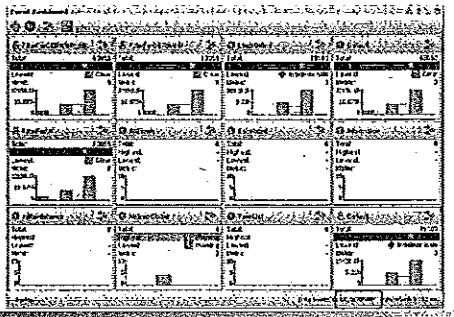


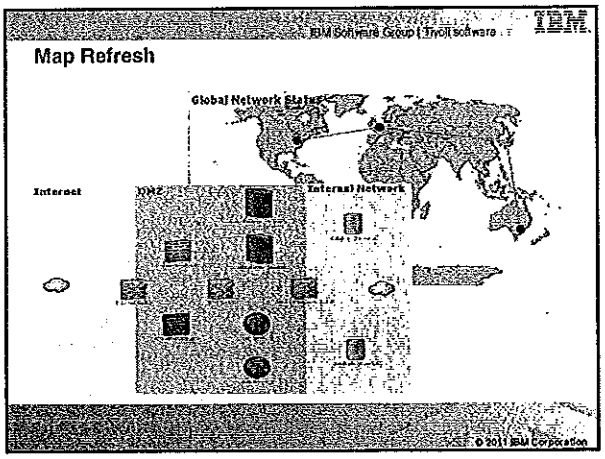
Event Dashboard

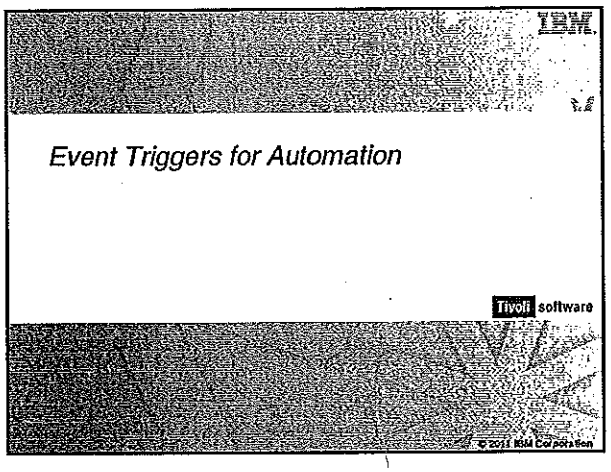
Tivoli software

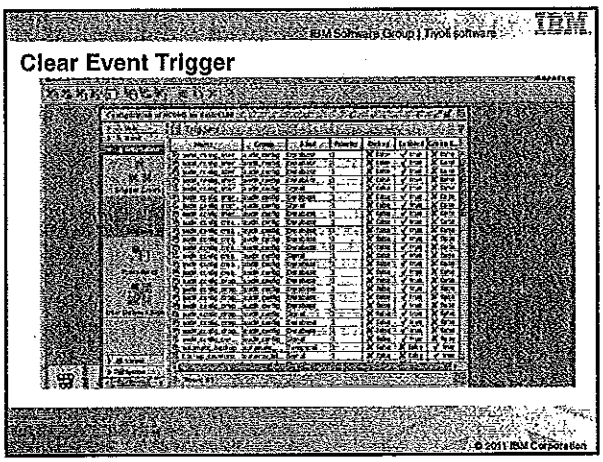
IBM Software Group | Tivoli software

Event Dashboard

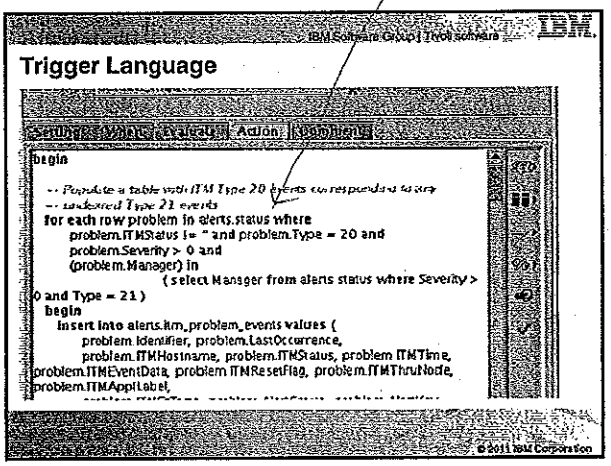








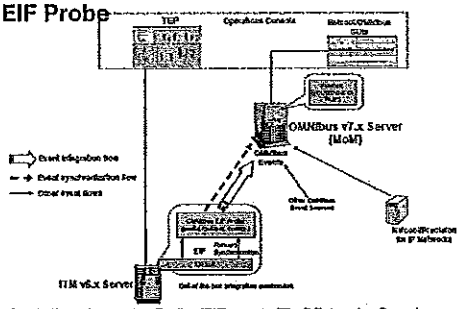
event stored in here.



Event Clear Trigger and Dashboard Status Update Demonstration

Tivoli software

Tivoli EIF Probe



- Sends Event Integration Facility (EIF) events (Tivoli Enterprise Console Events) to OMNIbus.
- Maps TEC event fields to OMNIbus fields.

What we need for sending EIF events

- Postzmsg
- Omnibus IP and receive ports and put it into a config file with the below contents:

```
#
#-------------------------------------------------------------------
# Licensed Materials - Property of IBM
# "Restricted Materials of IBM"
#
# Sample program file showing updates for OMNIbus to allow error
#     events to be sent to EIF probe
#
# (C) Copyright IBM Corp. 2007 All Rights Reserved.
#-------------------------------------------------------------------
#
#
ServerLocation=192.168.10.5
ServerPort=9998
#BufferEvents=NO
BufEvtPath=/tmp/itm_sync_error.cache
#BufEvtMaxSize=4096
```

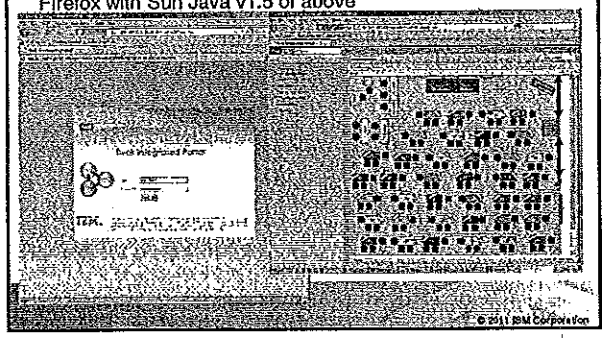

Syntax for postzmsg

- Create a RED color event:
  - postzmsg.exe –f my.conf –r CRITICAL –m contest EventID=101 situation_name="team1" satuation_status="Y" Integration_type="U" situation_origin=team1 situation_displayitem='team1' team1 ITM
- Create a GREEN color for clear event:
  - postzmsg.exe –f my.conf –r CRITICAL –m contest EventID=101 situation_name="team1" satuation_status="N" Integration_type="U" situation_origin=team1 situation_displayitem='team1' team1 ITM

pvt ... default.

Access to Omnibus Web GUI

- http://192.168.10.5:16310 and redirect to https using Firefox with Sun Java v1.5 or above



Follow Chapter 2 Tivoli Netcool Omnibus Exercise

THANK YOU!

Any Questions?

Chapter 2: Tivoli Netcool Omnibus

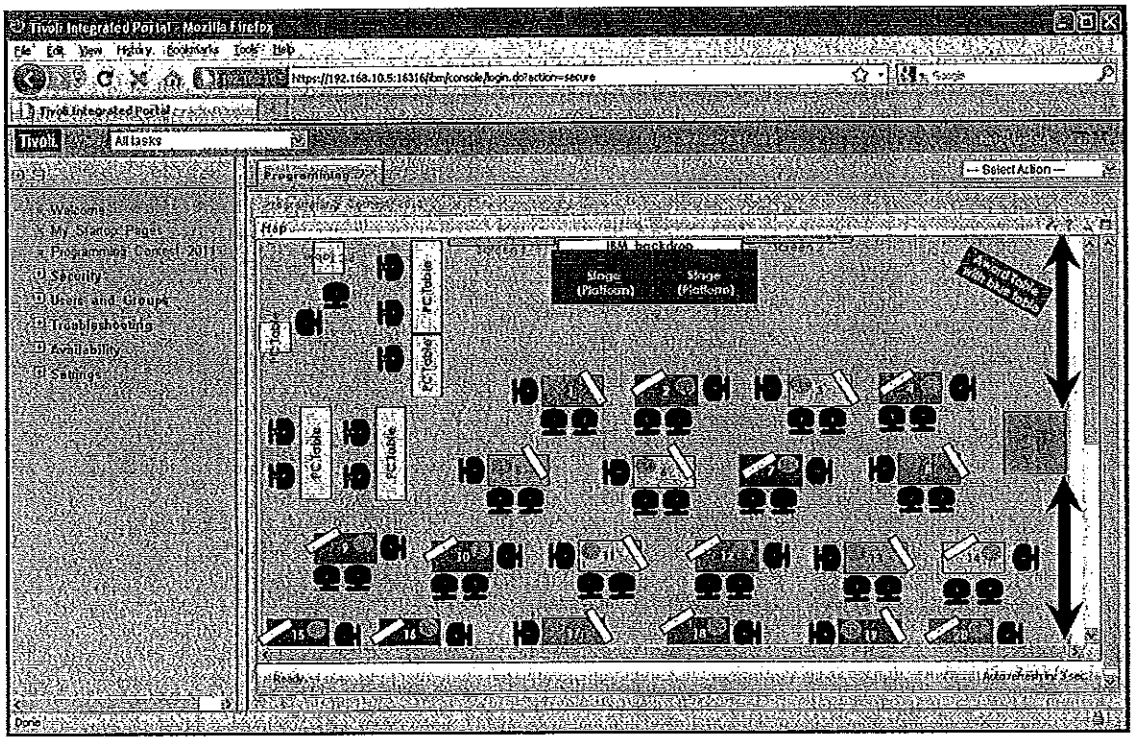
Objectives

In this exercise, we will learn:

- Review server interface of Omnibus
- Review and change Omnibus client scripts
- Observe the reaction of the Omnibus based on client request

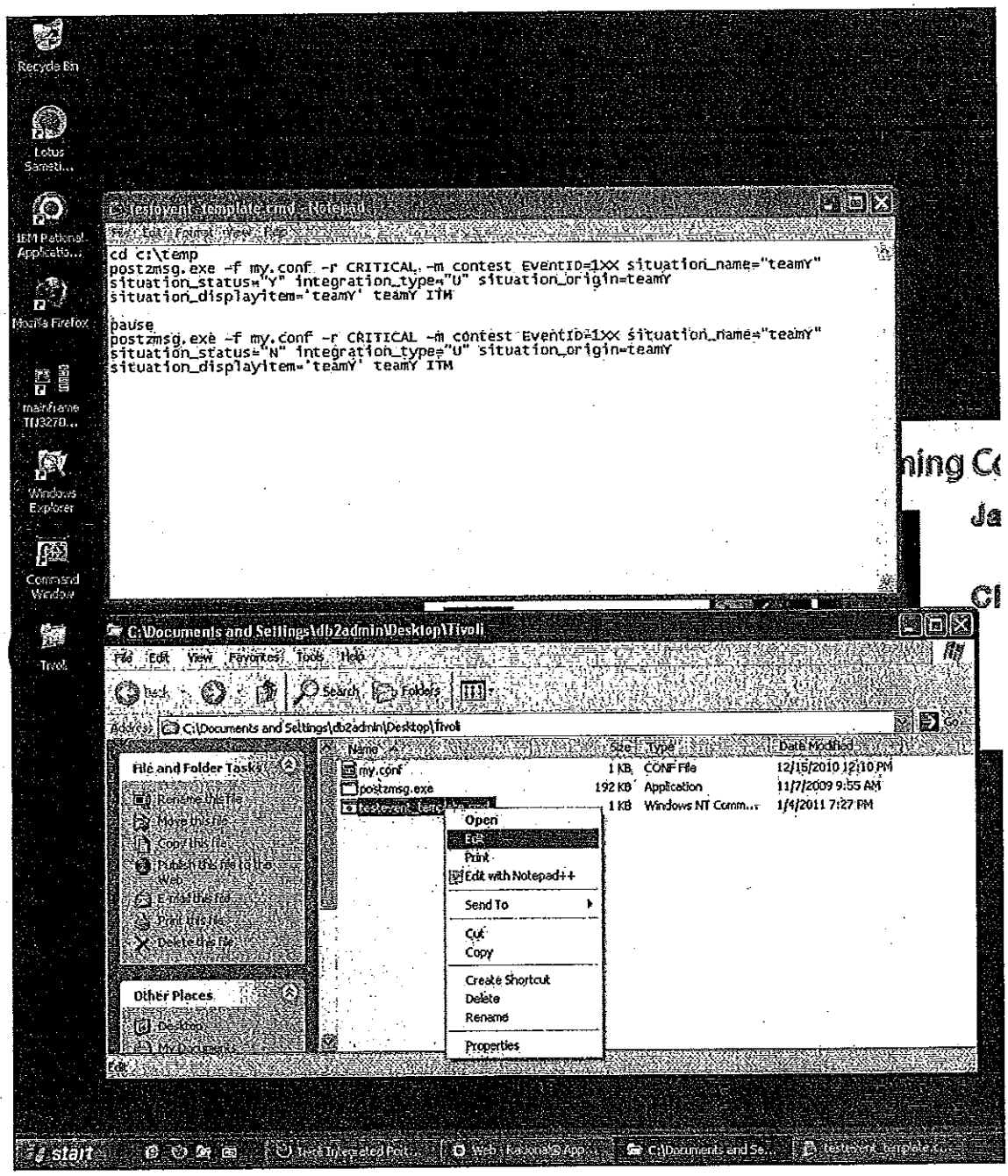
Exercises

1. Review the server interface of the Omnibus server as provided by the instructor



All the traffic lights in the tables are green in color. Now you will write a script to change the color of your table to red and then back to green.

2. Double click the folder "Tivoli" on the desktop, then select the file "testevent_template.cmd" then right clicked it and then press "Edit":



then an editor pops up with the content of the script file displayed in the editor.

3. You can see there are two commands in the script file. The first command is to change the traffic light of team<Y> to red, and the second command is to change the traffic light of team<Y> back to green.

4. Modify the two commands by change Y to your team number, and XX = Y with a prefix 0 if Y is single digit. For example, if Y = 2, then XX = 02. Then the two commands will become:

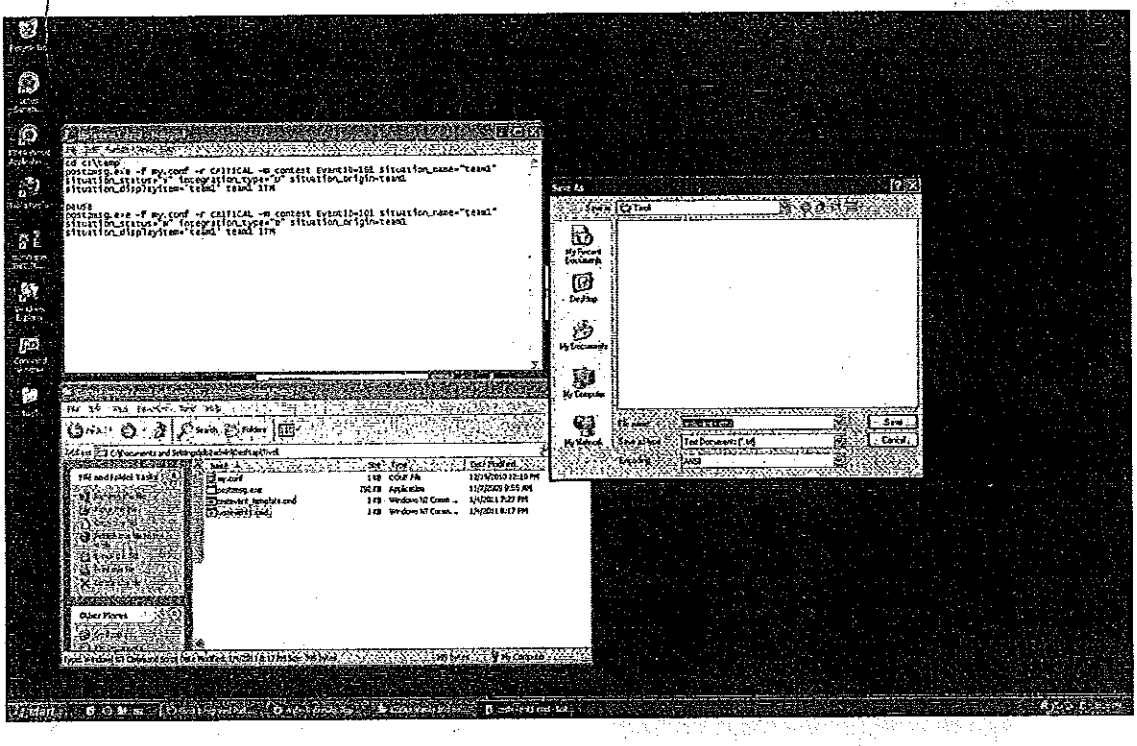
    postzmsg.exe –f my.conf –r CRITICAL –m contest EventID=101
    situation_name="team1" satuation_status="Y" integration_type="U"
    situation_origin=team1 situation_displayitem='team1' team1 ITM

    and

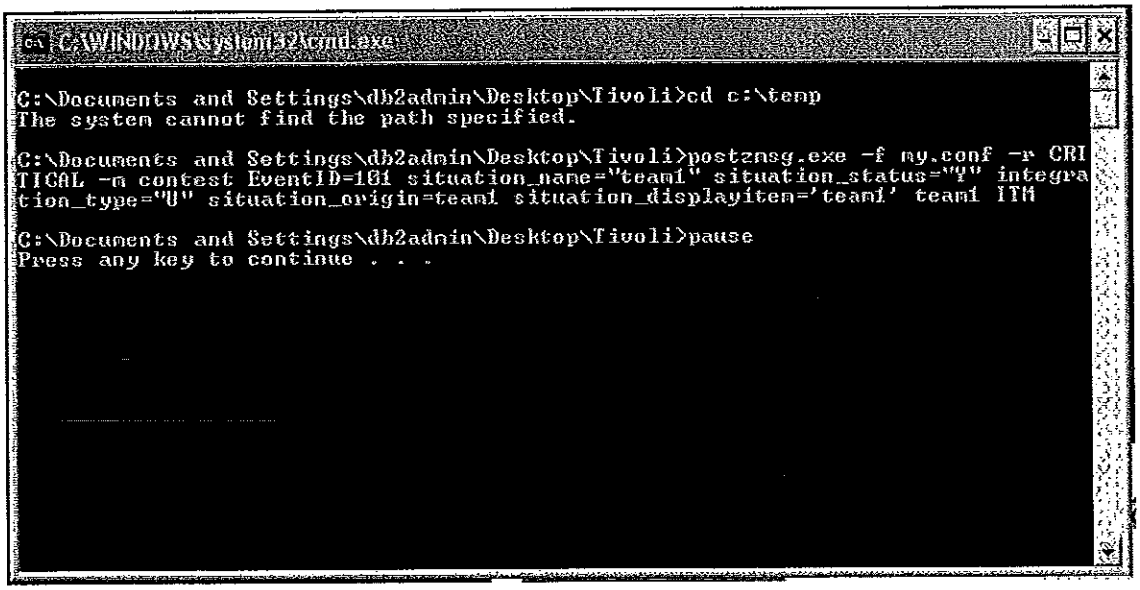
    postzmsg.exe –f my.conf –r CRITICAL –m contest EventID=101
    situation_name="team1" satuation_status="N" integration_type="U"
    situation_origin=team1 situation_displayitem='team1' team1 ITM

    respectively.

5. After the two commands are changed, run the command "File"->"Save As" from the editor and give a new name (e.g., testevent1.cmd)::



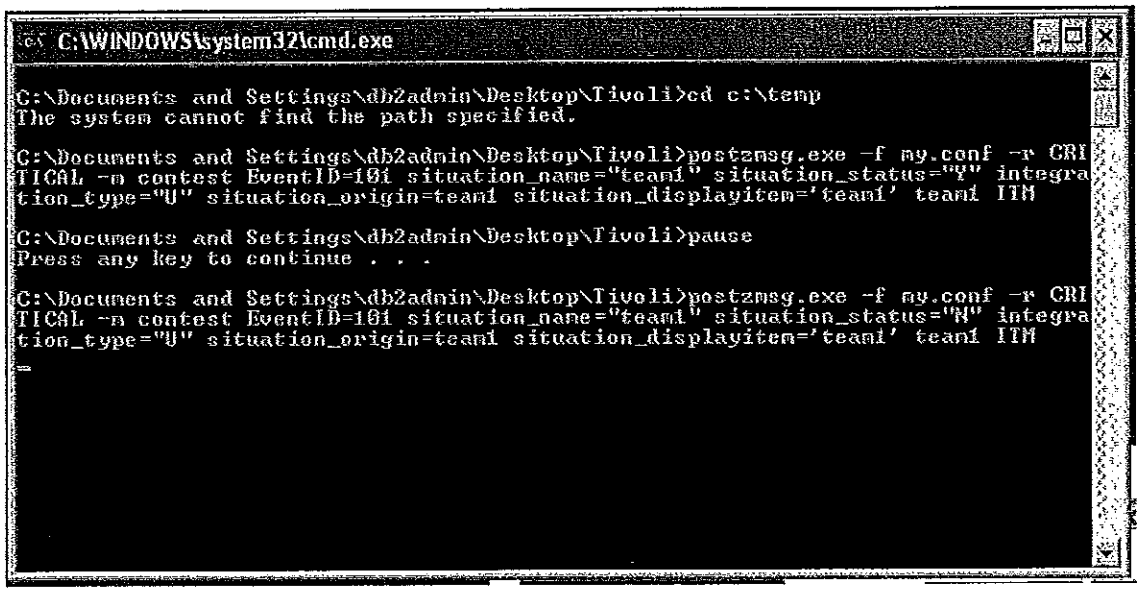
6. Double click the file "testevent1.cmd" to run the script file. You will see the following window:



The first command is run and then it will pause and wait for your input. Don't press any key right now.

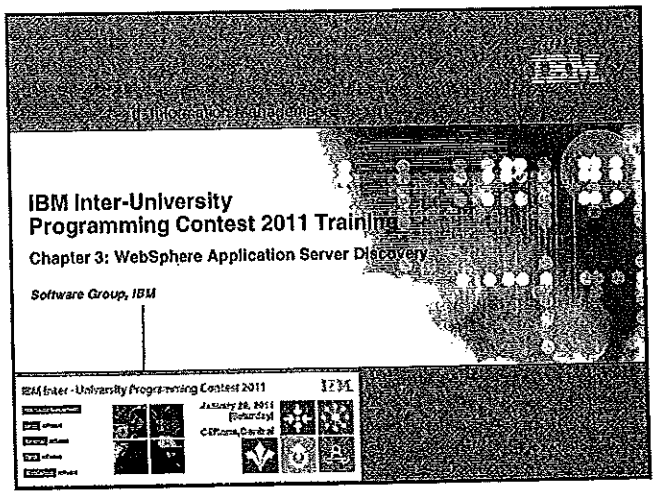
7. Look at the Omnibus server screen. Wait until the traffic light of your team changes to red. The screen will be refreshed every 5 seconds.

8. Press enter on the script window. The second command is run now and you will see the following output:



9. Look at the Omnibus server screen. Wait until the traffic light of your team changes back to green. The screen will be refreshed every 5 seconds.

10. This is the end of this exercise.



Agenda

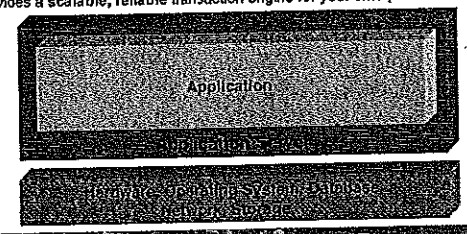
- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

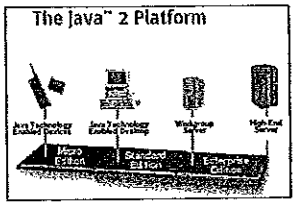
What is an Application Server?

- Provides the infrastructure for running applications that run your business
  - Insulates applications from hardware, operating system, database, network...
  - Provides a common environment and programming model for applications
    - Write once, run anywhere (WORA)
    - Platform for developing and deploying Web Services
  - Provides a scalable, reliable transaction engine for your enterprise



The Java Platforms

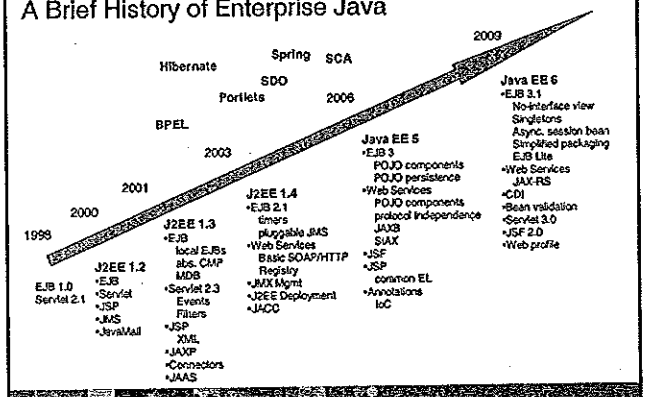
- **Java 2 Micro Edition (J2ME)**
  An optimized Java runtime environment for the consumer space. It covers the range of extremely tiny devices such as smart cards or a pager all the way up to the set-top box.
- **Java 2 Standard Edition (J2SE)**
  A complete environment for applications development on desktops & servers.
  - Java Runtime Environment, Standard Edition (JRE) - provides the Java APIs, Java virtual machine, and other components necessary to run applets and applications written in the Java programming language
  - Java Software Development Kit, Standard Edition (SDK) - A superset of the JRE that includes tools such as the compilers and debuggers necessary for developing applets and applications.
- **Java Enterprise Edition (JEE) – formerly J2EE**



Java Enterprise Edition (JEE)

- Built upon J2SE, JEE provides the specifications for developing multi-tier enterprise applications with Java
- Accelerates and simplifies enterprise application development
  - With applications based on standardized, modular components
  - Providing a complete set of services to those components
    - Standard APIs for common services such as database access, transaction management, messaging, etc.
  - Handling many details of application behavior automatically, without complex programming (e.g. transaction management, security)
  - Providing a foundation for the third party component market (e.g. JCA, JMS, JDBC)
- Improves systems and operations management by providing:
  - Packaging, deployment, and management standards for enterprise applications
- Fulfills the promise of true portability
  - Hardware, operating system and vendor

A Brief History of Enterprise Java



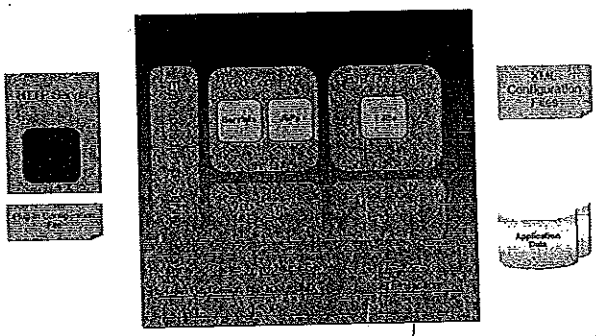
JEE and WebSphere Application Server

- Application servers, based on standards, can be perceived as commodities
- WebSphere Application Server (WAS) delivers more value to customers than just standards compliance
  - Quality of service, such as
    - Workload management
    - High availability and failover
    - Performance
  - Tooling support
    - Runtime embedded in Application Developer products
  - Programming model extensions
    - Future standards
    - Customer identified critical functions
  - Foundation for evolving WebSphere Software Platform

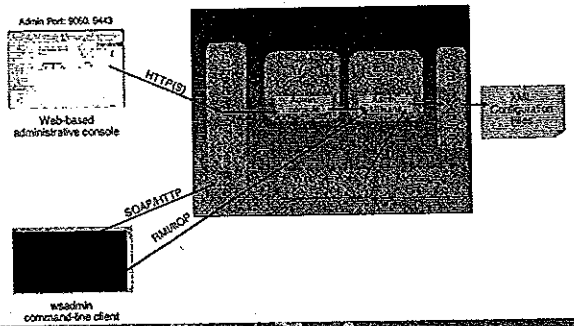
Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

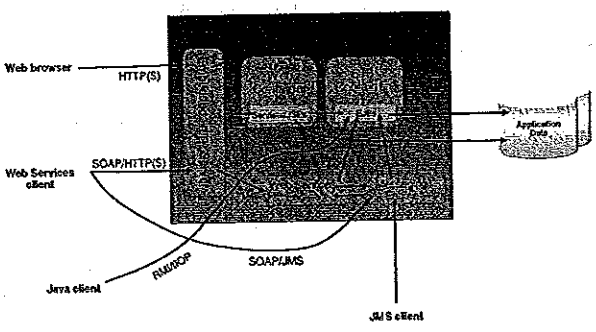
Basic Architecture

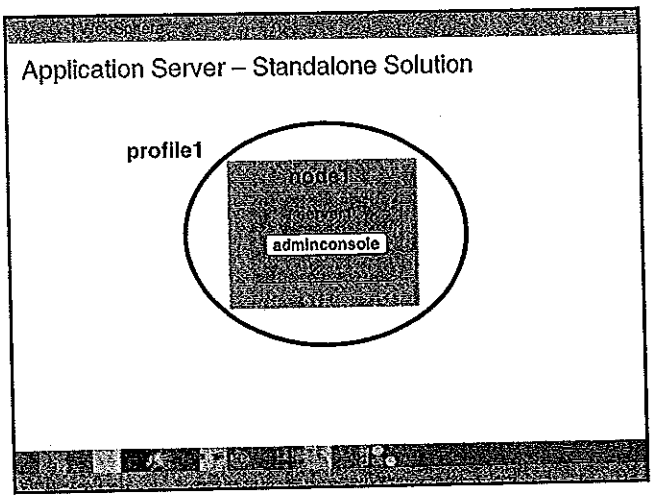


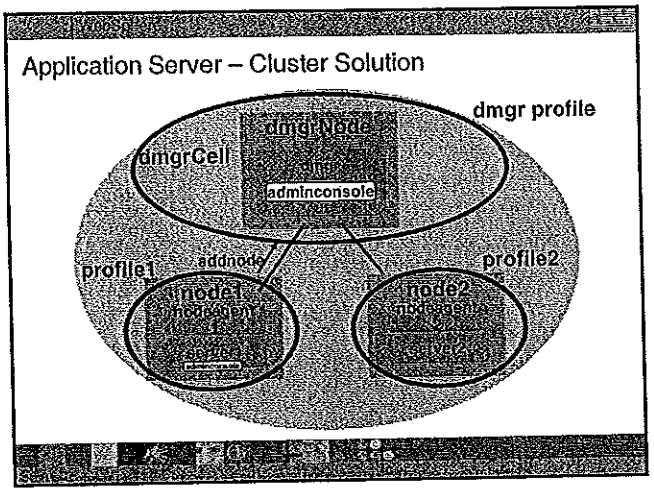
Administering the Server

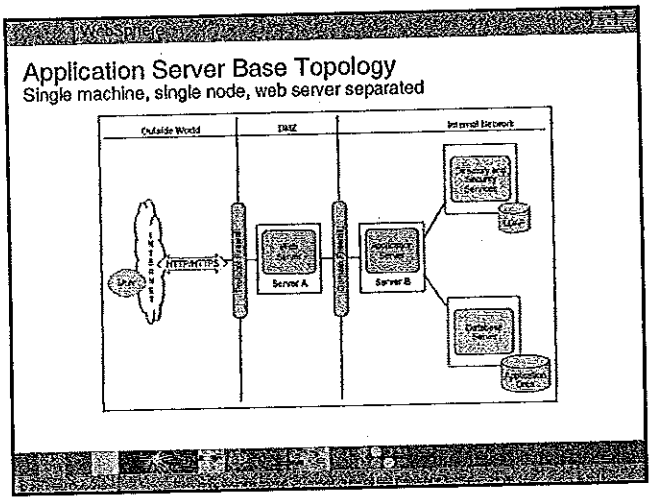


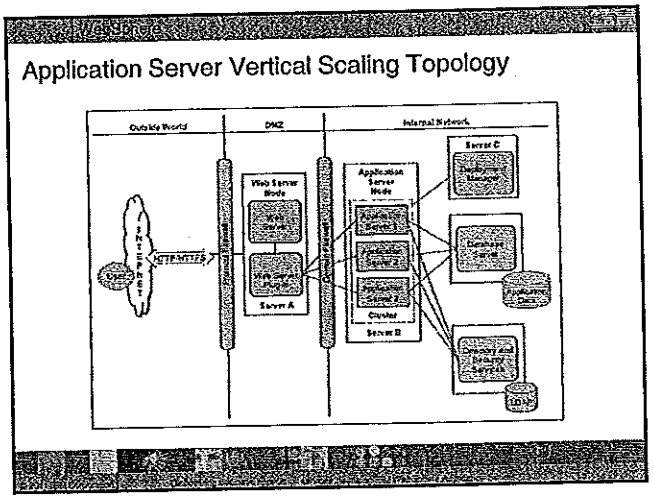
Accessing Server Resources



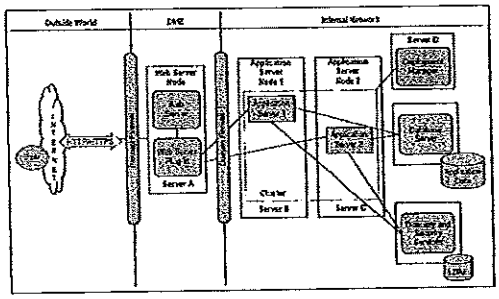








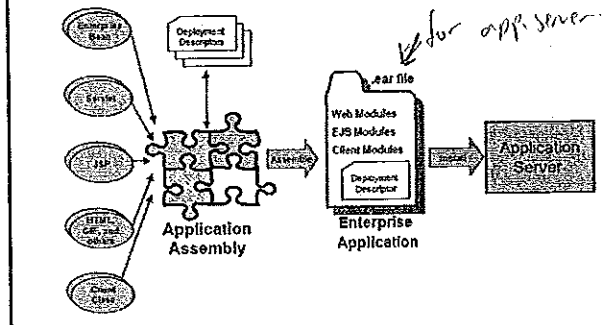
Application Server Horizontal Scaling Topology



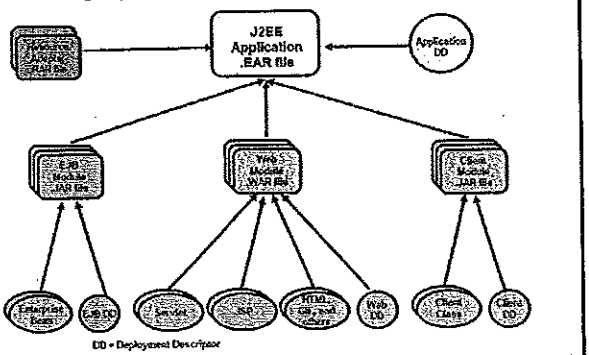
Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

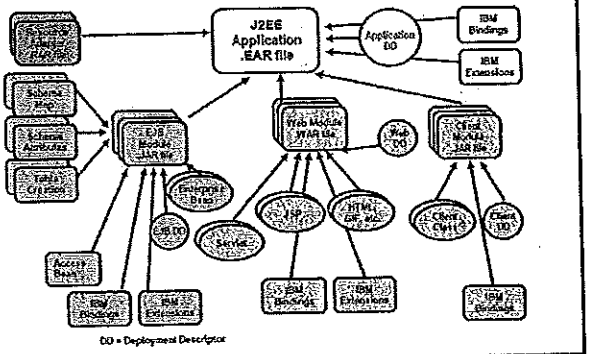
Overview of Application Assembly and Installation



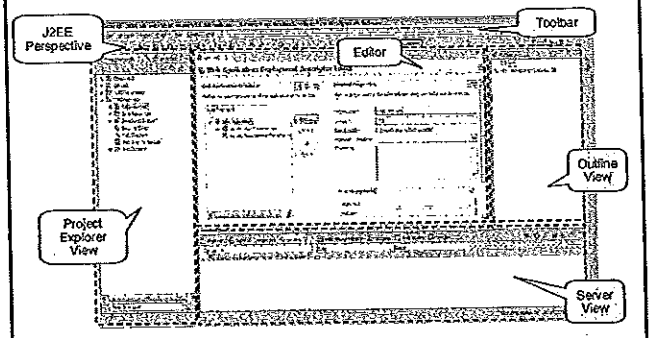
JEE Packaging



WebSphere Application Packaging



The Application Server Toolkit Workbench



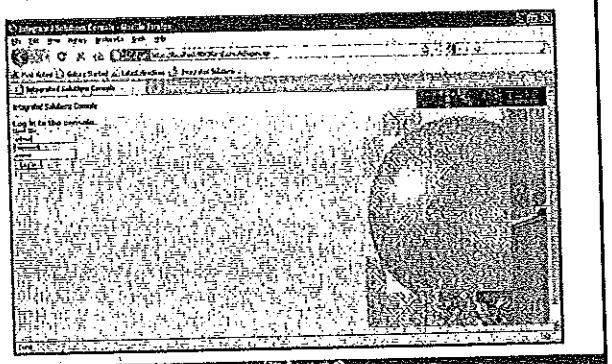
Agenda

- What Is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

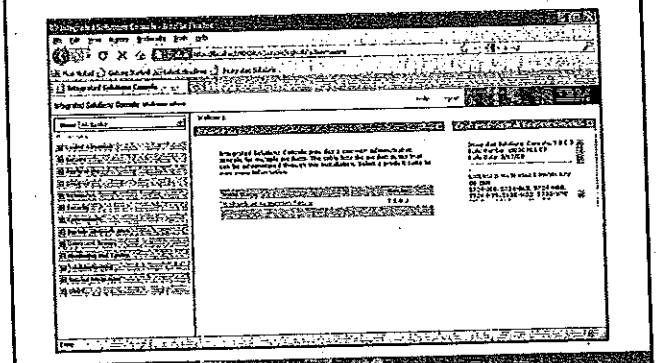
System Administration Tools

- WebSphere Integrated Solutions Console
  - Browser-based interface
- Command-line operation tools
  - Available in the bin directory
- wsadmin scripting
  - Interactive and batch modes
  - Supports JACL or Jython scripts
- Ant
  - Java-based build and automation tool
- Java-based JMX APIs
  - Programming interface for custom Java applications

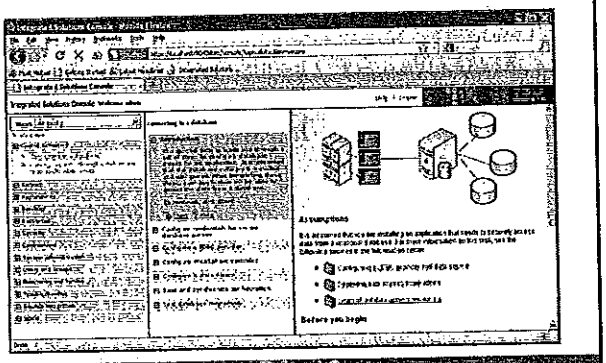
Integrated Solutions Console



Integrated Solutions Console – Welcome Page



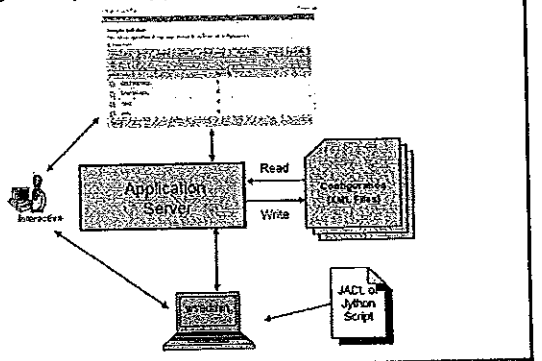
Integrated Solutions Console – Cheat Sheet



Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

Installing Enterprise Application



Installation Tasks

- Configure the application environment as required
  - Variables, virtual hosts, classpath, security, and so forth
- Configure application resources
  - JDBC provider, DataSources
- Install application

Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

Logging and Tracing

- Message logging (messages) and diagnostic trace (trace) are conceptually similar since they are using the same Java logging API
- They have important differences:
  - A message entry is an informational record that is intended for end users, systems administrators, and support personnel to view. The text of the message must be clear, concise, and interpretable by an end user.
  - A trace entry is an information record that is intended for service engineers or developers to use. As such, a trace record might be considerably more complex, verbose and detailed than a message entry.

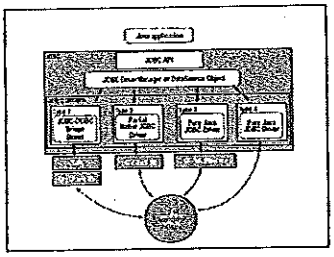
Log Files and Locations

- The destination and names for the log files are configurable.
  The default location is:
  <was_root>\profiles\<profile_name>\logs\<server_name>
- Log Files:
  - SystemOut.log and SystemErr.log - Standard JVM output and error log
  - startServer.log and stopServer.log
    - Startup and shutdown of the Application Servers
  - activity.log - Events that show a history of installation activities
    - Use Log Analyzer to read output from this file
  - trace.log – Output from diagnostic trace
    - Destination and name are configurable
  - http_plugin.log – Not in <was_root>
    - Location: <plugin_root>\logs\<webserver_name>

Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

JDBC Drivers and DataSource



- DataSource represents a real-world data source, e.g. DB2 for z/OS, in the Java programming language
- DataSource is registered with JNDI naming service for use in a JEE environment, e.g. WebSphere Application Server

Types of JDBC Driver

Type-1	Type-2	Type-3	Type-4
•Oldest type •JDBC API calls ODBC •No good performance •Not commonly used •Partially written in Java	•Platform- and Database-specific •Best performance •Partially written in Java •Platform-specific written in C	•Middleware translates JDBC API to database-specific call thru TCP/IP •No driver for DB2 for z/OS	•Full written by Java •Portable •DB2 Universal Driver for Java Common Connectivity (JCC) Pure Java (will though TCP/IP

- Only Type-2 and Type-4 drivers are supported by DB2 for z/OS.

Agenda

- What is an Application Server?
- Application Server Architectural Overview
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- JDBC Provider and DataSource
- Distributed Transaction
- Lab (Review WAS-lab application)

What is a Transaction?

- A financial program might transfer money from a savings account to a trading account by using the pseudocode below:
  - *begin transaction*
  - *debit savings account*
  - *credit trading account*
  - *commit transaction*
- Either all or none of the two steps must complete. Otherwise, data integrity is lost. Because the steps within a transaction are a unified whole, a transaction is often defined as an indivisible *unit of work*.
- A transaction can end in two ways:
  - with a commit for a successful transaction
  - with a rollback for a failed transaction
- As shown, the begin and commit statements mark the boundaries of the transaction, which is declared by application developer.

Code Pattern

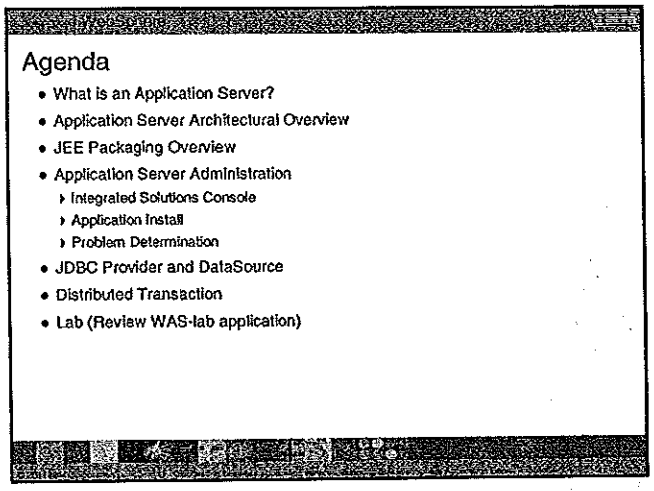
- javax.transaction.UserTransaction interface is used to mark the boundaries of a JTA transaction.
- Code pattern of a distributed transaction under an application server:

```
// Obtain UserTransaction object from the application server
InitialContext ctx = new InitialContext();
javax.transaction.UserTransaction utx = (UserTransaction) ctx.lookup("java:comp/UserTransaction");
// Use the begin method on a UserTransaction object to indicate
// the beginning of a distributed transaction.
utx.begin();
...
// Execute some SQL statements with one Connection object.
// Do not call Connection methods commit or rollback.
...
// Use the commit method on the UserTransaction object to
// drive all transaction branches to commit and indicate
// the end of the distributed transaction.
utx.commit();
```

- To rollback a transaction, use utx.rollback() instead.

Distributed Transaction in JEE

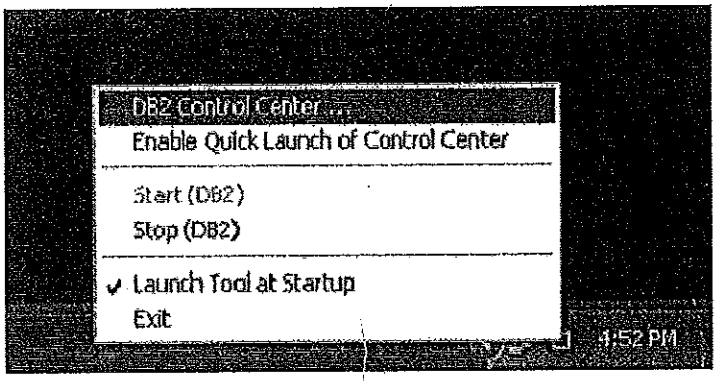
- Distributed transactions typically involve multiple connections to the same data source or different data sources from the same or different manufacturers.
- J2EE supports distributed transactions through Java Transaction API (JTA) and Java Transaction Service (JTS) specifications.
- JTS is seldom manipulated directly by application developers.
- JTA specifies standard Java interfaces between a *transaction manager* and the parties involved in a distributed transaction system: the *resource manager*, the application server, and the transactional applications. For example:
  - DB2 acts as a resource manager.
  - WAS assumes the role of a transaction manager.
- JDBC drivers with distributed transaction capability are required. For example:
  - com.ibm.db2.jdbc.DB2XADataSource class should be used when defining DB2 data sources involved in a distributed transaction within the WAS.



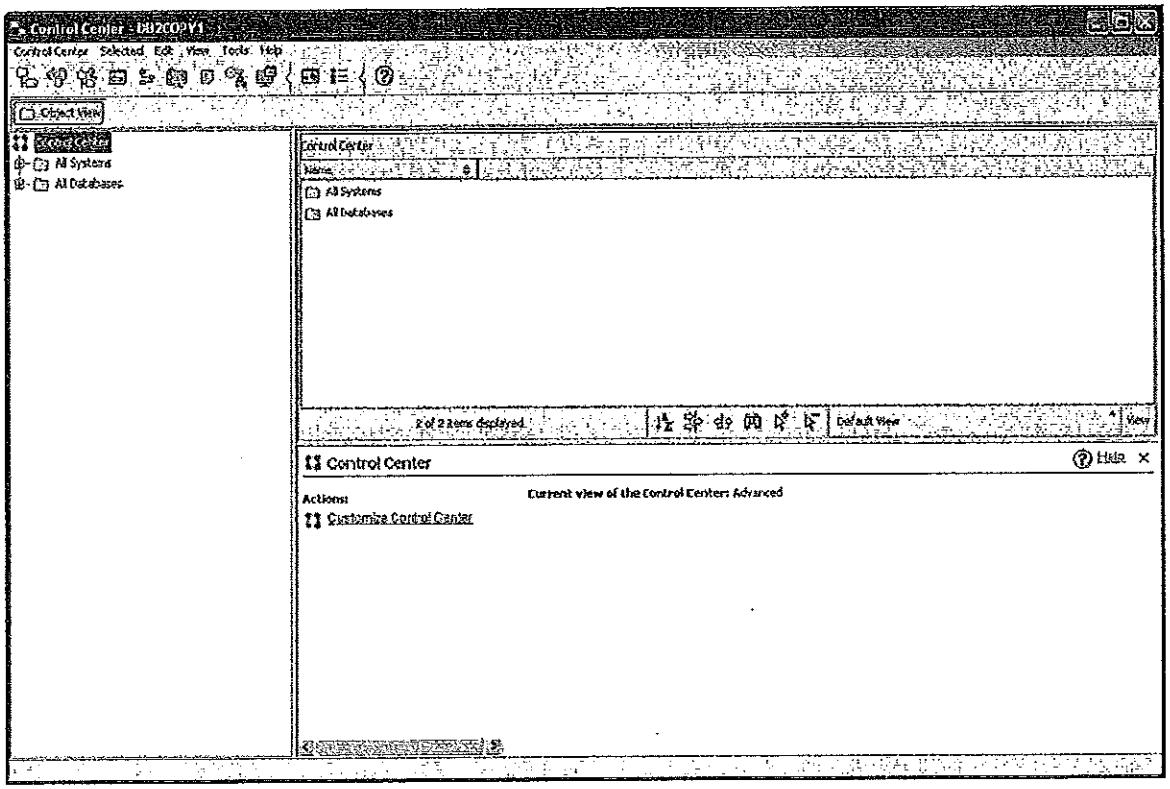
Chapter 3: Step by Step Lab Instructions for WAS-lab

Task A – Setup DB2 Database Tables

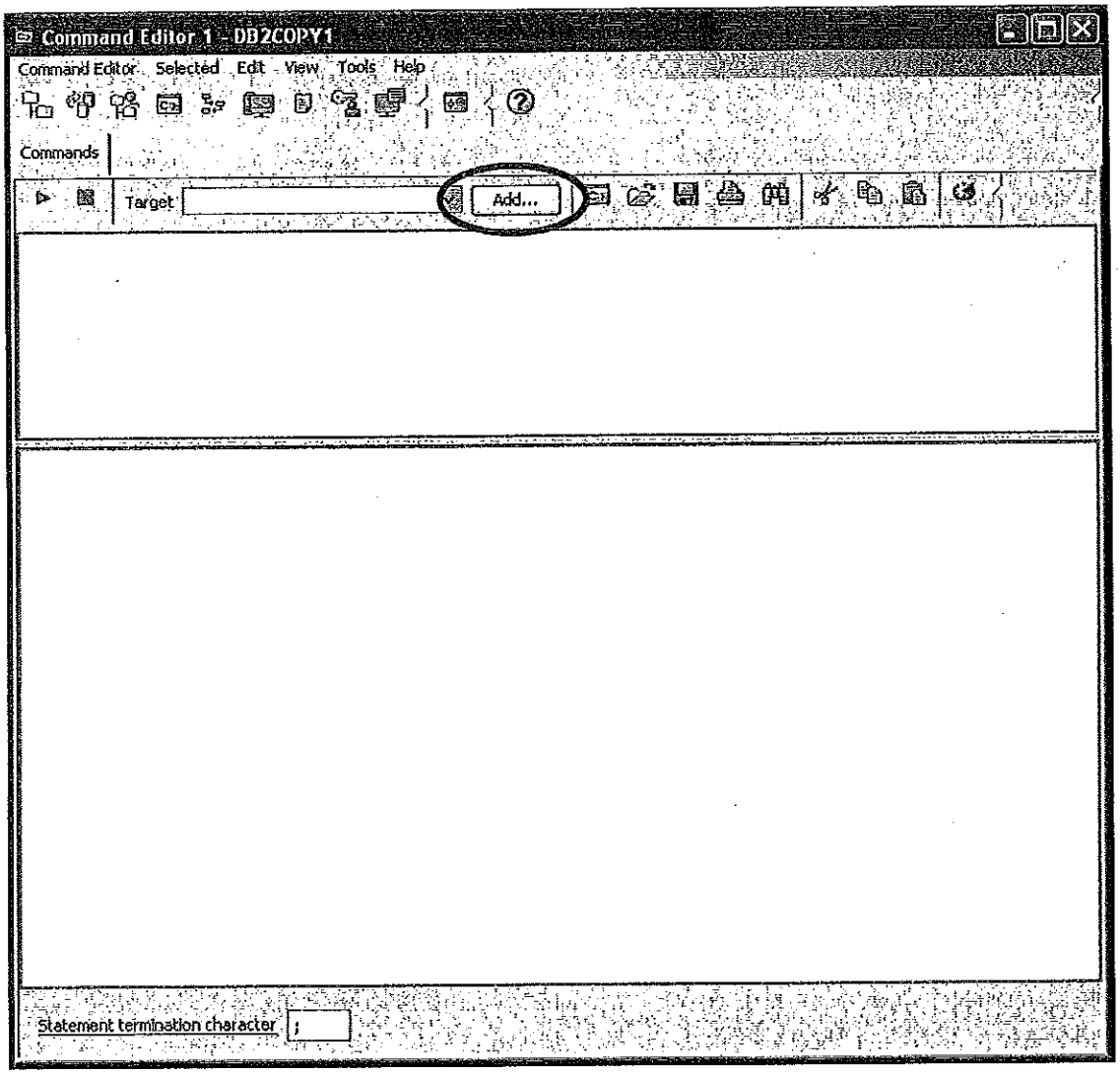
1. Right click on the DB2 tray icon on the taskbar and select **"DB2 Control Center"** .



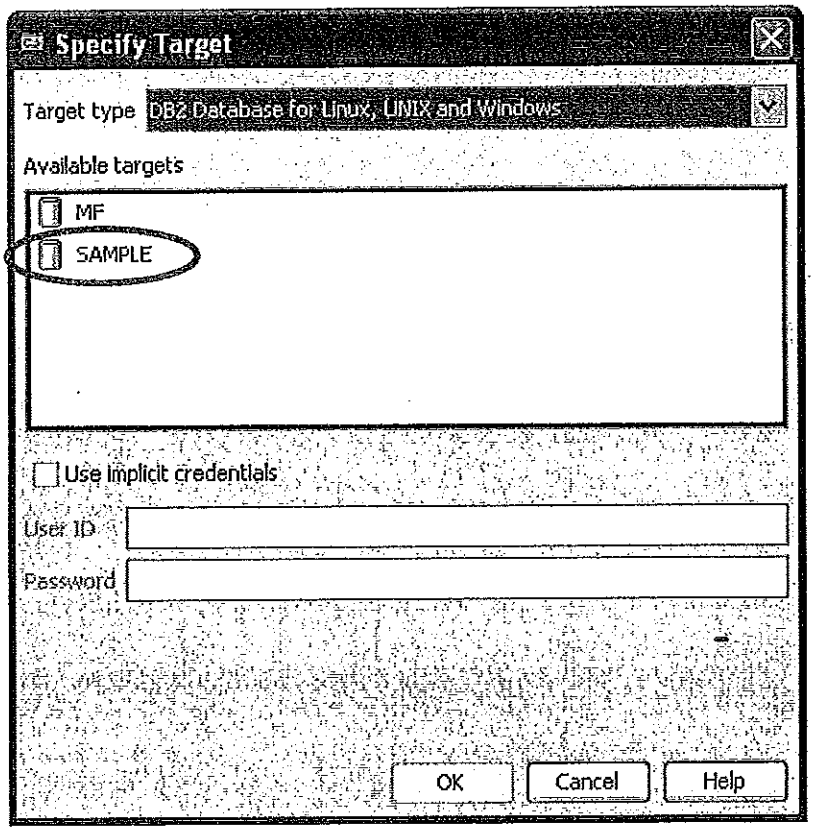
2. A dialogue called **"Control Center View"** is shown, click **"OK"** . Then the Control Center shows.



3. On the menu bar, click **"Tools → Command Editor"** and another Command Editor window will be shown. Click **"Add"** button.

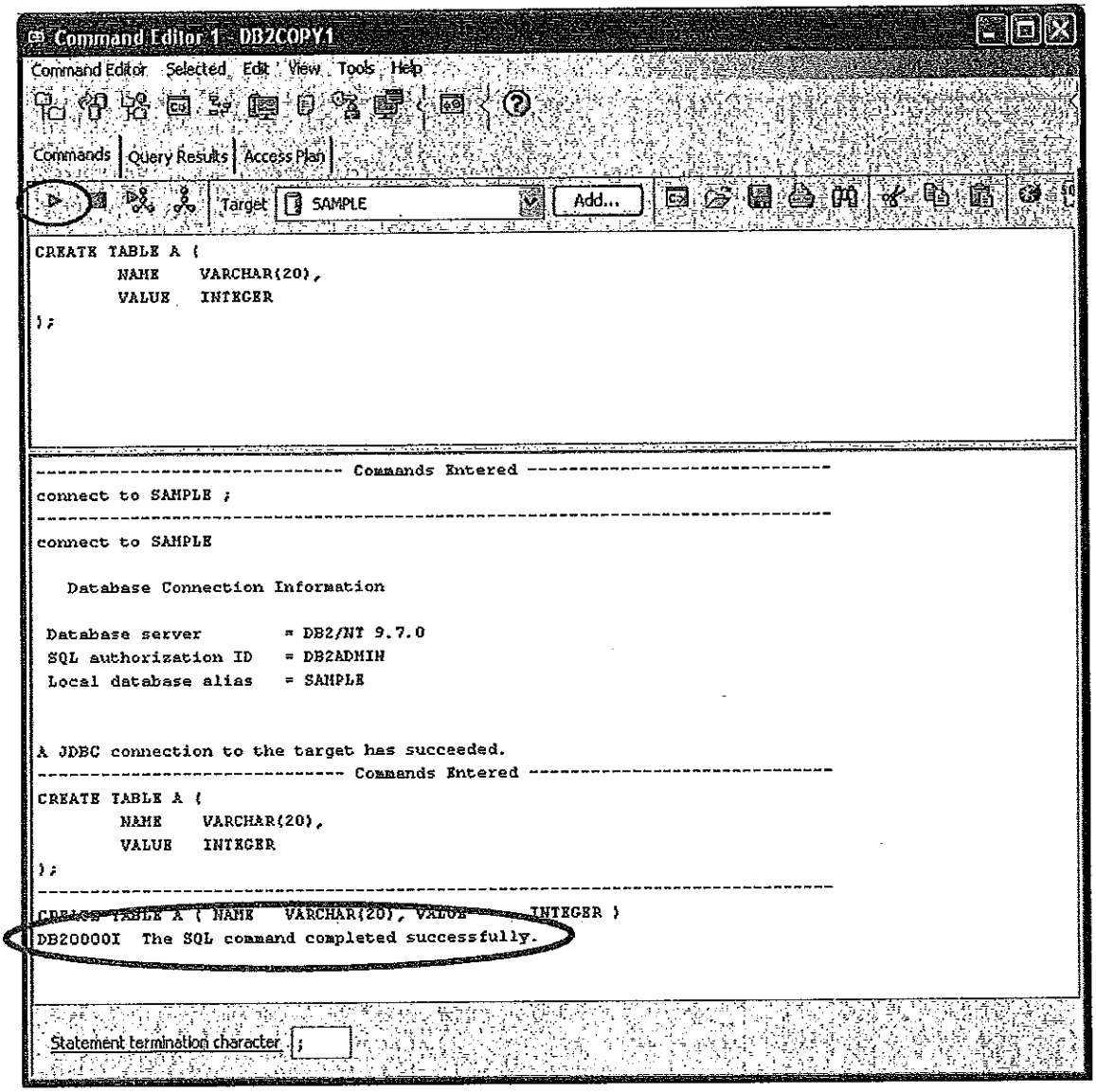


4. Select "**SAMPLE**" from the "**Specify Target**" dialogue box.



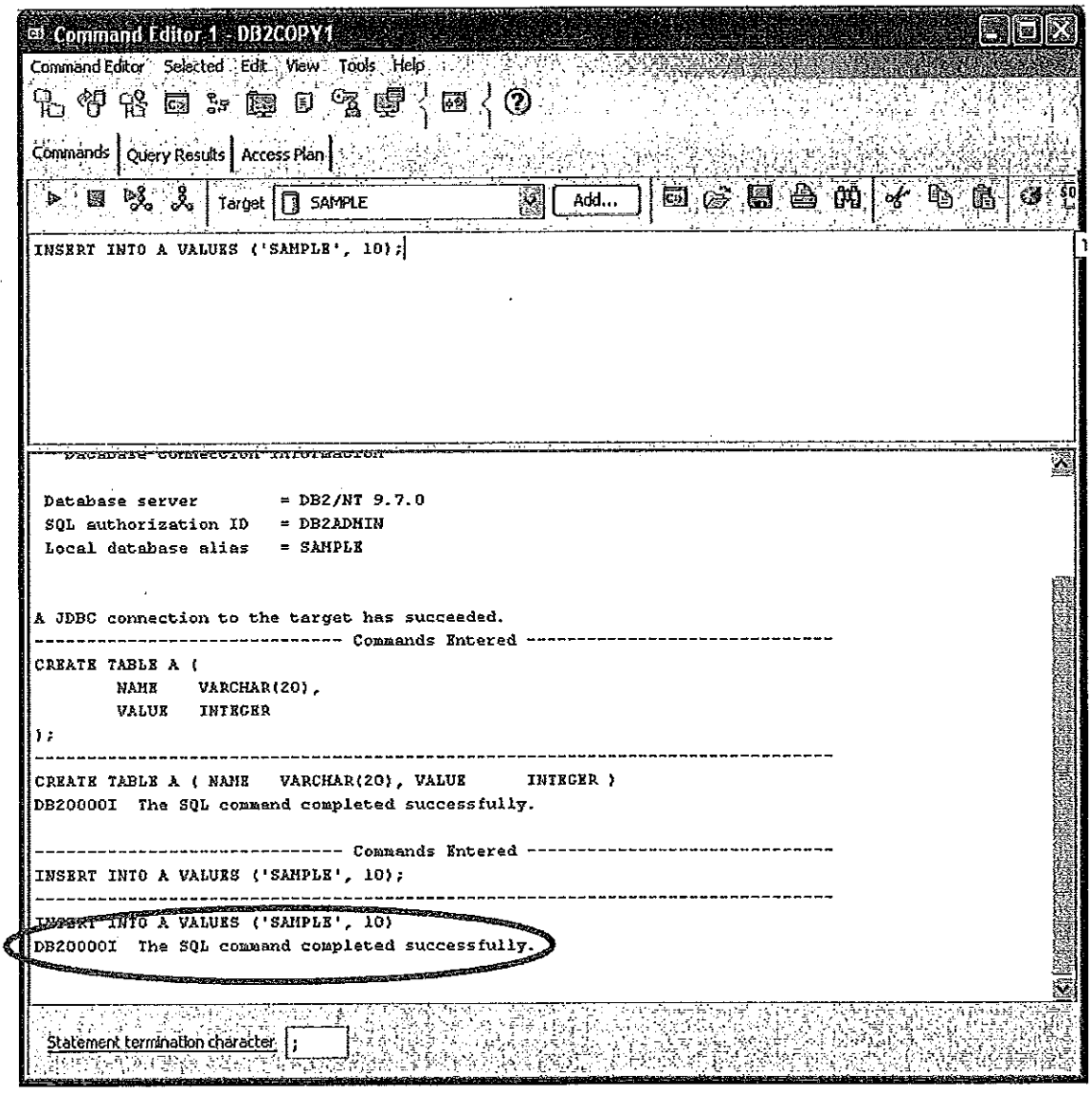
5. Enter the following SQL statement in the Command Editor and click the "**Run**" button. Check to see if "**The SQL command completed successfully.**" appears on the status window.

```
CREATE TABLE A (
        NAME VARCHAR(20) NOT NULL,
        VALUE INTEGER,
        PRIMARY KEY (NAME)
);
```

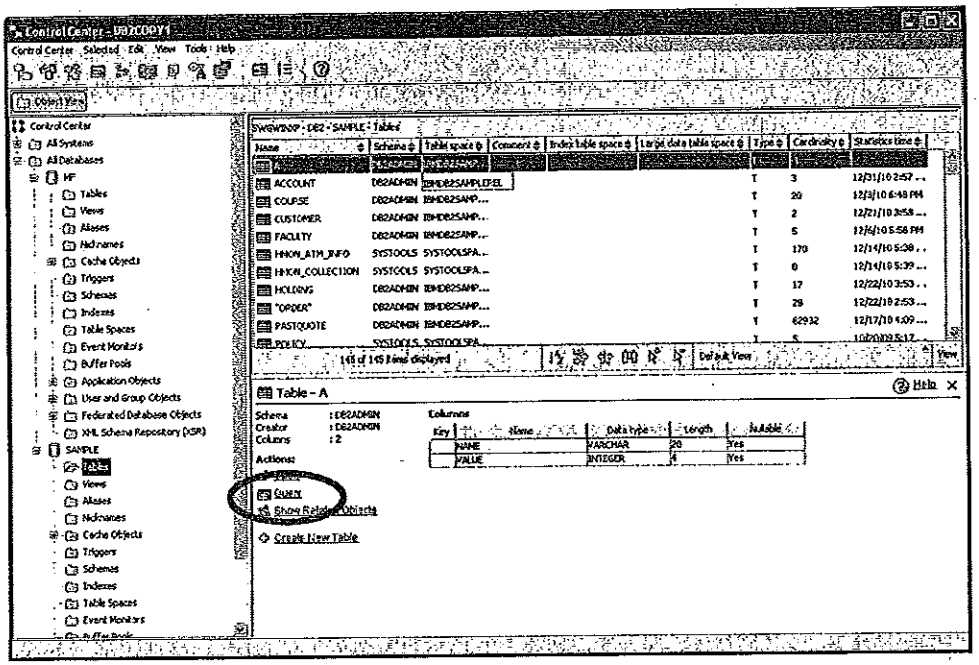


6. Enter the following SQL statement in the Command Editor and click the "**Run**" button. Check to see if "**The SQL command completed successfully.**" appears on the status window.

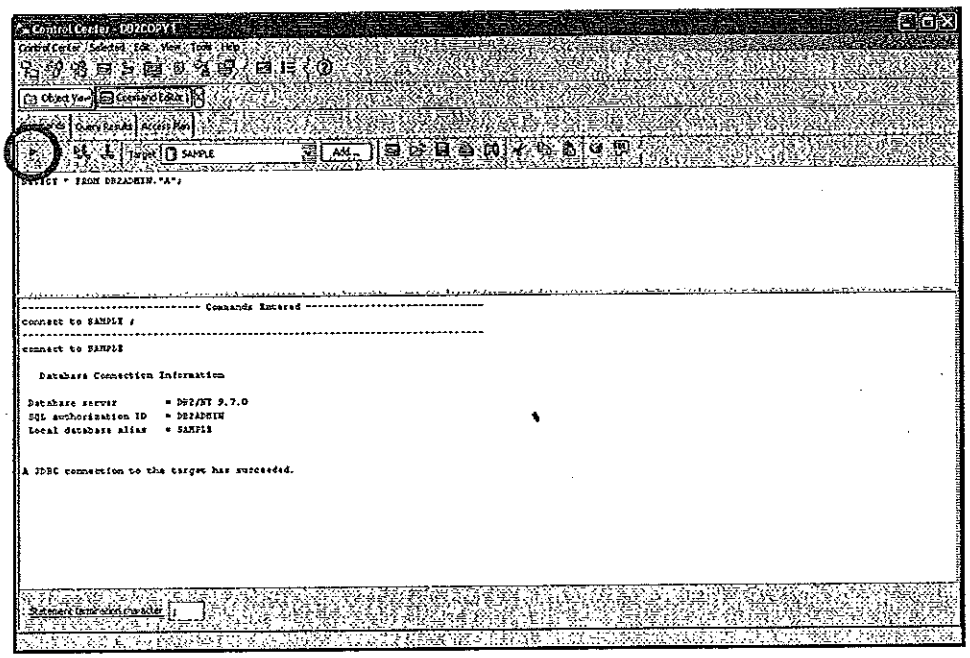
INSERT INTO A VALUES ('SAMPLE', 10);



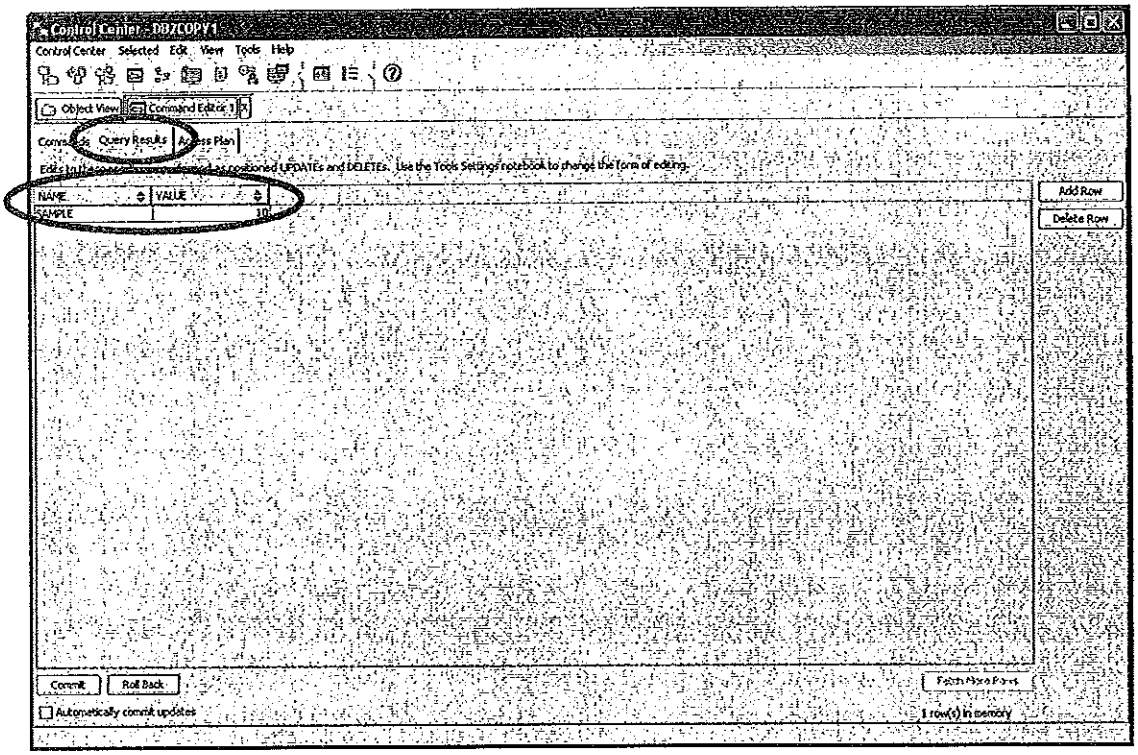
7. Close Command Editor by clicking **Command Editor → Exit** on the menu bar. In Control Cetner, expand **All Databases → SAMPLE → Tables** by clicking the (+) icon in the **Object View**. On the right panel, the created table A should be there. Click to select table A and then click **Query** link on the bottom.



8. A Command Editor will be shown again with a SQL statement already there. Click the **Run** button.



9. The **Query Results** tab will be shown with the records of table **A** displayed in a table. The field values should be consistent with the previous insert SQL statement. Close Command Editor by clicking **Command Editor** --> **Exit** on the menu bar.



10. Using similar steps from 3 to 9 above to prepare another table, called **B**, in the database **MF** with reference to the following SQL statements.

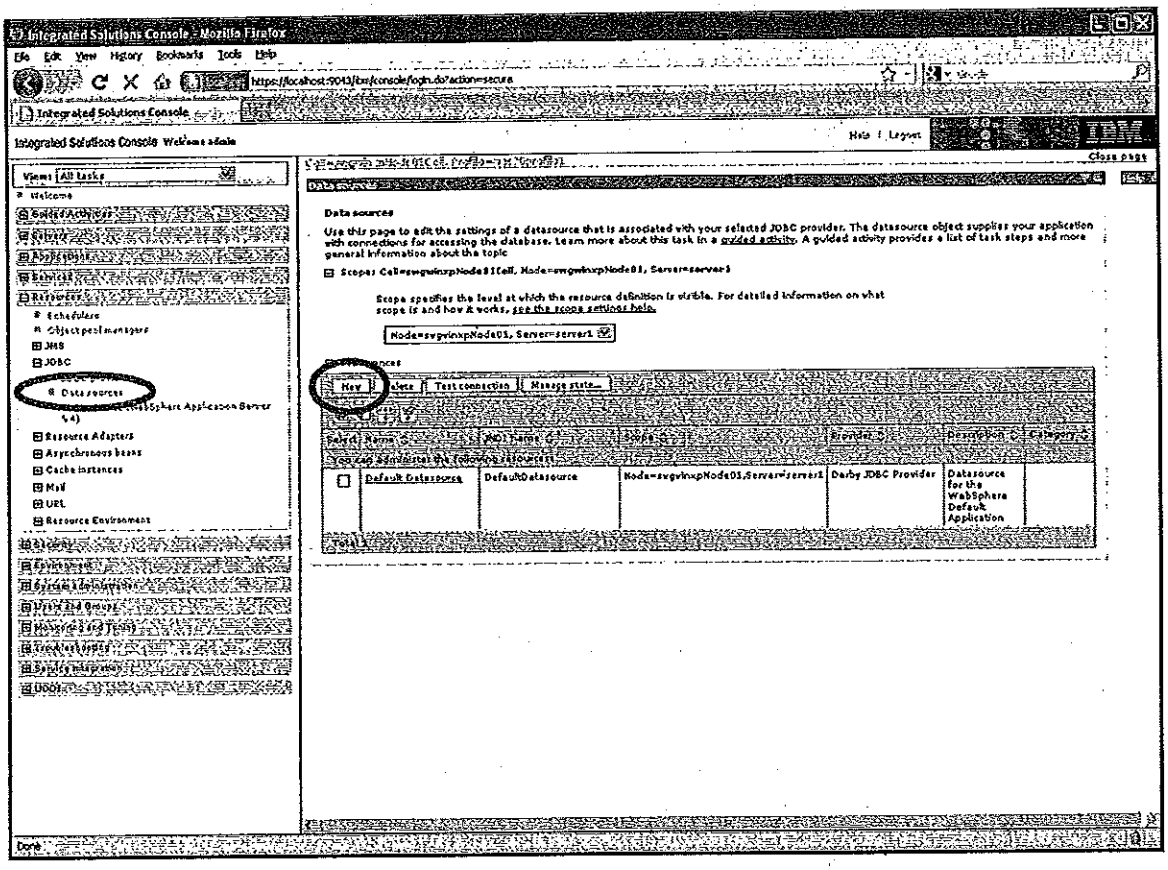
```
CREATE TABLE B (
        NAME VARCHAR(20) NOT NULL,
        VALUE INTEGER,
        PRIMARY KEY (NAME)
);

INSERT INTO B VALUES ('MF', 20);
```

Congratulations, you have finished Task A of this lab. Go ahead to Task B.

Task B – Configure Data Sources in WebSphere Application Server

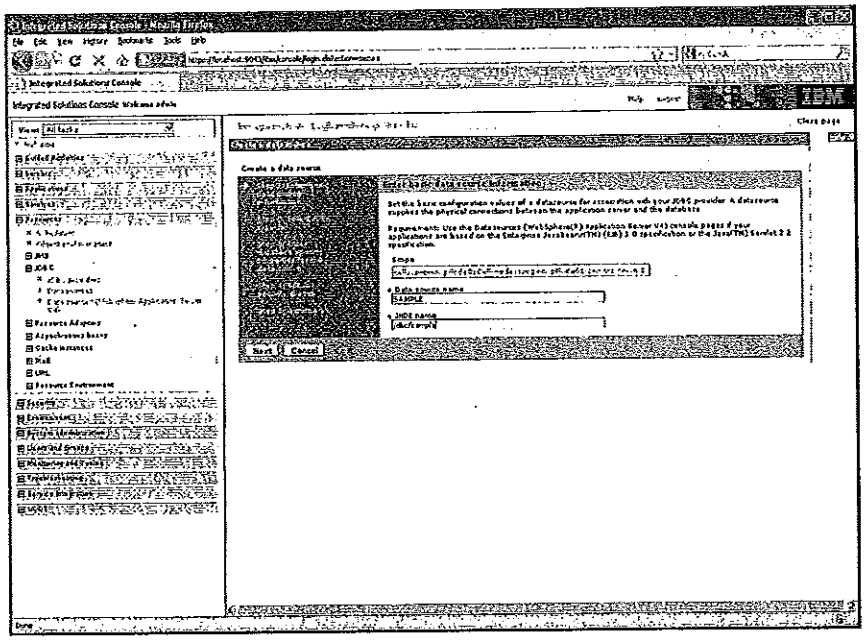
1. Open FireFox and enter the link (http://localhost:9060/ibm/console). Log in to the Integrated Solutions Console. Enter **admin** as username and **admin** as password, click **Log in**. The **Welcome** page shows.

2. Click **Resources → JDBC → Data sources**. Click the **New** button on the right panel.

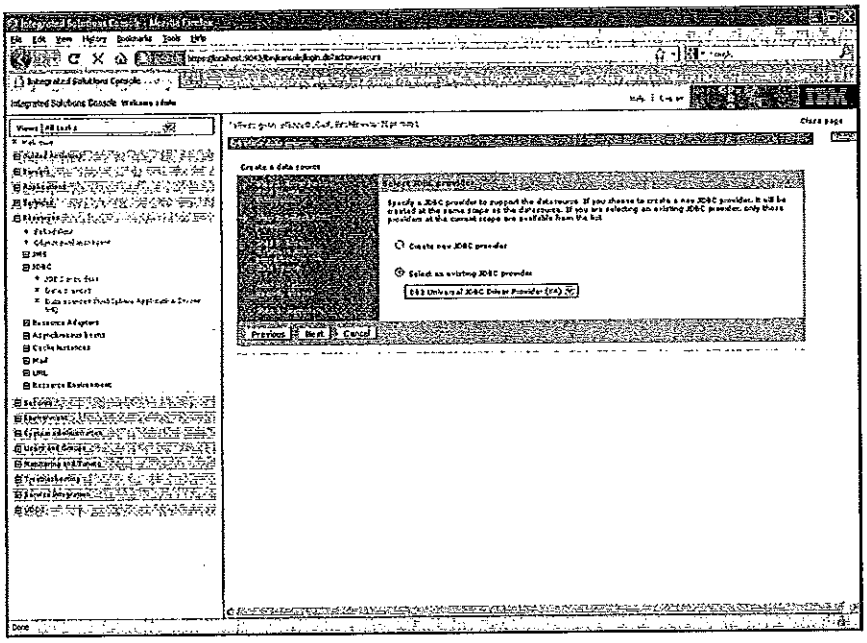


3. Enter the following details and click **Next.**

   Data source name: **SAMPLE**
   JNDI name: **jdbc/sample**

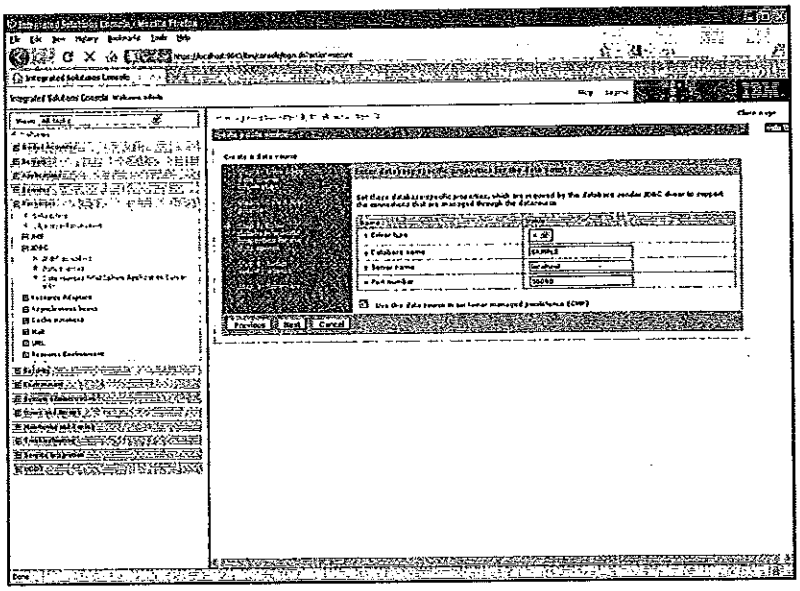


4. Select **DB2 Universal JDBC Driver Provider (XA)** in **Select an existing JDBC provider.** Click **Next.**.



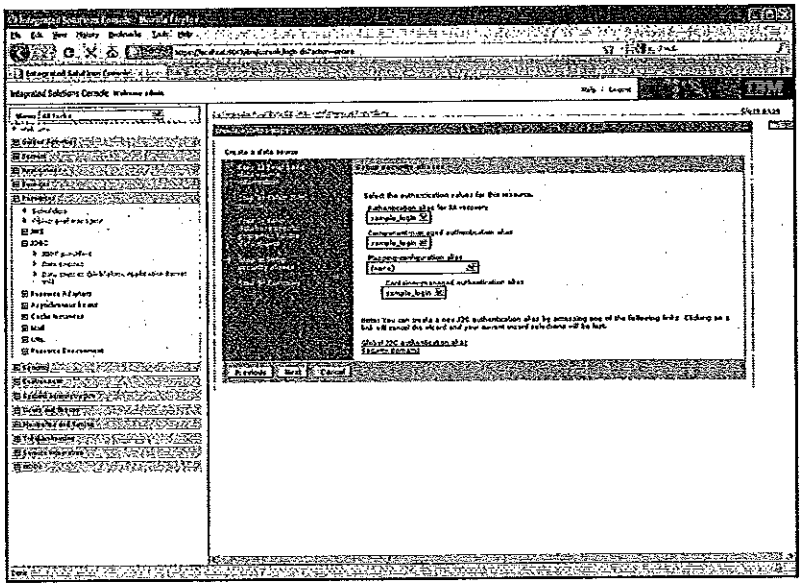
5. Enter the following details and click **Next.**

Driver type: **4**
Database name: **SAMPLE**
Server name: **localhost**
Port number: **50000**

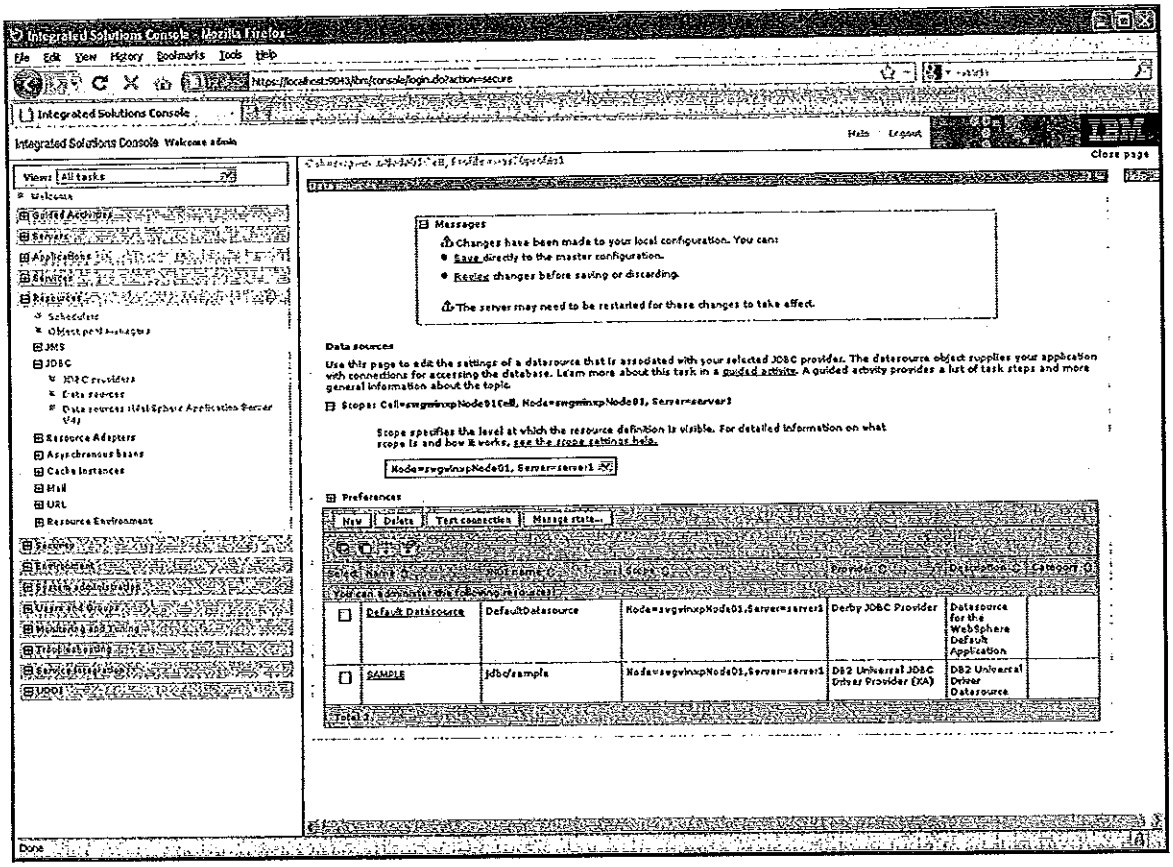


6. Select the following details and click **"Next"**.

Authentication alias for XA recovery: **sample_login**
Component-managed authentication alias: **sample_login**
Mapping-configuration alias: **(none)**
Container-managed authentication alias: **sample_login**



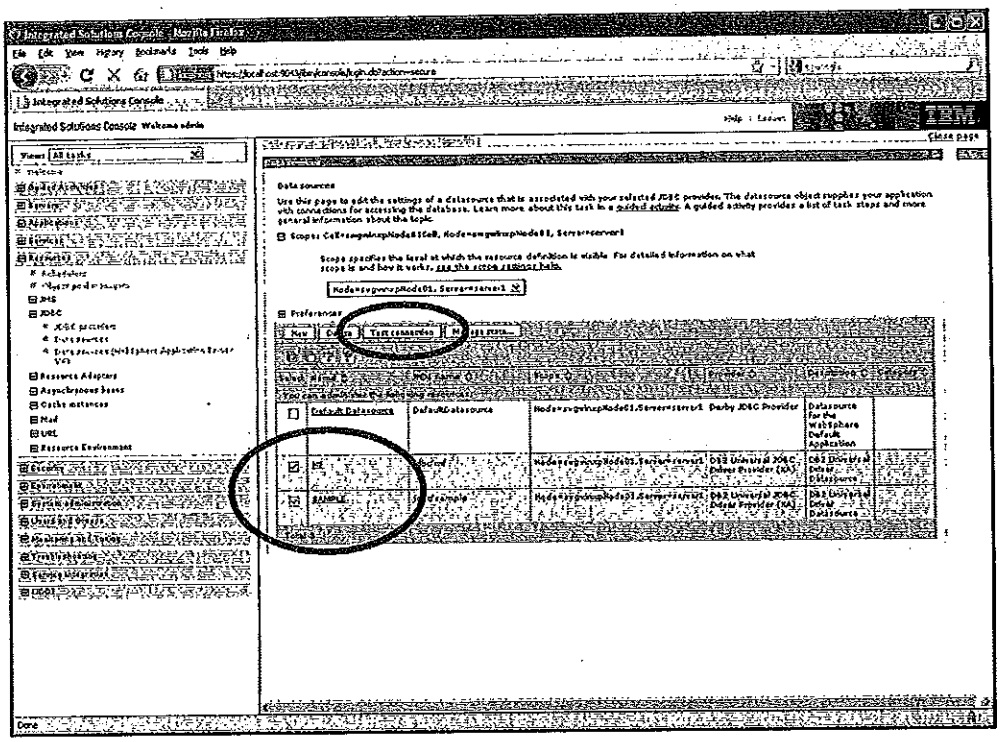
7. In **Step 5: Summary**, click **Finish**. Then click **Save** on the top of the screen to make the setting effective.



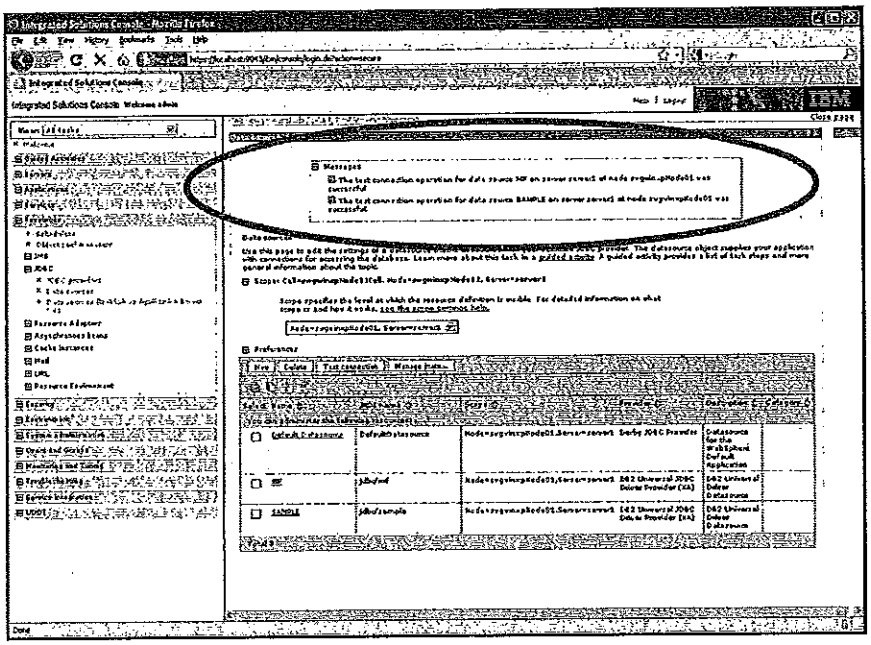
8. Using similar steps from 2 to 6 above to prepare another data source, called **MF**, with reference to the following details.

Data source name: **MF**
JNDI name: **jdbc/mf**
Driver type: **4**
Database name: **MF**
Server name: **localhost**
Port number: **50010**
Authentication alias for XA recovery: **mf_login**
Component-managed authentication alias: **mf_login**
Mapping-configuration alias: **(none)**
Container-managed authentication alias: **mf_login**

9. Go back to **Data sources** page, select **MF** and **SAMPLE** and then click the **Test connection** button.



10. The following screen should be shown to inform you test connections successful.



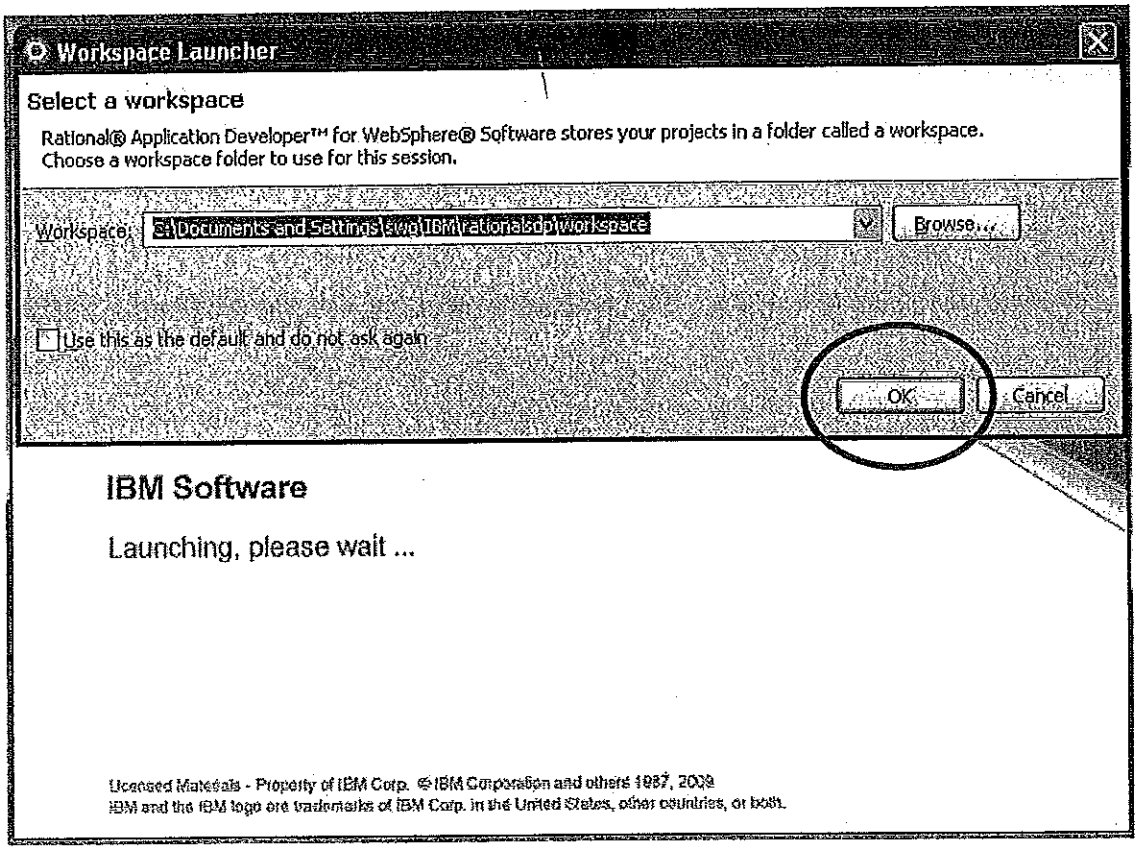
Congratulations, you have finished Task B of this lab. Go ahead to Task C.

Task C – Test and Modify Servlet Involving a Distributed Transaction

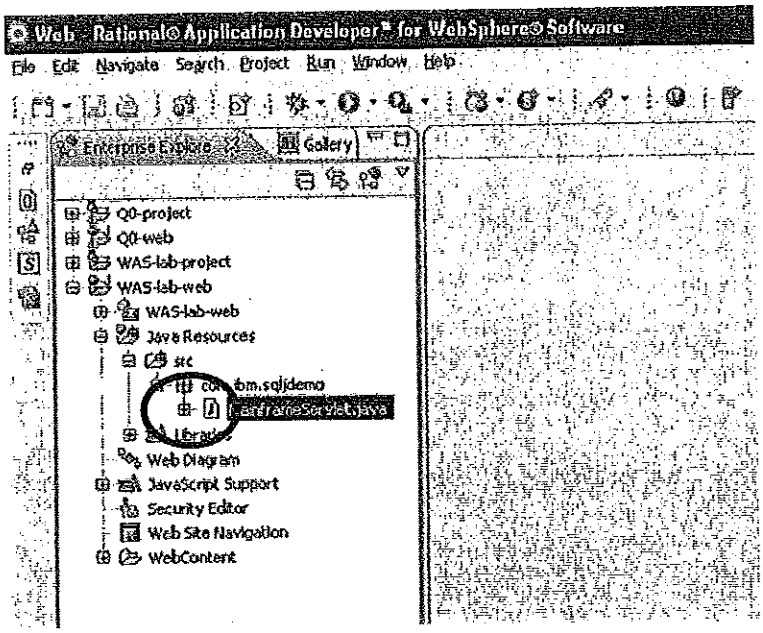
Use **db2admin** as username and **password** as password account log in the Windows VMware image.

1. Double click on the **IBM Rational Application Developer** icon (RAD) which is located on desktop

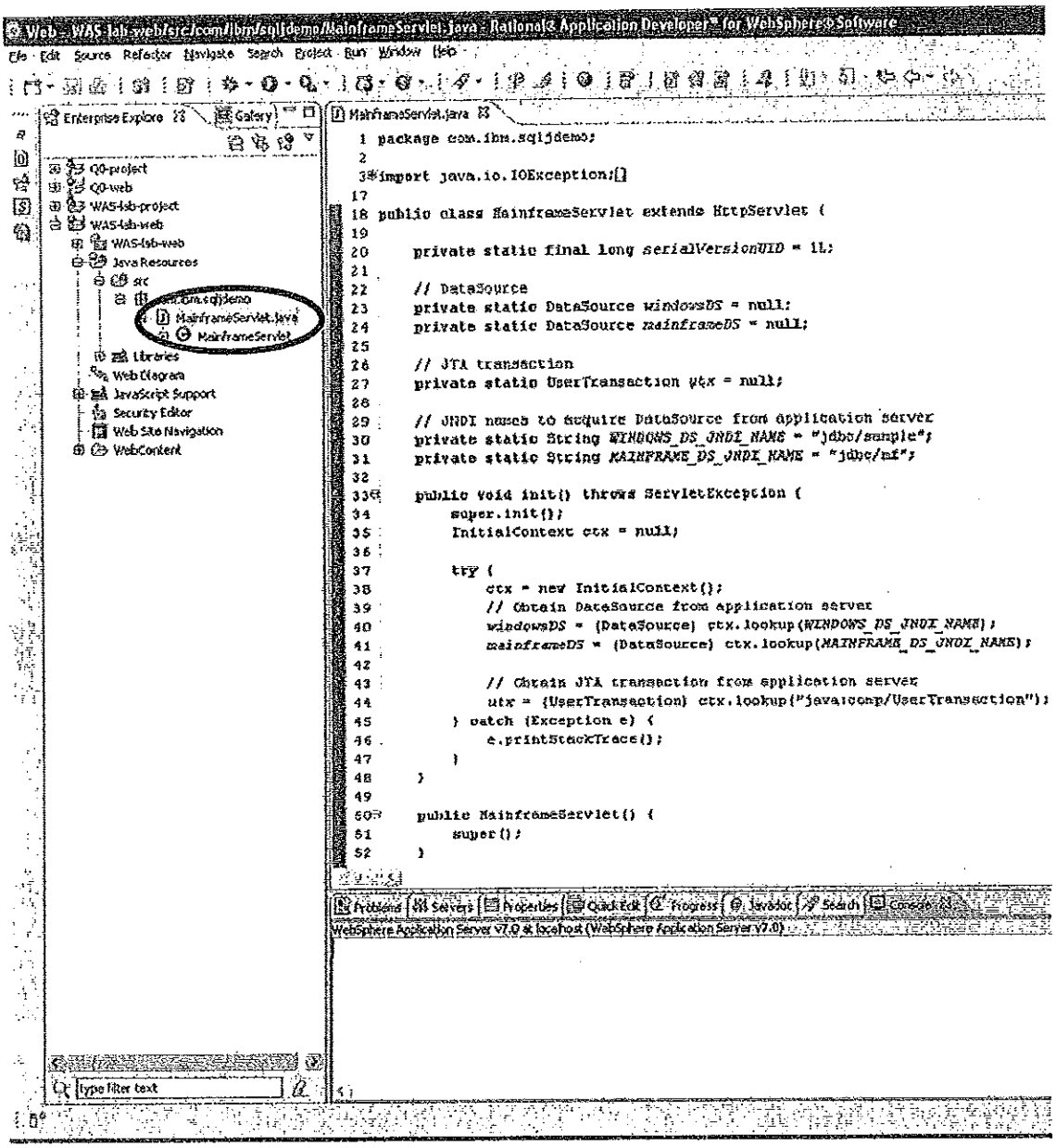
2. Click OK to accept the default workspace location



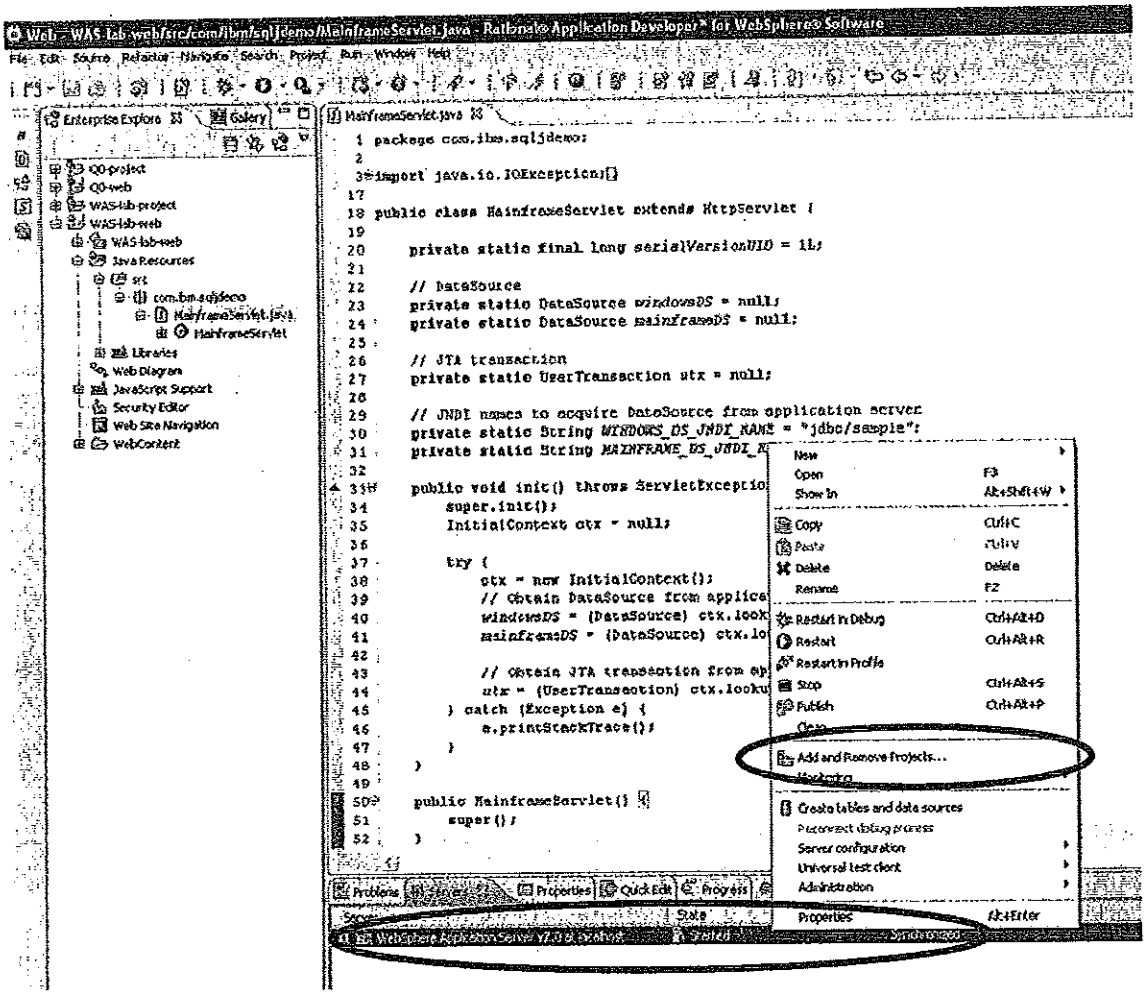
3. Expand the **WAS-lab-web project -> Java Resources -> src -> com.ibm.sqljdemo** by clicking (+) in the Enterprise Explorer.



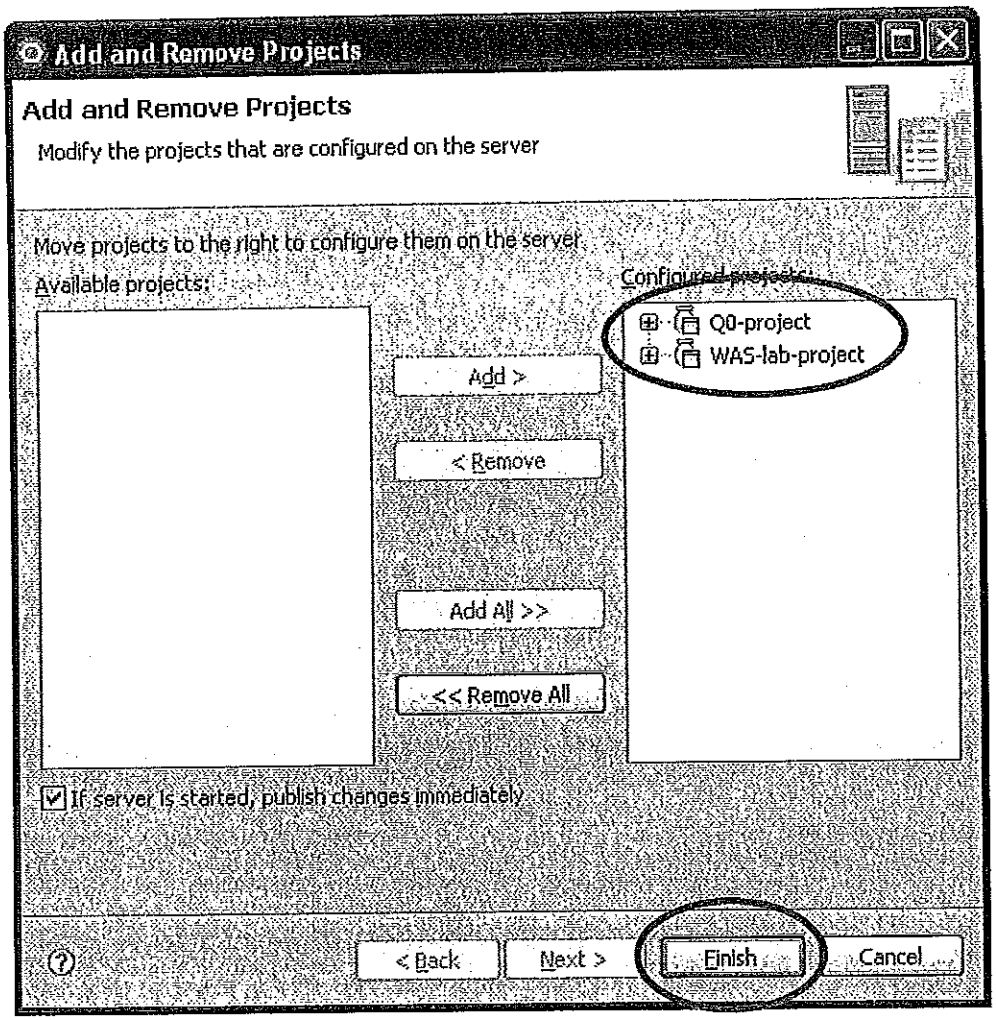
4. Double click **MainframeServlet.java**, modification of codes will be done here.



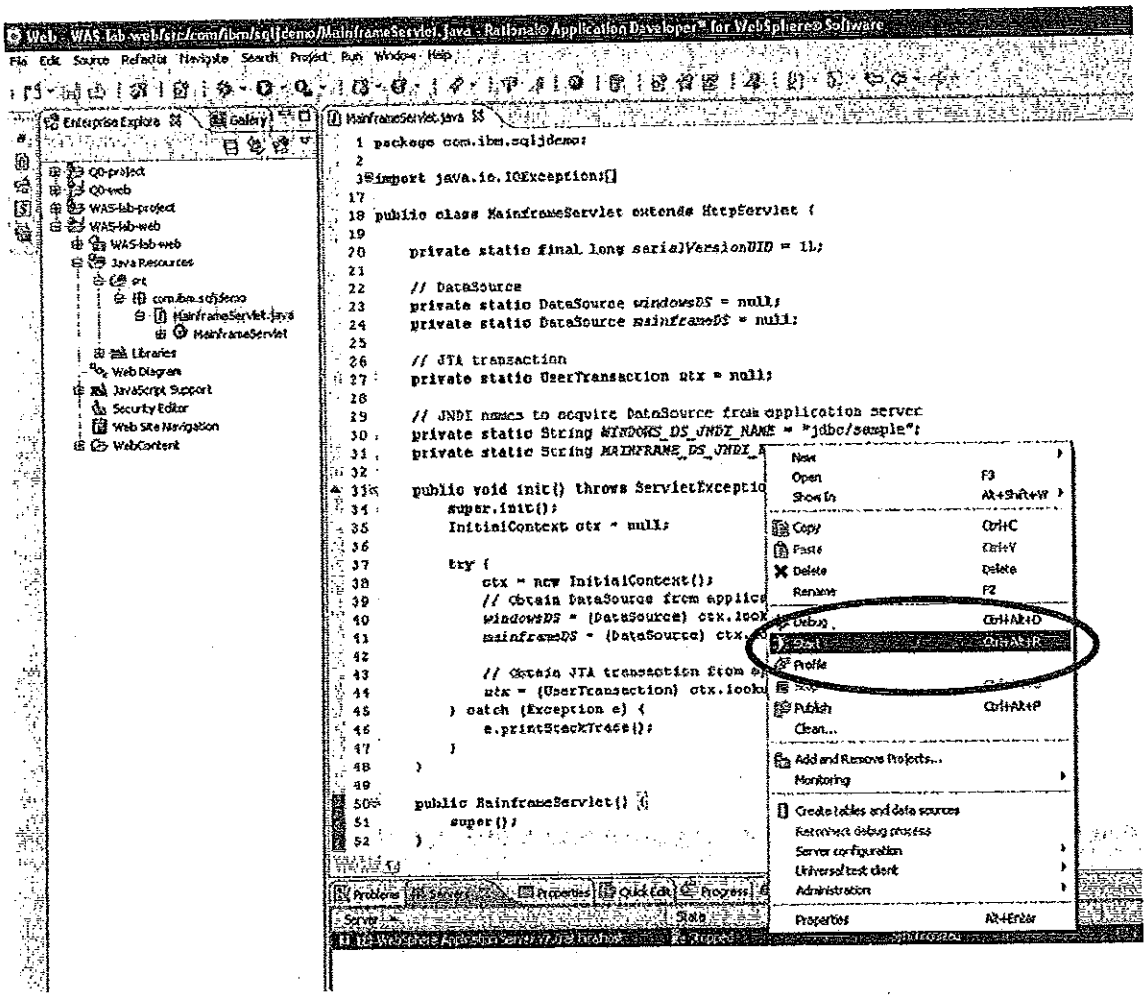
5. Right click on the **WebSphere Application Server v7.0 at localhost** in the server perspective, select **Add and Remove Projects** in order to test the project.



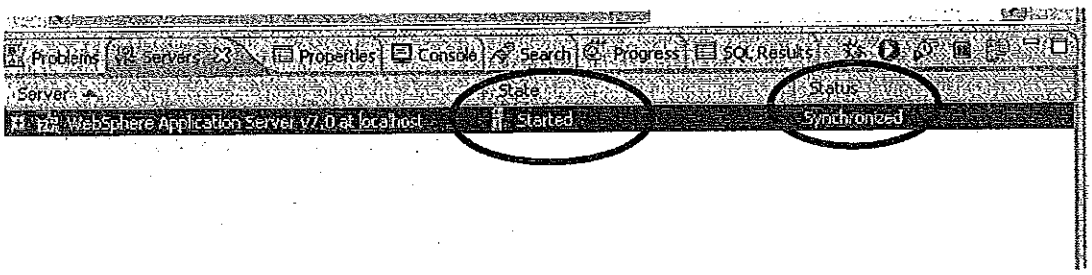
6. Make sure the project **WAS-lab-project** is in the Configured projects, otherwise, click **Add All** to add the project and click **Finish**



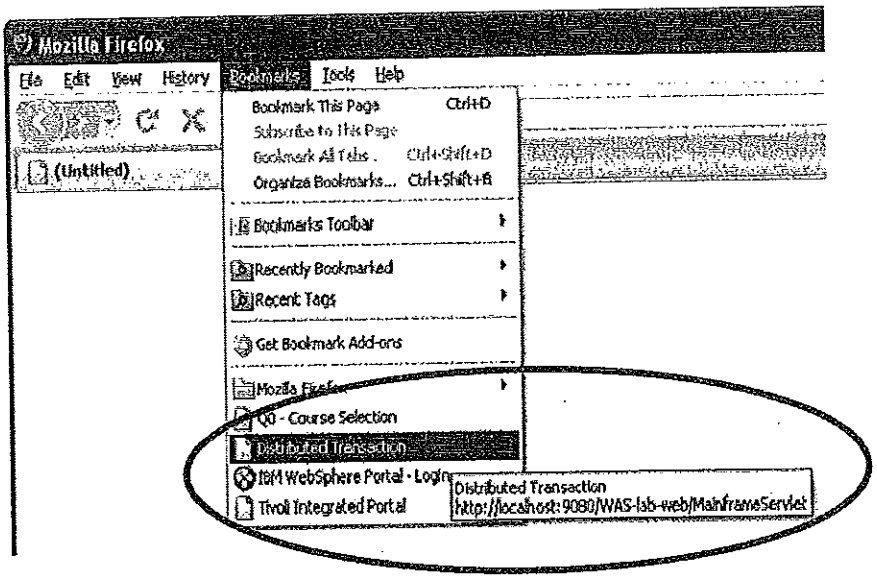
7. Start the WebSphere Application Server instance by right clicking **WebSphere Application Server v7.0 at localhost** and selecting **Start**.



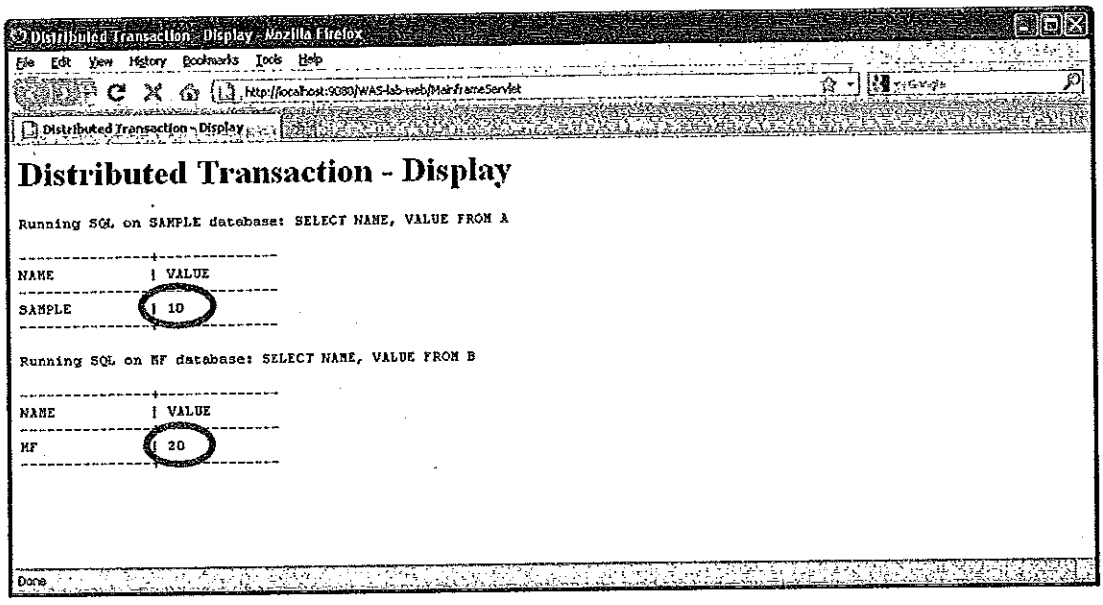
8. Check the server status, which should be "State = Started" and "Status = Synchronized" (server start-up time highly depends on the machine power, which should be within 1-5 minutes)



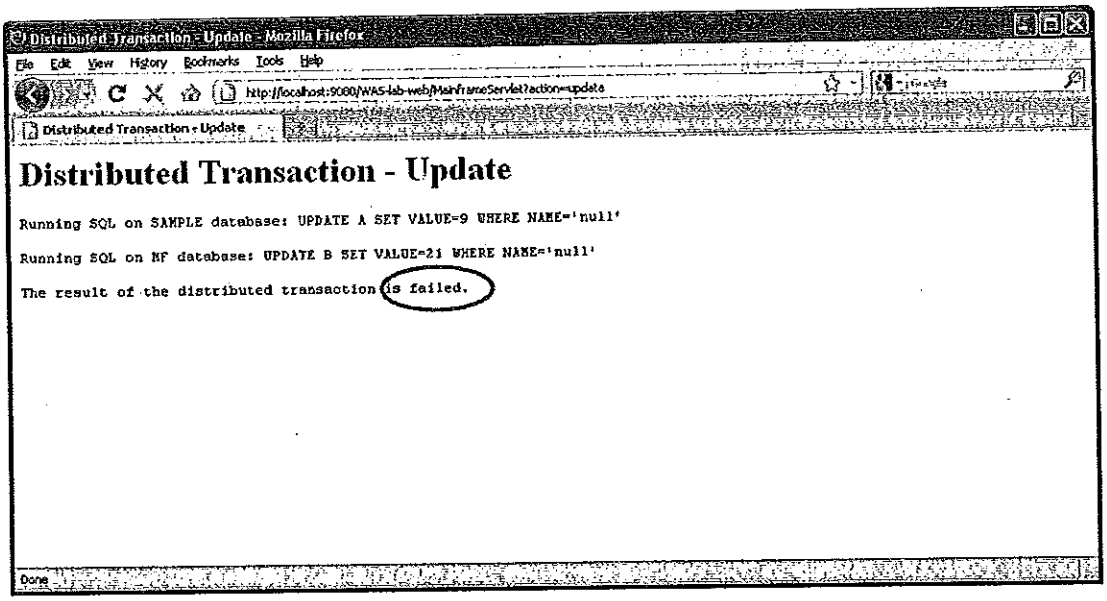
9. Open **FireFox** and navigate for the link (http://localhost:9080/WAS-lab-web/MainframeServlet) in Bookmarks.



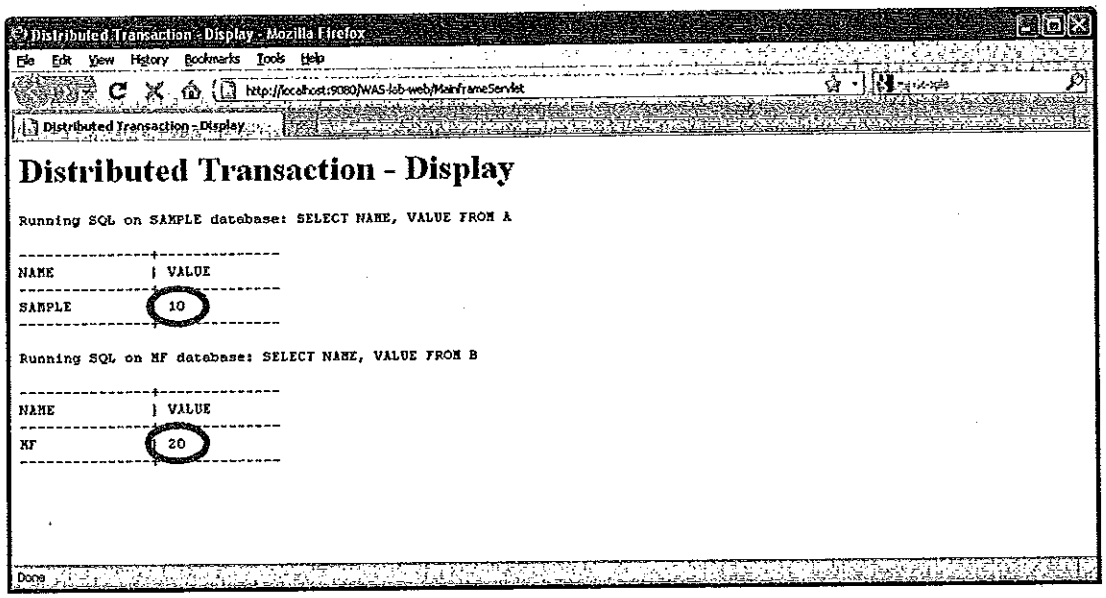
10. The following screen should appear, it shows a record of table **A** in **SAMPLE** database and another record of table **B** in **MF** database respectively.



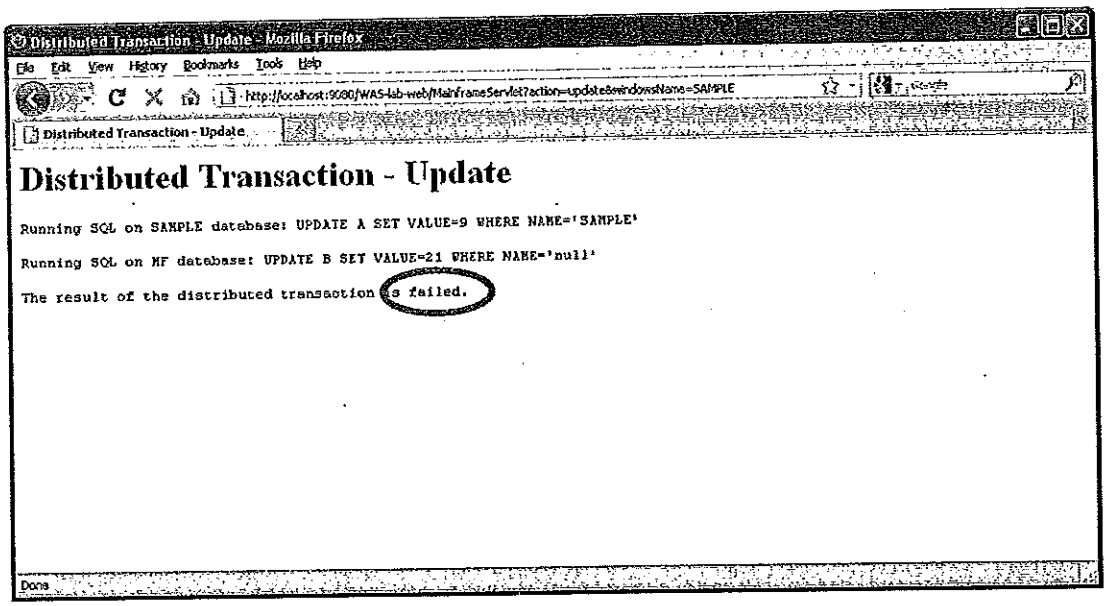
11. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=update). It shows the distributed transaction failed.



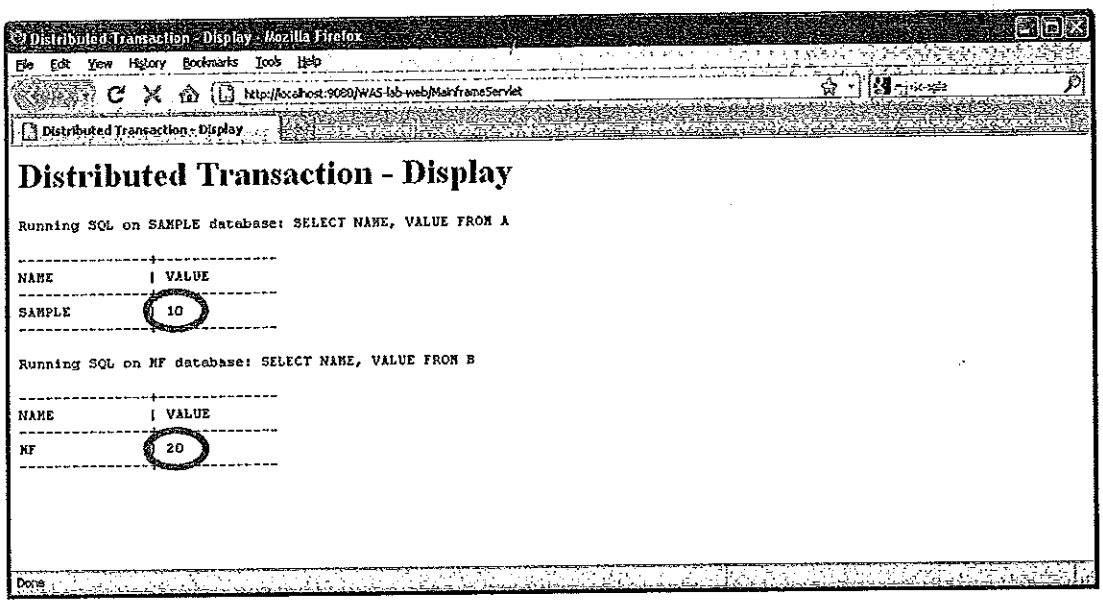
12. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=display). It shows the records in tables **A** and **B** and their values are not changed.



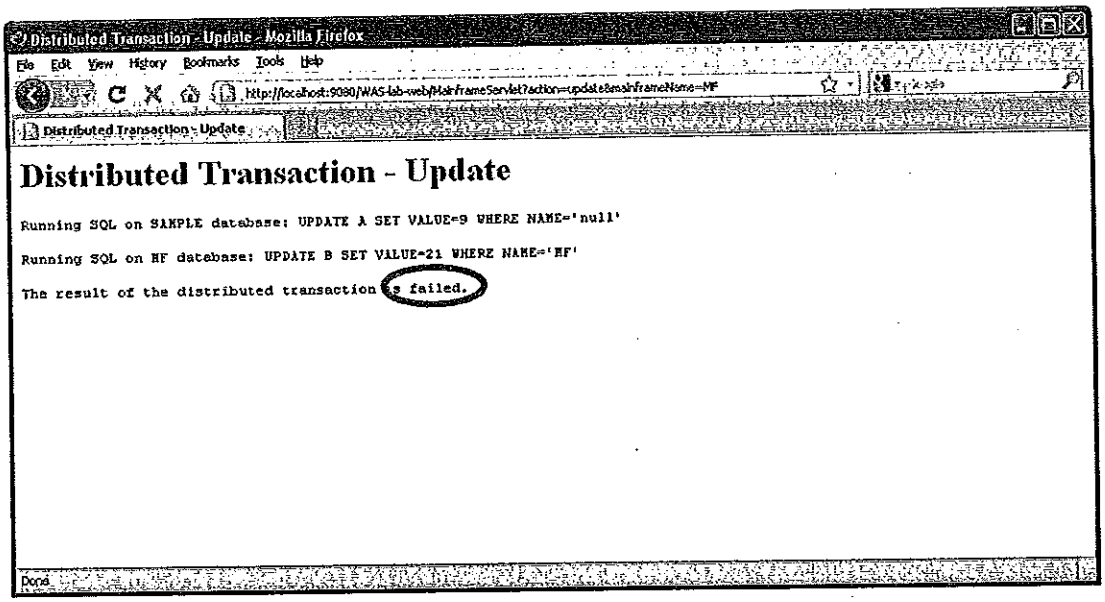
13. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=update&windowsName=SAMPLE). It shows the distributed transaction failed again.



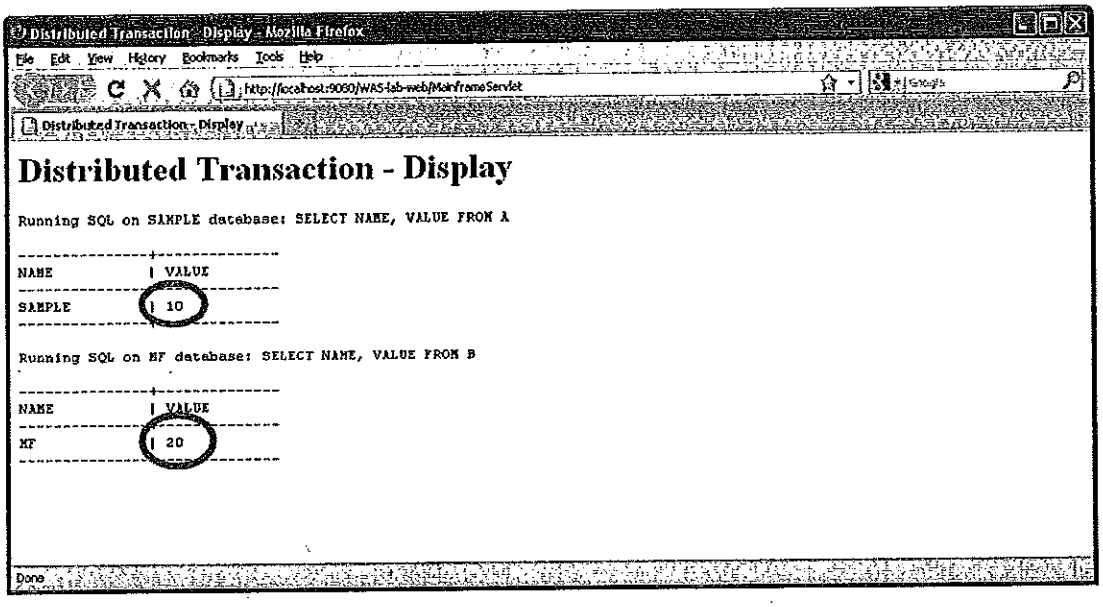
14. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=display). It shows the records in tables **A** and **B** and, again, their values are not changed.



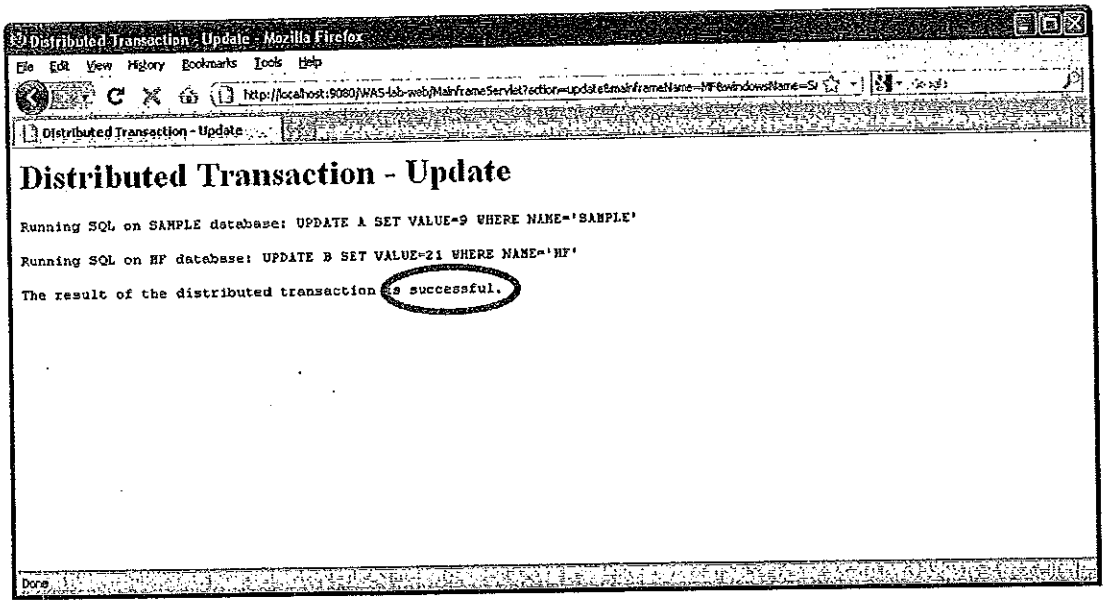
15. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=update&mainframeName=MF). It shows the distributed transaction failed again.



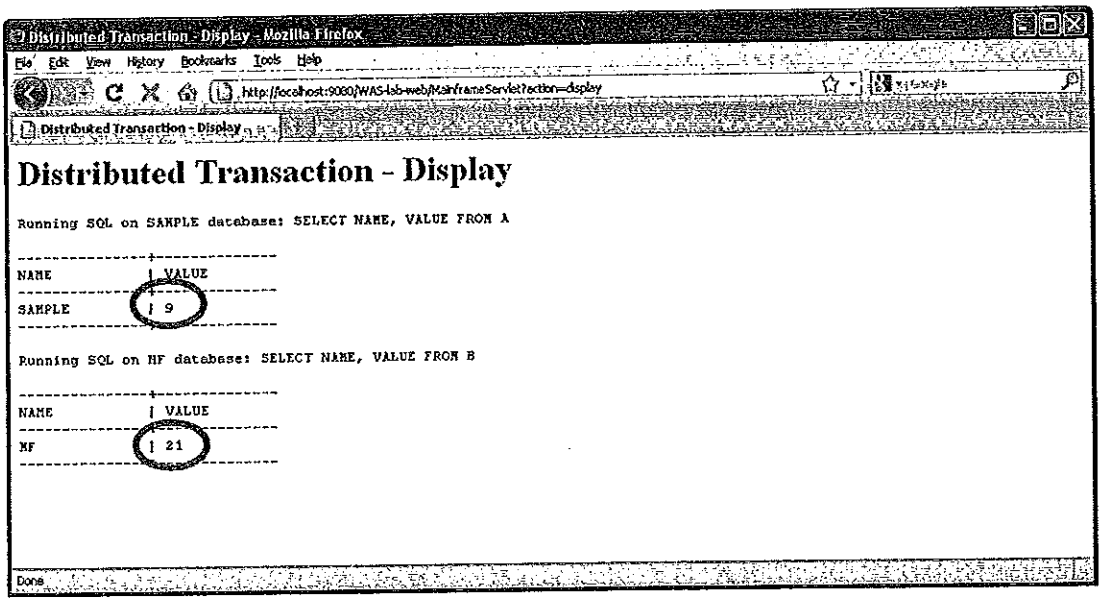
16. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=display). It shows the records in tables **A** and **B** and, again, their values are not changed.



17. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=update&mainframeName=MF&windowsName=SAMPLE). It shows the distributed transaction successful.



18. On the address bar, enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=display). It shows the records in tables **A** and **B** and their values are changed according to the SQL statements.



Source code explanation

The servlet *MainframeServlet.java* generates an html page. Take a look on the source code in the RAD, the method *request.getParameter("action")* [line 66] get the value of variable "action" from the URL.

```
64    public void doService(HttpServletRequest request,
65            HttpServletResponse response) throws ServletException, IOException {
66        String action = request.getParameter("action");
67
68        if ("display".equals(action)) {
69            doDisplayInfo(request, response);
70        } else if ("update".equals(action)) {
71            doUpdateInfo(request, response);
72        } else
73            doDisplayInfo(request, response);
74    }
75
```

If the value of "action" is equal to "display", it calls the method *doDisplayInfo*. If the value of "action" is equal to "update", it calls *doUpdateInfo*.

```
 1 package com.ibm.sqljdemo;
 2
 3 import java.io.IOException;
17
18 public class MainframeServlet extends HttpServlet {
19
20     private static final long serialVersionUID = 1L;
21
22     // DataSource
23     private static DataSource windowsDS = null;
24     private static DataSource mainframeDS = null;
25
26     // JTA transaction
27     private static UserTransaction utx = null;
28
29     // JNDI names to acquire DataSource from application server
30     private static String WINDOWS_DS_JNDI_NAME = "jdbc/sample";
31     private static String MAINFRAME_DS_JNDI_NAME = "jdbc/mf";
32
33     public void init() throws ServletException {
34         super.init();
35         InitialContext ctx = null;
36
37         try {
38             ctx = new InitialContext();
39             // Obtain DataSource from application server
40             windowsDS = (DataSource) ctx.lookup(WINDOWS_DS_JNDI_NAME);
41             mainframeDS = (DataSource) ctx.lookup(MAINFRAME_DS_JNDI_NAME);
42
43             // Obtain JTA transaction from application server
44             utx = (UserTransaction) ctx.lookup("java:comp/UserTransaction");
45         } catch (Exception e) {
46             e.printStackTrace();
47         }
```

Lines 23 to 24 declare the DataSource objects.

Line 27 declare the JTA transaction object that is used to manipulate the transaction programmatically.

Lines 30 and 31 are the JNDI names configured in the WAS data sources.

Lines 40 to 44 is the program code to obtain DataSource objects and JTA transaction object from the application server.

Line 76 shows the method *doDisplayInfo* that is to generate the html page to display the records of tables A and B respectively.

```
76    public void doDisplayInfo(HttpServletRequest request,
77            HttpServletResponse response) throws ServletException, IOException {
78        ServletOutputStream os = response.getOutputStream();
79
80        // SQL statement to be executed on SAMPLE database to return records
81        String windowsSQL = "SELECT NAME, VALUE FROM A";
82
83        // SQL statement to be executed on MF database to return records
84        String mainframeSQL = "SELECT NAME, VALUE FROM B";
85
86        try {
87            // Get connections from application server
88            Connection windowsConn = windowsDS.getConnection();
89            Connection mainframeConn = mainframeDS.getConnection();
90
91            os.print("<html><head><title>Distributed Transaction - Display</title></head>");
92            os.print("<body>");
93            os.println("<h1>Distributed Transaction - Display</h1>");
94
95            // Create statements in connections
96            Statement windowsStmt = windowsConn.createStatement();
97            Statement mainframeStmt = mainframeConn.createStatement();
98
99            // Execute SQL statements and get records
100           ResultSet windowsRs = windowsStmt.executeQuery(windowsSQL);
101           ResultSet mainframeRs = mainframeStmt.executeQuery(mainframeSQL);
102
103           os.println("<pre>");
104           os.println("Running SQL on SAMPLE database: " + windowsSQL);
105           os.println();
106           os.println("----------------+--------------");
107           os.println("NAME            | VALUE        ");
108           os.println("----------------+--------------");
```

Lines 81 to 84 define the select SQL statements to be execute to get the records from tables A and B respectively.

Lines 88 and 89 are to get DataSource Connection objects from the application server.

Lines 96 and 97 create the JDBC Statement objects to be executed.

Lines 100 and 101 are to execute the SQL statements and return the records to the ResultSet objects.

```
102
103             os.println("<pre>");
104             os.println("Running SQL on SAMPLE database: " + windowsSQL);
105             os.println();
106             os.println("---------------+--------------");
107             os.println("NAME           | VALUE        ");
108             os.println("---------------+--------------");
109             while (windowsRs.next()) {
110                 os.print(windowsRs.getString("NAME"));
111                 os.print("          | ");
112                 os.println(windowsRs.getInt("VALUE"));
113             }
114             os.println("---------------+--------------");
115             os.println();
116             os.println("Running SQL on MF database: " + mainframeSQL);
117             os.println();
118             os.println("---------------+--------------");
119             os.println("NAME           | VALUE        ");
120             os.println("---------------+--------------");
121             while (mainframeRs.next()) {
122                 os.print(mainframeRs.getString("NAME"));
123                 os.print("             | ");
124                 os.println(mainframeRs.getInt("VALUE"));
125             }
126             os.println("---------------+--------------");
127             os.println("</pre>");
128             os.print("</body></html>");
129             os.close();
130
131             // Release resources
132             mainframeStmt.close();
133             windowsStmt.close();
134             mainframeConn.close();
135             windowsConn.close();
```

Lines 109 to 113 are a loop body to iterate the ResultSet returned from the SAMPLE database to display the record values on the HTML page. Lines 121 to 125 function the same but for the MF database.

Lines 132 to 135 release the resources back to the application server.

Line 143 is the method *doUpdateInfo* that is to perform a distributed transaction involving databases SAMPLE and MF.

```
143  public void doUpdateInfo(HttpServletRequest request,
144          HttpServletResponse response) throws ServletException, IOException {
145      ServletOutputStream os = response.getOutputStream();
146
147      // Request parameters from URL
148      String windowsName = request.getParameter("windowsName");
149      String mainframeName = request.getParameter("mainframeName");
150
151      // Flag indicacing the result of distributed transaction
152      boolean success = false;
153
154      // SQL statements to update the respective table in SAMPLE and MF databases
155      String windowsSQL = "UPDATE A SET VALUE=9 WHERE NAME='" + windowsName + "'";
156      String mainframeSQL = "UPDATE B SET VALUE=21 WHERE NAME='" + mainframeName + "'";
157
158      try {
159          // Mark distributed transaction begin
160          utx.begin();
161
162          // Get connections from application server
163          Connection windowsConn = windowsDS.getConnection();
164          Connection mainframeConn = mainframeDS.getConnection();
165
166          // Create statements in connections
167          Statement windowsStmt = windowsConn.createStatement();
168          Statement mainframeStmt = mainframeConn.createStatement();
169
170          // Execute SQL statements and get updated count
171          int windowsCount = windowsStmt.executeUpdate(windowsSQL);
172          int mainframeCount = mainframeStmt.executeUpdate(mainframeSQL);
173
174          // Business logic to determine transaction successful or not
175          //      windowsCount == 1 means updating SAMPLE database successfully
176          //      mainframeCount == 1 means updating MF database successfully
```

Lines 148 and 149 are to get the parameters from the URL which will be passed to the SQL statements for execution.

Lines 155 and 156 are the update SQL statements to modify record value of table A in the SAMPLE database and that of table B in the MF database.

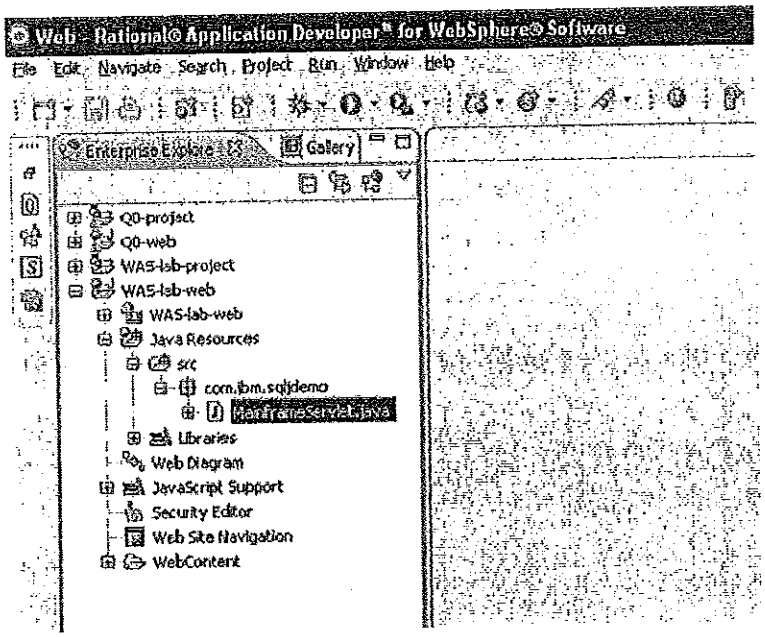
Line 160 is to mark the start of the distributed transaction boundary.

Lines 171 and 172 are the actual execution of the update SQL statements. The method *executeUpdate* returns the number of records affected after the SQL statement is executed. In our sample tables, each should have only one record and therefore the number of records affected in a successful case is one.

```
177             //      if updating both databases successfully, commit the transaction;
178             //      otherwise rollback the transaction
179             if ((windowsCount == 1) && (mainframeCount == 1)) {
180                 utx.commit();
181                 success = true;
182             } else {
183                 utx.rollback();
184                 success = false;
185             }
186
187             // Release resources
188             mainframeStmt.close();
189             windowsStmt.close();
190             mainframeConn.close();
191             windowsConn.close();
192         } catch (SQLException se) {
193             System.out.println("SQLException:" + se.toString());
194         } catch (Exception e) {
195             System.out.println("Exception:" + e.toString());
196         }
197
198         os.print("<html><head><title>Distributed Transaction - Update</title></head>");
199         os.print("<body>");
200         os.println("<h1>Distributed Transaction - Update</h1>");
201         os.println("<pre>");
202         os.println("Running SQL on SAMPLE database: " + windowsSQL);
203         os.println();
204         os.println("Running SQL on MF database: " + mainframeSQL);
205         os.println();
206         if (success)
207             os.println("The result of the distributed transaction is successful.");
208         else
209             os.println("The result of the distributed transaction is failed.");
210         os.println("</pre>");
```

Lines 179 to 185 are the business logic to determine if the distributed transaction should be committed or rolled back. If both numbers of record affected of the tables meet the all the required success conditions, the distributed transaction will be committed by calling *utx.commit()* method; otherwise *utx.rollback()* will be called to cancel all changes made within the transaction boundaries.

19. Open MainframeServlet.java in the RAD if you have not done so.



20. Now modify the program code to change back to the original record values.

Modify the line 155 to

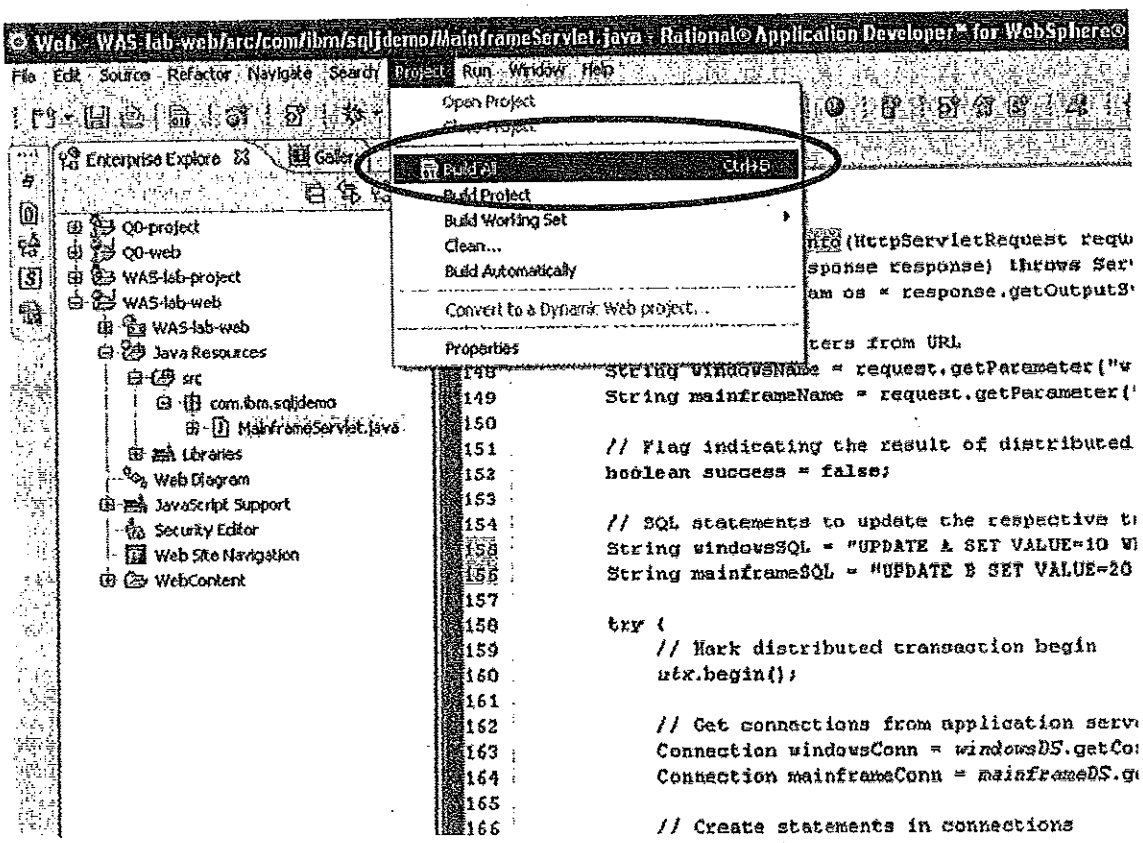
String windowsSQL = "UPDATE A SET VALUE=10 WHERE NAME='" + windowsName + "'";

Modify the line 156 to

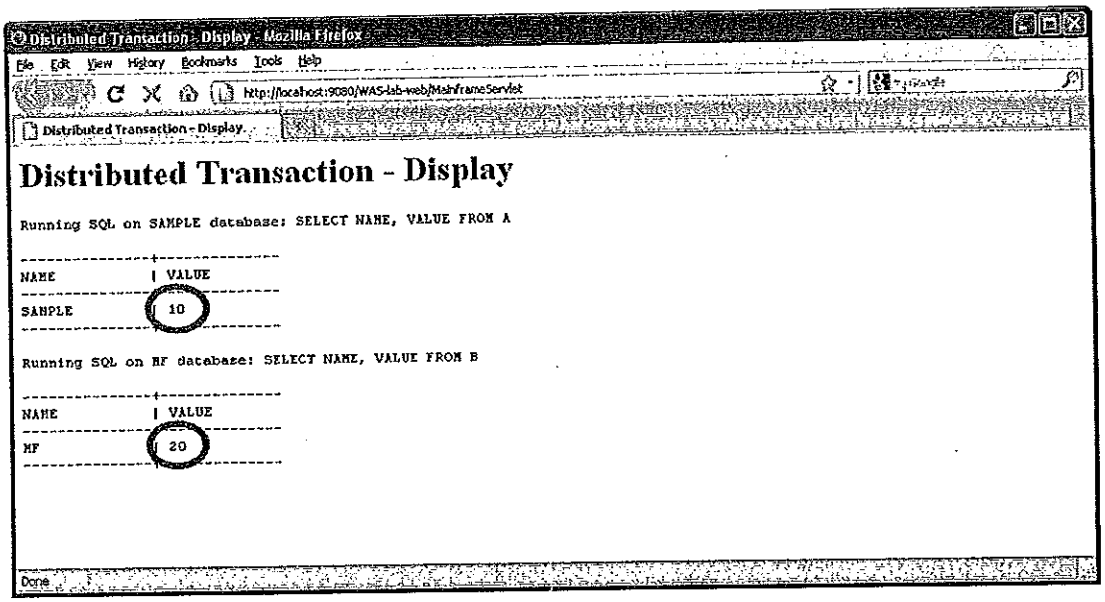
String mainframeSQL = "UPDATE B SET VALUE=20 WHERE NAME='" + mainframeName + "'";

```
148         String windowsName = request.getParameter("windowsName");
149         String mainframeName = request.getParameter("mainframeName");
150
151         // Flag indicating the result of distributed transaction
152         boolean success = false;
153
154         // SQL statements to update the respective table in SAMPLE and MF databases
155         String windowsSQL = "UPDATE A SET VALUE=10 WHERE NAME='" + windowsName + "'";
156         String mainframeSQL = "UPDATE B SET VALUE=20 WHERE NAME='" + mainframeName + "'";
157
158         try {
159             // Mark distributed transaction begin
160             utx.begin();
161
162             // Get connections from application server
163             Connection windowsConn = windowsDS.getConnection();
164             Connection mainframeConn = mainframeDS.getConnection();
165
166             // Create statements in connections
167             Statement windowsStmt = windowsConn.createStatement();
168             Statement mainframeStmt = mainframeConn.createStatement();
169
```

21. Save the file and select **Project -> Build All** to compile the code and commit the changes on the WebSphere Application Server.



22. Open FireFox and enter the link (http://localhost:9080/WAS-lab-web/MainframeServlet?action=update&mainframeName=MF&windowsName=SAMPLE). It shows the distributed transaction successful.



23. Now revert the program code to change back to the SQL statements as follows. Then save the file and select **Project → Build All**. If you are not sure, please refer to step 21 above. **CAUTION! DO NOT RUN THE SERVLET IN FIREFOX AGAIN.** This step is mandatory for the lab exercise of Chapter 5 – Access Mainframe DB2 from WebSphere.

    Modify the line 155 to

    String windowsSQL = "UPDATE A SET VALUE=9 WHERE NAME='" + windowsName + "' ;

    Modify the line 156 to

    String mainframeSQL = "UPDATE B SET VALUE=21 WHERE NAME='" + mainframeName + "' ;

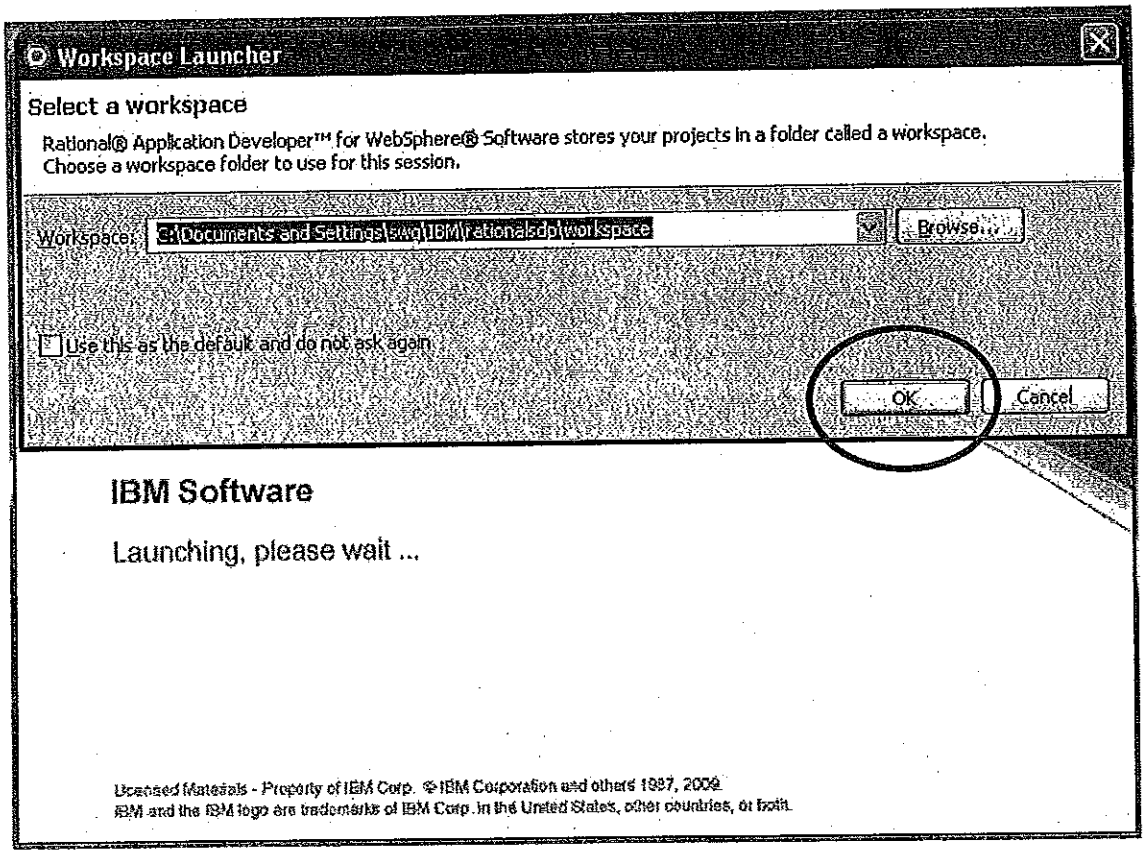
Congratulations, you have finished Task C of this lab. This is the end of the lab.

Supplementary Sheet for Chapter 3 – Task B

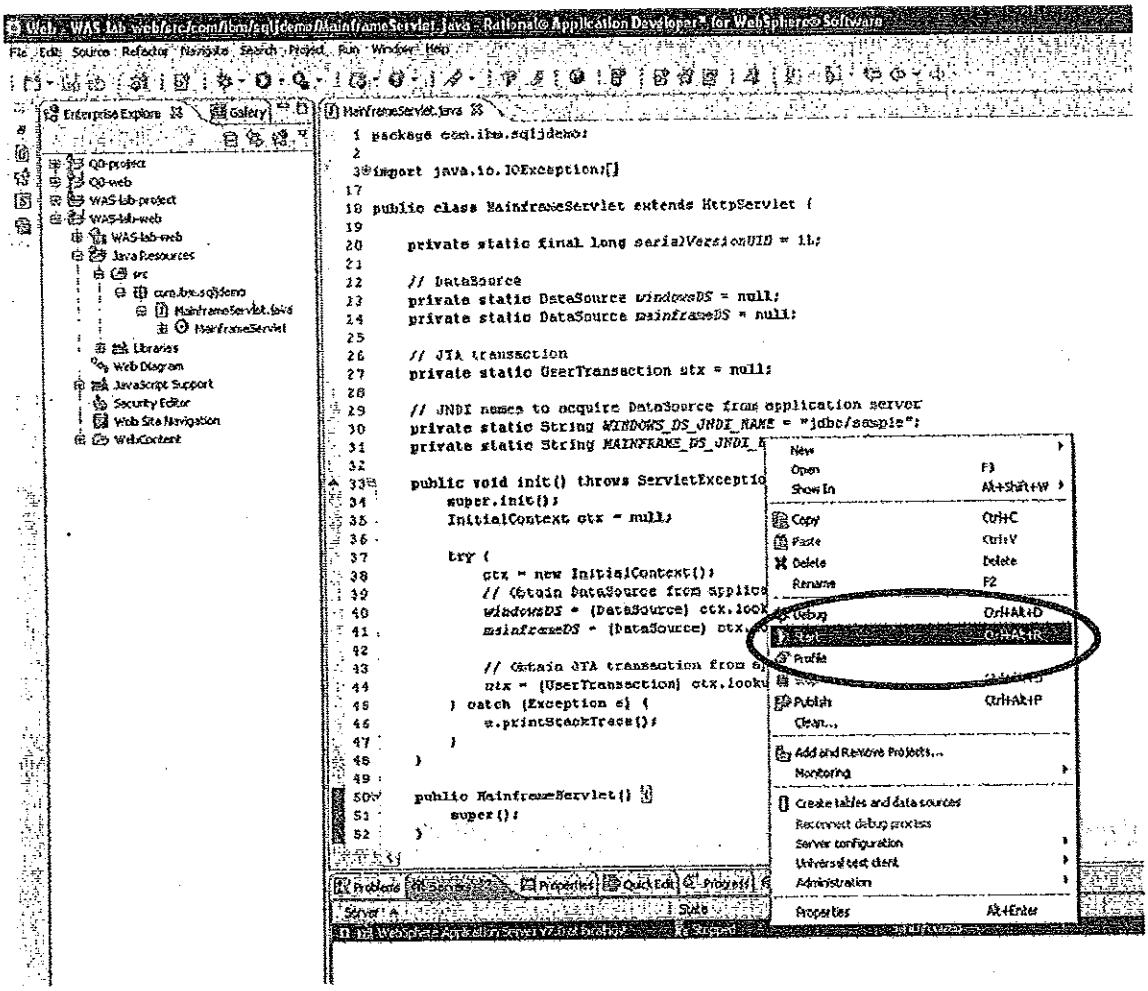
Before executing Step 1 of Task B, please start RAD and WebSphere Application Server.

1. Double click on the **IBM Rational Application Developer** icon (RAD) which is located on desktop

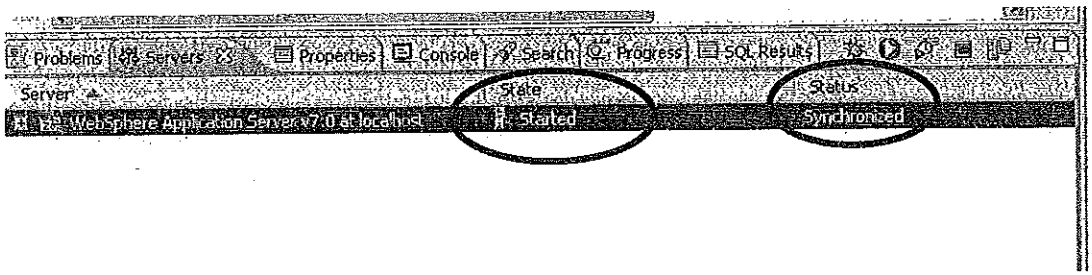
2. Click OK to accept the default workspace location



3. Start the WebSphere Application Server instance by right clicking **WebSphere Application Server v7.0 at localhost** and selecting **Start**.



4. Check the server status, which should be "State = Started" and "Status = Synchronized" (server start-up time highly depends on the machine power, which should be within 1-5 minutes)



You may now go back to follow Step 1 of Task B to continue the lab.

IBM

Information Management

IBM Inter-University Programming Contest 2011 Training

Chapter 4: WebSphere Portal Introduction

Software Group, IBM

IBM Inter-University Programming Contest 2011

TechWorks

Agenda

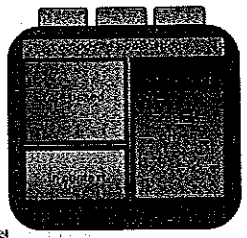
- Value Proposition of a Portal
- What is a Portal?
- WebSphere Portal V7.0 Architecture
- Methods to quickly populate your Portal with existing content and applications
  - Feeds
  - Static HTML
  - Dynamic web applications
- WebSphere Portal delivers Exceptional Mobile Experience

TechWorks

WebSphere Portal – What it Does

WebSphere Portal:

- provides a framework,
- for delivering role based, contextual & process driven composite applications,
- that make more people, more productive,
- in the context of what they do everyday.

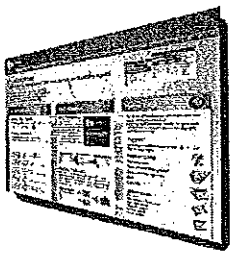


TechWorks

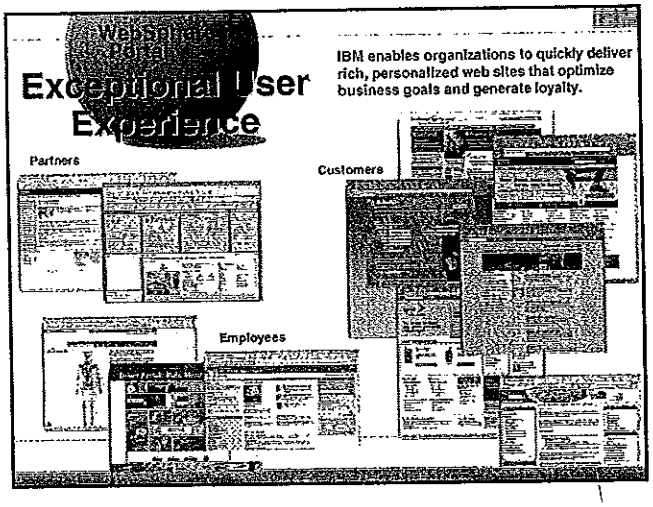
WebSphere Portal Strategy

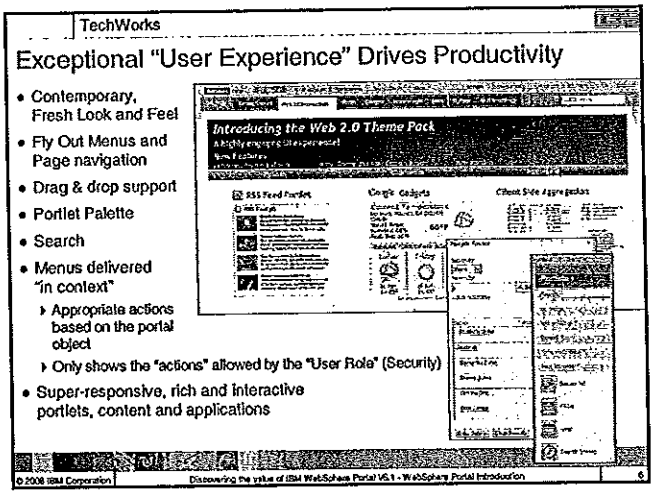
Delivering Exceptional User Experiences:

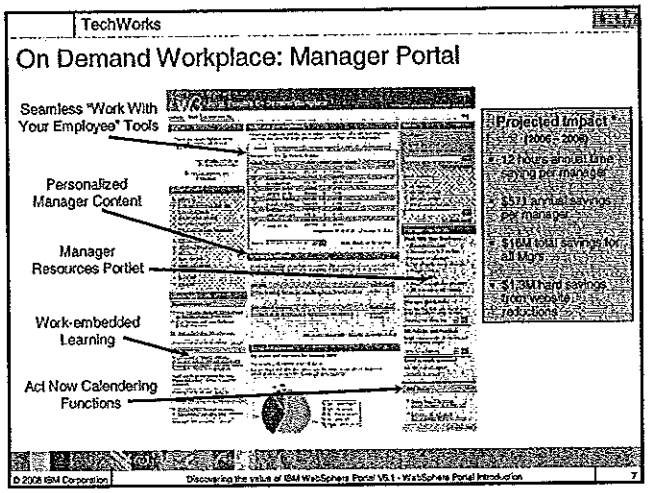
- Deliver the user experiences they choose to their partners, employees, customers, or citizens, with flexibility for change and based on open standards.
- Quickly leverage existing investments, through market-leading composite application tooling and robust framework
- Create highly personalized applications that adapt to users' context, community, role, actions, location, and preferences
- Interact with information from the user's device of choice
- Deliver a front-end to SOA, enabling business flexibility and agility
- Speed time to value with Prebuilt Portal snap-ons for specific business problems

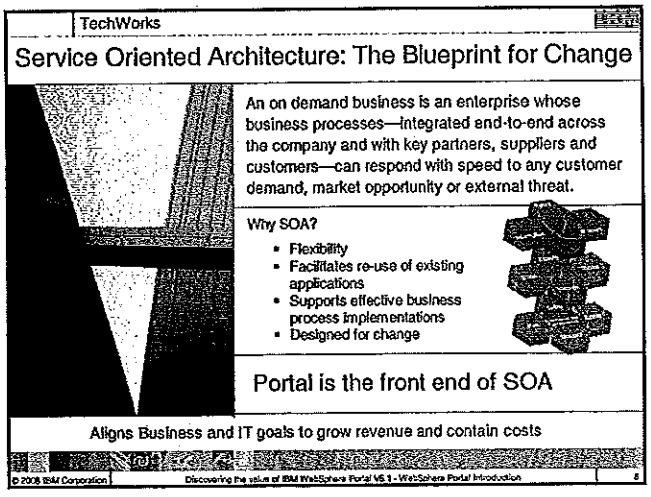


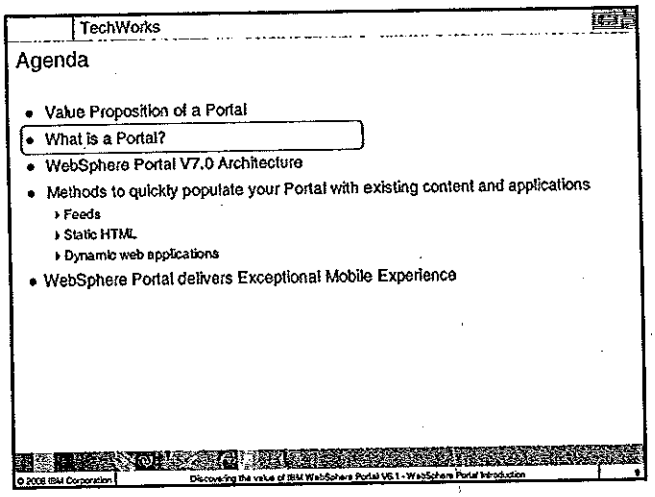
Sophie orientated

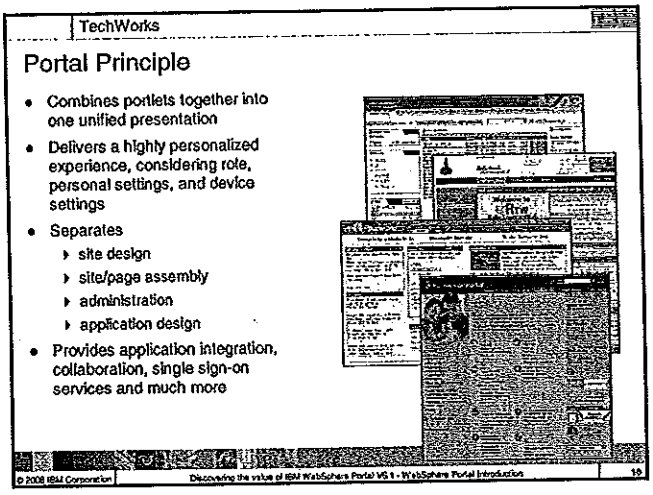

Exceptional User Experience
IBM enables organizations to quickly deliver rich, personalized web sites that optimize business goals and generate loyalty.
Partners
Customers
Employees

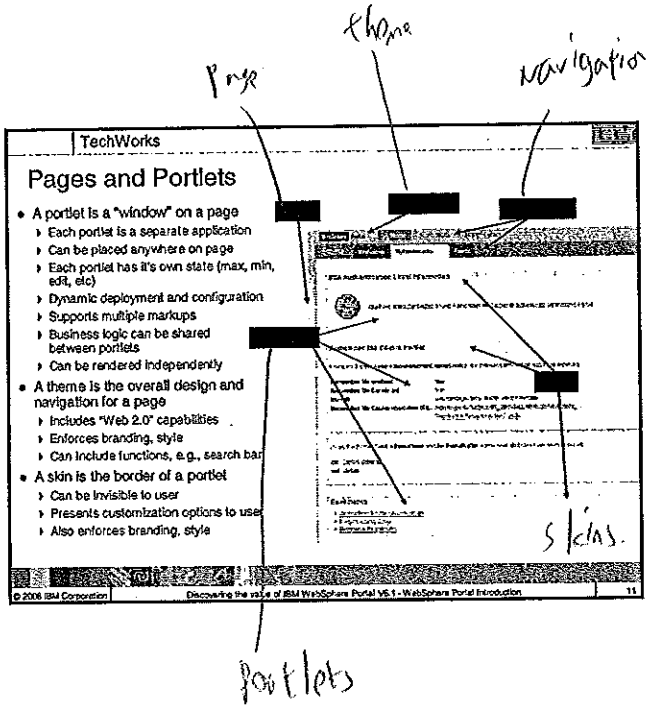

TechWorks
Exceptional "User Experience" Drives Productivity
• Contemporary, Fresh Look and Feel
• Fly Out Menus and Page navigation
• Drag & drop support
• Portlet Palette
• Search
• Menus delivered "in context"
‣ Appropriate actions based on the portal object
‣ Only shows the "actions" allowed by the "User Role" (Security)
• Super-responsive, rich and interactive portlets, content and applications
Introducing the Web 2.0 Theme Pack

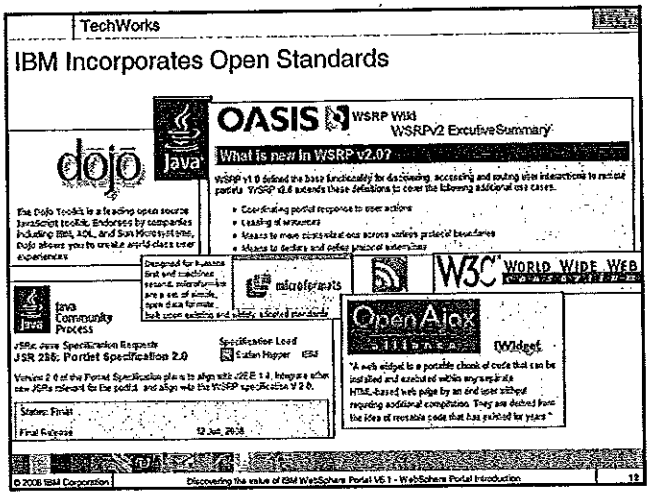

TechWorks
On Demand Workplace: Manager Portal
Seamless "Work With Your Employee" Tools
Personalized Manager Content
Manager Resources Portlet
Work-embedded Learning
Act Now Calendering Functions
Projected Impact

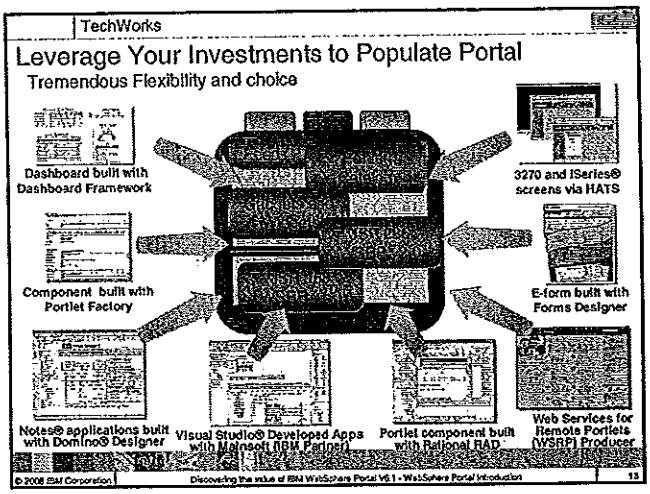

TechWorks
Service Oriented Architecture: The Blueprint for Change
An on demand business is an enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers and customers—can respond with speed to any customer demand, market opportunity or external threat.
Why SOA?
• Flexibility
• Facilitates re-use of existing applications
• Supports effective business process implementations
• Designed for change
Portal is the front end of SOA
Aligns Business and IT goals to grow revenue and contain costs

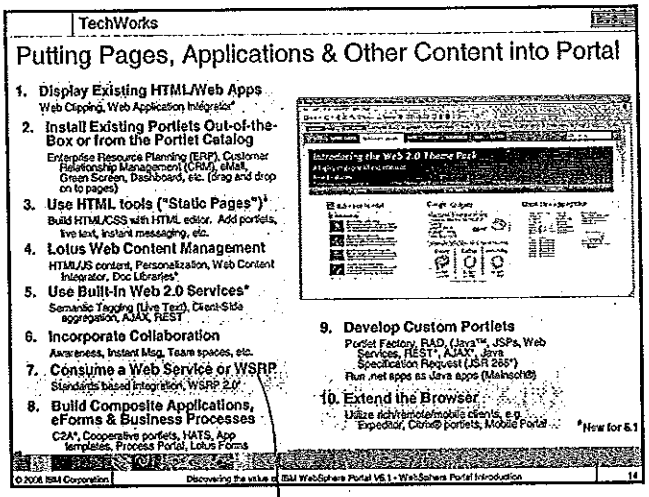

TechWorks
Agenda
Value Proposition of a Portal
What is a Portal?
WebSphere Portal V7.0 Architecture
Methods to quickly populate your Portal with existing content and applications
Feeds
Static HTML
Dynamic web applications
WebSphere Portal delivers Exceptional Mobile Experience


TechWorks
Portal Principle
Combines portlets together into one unified presentation
Delivers a highly personalized experience, considering role, personal settings, and device settings
Separates
site design
site/page assembly
administration
application design
Provides application integration, collaboration, single sign-on services and much more

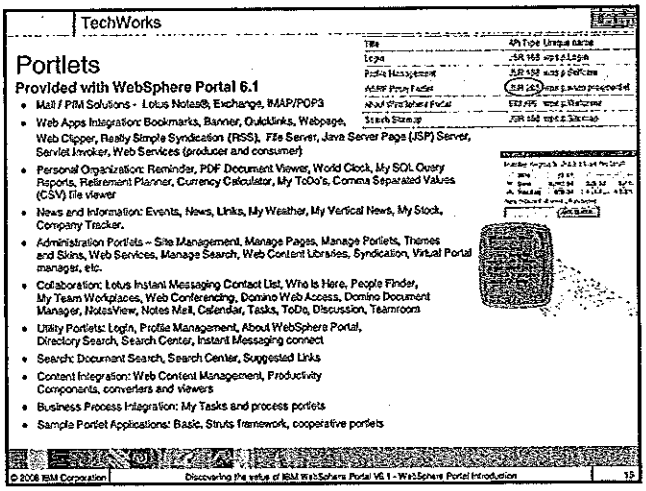

TechWorks
Pages and Portlets
A portlet is a "window" on a page
Each portlet is a separate application
Can be placed anywhere on page
Each portlet has it's own state (max, min, edit, etc)
Dynamic deployment and configuration
Supports multiple markups
Business logic can be shared between portlets
Can be rendered independently
A theme is the overall design and navigation for a page
Includes "Web 2.0" capabilities
Enforces branding, style
Can include functions, e.g., search bar
A skin is the border of a portlet
Can be invisible to user
Presents customization options to user
Also enforces branding, style

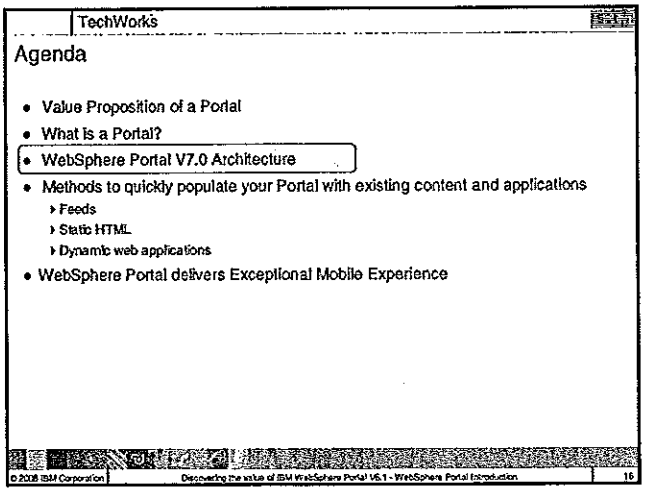

TechWorks
IBM Incorporates Open Standards
OASIS
WSRP Wiki
WSRPv2 ExcutiveSummary
What is new in WSRP V2.0?
dojo
Java
W3C
WORLD WIDE WEB
microformats
OpenAjax
Widget
Java Community Process
JSR 286: Portlet Specification 2.0

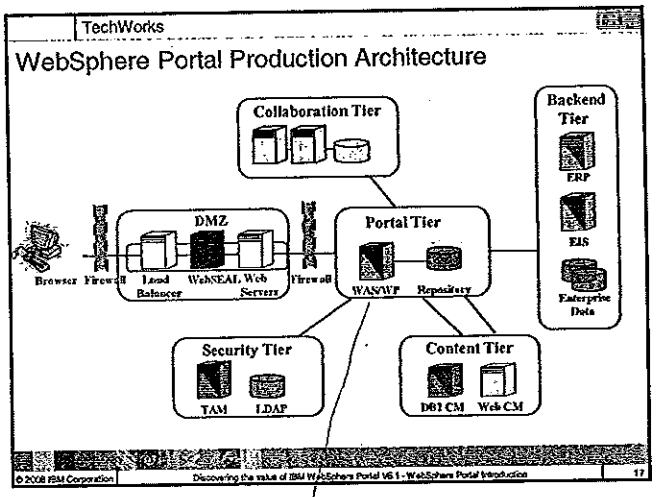


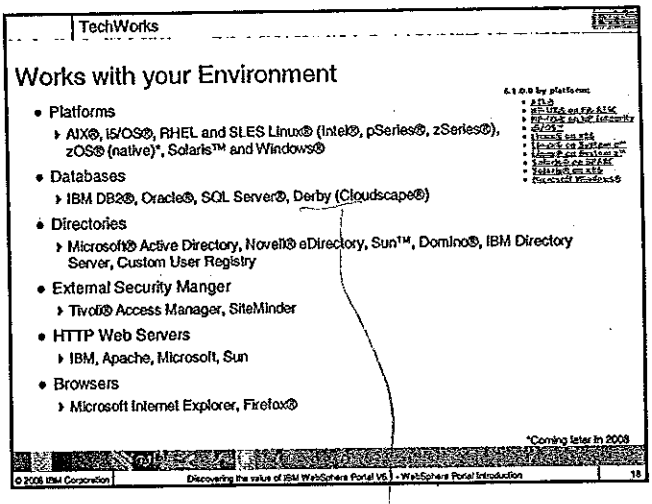


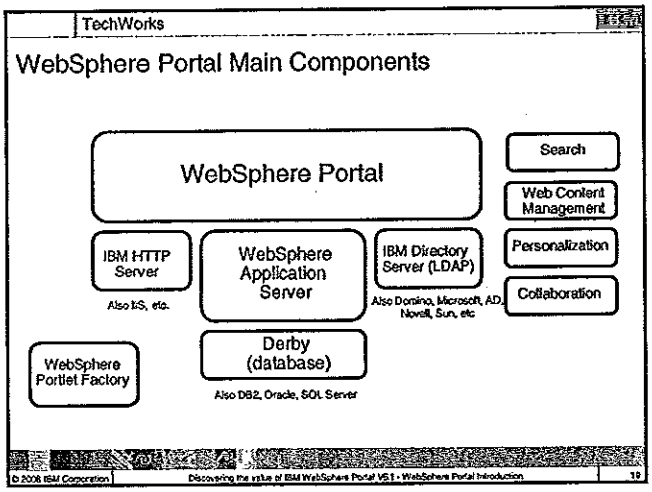
Remote Portlet

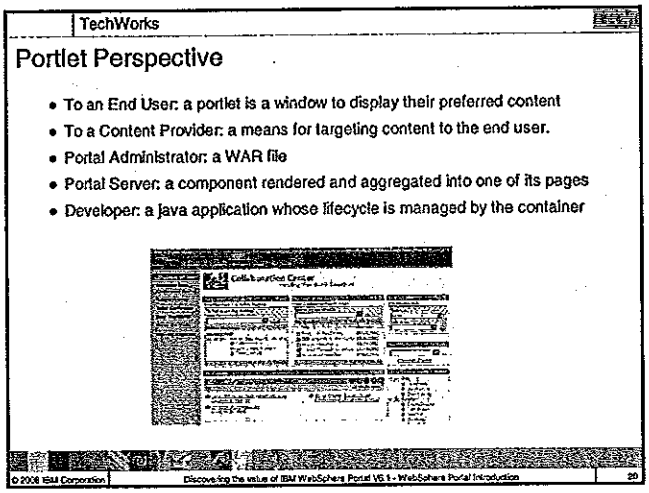





TechWorks
WebSphere Portal Production Architecture
Collaboration Tier
Backend Tier
ERP
EIS
Enterprise Data
DMZ
Portal Tier
Browser
Firewall
Load Balancer
WebSEAL Web Servers
Firewall
WAS/WP
Repository
Security Tier
TAM
LDAP
Content Tier
DB2 CM
Web CM
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V6.1 - WebSphere Portal Introduction
17


TechWorks
Works with your Environment
• Platforms
› AIX®, i5/OS®, RHEL and SLES Linux® (Intel®, pSeries®, zSeries®), zOS® (native)*, Solaris™ and Windows®
• Databases
› IBM DB2®, Oracle®, SQL Server®, Derby (Cloudscape®)
• Directories
› Microsoft® Active Directory, Novell® eDirectory, Sun™, Domino®, IBM Directory Server, Custom User Registry
• External Security Manger
› Tivoli® Access Manager, SiteMinder
• HTTP Web Servers
› IBM, Apache, Microsoft, Sun
• Browsers
› Microsoft Internet Explorer, Firefox®
*Coming later in 2008
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V6.1 - WebSphere Portal Introduction
18

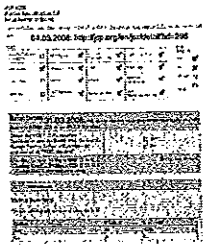

TechWorks
WebSphere Portal Main Components
WebSphere Portal
Search
Web Content Management
Personalization
Collaboration
IBM HTTP Server
Also IIS, etc.
WebSphere Application Server
IBM Directory Server (LDAP)
Also Domino, Microsoft, AD, Novell, Sun, etc
Derby (database)
Also DB2, Oracle, SQL Server
WebSphere Portlet Factory
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V6.1 - WebSphere Portal Introduction
19


TechWorks
Portlet Perspective
• To an End User: a portlet is a window to display their preferred content
• To a Content Provider: a means for targeting content to the end user.
• Portal Administrator: a WAR file
• Portal Server: a component rendered and aggregated into one of its pages
• Developer: a Java application whose lifecycle is managed by the container
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V6.1 - WebSphere Portal Introduction
20

TechWorks

Portlet Standards

- Java™ Specification Request (JSR) 286 – Java Portlet Specification V2.0
  - IBM is leading this JSR, all major Java technology portal (commercial and open source) vendors represented
  - Approved: March 04th 2008 (WebSite complete April 9th)
- Web Services for Remote Portlets (WSRP) V2.0
  - Standard protocol for accessing portlets as web service
  - Defined at OASIS, chaired by IBM, Approved: March 31st 2003
- Common goals
  - Enable coordination between portlets and allow building composite applications based on portlet components (events and render context (e.g. public render parameters))
- WebSphere Portal seamlessly integrates JSR168, JSR286 and WSRP 2.0 Portlets.
  - Also integrates IBM Portlet API and WSRP 1.0



TechWorks

Agenda

- Value Proposition of a Portal
- What is a Portal?
- WebSphere Portal V7.0 Architecture
- Methods to quickly populate your Portal with existing content and applications
  - Feeds
  - Static HTML
  - Dynamic web applications
- WebSphere Portal delivers Exceptional Mobile Experience

TechWorks

Various Methods to Display Portal Content

- Provide links to existing content*
- Display existing static pages*
- Include dynamic web applications*
- Leverage feed sources*
- Build custom portlets
- Supplied email and Lotus portlets
- Download vendor portlets from IBM® Portal Catalog*
- Render Web Services for Remote Portlets (WSRP)*
- Web Content Manager
- Others?

* We will cover these during this presentation

TechWorks

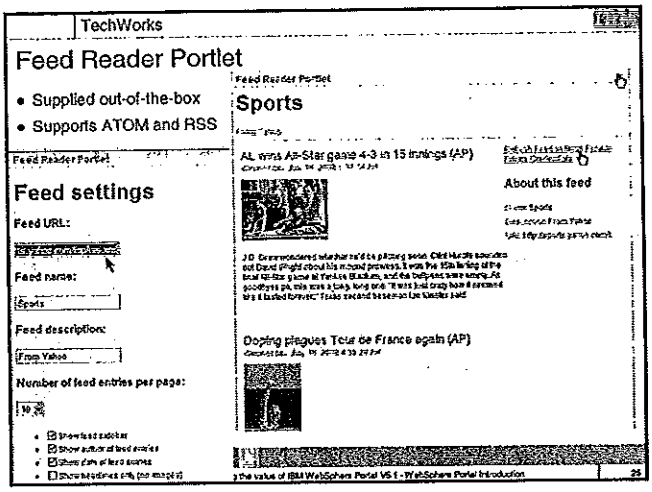
Quickly Populate your Portal

- Value of WebSphere Portal
  - Quickly and easily consolidate existing content and applications
- WebSphere Portal V6.1 includes numerous out-of-the-box features
- We will highlight the following:
  - Really Simple Syndication (RSS) Feeds
  - Bookmarks and Links
  - Google Gadgets
  - Static Web Pages
  - Web Application Integrator
  - Portal Catalog
  - Web Services for Remote Portlets (WSRP)

TechWorks

Feed Reader Portlet

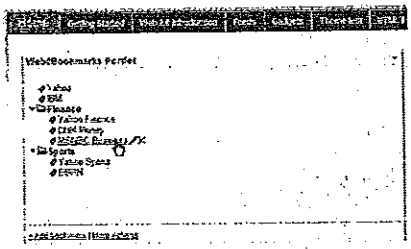
- Supplied out-of-the-box
- Supports ATOM and RSS



TechWorks

Links and Bookmarks

- New Web 2.0 Bookmarks Portlet
- Supplied out-of-the-box
- Web 2.0 user interface makes it fast and easy to use!
- Can bookmark external URLs and internal Portal pages



TechWorks

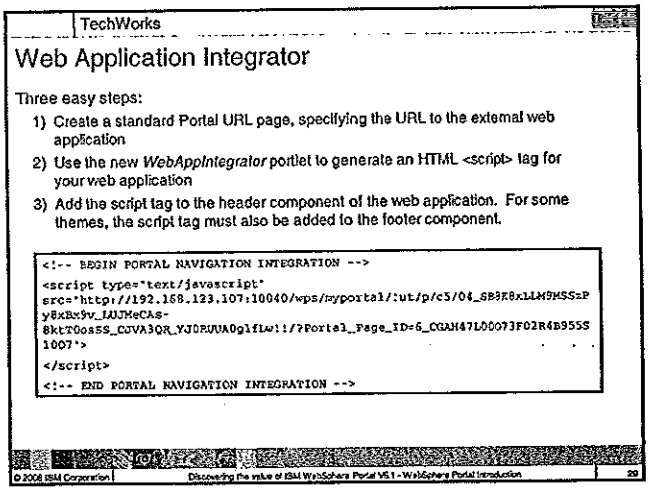
Google Gadgets

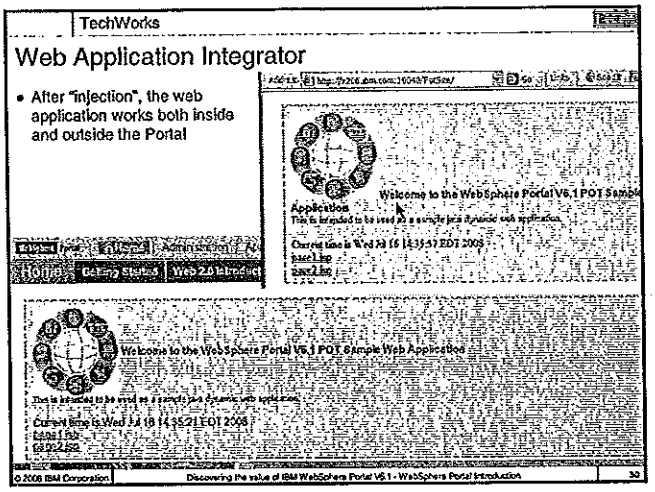
- Place any Google gadget on a page
- Download the IBM Portlet for Google Gadgets™ from the Portal catalog
- Seamless integration with thousands of web-based solutions

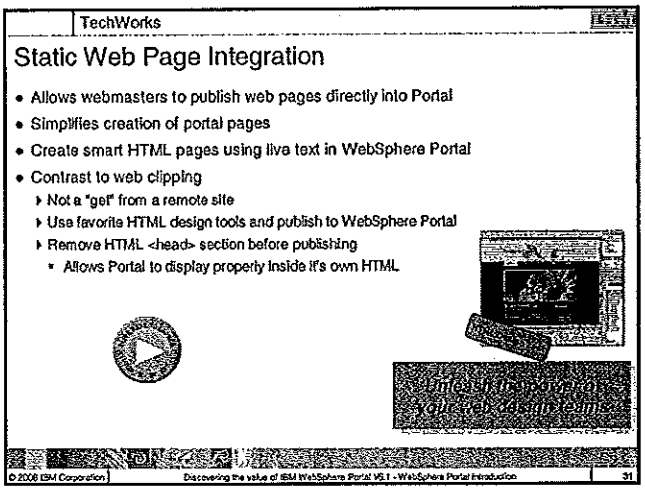
TechWorks

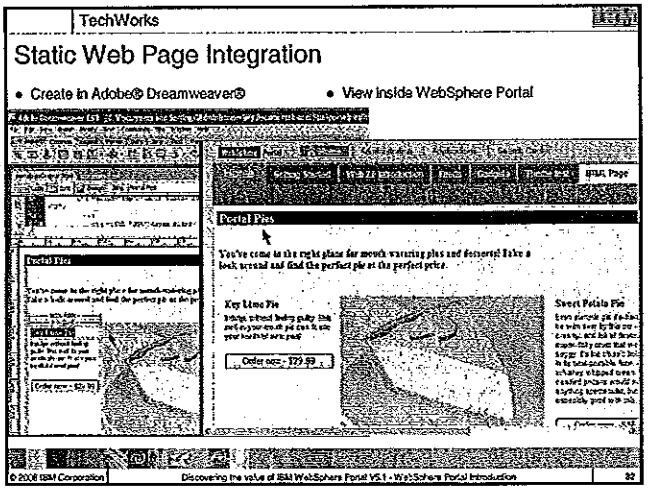
Web Application Integrator

- Centralizes access to external web applications within the enterprise
- Integration of web applications occurs by injecting Portal navigation into external web applications themselves
- No portlet development required
- Does NOT use an IFrame element
- Portal markup is "injected" into an external web application by adding an HTML <script> tag to the web app
- User believes he/she is still within the Portal environment
  - In reality, they are natively browsing the external web application
- The integrated web application can use any technology (e.g. .NET, Ruby on Rails™, PHP, PEARL, Domino®, J2EE, etc.)
- Download this portlet from IBM Portal catalog


TechWorks
Web Application Integrator
Three easy steps:
1) Create a standard Portal URL page, specifying the URL to the external web application
2) Use the new WebAppIntegrator portlet to generate an HTML <script> tag for your web application
3) Add the script tag to the header component of the web application. For some themes, the script tag must also be added to the footer component.
<!-- BEGIN PORTAL NAVIGATION INTEGRATION -->
<script type="text/javascript"
src="http://192.168.123.107:10040/wps/myportal/:ut/p/c5/04_SB8K8xLLM9MSSzP
y8xBz9v_LUJMeCAs-
8ktTOosSS_CGVA3QR_YJURUUA0glfLw!!/?Portal_Page_ID=6_CGAH47L00073F02R4B955S
1007">
</script>
<!-- END PORTAL NAVIGATION INTEGRATION -->
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V5.1 - WebSphere Portal Introduction
29


TechWorks
Web Application Integrator
• After "injection", the web application works both inside and outside the Portal
Welcome to the WebSphere Portal V6.1 POT Sample Application
Welcome to the WebSphere Portal V6.1 POT Sample Web Application
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V5.1 - WebSphere Portal Introduction
30


TechWorks
Static Web Page Integration
• Allows webmasters to publish web pages directly into Portal
• Simplifies creation of portal pages
• Create smart HTML pages using live text in WebSphere Portal
• Contrast to web clipping
‣ Not a "get" from a remote site
‣ Use favorite HTML design tools and publish to WebSphere Portal
‣ Remove HTML <head> section before publishing
▪ Allows Portal to display properly inside it's own HTML
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V5.1 - WebSphere Portal Introduction
31


TechWorks
Static Web Page Integration
• Create in Adobe® Dreamweaver®
• View inside WebSphere Portal
Portal Pies
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V5.1 - WebSphere Portal Introduction
32

TechWorks

Agenda

- Value Proposition of a Portal
- What is a Portal?
- WebSphere Portal V7.0 Architecture
- Methods to quickly populate your Portal with existing content and applications
  - Feeds
  - Static HTML
  - Dynamic web applications
- WebSphere Portal delivers Exceptional Mobile Experience

TechWorks

Techniques for Creating Exceptional Mobile Experiences

Browser Based, Web Applications

Accessible over the Internet without need to download software

Uses device browser to display content

Hybrid ? Both Web and Native Components

Native looking applications which utilize the browser interface to deliver content

Provide the ability to use native device features without writing code for each device

Native

Able to make use of phone's native features such as camera, GPS, accelerometer, calendar, etc..

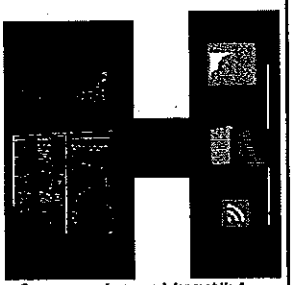
Supports the richest of user experiences (e.g., gaming applications)

TechWorks

Why Leverage a Portal Framework for Mobile Solution Delivery?

To create mobile applications that:

- **Integrate** with your back-end systems, including web content management
- Are **personalized** to a user's role, preferences, and behavior
- Support **end-user customization**
- Are **secure**
- Can be measured and optimized using **analytics**
- **Can adapt** their presentation and functionality according to the device



One common framework for mobile & web applications

TechWorks

Websphere Portal Themes for Smartphones

Extend Portal web experience to mobile devices though the use of Portal themes for smartphones

Optimizes look & feel for smaller screens (site branding, navigation, page layout)

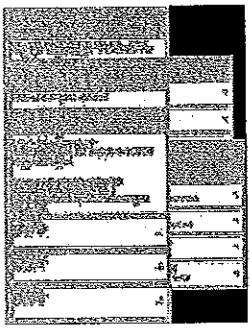
Optimizes page download size for mobile connections

Create native-looking smartphone Portal themes using familiar technologies and skillsets (HTML, CSS, Javascript)

Two mobile theme options provided:

- Dedicated mobile themes can serve dedicated mobile pages and content
- Single theme can detect the device and serve mobile presentation or full site and content.

Sample themes can be downloaded on Portal Solutions catalog



TechWorks

WebSphere Portlet Factory for Smart Phones

Automate the creation of smartphone-optimized applications, eliminating coding and speeding time to market

Support your multi-channel strategy. Create once and run on multiple devices and in multiple formats:

- Mobile-optimized applications (iPhone, Blackberry, Android, iPad)
- Portlets or widgets (WebSphere Portal)
- Web application (WebSphere Application Server)

Quickly integrate your back-end systems into your mobile applications with a robust set of connectors (SAP, Domino, relational DB, web/REST services, PeopleSoft, Siebel, and more)

Create native-looking smartphone applications using familiar technologies and skillsets (HTML, CSS, Javascript)

Leverage the smartphone application samples to speed development and learn best practice techniques, including how to best support geolocation and phones' orientation features

TechWorks

Mobile Portal Accelerator

WebSphere Portal extension that delivers online content and applications to virtually all mobile devices.

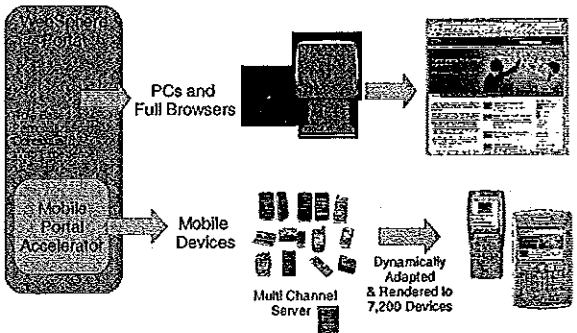
Adapt your content to over 7,200 mobile devices. MPA provides intelligent Multi-Channel Server capabilities that dynamically renders content to specific device characteristics and capabilities.

Speed your development with Mobile Portal Toolkit which enables both the development and testing of XDIME (XHTML with Device Independent Markup Extensions) portlets in IBM Rational Application Developer.

Keep pace with the latest mobile devices with IBM Mobile Portal Accelerator Device Update. Provides regular updates to mobile device definitions as they become available from device manufacturers.

TechWorks

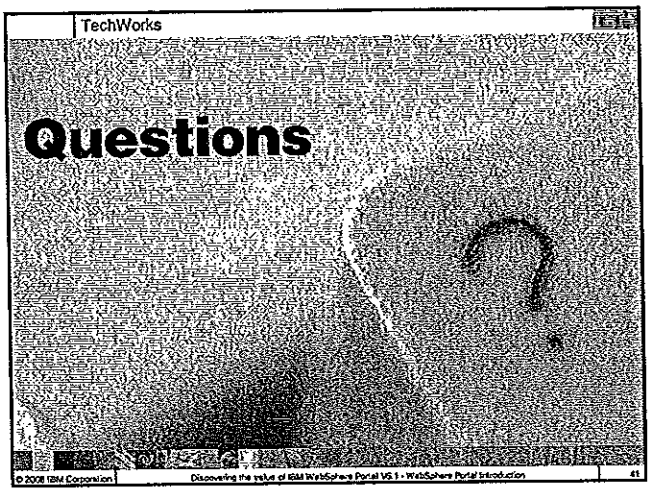
Mobile Portal Accelerator - Mobile Multi-Channel Capabilities

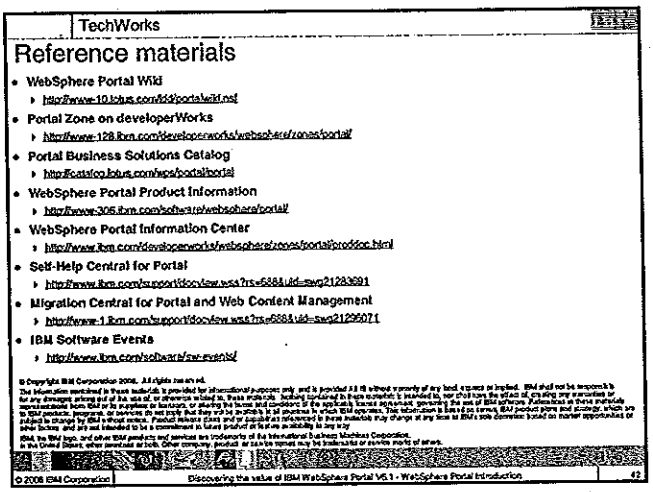


TechWorks

Presentation summary

- You should now be able to:
  - Describe the business value of WebSphere Portal
  - Explain how companies are using WebSphere Portal
  - Review how IBM uses WebSphere Portal
  - Explain the basic concepts of a portal
  - Identify the benefits of IBM Portal to IT and the business users
  - Explain the value of Web 2.0
  - Describe how Web 2.0 is incorporated into WebSphere Portal
  - Identify easy methods to quickly populate your Portal with existing content and applications
  - Use out-of-the-box WebSphere Portal features
  - Describe how WebSphere Portal delivers exceptional web experience to different end devices


TechWorks
Questions
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V6.1 - WebSphere Portal Introduction
41


TechWorks
Reference materials
WebSphere Portal Wiki
http://www-10.lotus.com/ldd/portalwiki.nsf
Portal Zone on developerWorks
http://www-128.ibm.com/developerworks/websphere/zones/portal/
Portal Business Solutions Catalog
http://catalog.lotus.com/wps/portal/portal
WebSphere Portal Product Information
http://www-306.ibm.com/software/websphere/portal/
WebSphere Portal Information Center
http://www.ibm.com/developerworks/websphere/zones/portal/proddoc.html
Self-Help Central for Portal
http://www.ibm.com/support/docview.wss?rs=688&uid=swg21283691
Migration Central for Portal and Web Content Management
http://www-1.ibm.com/support/docview.wss?rs=688&uid=swg21296071
IBM Software Events
http://www.ibm.com/software/sw-events/
© 2008 IBM Corporation
Discovering the value of IBM WebSphere Portal V6.1 - WebSphere Portal Introduction
42

Chapter 4: IBM WebSphere Portal

Overview

In this lab, you will integrate existing web content into your new IBM® WebSphere® Portal. This shows you how easily and quickly you can consolidate existing content into your Portal. You will also configure the portlets and modify your web content source to achieve communication between portlets.

- **Web Page Portlet** – configure and display existing web content

When finished, you will see a portlet for each exercise, and the specified content displayed inside each portlet.

Requirements

Image Requirements:

This lab exercise uses the Portal v7.0 image.

Hardware Requirements:

PC Hardware with a minimum of 3 gigabytes of RAM and 40 gigabytes of free disk space.

Software Requirements:

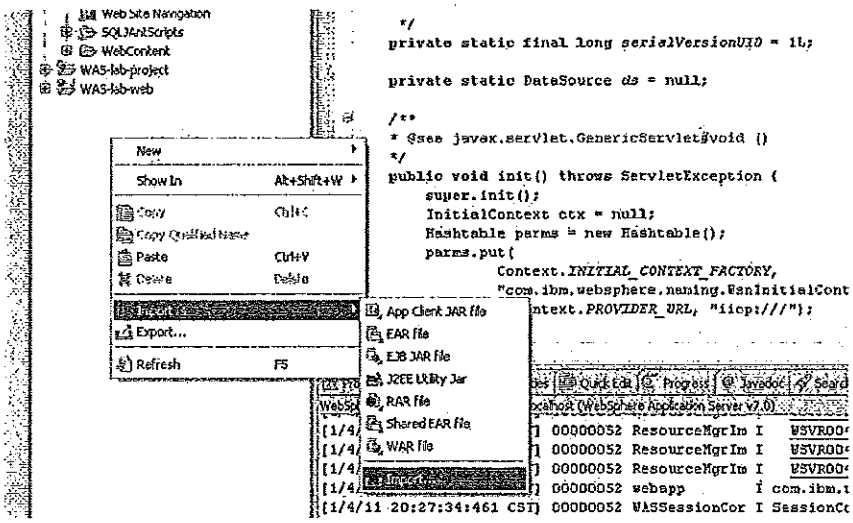
VMware Workstation version 6

Create a New Servlet

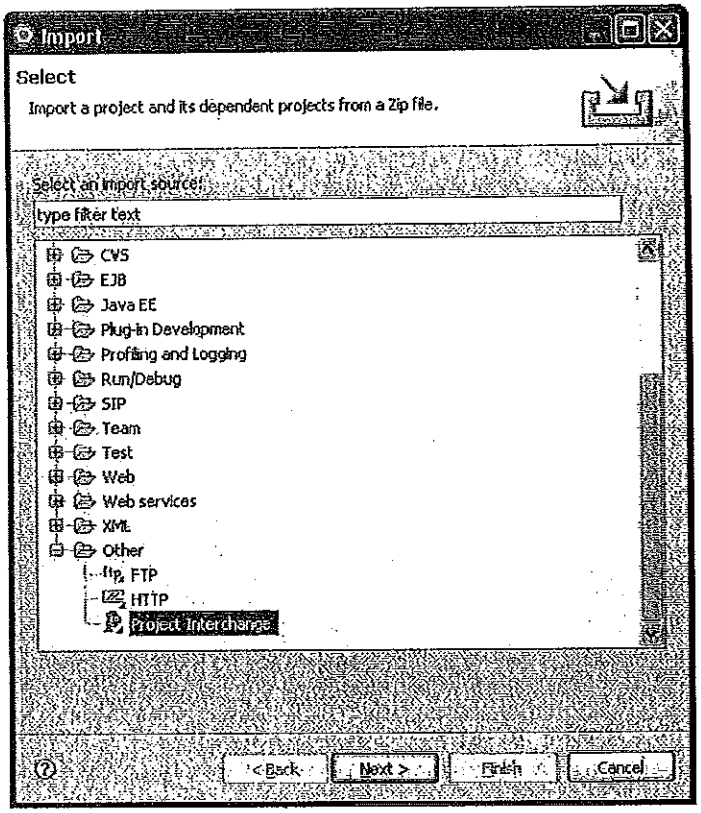
Import a project into RAD

__1. Open Rational Application Developer (RAD) and start WebSphere Application Server (WAS) if you have not done so.

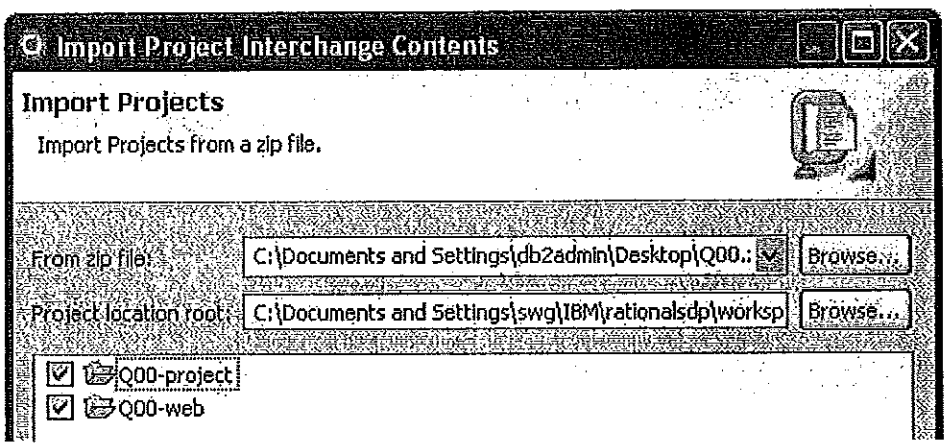
__2. On the left sidebar of RAD, right-click and select **Import** and **Import...**.



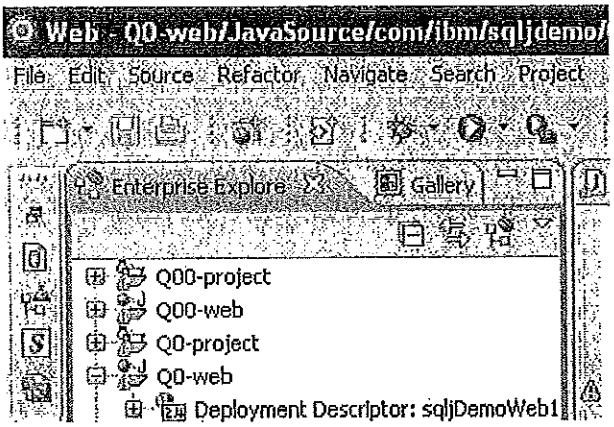
__3. Select **Project Interchange**. Click **Next**.



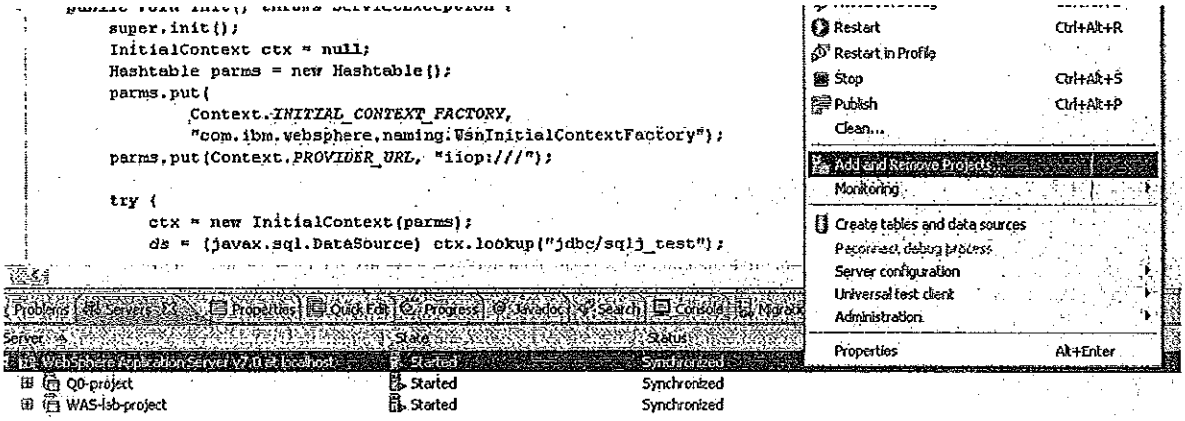
__4. Click **Browse...** and select **Q00.zip** on Desktop. Select both boxes of **Q00-project** and **Q00-web**. Then click **Finish**.



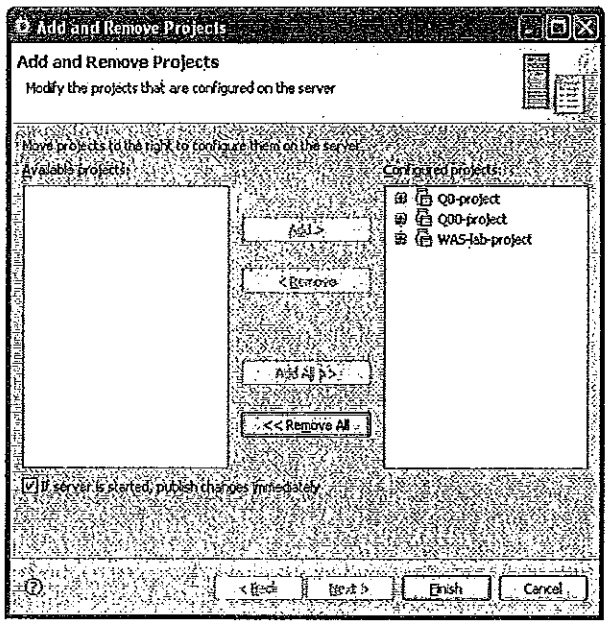
__5. You will see Q00-project and Q00-web on the left sidebar.



__6. Go to **Servers** at the bottom of RAD and right-click **WebSphere Application Server v7.0 at localhost**. Then select **Add and Remove Projects....**

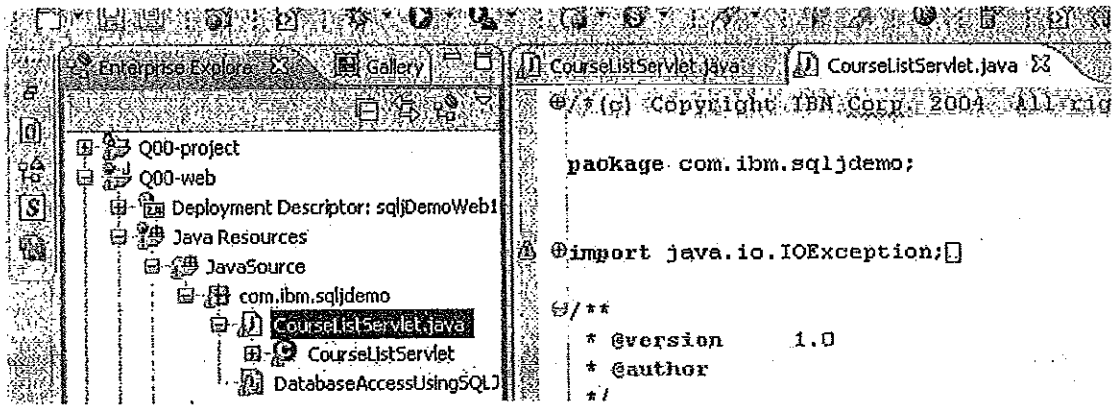


__7. Select **Q00-project** and click **Add >**.and **Finish**.



Create New Web Content

__8. Open **CourseListServlet.java** in Q00-web.



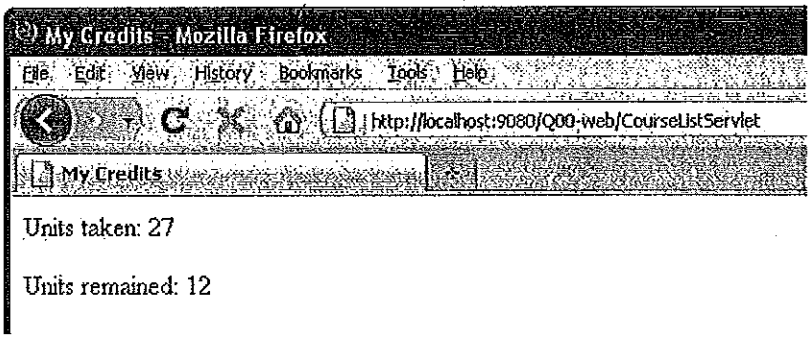
__9. Go to line 111 and change ***Q0 – Course Selection*** to ***My Credits***.

```
        /** START OF HTML HEADING **/

        os.println("<HTML>");
        os.println("<HEAD>");
        os.println("  <TITLE> My Credits </TITLE>");
        os.println("</HEAD>");
        os.println("");

        /** END OF HTML HEADING **/
```

__10. Delete from line 117 to line 122, line 124 and from line 131 to line 287, and add the 2 lines shown below. Then click **Ctrl-S** to save and compile.

```
CourseListServlet.java   *CourseListServlet.java

    public void doService(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        ServletOutputStream os;
        os = resp.getOutputStream();

        /** START OF HTML HEADING **/

        os.println("<HTML>");
        os.println("<HEAD>");
        os.println("    <TITLE> My Credits </TITLE>");
        os.println("</HEAD>");
        os.println("");

        /** END OF HTML HEADING **/

            displayCourseList(req, resp, os);
    }

    private void displayCourseList(HttpServletRequest req, HttpServletResponse resp, ServletOutputStream os){
        try {
            os.println("<BODY>");
            os.println("<P>Units taken: 27</P>");
            os.println("<P>Units remained: 12</P>");
            os.println("</BODY>");
            os.println("</HTML>");
            os.close();
        } catch (IOException ie) {
            System.out.println("IOException:" + ie.toString());
        } catch (Exception e){
            System.out.println("Exception:" + e.toString());
        }
    }
```

__11. Open a new browser and enter the URL http://localhost:9080/Q00-web/CourseListServlet and you should see the new web content in the servlet you just modified.

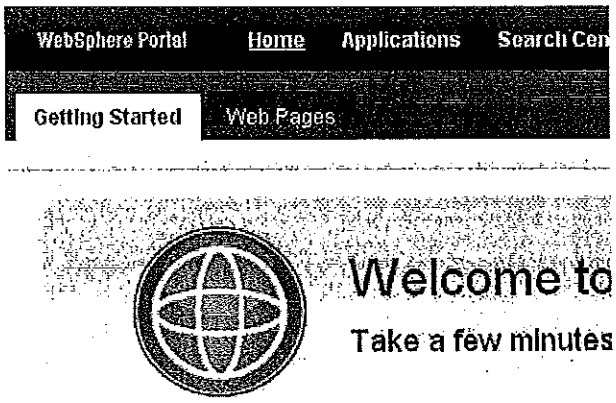


Configure a Web Page Portlet

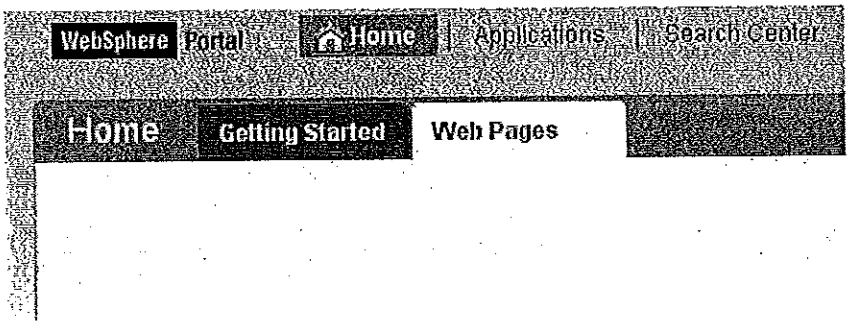
Add Web Page Portlets to a Portal Page

Follow these steps to add two web page portlets to a portal page.

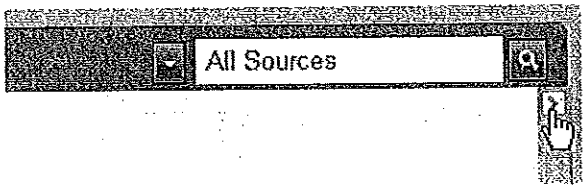
__1. Open a new browser and enter URL http://192.168.10.2/wps/portal. Login to WebSphere Portal using **team*X*/team*X*** as your credentials. ***X*** is your team number. For example team 1 uses **team1/team1**, team 30 uses **team30/team30.**



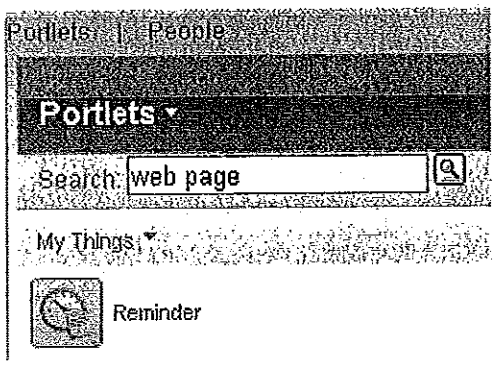
__2. Click the **Web Pages** tab. The page is blank, since no portlets are on this page.



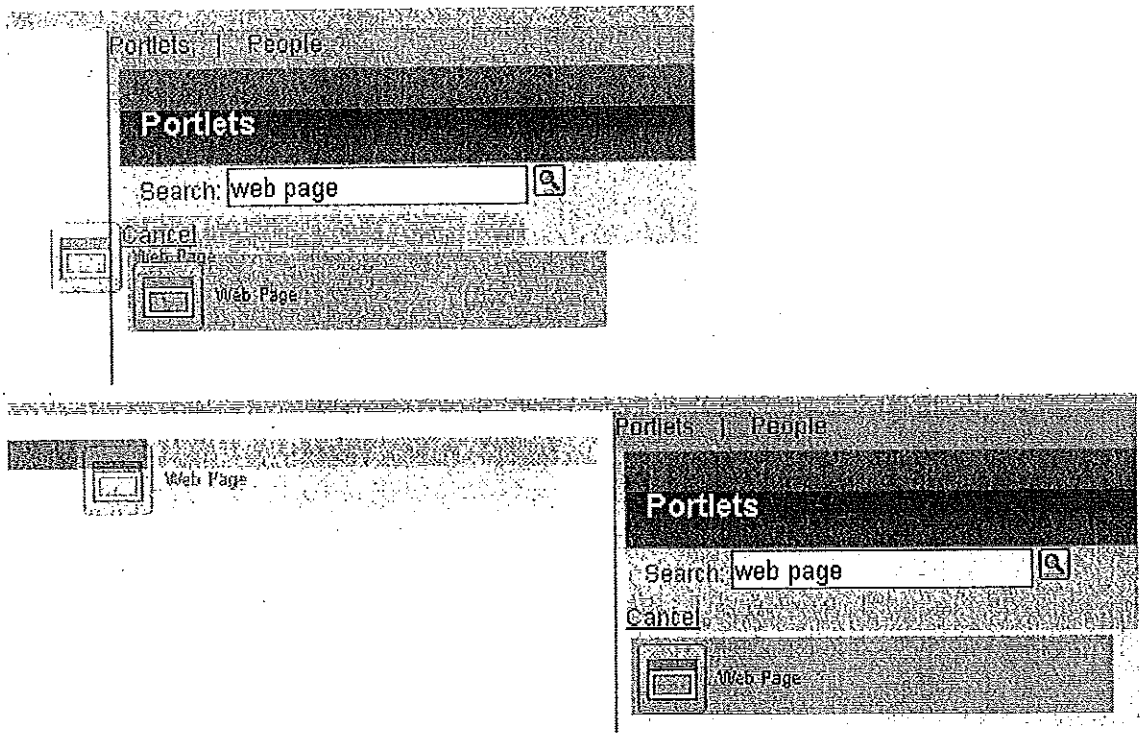
__3. Click on the Portal's "Expand Palette" icon.



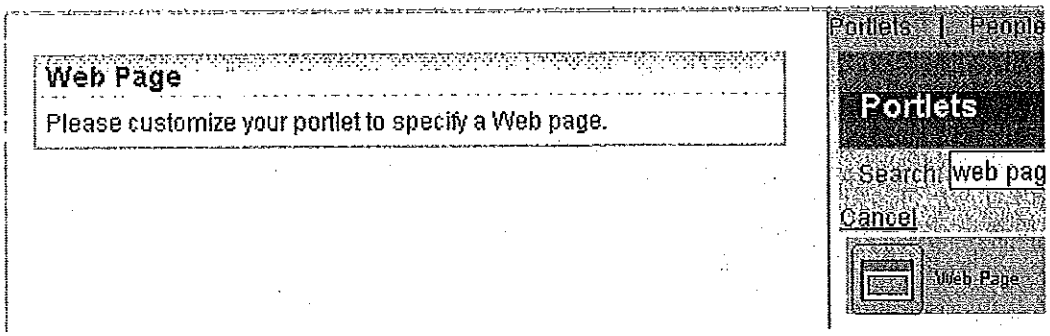
__4. TYPE *web page* in the search field. Click the  icon.



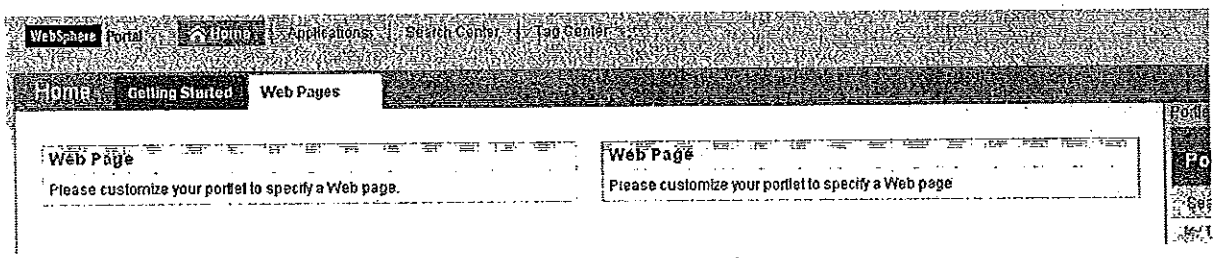
__5. Drag and drop the portlet onto the top of the blank Portal page. Place it on top of one of the colored bars in the white space.



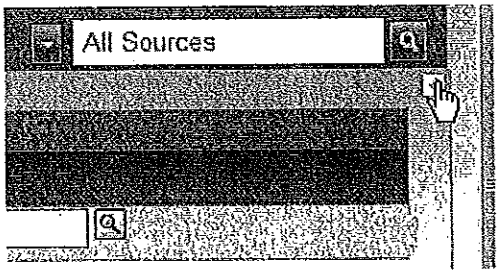
__6. You should see the portlet. If it did not work, try again.



__7. Add one more to the right hand side of the first one.

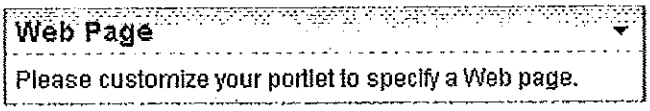


__8. Hide the portlet palette by clicking on expand palette the icon again.

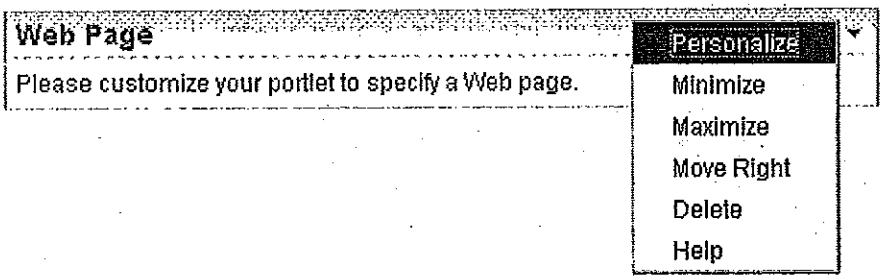


Configure the portlet

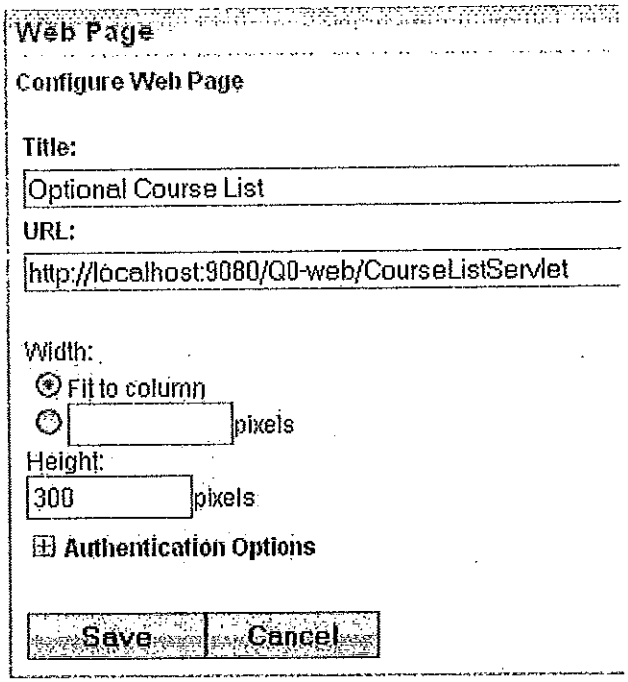
__9. Hover over the top-right corner of the portlet. You will see a drop-down arrow.



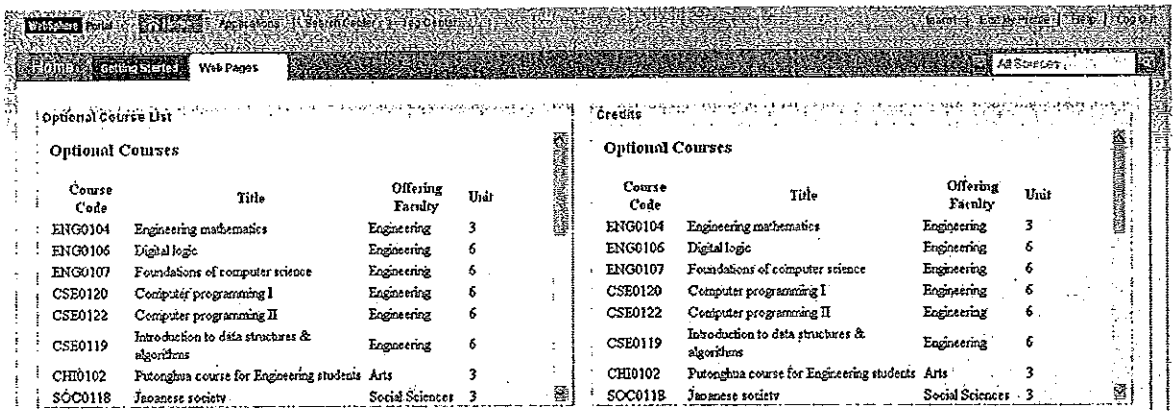
__10. Click on the arrow and select **Personalize.**



__11. Type ***Optional Course List*** in the Title field and ***http://localhost:9080/Q0-web/CourseListServlet*** in the URL filed. Then click **Save**. The Title you enter will be the name of the iframe in this Web Page portlet. You will need to use this iframe name later to edit the web content source.



__12. Do the same for the other portlet, but enter ***Credits*** as the Title. Then you should see both portlets displaying the Option Course list.



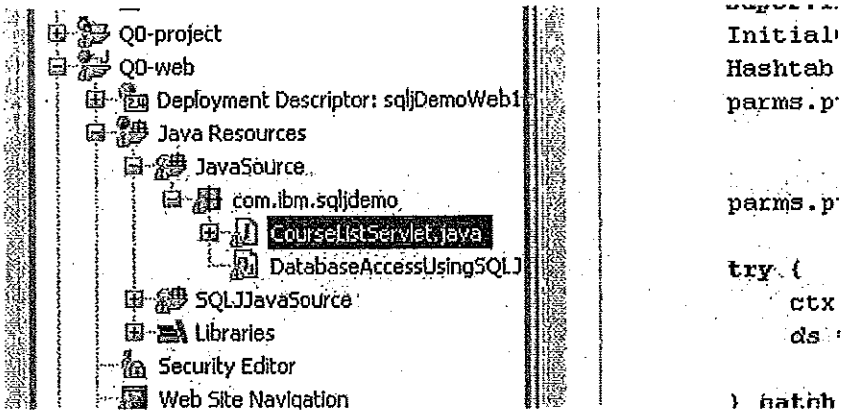
__13. **Optional:** You can modify the size of the Web Page portlet (size of the iframe) by entering the pixels of width and height.

Develop Communications between Portlets

Trigger Refresh Action between Portlets

The following steps show how to edit the web content source in RAD so that when a button is clicked in a portlet, the other portlet will refresh and show the web content that you would like to display. You will add a button named "My Credits" in the "Optional Course List" portlet. When you click the button, the "Credits" portlet will automatically refresh to the "My Credits" web page, which is the new web content (servlet) that you created at the beginning of this exercise.

__1. Open **CourseListServlet.java** in Q00-web in RAD.



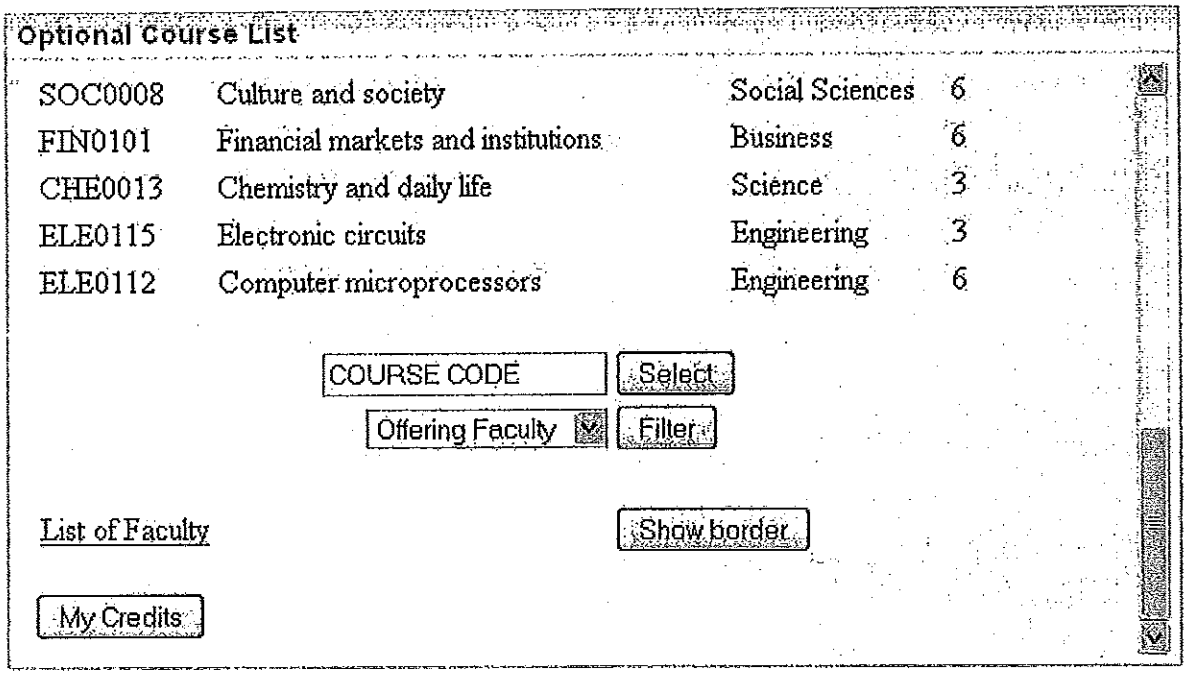
__2. Add the following lines after line 111. Notice that you have put the iframe name ***Credits*** and the path ***"../Q00-web/CourseListServlet\"*** in the source. It means that in this function "refresh", it triggers the portlet with iframe name ***Credits*** to refresh and display the new URL ***http://localhost:9080/Q00-web/CourseListServlet***.

```
        /** START OF HTML HEADING **/

        os.println("<HTML>");
        os.println("<HEAD>");
        os.println("    <TITLE> Q0 - Course Selection </TITLE>");
        os.println("    <SCRIPT TYPE=\"text/javascript\">");
        os.println("    function refresh(){");
        os.println("        window.parent.Credits.location.assign(\"../Q00-web/CourseListServlet\");");
        os.println("    }");
        os.println("    </SCRIPT>");
        os.println("</HEAD>");
        os.println("");

        /** END OF HTML HEADING **/

        String tran = req.getParameter("action");
```

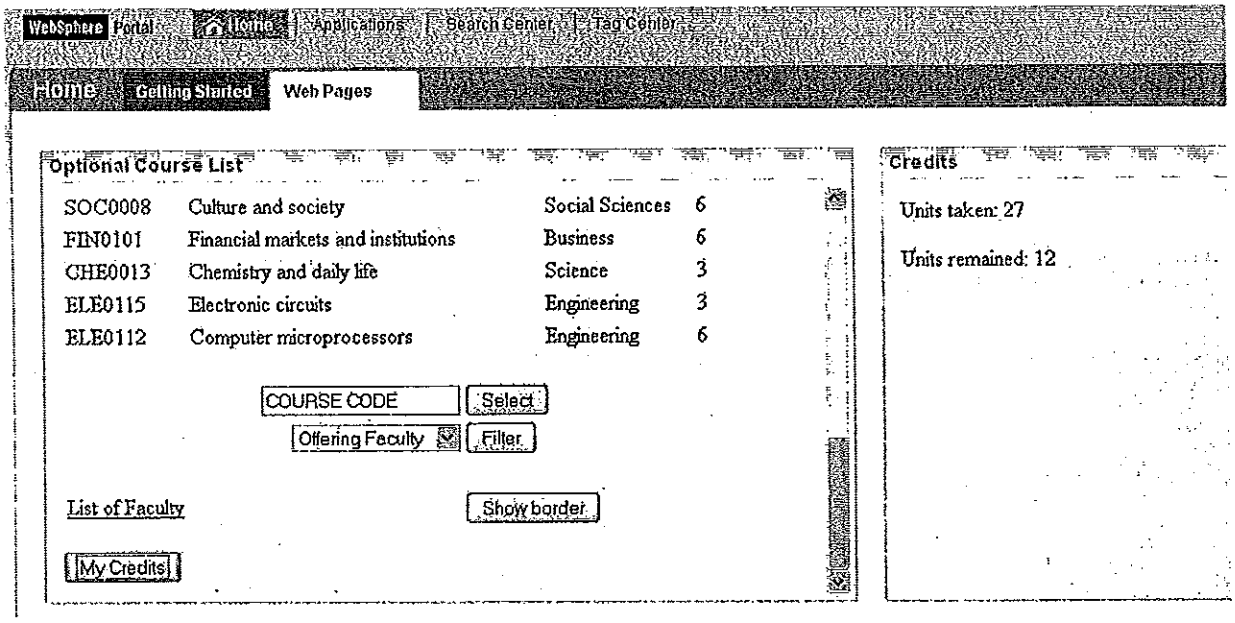
__3. Add the following line after line 270. This adds a button that triggers the "refresh" function to the end of the "Optional Course List" portlet.

```
            os.println("          </TD>");
            os.println("    </TR>");
            os.println("</TABLE>");

            os.println("<BUTTON ONCLICK=\"refresh()\">My Credits </BUTTON>");

            os.println("</BODY>");
            os.println("</HTML>");
            os.close();
        } catch (IOException ie) {
            System.out.println("IOException:" + ie.toString());
```

__4. Click **Ctrl-S** to save and compile. Then go back to WebSphere Portal and refresh the portal page. Scroll down the "Optional Course List" portlet and you should see the button "My Credits".



__5. Click the **My Credits** button and you should see the portlet on the right has refreshed to display the "My Credits" web page that created earlier.



End of Exercise

IBM

Information Management

IBM Inter-University Programming Contest 2011 Training

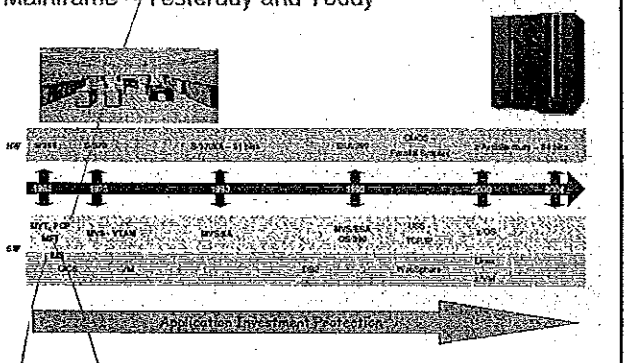
Chapter 5: Access Mainframe DB2 from WebSphere

Software Group, IBM

Chapter Outline

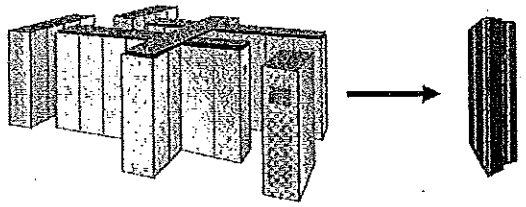
- Mainframe History
- Mainframe Major Hardware Features
- Operating Systems running on Mainframe
- DB2 Connect Topologies

Mainframe – Yesterday and Today



Information Management | IBM

The changing mainframe footprint

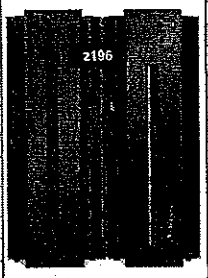


9021-9X2 - 470 MIPS in 1994
Bipolar, Water Cooled Mainframes transition to smaller CMOS technology

Information Management | IBM

zEnterprise z196 Functions and Features (GA Driver 86 – August, 2010)

Five hardware models	2 New OSA CHPIDs – OSX and OSM
Quad core CPU chip	Three subchannel sets per LCSS
Up to 80 processors configurable as CPs, zAAPs, zIPs, IFLs, ICFs, or optional SAPs	8 slot, 2 domain I/O drawer
Increased capacity processors	Concurrent I/O drawer add, remove, replace
Out of order instruction execution	FICON discovery and autoconfiguration
Over 100 new and enhanced instructions	Doubled HiperSockets to 32
Improved processor cache design	Physical Coupling Links increased to 80
Up to 15 subcapacity CPs at capacity settings 4, 5, or 6	Doubled Coupling CHPIDs to 128
Up to 3 TB of Redundant Array of Independent Memory (RAIM)	CFCC Level 17
Unified Resource Manager suites	Optional water cooling
Cryptographic enhancements	Optional High Voltage DC power
On Demand enhancements	Optional overhead I/O cable exit
Energy efficiencies	STP enhancements
	zBX-002 with IBM Smart Analytics Optimizer, IBM Blades



Information Management | IBM

IBM Disk Storage

Mainframe environment > IBM disk storage: DS8000

A more recent disk system, the DS8000 series, is based on ESS 800 technology, so it also works with open systems and has built-in hardware-based disaster recovery functions. These built-in functions maintain a synchronous copy—always up-to-date with the primary copy—of data in a remote location. Customers use this backup copy to quickly recover from failures without losing any transactions or data.

The amazing difference between the ESS 800 and the DS8000 is capacity: With the DS8000, IBM has increased potential disk storage capacity from 55 up to 192 terabytes, in a footprint that is approximately 20% smaller than the ESS 800.

Information Management | IBM

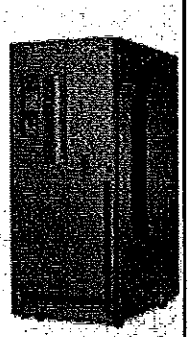
IBM Tape Storage

Mainframe environment > IBM tape storage

Just as disk storage improves significantly from one generation of machines to the next, tape storage devices also have come a long way since the archetypal reel-to-reel machines of the early 1960s. Although modern tape storage machines might be an equivalent size to the reel-to-reel machines, their design and technology is far more advanced and provides exponentially improved capacity.

An example of a modern tape storage device is the IBM System Storage™ TS3500 Tape Library, which resides in one base frame that can be connected with up to 15 additional expansion frames. The TS3500 base and expansion frames each have a footprint of approximately 10 square feet (almost one square meter), so a full bank of frames has a footprint of 160 square feet (almost 15 square meters).

Inside the TS3500 frames are tape controllers or drives, which connect the tape library with the z9 EC, and tape cartridges, which store data.



Who uses mainframes?

- Most Fortune 1000 companies use a mainframe environment
- 60% of all data available on the Internet is stored on mainframe computers
- Typical usage
  - Large-scale transaction processing
    - Thousands of transactions per second
  - Support thousands of users and application programs simultaneously accessing resources
  - Terabytes of information in databases
  - Large-bandwidth communications
- There are more CICS transactions processed daily than Web pages served

The foundation of modern business!
The new world needs Integration with the legacy Business!

Why customers choose zSeries ?

- Highest availability
  - 52 * 7 * 24 99.999%
  - Planned as well as unplanned outages
- Capability & Scalability
  - Multiple LPARs and I/O sharing
  - Load balancing between workloads
  - Workload priority management
- Largest database capability
  - Multi-terabytes
  - On line backup/recovery
- Maximum investment protection & leverage
  - Use existing zSeries skills
  - Incremental upgrades
  - Platform compatibility and stability over many years
- Security
  - Business data is critical and sensitive company & customer information
  - Potential damage through unauthorized access often underestimated
  - zSeries is the acknowledged industry leader in all aspects of IT security

Mainframe Operating Systems

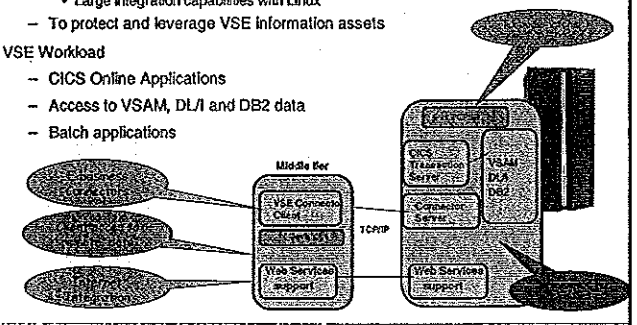
- z/VSE
- z/VM
- z/Linux
- z/TPF
- z/OS

z/VSE

- z/VSE - Replacement for VSE/ESA
  - To enable network integration and infrastructure simplification
    - Large integration capabilities with Linux
  - To protect and leverage VSE information assets

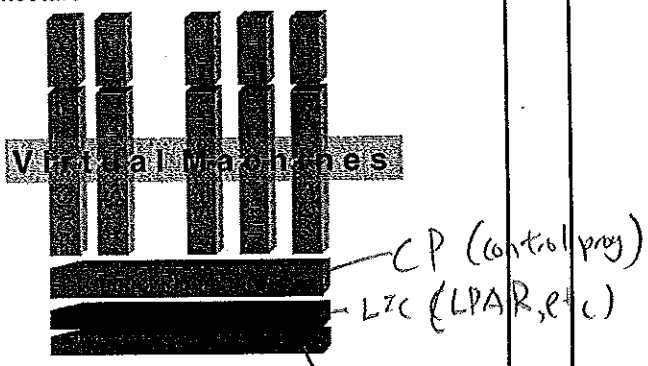
VSE Workload

- CICS Online Applications
- Access to VSAM, DL/I and DB2 data
- Batch applications



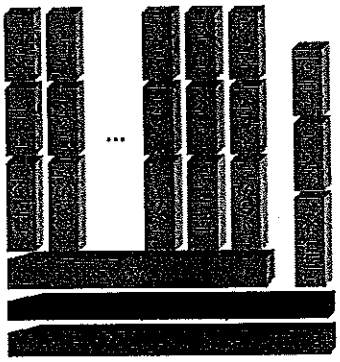
Information Management

z/VM architecture



Information Management

Linux on zSeries ...



Information Management

Linux on zSeries (zLinux)

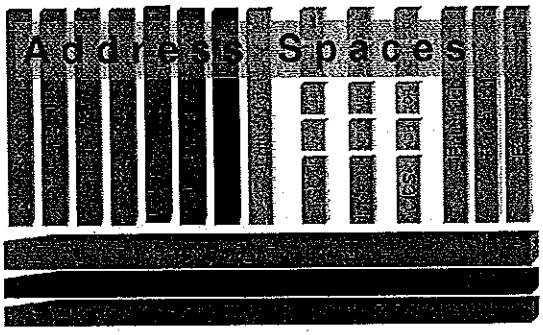
- What is it?
  - A native zSeries operating environment
    - Pure Linux, an ASCII environment (vs. EBCDIC on z/OS & z/VM)
    - Exploits IBM zSeries hardware, including IEEE floating point
    - Linux for S/390 - 32-bit
    - Linux for zSeries - 64-bit
  - Not a unique version of Linux or other operating system
  - Not a replacement for other IBM zSeries operating systems
- zSeries Linux benefits
  - The most reliable hardware platform available
  - Scalability
    - Both Physical and Logical
    - Non-disruptive capacity upgrade on demand
  - Designed to support mixed work loads
    - Complete work load isolation
    - High speed inter-server connectivity
    - High Internal Bandwidth
    - Virtualization

Information Management

z/Transaction Processing Facility Enterprise Edition (z/TPF) v1.1

- High-performance operating system
  - 64bit architecture,
  - large in-memory data tables.
  - Up to 32 z/TPF images connectable.
- Special purpose system
  - High volume transaction processing
  - High availability
- Open development environment - z/TPF
  - uses the GNU tool chain
  - can share applications, tooling, and development infrastructure with, Linu
- Customer usage
  - Airline reservation systems
  - Credit card companies
  - Public utilities (eg. Railways)
- The z/Transaction Processing Facility Database Facility (z/TPFDF) (5748-F15), provides database management functions to z/TPF.
  - To virtualize the physical database.

z/OS – The ultimate zSeries OS – Components …



z/OS component terminology

JES2 or JES3 - Job Entry Subsystem
TSO - Time Sharing Option
ISPF/PDF – Interactive System Productivity Facility
WLM – Work Load Manager
RMF – Resource Measurement Facility
SMF – System Measurement Facility
RACF – Resource Access Control Facility
Communications Server for z/OS - VTAM and TCP/IP
DFSMS – Data Facility Storage Management Subsystem
USS – Unix System Services

AM - access method

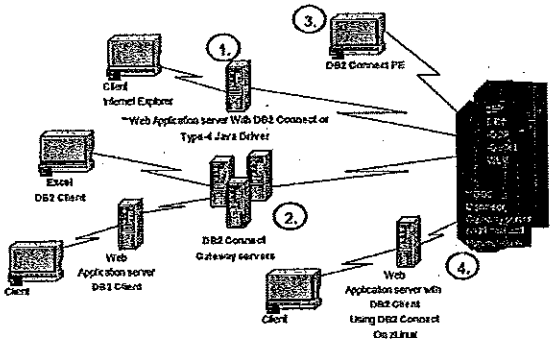
z/OS provides the platform for access to Data and Databases

- **Traditional data access methods & formats on z/OS**
  - VSAM (KSDS, ESDS, RRDS, Linear),
  - Sequential (SAM, OSAM,BDAM, ..),
  - Partitioned (PDS, PDSE)
- **Hierarchical File System for Unix System Services**
  - **/etc/....**
- **z/OS Database**
  - Hierarchical Database
    - IMS Database Manager, also called DL/I
  - Relational Database
    - DB2
  - 3rd Party Vendors – IDMS, CA-Datacom, Adabas, Oracle
- **Information Integration with existing z/OS data sources**
  - "Websphere Information Integrator Classic Federation for z/OS"

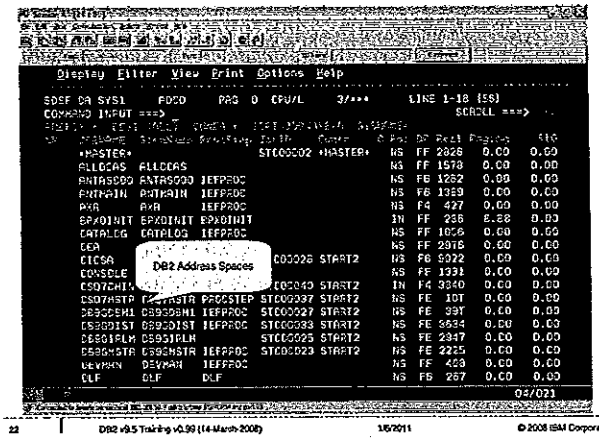
The classic mainframe TSO 'green screen'

Information Management

Summary of DB2 Connect Topologies - Roadmap:



Information Management



Information Management

System z ISV Ecosystem

Dramatic growth responding to market demand

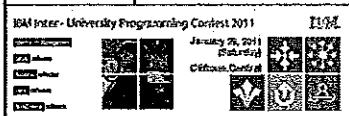
- 1,142 ISVs added to the System z portfolio in YTD 2009
- 900 New System z applications & tools YTD 2009
- 1,700 + ISVs developing for our System z Ecosystem
- 3,500 + applications available for z/OS
- 3,000 + applications available for Linux on System z
- 6,300 + applications available for System z

Information Management

IBM Inter-University Programming Contest 2009 Training

Chapter 5: DB2 Operation Overview

Software Group, IBM



1/6/2011

Chapter 5: Access Mainframe DB2 From WebSphere

Objectives

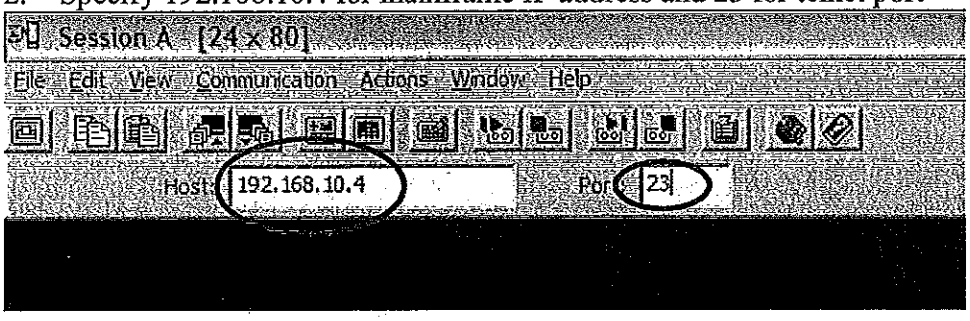
In this exercise, we will learn:

- How to login Mainframe using TN3270 terminal emulator
- How to delete table using DB2 tool on Mainframe
- How to create table, index and insert 1 record using DB2 tool on Mainframe
- How to login WebSphere Administration Console
- How to create authentication alias for Mainframe access
- How to modify local Data source to access Mainframe DB2
- How to run the servlet to access Mainframe DB2

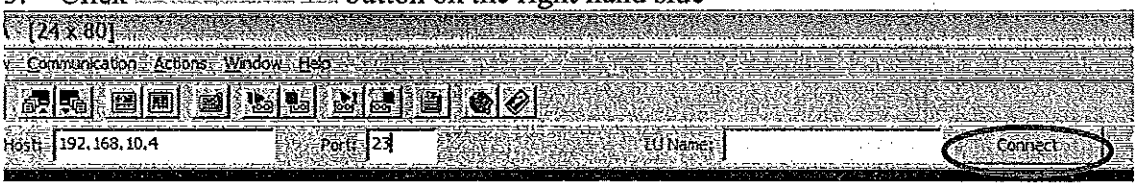
Exercises

1. Click the icon to start TN3270 terminal emulator from Desktop:

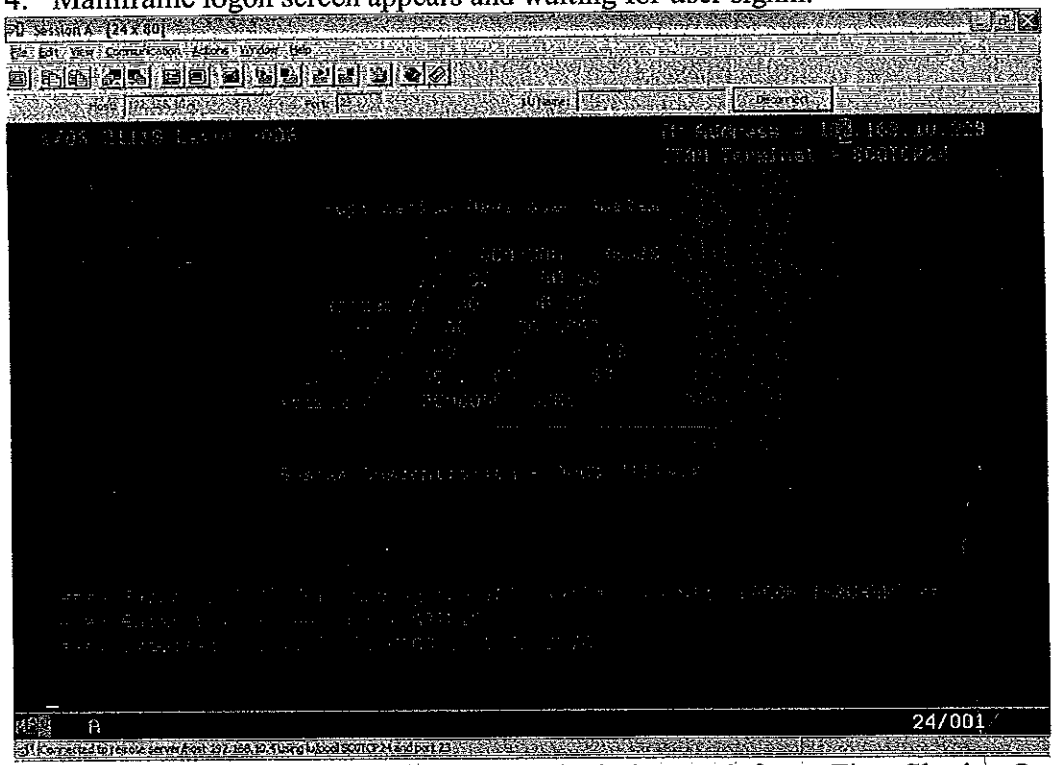
2. Specify 192.168.10.4 for mainframe IP address and 23 for telnet port



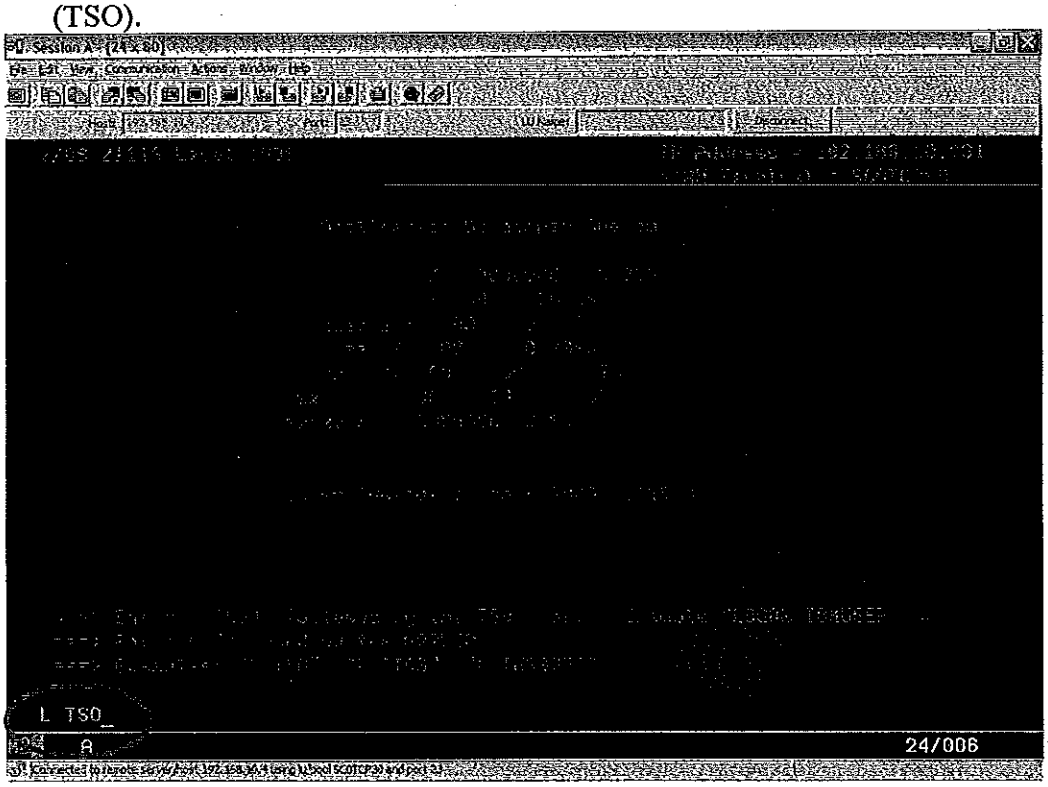
3. Click Connect button on the right hand side



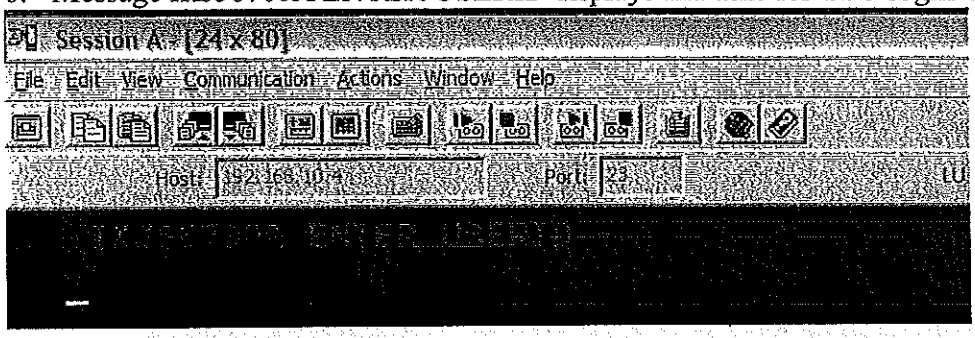
4. Mainframe logon screen appears and waiting for user signin.



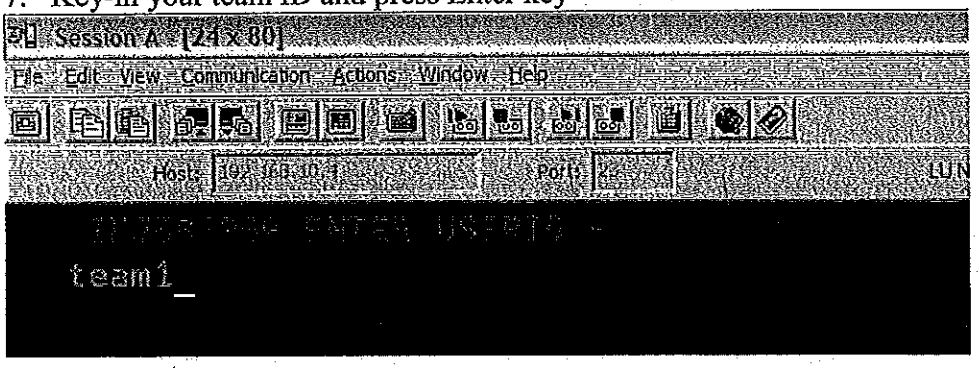
5. Key-in 'L TSO' and press Enter key to login into mainframe Time Sharing Option (TSO).



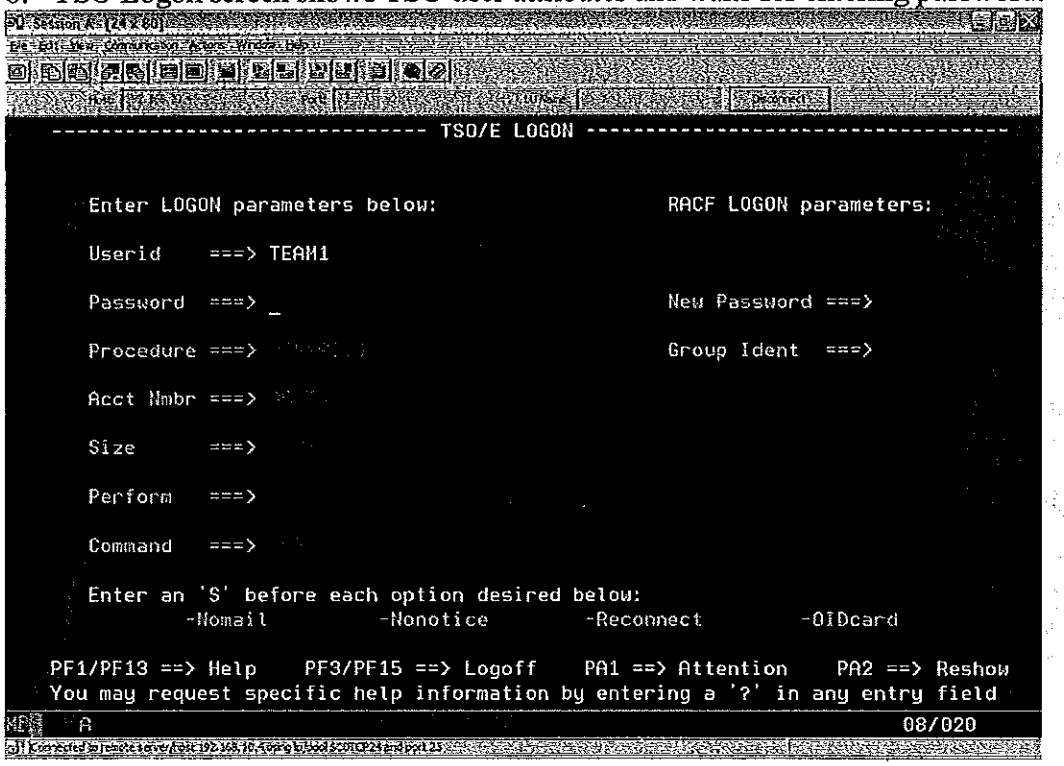
6. Message IKJ56700A ENTER USERID displays and asks for TSO login USER ID



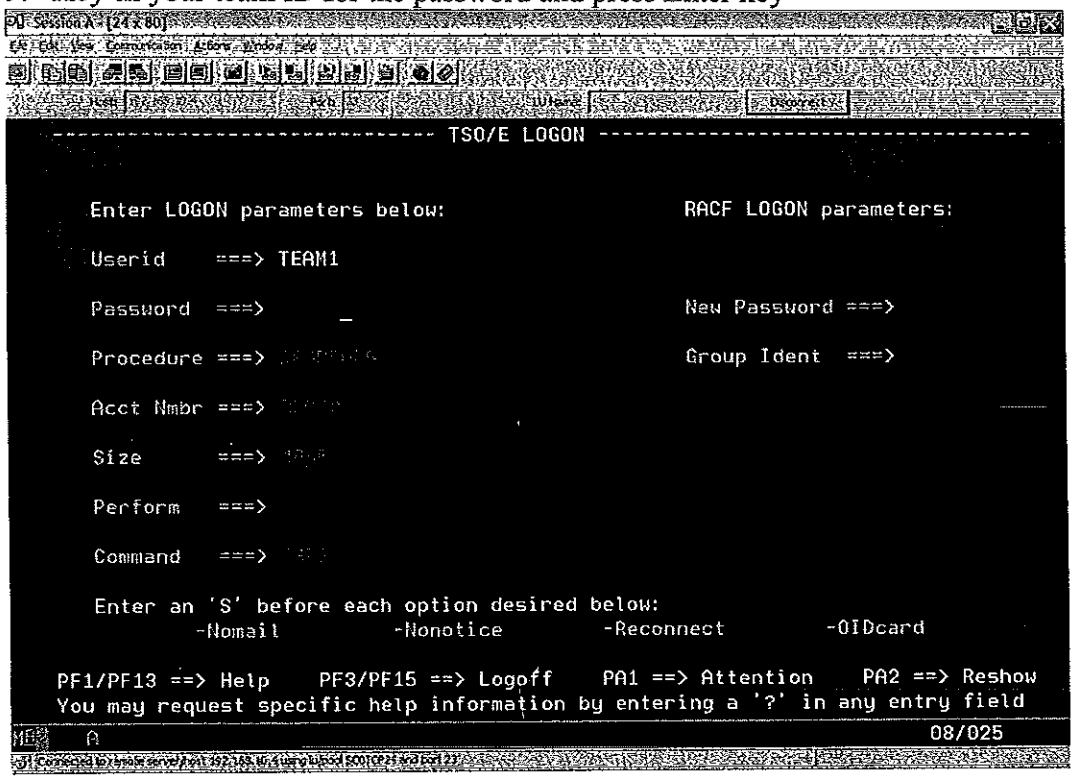
7. Key-in your team ID and press Enter key



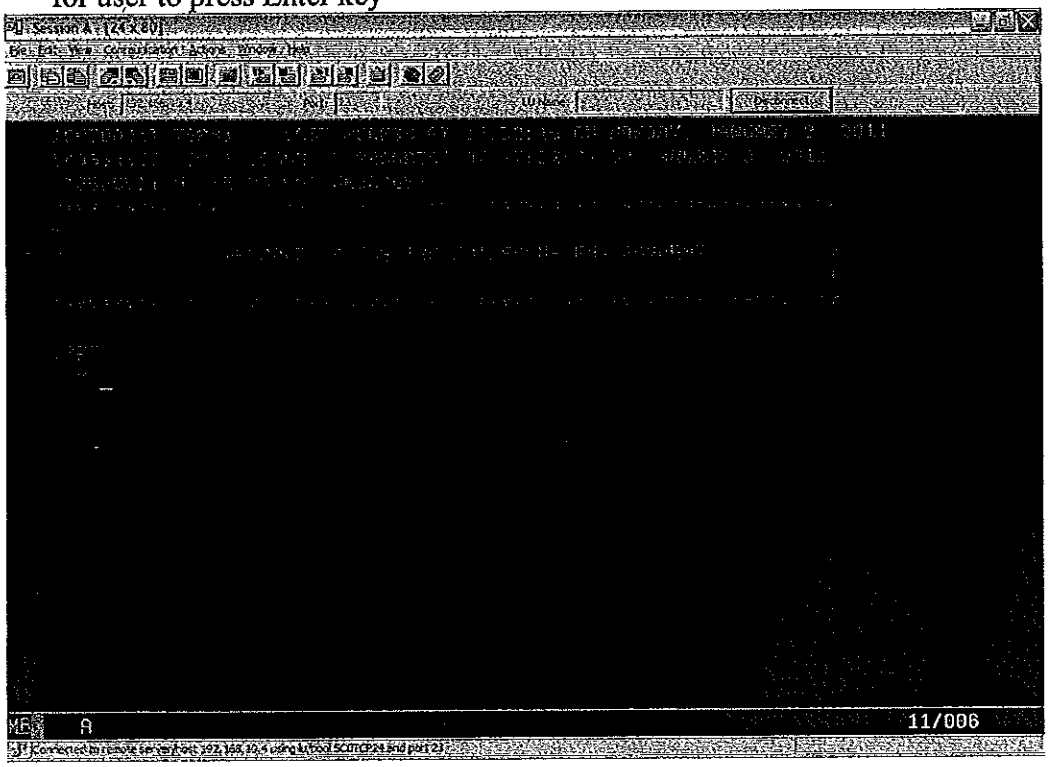
8. TSO Logon screen shows TSO user attributes and waits for entering password.



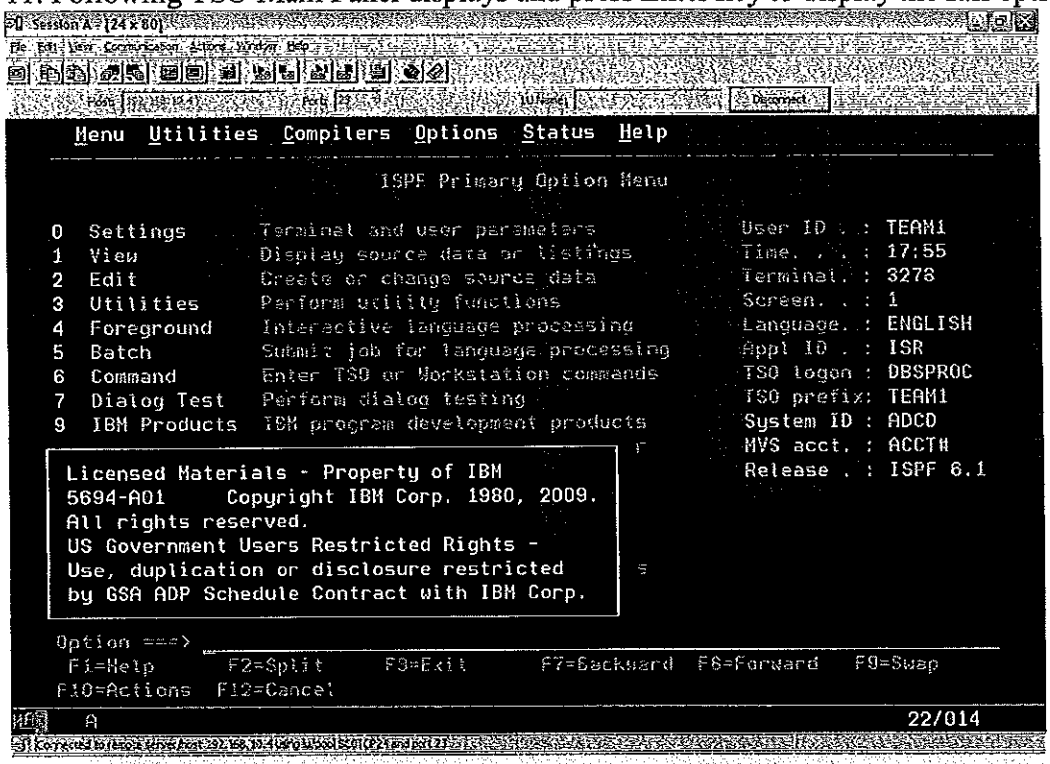
9. Key-in your team ID for the password and press Enter key



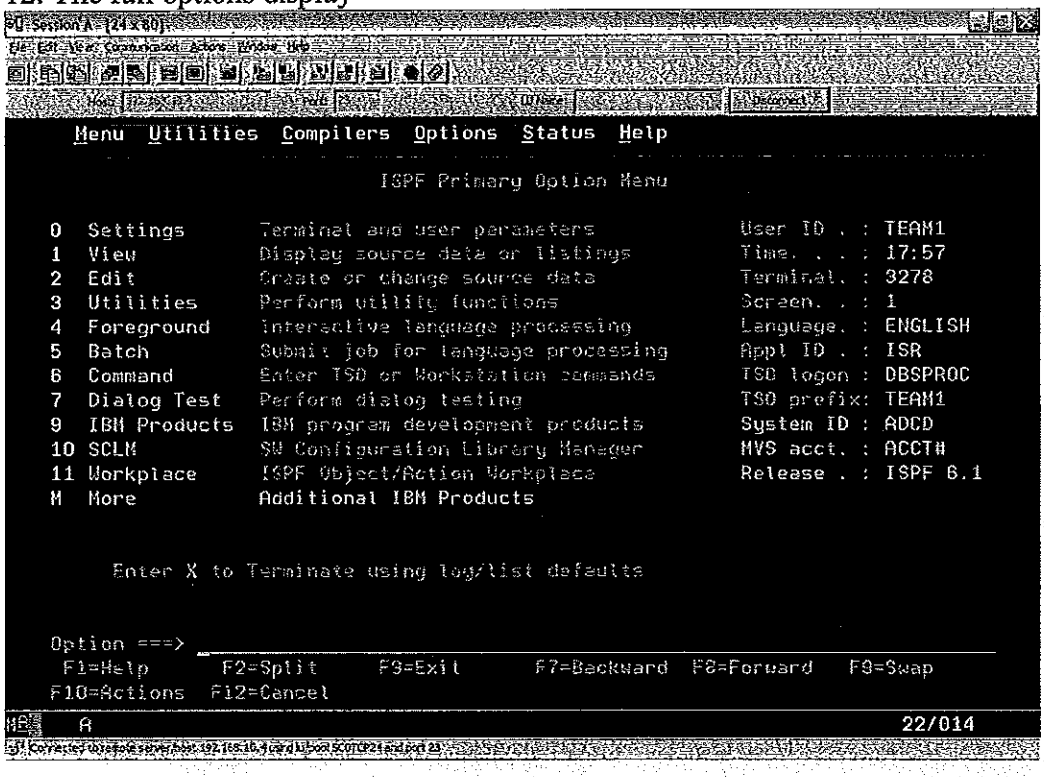
10. After entering correct password, the following TSO welcome page displays and wait for user to press Enter key



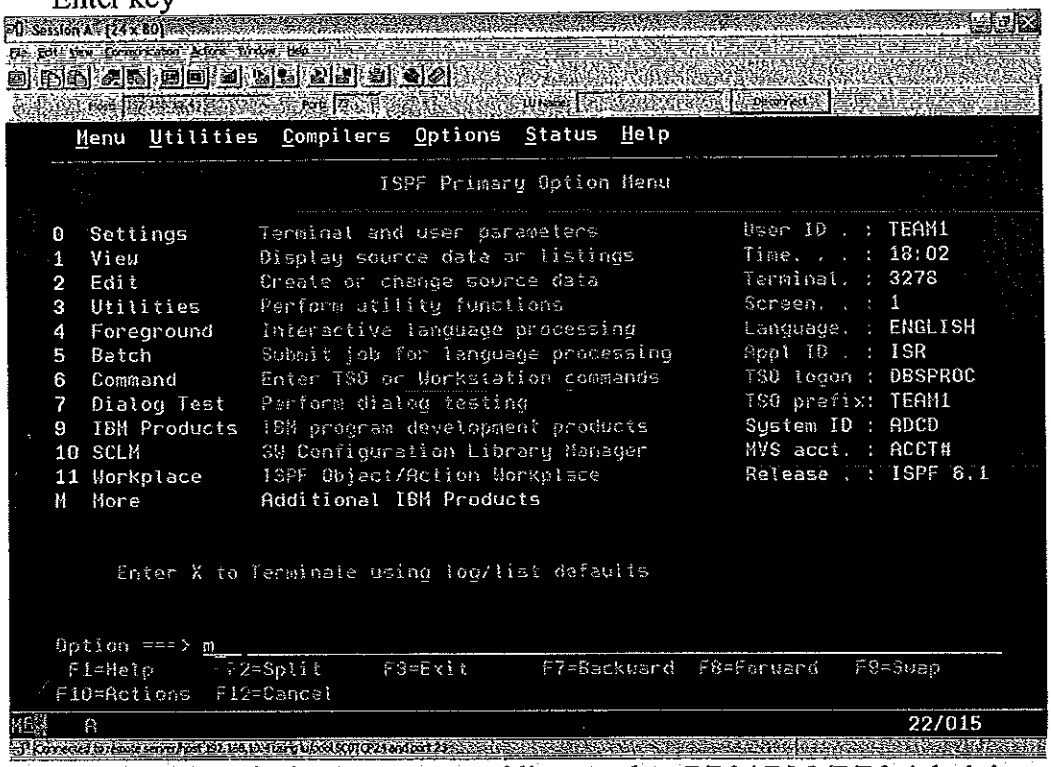
11. Following TSO Main Panel displays and press Enter key to display the full options.



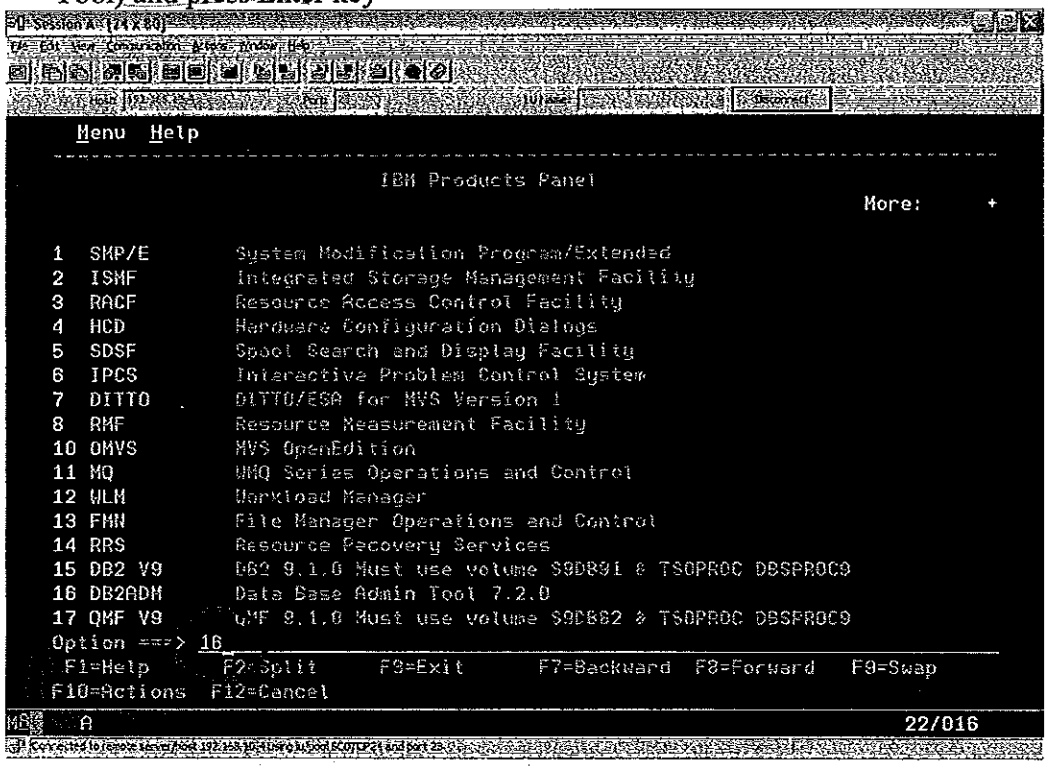
12. The full options display



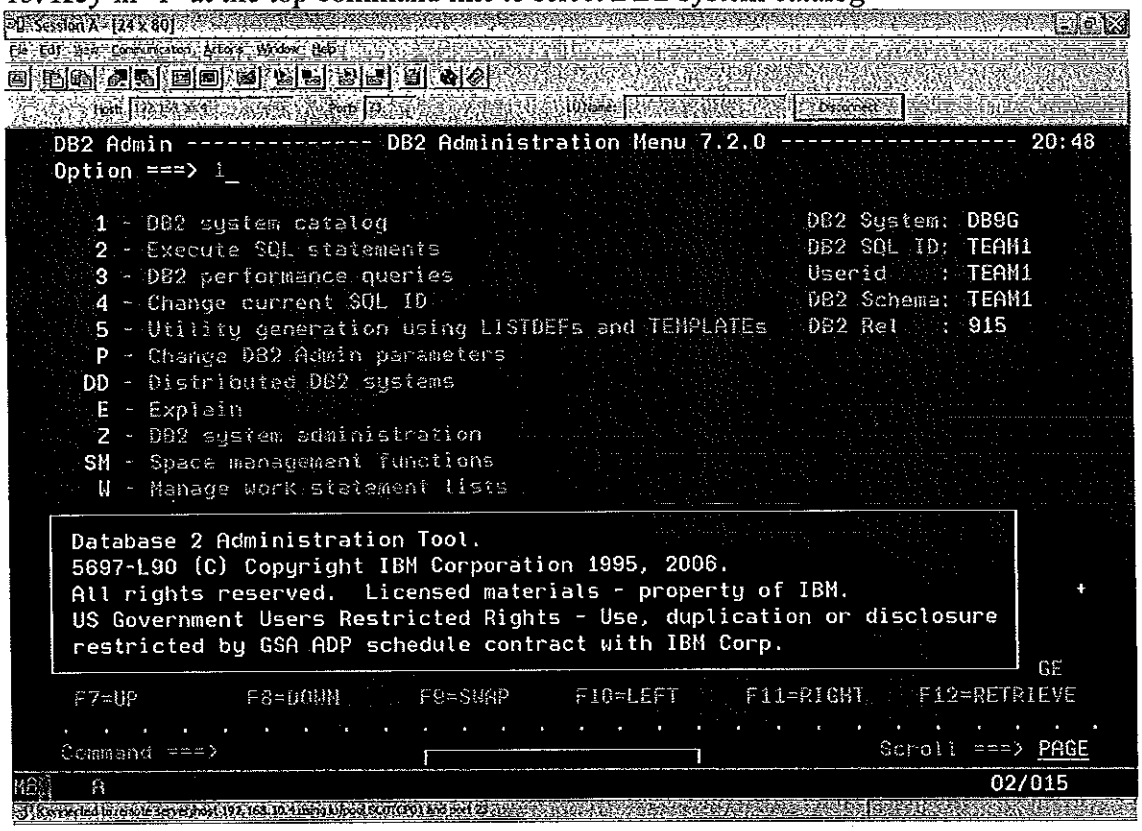
13. Key-in 'm' at the bottom command line to select Additional IBM products and press Enter key



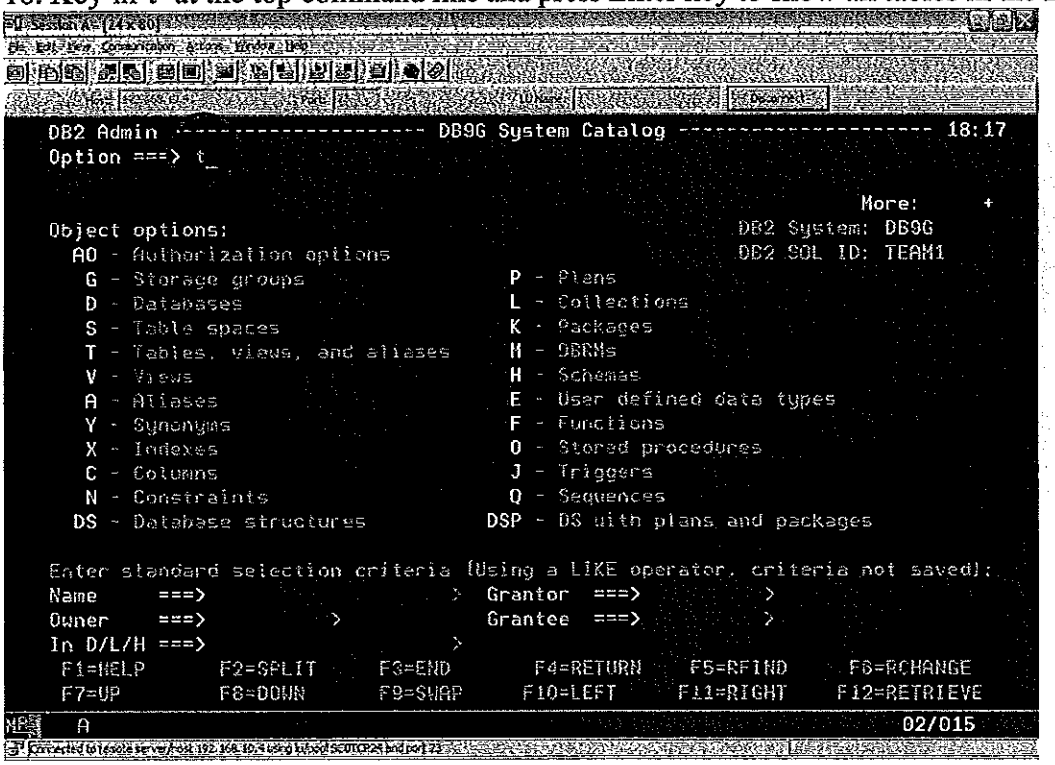
14. Key-in '16' at the bottom command line to select DB2ADM (DB2 Administrator Tool) and press Enter key



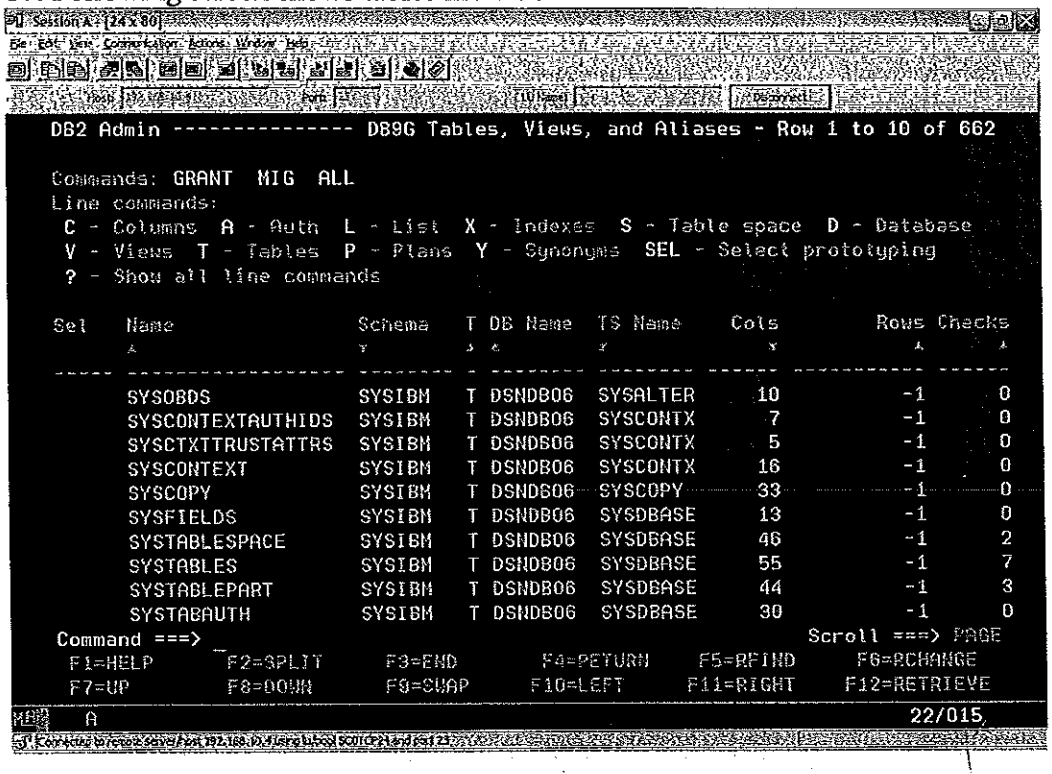
15. Key-in '1' at the top command line to select DB2 system catalog



16. Key-in't' at the top command line and press Enter key to show all tables in the DB2.

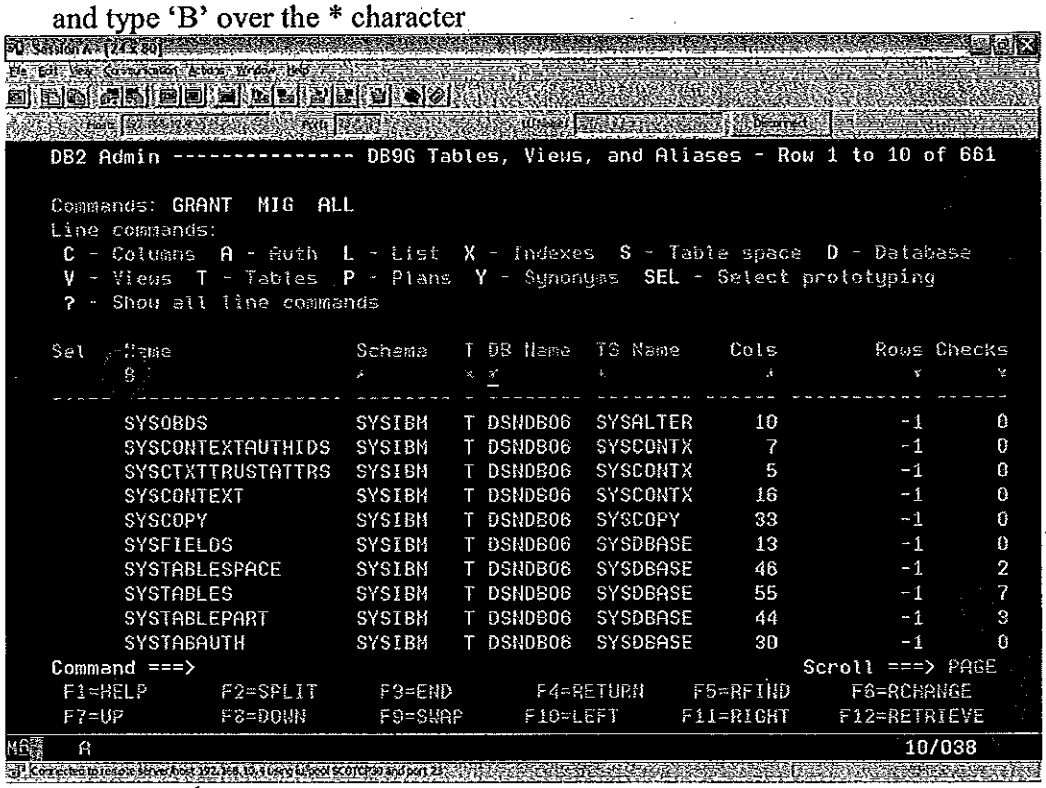


17. Following screen shows tables in DB2.



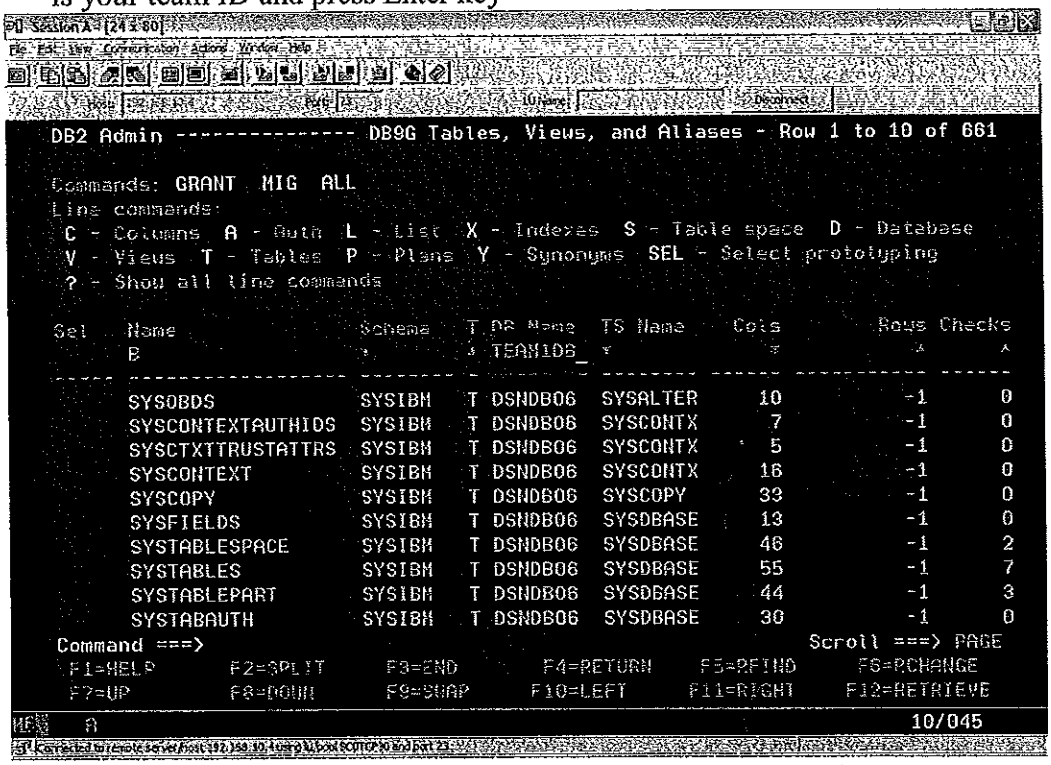
```
DB2 Admin --------------- DB9G Tables, Views, and Aliases - Row 1 to 10 of 662

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Schema   T DB Name  TS Name    Cols        Rows Checks
      ^                   v        ^ ^        v          v           ^    ^
----- ------------------- -------- - -------- -------- ------ ----------- ------
      SYSOBDS             SYSIBM   T DSNDB06  SYSALTER     10          -1      0
      SYSCONTEXTAUTHIDS   SYSIBM   T DSNDB06  SYSCONTX      7          -1      0
      SYSCTXTTRUSTATTRS   SYSIBM   T DSNDB06  SYSCONTX      5          -1      0
      SYSCONTEXT          SYSIBM   T DSNDB06  SYSCONTX     16          -1      0
      SYSCOPY             SYSIBM   T DSNDB06  SYSCOPY      33          -1      0
      SYSFIELDS           SYSIBM   T DSNDB06  SYSDBASE     13          -1      0
      SYSTABLESPACE       SYSIBM   T DSNDB06  SYSDBASE     46          -1      2
      SYSTABLES           SYSIBM   T DSNDB06  SYSDBASE     55          -1      7
      SYSTABLEPART        SYSIBM   T DSNDB06  SYSDBASE     44          -1      3
      SYSTABAUTH          SYSIBM   T DSNDB06  SYSDBASE     30          -1      0
Command ===>                                                  Scroll ===> PAGE
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA   A                                                                22/015
```

18. To show the table **B** in your TEAM database, move the cursor under Name, left click and type 'B' over the * character

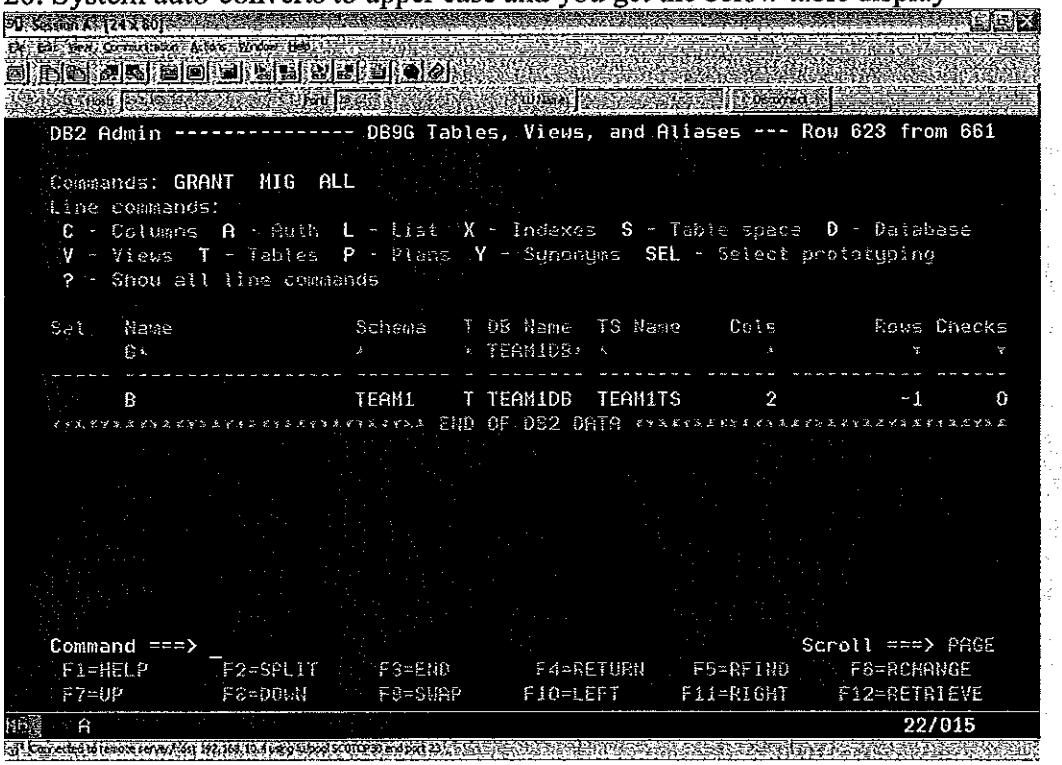


```
DB2 Admin --------------- DB9G Tables, Views, and Aliases - Row 1 to 10 of 661

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Schema   T DB Name  TS Name    Cols        Rows Checks
      B                   ^        ^ ^        ^          ^           v    v
----- ------------------- -------- - -------- -------- ------ ----------- ------
      SYSOBDS             SYSIBM   T DSNDB06  SYSALTER     10          -1      0
      SYSCONTEXTAUTHIDS   SYSIBM   T DSNDB06  SYSCONTX      7          -1      0
      SYSCTXTTRUSTATTRS   SYSIBM   T DSNDB06  SYSCONTX      5          -1      0
      SYSCONTEXT          SYSIBM   T DSNDB06  SYSCONTX     16          -1      0
      SYSCOPY             SYSIBM   T DSNDB06  SYSCOPY      33          -1      0
      SYSFIELDS           SYSIBM   T DSNDB06  SYSDBASE     13          -1      0
      SYSTABLESPACE       SYSIBM   T DSNDB06  SYSDBASE     46          -1      2
      SYSTABLES           SYSIBM   T DSNDB06  SYSDBASE     55          -1      7
      SYSTABLEPART        SYSIBM   T DSNDB06  SYSDBASE     44          -1      3
      SYSTABAUTH          SYSIBM   T DSNDB06  SYSDBASE     30          -1      0
Command ===>                                                  Scroll ===> PAGE
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA   A                                                                10/038
```

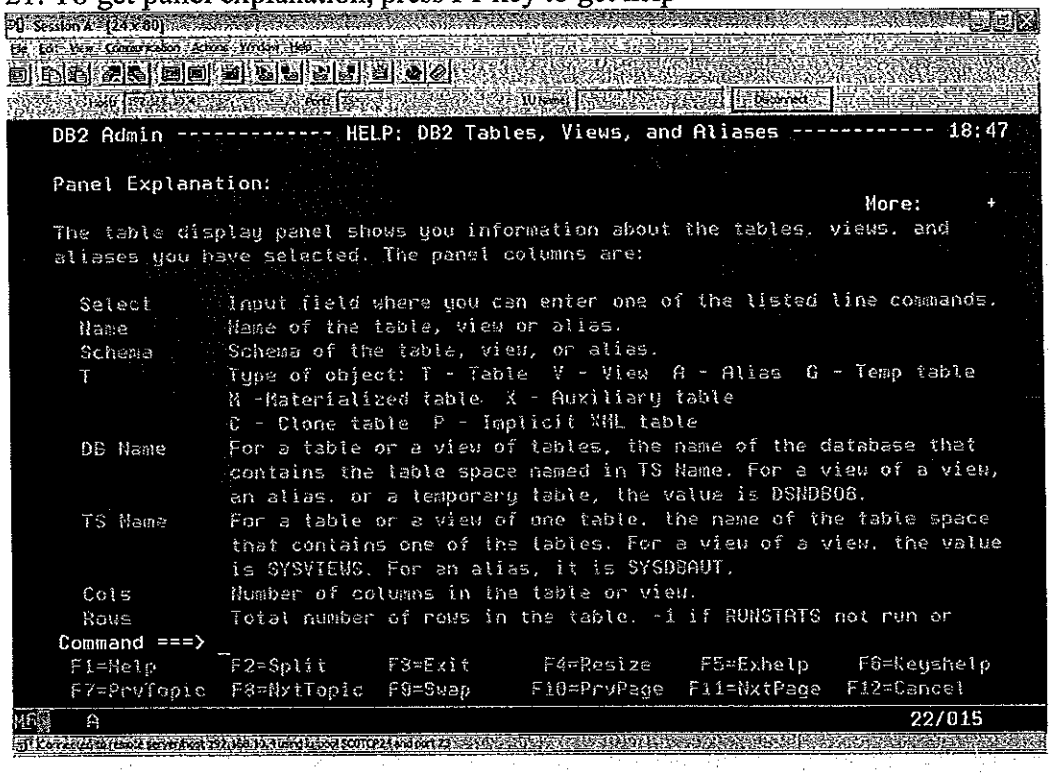
19. Press Tab key 3 times, the cursor under DB Name, type 'TEAMxDB' where TEAMx is your team ID and press Enter key



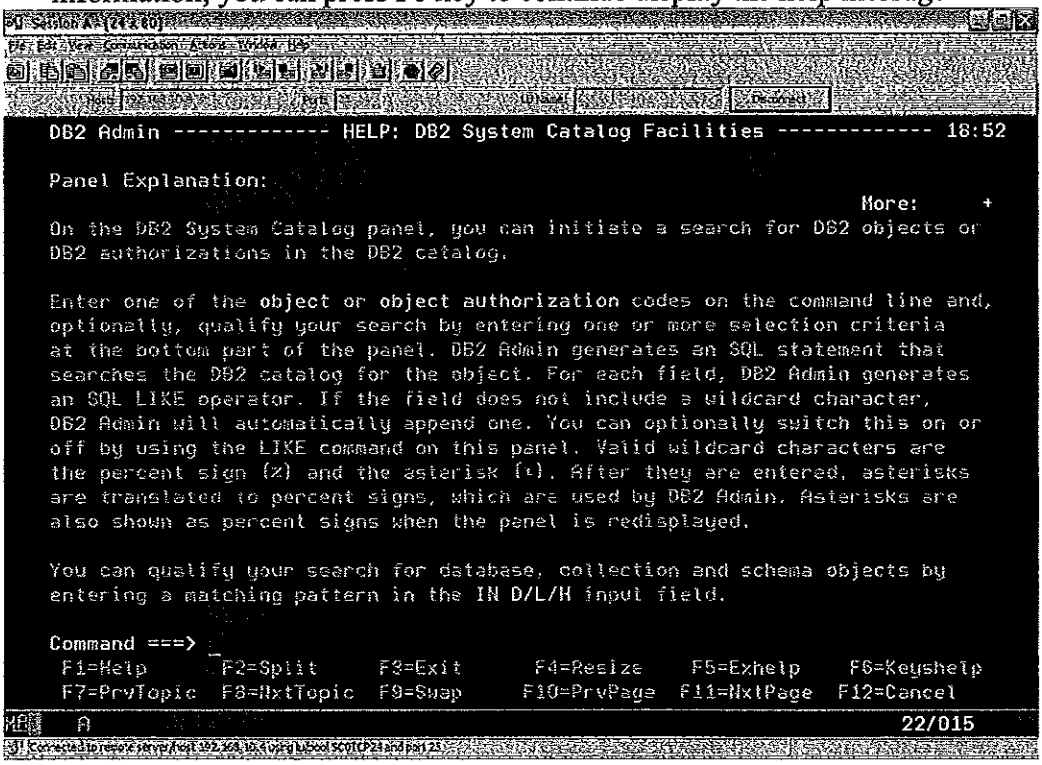
20. System auto-converts to upper case and you get the below table display



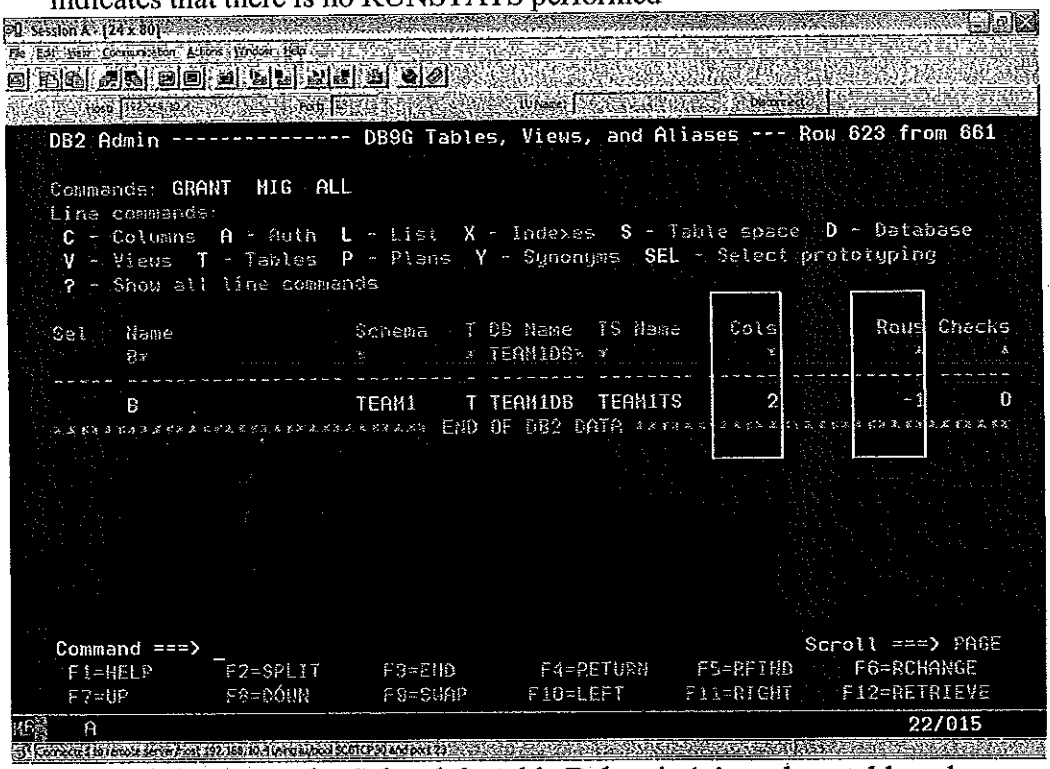
21. To get panel explanation, press F1 key to get help



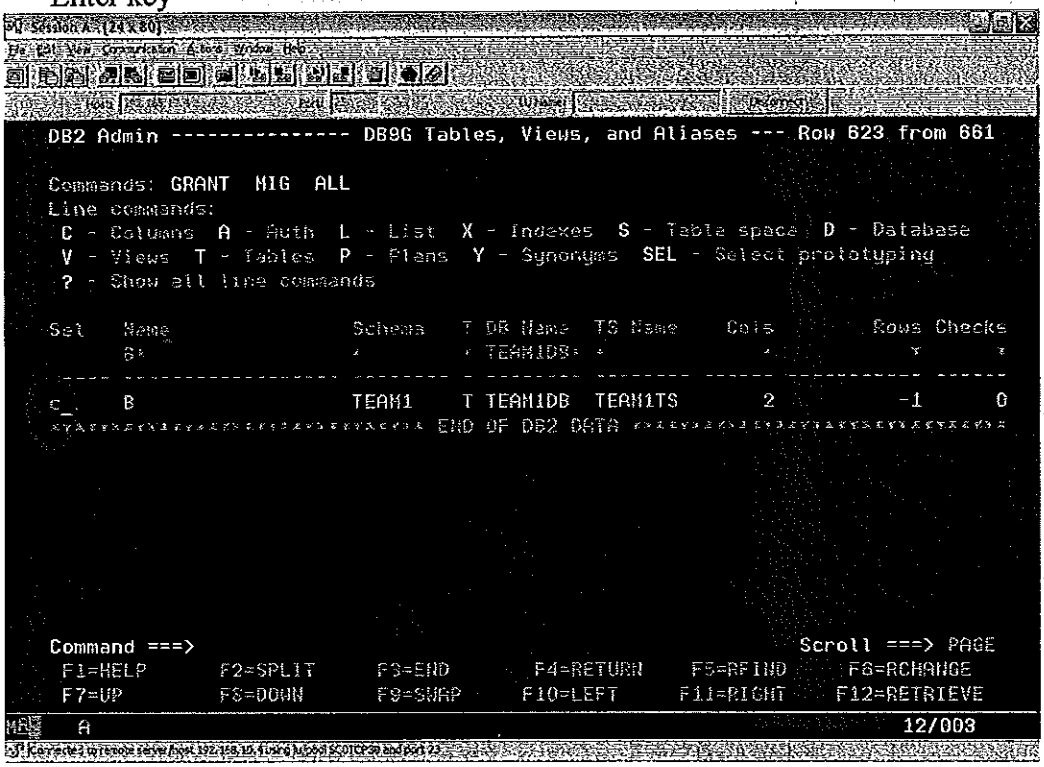
22. When 'More +' appears at upper right hand corner to indicate that it has more information, you can press F8 key to continue display the help message



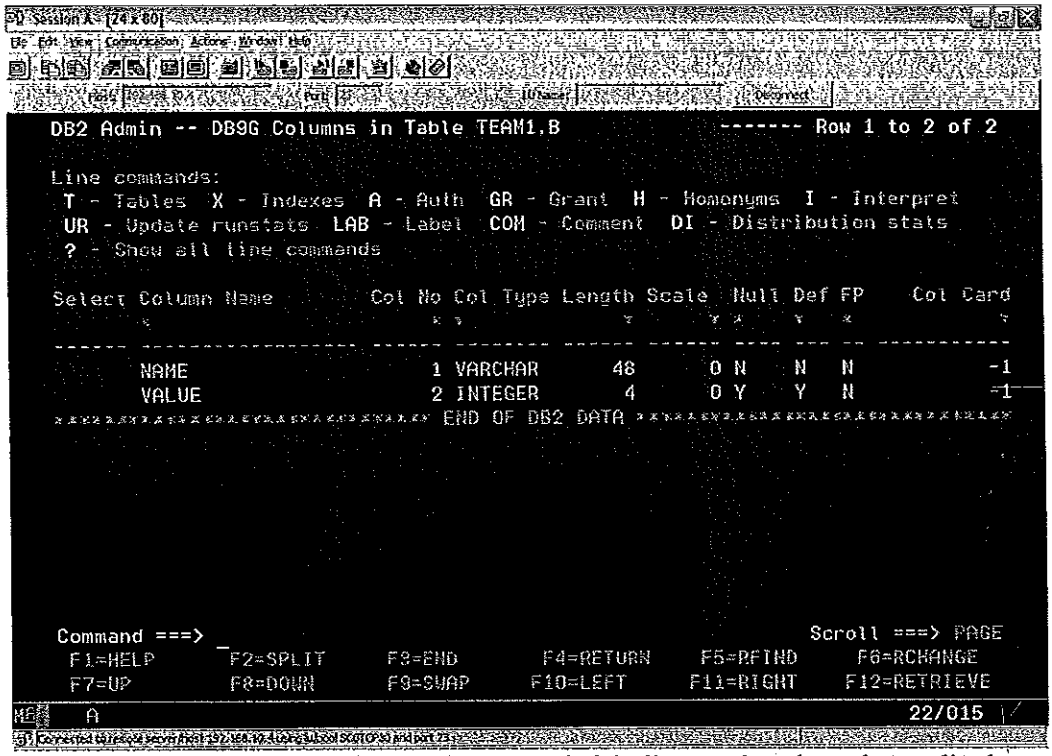
23. Press F3 key to return the calling panel and this table has 2 columns and 'ROWS -1" indicates that there is no RUNSTATS performed



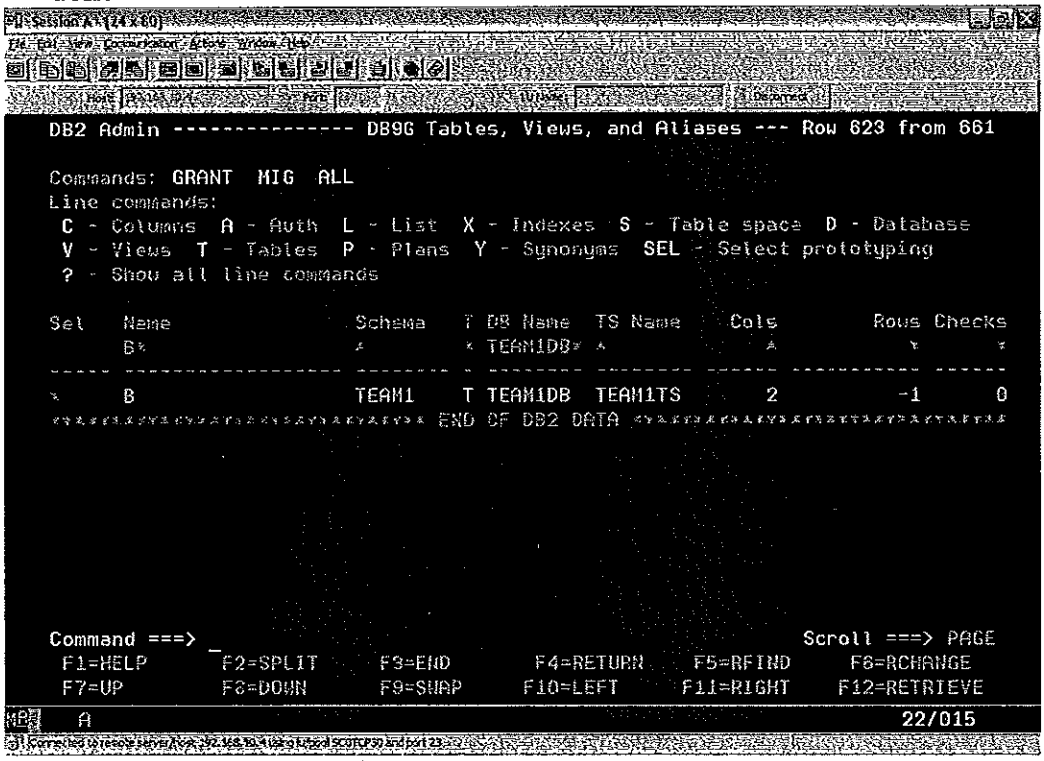
24. Move the cursor under Sel and the table **B**, key-in 'c' to show table column and press Enter key



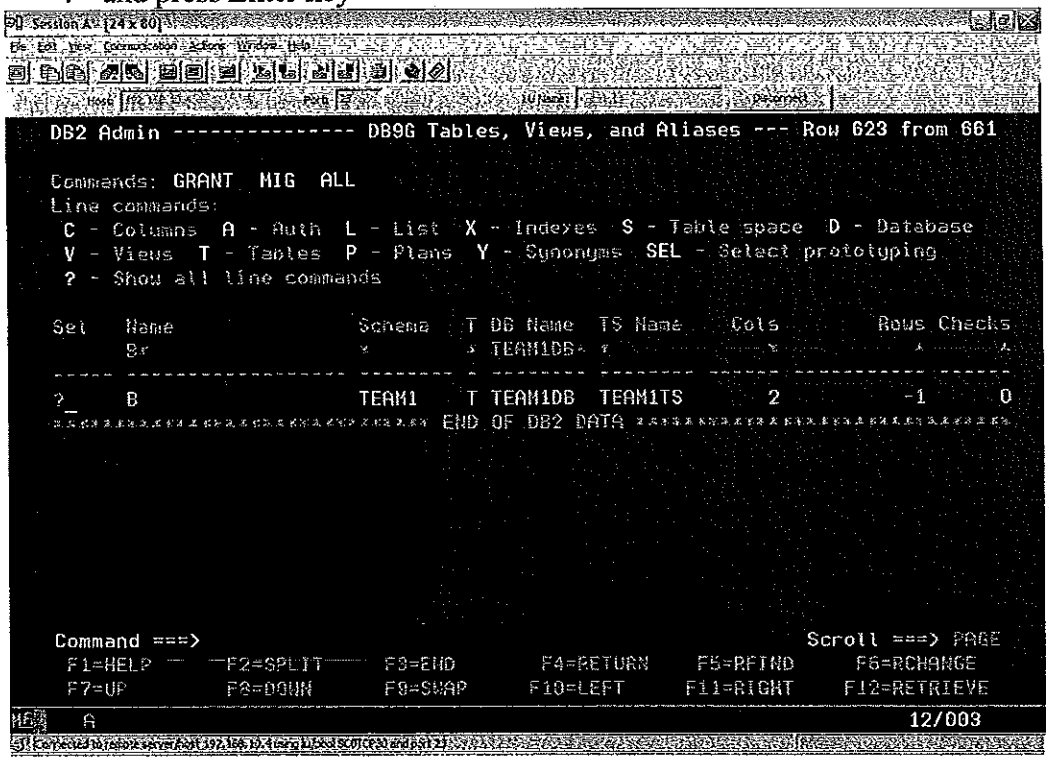
25. It shows columns information:



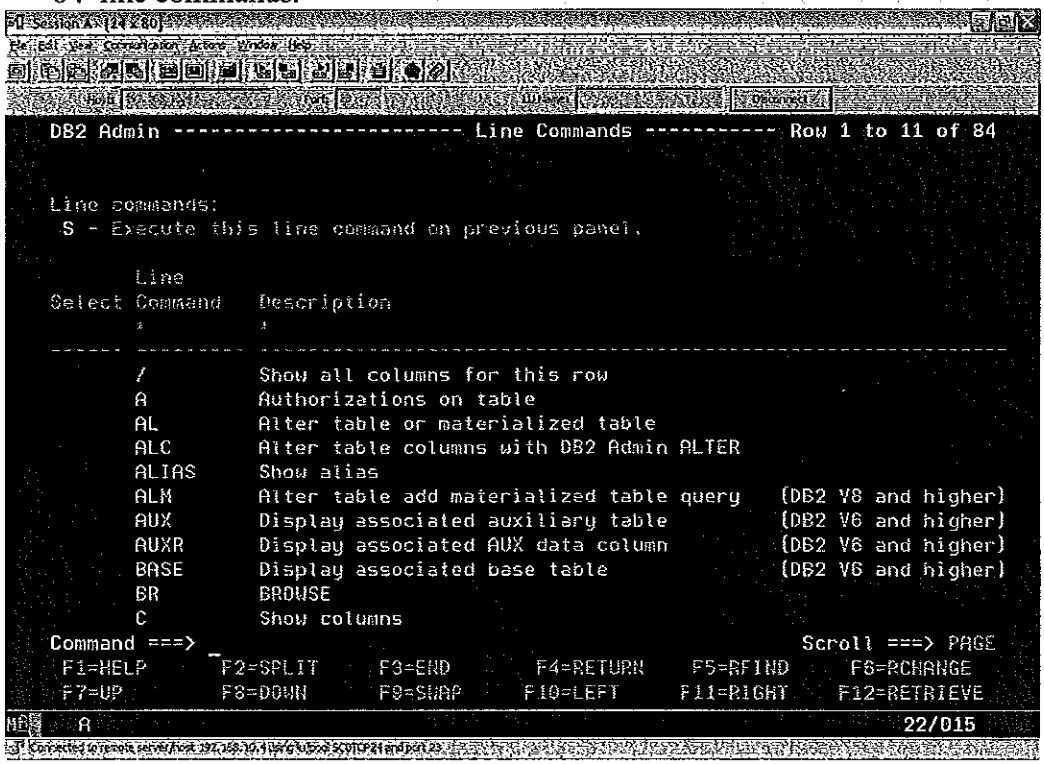
26. Press F3 key to return the previous panel, * indicates that the column display action is done



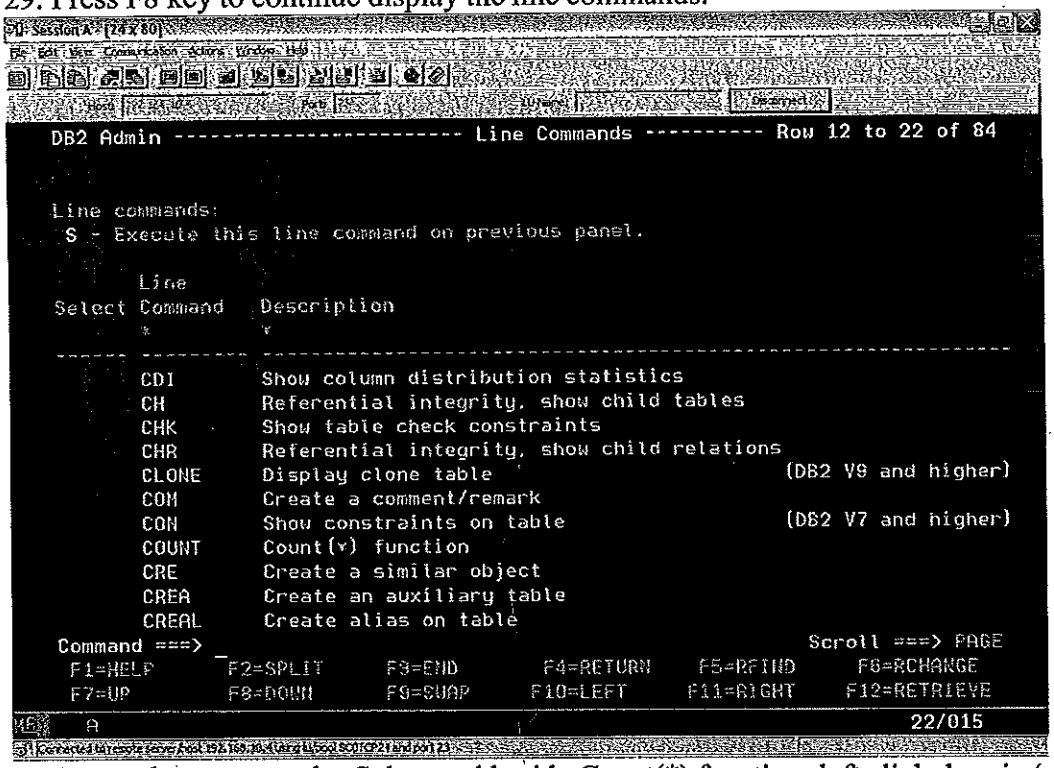
27. To get help of using line command(s), move the cursor under Sel and the table, key-in '?' and press Enter key



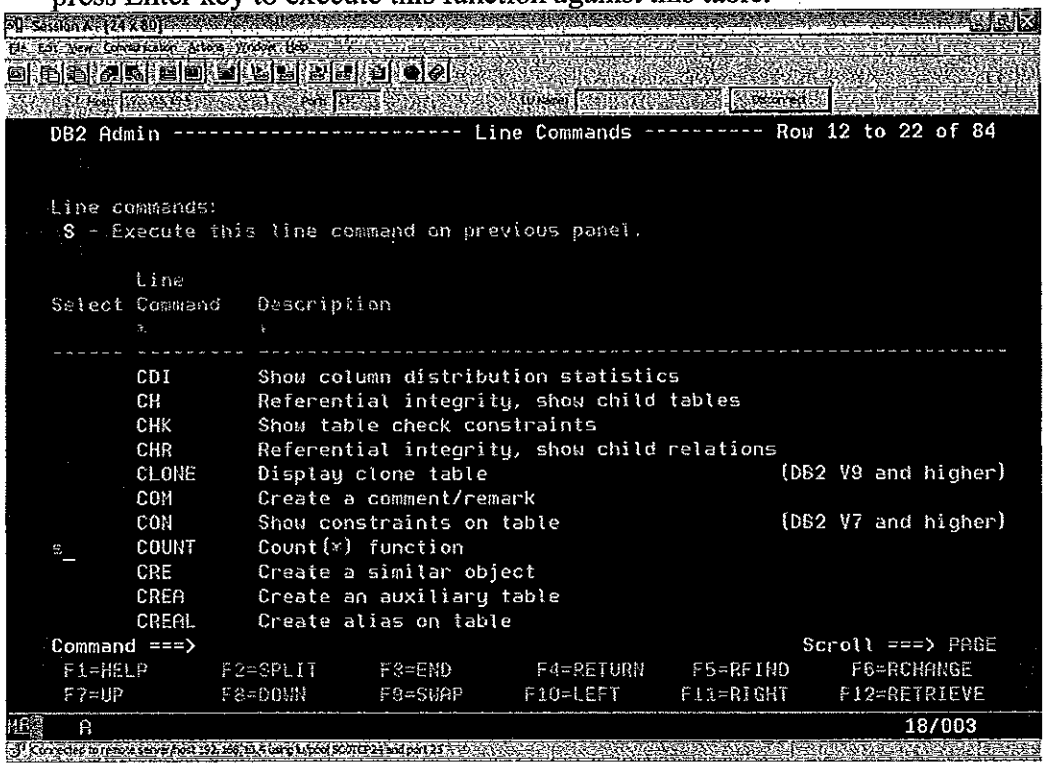
28. It shows the line commands and descriptions. The right upper corner shows there are 84 line commands.



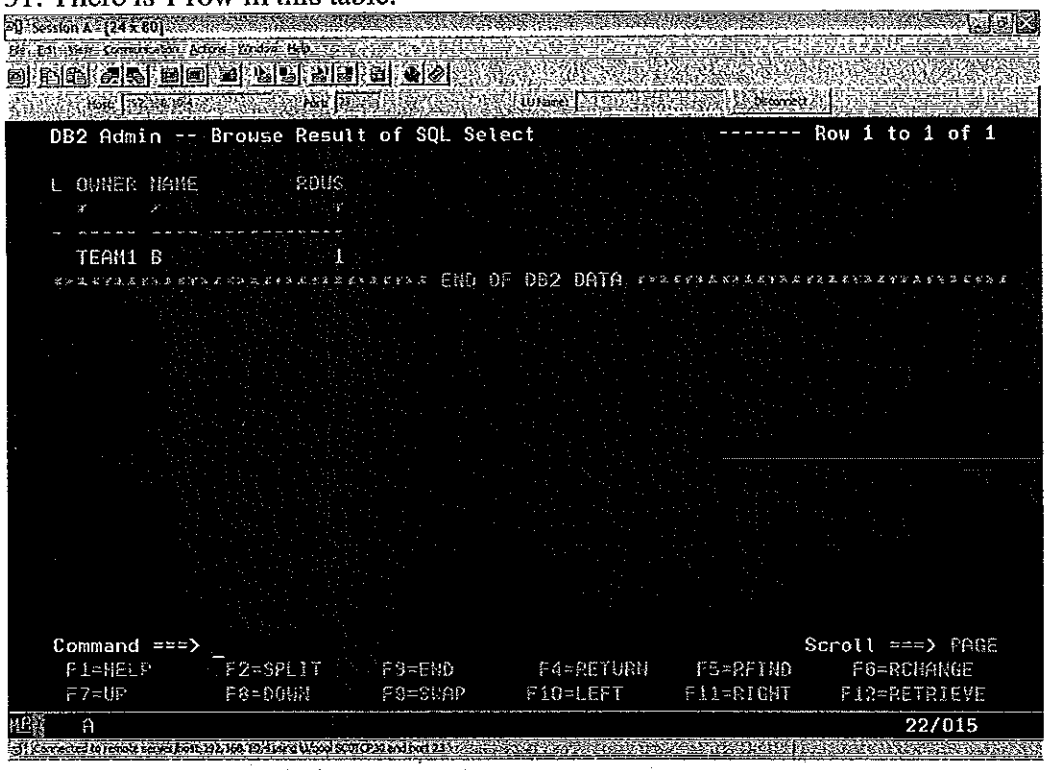
29. Press F8 key to continue display the line commands:



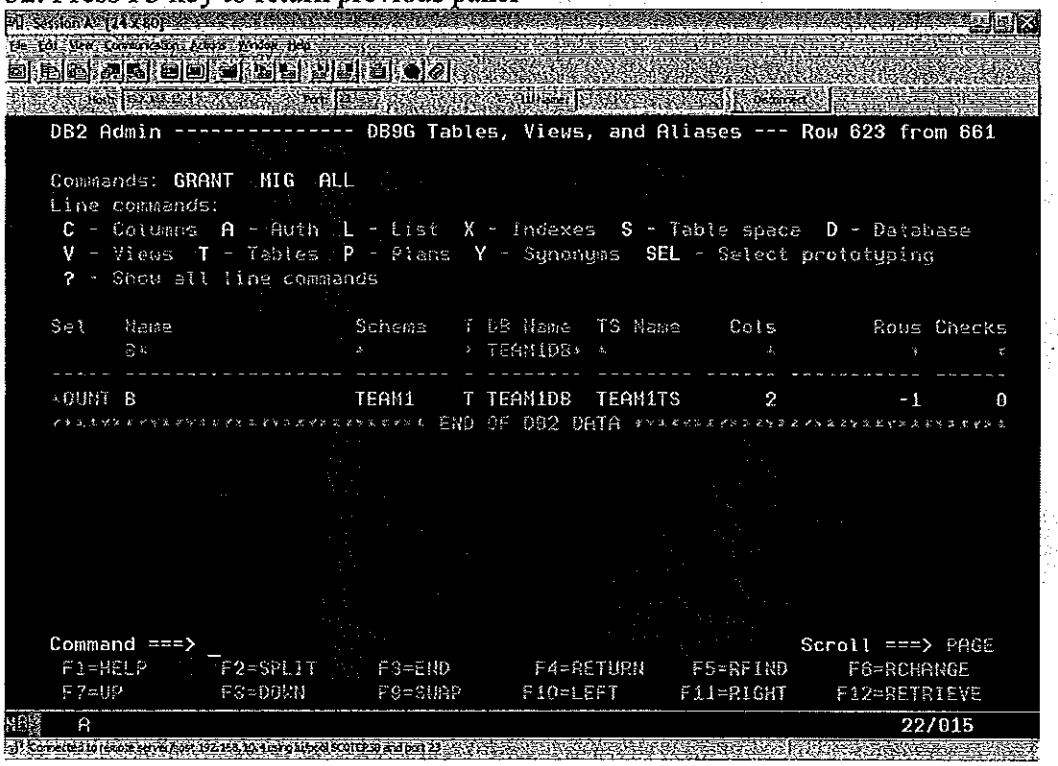
30. Move the cursor under Select and beside Count(*) function, left click, key-in 's' and press Enter key to execute this function against this table:



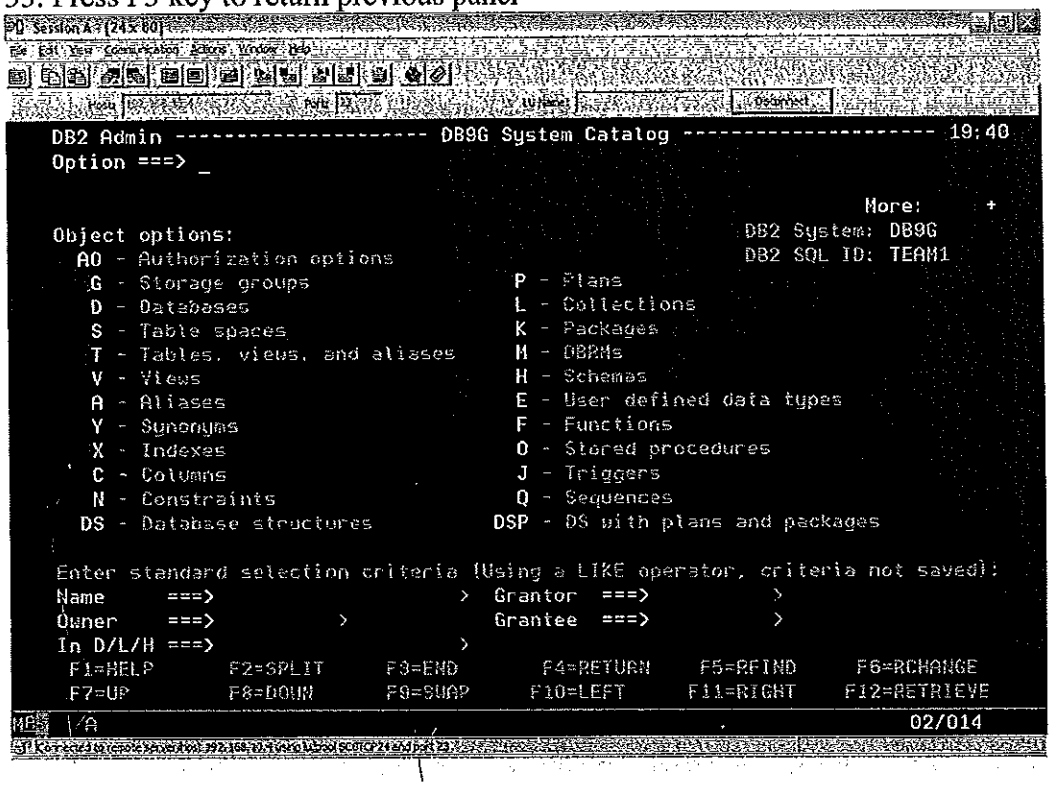
31. There is 1 row in this table:



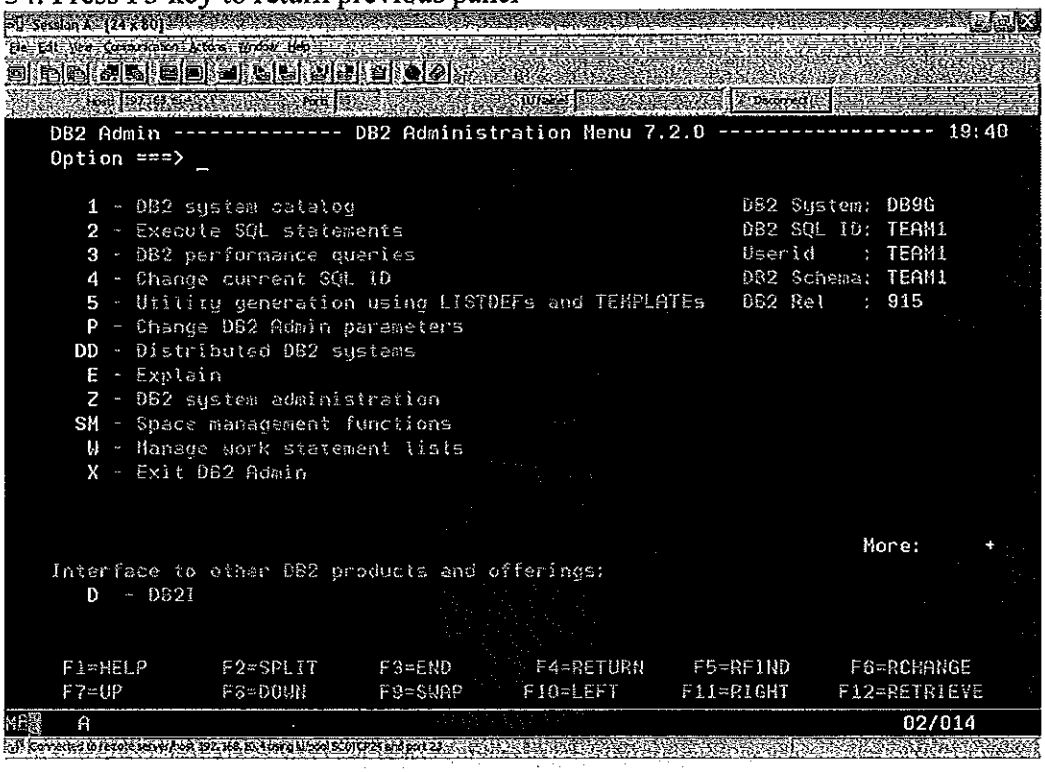
32. Press F3 key to return previous panel



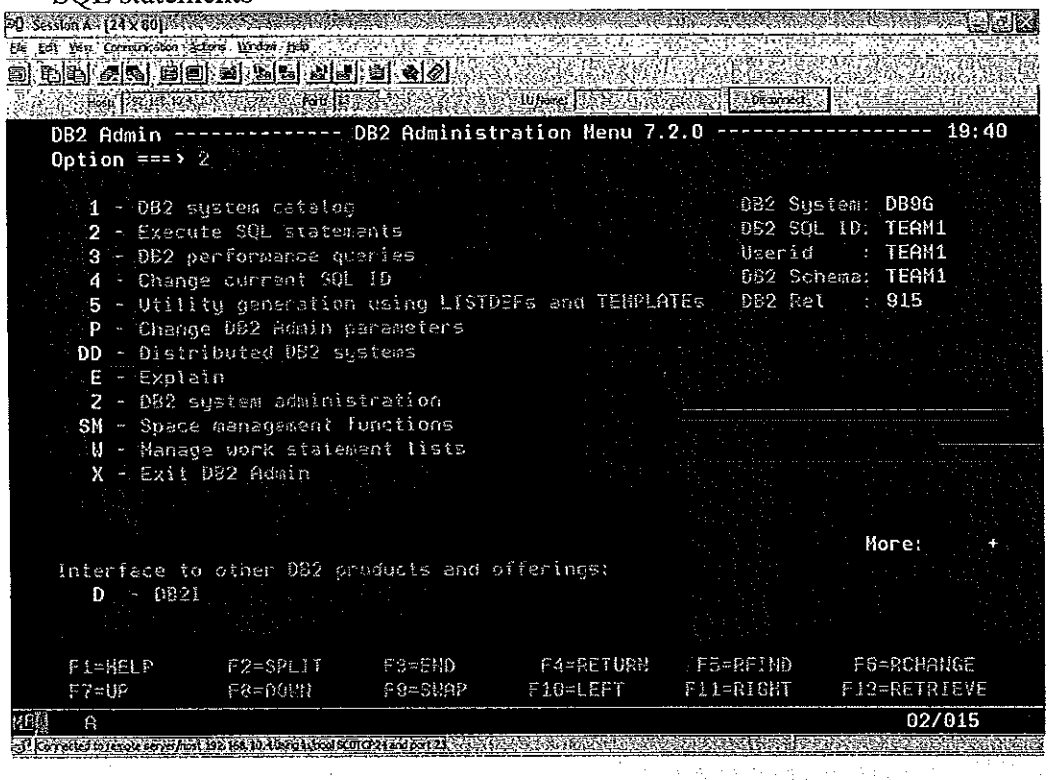
33. Press F3 key to return previous panel



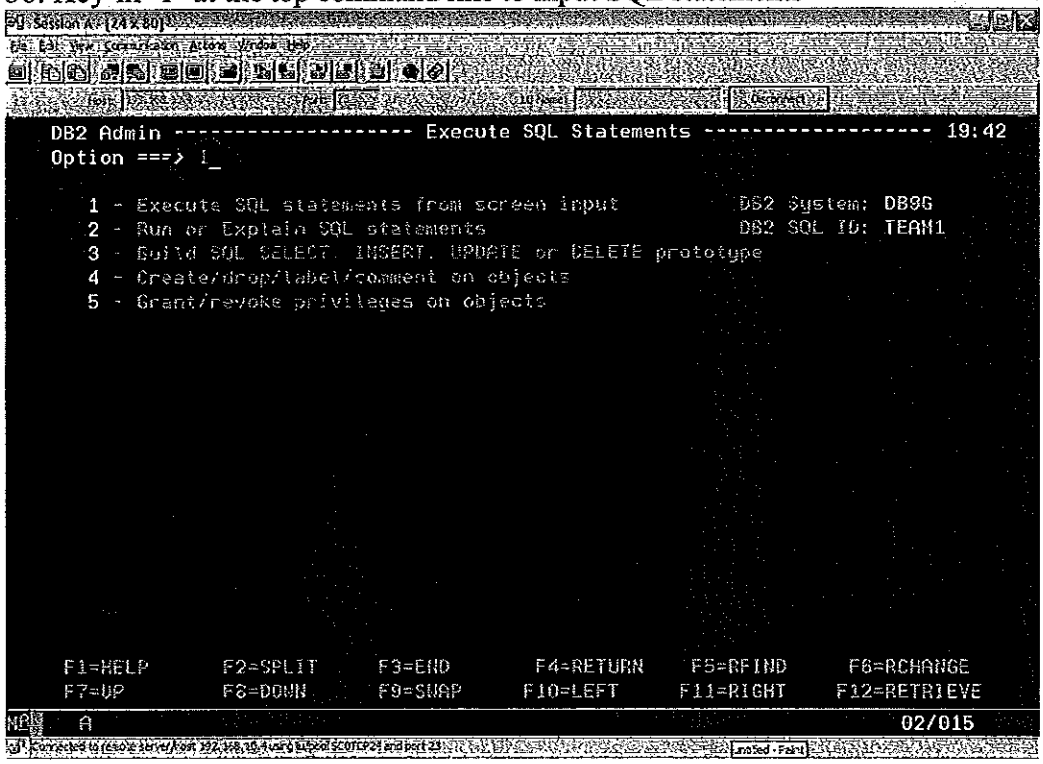
34. Press F3 key to return previous panel



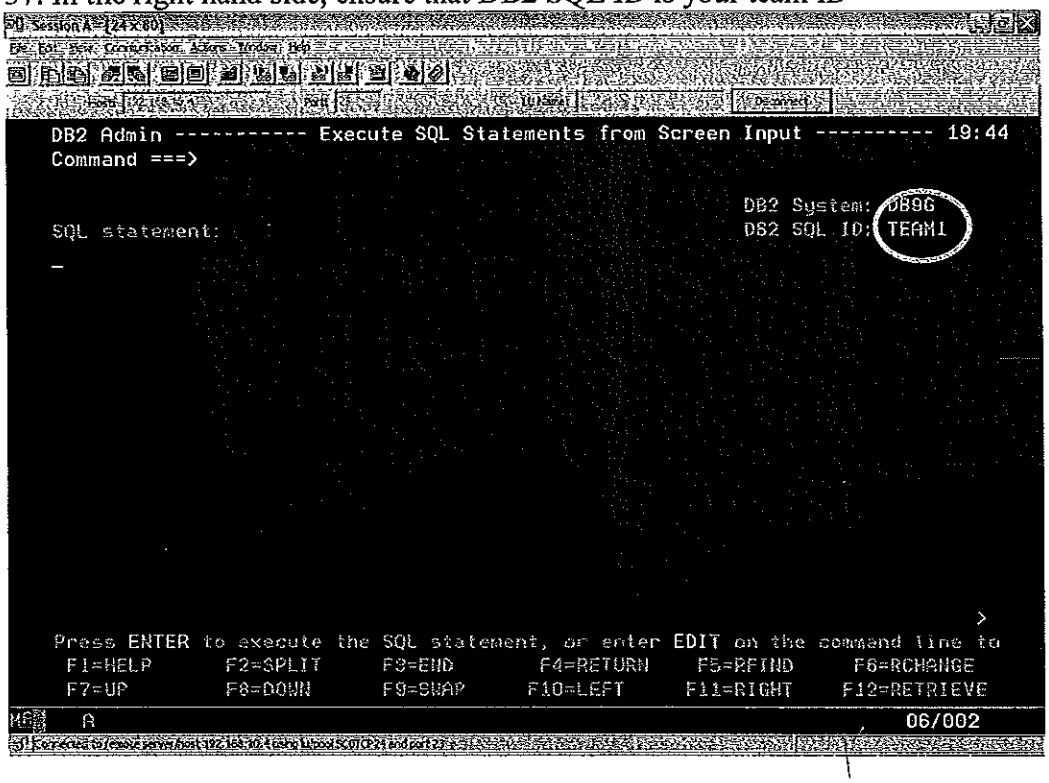
35. Key-in '2' at top command line and press Enter key to provide a panel to execute SQL statements



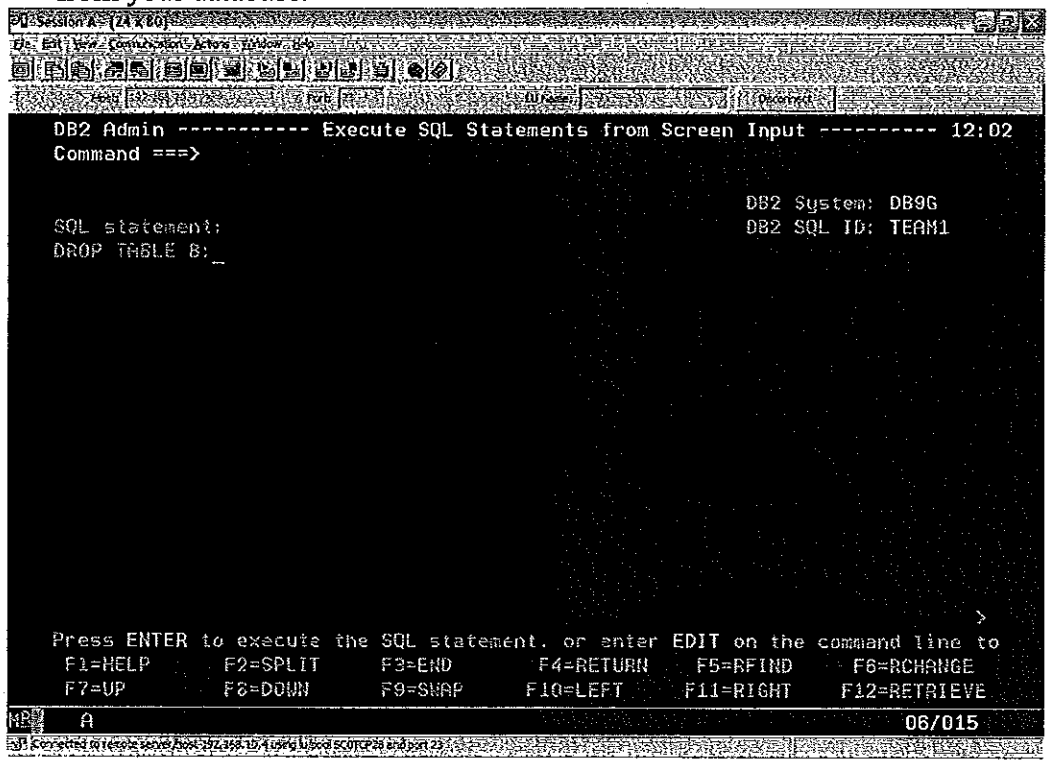
36. Key-in '1' at the top command line to input SQL statements



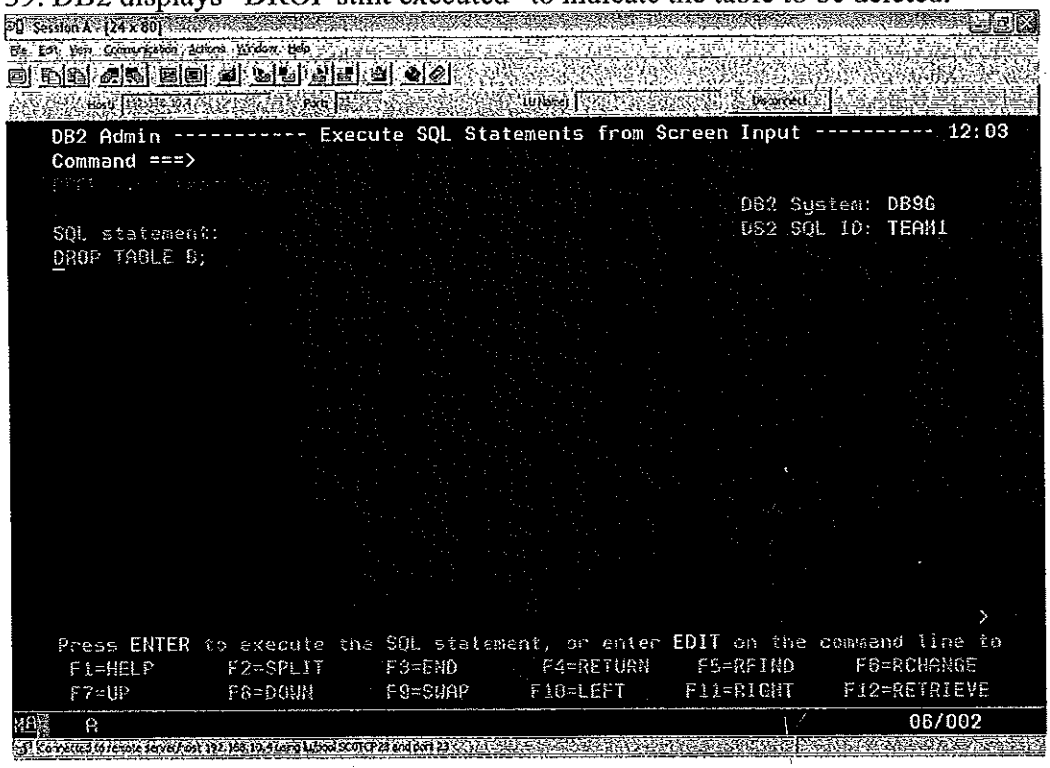
37. In the right hand side, ensure that DB2 SQL ID is your team ID



38. Under SQL statement, key-in **DROP TABLE B;** and press Enter key to delete table from your database:



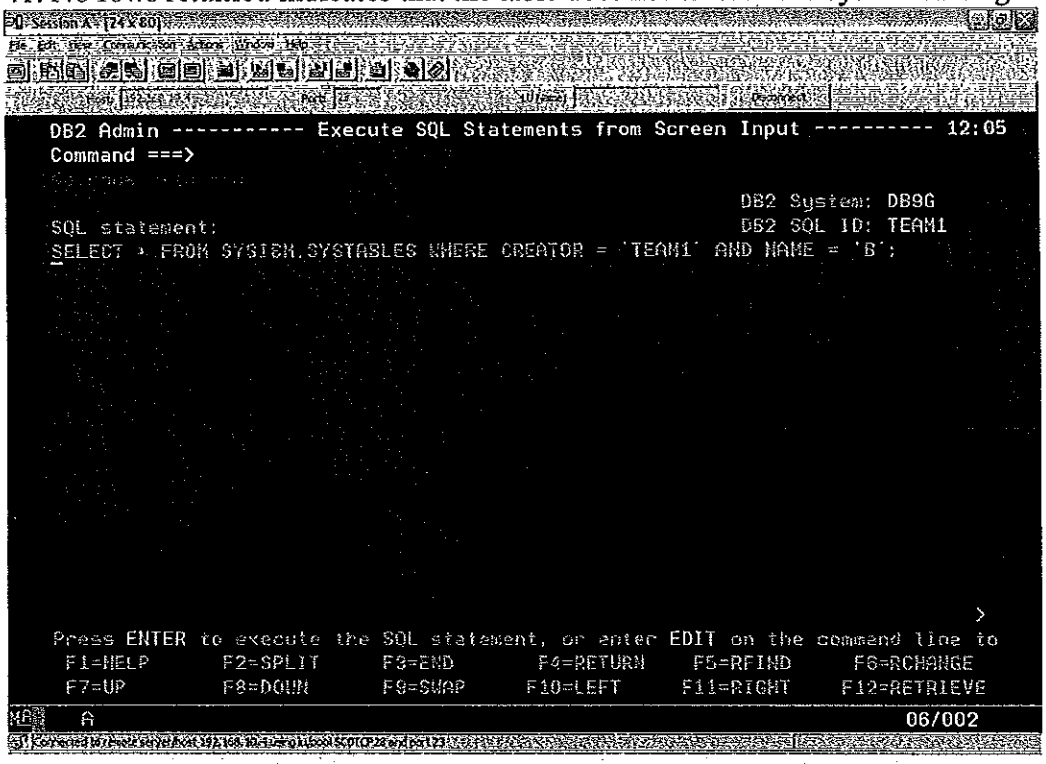
39. DB2 displays "DROP stmt executed" to indicate the table to be deleted.



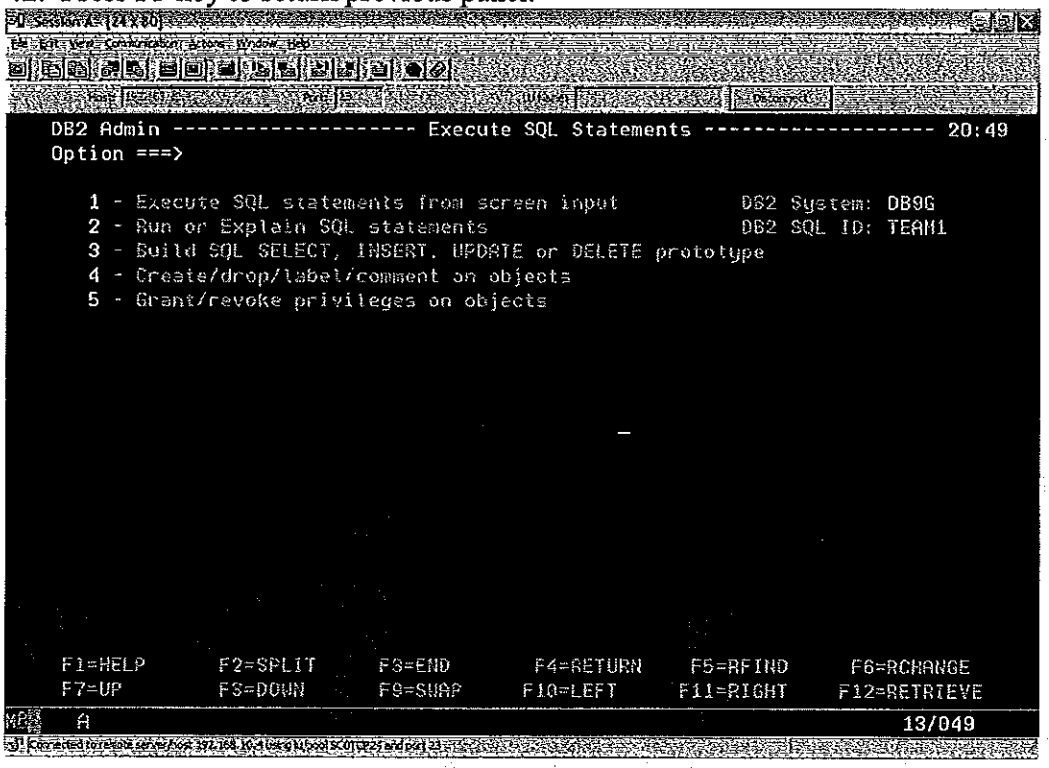
40. To verify table to be deleted, key-in the following statement with replacing TEAMx to your team ID and press Enter key

```
SELECT * FROM SYSIBM.SYSTABLES WHERE CREATOR =
'TEAMx' and NAME = 'B';
```

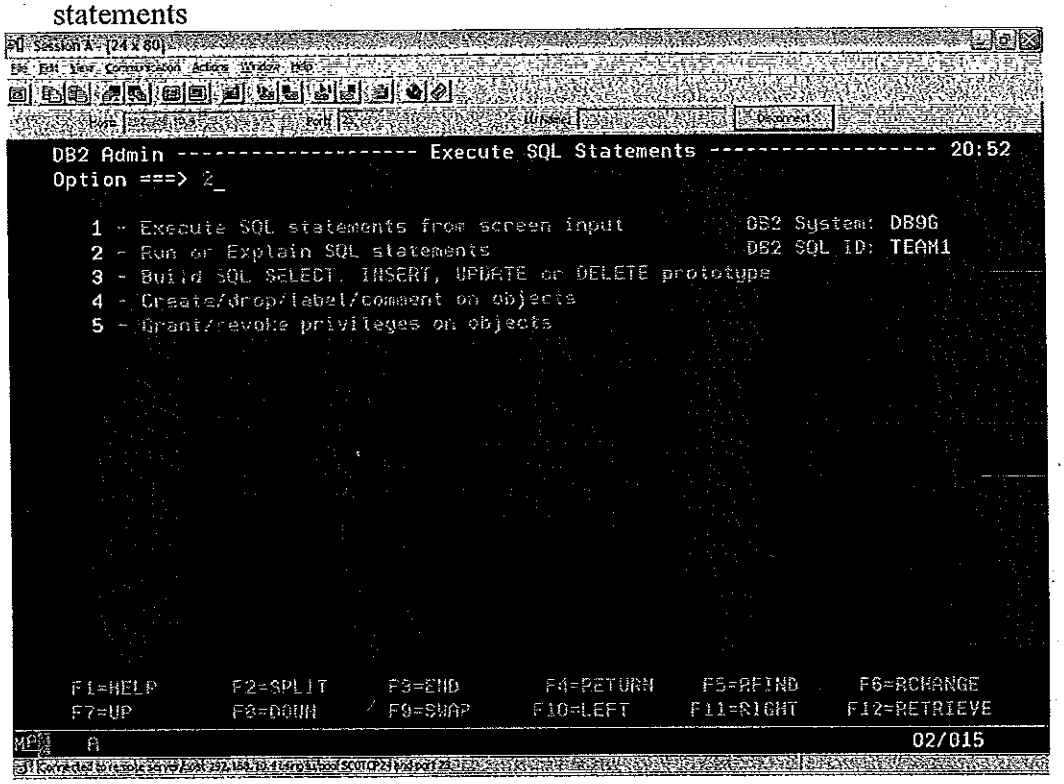
41. No rows returned indicates that the table does not exist in the system catalog



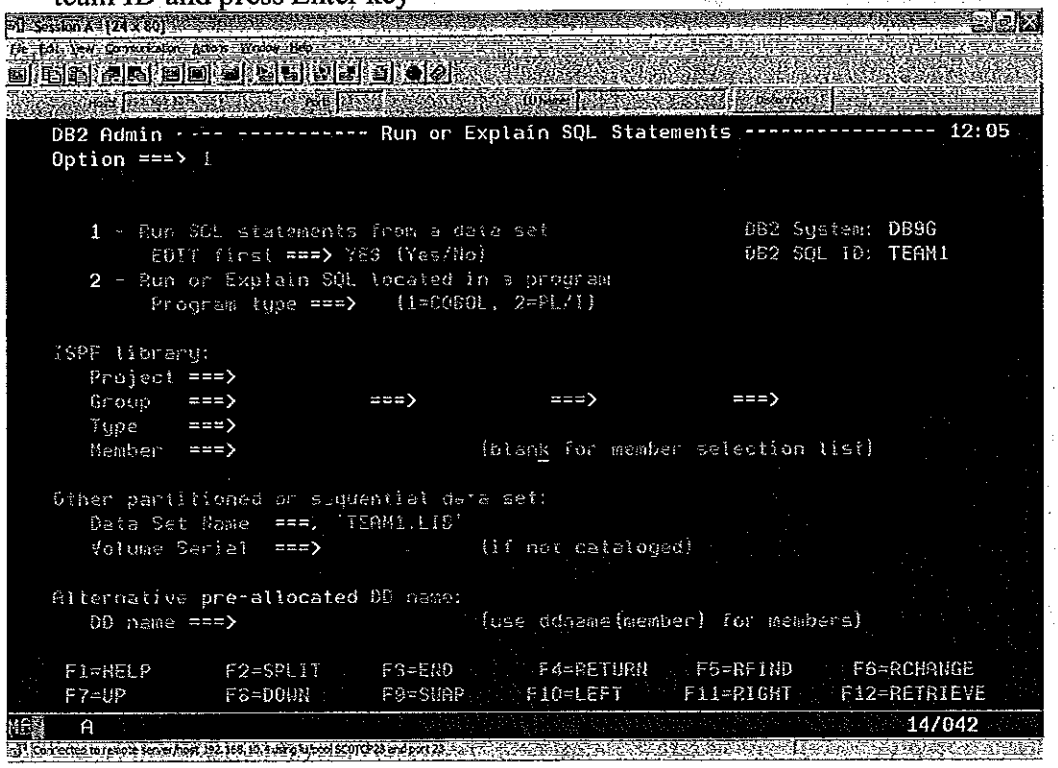
42. Press F3 key to return previous panel:



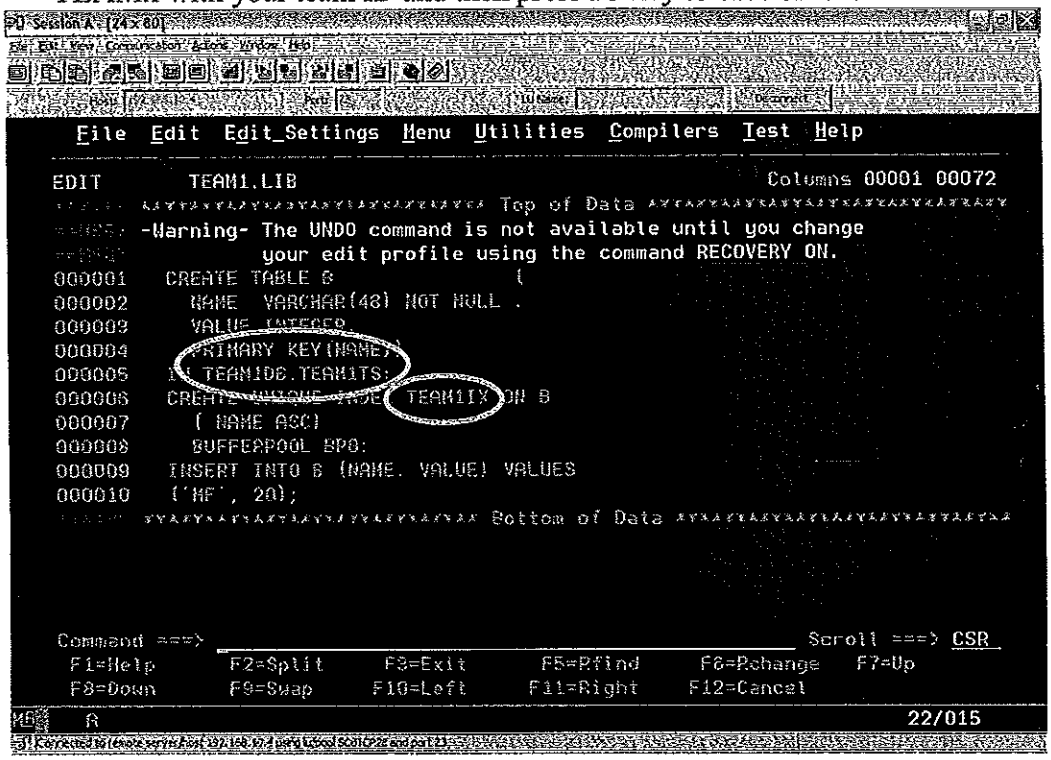
43. Key-in '2' at the top command line and press Enter key to RUN saved SQL statements



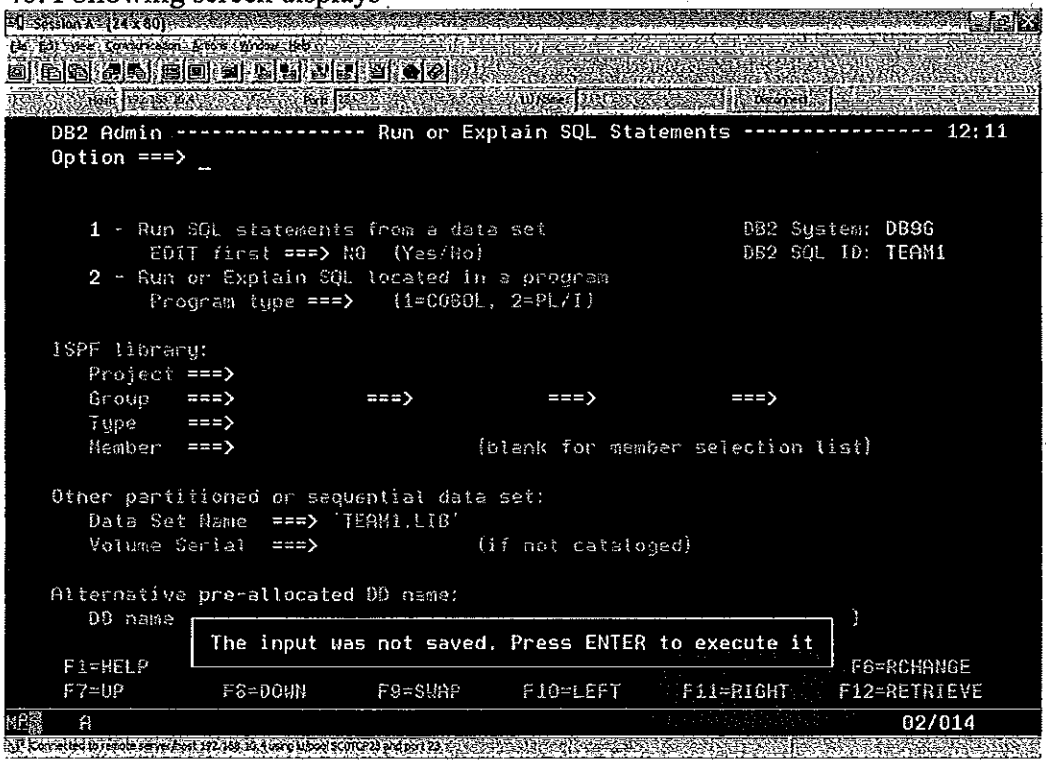
44. Key-in '1' at the top command line, specify 'TEAMx.LIB' where TEAMx is your team ID and press Enter key



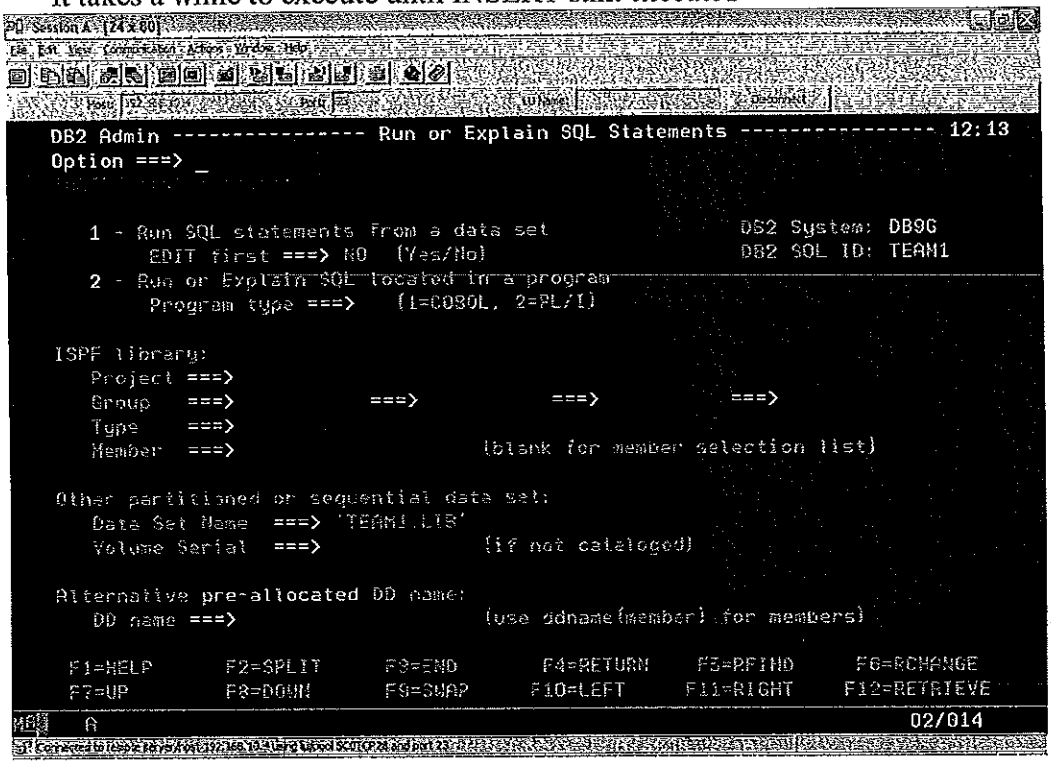
45. Ensure TEAMxDB.TEAMxTS and TEAMxIX are your team ID. If not, replace TEAMx with your team ID and then press F3 key to save or exit



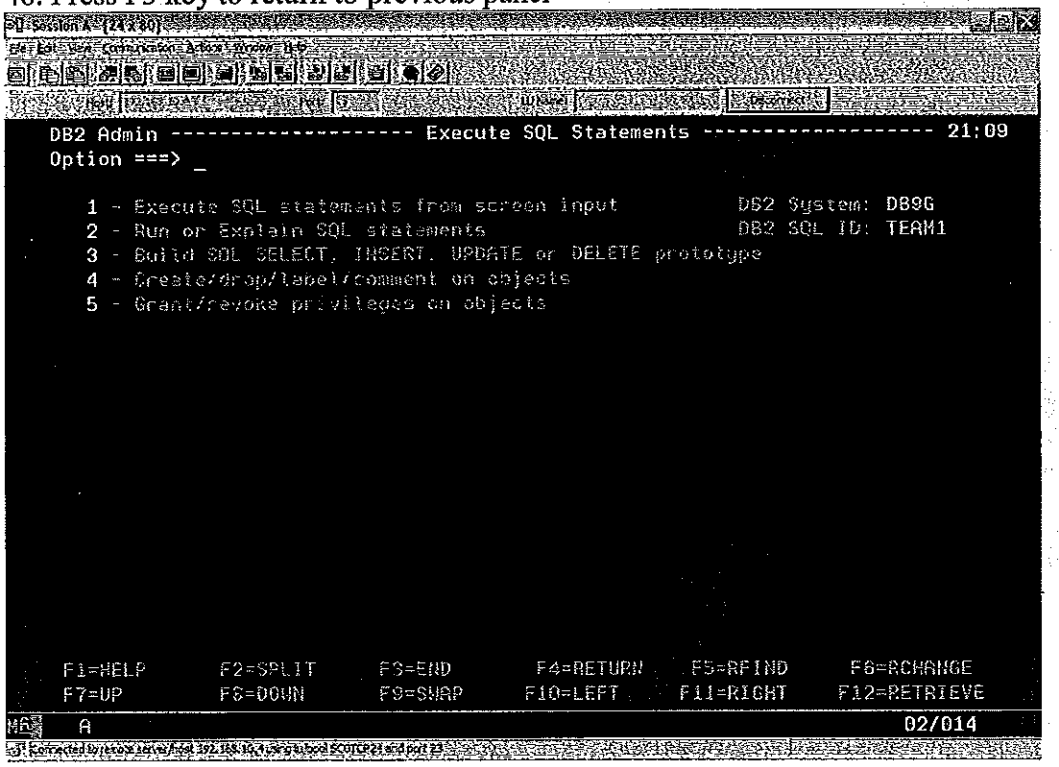
46. Following screen displays



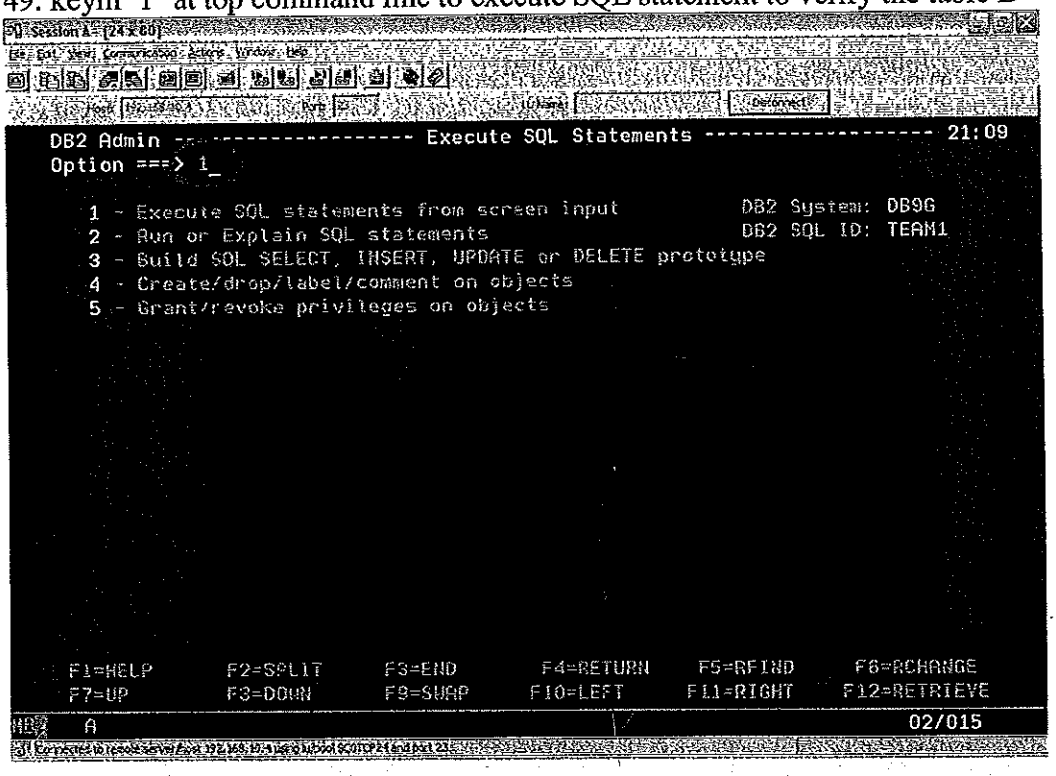
47. Press Enter key to execute, it creates table B, index and insert one record into table B.
It takes a while to execute until INSERT stmt executed



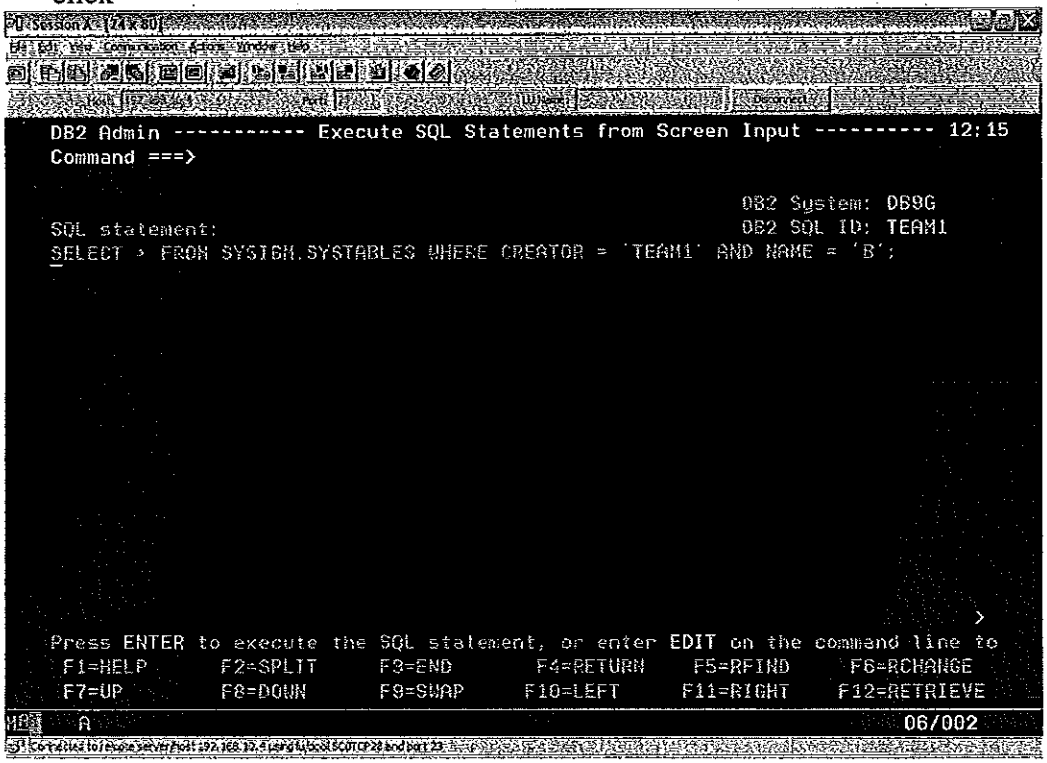
48. Press F3 key to return to previous panel

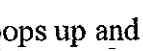


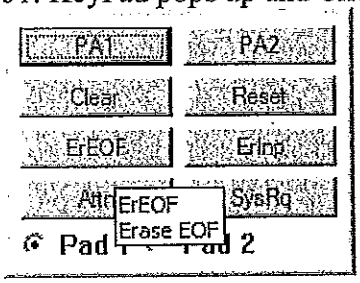
49. keyin '1' at top command line to execute SQL statement to verify the table **B**



50. To clear the full statement left over, move the cursor under the **SELECT** and right click



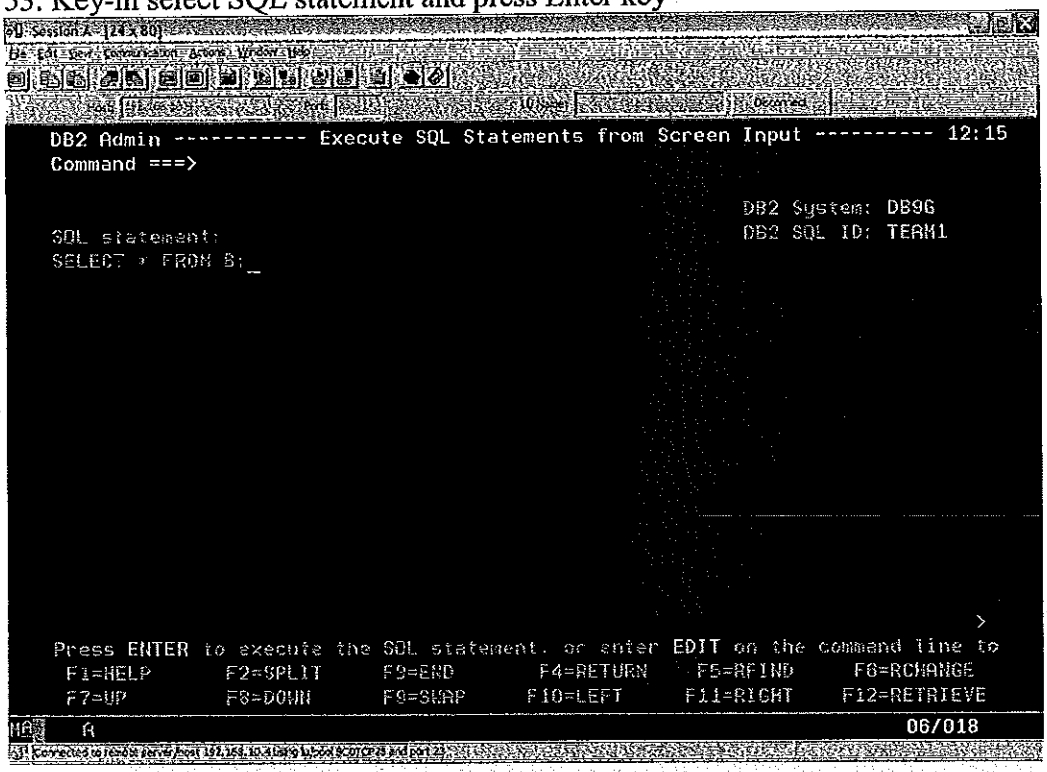
51. KeyPad pops up and Click '  ' to erase to EOF



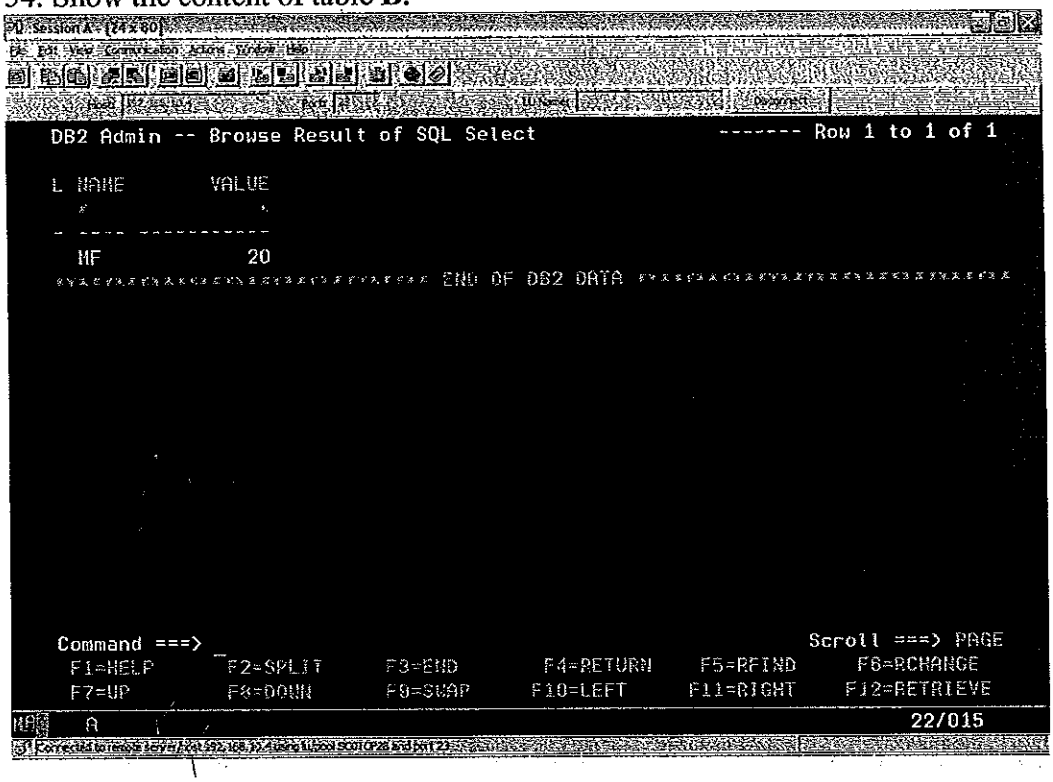
52. Use below SQL statement to view the table content

```
SELECT * FROM B;
```

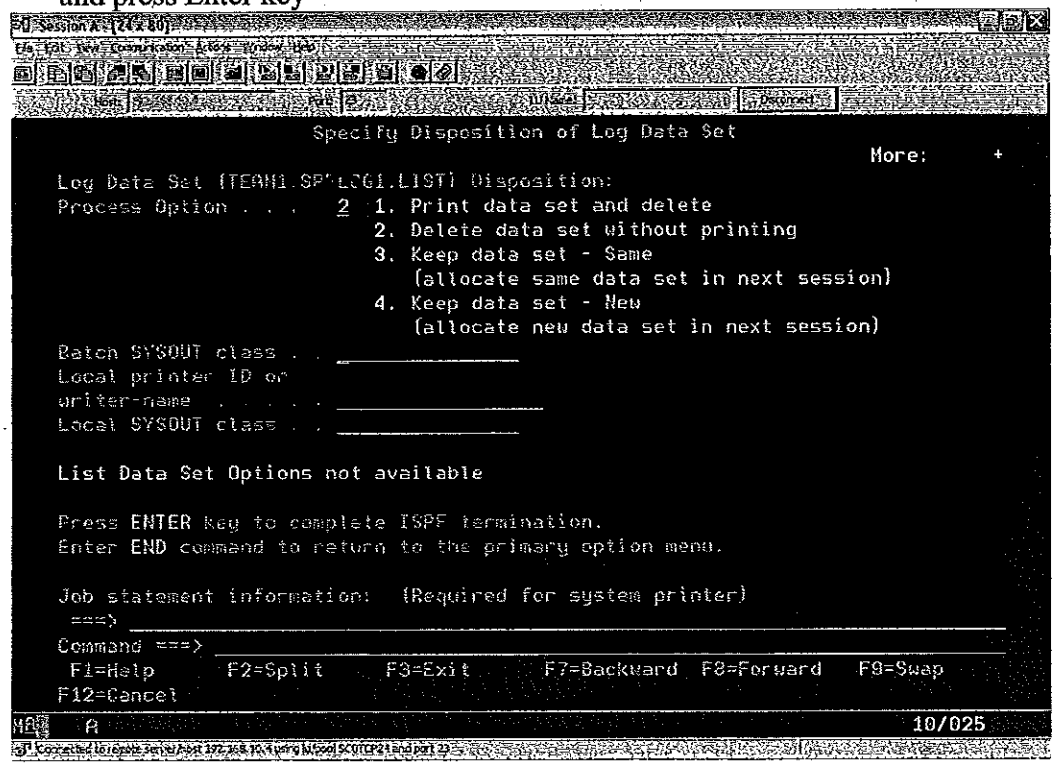
53. Key-in select SQL statement and press Enter key



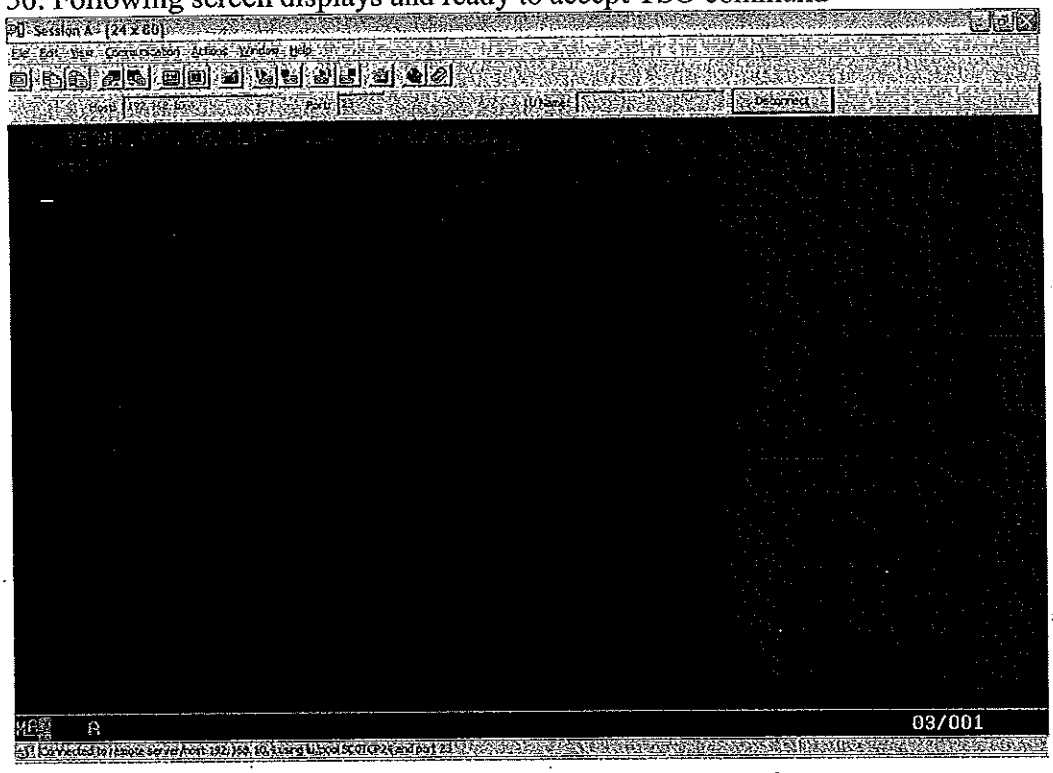
54. Show the content of table **B:**



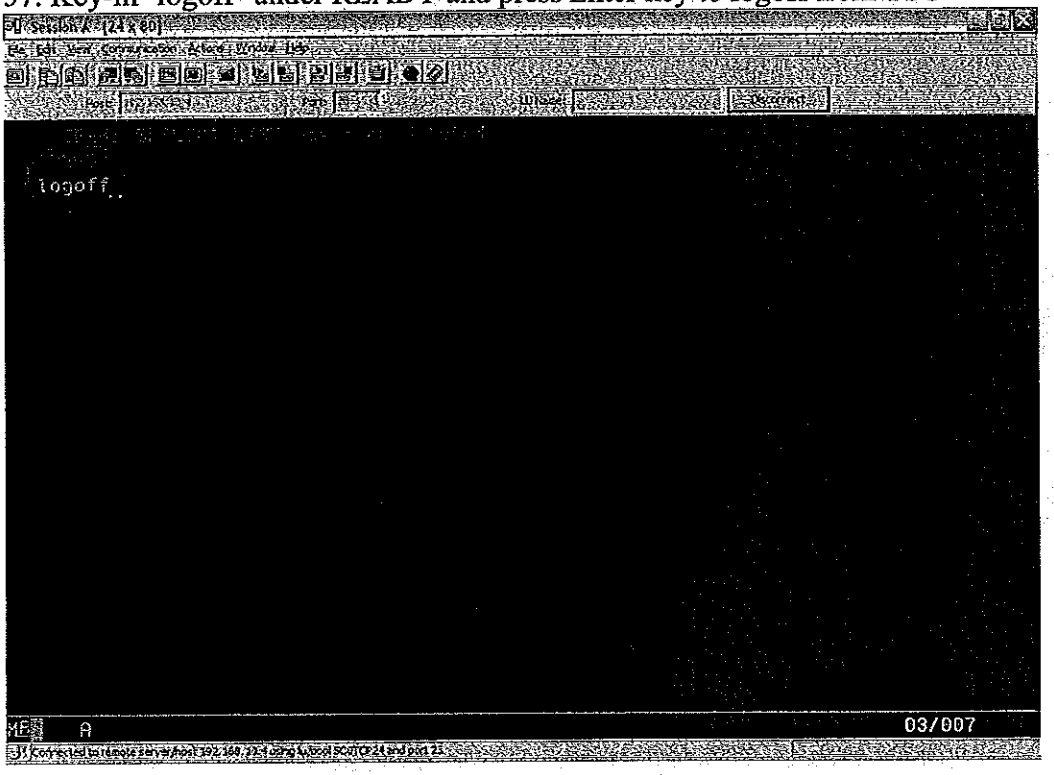
55. To exit TSO, repeat press F3 key until the following screen appears, then key-in '2' and press Enter key



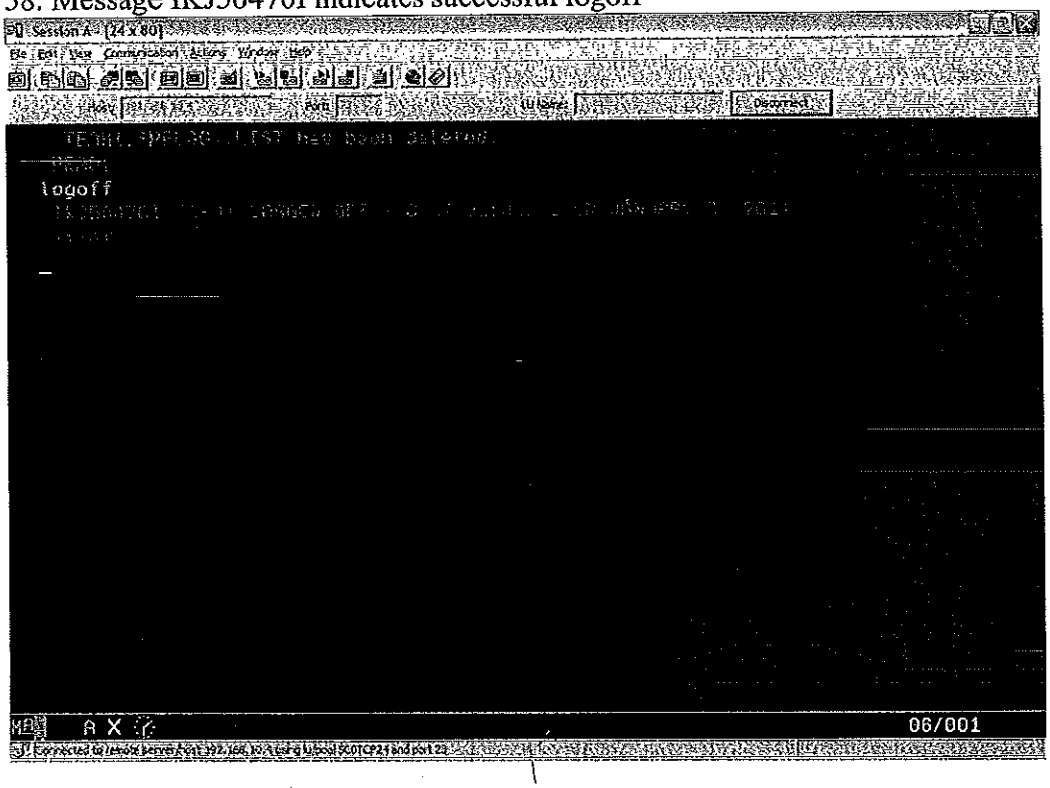
56. Following screen displays and ready to accept TSO command



57. Key-in 'logoff' under READY and press Enter key to logoff from TSO

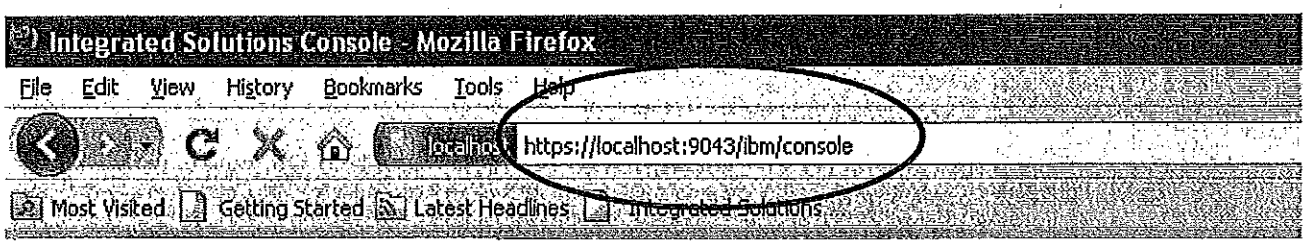


58. Message IKJ56470I indicates successful logoff

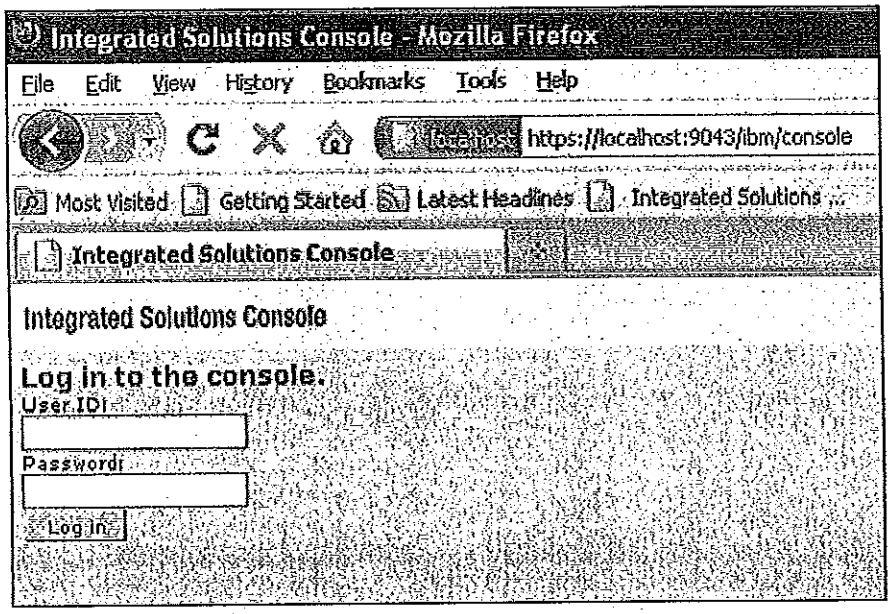


59. Click Mozilla Firefox to start web browser from Desktop

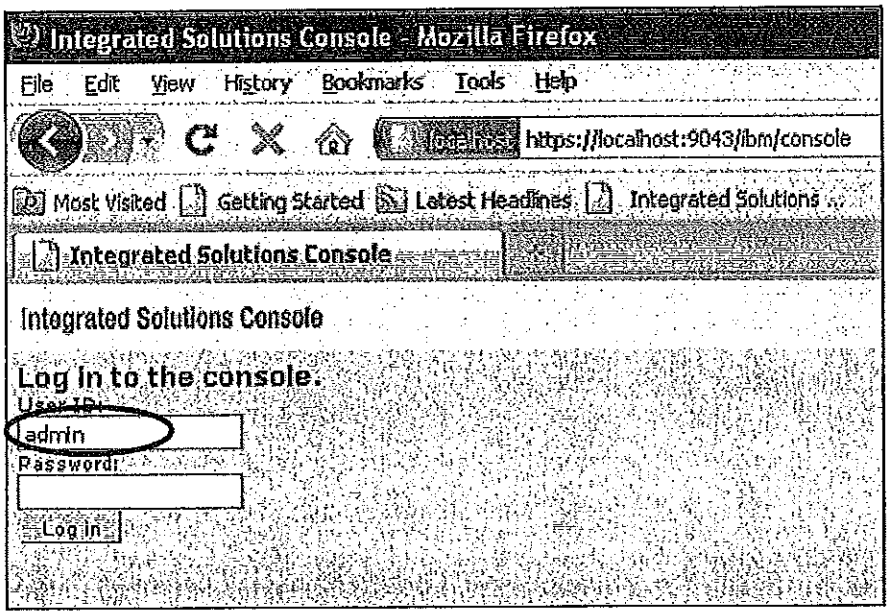
60. Enter https://localhost:9043/ibm/console and press Enter to WebSphere Administration Console to do configuration.



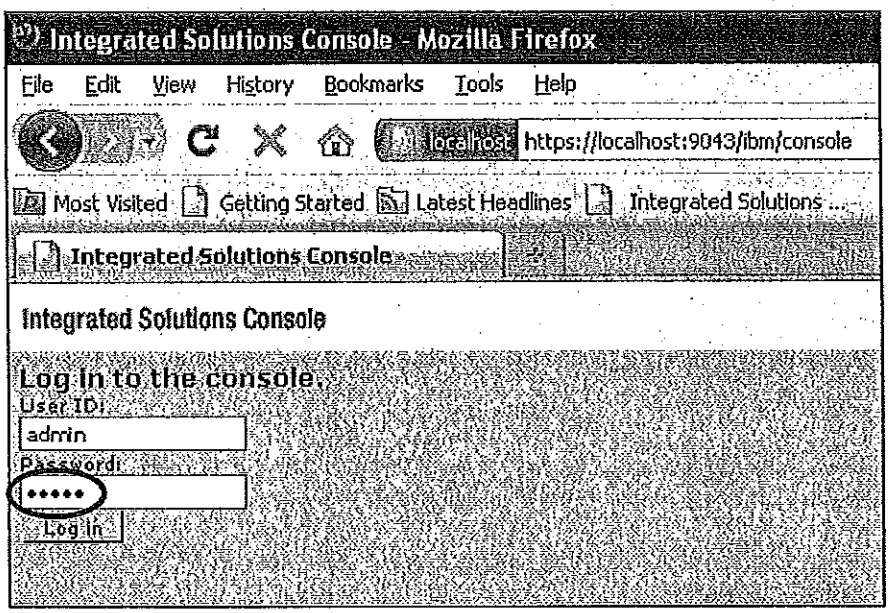
61. Following screen displays and ask for User ID: and Password to logon Administration Console:



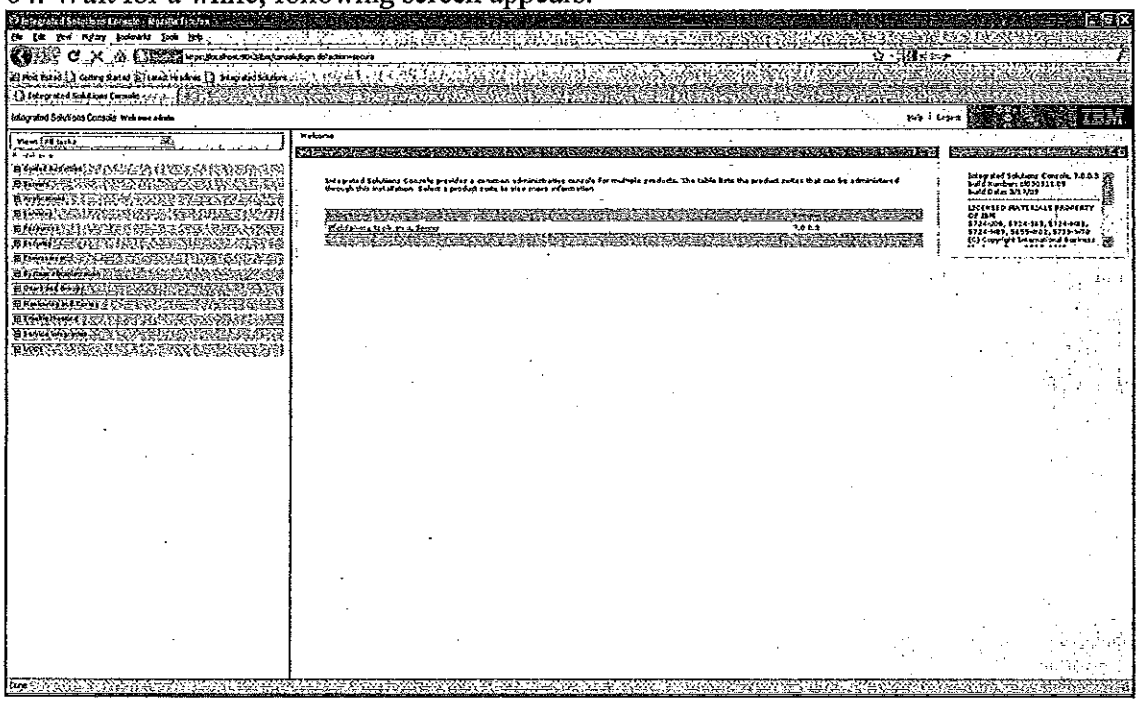
62. Enter admin for User ID:



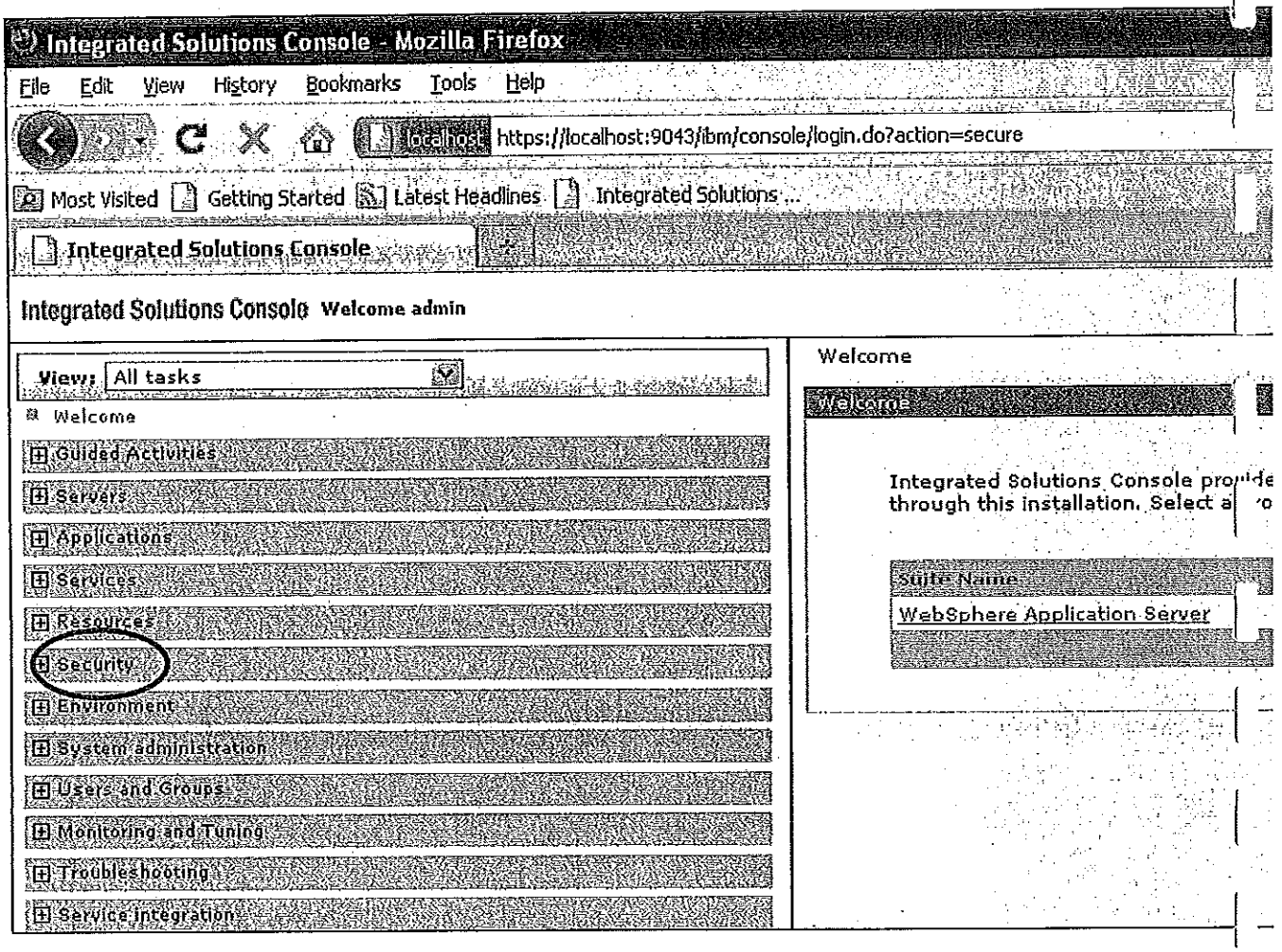
63. Enter admin for Password: and Click "Log in" to logon WebSphere Administration Console



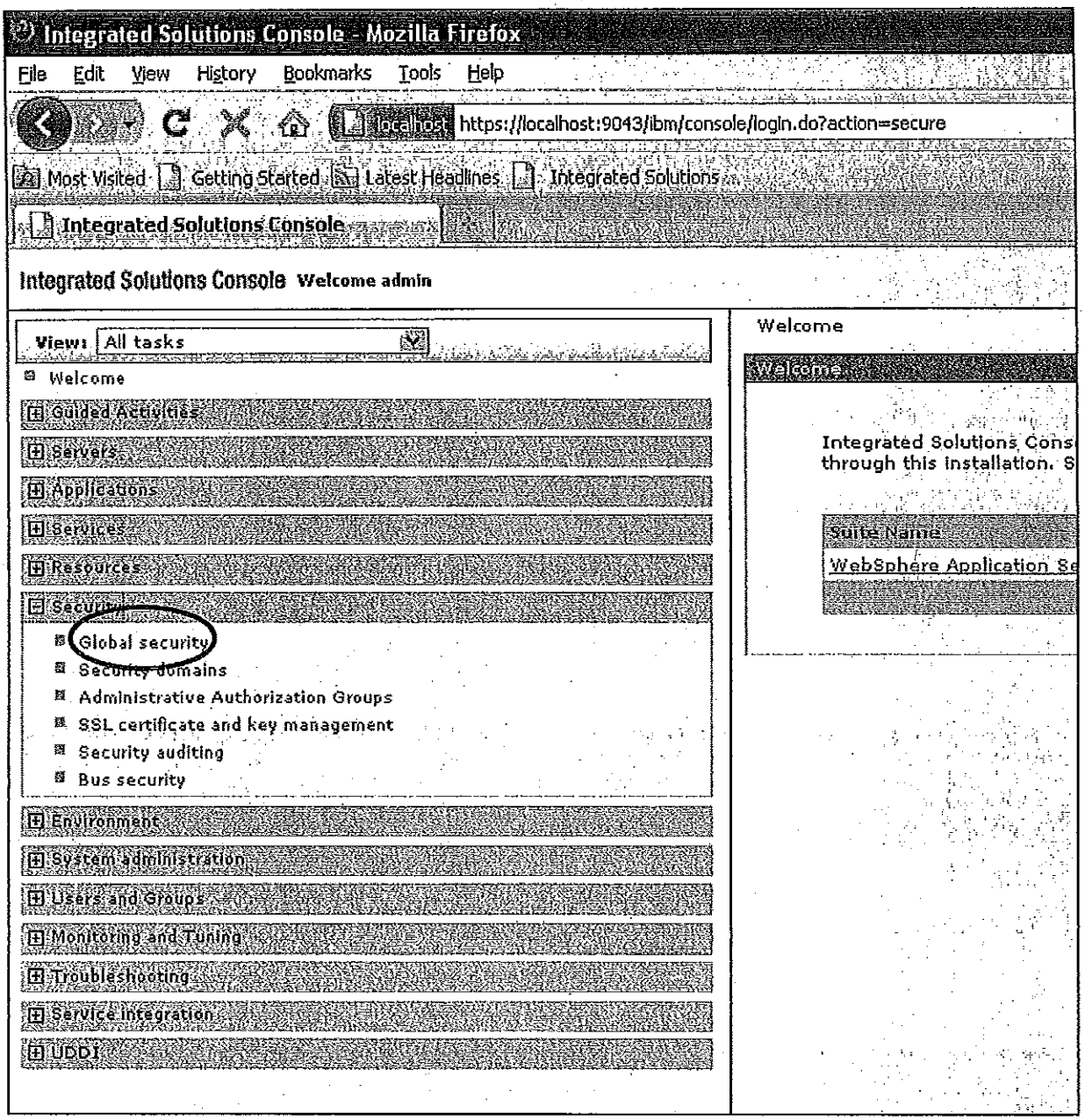
64. Wait for a while, following screen appears:



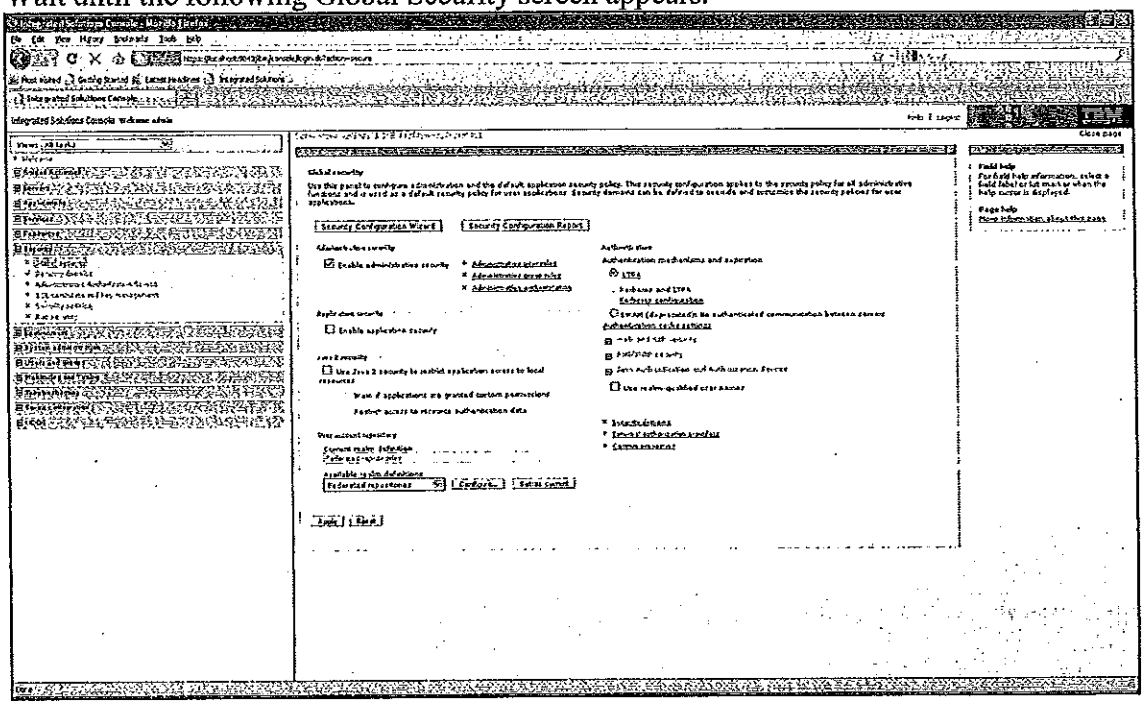
65. In the left hand side, click '⊞' Security to expand the Security settings



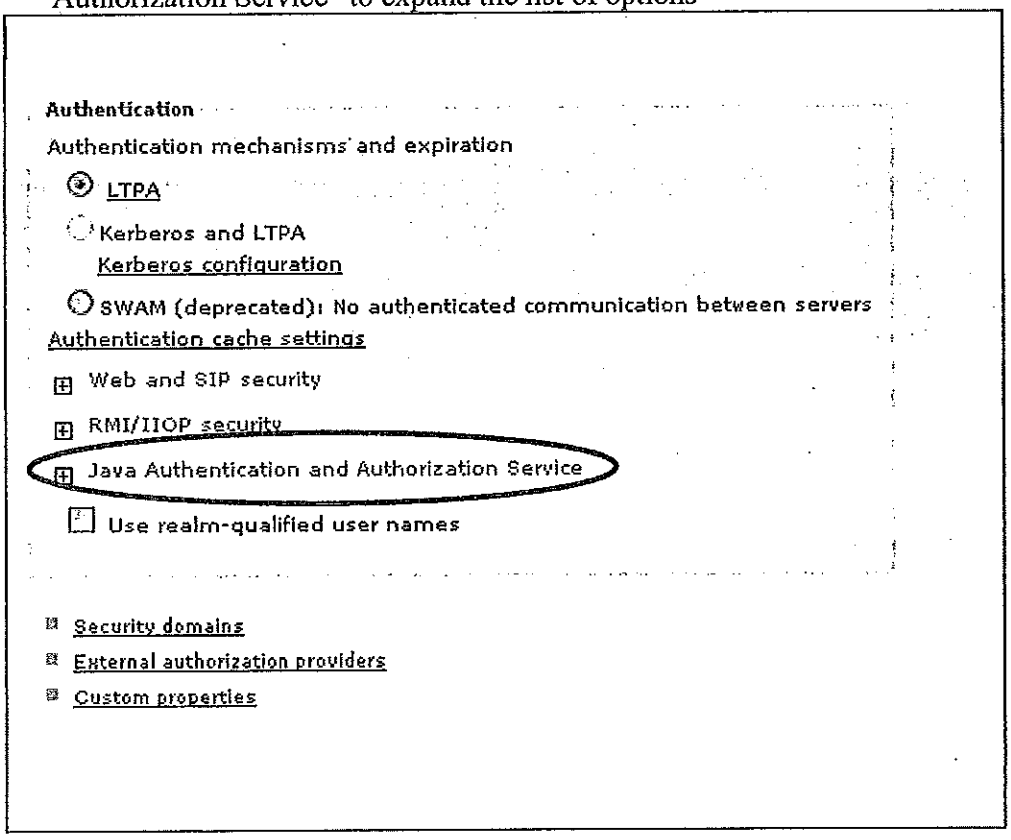
66. Click "Global Security" to create authentication alias for Mainframe access:



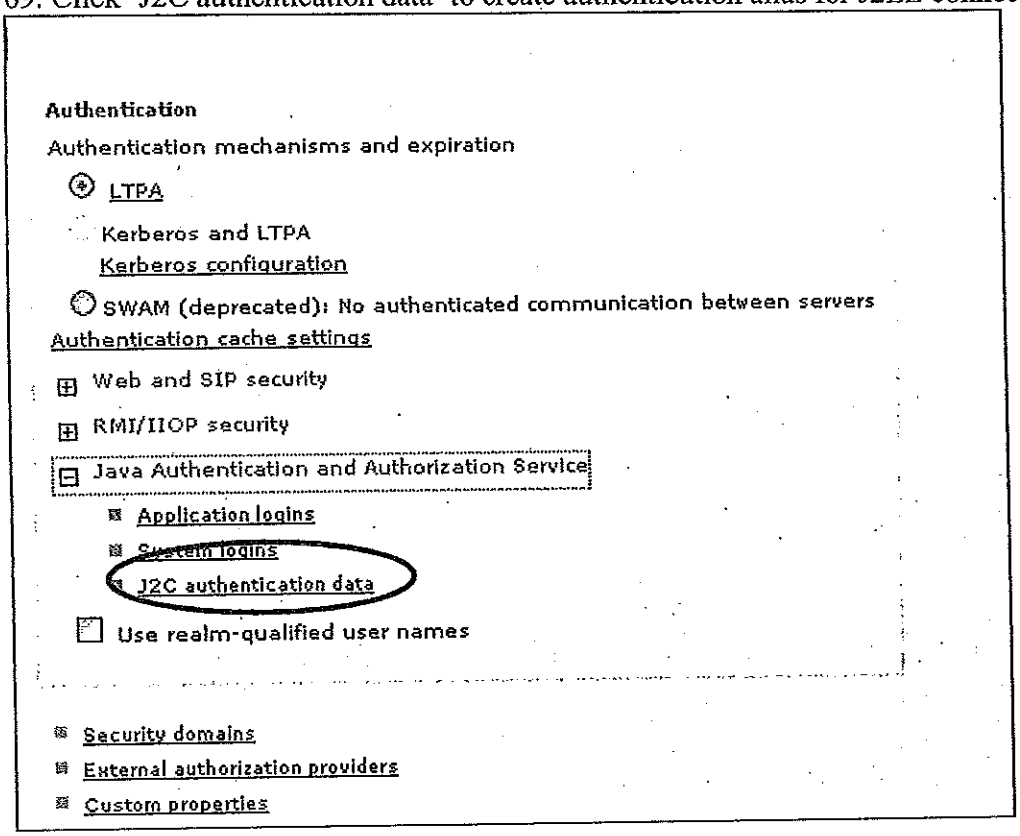
67. Wait until the following Global Security screen appears:



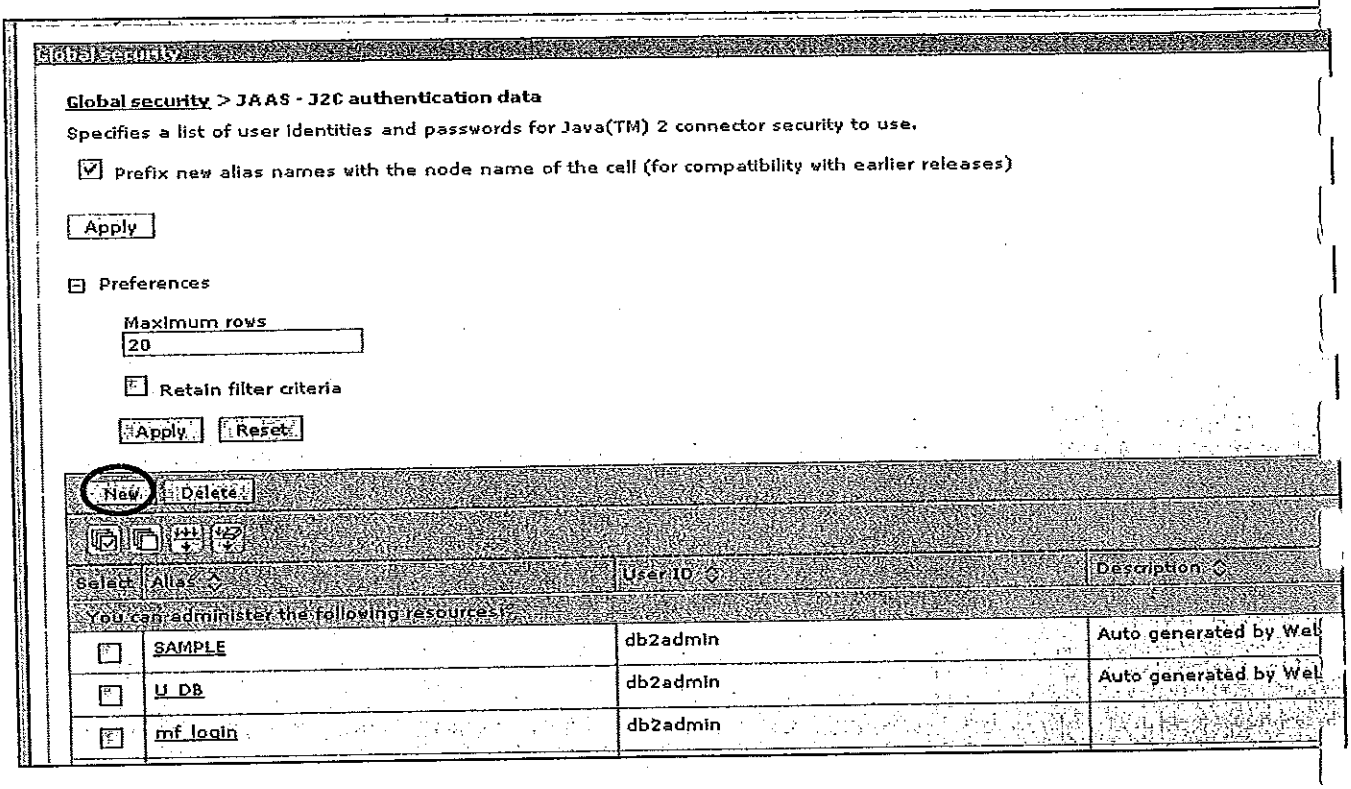
68. In the right hand side, under Authentication box, click "⊞" Java Authentication and Authorization Service" to expand the list of options



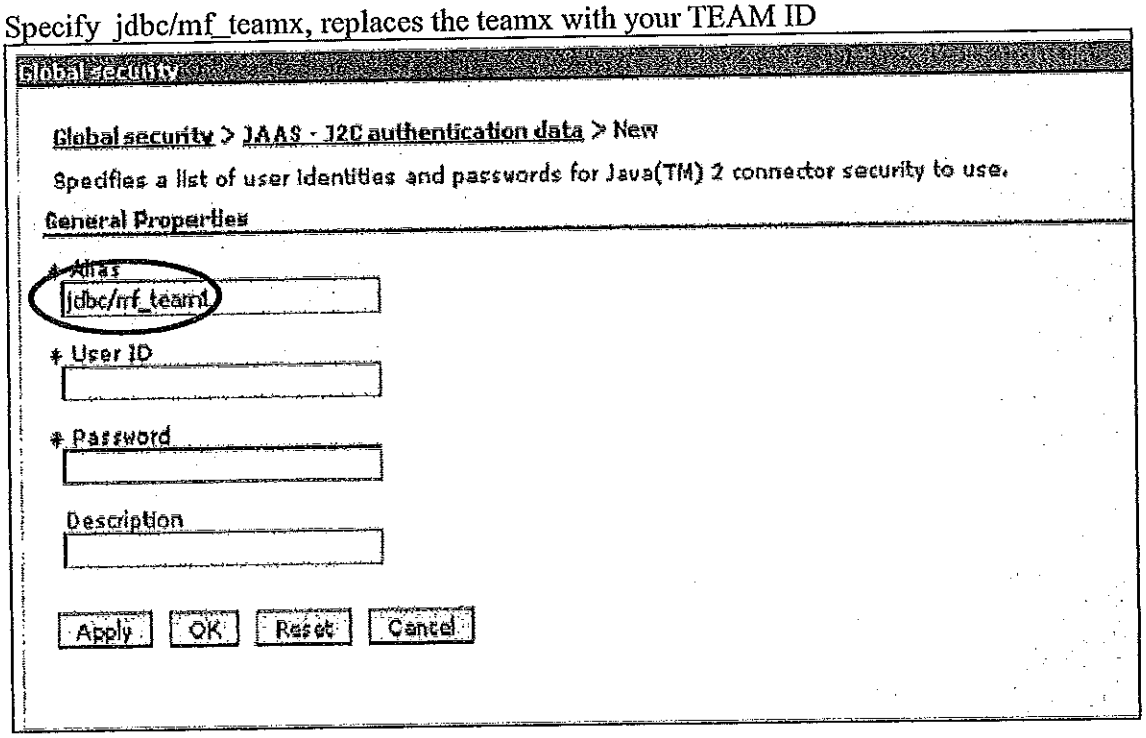
69. Click 'J2C authentication data' to create authentication alias for J2EE connector



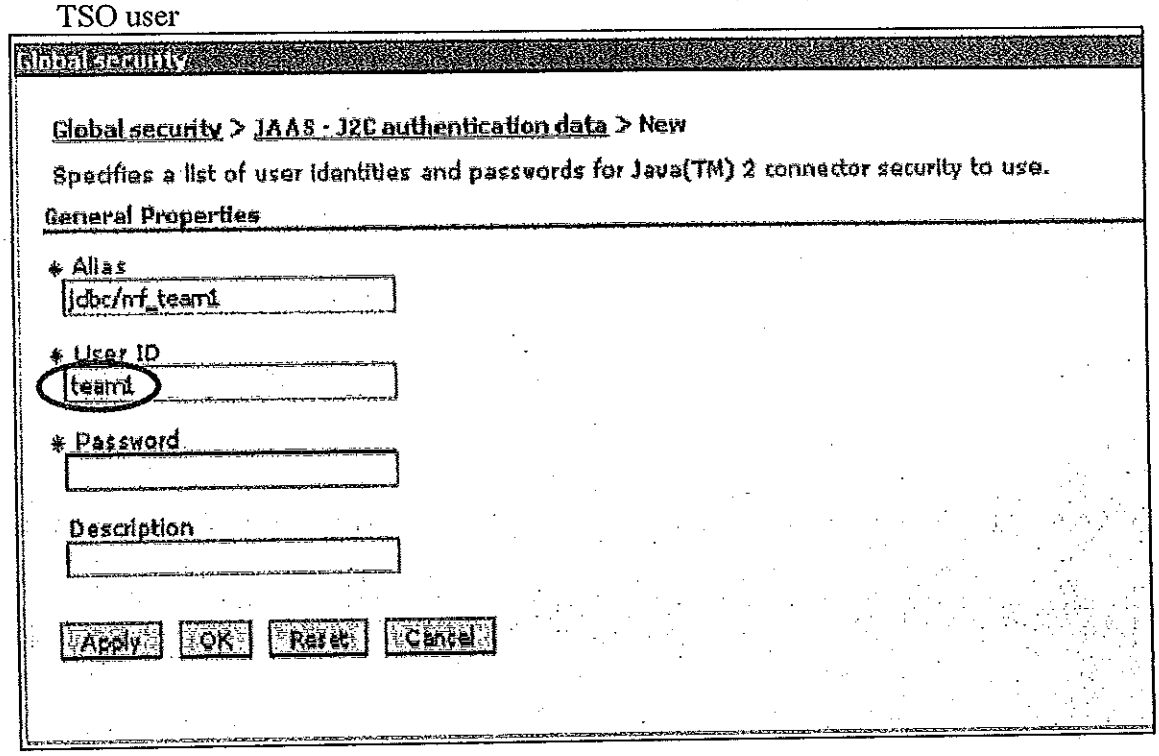
70. Click '[New]' button to create new authentication alias to access Mainframe



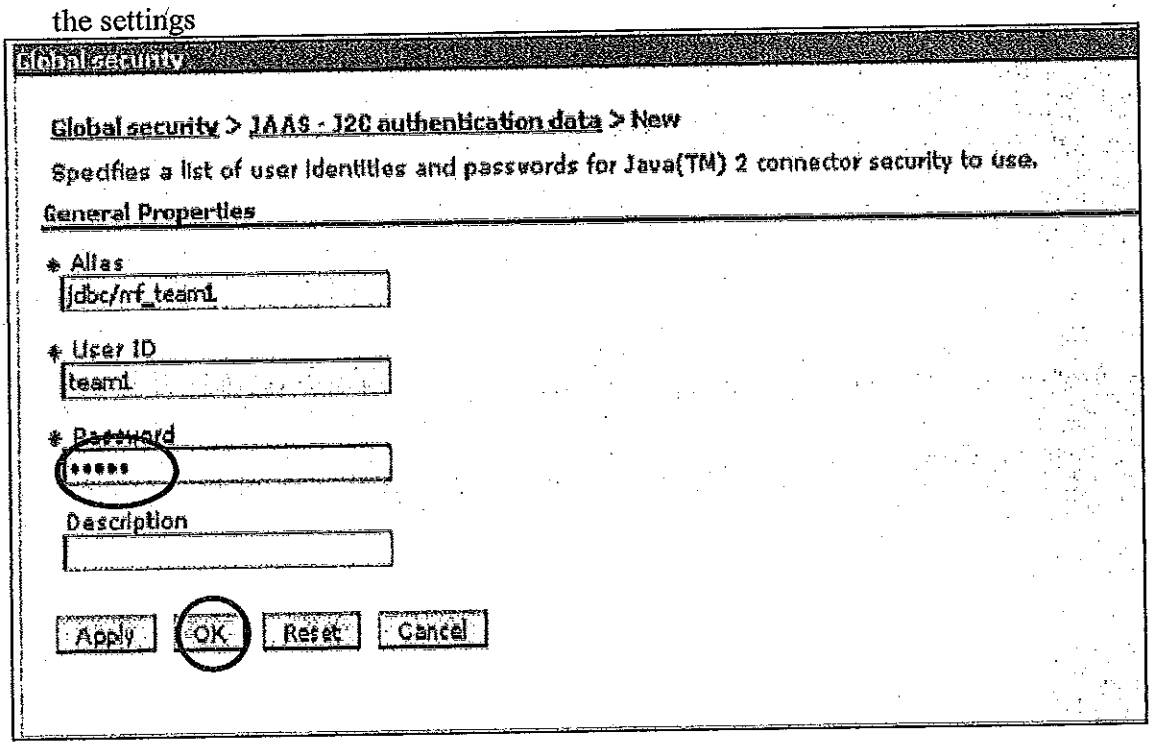
71. Specify jdbc/mf_teamx, replaces the teamx with your TEAM ID



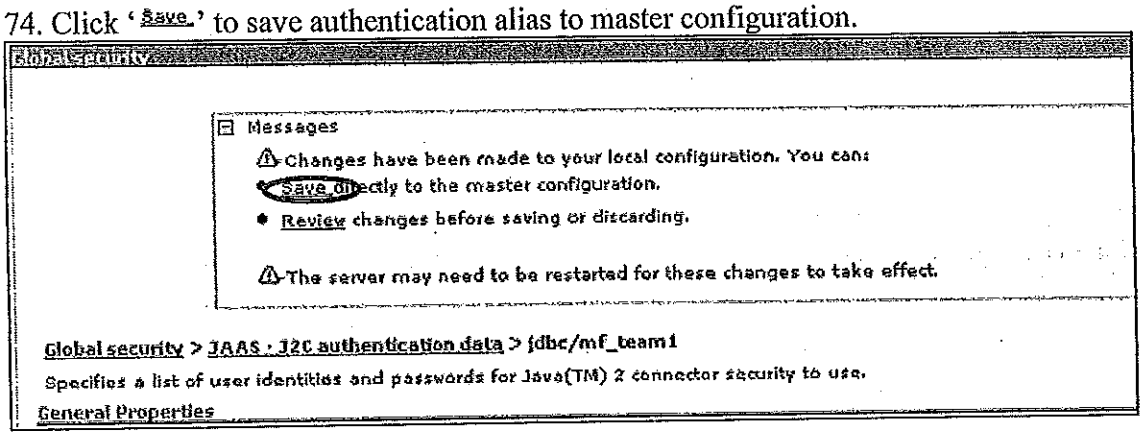
72. Provides your team ID for User ID which has been defined in Mainframe as DB2 and TSO user



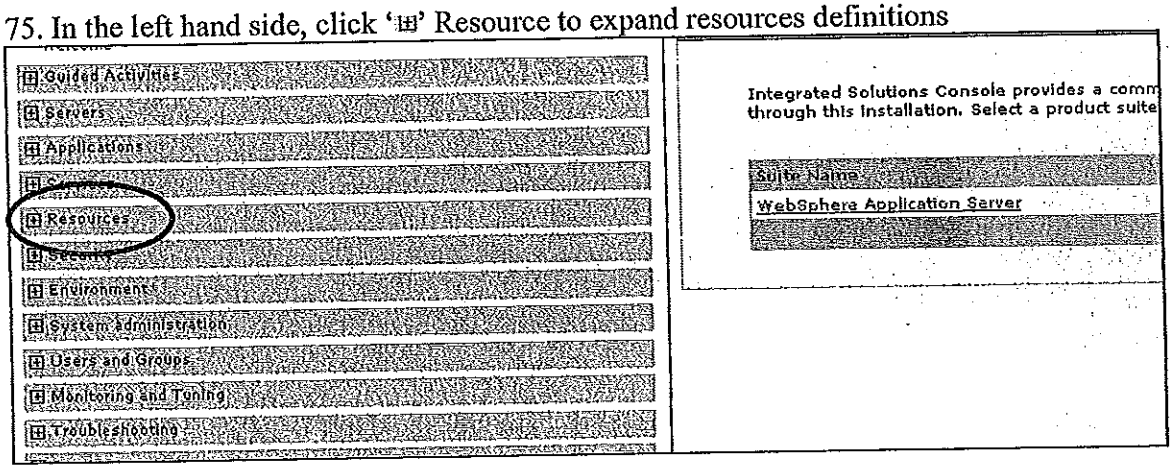
73. Provides the password which is same as the team ID and Click '[OK]' to confirm the settings



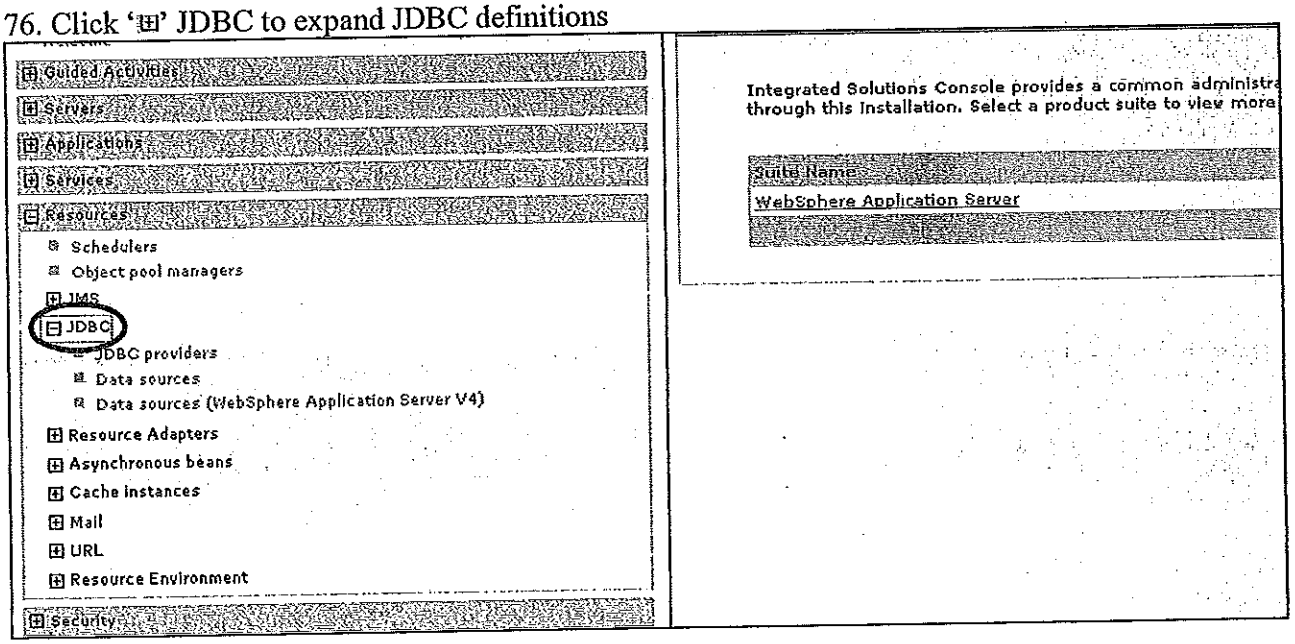
74. Click '<u>Save</u>' to save authentication alias to master configuration.



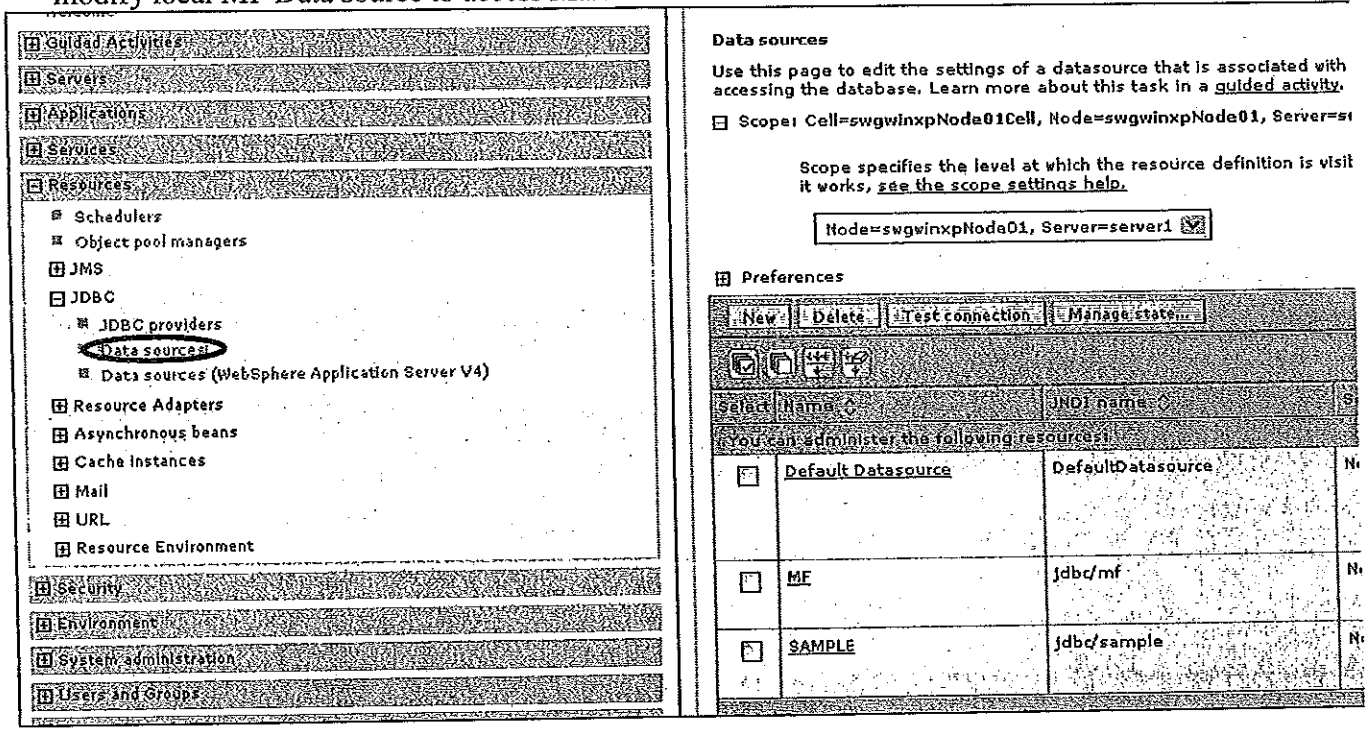
75. In the left hand side, click '⊞' Resource to expand resources definitions



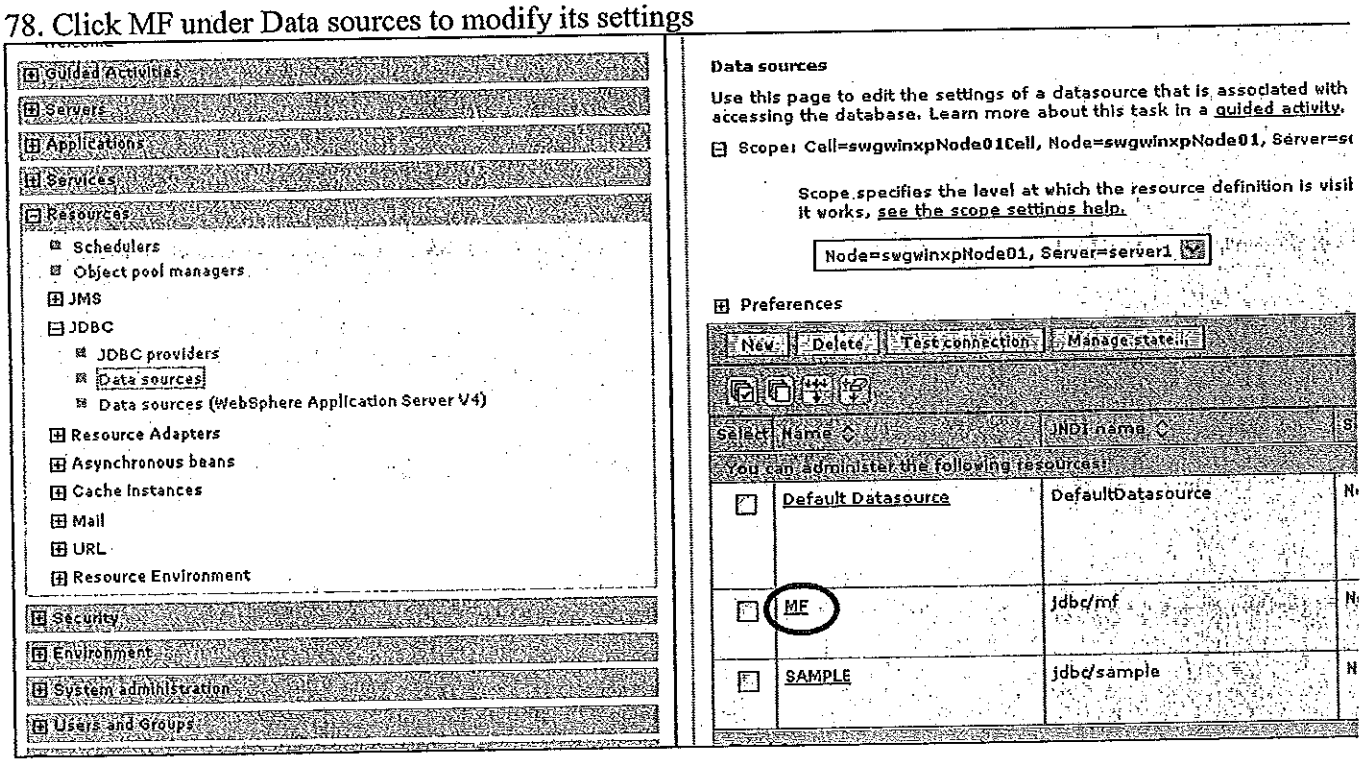
76. Click '⊞' JDBC to expand JDBC definitions



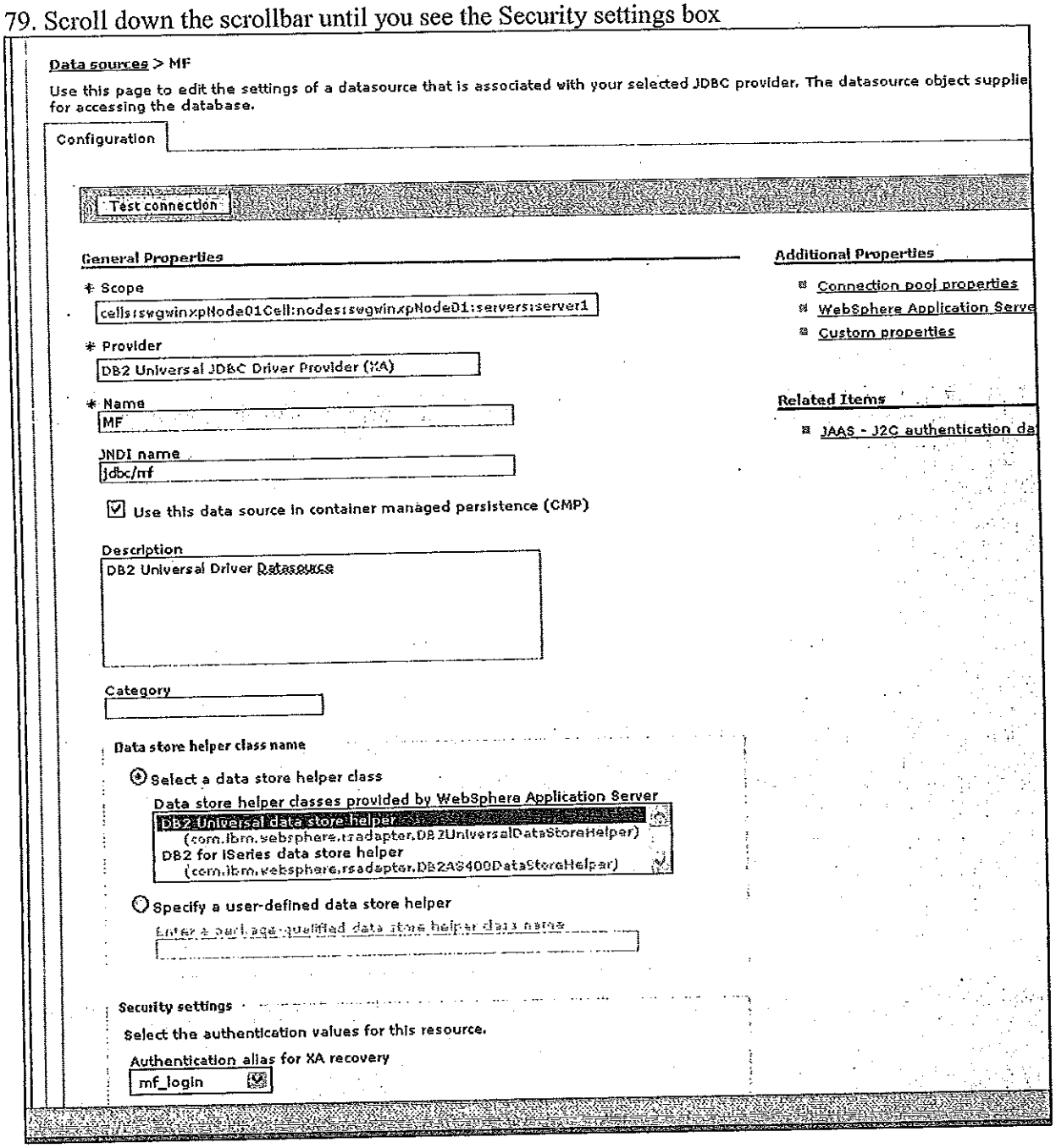
77. Click Data sources and wait until below Data source to be displayed. We are going to modify local MF Data source to access Mainframe DB2



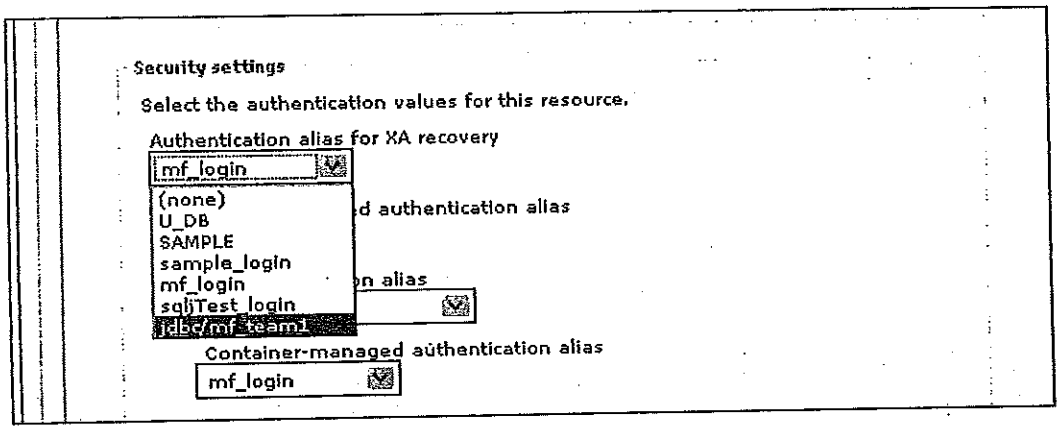
78. Click MF under Data sources to modify its settings



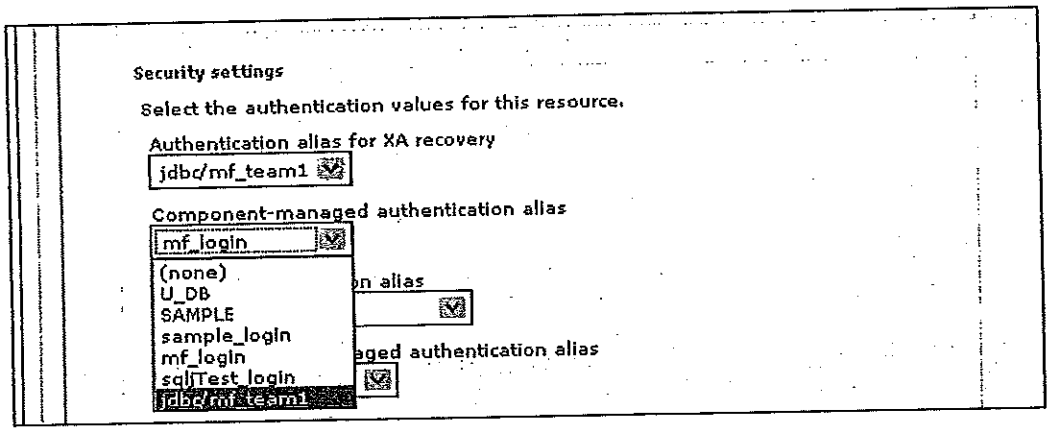
79. Scroll down the scrollbar until you see the Security settings box



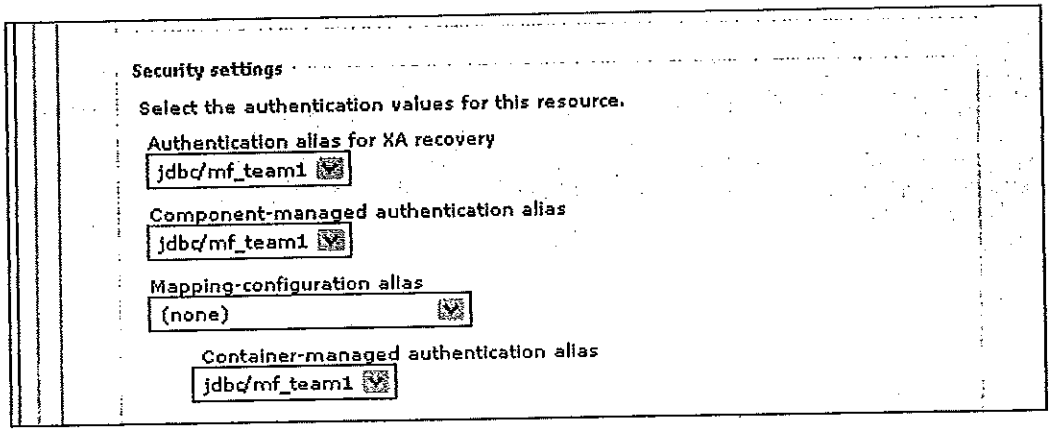
80. In Security settings box, click '▼' under Authentication alias for XA recovery, to select jdbc/mf_teamx which is defined at the beginning of this lab.



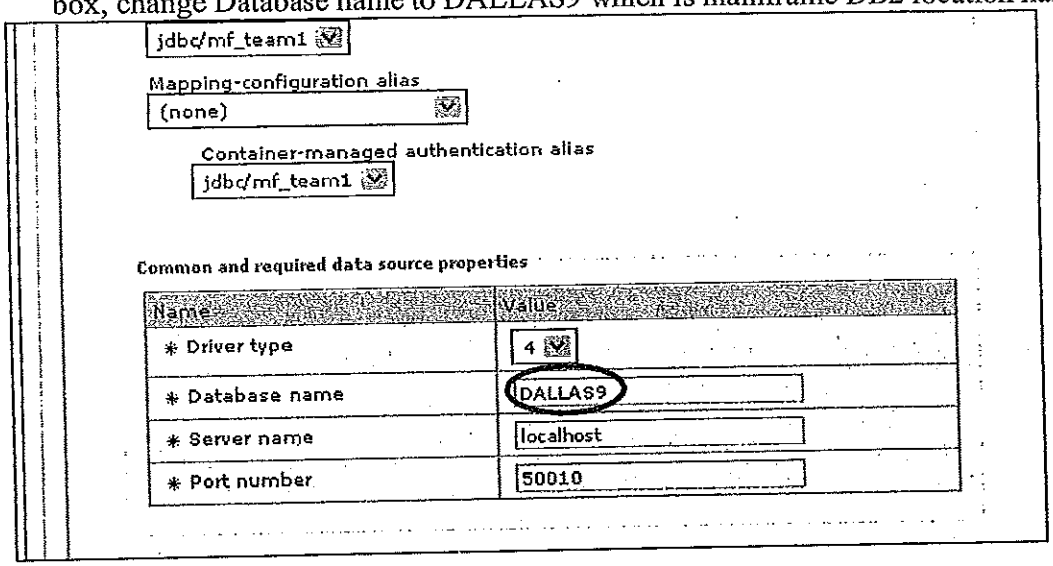
81. In Security settings box, click '▼' under Component-managed authentication alias, select jdbc/mf_teamx which is defined at the beginning of this lab.



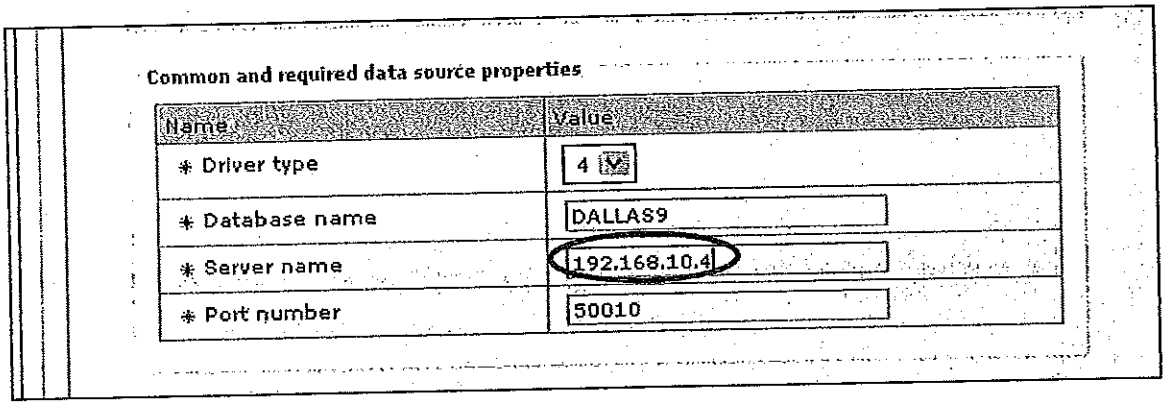
82. In Security settings box, click '▼' under Container-managed authentication alias, select jdbc/mf_teamx which is defined in the beginning of this lab.



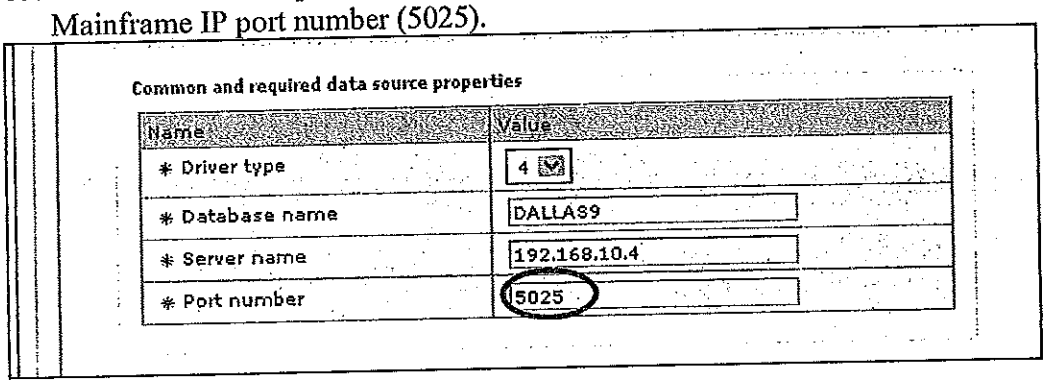
83. Scroll down to Common and required data source properties under Security settings box, change Database name to DALLAS9 which is mainframe DB2 location name.



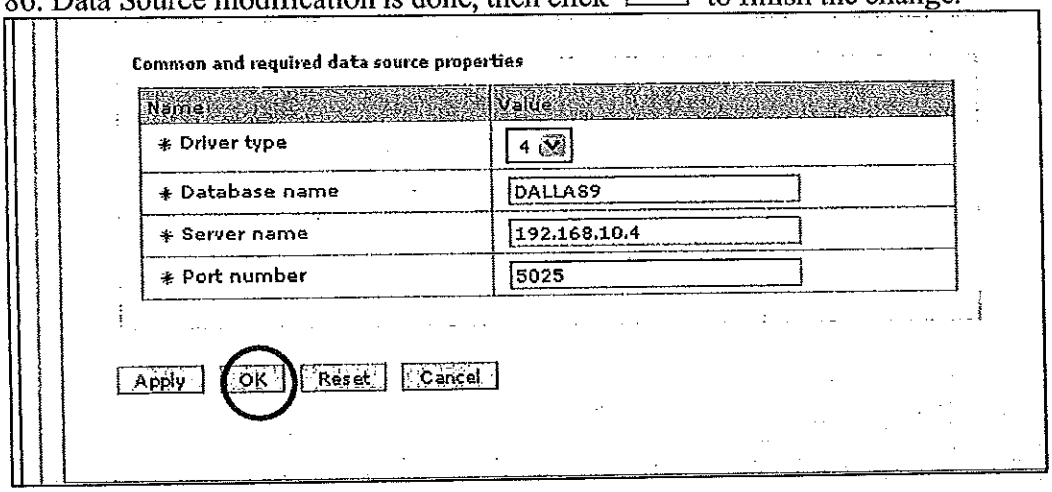
84. In Common and required data source properties box, change Server name to Mainframe IP address (192.168.10.4)



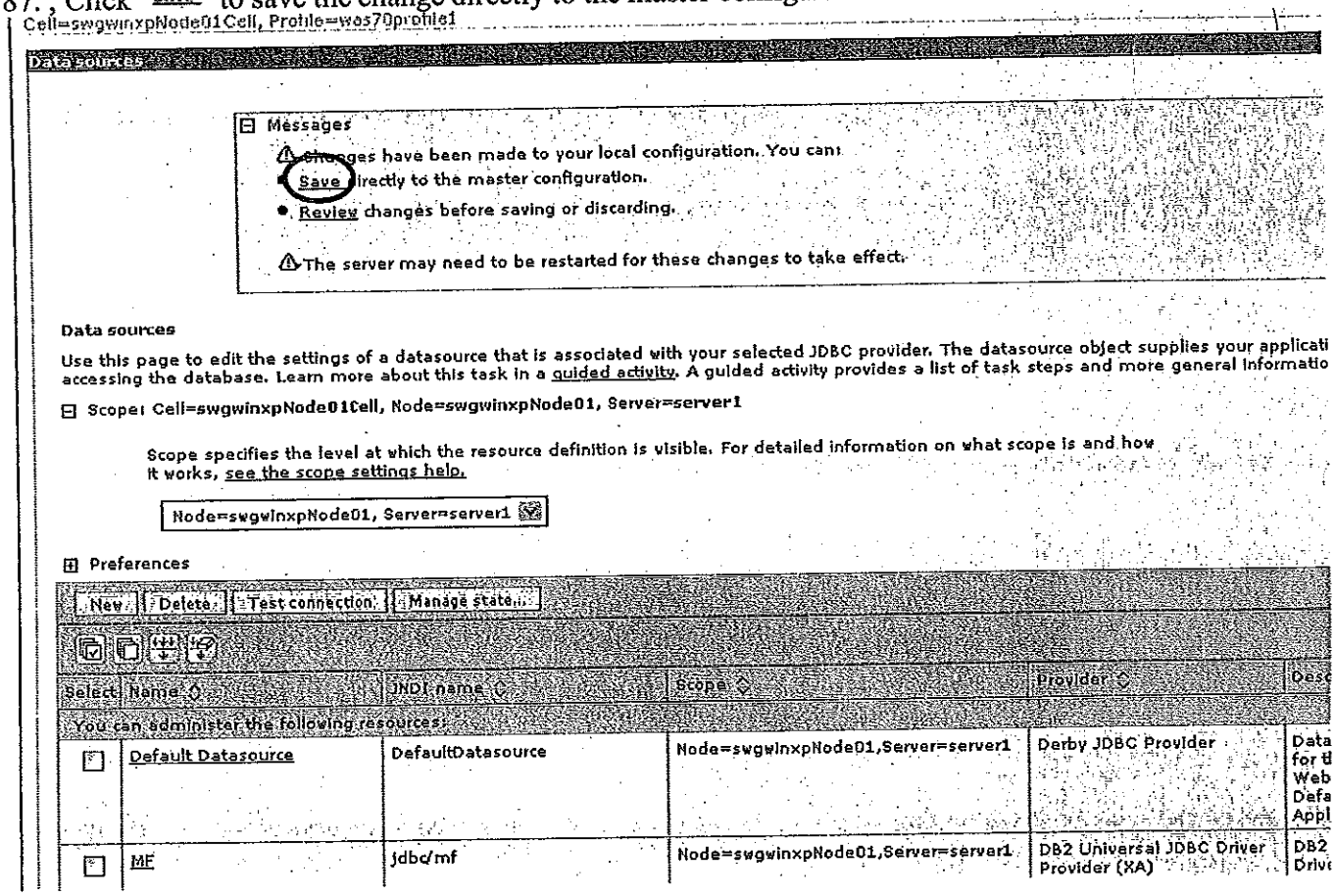
85. In Common and required data source properties box, change Port number to Mainframe IP port number (5025).



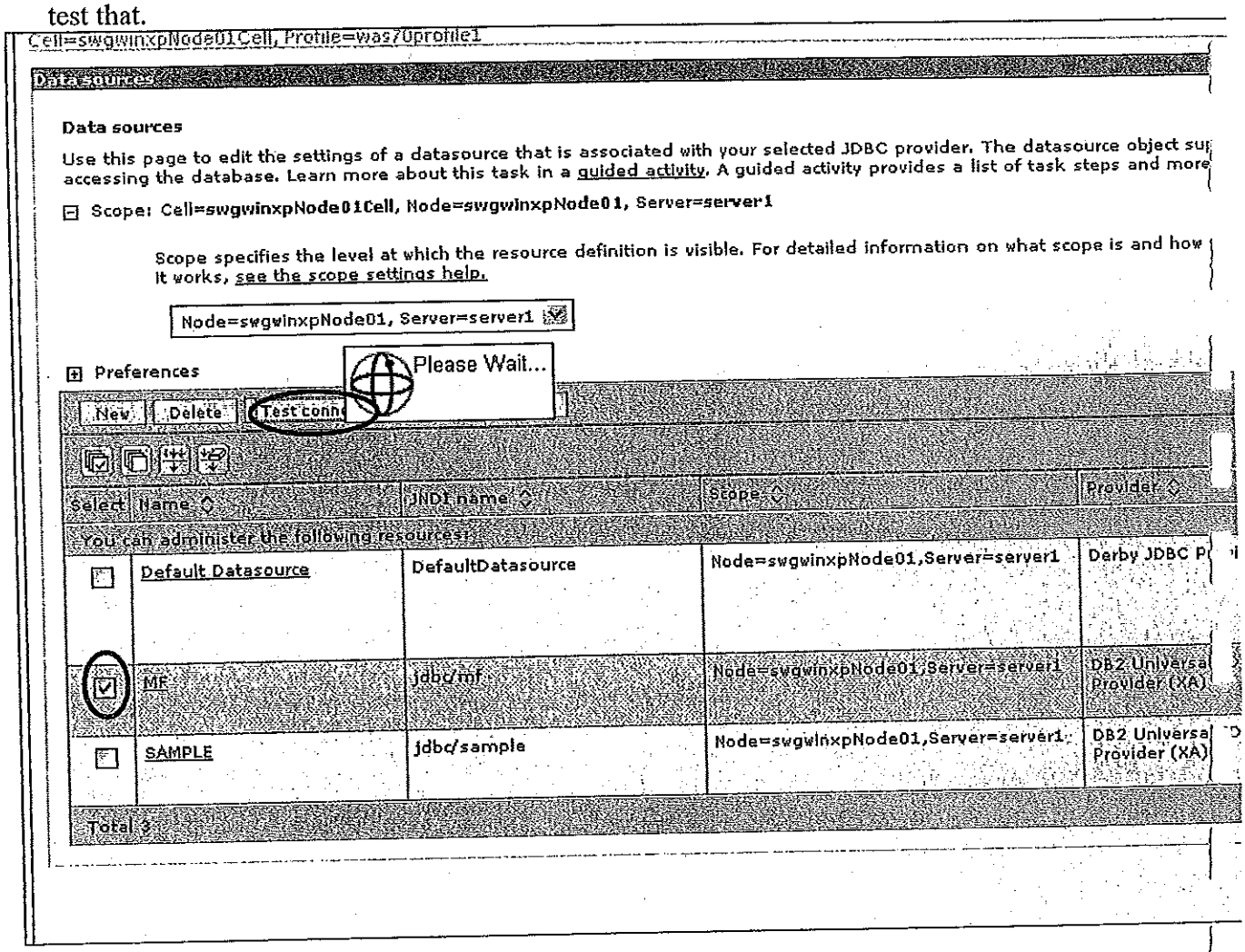
86. Data Source modification is done, then click '[OK]' to finish the change.



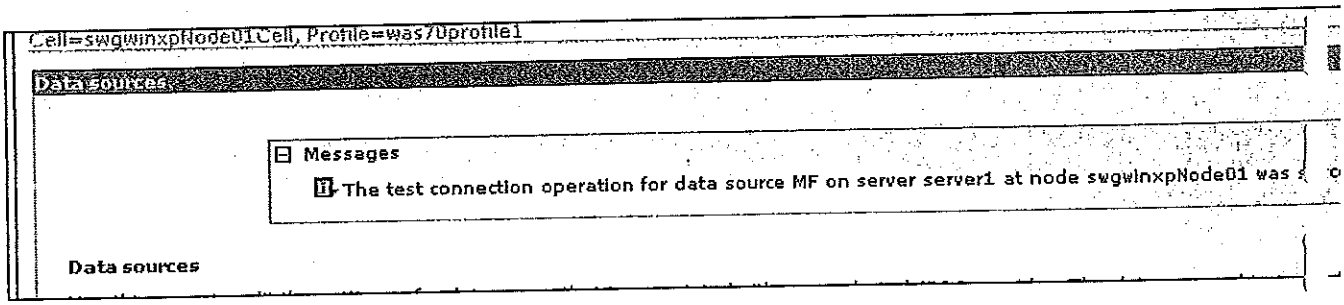
87. , Click '[Save]' to save the change directly to the master configuration



88. To verify whether the connection between the Websphere and Mainframe DB2 is good or not, click '☐' of modified MF data source and then click ' Test connection ' to test that.

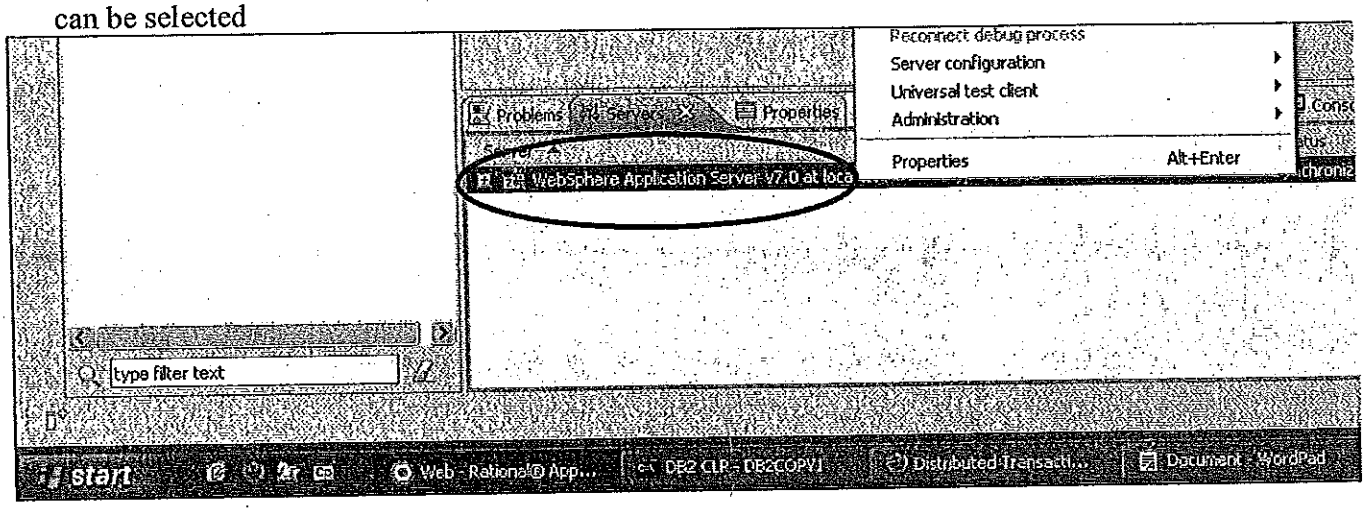


89. Successful Test connection Message displays as below:

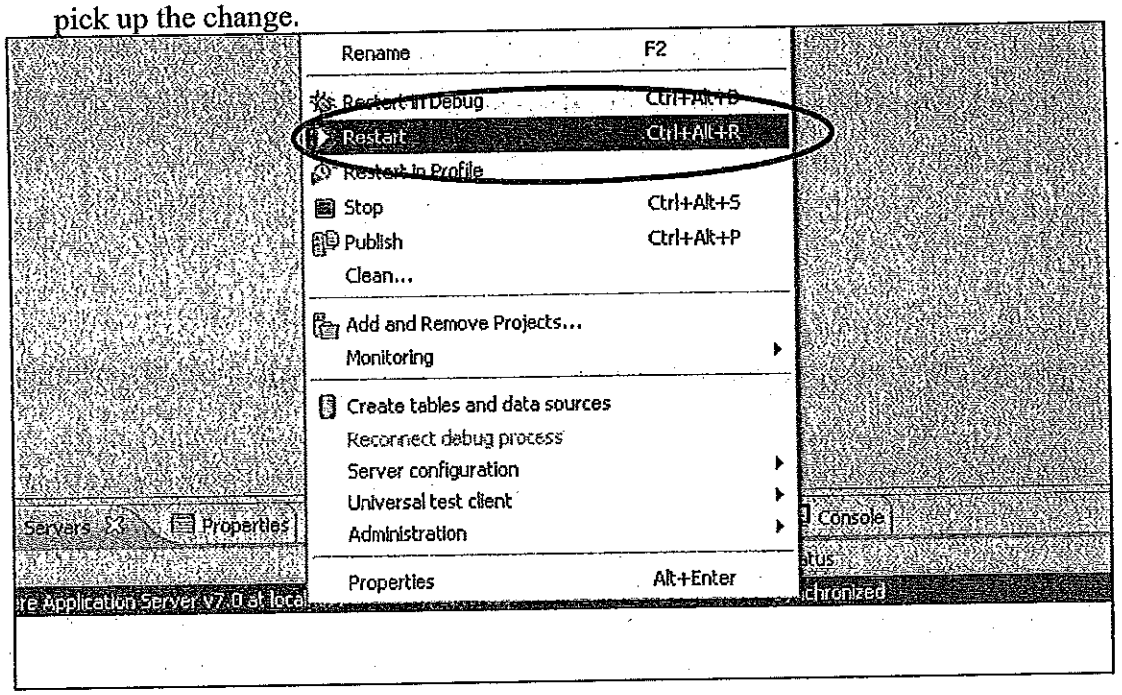


90. Need to recycle WebSphere to pick up the change. Click Web - Rational® App... in the task bar at the bottom of desktop to switch to RAD application development environment.

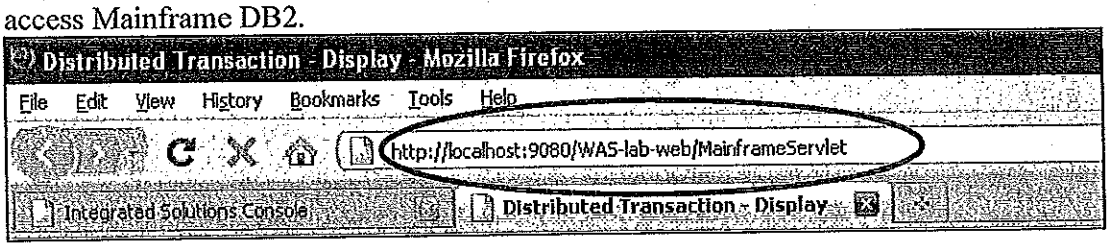
91. Right Click WebSphere Application Server v7.0 at loc, it pops up a list of options that can be selected



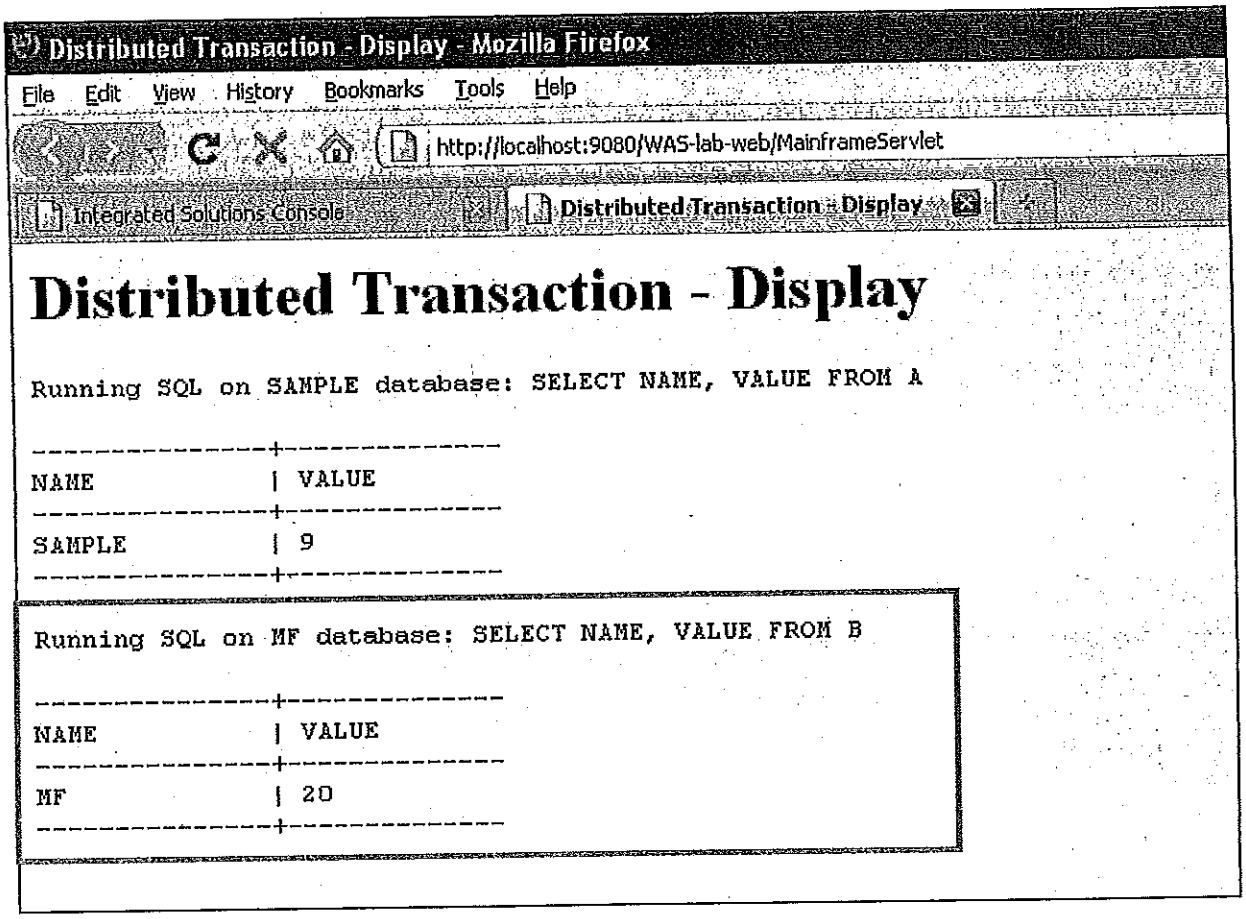
92. Click Restart to cycle WebSphere Server to clean up connection pool buffer and pick up the change.



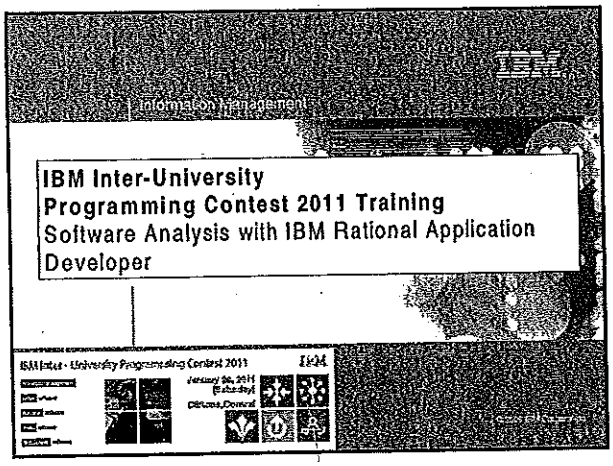
93. Key-in http://localhost:9080/WAS-lab-web/MainframeServlet to start Servlet to access Mainframe DB2.



94. The values in the green box are stored in Mainframe DB2 table. You have successfully completed this LAB, Well done. Congratulation.



95. Well done. Congratulation.



Agenda

- Introduction to Integrated Development Environment (IDE)
- Introduction to Static Analysis
- Real World Static Analysis Examples
- Lab

IBM.

What is Integrated Development Environment (IDE) ?

- An integrated development environment (IDE) also known as integrated design environment or integrated debugging environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of:
  - Source code editor
  - Compiler and/or an interpreter
  - Code visualisation tool
  - Build automation tools
  - Debugger or Profiler
  - Version Control tool or adaptor
  - Runtime test environment integration

IBM.

Eclipse (Open Source) IDE

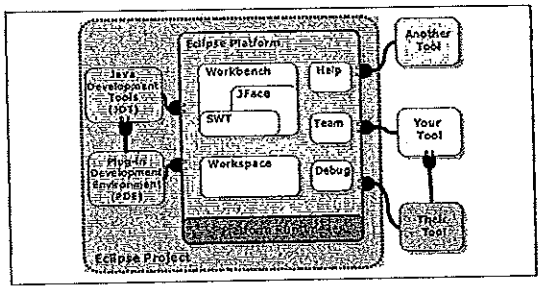
www.eclipse.org (free to download)

- Multi-language software development environment comprising an IDE and a plug-in system to extend it. It is written primarily in Java and can be used to develop applications in Java and, by means of the various plug-ins, in other languages as well, including C, C++, COBOL, Python, Perl, PHP, and others. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C, Eclipse JDT for Java and Eclipse PDT for PHP.

Release	Date	Platform version
Eclipse 3.0	28 June 2004	3.0
Eclipse 3.1	28 June 2005	3.1
Callisto	30 June 2006	3.2
Europa	29 June 2007	3.3
Ganymede	25 June 2008	3.4
Galileo	24 June 2009	3.5
Helios	23 June 2010	3.6

IBM.

Eclipse Architecture

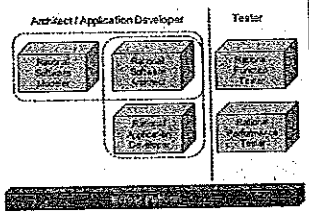


IBM

Commercial Tool: IBM Rational Application Developer

IBM Rational Application Developer for WebSphere Software (RAD) is an integrated development environment (IDE) that extends Eclipse,for:

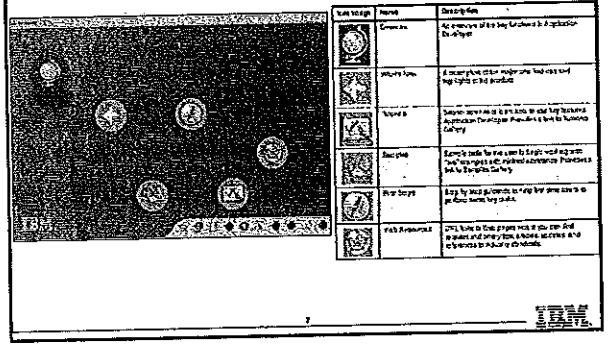
- Visually designing,
- Constructing,
- Testing, and deploying Web application, web services, portals, and
- Java Enterprise Edition (JEE) applications.



Note: IBM Rational Application Developer v7.0 is based on Eclipse v3.4

IBM

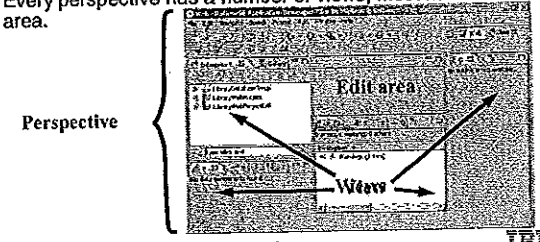
Workbench basic



IBM

Perspectives, views, and editors

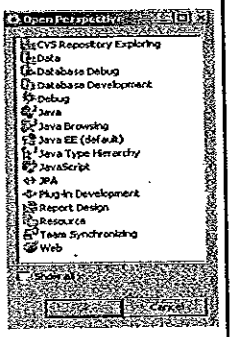
- A **perspective** is a collection of views, toolbar icons, and menus, grouped to accomplish a specific type of work.
- A **view** supports editors and presents information differently.
- **Editors** allow you to modify the code.
- Every perspective has a number of views; most have an edit area.



IBM

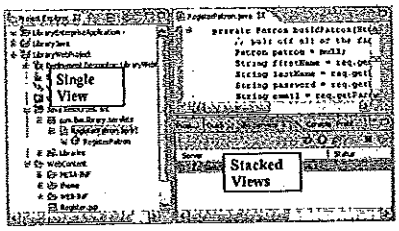
Perspectives

- A perspective defines the initial set and layout of views in the Workbench window.
- Each perspective provides functionality aimed at accomplishing a specific type of task, or works with specific types of resources.
- Perspectives control what appears in certain menus and toolbars.
  - They define visible action sets, which you can change to customize a perspective.



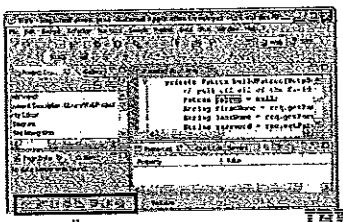
Views

- There are two states for views
  - Single
    - There are no other views at that position in the perspective
  - Stacked
    - There are other available views, represented by tabs at the top of the view pane



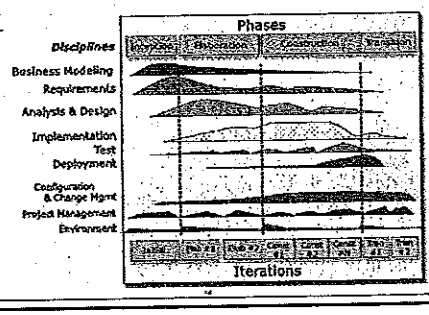
Fast views

- You can quickly open and close fast views.
  - This functionality is available via the fast view bar
- There are two ways to create a fast view:
  - Drag a view onto the shortcut bar
  - Right-click the view icon and choose **Fast View**
- To restore a fast view, right-click the icon in the shortcut bar, and clear the **Fast View** option.



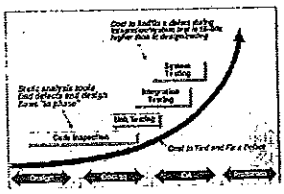
Rational Unified Process (RUP) Project Lifecycle

- A typical software development process consists of the following phases and disciplines
  - Each phase is broken into Iterations



The high cost of fixing defects

- A single defect can cost between $12-18K.
- Thousands of potential defects in a large piece of software can cost between $600K-$2.7M.



IBM

A real-world example: Ford recall

- A two-year investigation determined that the problem was system-related.
- Pain includes:
  - costs of the recall itself
  - bad PR
  - customer satisfaction
  - potential safety lawsuits
- Potential cost to manufacturer of **$54 million dollars.**

IBM

Mechanisms for identifying defects early in the lifecycle

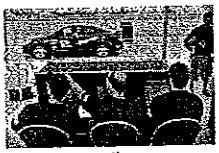
- Agile Process
  - Test Driven Development
- Peer Code Review

IBM

give [illegible] opinion.

Agile Process - Test Driven Development

- A technique involving short iterations, where:
  - New test cases covering new functionality are written first.
  - Next, the production code necessary to pass tests is implemented.
  - Finally, the software is refactored to accommodate changes.



IBM

Static analysis : 100% coverage
runtime analysis : 2 to 20

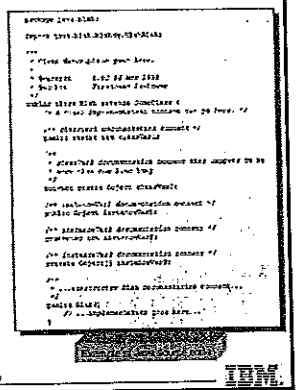
Peer code review

- A systematic examination of source code by human experts.
  - Intended to find and fix mistakes overlooked in the initial development phase.
  - Improves both the overall quality of software and the developers' skills.

IBM.

Adherence to coding guidelines

- **Coding guidelines** are important for a number of reasons:
  - 80% of the lifetime cost of a piece of software goes to maintenance.
  - Hardly any software is maintained for its whole life by the original author.
  - Adhering to coding guidelines improves the readability of the software, allowing engineers to understand new code more quickly and thoroughly.



IBM.

The static analysis tool

- Processes such as agile development and peer code review are not enough.
- More and more, we need the assistance of automated analysis tools such as **IBM Rational Application Developer** that can ensure adherence to coding guidelines.
- The best time to find problems is to review the source code as it is written.

Note that static analysis is simply another tool to improve code quality. It is not a complete replacement for manual code reviews.

IBM.

What is static analysis?

- **Static analysis** is the study of source and/or binary code that is not currently executing.
- Static analysis can:
  - Ensure that the source adheres to a predefined coding standard.
  - Detect common performance problems.
  - Understand the dependencies of the imports of each class.

IBM.

Static analysis vs. runtime analysis

- **Static analysis**
  - The study of source and/or binary code that is not currently executing.
  - Examines architectural elements of software to find problems relating to dependencies in code (What is the impact of changing a class?)
  - Examines the complexity of code (Are there too many paths through the code?)
  - Simulates data movement through a system testing for data security problems (does the password get passed outside the system?)
- **Runtime analysis**
  - Understanding software component behavior by using data collected during the execution of the component.
  - Provides information about how the developed component - or the whole application - behaves when it runs.
  - Provides explanations for various exposed or potential misbehaviors.

IBM

Categories of static analysis – Code Review

- Performs automated code parsing
  - Each source file is loaded and passed through a **parser** that looks for particular code patterns violating a set of established rules.
    - In some languages like C++, rules are built into the compiler or available in external programs like Lint.
    - In other languages like Java, the compiler does little in the way of automated code review.
- Code Review is a good tool to:
  - Enforce coding standards
  - Find basic performance problems
  - Find possible API abuse.

IBM

Categories of static analysis – Code Dependency

- Does not examine the format of individual source files.
- Examines the **relationships between** source files (typically classes) to build a map of the overall architecture of a program.
- Commonly used to discover known design patterns (good) or common anti-patterns (bad) in code.

IBM

Categories of static analysis – Code Complexity

- Analyzes the program code and compares it to established software metrics
  - Determines if it is unnecessarily complex.
- If a particular piece of code exceeds a given threshold, it can be flagged as a candidate for **refactoring** to help improve maintainability.

IBM

Categories of static analysis – **Trending**

- Trend analysis does not use code artifacts directly.
- It is the study of improvements/degradations in code quality based on other forms of analysis.
  - Analyzing the results of analysis.
- Results generated by trend analysis typically appeal to managers and executives.
  - They make a statement about the direction of quality improvements, answering the question "Is the code getting better or worse?

IBM

Real-world Example in Java #1

- Rule: "Always surround if and loop statements with curly braces"

```
if( condition )
     methodCall();
     anotherMethodCall();
```

- What was the developer's intended behavior?
- Is this a real bug?

IBM

Real-world example in Java #1 **Solution**

- Both method calls should have been called when the "if" was true

```
if( condition ) {
     methodCall();
     anotherMethodCall();
}
```

- This was a real bug!

IBM

Real-world example in Java #2

- Rule: "Avoid returning null instead of iterator"

```
public Iterator myMethod() {
     List list = getList();
     if( trueCondition ) {
          return list.iterator();
     }
     return null;
}
```

- What's wrong with this?

cost!

return null object

IBM

Real-world example in Java #2

- Inline usage of myMethod()

```
for( Iterator it = myMethod(); it.hasNext(); ) {
     // Do something
}
```

- Under the right conditions myMethod() can return null resulting in NPE
- This is particularly bad because "the right condition" may not happen until a customer executes the code.

IBM

Real-world example in Java #2 Solution

- Modify myMethod()

```
public Iterator myMethod() {
     List list = getList();
     if( trueCondition ) {
          return list.iterator();
     }
     return new ArrayList(0).iterator();
}
```

IBM

Real-world example in Java #3

- Rule: "Avoid multiple invocations of the same method"

```
public static void satisfy( List fullList, IRuleFilter filter ) {
    for( Iterator it = fullList.iterator(); it.hasNext(); ) {
        ASTNode node = (ASTNode)it.next();

        boolean satisfied = filter.satisfies( node );
        if (!filter.isSuccessful()
              || ((filter.isInclusive() && !satisfied)
              || (!filter.isInclusive() && satisfied))) {
            it.remove();
        }
    }
}
```

- Why does filter.isInclusive() get called twice?

cost! (run 2 times)
use locally to store result
(performance problem)

IBM

Real-world example in Java #3 Solution

- Sometimes a seemingly harmless call is expensive
- Add a temporary variable

```
public static void satisfy( List fullList, IRuleFilter filter ) {
    boolean inclusive = filter.isInclusive();
    for( Iterator it = fullList.iterator(); it.hasNext(); ) {
        ASTNode node = (ASTNode)it.next();

        boolean satisfied = filter.satisfies( node );
        if (!filter.isSuccessful()
              || ((inclusive && !satisfied)
              || (!inclusive && satisfied))) {
            it.remove();
        }
    }
}
```

IBM

rad: 6xx rules!

Real-world example in Java #4

- Rule: "Consider using HashSet instead of List"

```
List employees = new ArrayList();
...
if (!employees.contains(emp)) {
   employees.add(emp);
}
```

- What is wrong with this code?

Real-world example in Java #4 Solution

- Sets assure uniqueness without requiring programmers to check for duplicates

```
Set employees = new HashSet();
...
employees.add(emp);
...
```

- Better performance since it does not require searching through the entire collection
- If order is important use a LinkedHashSet

Lab Overview

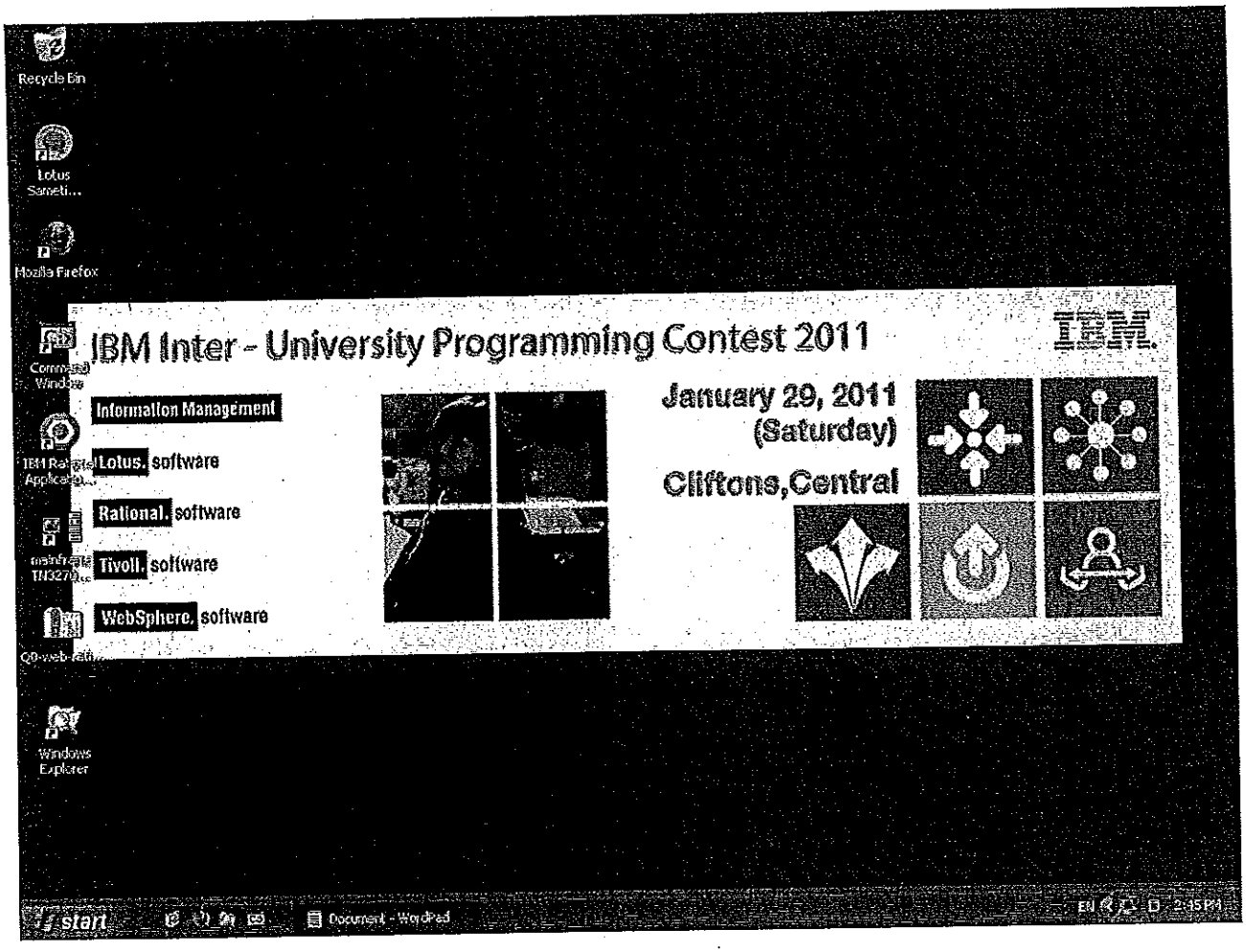
- Create Rule for Analysis
- Configuration of Analysis
- Analysis the project

Enjoy your lab !

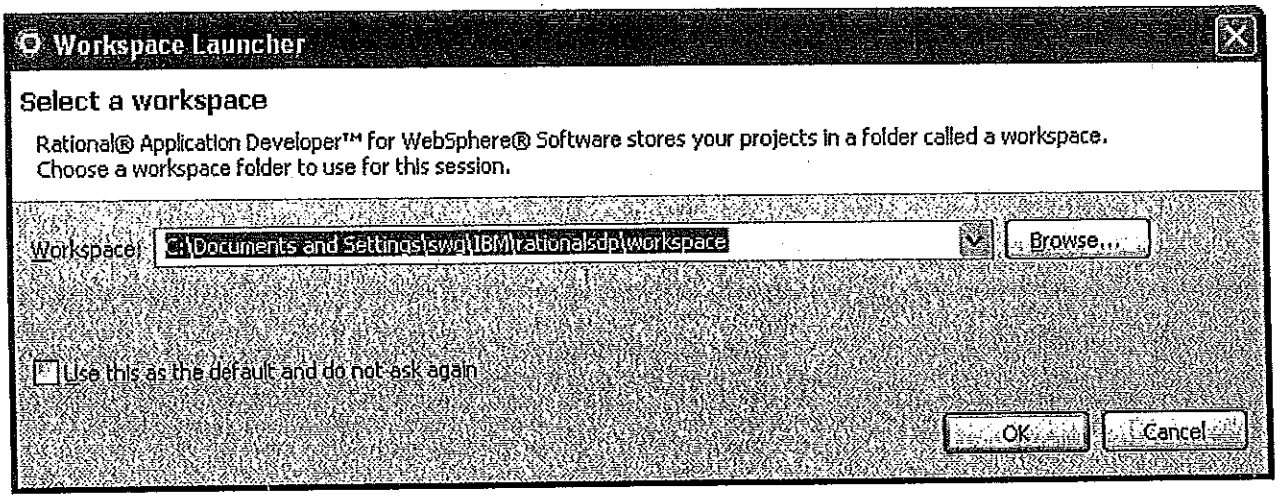
Chapter 6a: Software Analyzer using IBM Rational Application Developer

Start IBM Rational Application Developer (RAD) to develop J2EE application which will be used during the programming contest. RAD is a GUI integrated development environment (IDE) which is used to perform J2EE and web programming. The resultant programs will be published to WebSphere Application Server so that it can be executed in the web environment.

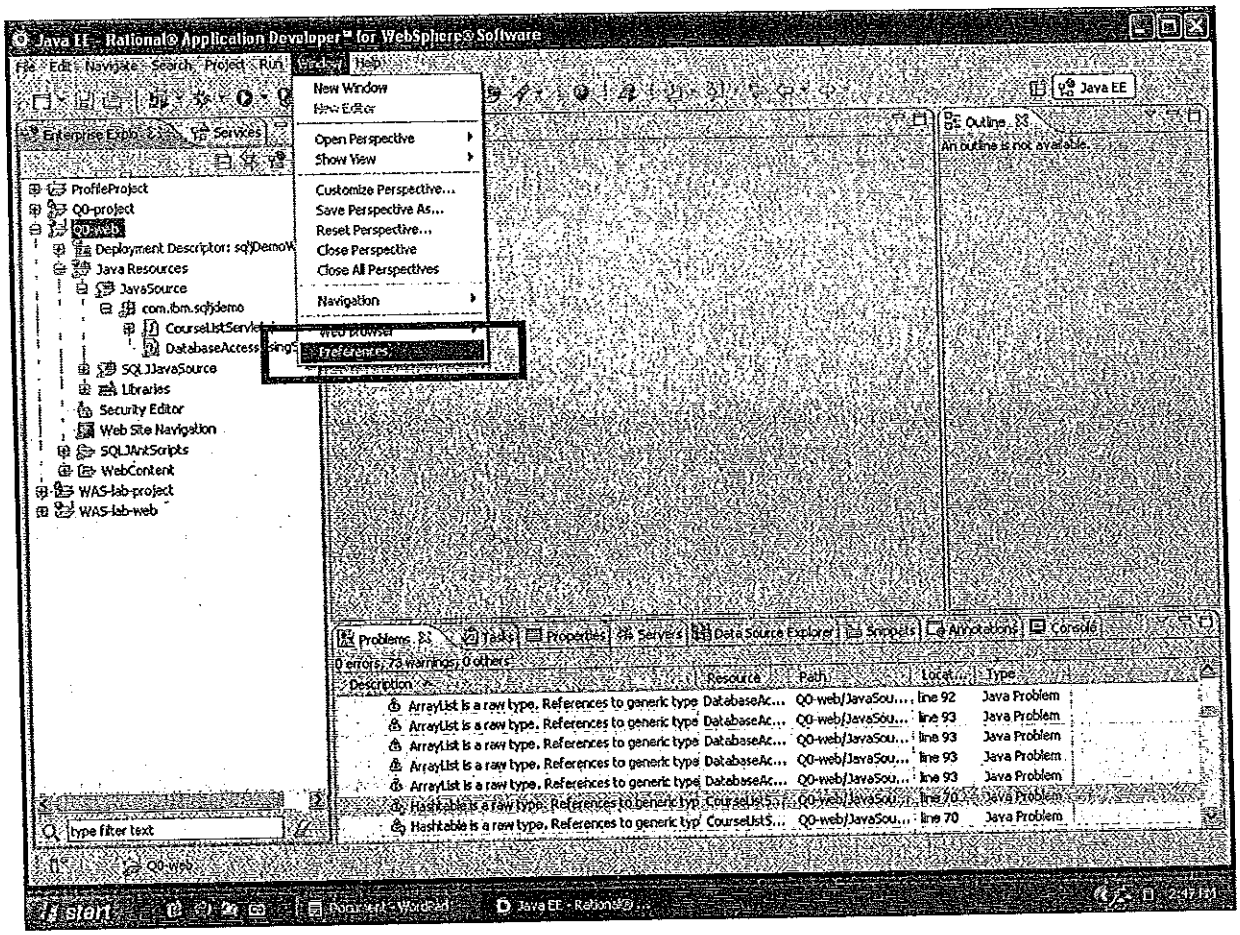
1. Double clicked the icon "IBM Rational Application Developer" on Desktop to start.



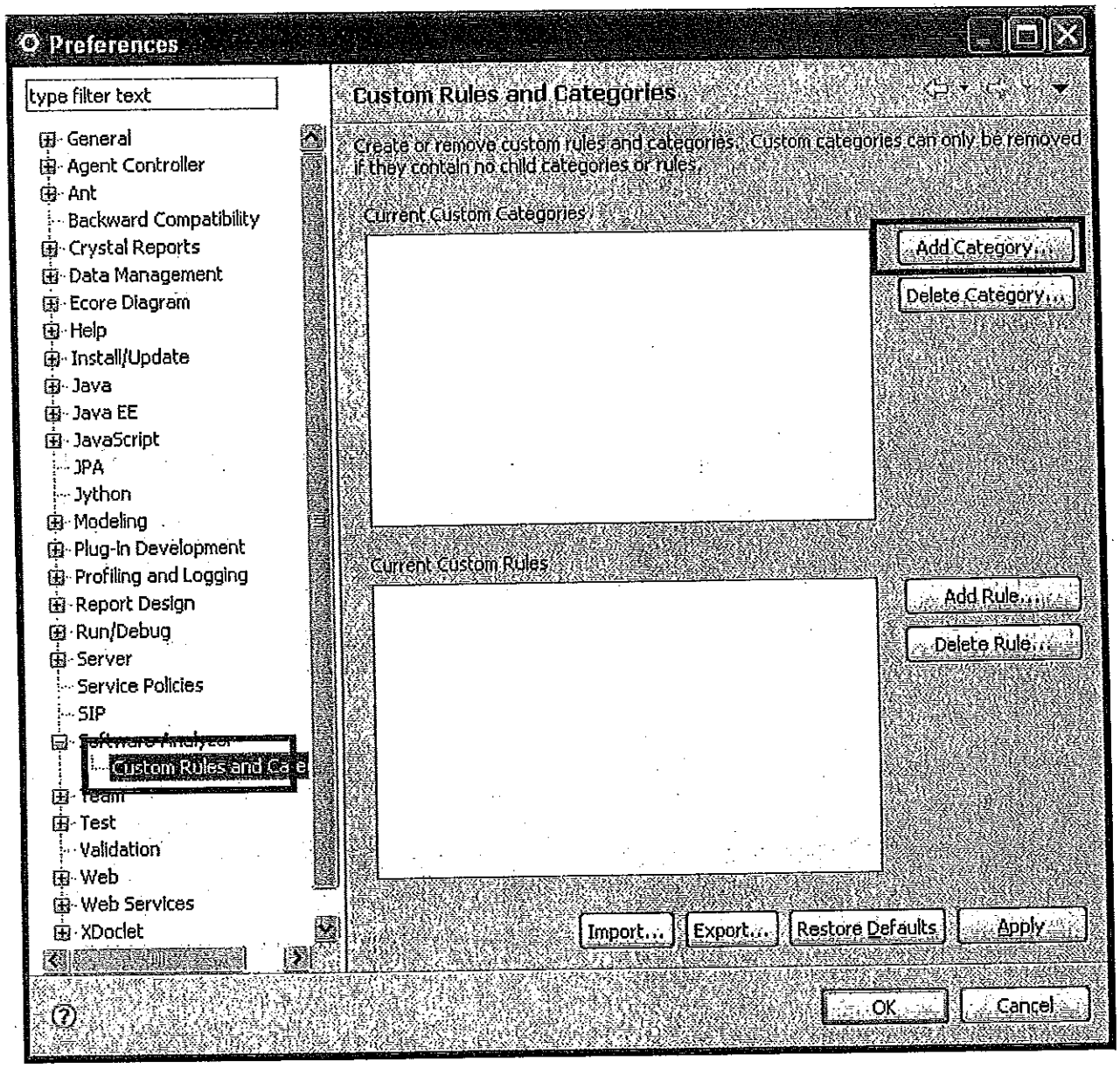
2. Banner will appear and a 'Workspace Launcher' window will appear which allows you to choose the base location of the workspace, type: "*C:\Documents and Settings\swg\IBM\rationalsdp\workspace*" as workspace.



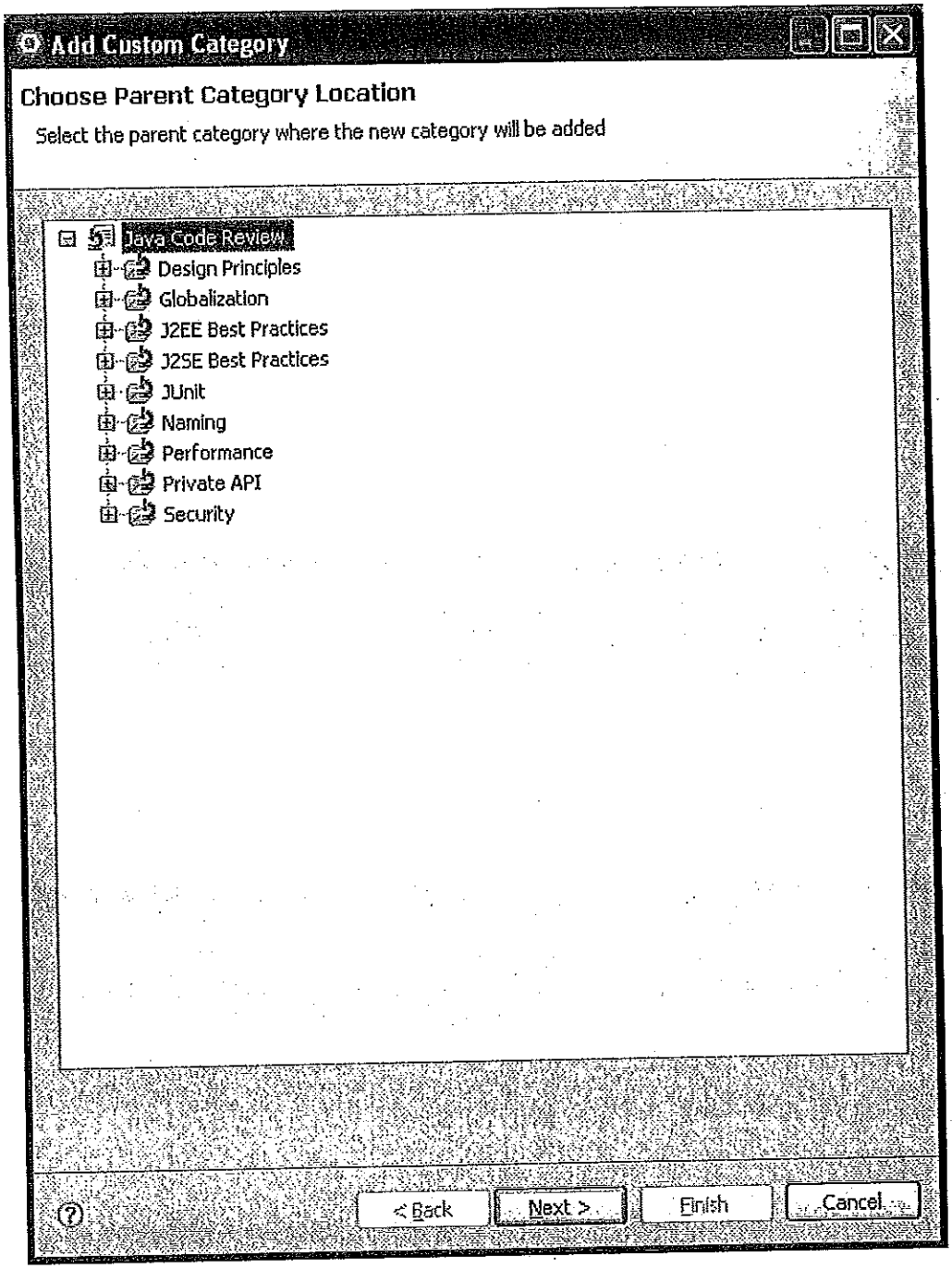
3. To config a rule, select window > Preferences



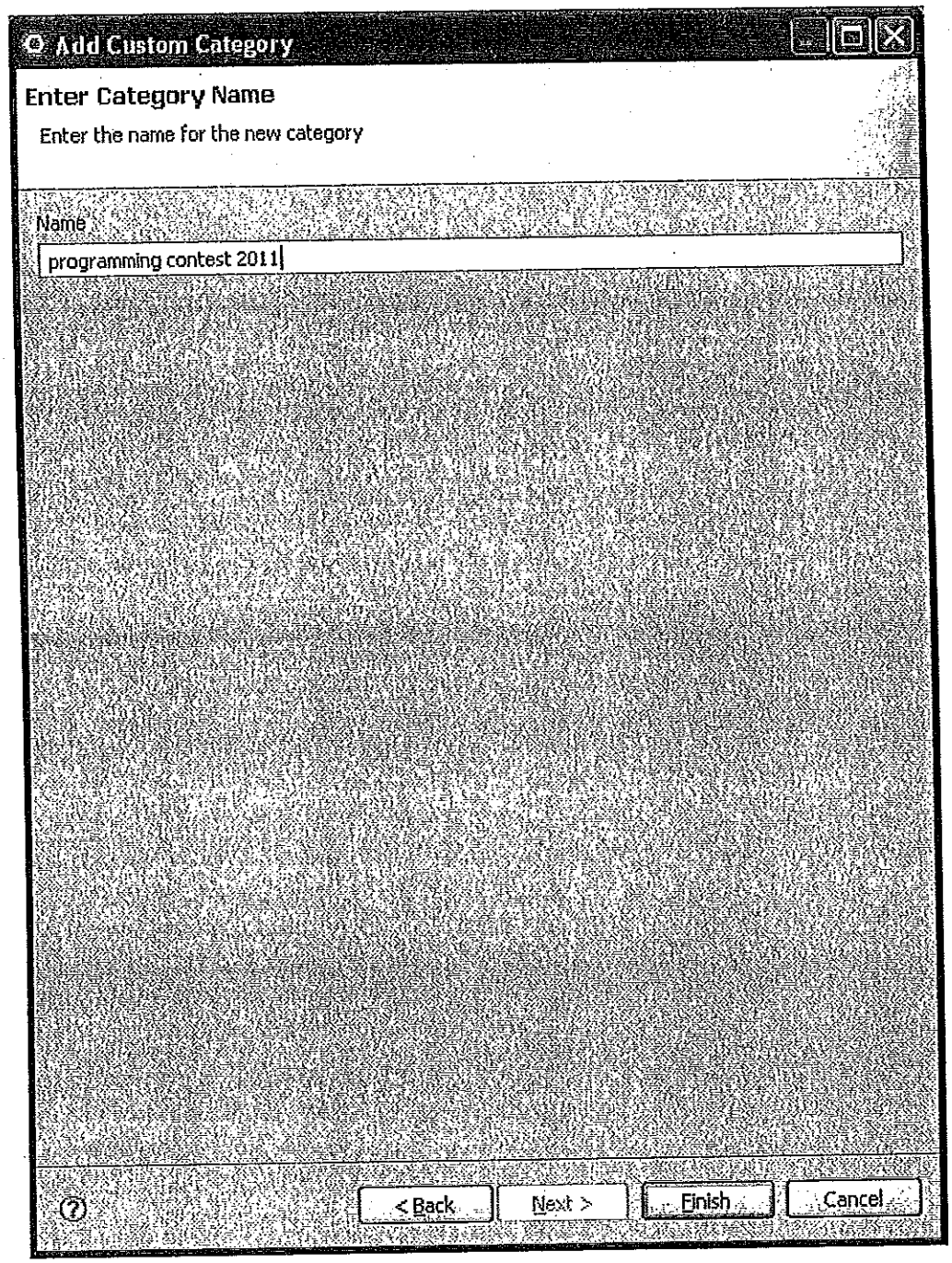
4. Select Software Analyzer > Custom Rules and Categories



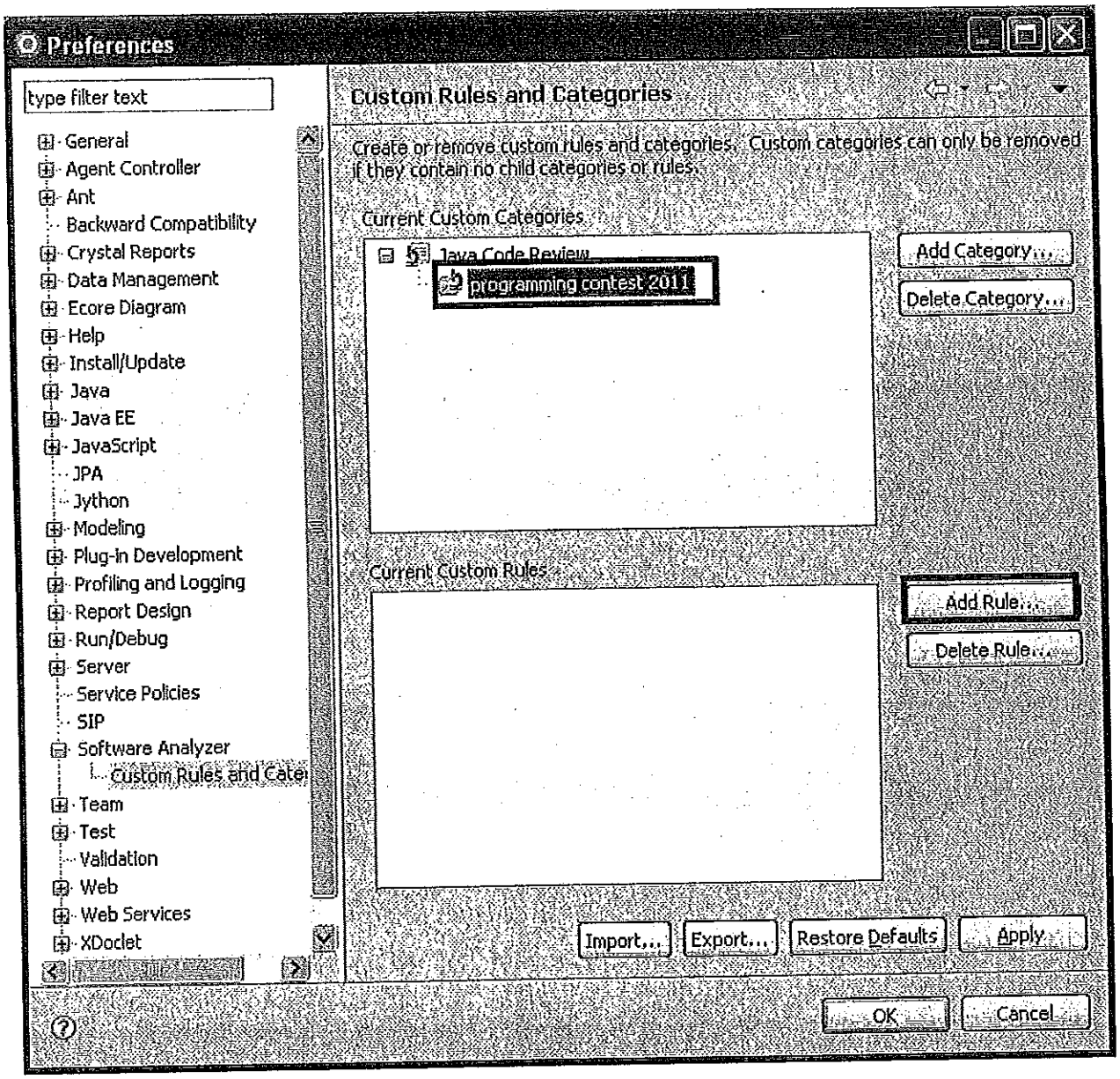
5. Choose Parent Category Location as "Java Code Review"



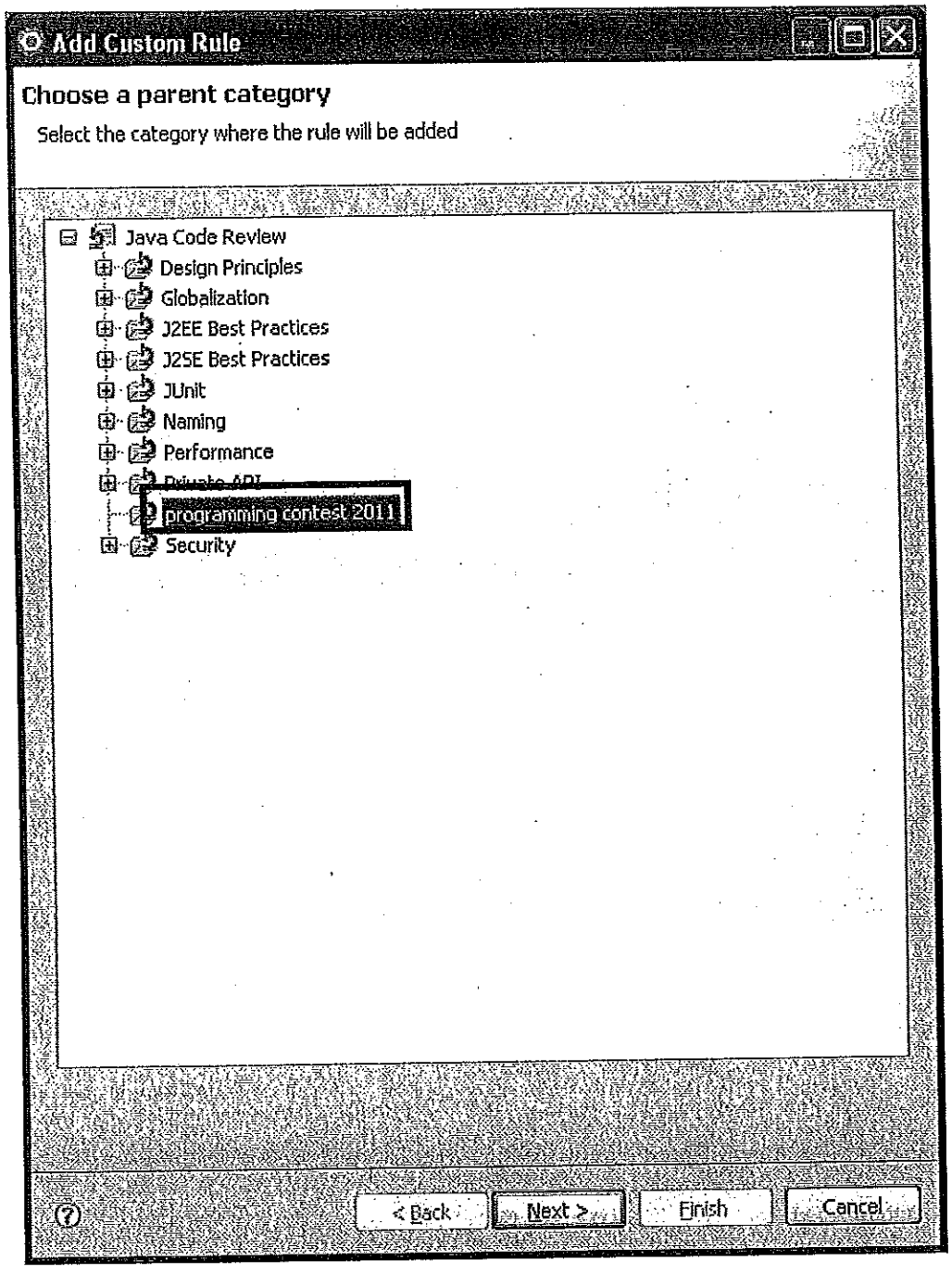
6. Enter Category Name as "programming contest 2011"



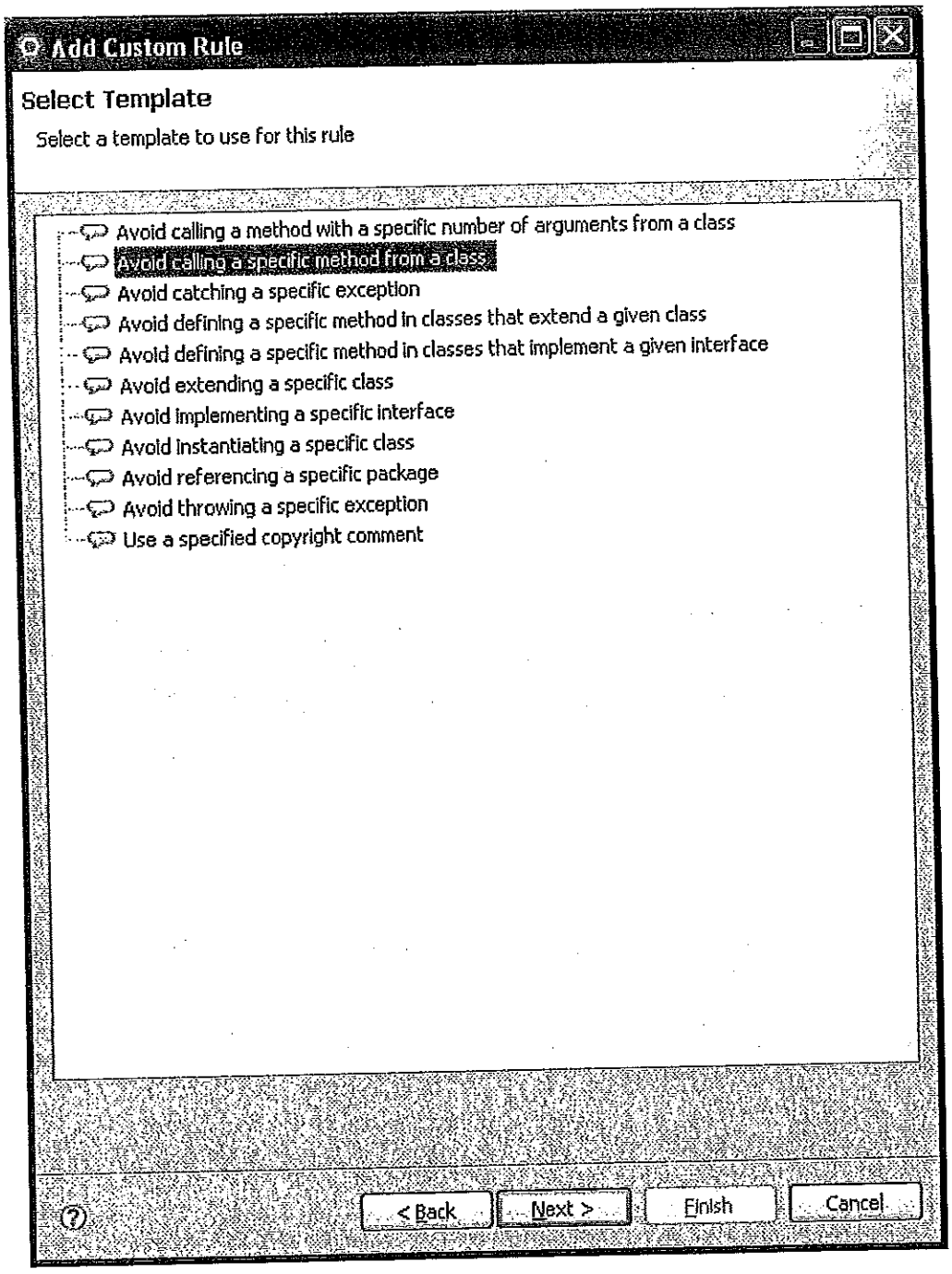
7. Confirm the category created, and "Add Rule..."



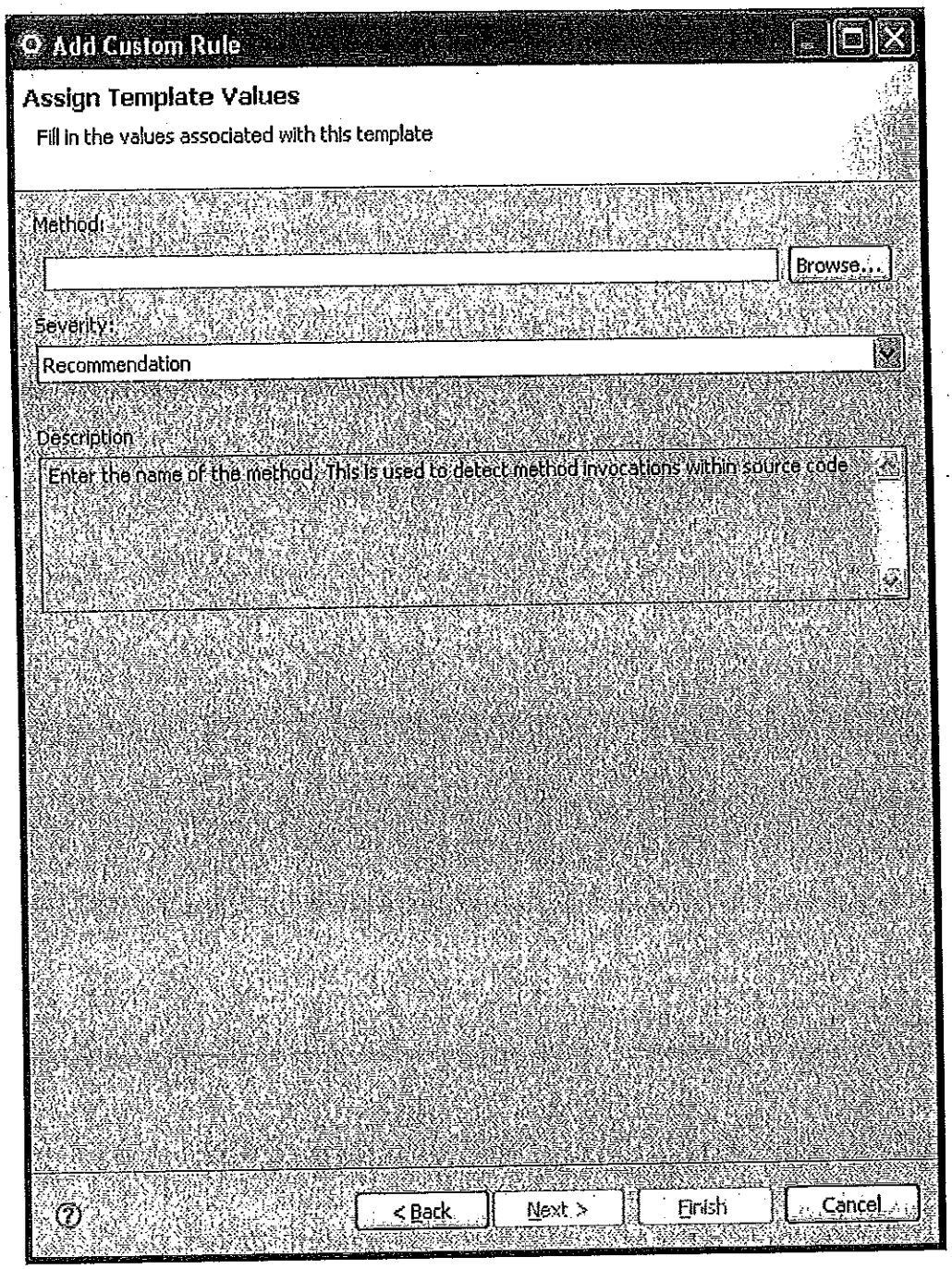
8. Select the parent category as "programming contest 2011"



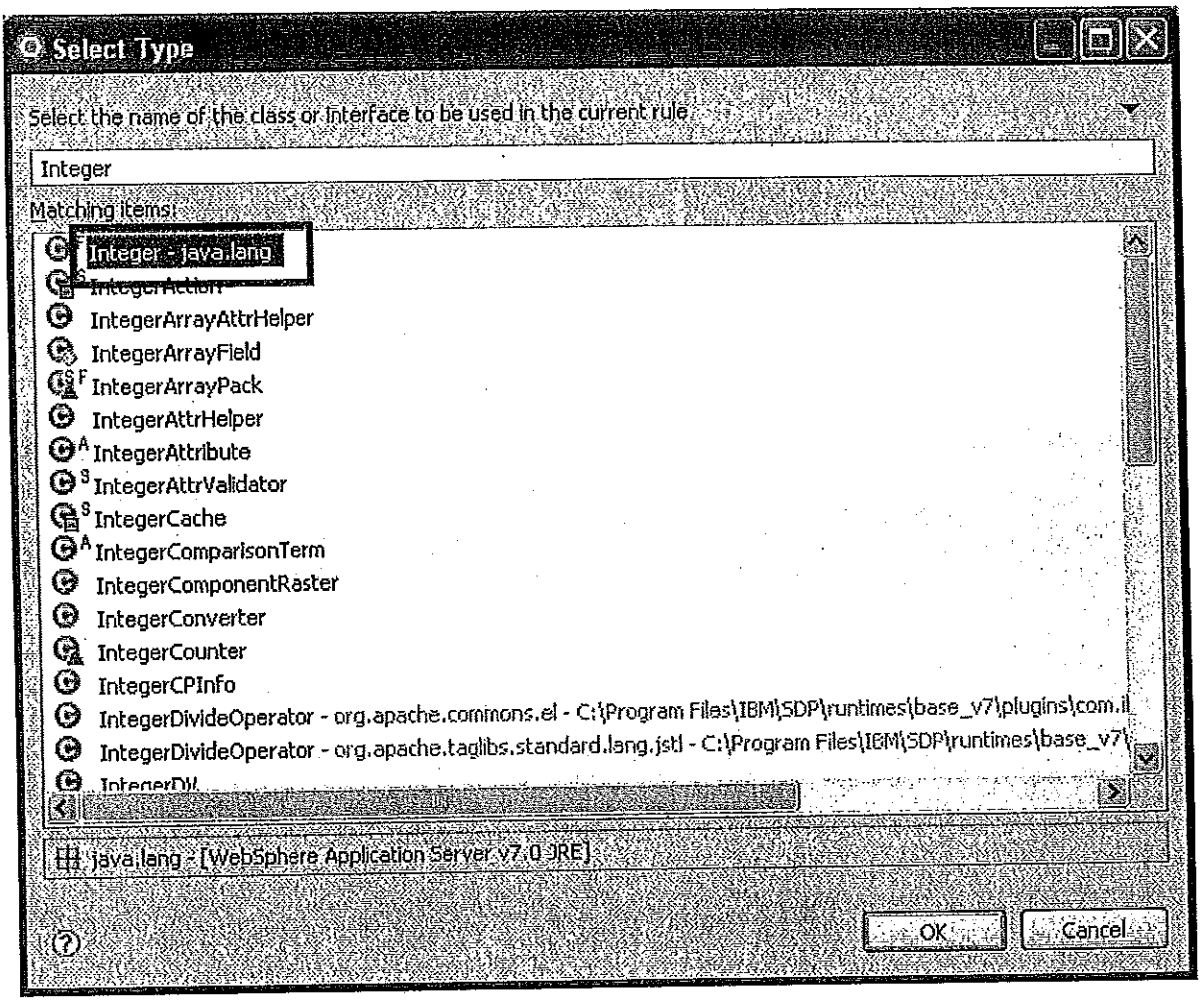
9: select template as "Avoid calling a specific method from a class"



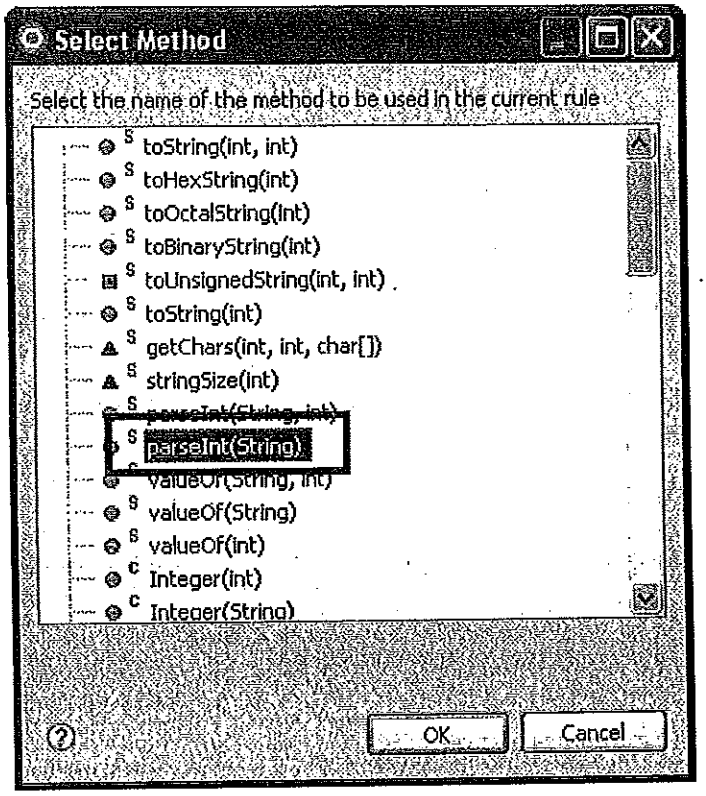
9. Assign Template Values



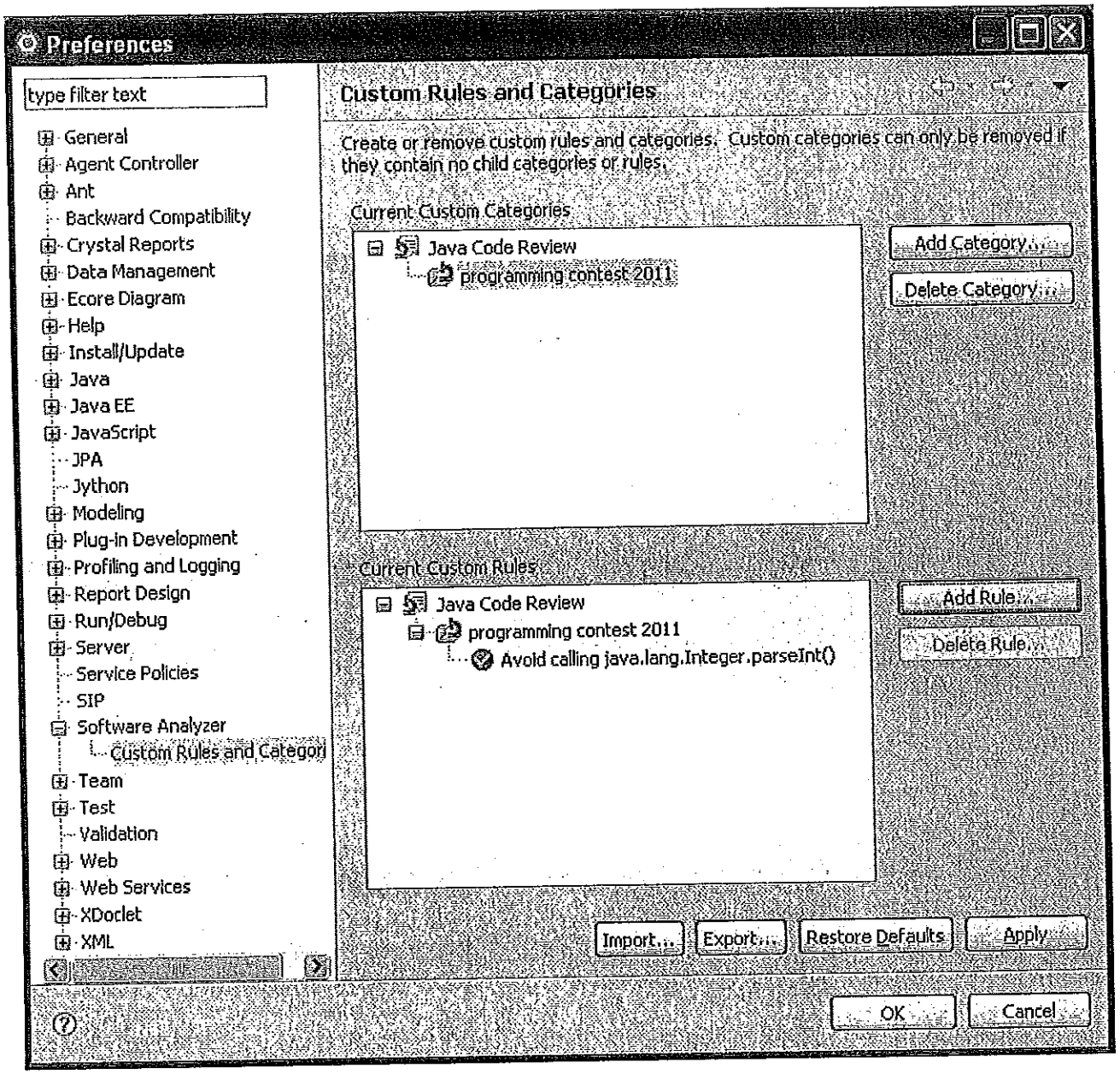
10. Type "Integer" to select matching class.



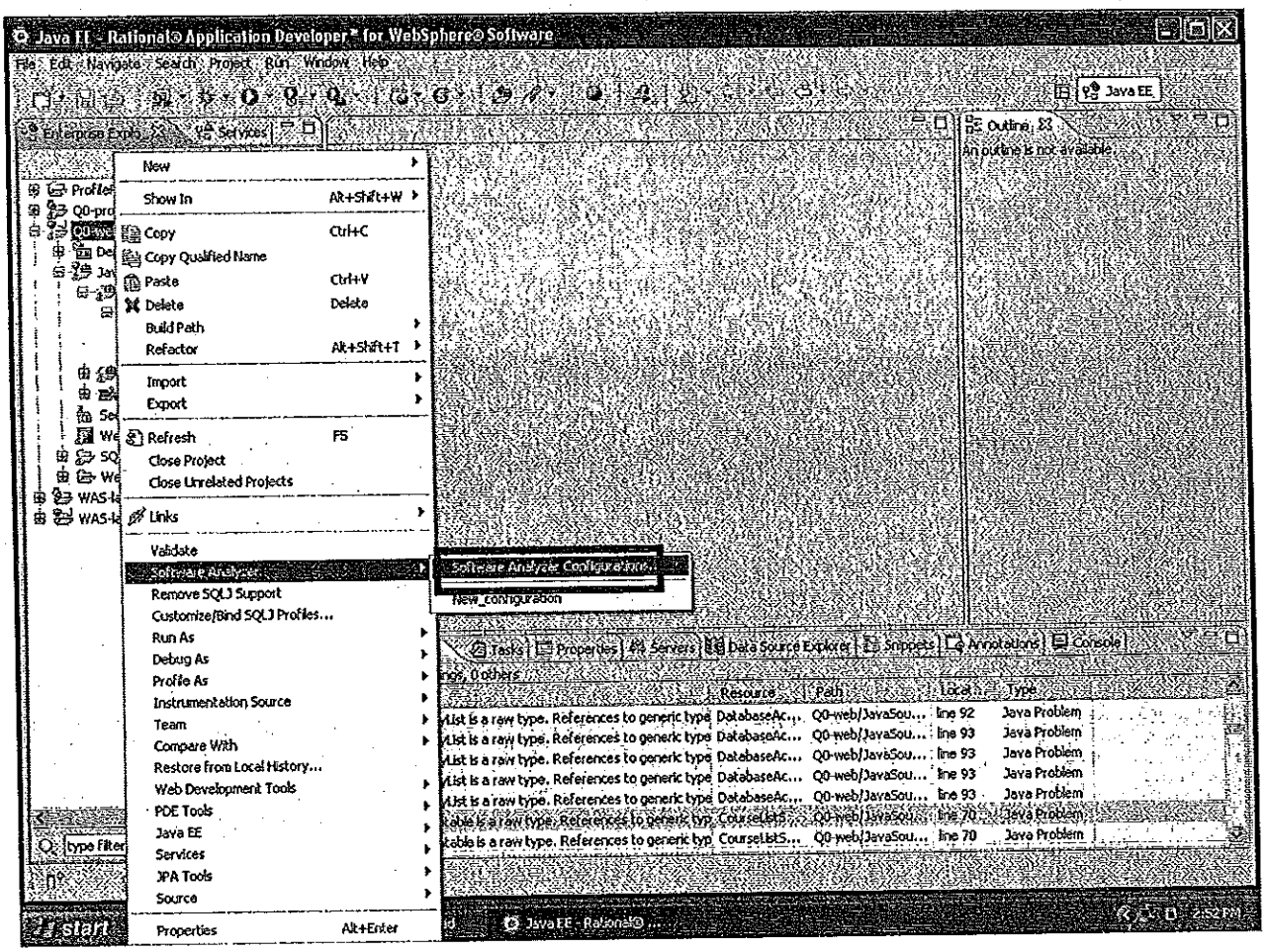
11. Select parseInt(String) as match method.



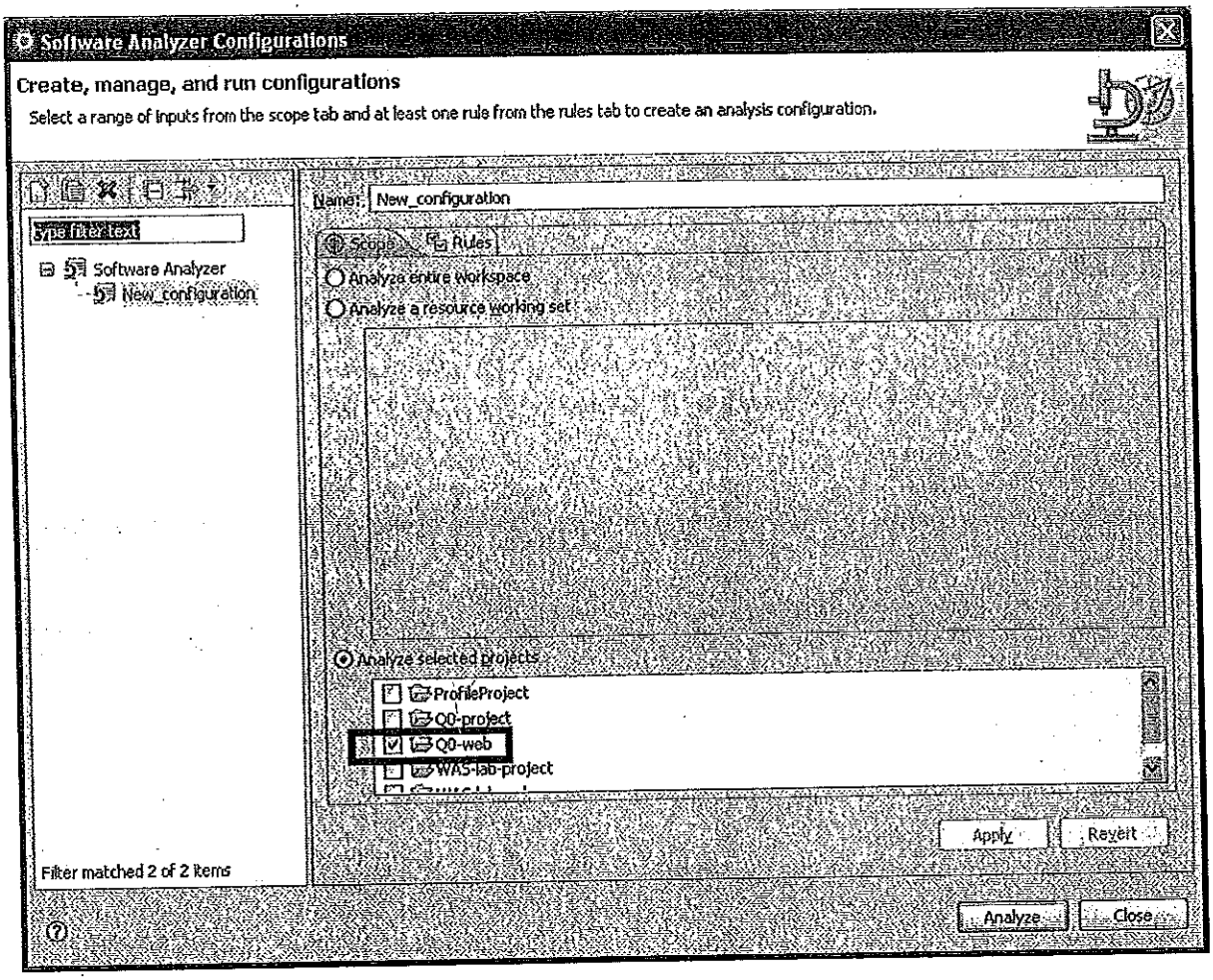
12, confirm the custom rules as follow.



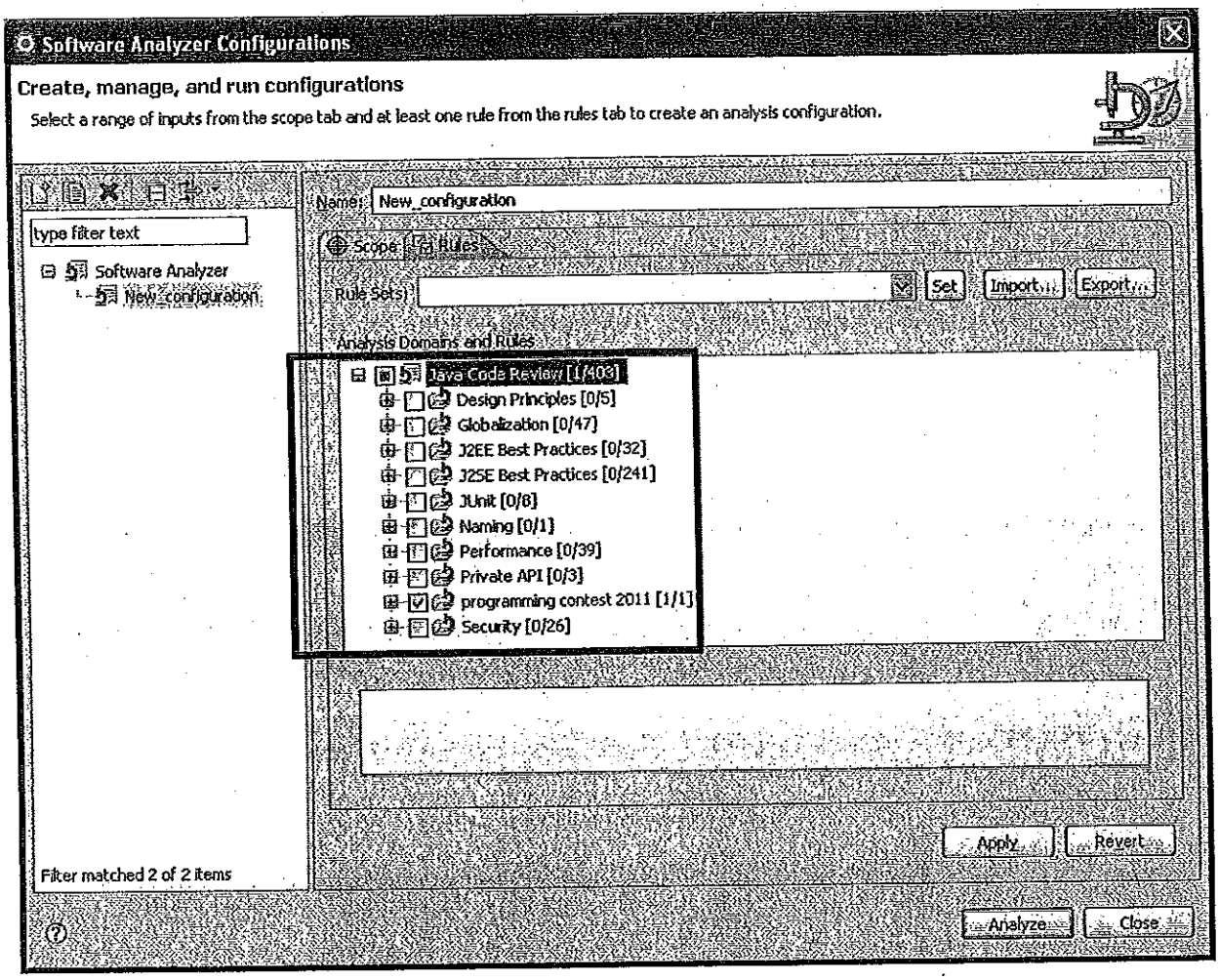
13. Right click the project, and select Software Analyzer > Software Analyzer Configuration…



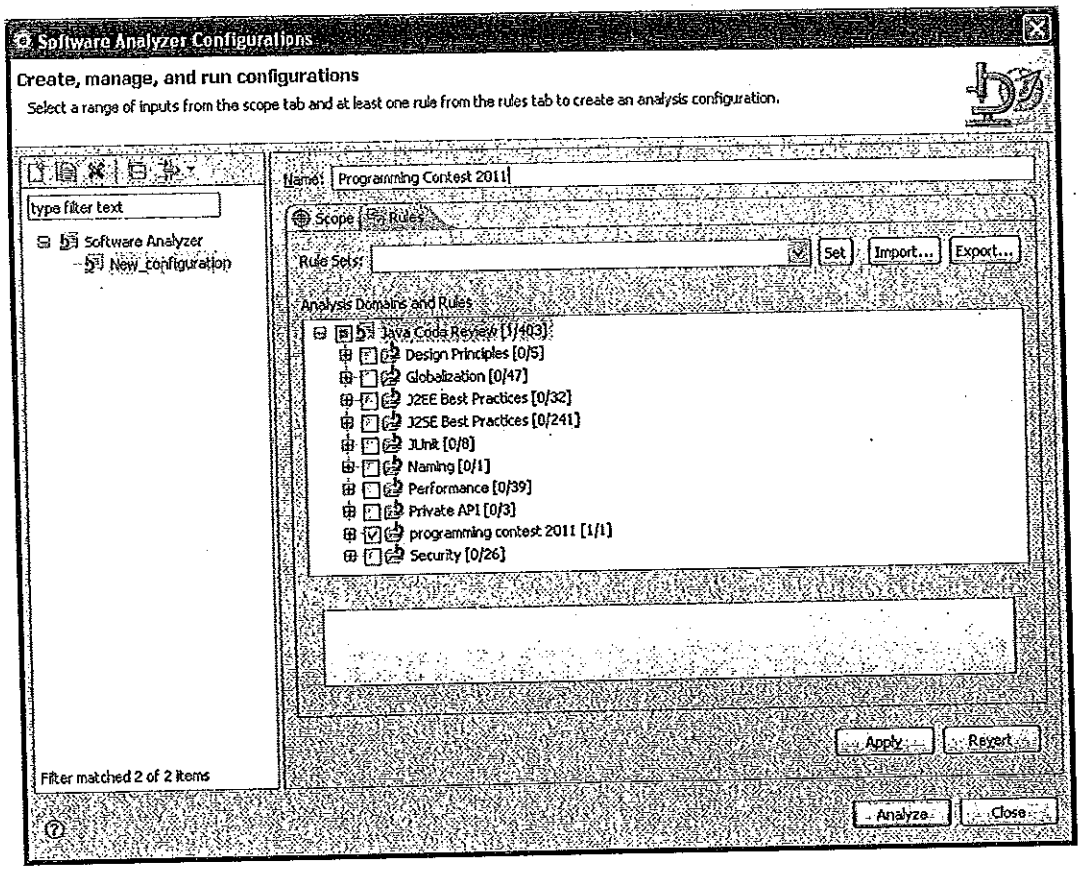
14. Select the "Q0-web" as Analyze selected projects



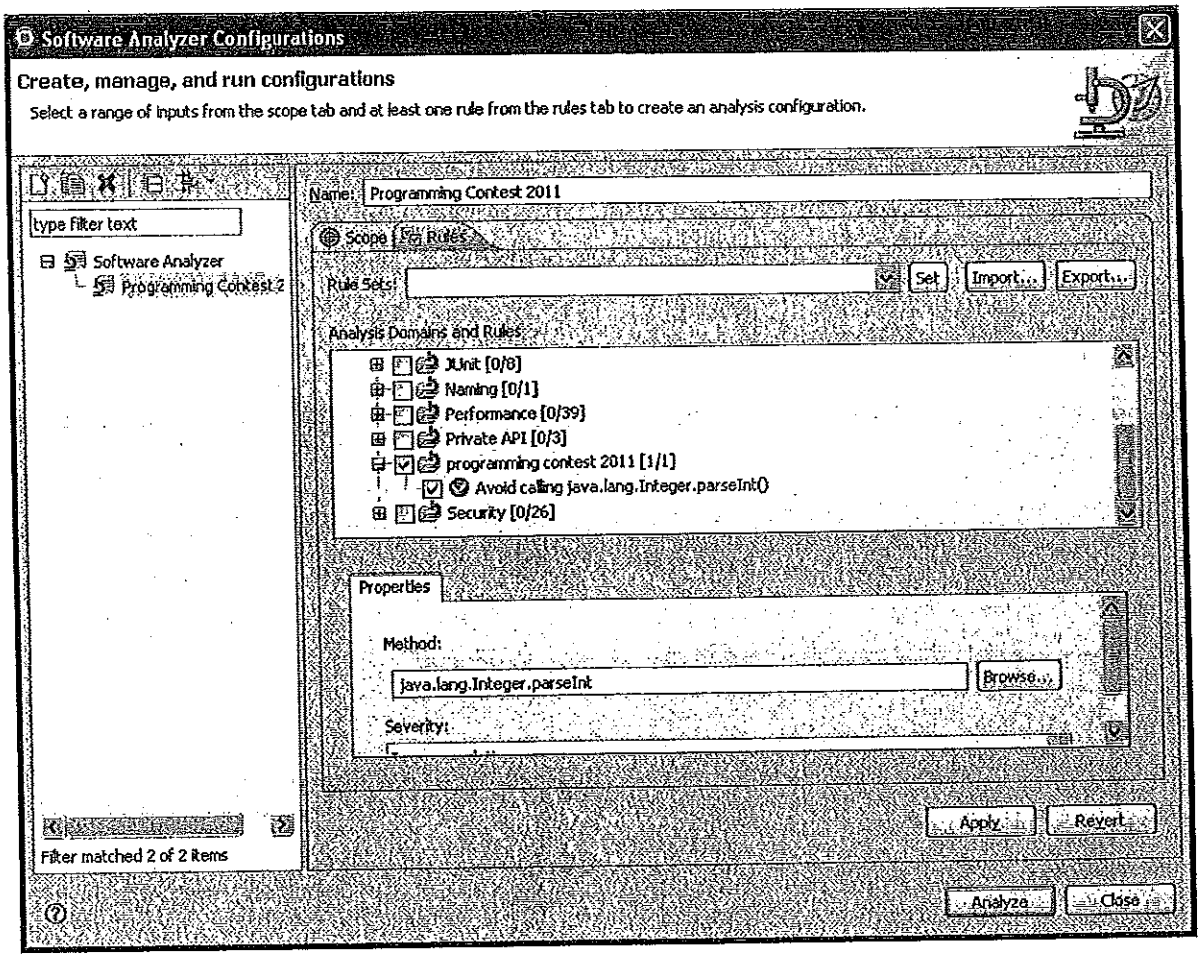
15. Clear the Checkboxs in Analysis Domains and Rules, and check "programming contest 2011"



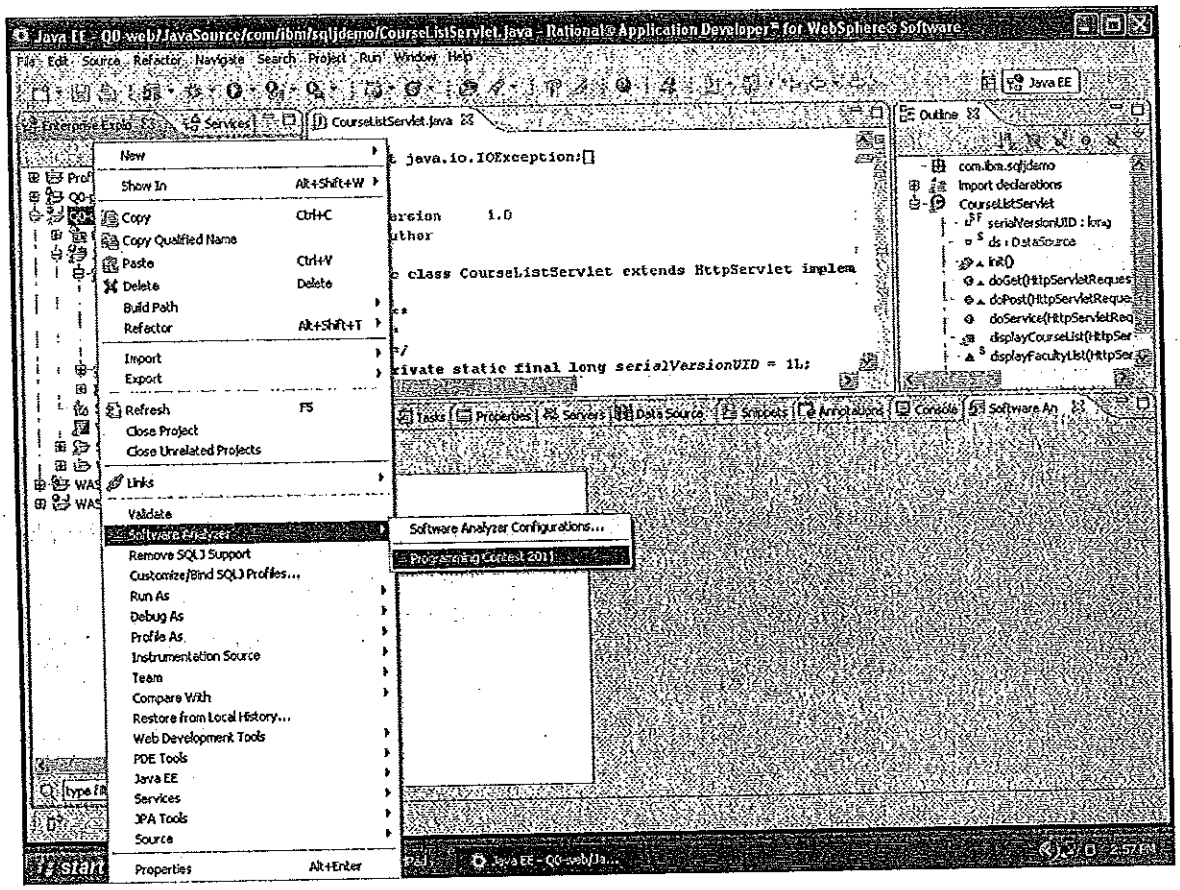
16. Rename to "Programming Contest 2011"



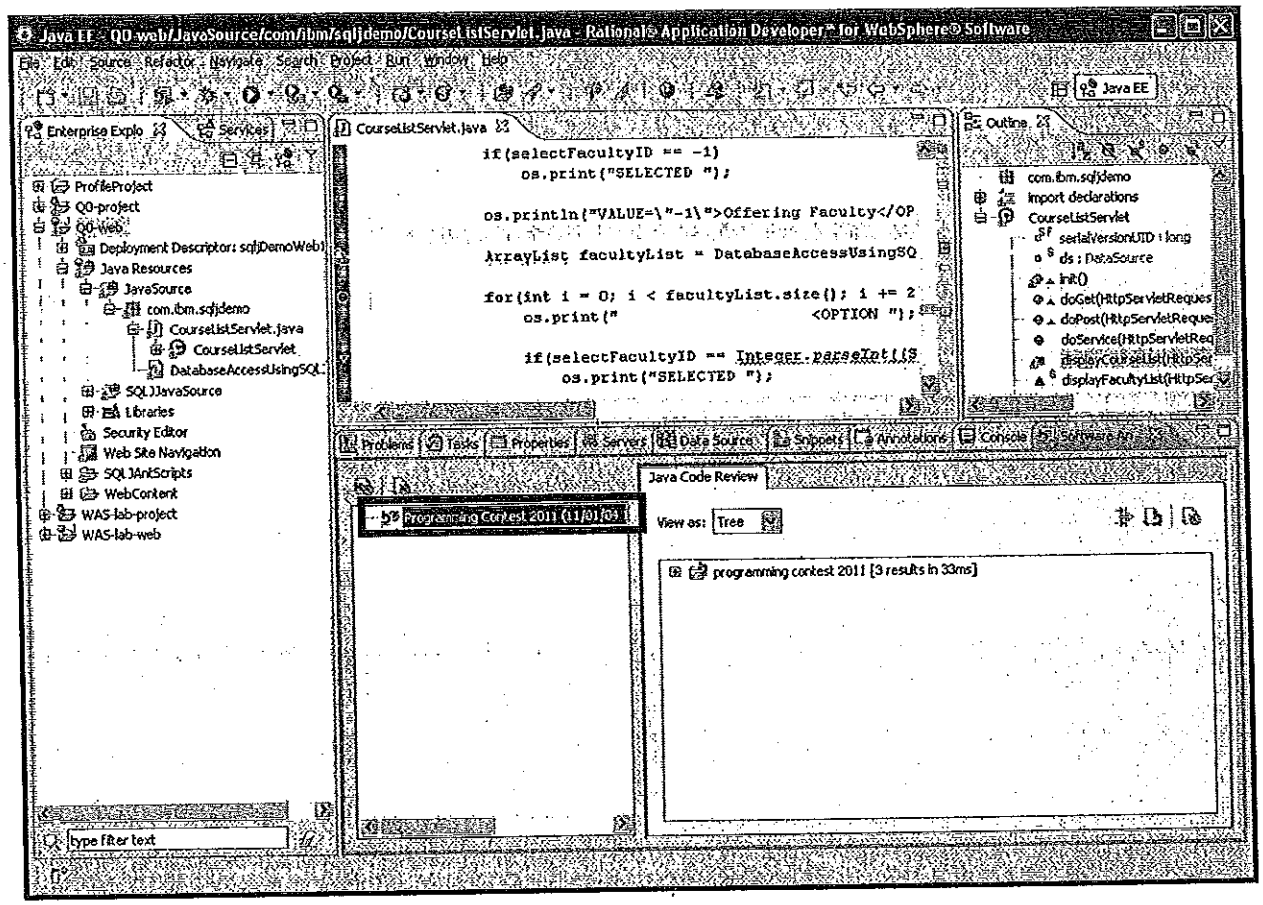
17. Check the "Avoid calling java.lang.Integer.parseInt()"



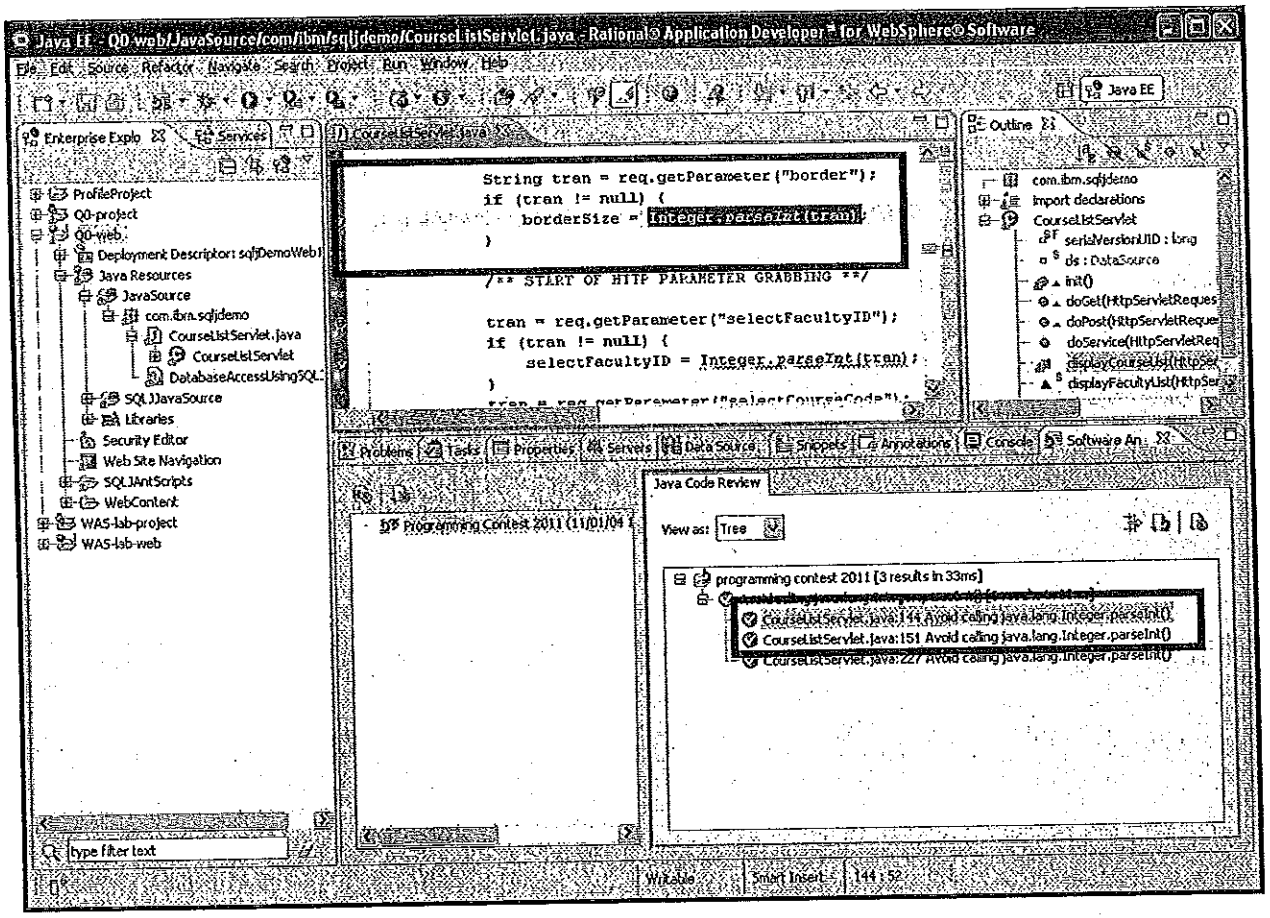
18. Run the analyzer with click Software Analyzer > Programming Contest 2011.



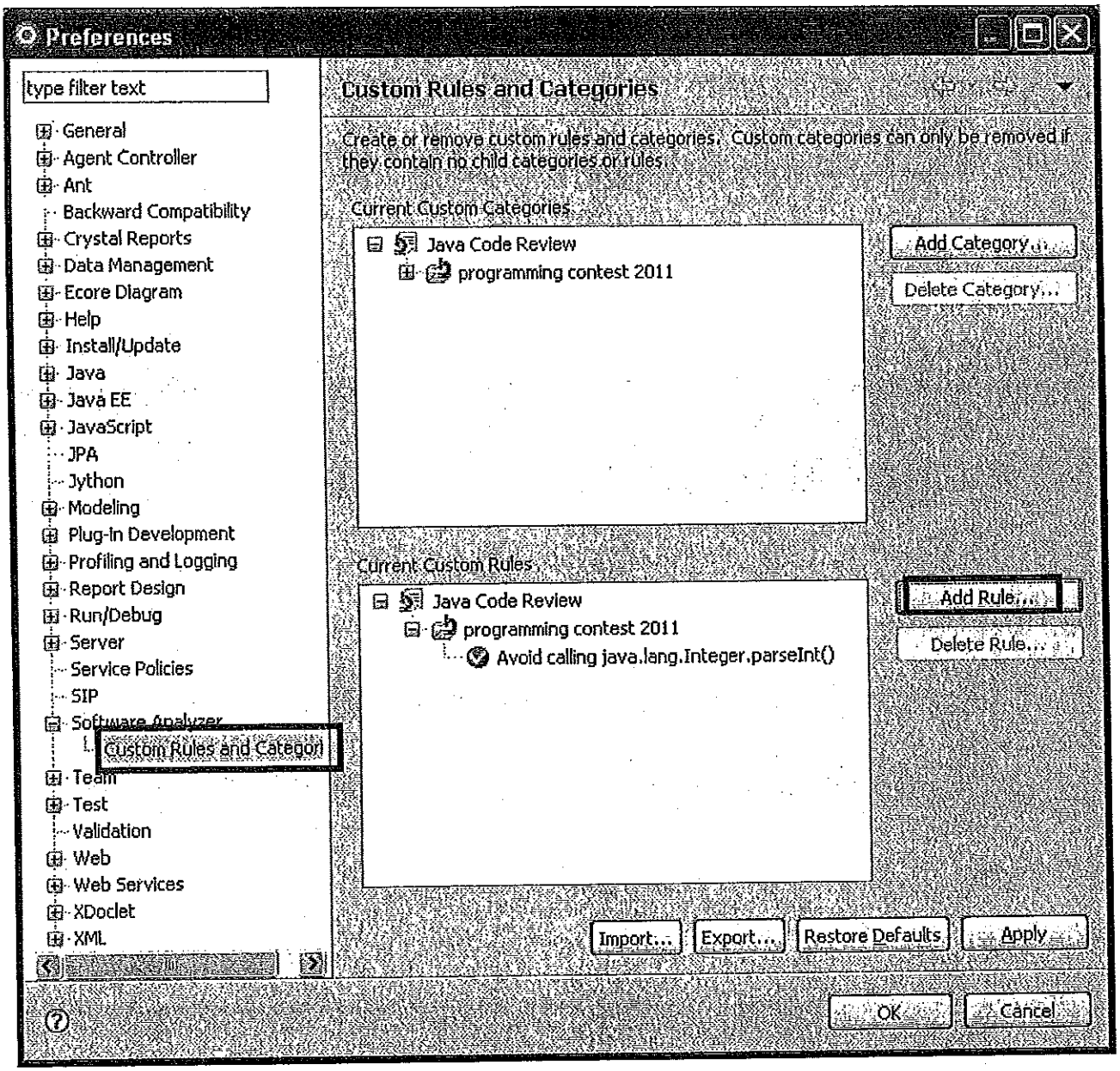
19. View the analysis result



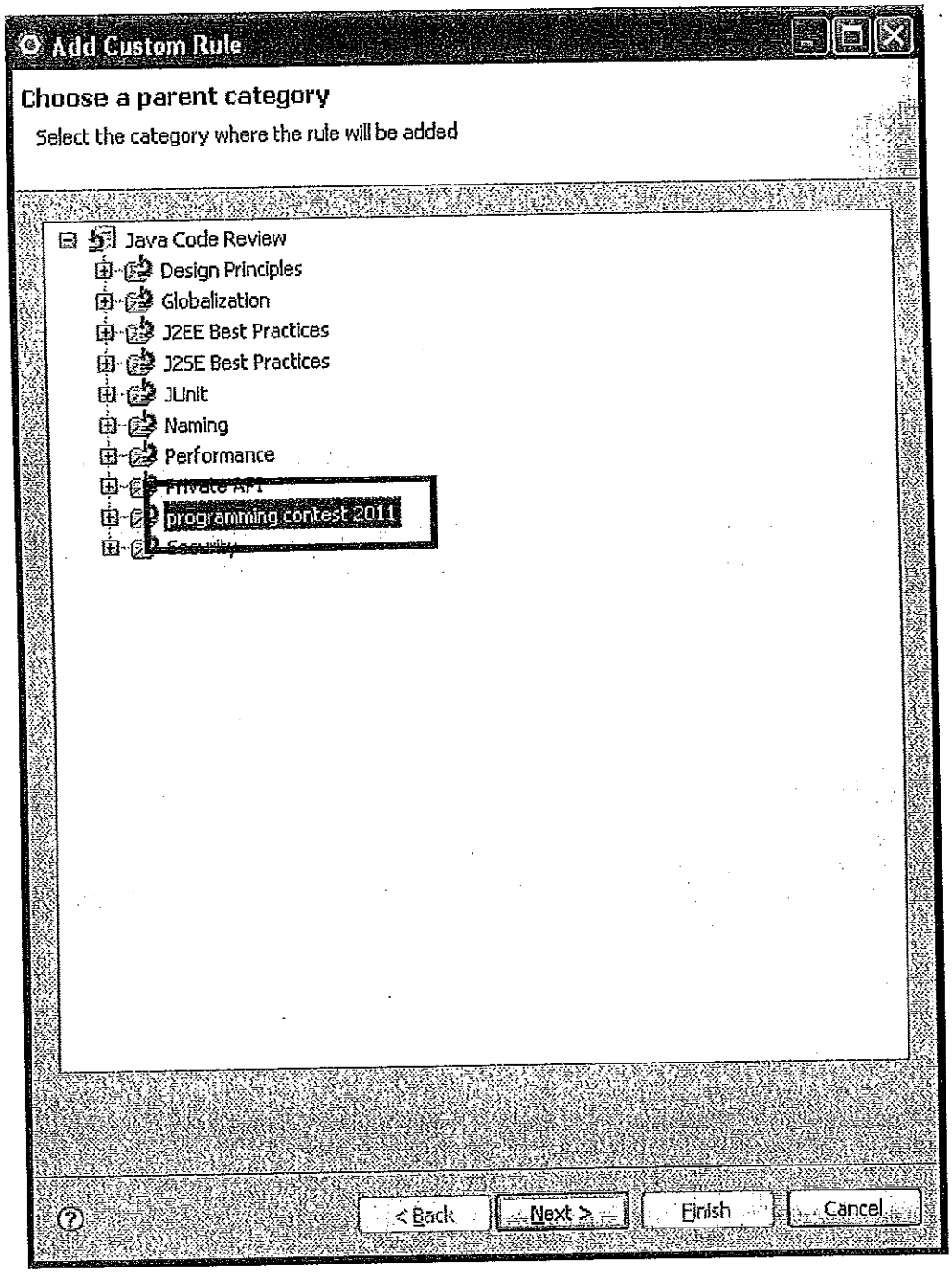
20. Click the node in Java Code Review result pane, will locate to the source with matching pattern.



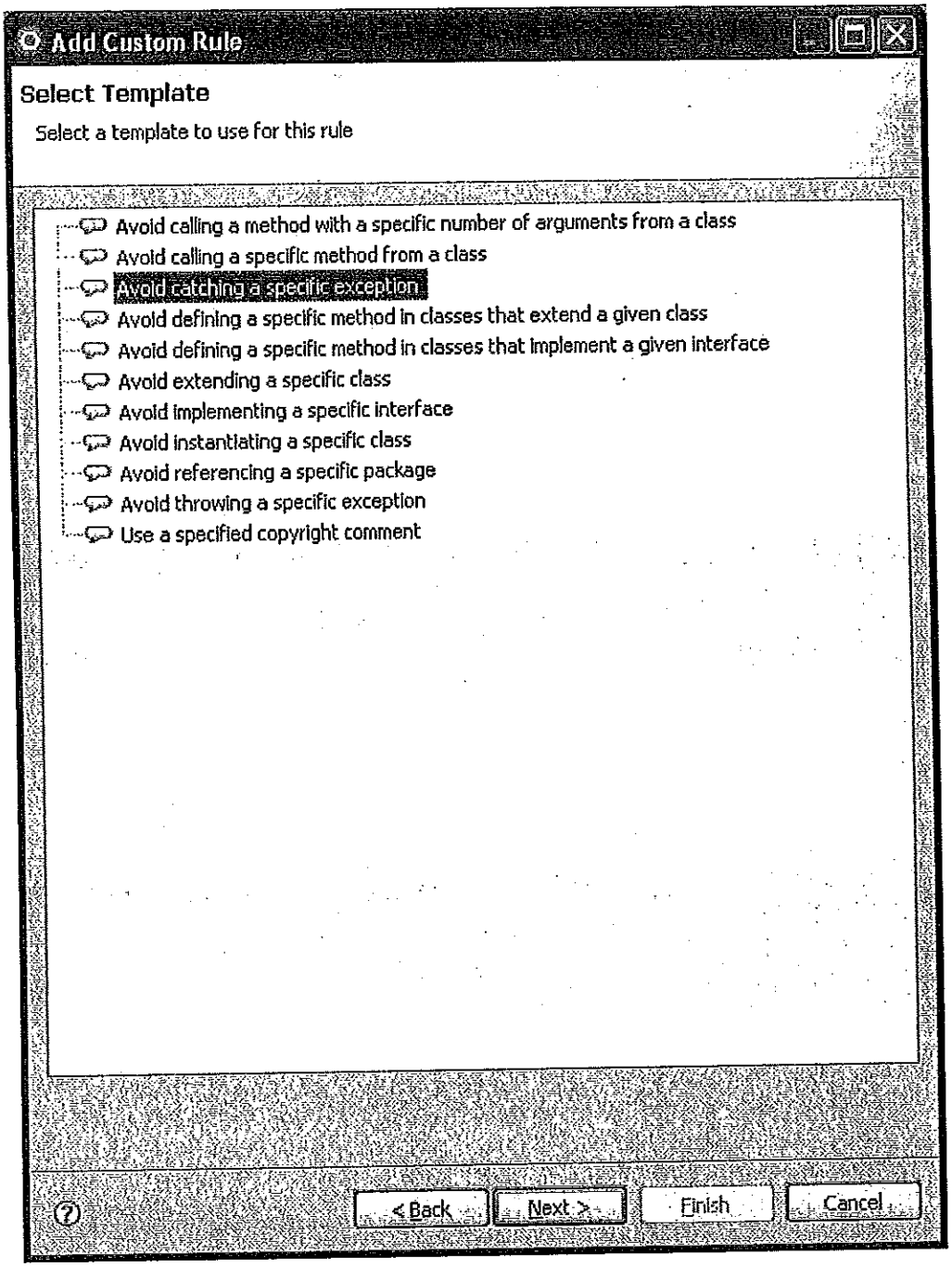
21. In Window > Preferences, select custom rules and category. click Add Rule…



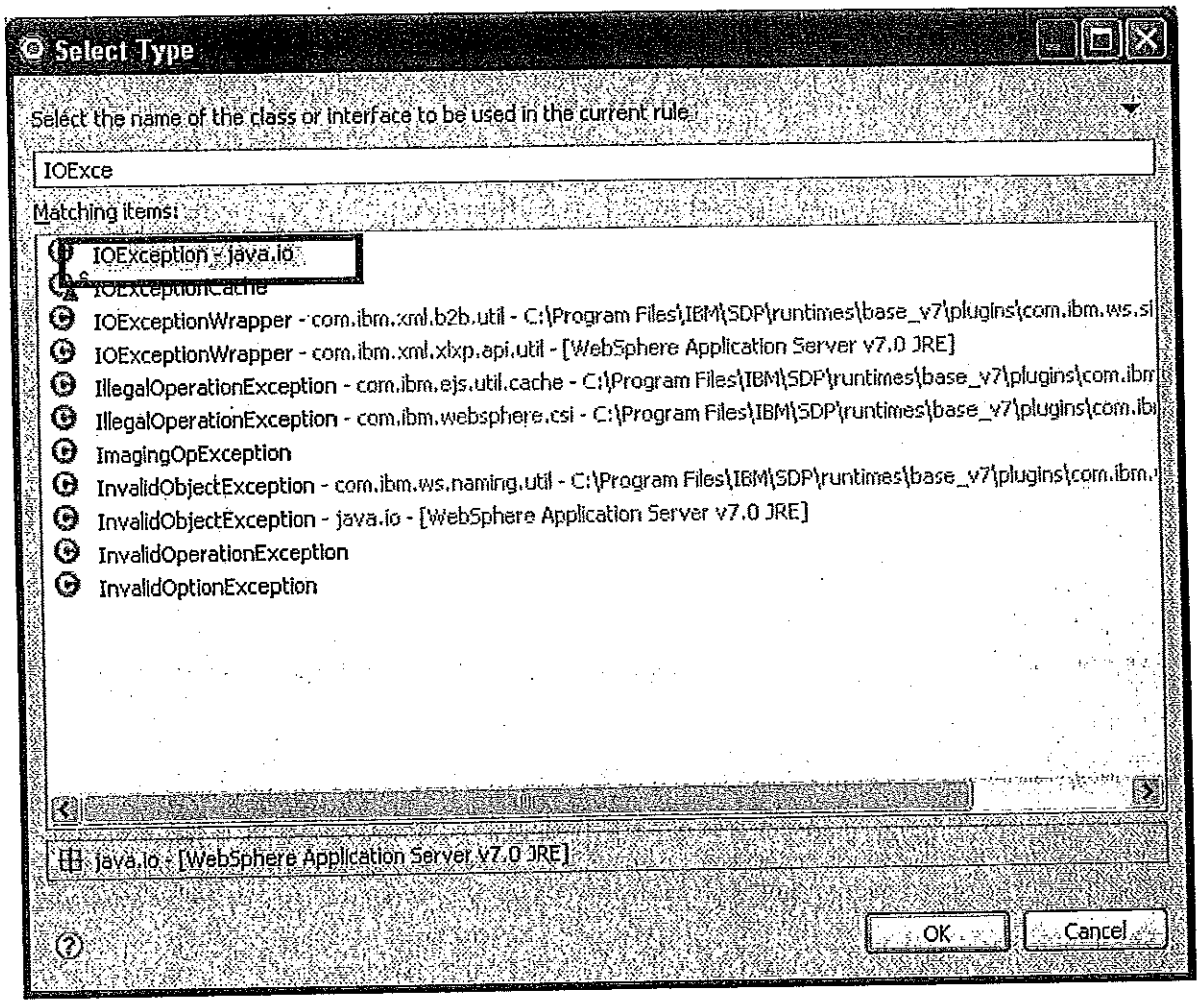
22. Select programming contest 2011 as parent category



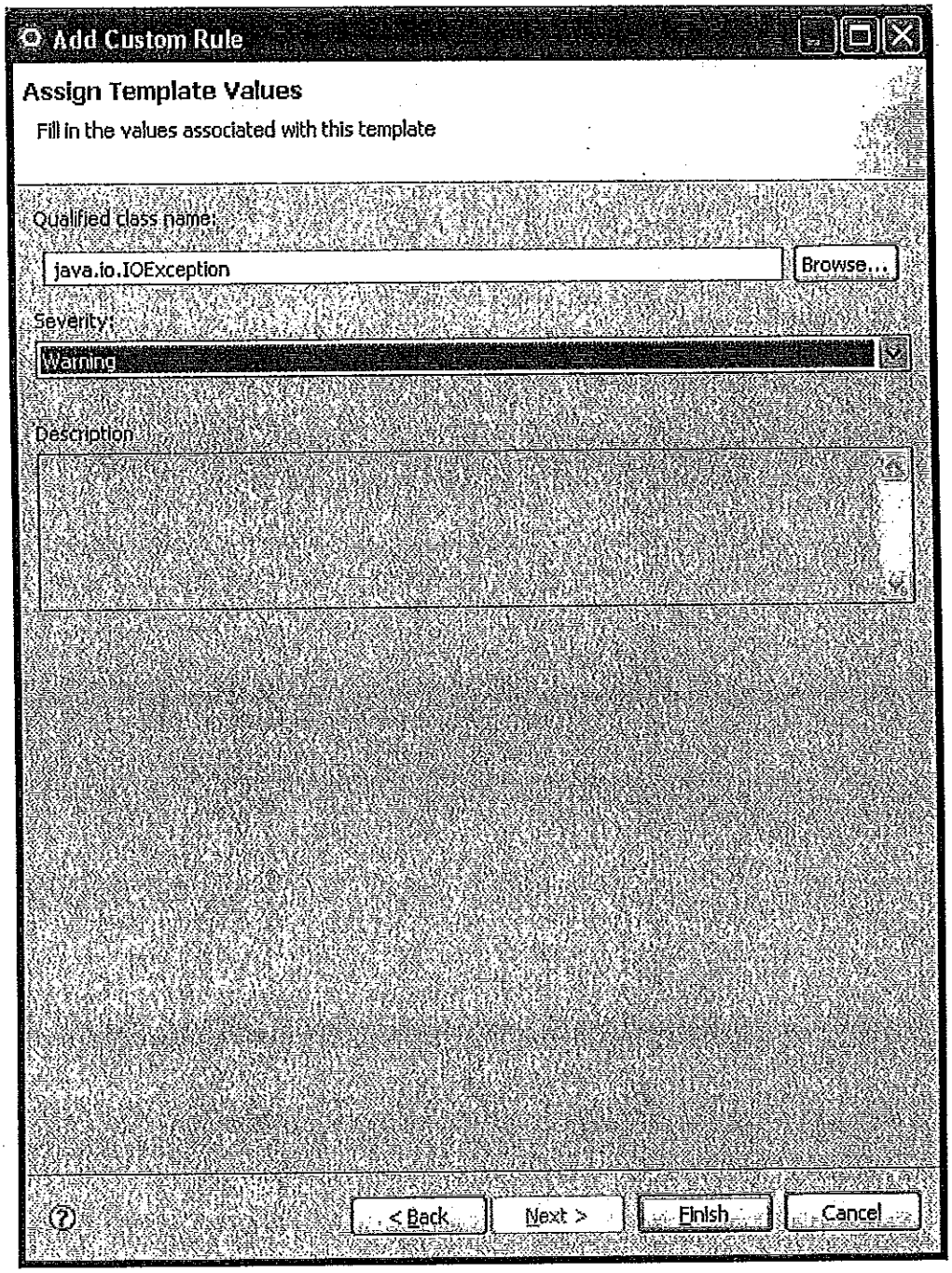
23. Select template as "Avoid catching a specific exception"



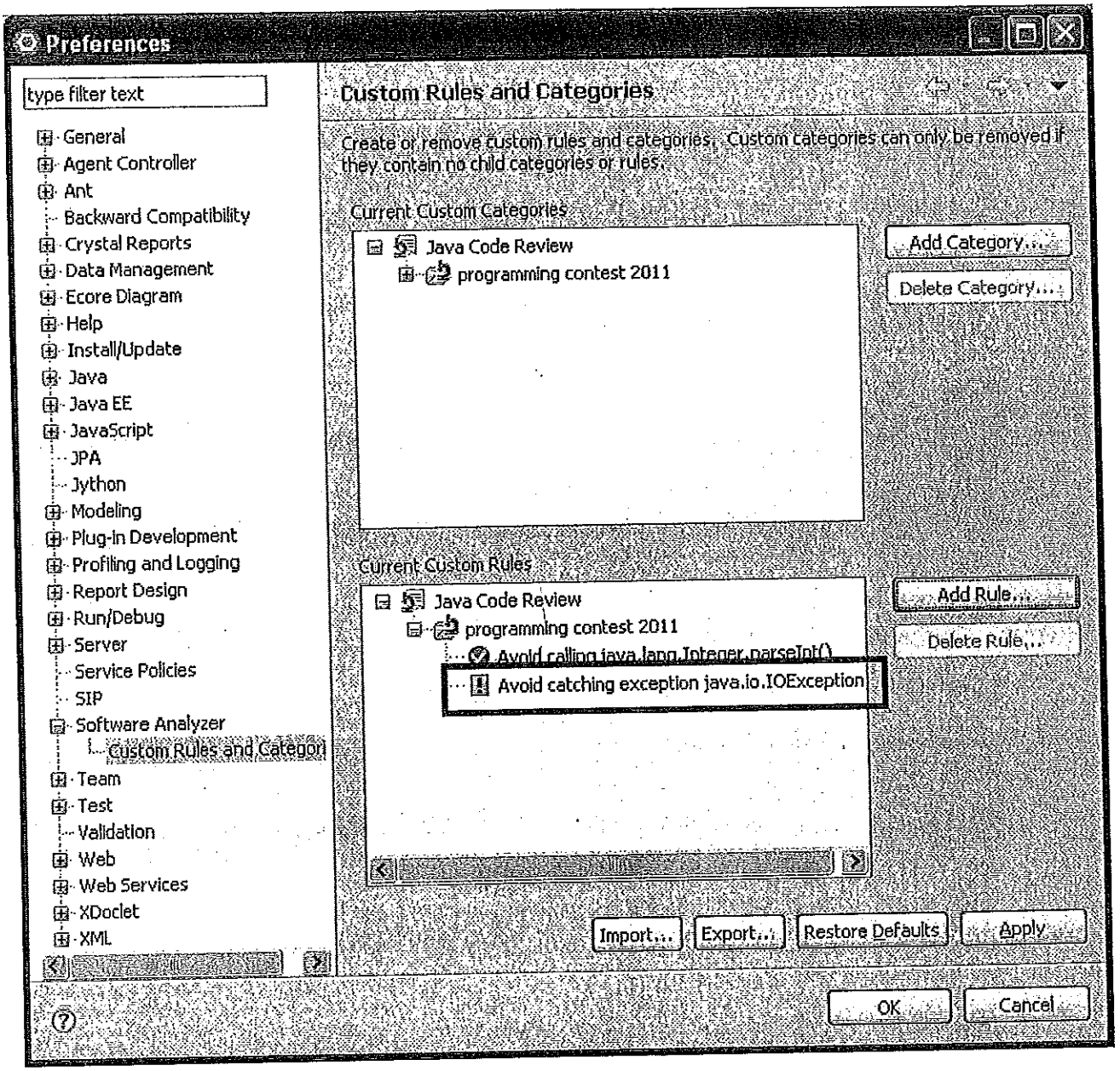
24. Look up IOException



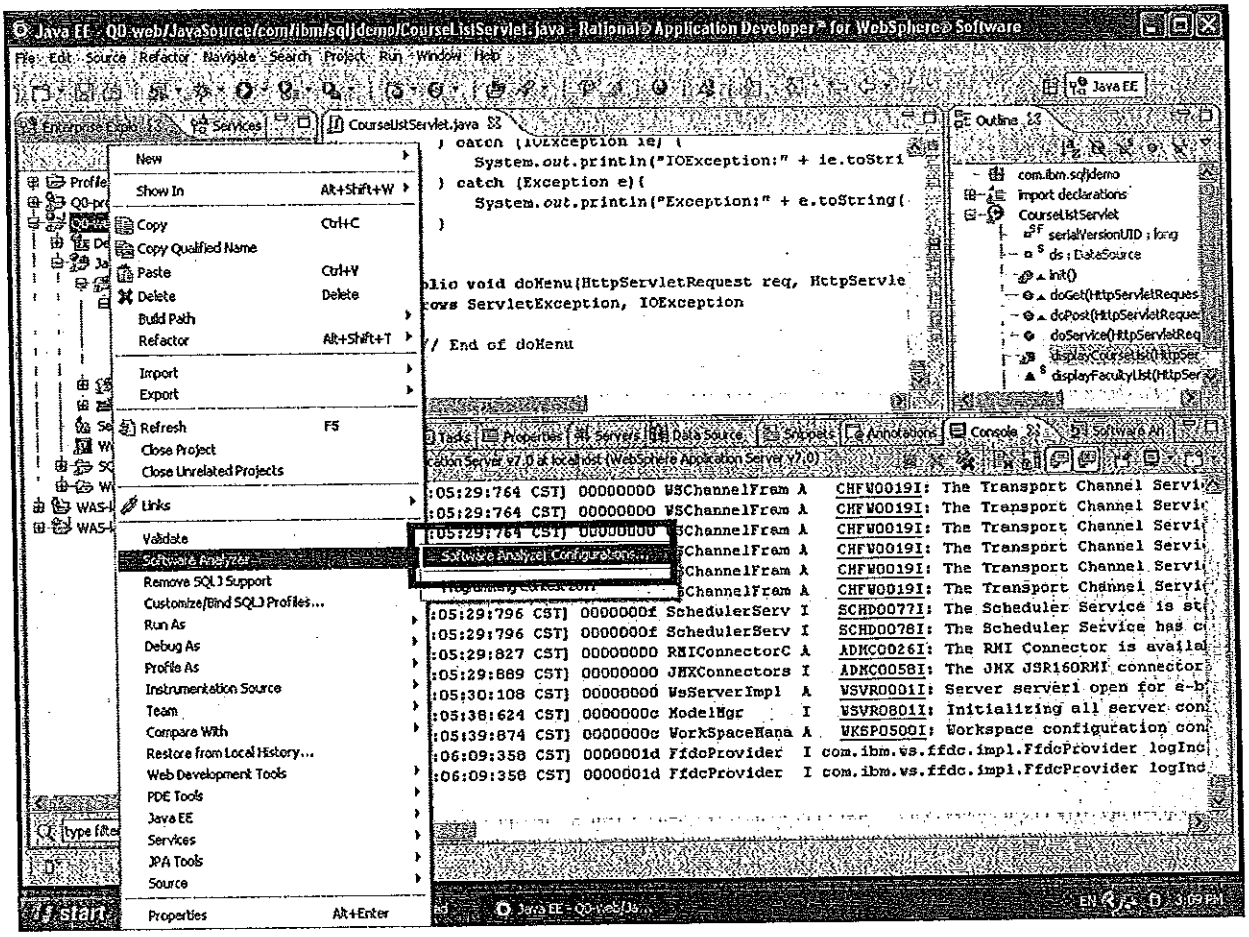
25. Change the severity to "Warning"



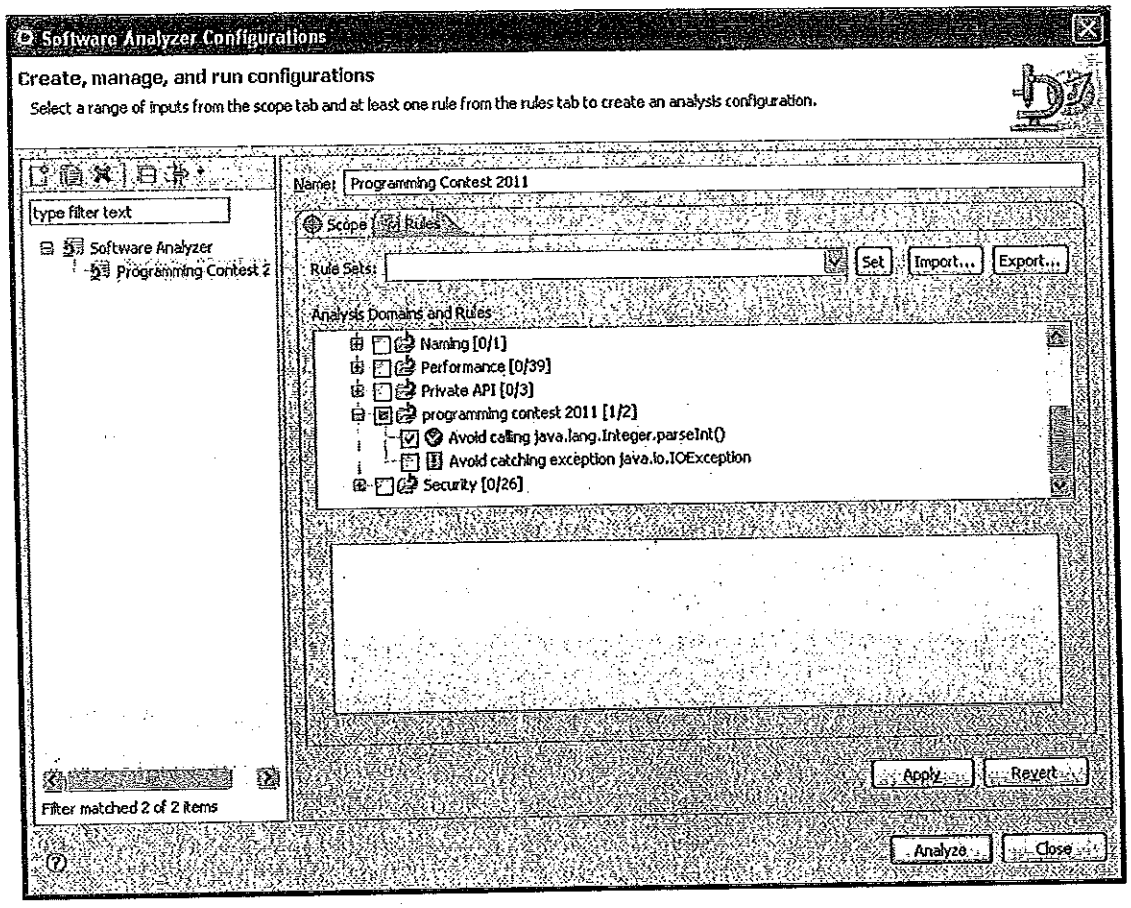
26. Another rule created.



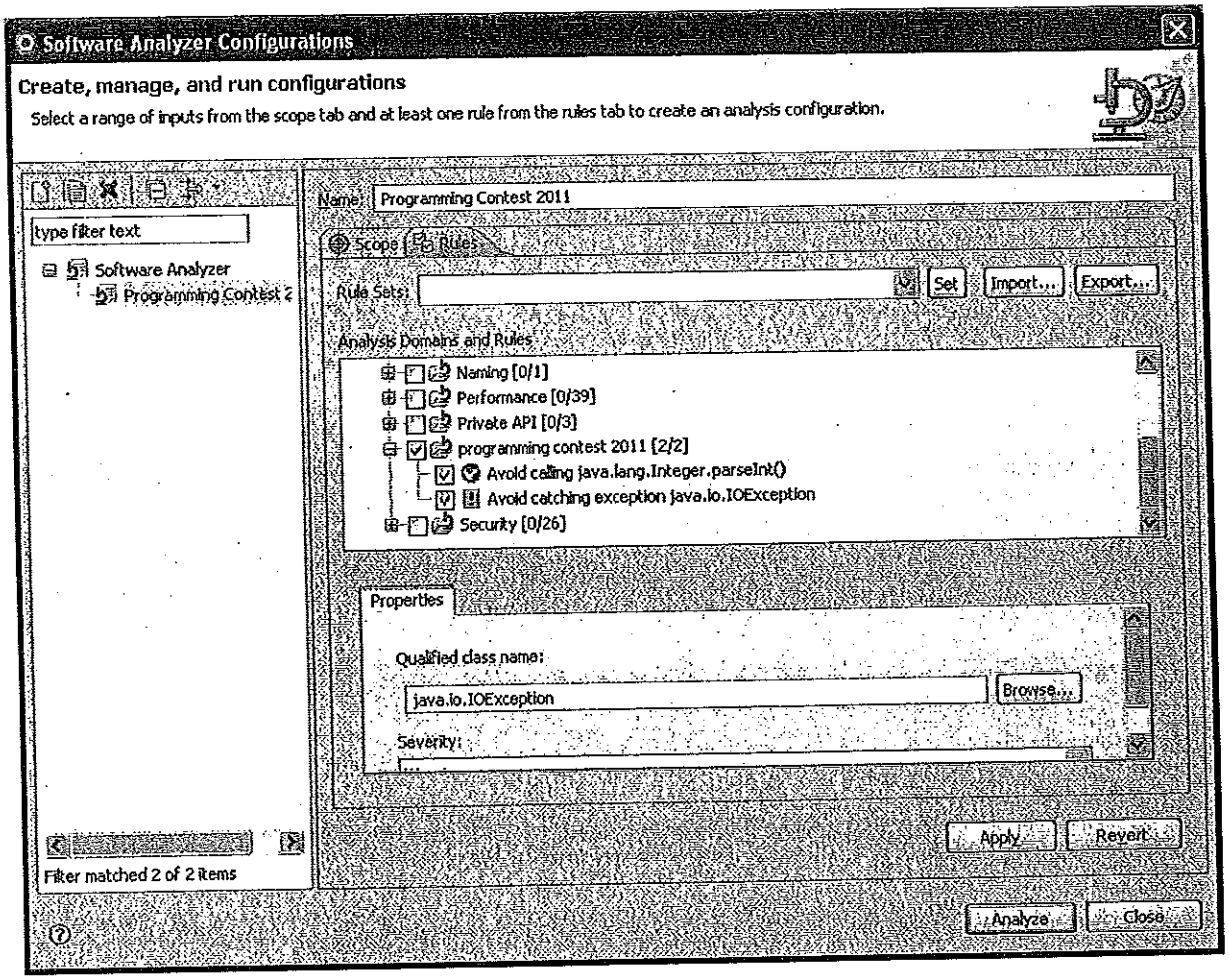
27. Config the Software Analyzer Configuration.



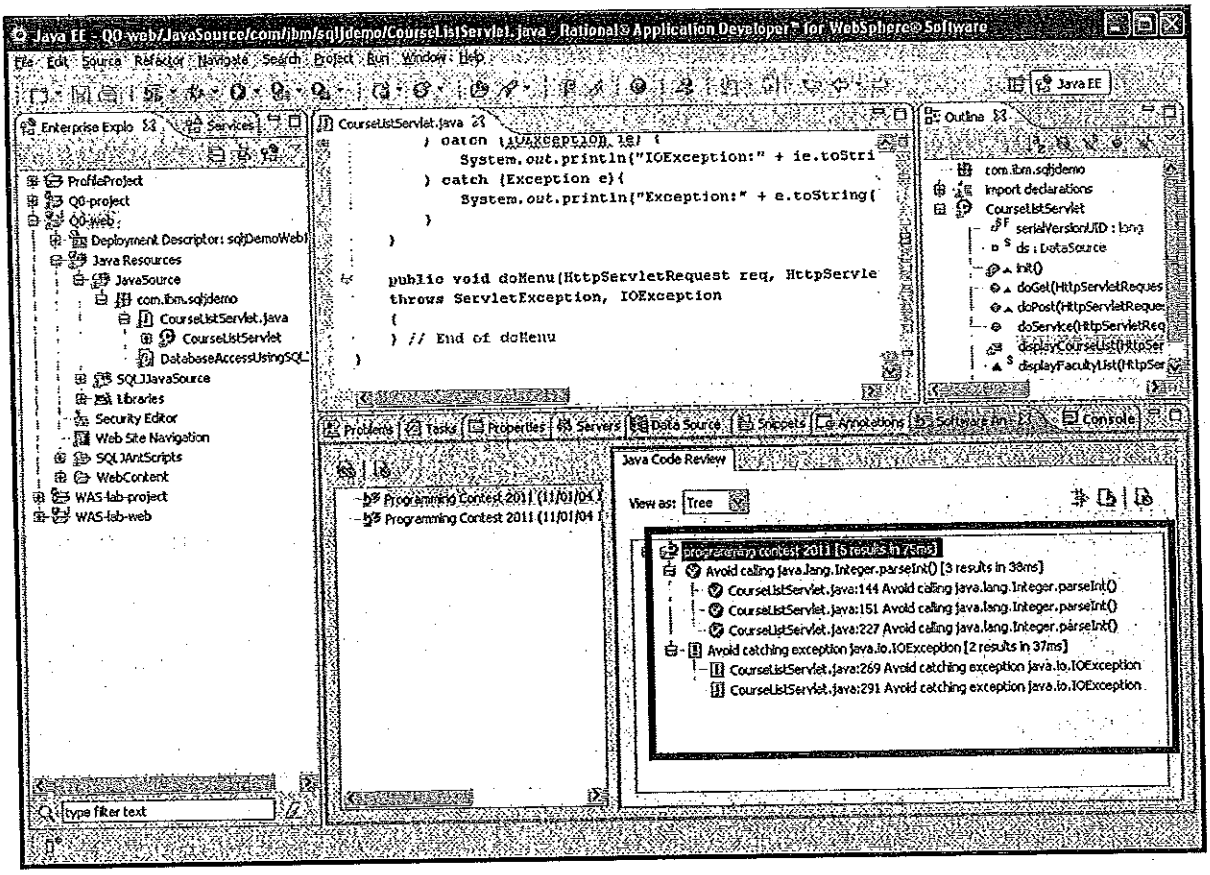
28. Goto the node programming contest 2011.



29. Check the "Avoid Catching exception java.io.IOException" and Apply.



30. Run the analyzer again, and you can see more results.

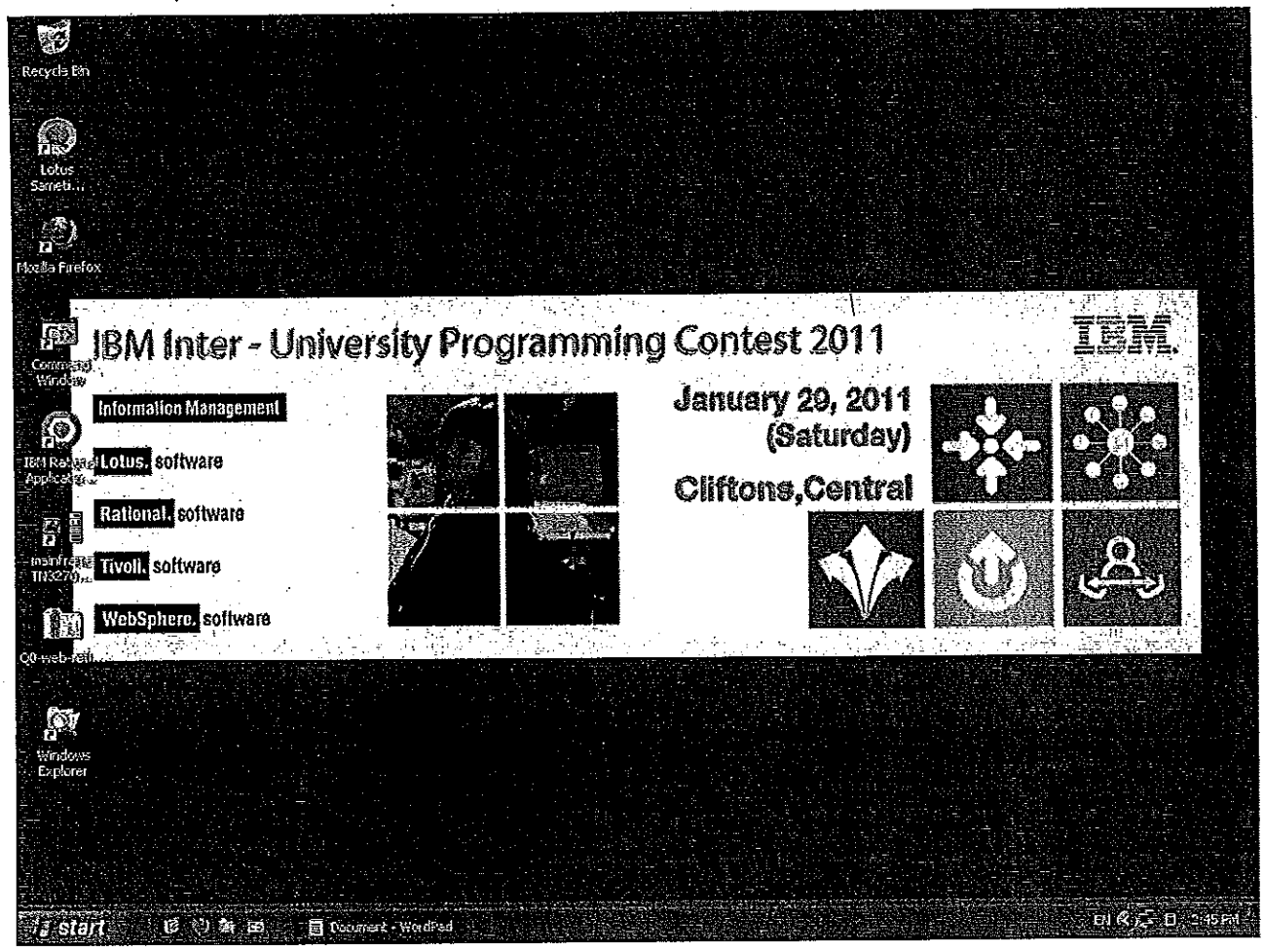


---End of Lab---

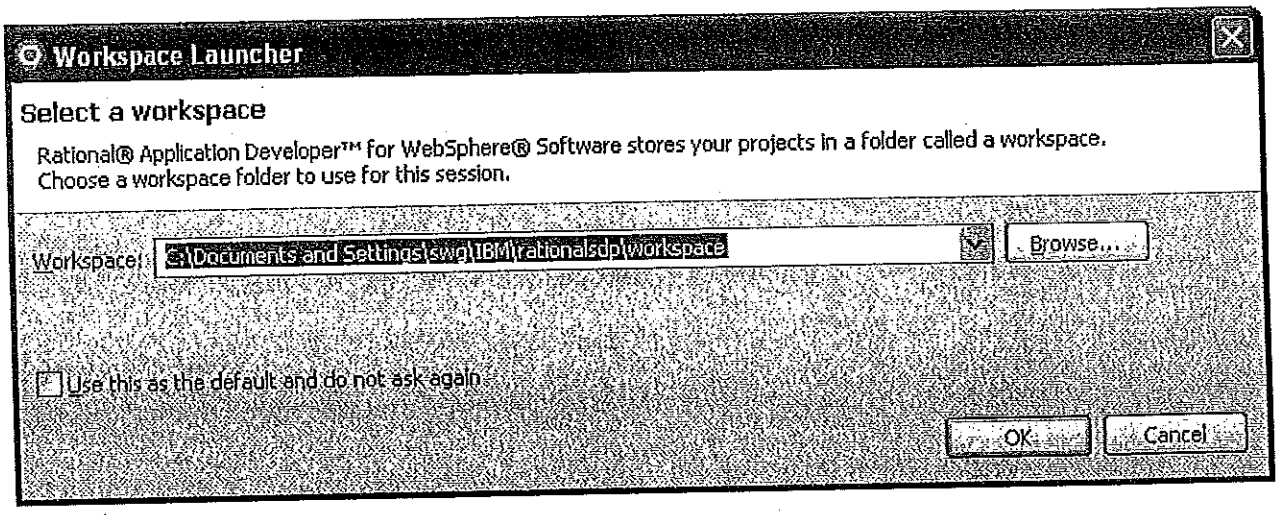
Chapter 6b: Debug using IBM Rational Application Developer

Start IBM Rational Application Developer (RAD) to develop J2EE application which will be used during the programming contest. RAD is a GUI integrated development environment (IDE) which is used to perform J2EE and web programming. The resultant programs will be published to WebSphere Application Server so that it can be executed in the web environment.

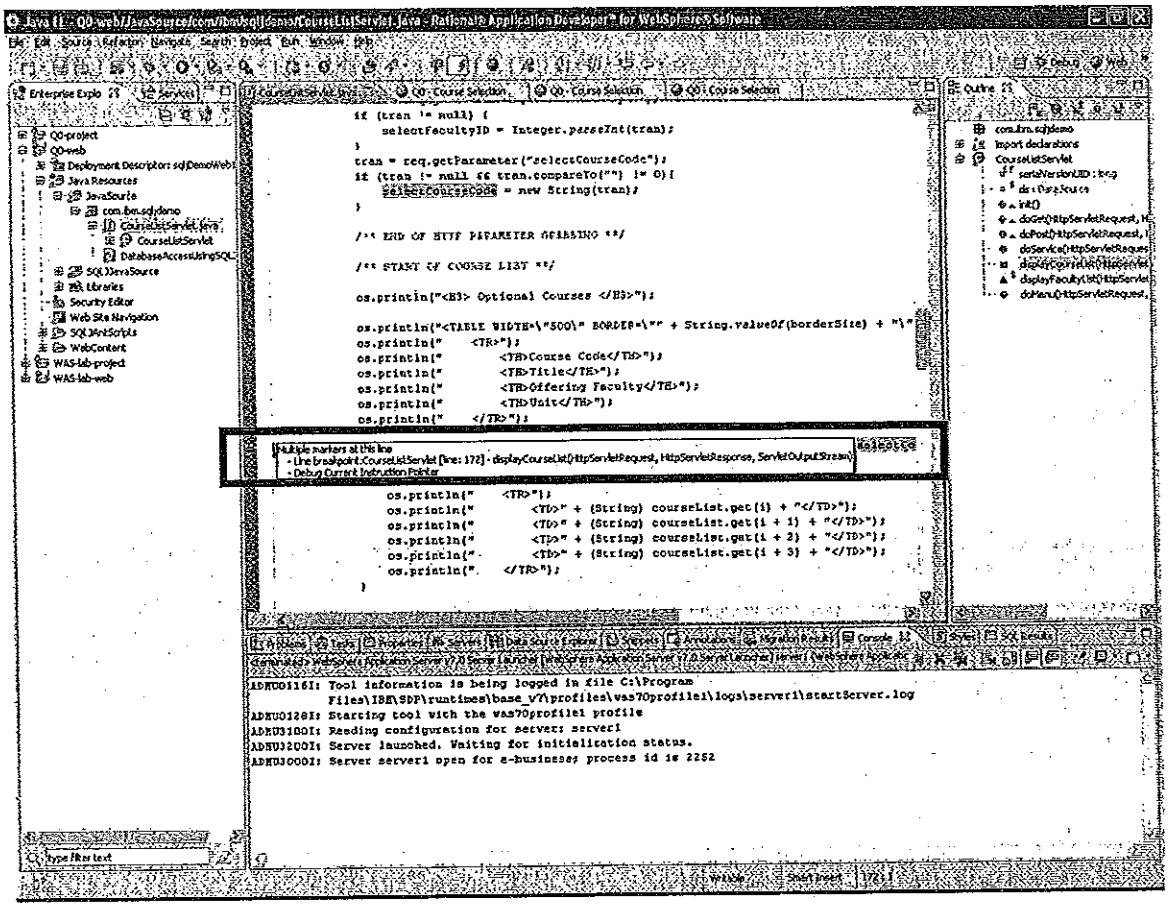
1. Double clicked the icon "IBM Rational Application Developer" on Desktop to start.



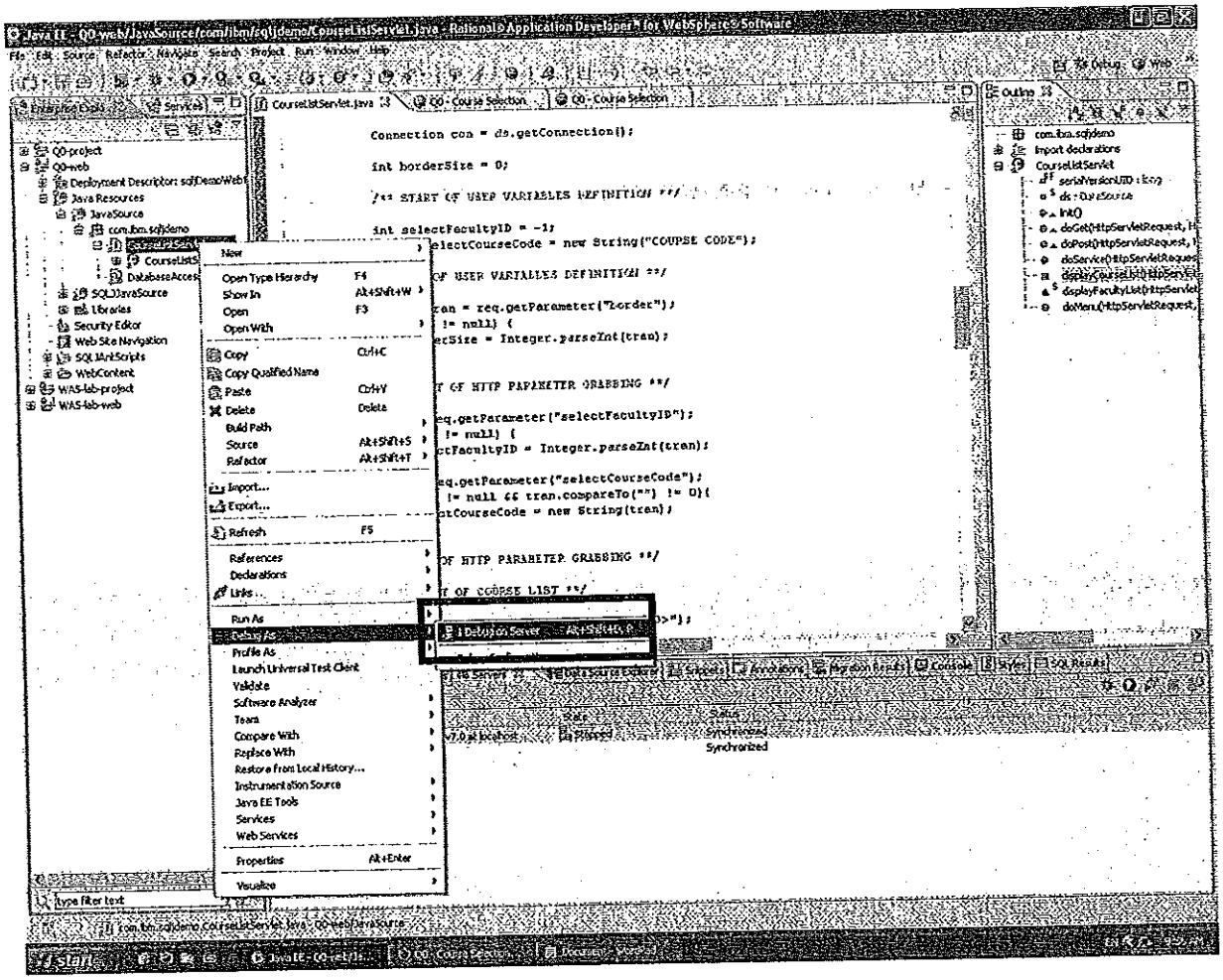
2. Banner will appear and a 'Workspace Launcher' window will appear which allows you to choose the base location of the workspace, type: "*C:\Documents and Settings\swg\IBM\rationalsdp\workspace*" as workspace.



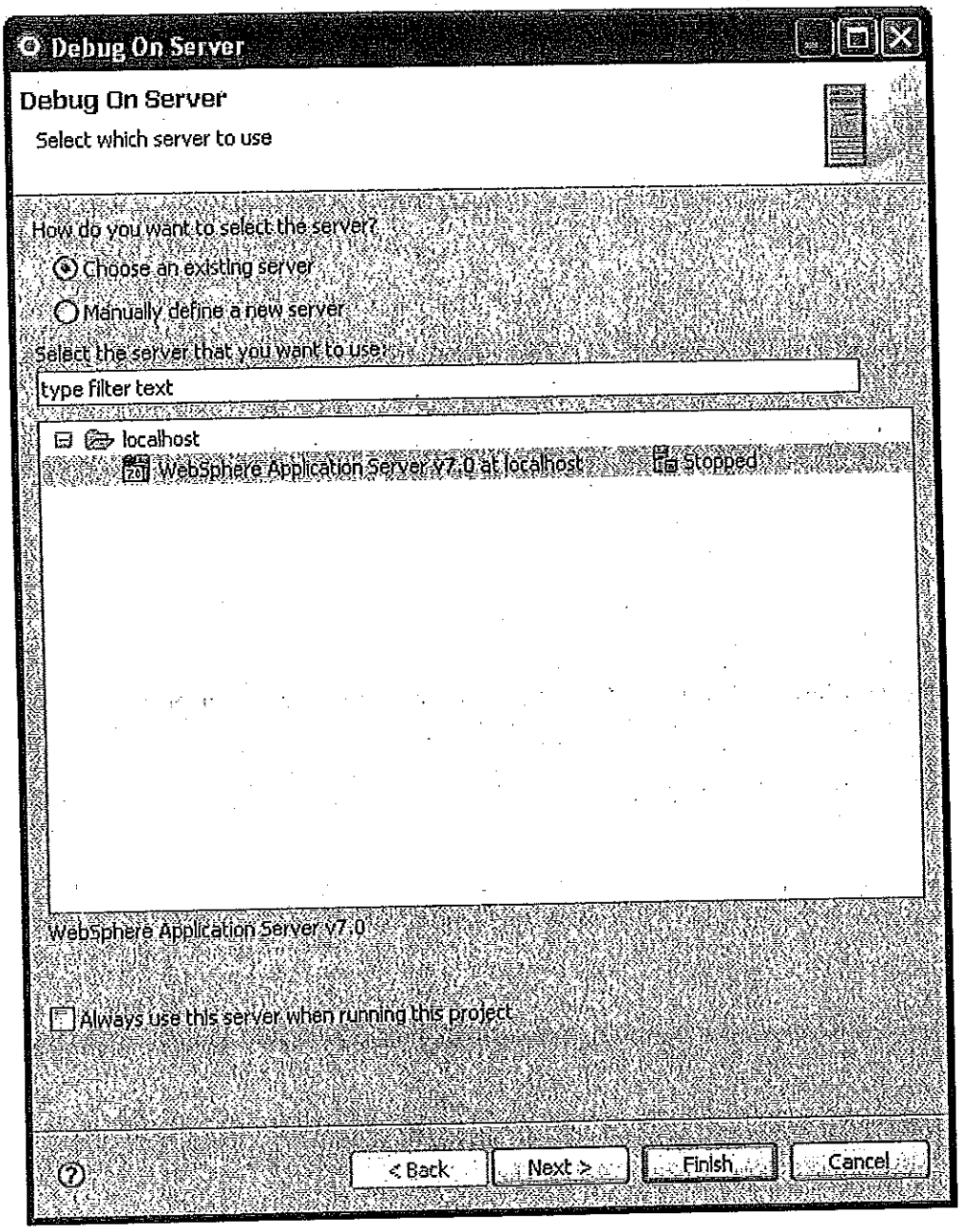
3. Open the source CourseListServlet.java. Double click the blue left bar at line 172 to create breakpoint



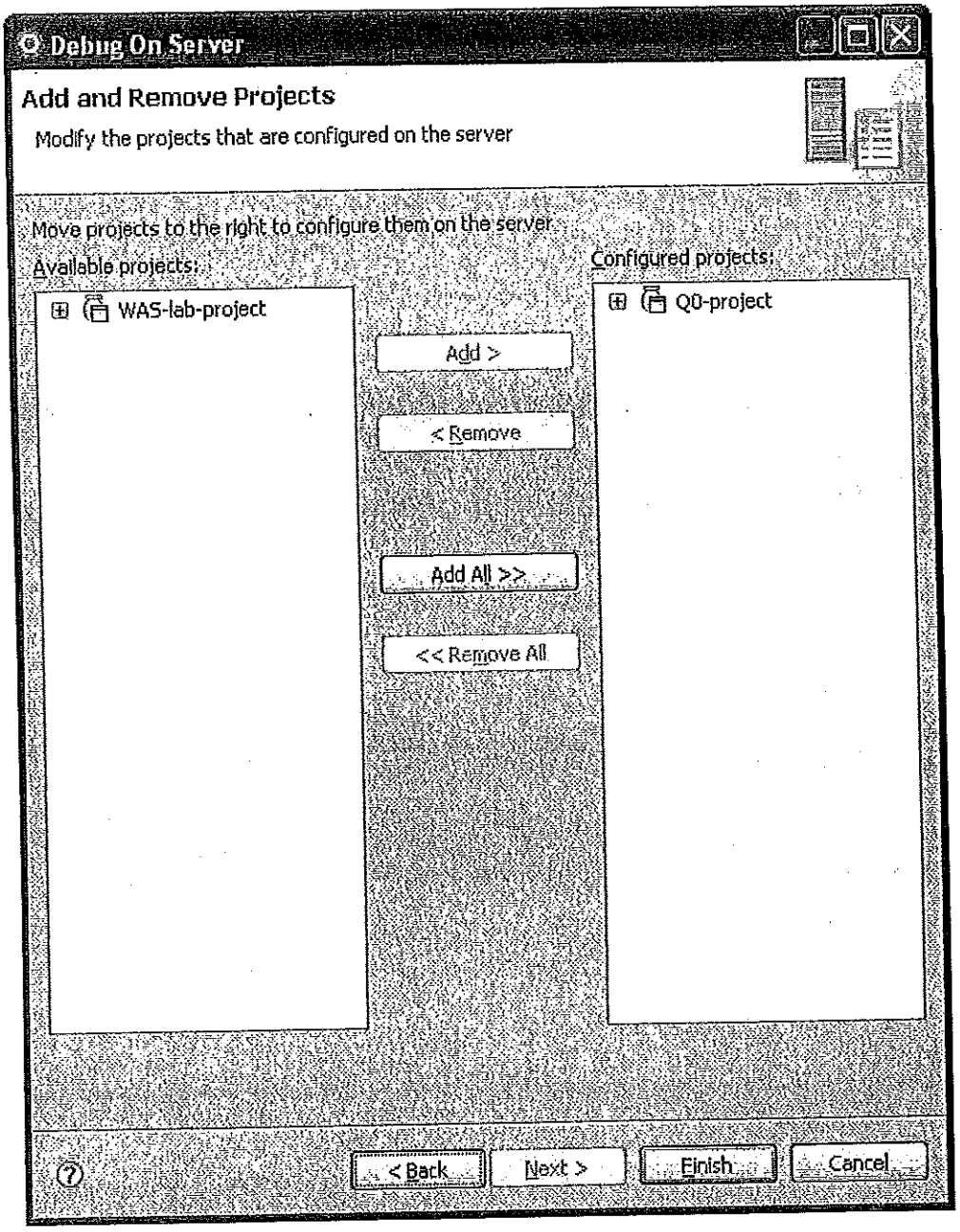
4. Right click the CourseListServlet.java file node, click Debug As > Debug on Server.



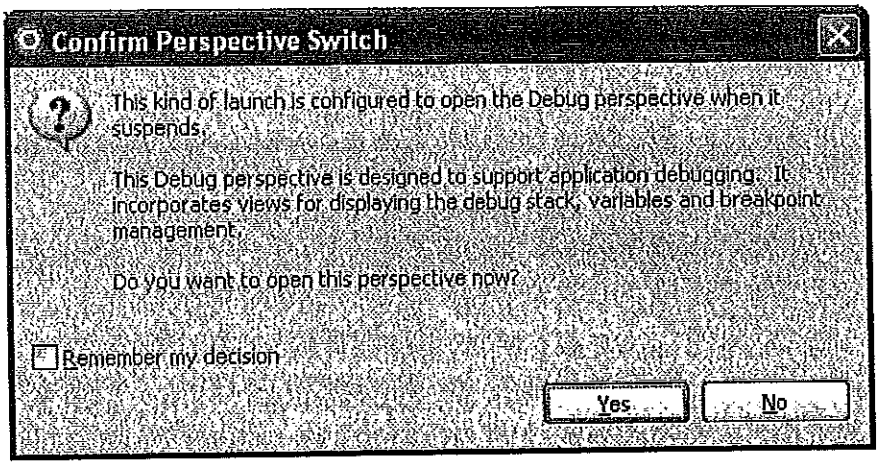
5. Choose WebSphere Application Server v7.0 at localhost as the server.



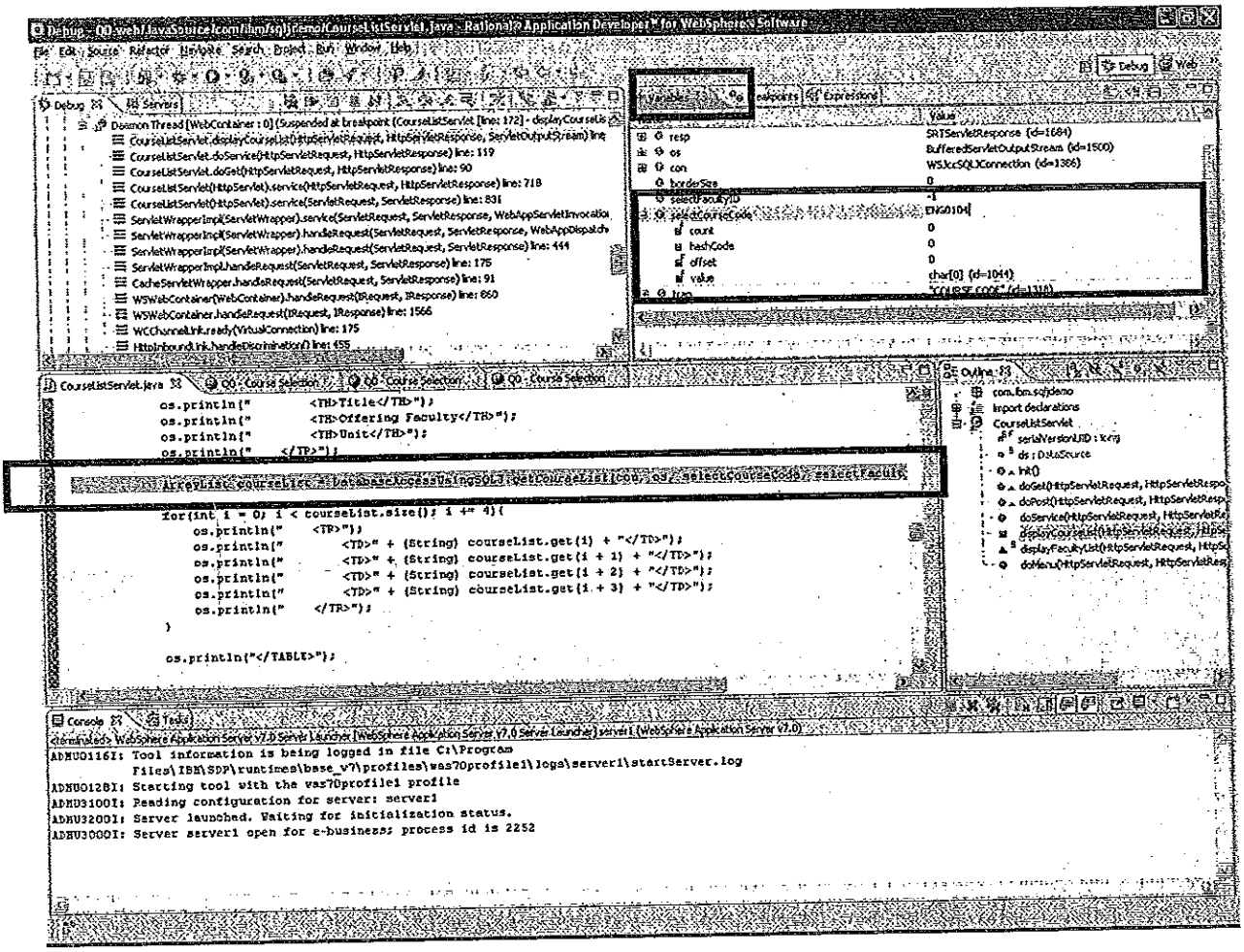
6. Include Q0-project in the Project to debug



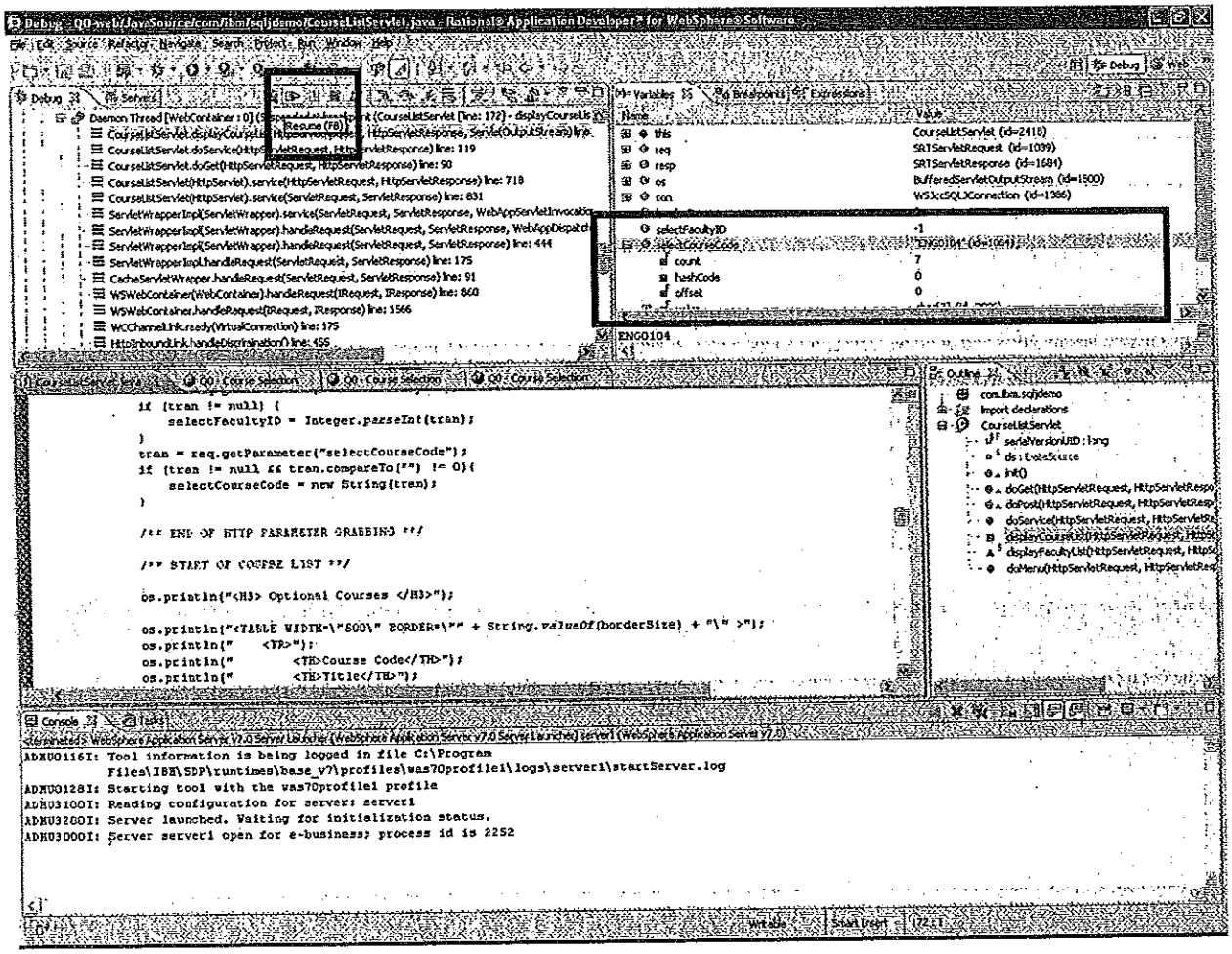
7. It may switch to Debug perspective, click Yes.



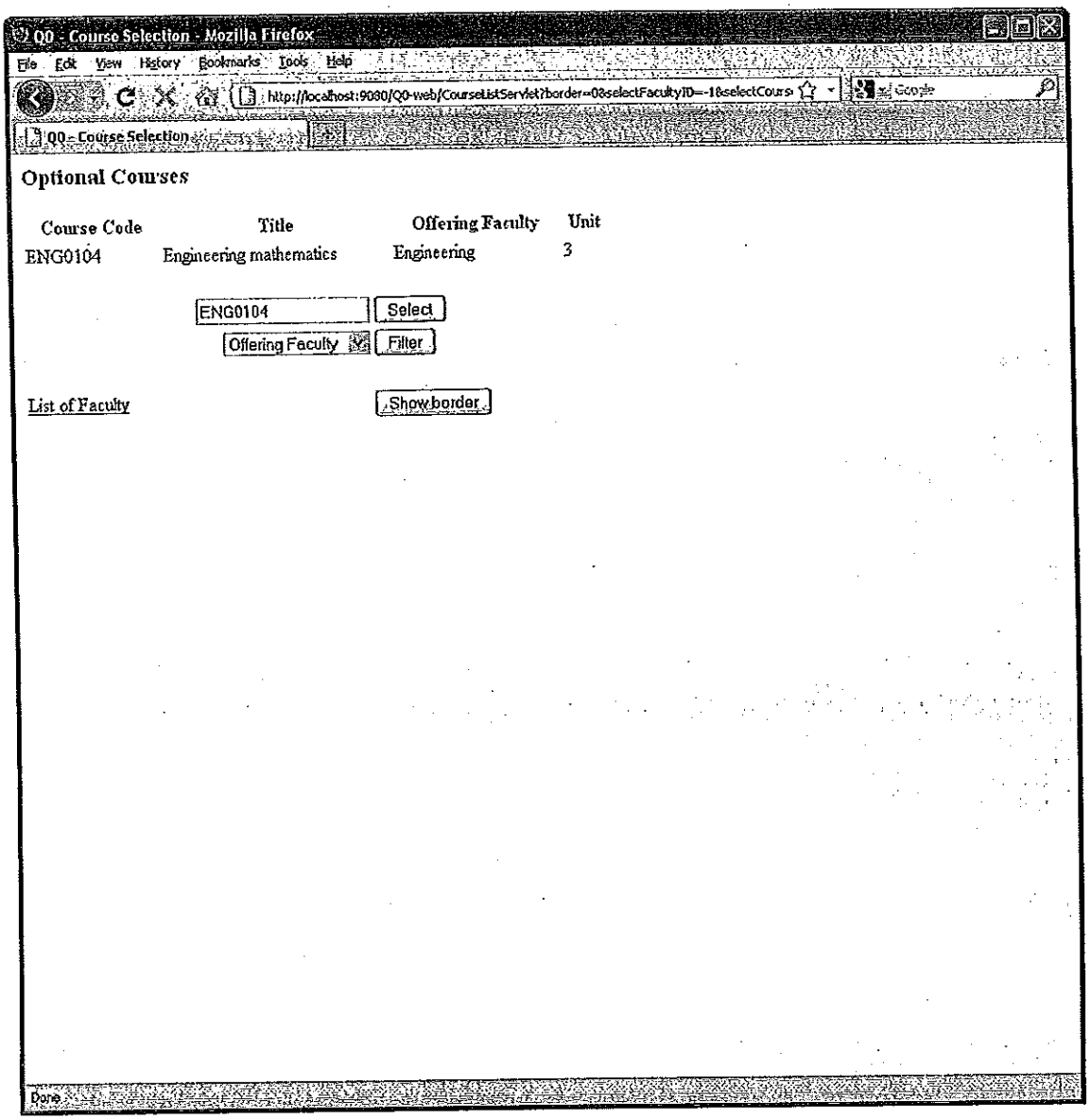
8. Prompt the debug perspective, and locate to the breakpoint just defined on line 172. And in Variables View, you may found the node selectCourseCode



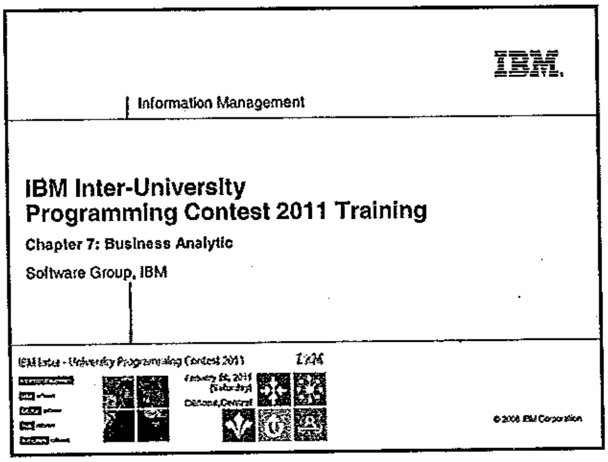
9. click the value of selectCourseCode, edit it to "ENG0104", leave the box to save. Click resume.

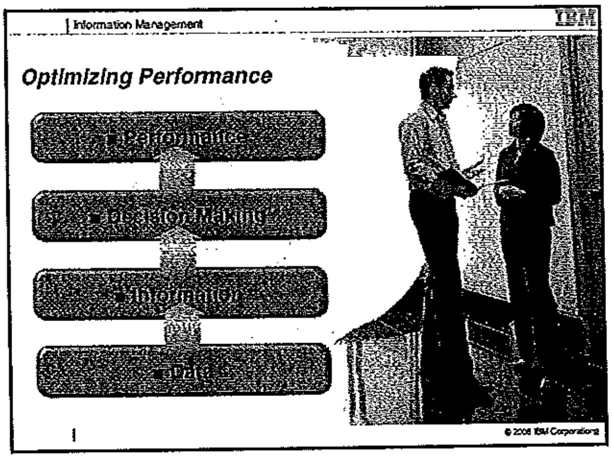


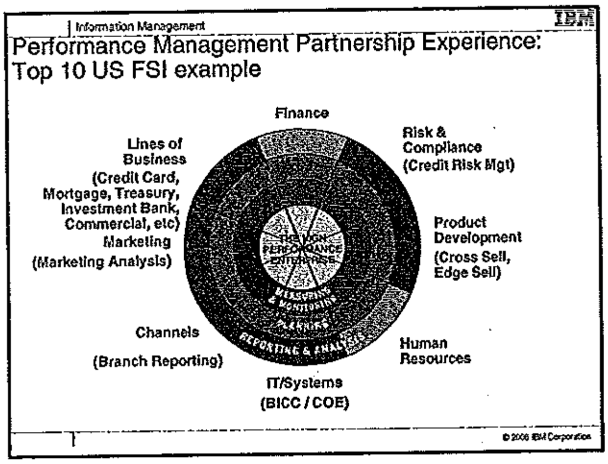
10. Return to the browser, you may found the value of edit box changed to "ENG0104". Which the value has been updated in debug.

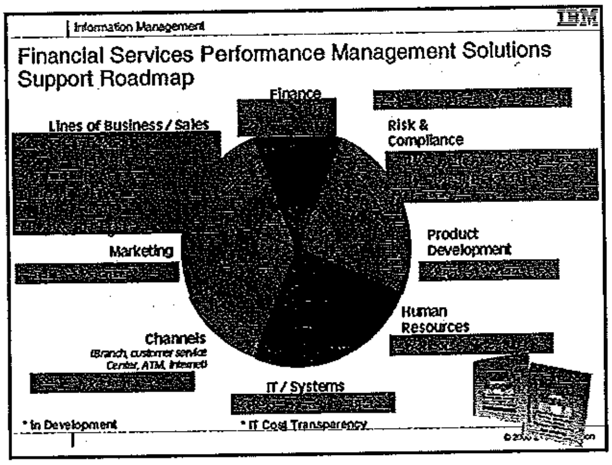


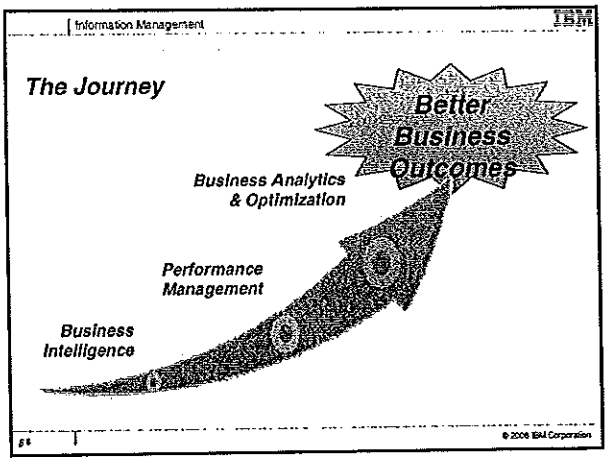
--end of lab--

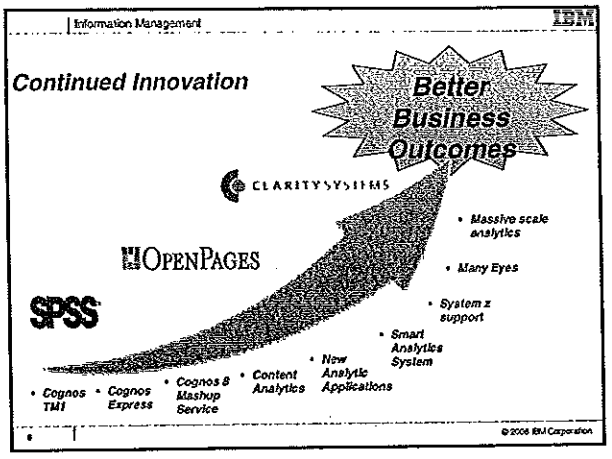

IBM.
Information Management
IBM Inter-University
Programming Contest 2011 Training
Chapter 7: Business Analytic
Software Group, IBM
© 2006 IBM Corporation


Information Management
IBM
Optimizing Performance
Performance
Decision Making
Information
Data
© 2006 IBM Corporation

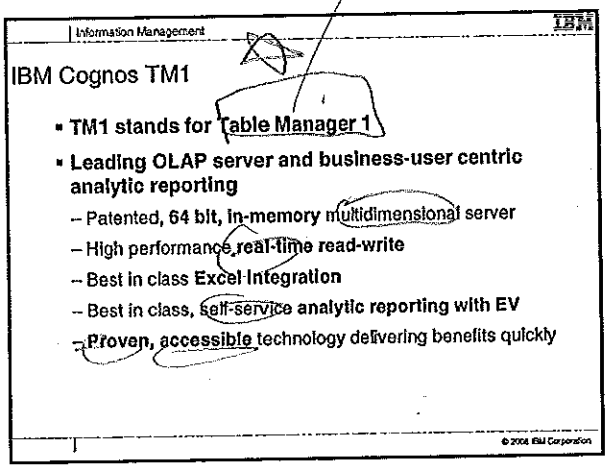

Information Management
IBM
Performance Management Partnership Experience:
Top 10 US FSI example
Finance
Risk & Compliance (Credit Risk Mgt)
Product Development (Cross Sell, Edge Sell)
Human Resources
IT/Systems (BICC / COE)
Channels (Branch Reporting)
Lines of Business (Credit Card, Mortgage, Treasury, Investment Bank, Commercial, etc)
Marketing (Marketing Analysis)
REPORTING & ANALYSIS
PLANNING
© 2006 IBM Corporation

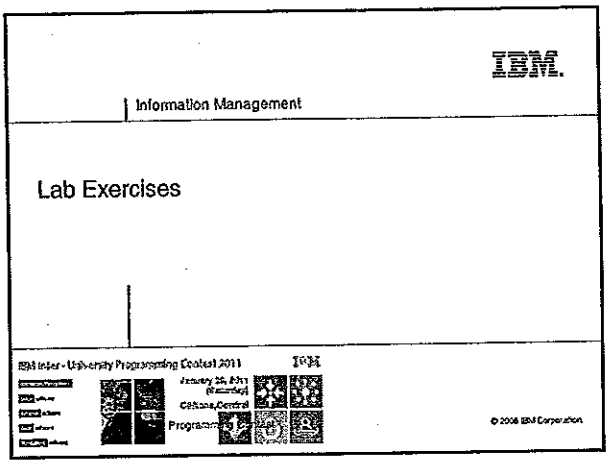

Information Management
IBM
Financial Services Performance Management Solutions
Support Roadmap
Finance
Lines of Business / Sales
Risk & Compliance
Product Development
Marketing
Human Resources
Channels
(Branch, customer service Center, ATM, Internet)
IT / Systems
* In Development
* IT Cost Transparency





in-ram database
real time.

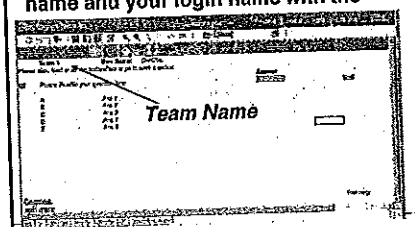


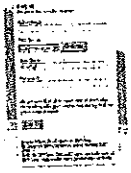


Information Management

Multiple Choice Input Process

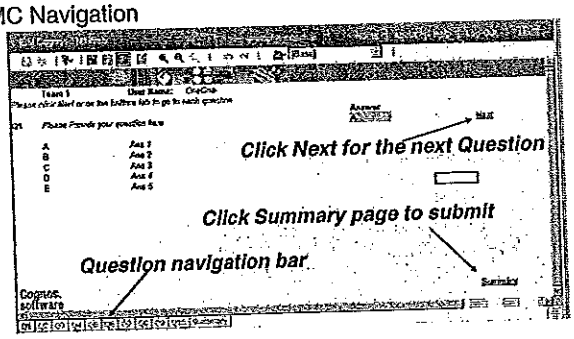
- **Login Screen**
  - User Login / Password
- **After login, it will show your team name and your login name with the**





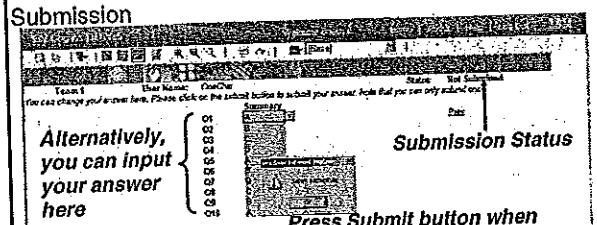
Information Management

MC Navigation



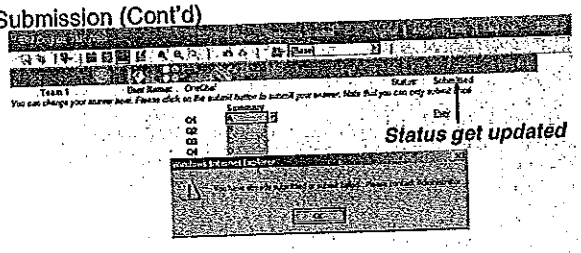
Information Management

Submission



Information Management

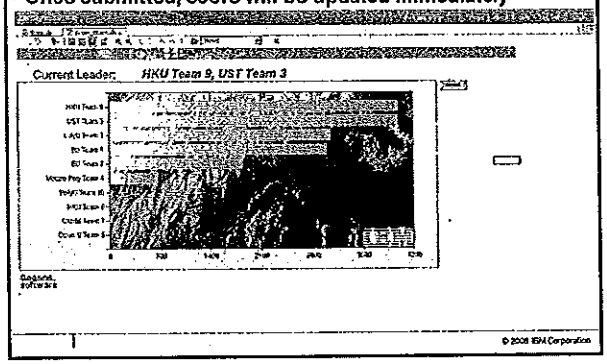
Submission (Cont'd)



Information Management

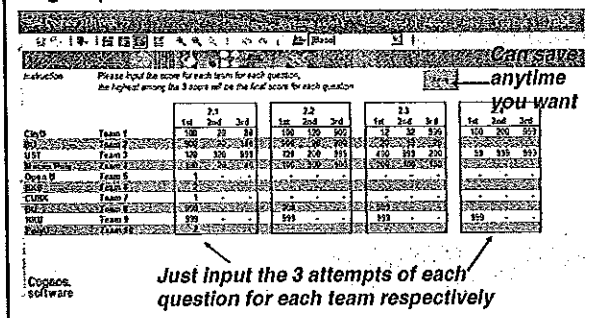
Score Dashboard

- **Once submitted, score will be updated immediately**



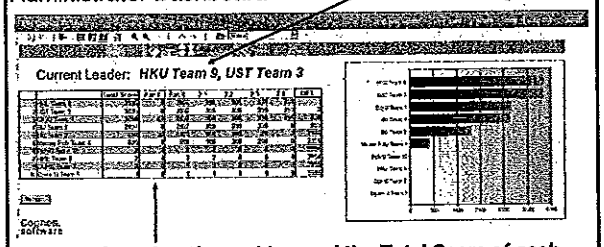
Information Management

Judge Input Score



Information Management

Administrator Dashboard



Can view the ranking and the Total Score of each team, and the marks difference from the leader, but this is a read-only dashboard

Chapter 7: Multiple Choice Screen Walkthrough

Objectives

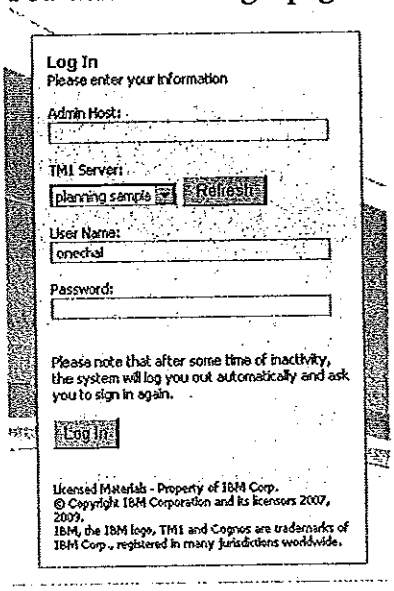
In this exercise, we will learn:

- How to connect to the multiple choice screen
- How to log in
- How to answer each multiple choice question
- How to submit your answer once you are finished
- How to examine your submission status

Exercises

1. In the VM that is started in Chapter 1, open an Internet Explorer and type the URL http://cogserver/tm1web

   You will see the login page like this,

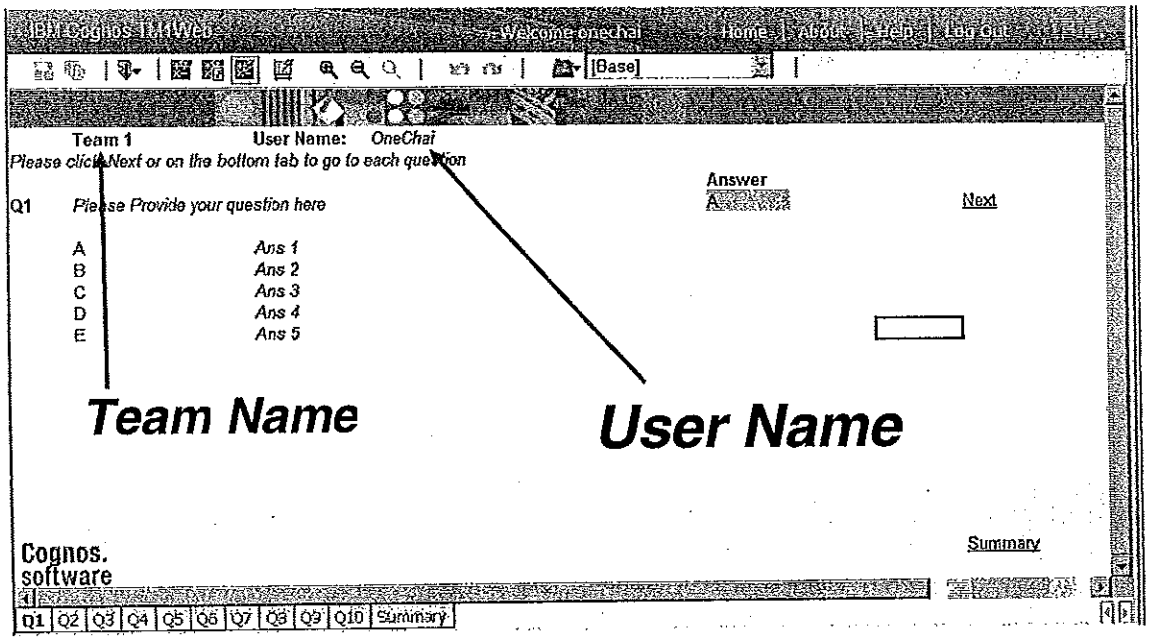


user name : team11

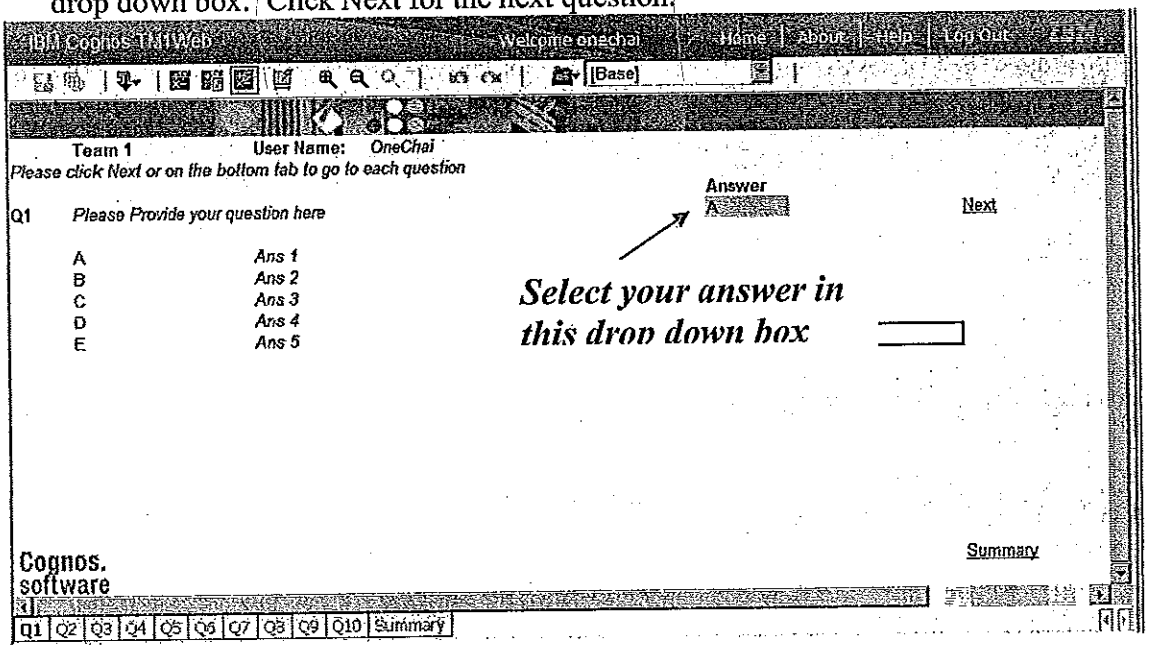
password : password

2. In the Log In screen, just leave the Admin Host blank, and select "planning sample" in the TM1 Server.

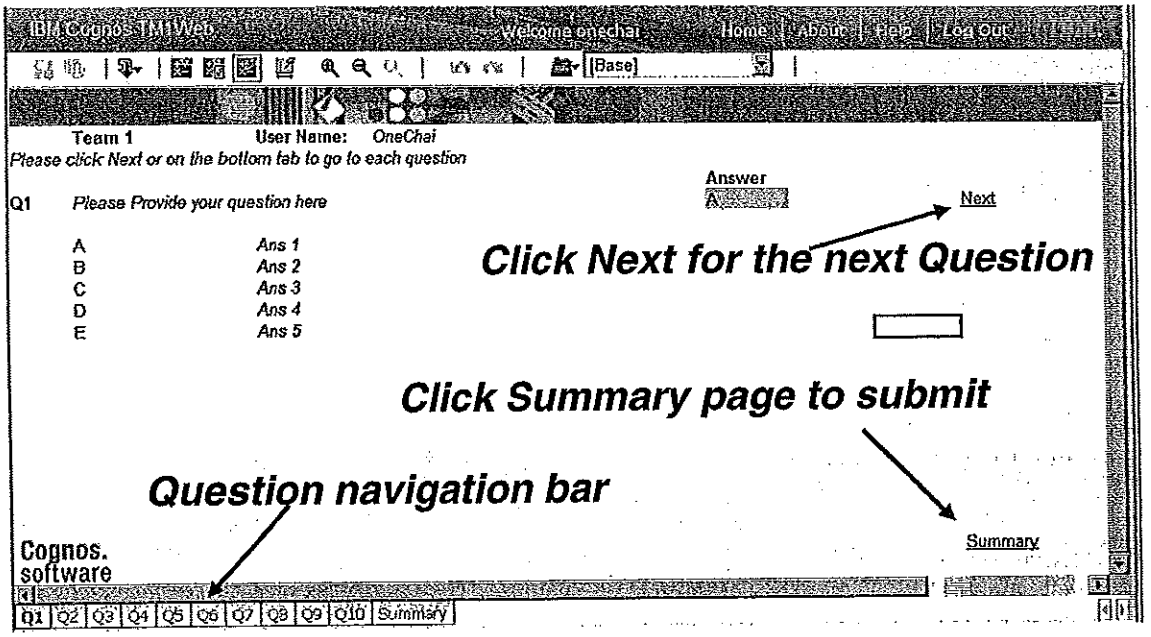
   Next, Login into the multiple choice application by the supplied credentials. After you had logged in, you can check if the team name and user name is correct.



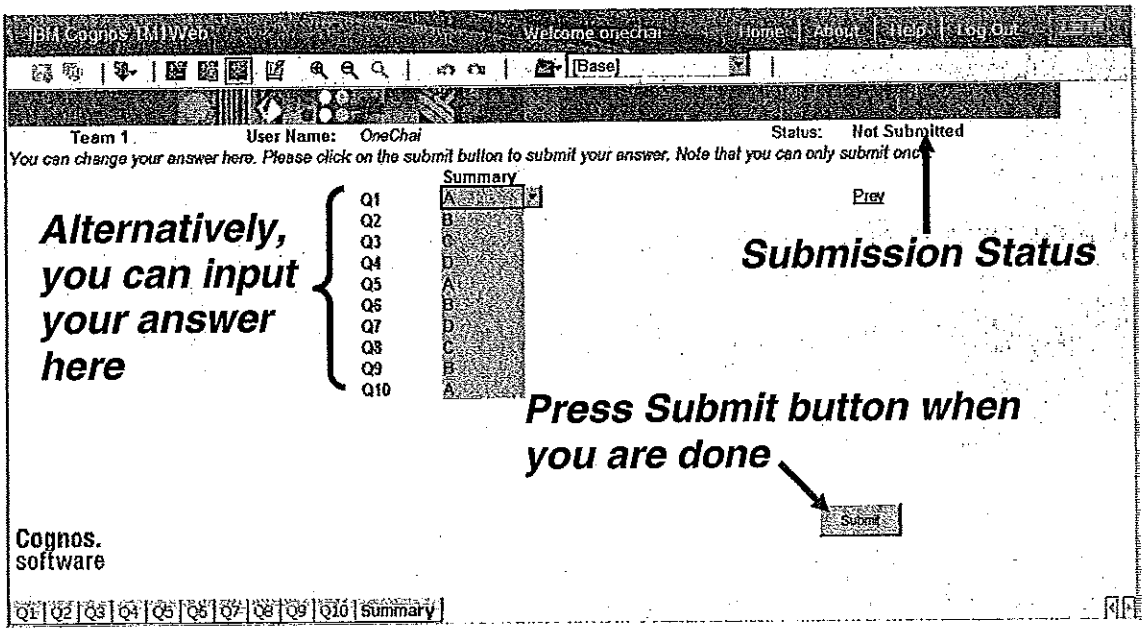
3. After logged in, the first question is displayed and selects the correct answer in the drop down box. Click Next for the next question.



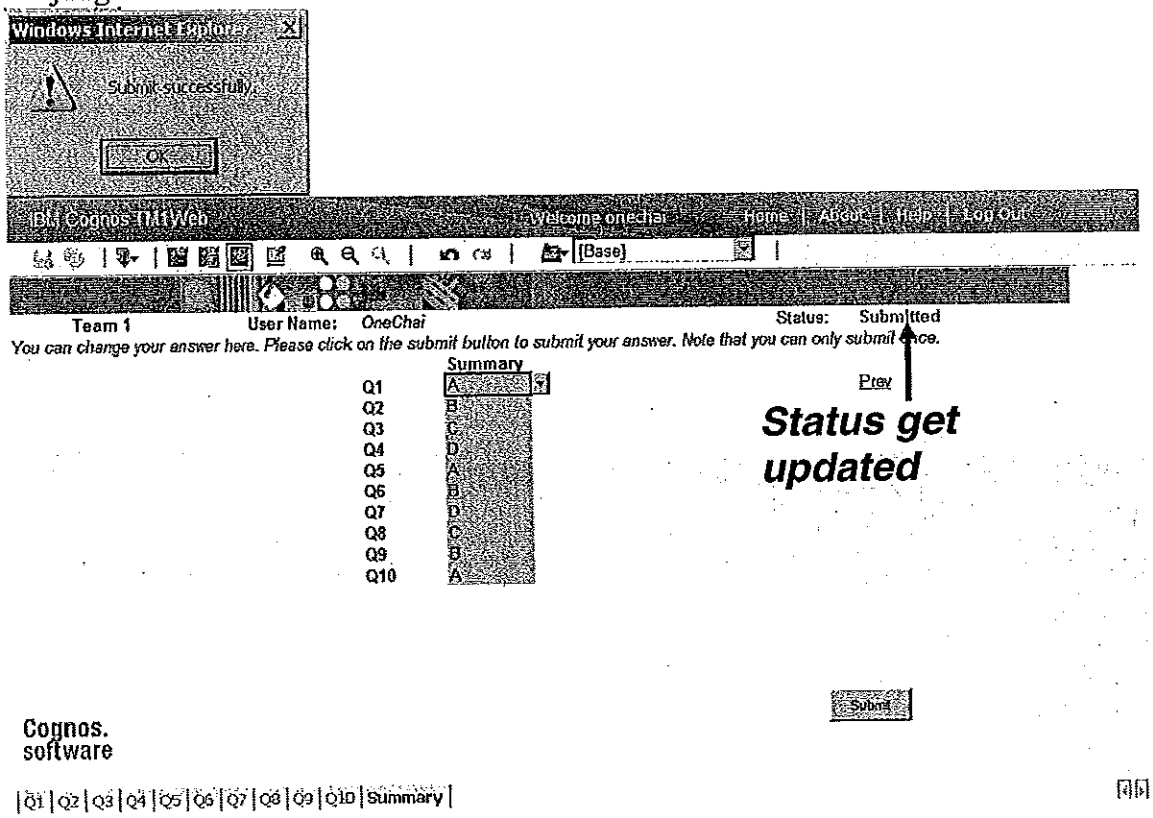
4. To navigate to different questions directly, you may access it through the bottom tab.



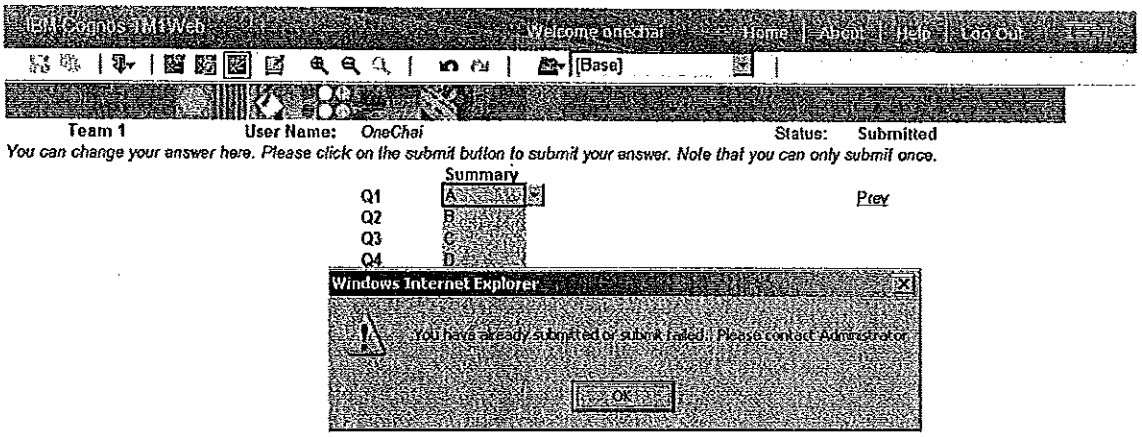
5. In order to submit your answer, you must go to the summary page and review your answer. For your convenience, you may also answer the question in this summary page directly from each of the drop down box besides the question number. Note that your Status now should be "Not Submitted". After you confirm it, press the button "Submit".



6. Once you submitted, a confirmation box should be appeared. Also check the status should be changed to "Submitted". If you got an error message, please contact your judge.



7. If you try to resubmit your answer, you will get an error message like this.



8. Once you have submitted your answer, the score will be appeared in the Score Dashboard after a refresh.