



# Introduction to Rational System Developer for System z (also known as RDz)

Basic Navigation  
Creating PDS  
Running a Program

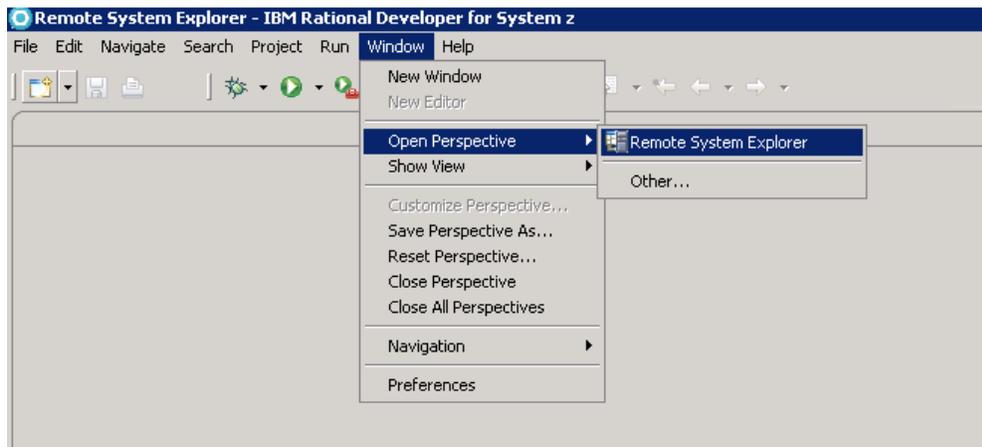
## Workbench fundamentals

The workbench is the user interface for Rational® Developer for System z. The workbench features an integrated development environment with customizable *perspectives* that support role-based development. The workbench provides a common way for all members of your project team to create, manage, and navigate resources easily. It consists of interrelated *views* and *editors*. Views provide different ways of looking at the resources you are working on. Editors allow you to create and modify code. More than one workbench window can be open on the desktop at any given time. To access the Rational Developer workbench we must first establish connections to the system.

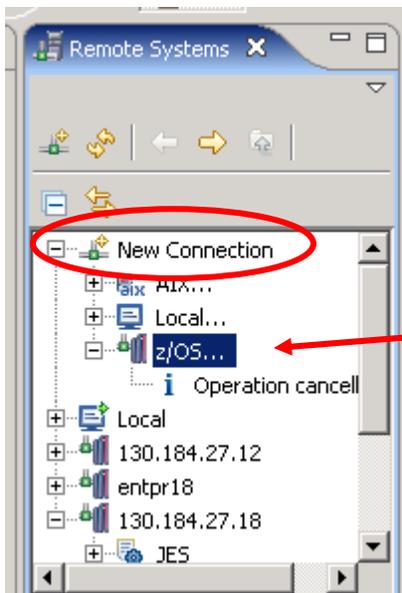
### Create a connection for z/OS Perspective

To use RDz you must first start the software and then create a connection to the z/OS system that we will be using.

If you do not see a tab on your page for “Remote Systems”, navigate to the top menu bar and hit Window -> Open Perspective -> Remote System Explorer.



This will open a view as shown below. If you do not see a connection for a system called EPRS18 or enterprs.waltoncollege.uark.edu, you will have to create one. Double click on “New Connection”



Select a z/OS connection. Enter the following information:

A screenshot of the 'New Connection' dialog box for a 'Remote z/OS System Connection'. The dialog is titled 'New Connection' and 'Remote z/OS System Connection'. It contains the following fields and options:

- Parent profile: TS-IBM1
- Host name: enterprs.waltoncollege.uark.edu
- Connection name: enterprs.waltoncollege.uark.edu
- Description: (empty text box)
- Verify host name

At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present.

Hit the “Next” button until you encounter the screen to enter the JES Job Monitor port—make sure that it is set to 6715.

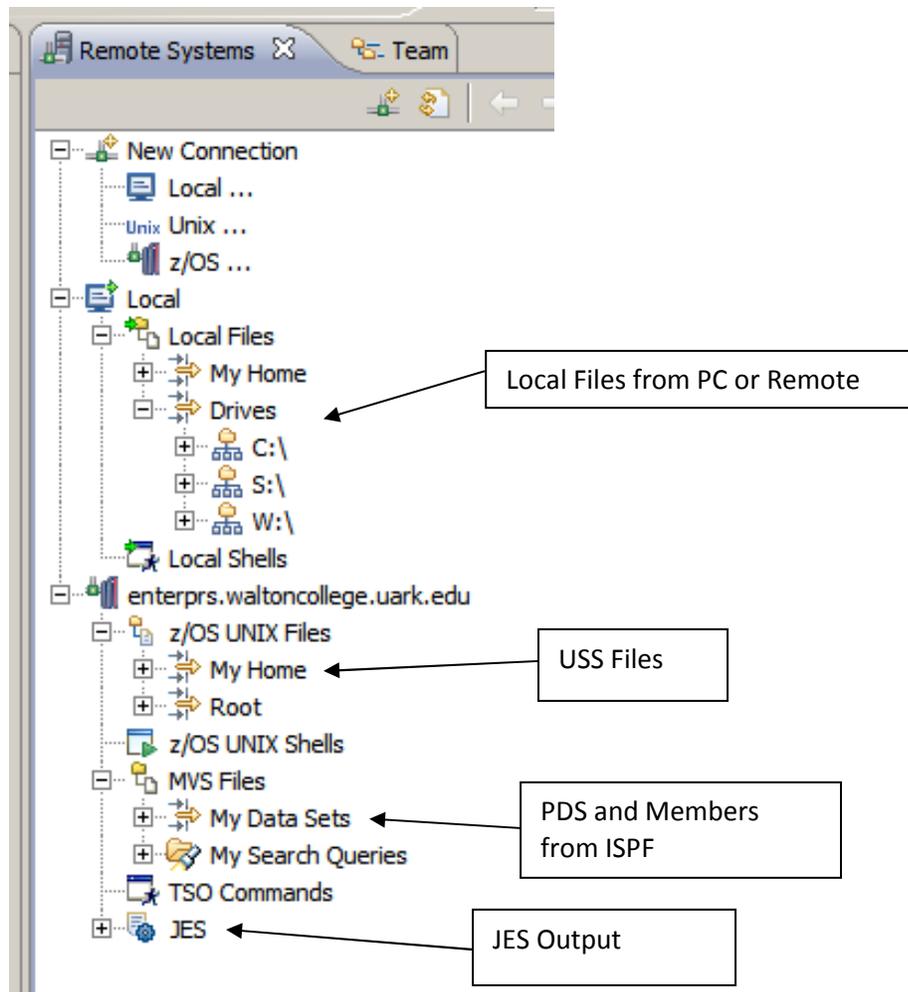
Hit “Finish”. You will also need to supply your userID and password to RDz on the popup box. You may receive a popup screen that says “Connection enterprs.waltoncollege.uark.edu has not been secured using SSL. Proceed anyway?” You may answer “Yes” to proceed. Now we are logged into the z/OS system.

Now we are connected to the system, we are ready to learn something about RDz and how it works. RDz is based on an open-source development environment called Eclipse. If you are already familiar with this environment, you will find most things in the RDz environment very familiar.

## **Perspectives**

Perspectives are combinations of views and editors that show various aspects of the project resources and are organized by developer role or task. For example, a Java™ developer would work most often in the Java perspective, while a Web designer would work in the Web perspective. You will find that you can do everything that we have already done in the ISPF environment in RDz!

First, let’s take a look at the “remote systems” panel, which should be on the right of your screen.



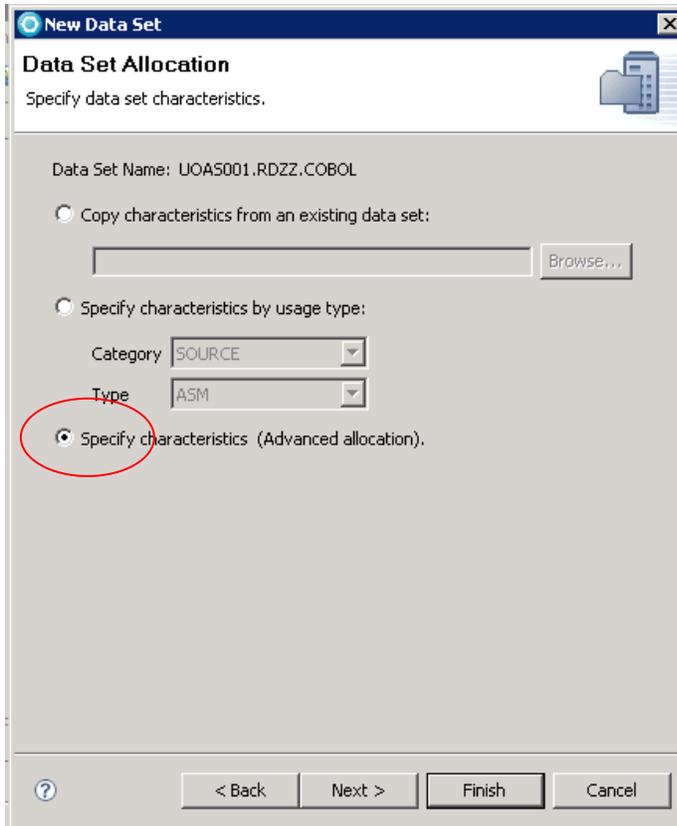
In the remote explorer, we can see several different types of files. . . files from our remote desktop, JES output files, MVS from our mainframe system and even UNIX files from our system. Most of these will be blank for your ID right now, since we have not created any programs quite yet. We will see very quickly how simple it is to create and move files between these different environments. Instead of having to go through that tedious upload/download process, which we did in the ISPF environment, we can now simply drag and drop our content into PDS members.

To illustrate how easy it is to create a PDS and member, let's begin with a simple exercise. To allocate a new PDS we simply right-click on MVS Files and then "Allocate PDS"



Select a name for your PDS—for this exercise we will need to create **three** PDS's. In the RDz environment the NAME of the PDS is VERY IMPORTANT!!!! For Exercise 3 we will need a PDS called UOASxxx.RDZ.COBOl to hold our Cobol programs. IT MUST have "COBOL" in the third part of the PDS name. This tells RDz to enable syntax check on these members.

Hit "NEXT"



**New Data Set**

**Data Set Allocation**  
Specify data set characteristics.

Data Set Name: UOAS001.RDZZ.COBOL

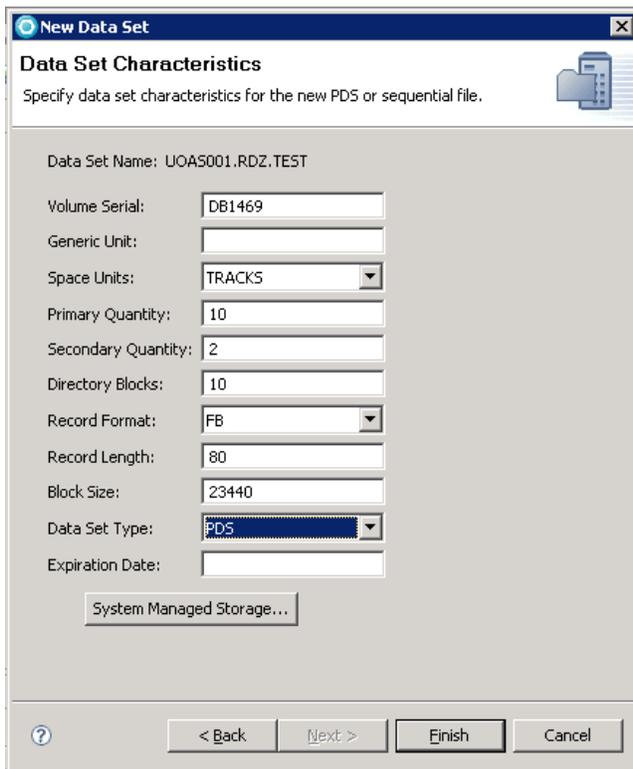
Copy characteristics from an existing data set:  
[Text Field] [Browse...]

Specify characteristics by usage type:  
Category: [SOURCE]  
Type: [ASM]

Specify characteristics (Advanced allocation).

[?] [ < Back ] [ Next > ] [ Finish ] [ Cancel ]

Hit the "Next" tab twice, and we come to the screen where we input our space requirements:



**New Data Set**

**Data Set Characteristics**  
Specify data set characteristics for the new PDS or sequential file.

Data Set Name: UOAS001.RDZ.TEST

Volume Serial: [DB1469]  
Generic Unit: [Text Field]  
Space Units: [TRACKS]  
Primary Quantity: [10]  
Secondary Quantity: [2]  
Directory Blocks: [10]  
Record Format: [FB]  
Record Length: [80]  
Block Size: [23440]  
Data Set Type: [PDS]  
Expiration Date: [Text Field]

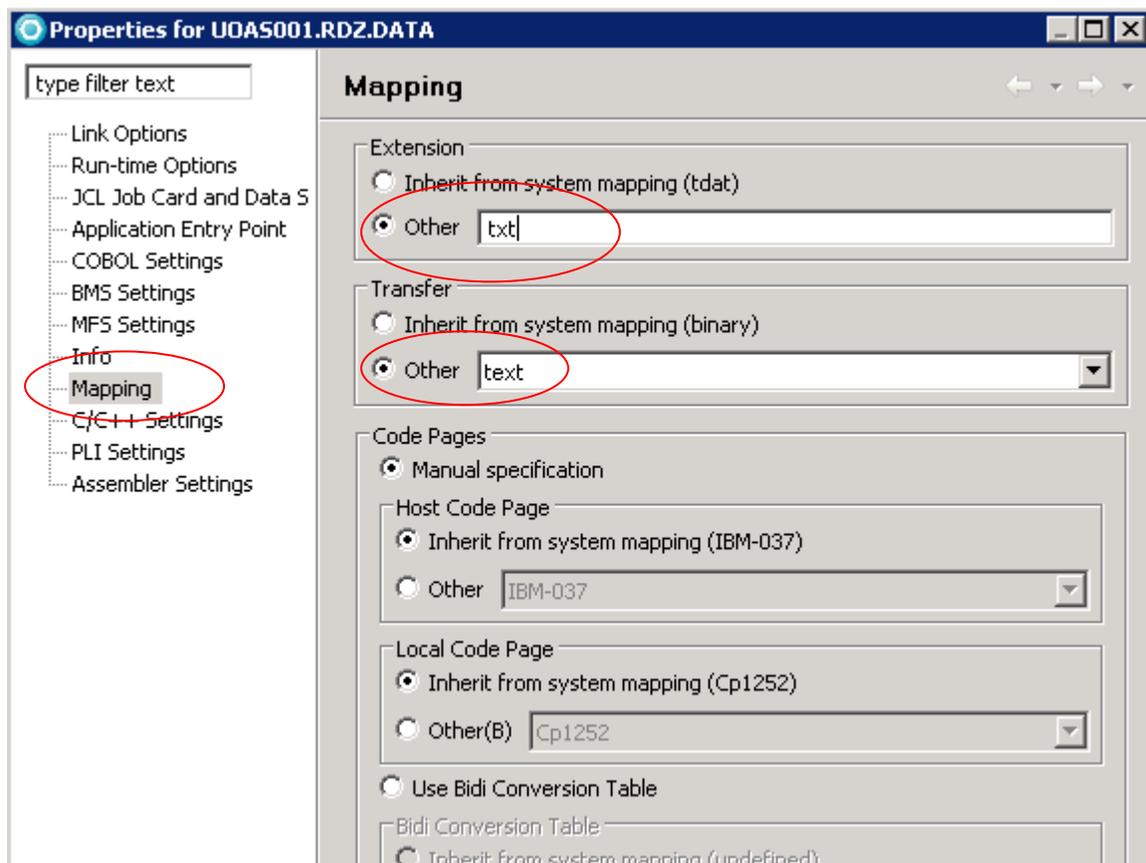
[System Managed Storage...]

[?] [ < Back ] [ Next > ] [ Finish ] [ Cancel ]

Hit “Finish”.

Next, we need to create our PDS for JCL. Following the same steps as above, create a PDS named UOASxxx.RDZ.JCL.

Last, we need to create a PDS to hold our Data files. Again, we allocate a PDS, calling the PDS UOASxxx.RDZ.DATA. This time we are going to have to edit the properties of the dataset so that we can transfer text files into the file. Right click on your data PDS and hit “Properties”.

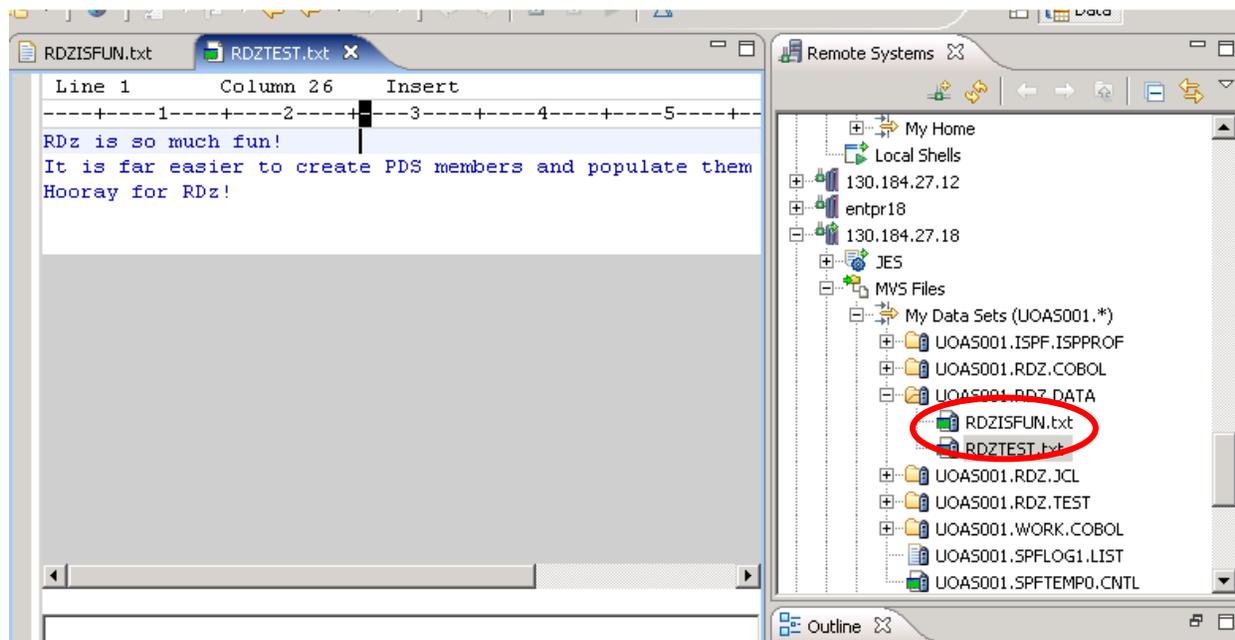


Now, just for fun let’s see how easy it is to populate a PDS member in RDz from a file on the local workstation. Create a file in notepad on your remote desktop with the following text:

RDz is so much fun!  
It is far easier to create PDS members and populate them with data from the workstation.  
Hooray for RDz!

Save the file to your local workstation with the name RDZISFUN. In the remote systems panel, simply right click on the local file that you just created and saved. Scroll down and find the PDS

for DATA files. Select the PDS for DATA files and right-click on it. Select paste. A new member appears under the PDS with the name RDZISFUN.



If you Double-click on your new member, the panel in the center will show you the text that was populated in this member. Now wasn't that much easier than old Upload/Download facility?

### Creating Program, Data and JCL for Exercise #3

Our next step is to create the COBOL program, data file and JCL which in RDz which we will use in the third exercise. We have already created PDS's of the correct type for data, JCL and the COBOL program, so now let's create our members in each of the PDS. In your UOASxxx.RDZ.JCL PDS, create a member called COBJCL.

Copy the following JCL statements into the blank member and Save (**HINT: You may have to open a blank Notepad file in TS-IBM desktop and copy/paste the code in from your lab PC**):

```
//COBRDZ JOB (COBOLPROG),'userid',NOTIFY=userid,
// CLASS=A,MSGLEVEL=(1,1),TIME=1,MSGCLASS=A
//STEP1 EXEC PROC=IGYWCLG
//COBOL.SYSIN DD DSN=userid.RDZ.COBOLO(COBOL),DISP=SHR
//GO.DISKFL1 DD DSN=userid.RDZ.DATA(RDZOUT),DISP=SHR
//GO.READER DD DSN=userid.RDZ.DATA(COBDAT),DISP=SHR
```

Now create our COBOL program in UOASxxx.RDZ.COBOL and save the following program. COBOL Statements save in member UOASXXX.RDZ.COBOL(COBOL):

```

IDENTIFICATION DIVISION.
PROGRAM-ID. COBTEST.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT CARD-IN      ASSIGN TO UT-S-READER.
    SELECT DISK-FILE1  ASSIGN TO UT-S-DISKFL1.
DATA DIVISION.
FILE SECTION.
FD DISK-FILE1 LABEL RECORDS ARE STANDARD
                RECORDING F
                BLOCK CONTAINS 0 RECORDS.

01 DSK-RCD1.
   02 FILLER PIC X(80).
FD CARD-IN LABEL RECORDS OMITTED
            RECORDING F.

01 CRD-REC.
   02 CRD-NAME PIC X(30).
   02 FILLER PIC X(23).
   02 CRD-U-ID PIC X(7).
   02 FILLER PIC X.
   02 CRD-PASSW PIC X(6).
   02 FILLER PIC X(13).
WORKING-STORAGE SECTION.
01 EOF PIC X VALUE 'N'.
01 HOLD-INST.
   02 CRD-CRSE PIC X(30).
   02 FILLER PIC X(23).
   02 CRD-INST-ID PIC X(7).
   02 FILLER PIC X.
   02 CRD-PWD PIC X(6).
   02 FILLER PIC X(13).
01 OUTPUT-INST.
   02 CRD-OUT-ID PIC X(7).
   02 FILLER PIC X(2) VALUE SPACES.
   02 CRD-OUT-PWD PIC X(6).
   02 FILLER PIC X(65) VALUE SPACES.
PROCEDURE DIVISION.
    OPEN INPUT CARD-IN
        OUTPUT DISK-FILE1.
    READ CARD-IN INTO HOLD-INST
        AT END
            MOVE 'Y' TO EOF.
    PERFORM 1000-PROCESS-IT
        UNTIL EOF = 'Y'.
    CLOSE CARD-IN
        DISK-FILE1.
    STOP RUN.
2000-RD-CRD-RT.
    READ CARD-IN INTO HOLD-INST
        AT END
            MOVE 'Y' TO EOF.
1000-PROCESS-IT.
    MOVE CRD-INST-ID TO CRD-OUT-ID.
    MOVE CRD-PWD TO CRD-OUT-PWD.
    WRITE DSK-RCD1 FROM OUTPUT-INST.
    PERFORM 2000-RD-CRD-RT.

```

Now create the Datafile used to run this program:

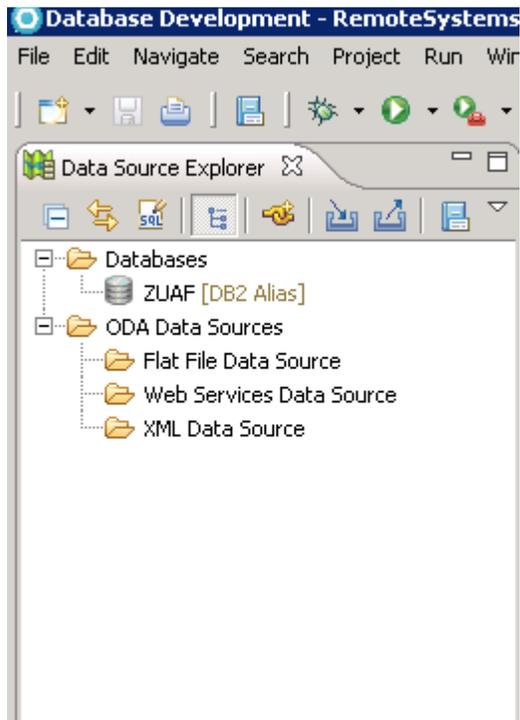
Data file save in member UOASXXX.RDZ.DATA(COBDAT) :

ANDERSON ESSABIA ANNIECE	XXXXXXXXXX	ISYS4283	002	EANDERS	PWD001
ANIPA EKO A	XXXXXXXXXX	ISYS4283	002	EAANIPA	PWD002
BLOMBERG BETTY ANN	XXXXXXXXXX	ISYS4283	002	BBLOMBE	PWD003
CORDIERO SHELDON JOHN	XXXXXXXXXX	ISYS4283	002	SCORDEI	PWD004
NESTRUD CHRISTOPHER CHARLES	XXXXXXXXXX	ISYS4283	002	CCN	PWD005
RODRIGUEZ SARAH MARTHA	XXXXXXXXXX	ISYS4283	002	SRODRIG	PWD006
SIMMONS KIMBERLY M	XXXXXXXXXX	ISYS4283	002	KMS06	PWD007
WIDEMAN JOHN RANDELL	XXXXXXXXXX	ISYS4283	002	JWIDEMA	PWD008

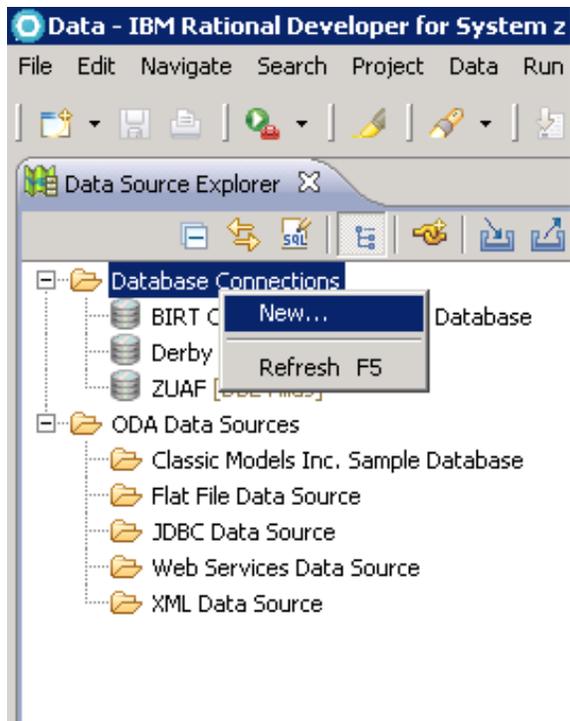
## DB2 –Create a Table:

We will have to login to DB2 in the Database Development Perspective to complete this assignment. The Database view should be located to the right of your RDz screen. If a panel similar to the screenshot below does not appear in the lower right area of your screen, on the top menu bar navigate to Window→Open Perspective→Other. Then from the next popup box, choose the “Database Development” perspective.

Locate the entry for “ZUAF” as indicated on the following screenshot:



**If you do not see ZUAF in this view you will have to create a new connection.** To do this, right click on Database Connections and click New.

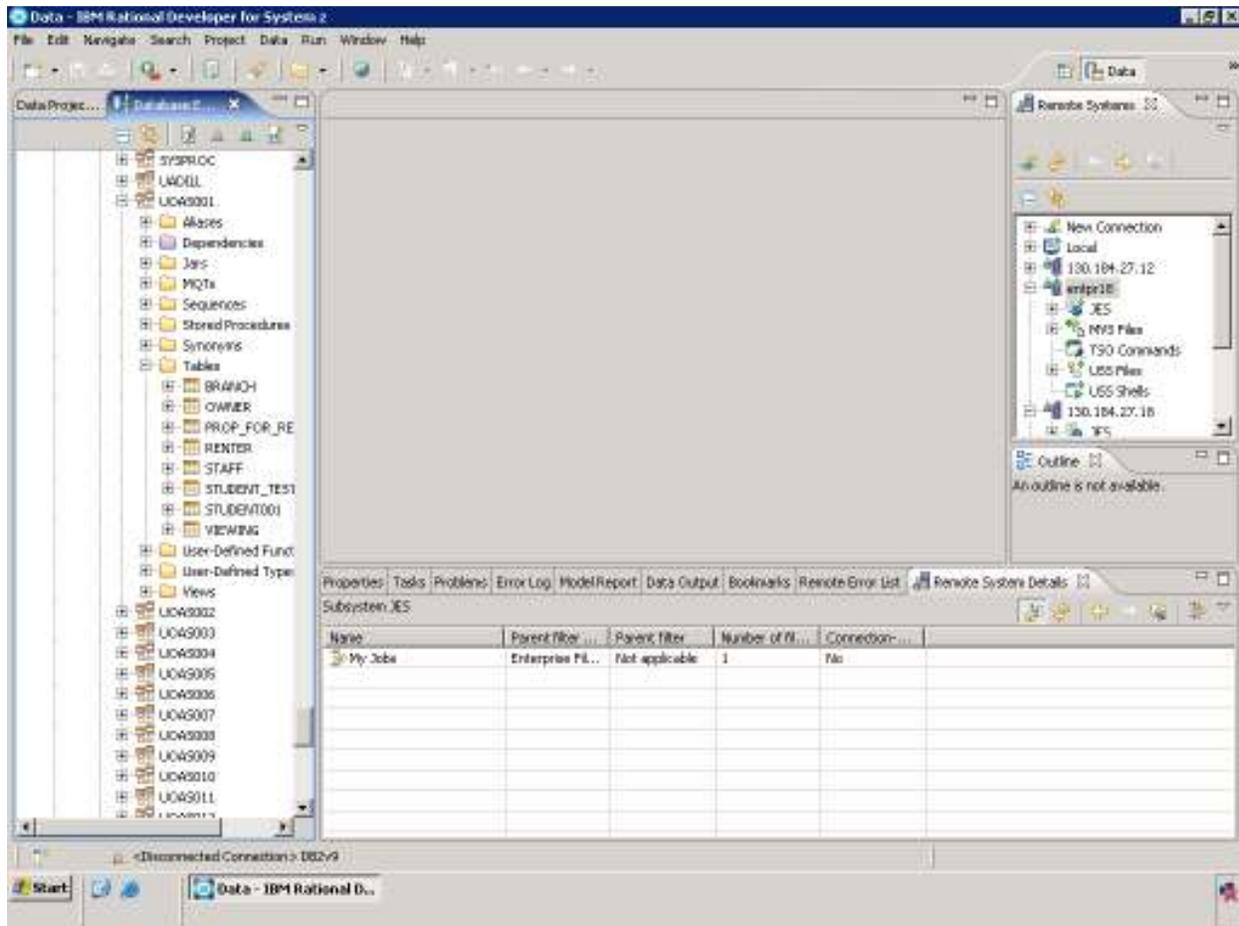


Input the following information:

On the popup which will appear, enter your assigned userID once again:

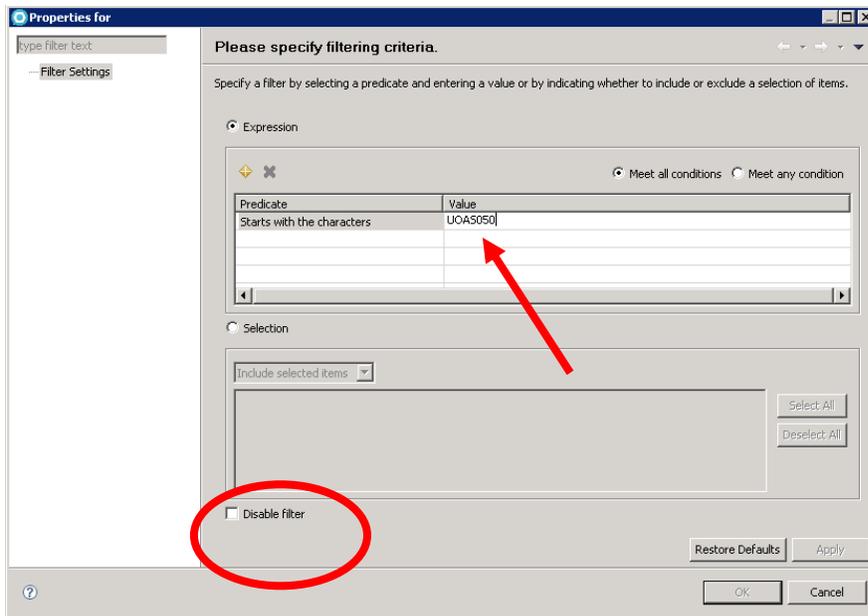
Once your login has been successfully accepted, you can open the ZUAF tab (a "+" will appear to the left of this item). From there you can open DB2v9 → ZUAF → Schemas.

In the “Schemas” tab, you should be able to scroll-down and find your database UOASxxx (with your number). Open your database and find the “Tables” tab. Unfortunately, we have no tables, quite yet. . .

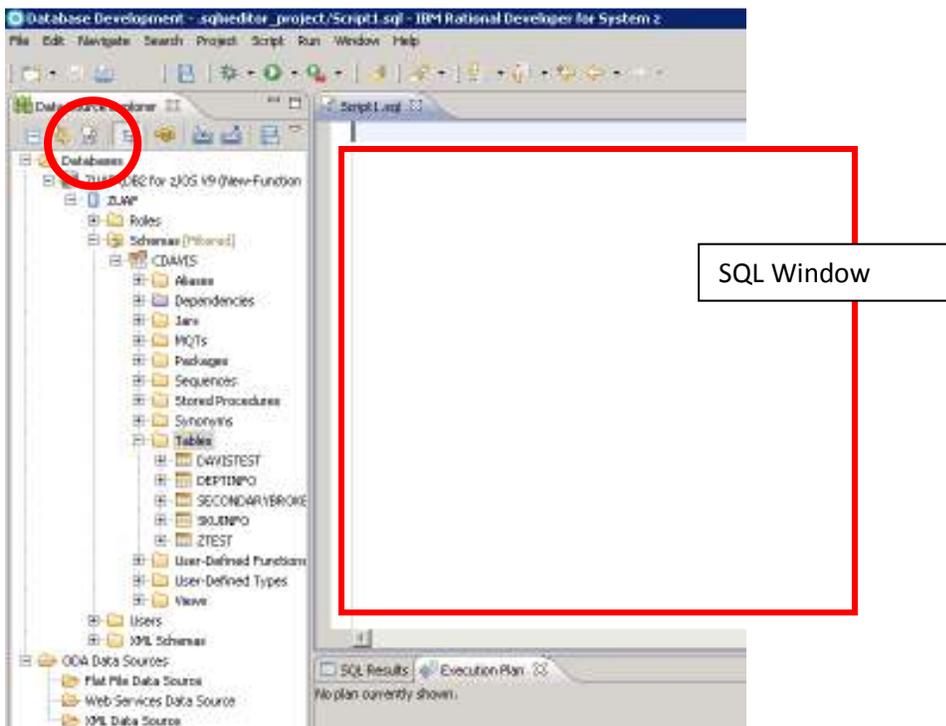


## Create a database table

A schema is a named collection of database tables and artifacts and their relationships. To limit the number of nodes displayed in the Database Explorer tree, a filter has been established in the example. To do so, right-click on the **Schemas** node, and specify the filter, as shown the following screenshot:



On the Datasource explorer tab, locate the SQL icon:



Notice that when you push it, the SQL window opens in the center of the page. Copy the following SQL code into the SQL Window:

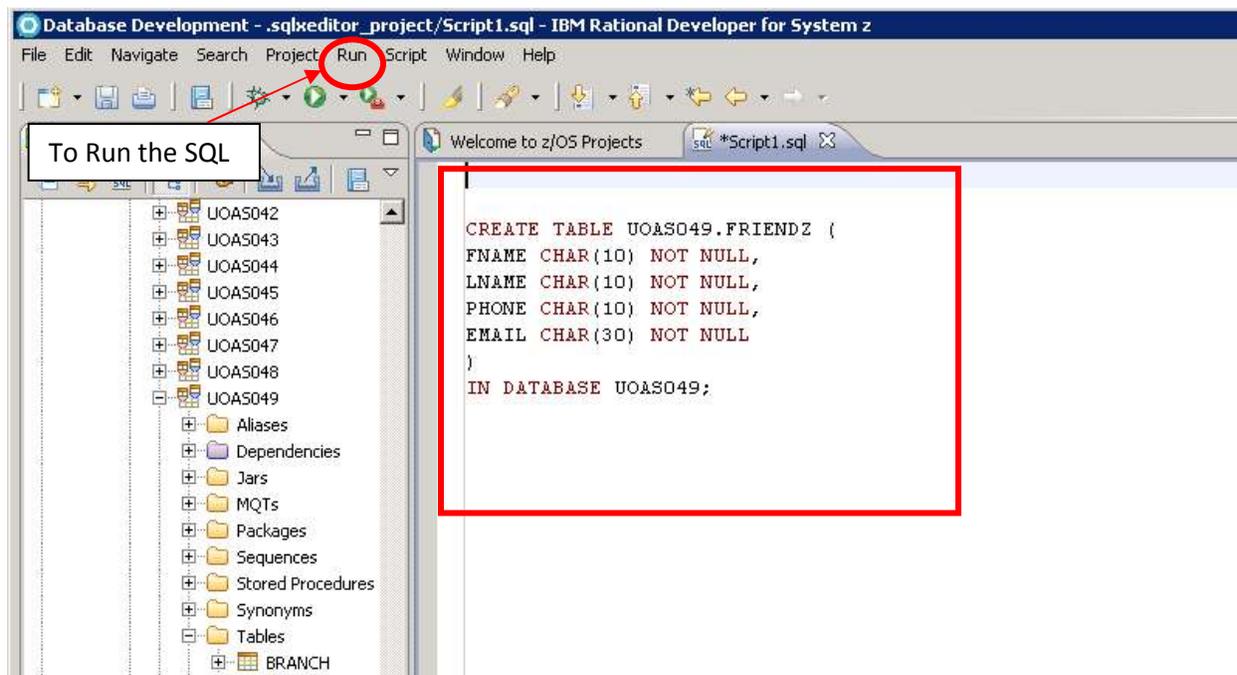
**Listing 1. SQL to create the table used in the example**

```

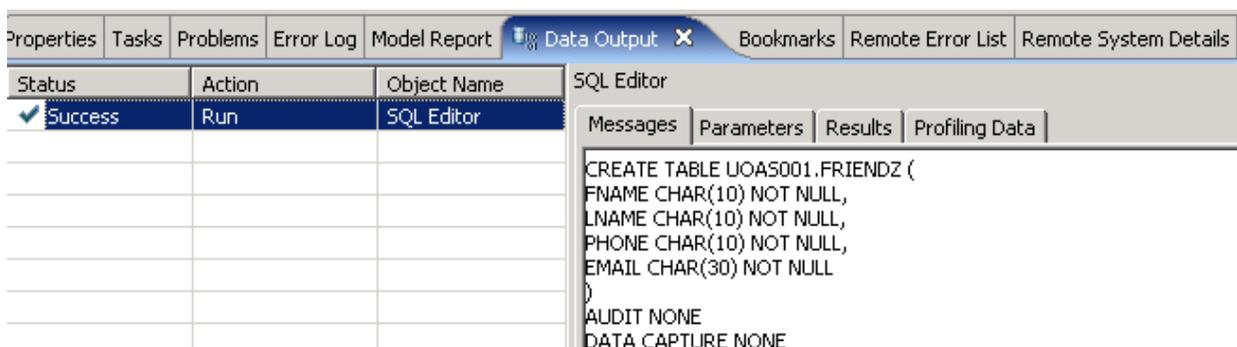
CREATE TABLE UOASxxx.FRIENDZ (
FNAME CHAR(10) NOT NULL,
LNAME CHAR(10) NOT NULL,
PHONE CHAR(10) NOT NULL,
EMAIL CHAR(30) NOT NULL
)
IN DATABASE UOASxxx                                (You will need to substitute your ID)

```

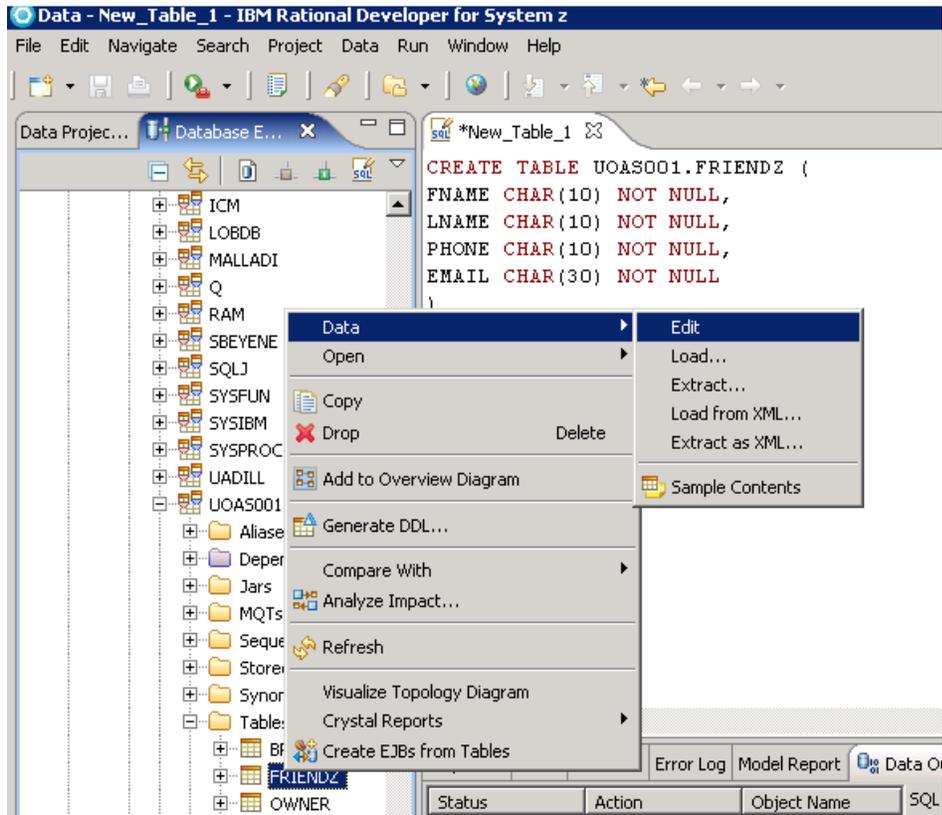
Notice that an example table create statement should appear in the SQL window in the center of the screen. You will want to overwrite these statements with the code provided in these exercises:



To run the SQL, go to the top menu bar selection: **RUN** → **Run SQL** or **Right-Click** in the SQL window and select **Run SQL**. You will see the results in the Data Output view, as shown below:



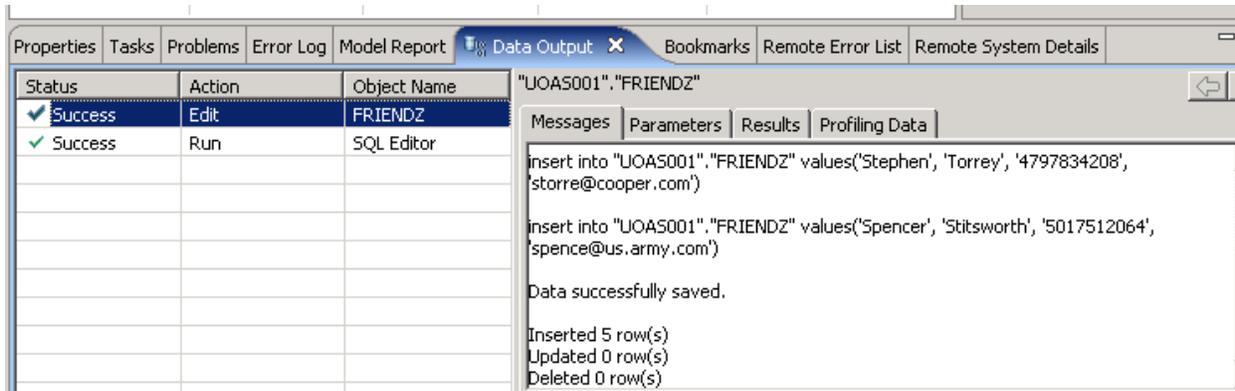
Inserting values into a table is very easy using the table editor within the data perspective. To use the table editor, find the table in the Database Explorer. You may need to refresh the Tables node to show the newly created table, as shown below. Highlight the table **FRIENDZ**, right-click, and then select **Data > Edit**. An edit window will open and you can type directly into the tabular form.



With the data editor open, you may add the information for 5 of your friends. I'm sure that they would appreciate it if you didn't post their real phone numbers ☺.

FNAME [CHAR(10)]	LNAME [CHAR(10)]	PHONE [CHAR(10)]	EMAIL [CHAR(30)]
David	Douglas	4795754500	ddouglas@uark...
Margaret	Llewellyn	4796529956	marge@hotmail...
Katie	McBride	4796482356	katie_turtle@ya...
Stephen	Torrey	4797834208	storre@cooper....
Spencer	Stitsworth	5017512064	spence@us.arm...
<new row>			

You need to save the document by clicking the disk icon on the main menu or using the two-key combination of Ctrl-S. You will see a confirmation in the Data Output view, as shown below:



The screenshot displays the RDz Data Output view. The left pane shows a table with the following data:

Status	Action	Object Name
✓ Success	Edit	FRIENDZ
✓ Success	Run	SQL Editor

The right pane shows the Data Output view for the object "UOAS001", "FRIENDZ". The Messages tab is selected, displaying the following output:

```
insert into "UOAS001"."FRIENDZ" values('Stephen', 'Torrey', '4797834208',
'storre@cooper.com')
insert into "UOAS001"."FRIENDZ" values('Spencer', 'Stitsworth', '5017512064',
'spence@us.army.com')
Data successfully saved.
Inserted 5 row(s)
Updated 0 row(s)
Deleted 0 row(s)
```