

PROGRESS EXCHANGE 2013

DISCOVER. DEVELOP. DELIVER.

OpenEdge MultiTenant Database Workshop

forgot to write something on this page



Progress Exchange 2013

Multitenant Workshop

Wednesday, October 9, 2013

Westin Boston Waterfront Hotel, Boston, MA

Welcome !

In this multi-tenant database programming workshop, you get to listen to the instructor's boring lectures and then you can try things for yourself to understand how things really work. There's no substitute for first-hand experience.

For this workshop we assume no prior knowledge of multitenant database concepts nor any familiarity with features new in OpenEdge 11. We do assume that you are familiar with some (not ancient) OpenEdge release and that you have some 4GL programming experience.

Each participant¹ has been asked to bring a portable computer equipped with wireless Internet access capability and Windows Remote Desktop (or a compatible program for some other operating system)

We have provided you with an already running Amazon EC2 virtual machine, preconfigured with Windows Server 2008, the OpenEdge 11.3.1 release and some additional artifacts for you to use in the various exercises.

In this virtual machine image, most of the OpenEdge product components have been installed. You will be using the 4GL compiler and runtime, the Enterprise Edition of the OpenEdge RDBMS, with multitenancy enabled, and the Database Administration Console section of OpenEdge Management.

In the directory "C:\mt" on your virtual machine you will find various files that we have graciously provided for you to use in the exciting and rewarding exercises. Here is a list of the files you will be using:

```
lab1.dfb  
lab1.st  
lab3.bak
```

¹ That means You !



forgot to write something on this page



LAB 0: Connecting to your OpenEdge 11 Virtual Machine

- Ladies and Gentlemen, start your computers!
Make sure your wireless network connection is working.
- Connect to the wireless network
- Open a browser (Google Chrome, Mozilla Firefox or Apple Safari)

For the morning multitenant workshop, please use this URL:

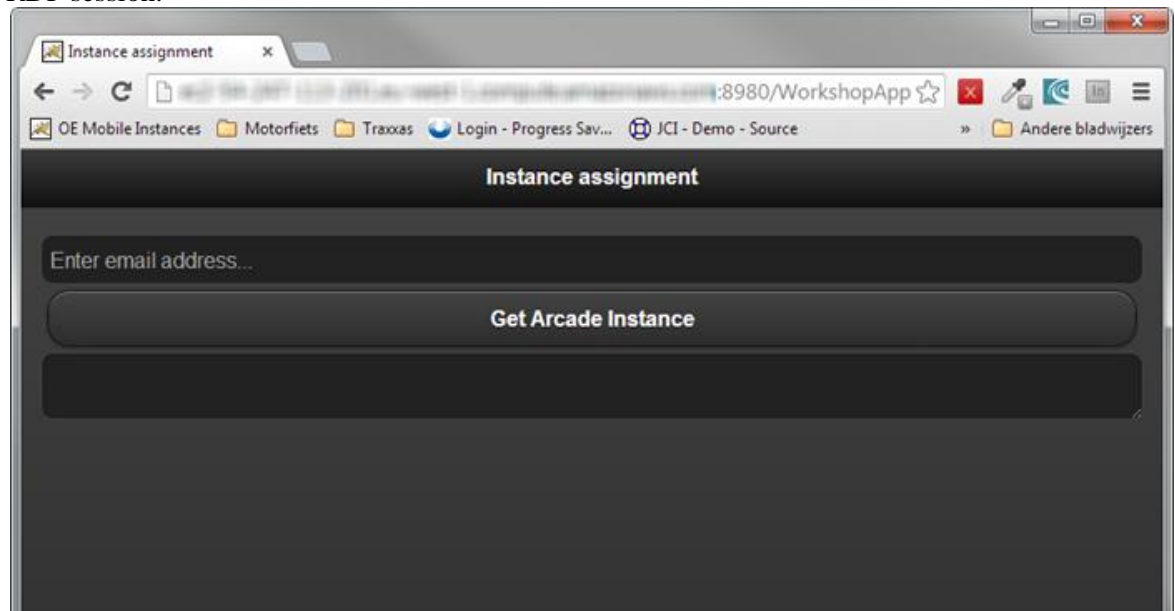
<http://23.23.210.136:8980/WorkshopApp>

For the afternoon multitenant workshop, please use this URL:

<http://54.225.237.144:8980/WorkshopApp>

- Enter your email address
- Press the button: Get Arcade Instance

This will give you a DNS for a running Arcade instance, which you can use to connect an RDP session.



- Start the Remote Desktop Connection utility program on you laptop.
In Windows 7 it is in Start->All Programs->Accessories->Remote Desktop Connection.
In Mac OS X, you may find a program called Remote Desktop Connection in the Applications folder. If you don't have it, you can download it at no cost from Microsoft.
- Enter the machine address that was displayed in your web browser. Type carefully (note: copy and paste may be easier and more accurate).



The Windows user name is "\Administrator"

The password is "Exchange2013".

Click the "Connect" button.

You should now see a Windows desktop from your Amazon virtual machine.

That concludes "lab 0". Now pay attention to the first lecture.



forgot to write something on this page



LAB 1: Creating a multitenant database

In this lab we will create a multitenant database. We will also use the same database in the next lab.

To create the multi-tenant database, we will begin by creating an empty database from “thin air”². The steps are summarized below and then described in detail.

1. Start a proenv command window
2. Create a directory called lab1
3. Get the storage definition file lab1.st.
4. Create a void database using the storage definition file.
5. Copy an empty database into the void database.
6. Enable multi-tenancy with the proutil program.
7. Load the table and index definitions from a .df file.
8. Start the database server using the proserve command.
9. Stop the database server again, using the proshut command.

LAB 1 Detailed Steps

1. Open an OpenEdge proenv command environment

- Double click on the “proenv” icon on the desktop or choose All Programs -> Progress -> OpenEdge11.3 -> Proenv from the Windows Start menu. This will open a command-line window with the environment variables you will need already set to the appropriate values. You will be in the “working directory”. This is called C:\OpenEdge\WRK
- Create a directory called “lab1” and go there

```
proenv> mkdir lab1
proenv> cd lab1
```

The name of the database we will work with is named “lab1”.

2. Get the storage definition file

When we add tenant definitions to the database later, we will have to specify the names of areas in which their data partitions will be allocated. The file C:\mt\lab1.st contains the following type i and type ii storage area definitions:

```
#
# lab 1 storage definition file
# basic areas
#
b .
d "Schema Area":6,32;1 .
d "Data":7,64;8 .
d "Indexes":8,64;8 .
```

² Actually, we will make a copy of an empty database from the DLC directory.




```

#
# Additional tenant partition data areas
# Each tenant gets their own data area
#
d "T1":101,64;8 .
d "T2":102,64;8 .
d "T3":103,64;8 .
d "T4":104,64;8 .
d "T5":105,64;8 .
d "T6":106,64;8 .
d "T7":107,64;8 .
d "T8":108,64;8 .
d "T9":109,64;8 .

```

Note that we have defined a data area for each tenant's stuff (T1 through T9).

- Copy the lab1.st file from the c:\mt directory into your lab1 directory.

3. Create the lab1 database

- Create the void database structure:

```
proenv> prostrct create lab1 lab1.st
```

- Copy the 4 kb block size empty database into the void structure you just created.

```
proenv> procopy %DLC%/empty4 lab1
```

Now you have an empty database that contains just schema table definitions and the virtual system tables but no tables of your own.

4. Enable multi-tenancy for the database

- Next, enable the database for multi-tenancy. This allows use of the various multitenant features but does not make any other changes to the database. Execute the following command:

```
proutil lab1 -C enablemultitenancy
```

5. Load table definitions

- Copy the table and index definitions from the workshop directory

```
proenv> copy C:\mt\lab1.df lab1.df
```

- Connect to the database in single-user mode

```
proenv> pro lab1
```

- In the Tools menu, choose the Data Dictionary tool
- In the Admin menu, choose Load Data and Definitions
- Load the file called lab1.df.



- Exit from the 4GL session.

6. Start the database server

```
proserve lab1 -B 5000
```

Check to ensure the database is working.

7. Stop the database server

```
proshut lab1 -by
```



LAB 2: Creating tenants, domains, and users

In this lab we will add some tenants, domains, and users to the database we created earlier. Here is a summary of the steps

1. Start the database server using the proserve command
2. Start the OpenEdge Management console

The user name is “admin” and the password is “admin”.

3. In OpenEdge Management set up a new database connection for the.
4. Define a super tenant.
5. Define a security domain for the super tenant
6. Define a user for the super tenant
7. Define some regular tenants
8. Define a security domain for each regular tenant.
9. Define a user for each regular tenant

1. Start the database server

For this lab, we will use the same database we created in the previous one.

```
proserve lab1 -B 5000
```

You do not need to use any other startup configuration parameters

2. Start the OpenEdge Management multi-tenant configuration tool.

- Double-click the Management Console icon on the desktop or by double-clicking the Management Console icon on your desktop or choose All Programs -> Progress -> OpenEdge11.3 -> Management Console from the Windows Start menu.

When you see the dialog asking for a user id and password, enter “admin” for the user name and “admin” for the password.

You will find yourself in the main page of the OpenEdge Management Tool.

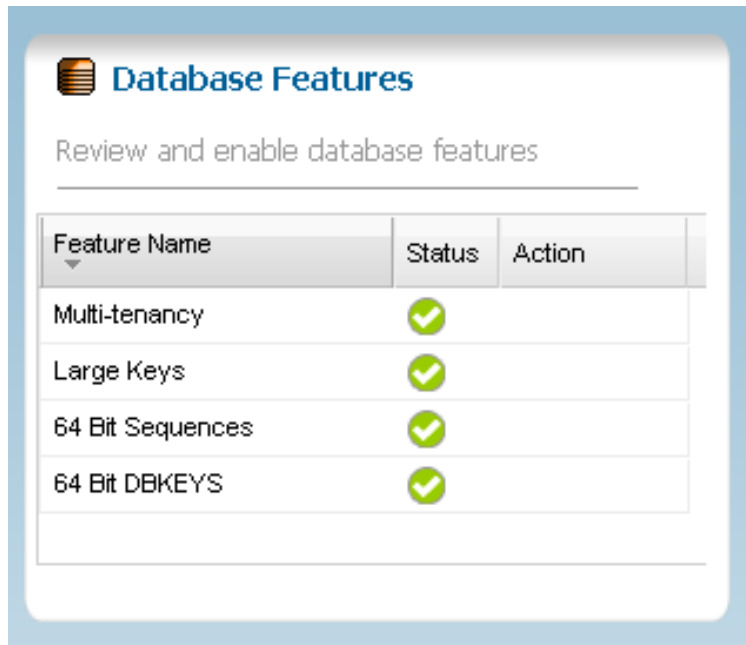
3. Connect to the database

We will connect to the unmanaged database, lab1, we just created in the previous lab.

- Click on the “Database Administration” tab. You will see a list of database connections on the left but the list will be empty.
- Click the “New” button.



- Enter “lab1” for the connection name, “C:\OpenEdge\WRK\lab1\lab1” for the database name, and select the only entry from the pick list for Container.
- Click the “Save” button. You should now see the database connection “lab1” listed in the panel on the left side. Note that this is an “unmanaged database” that is not managed by the OpenEdge Admin Server and that has no entries in its properties file. This is a capability that is new to OpenEdge Management.
- Click on "lab1" in panel on the left. You should see the “Database Features” pane illustrated below.



4. Create OpenEdge 11 tenant definitions

Next, we will add a few tenant definitions.

- Click on the arrow of the “Database Administration” tab and select New->Tenant.

4.1.1. Create the super tenant

- Add a new tenant that will be able to access all the data. This tenant’s name will be “super” and the tenant type will be “Super”. The panel will look like this when you have filled it in (except the connection name will be "lab1" instead of "aetf"):



Tenant name:

Description:

External ID:

Tenant type: Regular Super

Database connection:

Default data area:

Default index area:

Default LOB area:

Object allocation rule: Set new objects not to allocate space
 Set new objects to allocate space on tenant creation
 Set new objects to delay space allocation

Template: Do not use template
 Copy defaults from template

--None--

- Click on “Create tenant” to create the super-tenant.

4.1.2. Create a domain for the super tenant

- Click on “edit tenant”. Now we will add a security domain and a user for the super-tenant.
- Click on “Domains” and then on “New”.
- For the domain name, enter “super”. Choose `_oestable`³ for the Authentication system. Enter “letmein”⁴ for the access code. Confirm the access code.
- Click on “Commit”. This will create the domain.

4.1.3. Create a user for the super tenant

- Now click on “Users” and then on “new”.
- Enter “super” in the “User name” field. Choose “super” from the “Domain name” list. Enter “letmein” for the password. Confirm the password.
- Click on “Commit”. This will create the user "super@super".

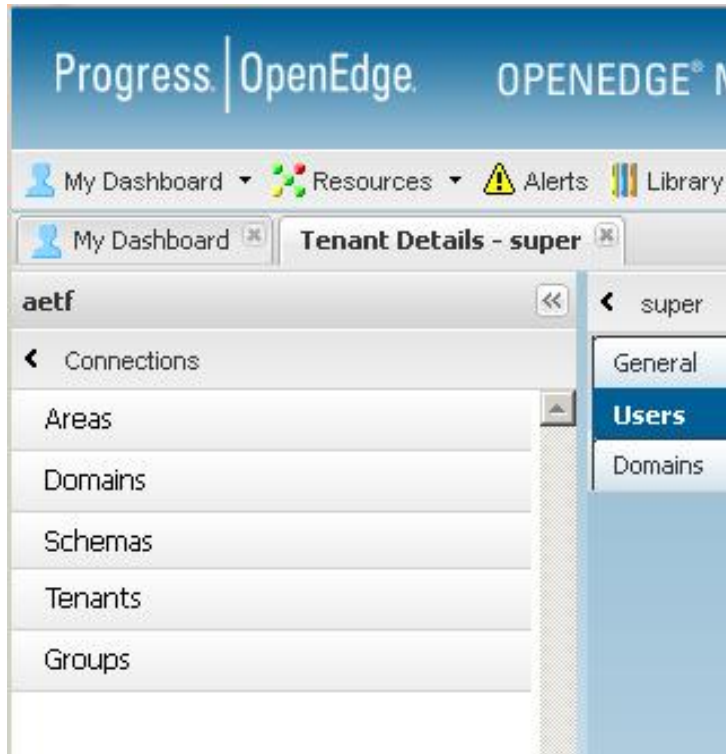
4.1.4. Create a regular tenant "chery"

- Click on the “Database connect details” tab

³ This means the database `_user` table will be used, as opposed to the other possibilities.

⁴ You can of course use whatever password if you prefer.

- Click on the database name "lab1" in the panel on the left side of the window. You should see stuff that looks like this:



- Create a regular tenant named "chery" in the same manner as the steps you followed a few minutes ago when you created the super-tenant.

The panel below shows what the definition for chery should look like (except the connection name will be "lab1" instead of "aetf") . Note that the area name T1 appears in 3 places. These are the storage area names for the areas for table partitions, index partitions, and LOB data partitions for the tenant chery.

To enter the 3 area names, do the following:

- Click on the little magnifying glass to the right of the area name fillin field for the tenant's default data area. You will see a filter dialog.
- Click on the "Filter" button. You will see a list of are names.
- Double click on the one called "T1". You will see the dialog disappear and the panel will return with "T1" in the fillin box.
- Click on the index area name fillin box and then enter T1.
- Click on the LOB area name fillin box and enter T1.
- Click in the "Description" fillin field.



- Finally, click on the "Create Tenant" button. The tenant named "chery" will be created. Note that if you click on "Generate tenant program" instead of "Create tenant", the new tenant *will not be created*. Instead, a 4GL program that creates the tenant will be generated and saved to a file for you to use later.

Now we will add a security domain and a user for the tenant "chery":

4.1.5. Add a domain for the tenant chery

- Click on "edit tenant".
- Click on "Domains" and then on "New".
- For the domain name, enter "chery".
- Choose _ousercontentable for the Authentication system. Enter "letmein" for the access code.
- Click on "Commit". Now the domain is created.

4.1.6. Add a user for the tenant chery

- Now click on "Users" and then on "new". Enter "bob" in the "User name" field. Choose "chery" from the "Domain name" list. Enter "letmein" for the password.
- Confirm the password



- Click on “Commit”. Now the user bob@chery will be created.

4.1.7. We are going to need more tenants, domains, and users.

Add the remaining regular tenants listed in the table that follows. You will follow the same steps as when you created the regular tenant "chery".

For each tenant listed (well, actually you can stop after you have finished with the one called "genericmotors" because we do not need them all) do the following 3 things:

- a) Create a regular tenant using the tenant name and area name given in the table, in the same manner as you created the tenant "chery"
- b) To the tenant, add a domain with the same name as the tenant
- c) To the domain, add a user named "bob" with a password of "letmein".

Note that the area names in the third column below are the same as those we used in the storage definition file you used to create the database.

Tenant Name	Tenant Type	Default data area name for tables, indexes, LOBs
chery	Regular	T1
fjord	Regular	T2
genericmotors	Regular	T3
h2omotors	Regular	T4
hinda	Regular	T5
motorgroupeast	Regular	T6
potomoc	Regular	T7
scubaroo	Regular	T8
toyola	Regular	T9

5. Check your work

Once you have created all of the tenants, in a 4GL session, execute the following code to verify that you got them all:

```
proenv> cd C:\OpenEdge\WRK\lab1
proenv> mpro lab1 -U super@super -P letmein
```

then

```
for each _tenant:
  display _tenant._tenantId _tenant._tenant-name.
end.
```



All the tenants you created should be shown. And one you didn't create and whose name is "Default". More about that later.

6. Put your toys away

- End your 4GL session.

```
quit.
```

- Close your browser window in the usual way to exit the Database Administration Console (DBAC).

- Stop the database server by executing this command:

```
proenv> proshut lab1 -by
```

- Close your proenv window.

You have now finished lab 2.

You may experiment while you wait for the others to finish.



forgot to write something on this page



LAB 3: Looking at the tenant's data

In this lab we will take a quick look at some data by running the same program a couple of times.

- Start a proenv window.
- Create a directory for lab 3 in the work directory C:\OpenEdge\WRK.

```
proenv> mkdir lab3
proenv> cd lab3
```

- Create a database by restoring from a backup we have provided for you.

```
proenv> prorest lab3 c:\mt\lab3.bak
```

This database contains a number of tenants, domains, and users, and some tables and data.

- Connect to the database (single user mode) as a user belonging to a regular tenant

```
proenv> pro lab3 -U bob@chery -P letmein
```

- Now execute the following trivial program:

```
for each customer:
  display custnum name.
end.
```

Note what data you see.

- Now exit from your 4GL session.
- Connect to the database again, but as a different user of another tenant

```
proenv> pro lab3 -U bob@fjord -P letmein
```

- Now execute the trivial program again

```
for each customer:
  display custnum name.
end.
```

Note what data you see.

You are finished with lab 3.

- Now exit from your 4GL session
- Close your proenv window.



LAB 4: Playing with CLIENT-PRINCIPAL

Now we will use the CLIENT-PRINCIPAL to determine our identity.

1. Start a proenv session

- Go to the directory we used before for lab 3.

```
proenv> cd C:\OpenEdge\WRK\lab3
```

- Connect to the database

```
proenv> pro lab3 -U bob@chery -P letmein
```

If you execute this program again,

```
for each customer:
  display custnum name.
end.
```

You will see chery's data as you did before.

2. Now create a client-principal token

- Run the following program:

```
def var hCp as handle.

create client-principal hCp.
hcp:initialize("bob@fjord").
hcp:seal("letmein").
set-db-client(hCp).

for each customer:
  display custnum name.
end.
```

Now you should see tenant fjord's data.

3. Look at some of the token's properties

The client-principal has quite a few inbuilt properties. Look at some of their values.

```
def var hCp as handle.

create client-principal hCp.
hcp:initialize("bob@fjord").
hcp:seal("letmein").
```



```

set-db-client(hCp).
MESSAGE
  "Database 'aetf' UserID" USERID('aetf') SKIP(2)
  "SESSION-ID" hCp:SESSION-ID SKIP
  "USER-ID" hCp:USER-ID SKIP
  "DOMAIN-NAME" hCp:DOMAIN-NAME SKIP(1)
  "AUDIT-EVENT-CONTEXT" hCp:AUDIT-EVENT-CONTEXT SKIP
  "CLIENT-TTY" hCp:CLIENT-TTY SKIP
  "CLIENT-WORKSTATION" hCp:CLIENT-WORKSTATION SKIP
  "DB-LIST" hCp:DB-LIST SKIP
  "DOMAIN-DESCRIPTION" hCp:DOMAIN-DESCRIPTION SKIP
  "DOMAIN-TYPE" hCp:DOMAIN-TYPE SKIP
  "INSTANTIATING-PROCEDURE" hCp:INSTANTIATING-PROCEDURE SKIP
  "LOGIN-EXPIRATION-TIMESTAMP" hCp:LOGIN-EXPIRATION-TIMESTAMP
SKIP
  "LOGIN-HOST" hCp:LOGIN-HOST SKIP
  "LOGIN-STATE" hCp:LOGIN-STATE SKIP
  "QUALIFIED-USER-ID" hCp:QUALIFIED-USER-ID SKIP
  "ROLES" hCp:ROLES SKIP
  "SEAL-TIMESTAMP" hCp:SEAL-TIMESTAMP SKIP
  "STATE-DETAIL" hCp:STATE-DETAIL SKIP
  "TYPE" hCp:TYPE SKIP(1)
  "LIST-PROPERTY-NAMES" hCp:LIST-PROPERTY-NAMES() SKIP(1)
  "TENANT-ID" hCp:TENANT-ID() SKIP
  "TENANT-NAME" hCp:TENANT-NAME()
VIEW-AS ALERT-BOX TITLE "After SEAL".

```

- Exit from your 4GL session.
- Close your proenv window.

You are now finished with lab 4.

Feel free to experiment while you wait for the others to finish.



LAB 5: Looking at data as the super-tenant.

- Go to the directory we used before for lab 3.

```
proenv> cd C:\OpenEdge\WRK\lab3
```

- Connect to the database as the super-tenant

```
proenv> pro lab3 -U super@super -P letmein
```

- Now execute this familiar program again,

```
for each customer:  
  display custnum name.  
end.
```

What do you see?

- Now try this:

```
for each customer  
  tenant-where tenant-id() > 0:  
  display custnum  
    name.  
end.
```

- and this:

```
for each customer  
  tenant-where tenant-id() > 0:  
  display BUFFER-TENANT-NAME (customer)  
    custnum  
    name.  
end.
```

- and this one:

```
SET-EFFECTIVE-TENANT ("fjord").  
for each customer:  
  display custnum  
    name.  
end.
```

You can also write a program for use by regular and super tenants. It should be obvious by now that it can do different things depending on who you are. For example:

```
if tenant-id() <> -1 then do:  
/*
```



```

    we are a user belonging to a regular tenant and can see
    only our own tenant data.
    all regular tenant users run this same part of the code
*/
display "Tenant" tenant-name() today string(time, "HH:MM")
    with frame theHeader centered.
for each customer:
    display custnum name
        with frame theData down centered
        title "Customer List".
end.
end.
else do:
/*
    we are a user belonging to the super-tenant and can see
    everyone's data
    supertenant users run this part
*/
display "All tenants" today string(time, "HH:MM")
    with frame superHeader centered.
for each customer tenant-where tenant-id() > 0:
    display buffer-tenant-name(customer) label "Tenant"
        custnum
        name
        with frame superData down centered
        title "Customer List".
end.
end.
pause.
quit.

```

- For extra credit, do the first program again but set up a CLIENT-PRINCIPAL for the super tenants before the for each
- Exit from your 4GL session.
- Close your proenv window.

You are finished with lab 5.

Feel free to experiment while you wait for the others to finish.





Progress Software

Progress Software Corporation (NASDAQ: PRGS) is a global software company that simplifies the development, deployment and management of business applications on premise or on any Cloud, on any platform and on any device with minimal IT complexity and low total cost of ownership.

Worldwide Headquarters

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA

Tel: +1 781 280-4000 Fax: +1 781 280-4095 On the Web at: www.progress.com

For regional international office locations and contact information, please refer to the Web page below: www.progress.com/worldwide

Progress and [ALL PRODUCT NAMES LISTED IN PUBLICATION] are trademarks or registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and other countries. Any other marks contained herein may be trademarks of their respective owners. Specifications subject to change without notice.

© 2013 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

