Apex

(Introduction to Apex)
Exercise Guide

Table of Contents

Exercise 0-2: Creating a Sandbox Organization

Goal:

Create a development environment that resembles production.

Scenario:

Universal Containers developers have successfully deployed their application to production. However, now the customers are so pleased that they want to add more features and change the current features.

Tasks:

1. Customize the admin user to use your email.
2. Download your lab files.
3. Create a Sandbox (Full) organization.

Time:

5 minutes

Instructions:

1. If you have not already done so, customize the admin user to use your email

   A. Navigate to *Your Name* **| My Settings | Personal | Personal Information**

   B. Modify the **Email** field to be an email you can access during class

   C. Click **Save**

2. Download your lab files.

   A. In your org, navigate to the **Documents** tab.

   B. Ensure that `Shared Documents` is selected for the **Folder** option, and click **Go!.**

   C. Select the **View** option for `LabFilesForDEV501.`

   D. When prompted, download the zip file to your Desktop.

   E. Unpack the zip file to a folder on the Desktop.

3. Create the sandbox (full) organization

   A. Navigate to **Setup | Data Management | Sandbox**

   B. Click **New Sandbox**

   C. On this screen, where you add Sandbox Information and choose the Sandbox License:

      i. **Name:** `DeOrgStudentID`

    ii.   **Description:** This is a development environment

    iii. **Type:** Full (Warning: Make sure you select this correctly!)

    iv. Click **Next**

D. On this screen, where you choose your Sandbox Options:

    i.   **Object data included:** All

    ii. **Include Chatter Data:** Selected

    iii. Click **Create**

Note: The time it takes to complete the full copy will depend on the length of the queue at the time requested. You will receive an email letting you know when it has been completed.

Review

1.   Why are you creating a sandbox?

_____

_____

_____

Exercise 1-1: Retrieving the Force.com Apex Code Developer's Guide

Goal:

Retrieve the Force.com Apex Code Developer's Guide PDF version and view the HTML version.

Scenario:

Developers can access detailed information about the Apex language available at salesforce.com. The Force.com Apex Developer's Guide provides a resource that can be referenced for questions regarding Apex language.

Tasks:

1. Open a browser window and navigate to the Developer Force website.
2. Download the Force.com Apex Code Developer's Guide PDF file.
3. View the HTML version.

Time:

10 minutes

Instructions:

1. Open a browser window and navigate to the developerforce website:

   http://developer.force.com.

2. Download the Force.com Apex Code Developer's Guide PDF file.

   A. From the site menu, navigate to **Technical Library | Documentation.**

   B. Under the Reference section, expand **App Logic | Apex Code.**

   C. Right-click on the PDF link next to **Apex Code | Developer's Guide**, then save the PDF to your desktop by choosing Save Target As…

3. View the HTML version: Click on the **HTML** link to view the web-based version of the Force.com Apex Code Developer's Guide.

Review

1. What type of information can you find in the Reference section of the documentation?

   _____

   _____

   _____

2. Does the Apex documentation contain the full specifications for SOQL syntax?

   _____

   _____

   _____

Exercise 1-2: Installing and Configuring the Force.com IDE

Goal:

Install and configure the Force.com IDE.

Scenario:

Universal Containers wants to use the Force.com IDE for its development needs. The company needs to understand how to install and configure the Force.com IDE before it can begin developing in its Salesforce org. Note: Supporting sites for the Force.com IDE are located at developer.force.com under the Tools section.
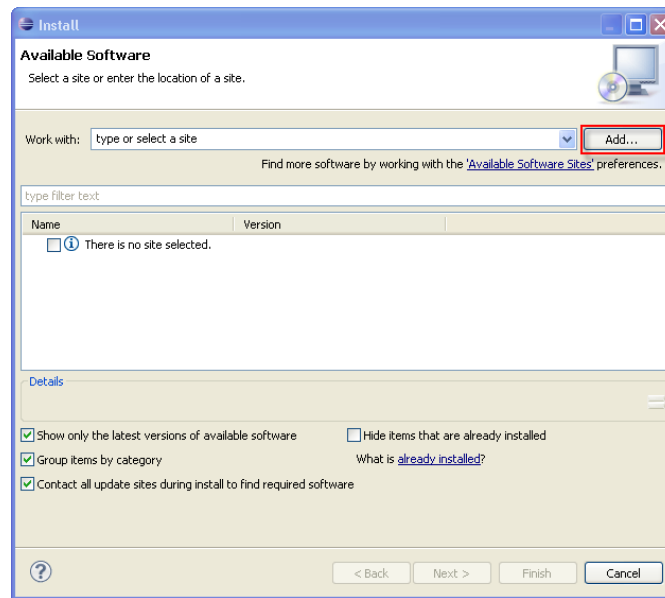http://wiki.developerforce.com/index.php/Force.com_IDE_nstallation_for_Eclipse_3.6

Tasks:

1. Add a Remote Site to Eclipse.
2. Download and install the Force.com IDE plug-in from the remote site.
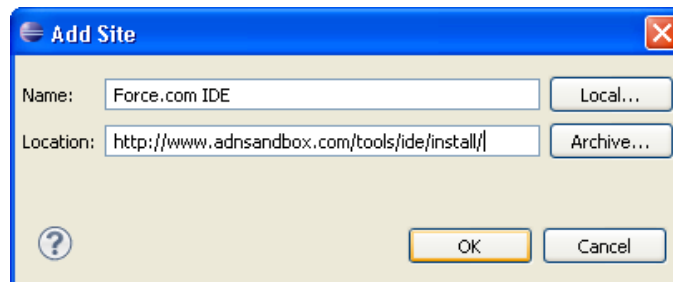3. Restart Eclipse.

Time:

30 minutes

Instructions:

1. Add a Remote Site to Eclipse.

    A. Double-click the Eclipse icon to launch Eclipse.

    B. In Eclipse, navigate to **Help | Install New Software…**
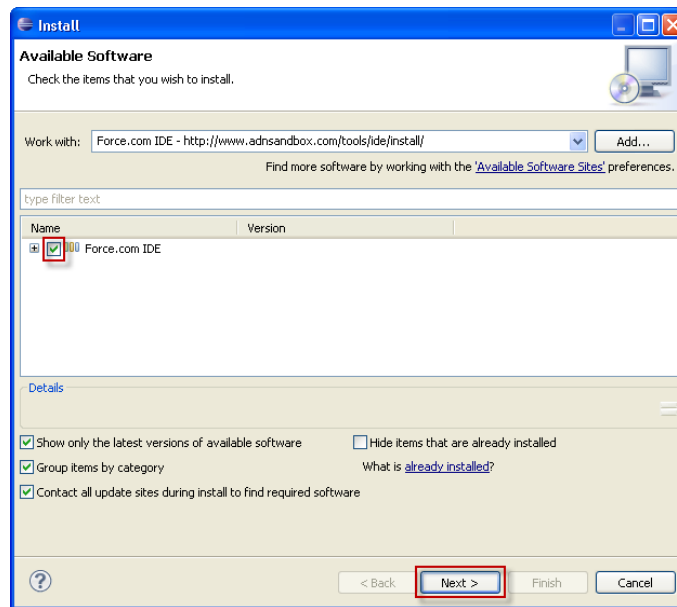
C. Click **Add**…

D. In the Add Site dialog box, type the following:

    i. **Name:** `Force.com IDE`

    ii. **Location:** `http://www.adnsandbox.com/tools/ide/install/`

E. Click **OK.**

2. Download and install the Force.com IDE Plug-In from the remote site.

A. Select the checkbox for `Force.com IDE`

B. Click **Next.**

C. In the **Install Details** dialog box, click **Next** again.

D. Select `I accept the terms`**,** and click **Finish.**

3. Restart Eclipse

A. After it takes a few minutes to install, Eclipse will prompt you to restart the application. Click **Yes.**

B. After Eclipse restarts, select **Window | Open Perspective | Other** and select **Force.com**. Click **OK.**

_____

Review

1. Do you have to use the Force.com IDE for Apex development? Is there another option?

   _____

   _____

   _____

2. After developing Apex code in a sandbox, how can the Force.com IDE help you move that code to production?

   _____

   _____

   _____

Exercise 1-3: Creating a Force.com IDE Project

Goal:

Create a Force.com Project using the Force.com IDE.

Scenario:

Universal Containers needs to create an environment for developers to be able create and update Salesforce code. To get their environment ready, they will need to prepare a project workspace within the Force.com IDE.

Tasks:

1. Create a new Force.com project.
2. Retrieve the Universal Containers organization code into the project.

Time:

15 minutes

Instructions:

1. Double-click the Eclipse shortcut on the desktop to launch Eclipse.

2. On the Eclipse menu bar, select **Window | Open Perspective | Other | Force.com** and then click **OK.**

3. Navigate to **File | New | Force.com Project.**

4. Enter the following information on the **Create New Force.com Project** dialog box:

    A. **Project name:** (enter your salesforce org username as the project name)

    B. **Username:** (enter your Salesforce username for this course, appended with '.dvorg')

    C. **Password:** (enter your salesforce password for this course)

    D. **Security Token:** (leave blank unless instructed differently)

    E. **Environment**: `Sandbox.`

        Note: Remember that if you are creating a project in sandbox, you will need to use those credentials for this project.

    F. Click **Next.**

    G. Select `Selected metadata components` and click **Choose…**

    H. Click **Select All** to choose all metadata component types and click **OK.**

  l. Click **Finish.**

5. Right-click the newly created project in the left-hand navigation pane and select **Properties.** Review the contents of the window and click **OK** when done.

_____

Review

1. When would we use the Security Token in the connection information?

 _____

 _____

 _____

2. When would we want to select `None` as our default project contents?

 _____

 _____

 _____

3. How do we ensure we see the specific menu items and layout associated with Force.com?

 _____

 _____

 _____

Exercise 1-4: Examining a Hello World Example

Goal:

Examine an Apex code sample and answer some discussion questions about it.

Scenario:

Universal Containers is new to Apex, so its developers want to start with the basics. Analyze the class and trigger and clarify some questions that they have.

Tasks:

Examine the following class and trigger code and answer the following questions.

Time:

10 minutes

Instructions:
1. Examine the following class and trigger code and answer the following questions.

Class

```
C1 public class HelloWorldPositionClass {
C2    public static void helloWorld(List<Position__c> positions){
C3      for (Position__c p:positions){
C4        if (p.Hello__c != 'World') {
C5          p.Hello__c =    'World';
C6        }
C7      }
C8    }
C9 }
```

Trigger

```
T1 trigger HelloWorldPositionTrigger on Position__c (before insert,
                                     before update) {
T2   List<Position__c> positions = Trigger.new;
T3   HelloWorldPositionClass.helloWorld(positions);
T4 }
```

Discussion Questions

1. What does this code do? Log in to your org and create and save some positions to verify this.

   _____

   _____

   _____

2. What do `before insert` and `before update` indicate in line `T1`? What other options are likely available?

   _____

   _____

   _____

3. What is `Trigger.new` in line `T2`? What does it contain?

   _____

   _____

   _____

4. What does `List<Position__c>` represent? Why is there `__c` after `Position`?

   _____

   _____

   _____

5. Why is it that the `helloWorld` method is called in line `T3` directly from the class, instead of being instantiated?

   _____

   _____

   _____

6. How many times does the `FOR` loop in line `C3` execute?

   _____

   _____

   _____

7. What does `p.Hello__c` represent? Why is it `p.Hello__c` instead of `positions.Hello__c`?

_____

_____

_____

8. Did we need the class here or could we have made this work just in a trigger? What might be a reason to architect the functionality with a class?

_____

_____

_____

9. What are some examples of ways you could use triggers?

_____

_____

_____