

→ → → →

Step 1: Click on Setup → Create Objects → New custom object
Label: MySale

Pular Label: MySales

Object Name: MySale

Record Name: MySale Description

Data Type: Text

→

Click on Save.

→

Step 2: Under MySale Go to Custom Field and Relationships → Click on New Custom Field

Creating 1st Field:-

→ select Data type as Auto Number → next
→ Enter the details → Field Label: PROD_ID → Display Format: MYS-{0000} →
→ Starting Number: 1001 → Field Name: PRODID → Next → Save & New

Creating 2nd Field:-

→ select Data type as Date → next
→ Enter the details → Field Label: Date of Sale → Field Name: Date_of_Sale
→ Default Value: Today()-1 → Next → Save & New

Creating 3rd Field:-

→ select Data type as Number → next
→ Enter the details → Field Label: Quantity Sold → Length:3 → Decimal places:0
→ Default Value: Show Formulae Editor:1 → Next → Save & New

Creating 4th Field:-

→ → select Data type as Currency next
→ → Enter the details → Field Label: Rate → Field Name: Rate → Length:4 → Decimal places:2
→ → Default Value: 10 → Next → Save & New

Creating 5th Field:-

→ → select Data type as Currency next
→ → MySale field → Quantity_Sold_c*Rate_c → next → save.

Now create an App

→ → → → → → → → → → → →
Setup Create App new → MyShop → Next → Select an Image → Next → Add Object MySales.

Now create an Tab

→ → → → → → → → → → → →
Setup Create Tab → New Custom Tab → Choose MySales object → select tab style → save.

On the top in the tab bar you can see the tab which has been created by you click on the tab you can see your object is opened just click on new button and provide the details mentioned.

Aim: Creating an Application in SalesForce.com using Apex programming Language.

Theory: Step1:

Log into your Sandbox or Developers Organization.

→ → →

Click on setup create objects new custom objects.
Enter Book for label.

Enter Books for plural label.

Click Save.

Step 2:

Now let's create a custom field.

In the custom field & relationship section of the Book Object click new.

Select Number for the datatype & next.

Enter Price for the field Label.

Enter 16 in the length text box.

Enter 2 in the decimal places & Next....next.... save.

Step 3:

→ →

Chilck setup Develop Apex Classes & click new
In the class Editor enter this class

```
public class MyHelloWorld{  
    public static void applyDiscount(Book_c[] books)  
    {  
        for(Book_c b:books)
```

```
{b.Price_c*=0.9;}  
}  
}
```

Step 4:

Add a trigger

A trigger is a piece of code that can execute objects before or after specific data manipulation language events occurred.



Click on setup → create → objects → click the object you have created ex:
Book Scroll down you can see Trigger Click on New

In the trigger Editor enter this class

```
trigger HelloWorldTrigger on Book_c(before insert)
```

```
{
```

```
    Book_c[] books=Trigger.new;
```

```
    MyHelloWorld.applyDiscount(books);
```

```
}
```

Step 5:



Click on setup → create → tabs → new custom tab → choose Book → next&.next&..save.



Click on tab Books → new → insert a name for Book → insert price for that book → click on save.

Experiment Title: Implementation of Para-Virtualization using VM Ware's Workstation/ Oracle's Virtual Box and Guest O.S.

Aim: Implementation of Virtual Box for Virtualization of any OS.

Some Terminologies used:

When dealing with virtualization (and also for understanding the following chapters of this documentation), it helps to acquaint oneself with a bit of crucial terminology, especially the following terms:

Host operating system (host OS). This is the operating system of the physical computer on which Virtual Box was installed. There are versions of Virtual Box for Windows, Mac OS X, Linux and Solaris hosts.

Guest operating system (guest OS). This is the operating system that is running inside the virtual machine. Theoretically, Virtual Box can run any x86 operating system (DOS, Windows, OS/2, FreeBSD, Open BSD), but to achieve near-native performance of the guest code on your machine, we had to go through a lot of optimizations that are specific to certain operating systems. So while your favorite operating system *may* run as a guest, we officially support and optimize for a select few (which, however, include the most common ones).

Virtual machine (VM). This is the special environment that Virtual Box creates for your guest operating system while it is running. In other words, you run your guest operating system "in" a VM. Normally, a VM will be shown as a window on your computers desktop, but depending on which of the various frontends of VirtualBox you use, it can be displayed in full screen mode or remotely on another computer. In a more abstract way, internally, VirtualBox thinks of a VM as a set of parameters that determine its behavior. They include

hardware settings (how much memory the VM should have, what hard disks VirtualBox should virtualize through which container files, what CDs are mounted etc.) as well as state information (whether the VM is currently running, saved, its snapshots etc.). These settings are mirrored in the VirtualBox Manager window as well as the VBoxManage command line program;

Guest Additions. This refers to special software packages which are shipped with VirtualBox but designed to be installed *inside* a VM to improve performance of the guest OS and to add extra features.

Starting Virtual Box:

After installation, you can start VirtualBox as follows:

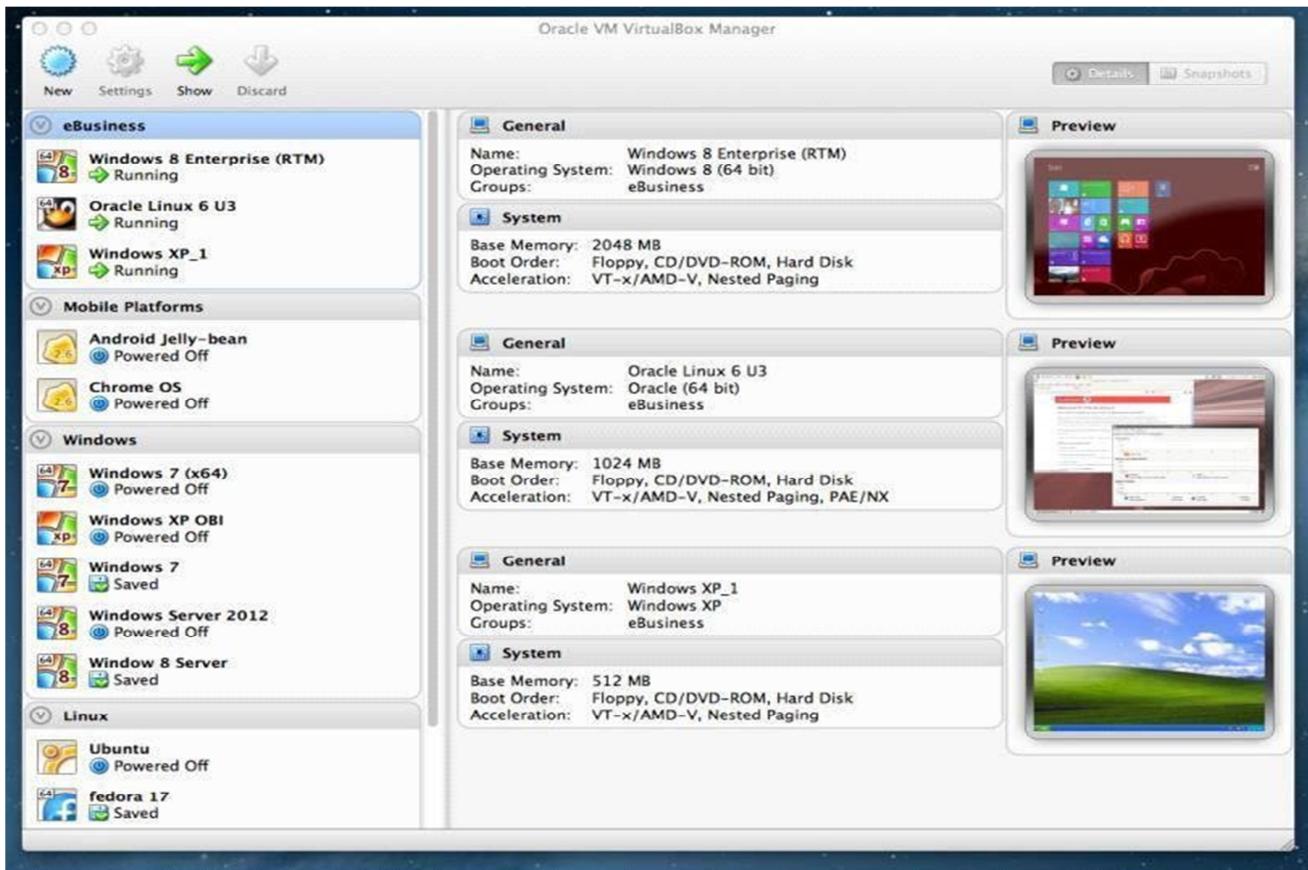
- On a Windows host, in the standard "Programs" menu, click on the item in the "VirtualBox" group. On Vista or Windows 7, you can also type "VirtualBox" in the search box of the "Start" menu.
- On a Mac OS X host, in the Finder, double-click on the "VirtualBox" item in the "Applications" folder. (You may want to drag this item onto your Dock.)
- On a Linux or Solaris host, depending on your desktop environment, a "VirtualBox" item may have been placed in either the "System" or "System Tools" group of your "Applications" menu. Alternatively, you can type VirtualBox in a terminal.

When you start VirtualBox for the first time, a window like the following should come up:



This window is called the "**VirtualBox Manager**". On the left, you can see a pane that will later list all your virtual machines. Since you have not created any, the list is empty. A row of buttons above it allows you to create new VMs and work on existing VMs, once you have some. The pane on the right displays the properties of the virtual machine currently selected, if any. Again, since you don't have any machines yet, the pane displays a welcome message.

To give you an idea what VirtualBox might look like later, after you have created many machines, here's another example:



Creating your first virtual machine:

Click on the "New" button at the top of the VirtualBox Manager window. A wizard will pop up to guide you through setting up a new virtual machine (VM)



On the following pages, the wizard will ask you for the bare minimum of information that is needed to

create a VM, in particular:

- The **VM name** will later be shown in the VM list of the VirtualBox Manager window, and it will be used for the VM's files on disk. Even though any name could be used, keep in mind that once you have created a few VMs, you will appreciate if you have given your VMs rather informative names; "My VM" would thus be less useful than "Windows XP SP2 with OpenOffice".
- For "**Operating System Type**", select the operating system that you want to install later. The supported operating systems are grouped; if you want to install something very unusual that is not listed, select "Other". Depending on your selection, Virtual Box will enable or disable certain VM settings that your guest operating system may require. This is particularly important for 64-bit guests (see [Section 3.1.2, "64-bit guests"](#)). It is therefore recommended to always set it to the correct value.

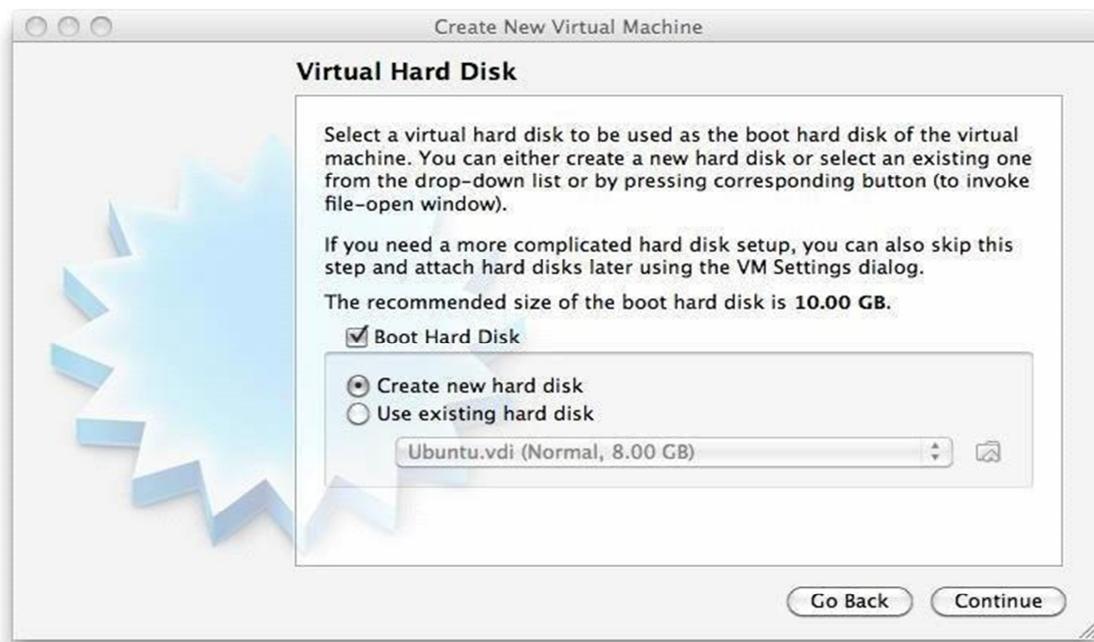
- On the next page, select the **memory (RAM)** that Virtual Box should allocate every time the virtual machine is started. The amount of memory given here will be taken away from your host machine and presented to the guest operating system, which will report this size as the (virtual) computer's installed RAM.

A Windows XP guest will require at least a few hundred MB RAM to run properly, and Windows Vista will even refuse to install with less than 512 MB. Of course, if you want to run graphics-intensive applications in your VM, you may require even more RAM.

So, as a rule of thumb, if you have 1 GB of RAM or more in your host computer, it is usually safe to allocate 512 MB to each VM. But, in any case, make sure you always have at least 256 to 512 MB of RAM left on your host operating system. Otherwise you may cause your host OS to excessively swap out memory to your hard disk, effectively bringing your host system to a standstill. As with the other settings, you can change this setting later, after you have created the VM.

4. Next, you must specify a **virtual hard disk** for your VM. There are many and potentially complicated ways in which VirtualBox can provide hard disk space to a VM (see [Chapter 5, *Virtual storage*](#) for details), but the most common way is to use a large image file on your "real" hard disk, whose contents VirtualBox presents to your VM as if it were a complete hard disk. This file represents an entire hard disk then, so you can even copy it to another host and use it with another VirtualBox installation.

The wizard shows you the following window:



Here you have the following options:

- To create a new, empty virtual hard disk, press the "New" button.
- You can pick an **existing** disk image file. The **drop-down list** presented in the window contains all disk images which are currently remembered by VirtualBox, probably because they are currently attached to a virtual machine (or have been in the past). Alternatively, you can click on the small **folder button** next to the drop-down list to bring up a standard file dialog, which allows you to pick any disk image file on your host disk.

Most probably, if you are using VirtualBox for the first time, you will want to create a new disk image. Hence, press the "New" button. This brings up another window, the "**Create New Virtual Disk Wizard**", which helps you create a new disk image file in the new virtual machine's folder.

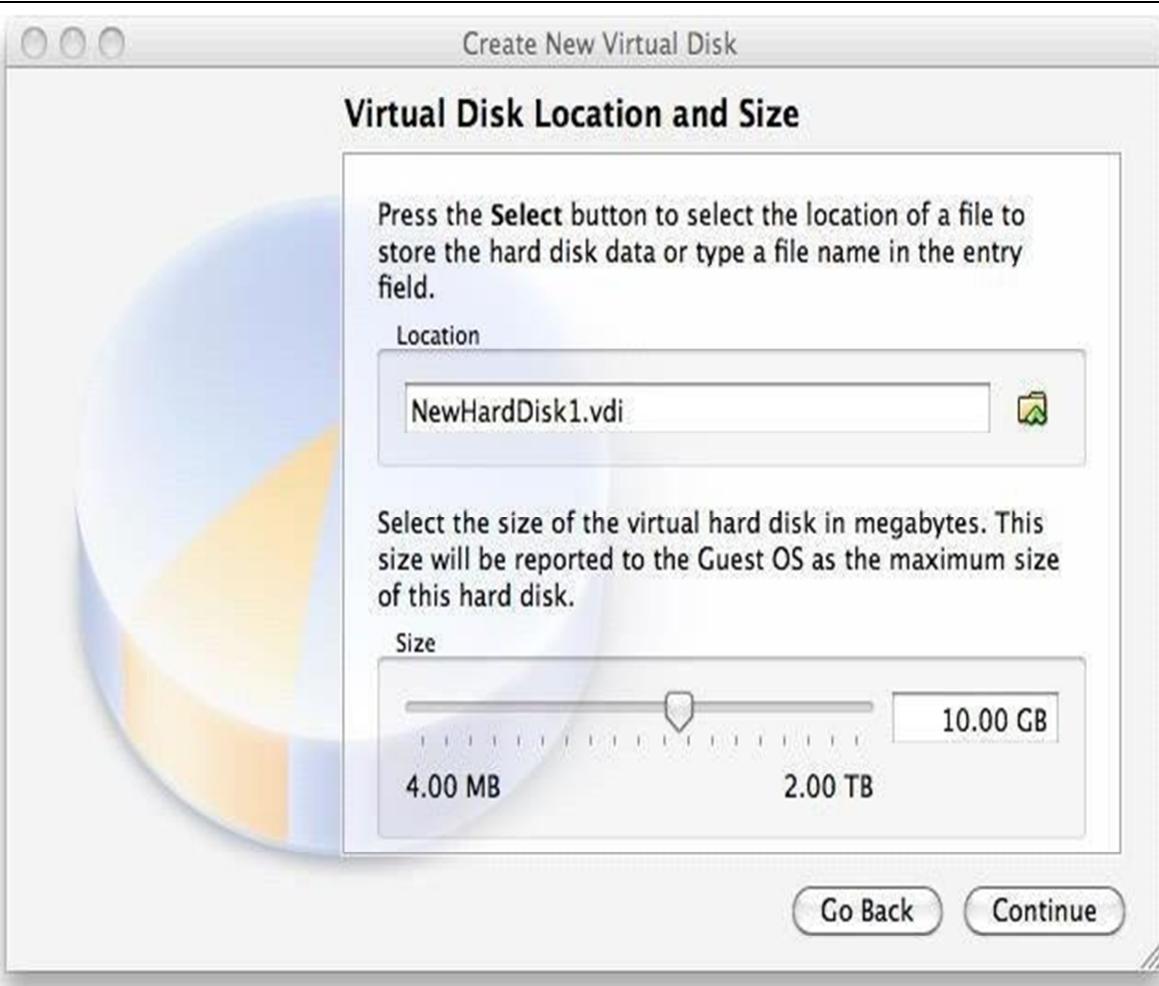
VirtualBox supports two types of image files:

- A **dynamically allocated file** will only grow in size when the guest actually stores data on its virtual hard disk. It will therefore initially be small on the host hard drive and only later grow to the size specified as it is filled with data.
- A **fixed-size file** will immediately occupy the file specified, even if only a fraction of the virtual hard disk space is actually in use. While occupying much more space, a fixed-size file incurs less overhead and is therefore slightly faster than a dynamically allocated file.

For details about the differences, please refer to [Section 5.2,Disk image files \(VDI, VMDK, VHD, HDD\)](#).

After having selected or created your image file, again press "**Next**" to go to the next page.

- After clicking on "**Finish**", your new virtual machine will be created. You will then see it in the list on the left side of the Manager window, with the name you entered initially.

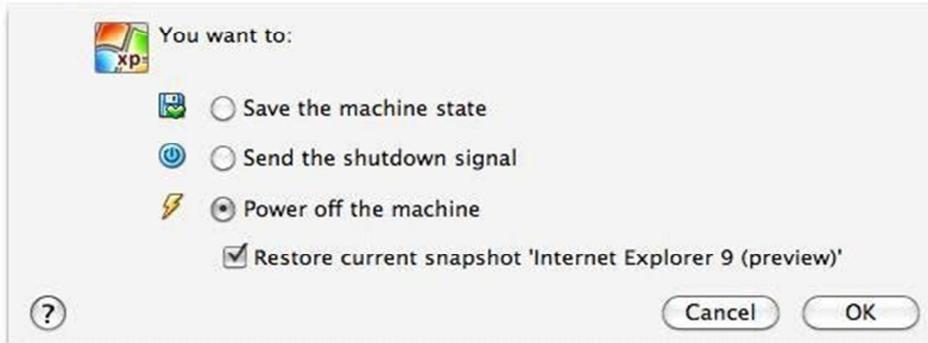


Running your virtual machine: To start a virtual machine, you have several options:

- Double-click on its entry in the list within the Manager window or
- select its entry in the list in the Manager window it and press the "Start" button at the top or
- for virtual machines created with VirtualBox 4.0 or later, navigate to the "VirtualBox VMs" folder in your system user's home directory, find the subdirectory of the machine you want to start and double-click on the machine settings file (with a .vbox file extension). This opens up a new window, and the virtual machine which you selected will boot up. Everything which would normally be seen on the virtual system's monitor is shown in the window. In general, you can use the virtual machine much like you would use a real computer. There are couple of points worth mentioning however.

Saving the state of the machine: When you click on the "Close" button of your virtual machine window (at the top right of the window, just like you would close any other window on your

system), VirtualBox asks you whether you want to "save" or "power off" the VM. (As a shortcut, you can also press the Host key together with "Q".)



The difference between these three options is crucial. They mean:

- **Save the machine state:** With this option, VirtualBox "freezes" the virtual machine by completely saving its state to your local disk. When you start the VM again later, you will find that the VM continues exactly where it was left off. All your programs will still be open, and your computer resumes operation. Saving the state of a virtual machine is thus in some ways similar to suspending a laptop computer (e.g. by closing its lid).
- **Send the shutdown signal.** This will send an ACPI shutdown signal to the virtual machine, which has the same effect as if you had pressed the power button on a real computer. So long as the VM is running a fairly modern operating system, this should trigger a proper shutdown mechanism from within the VM.
- **Power off the machine:** With this option, VirtualBox also stops running the virtual machine, but *without* saving its state. As an exception, if your virtual machine has any snapshots (see the next chapter), you can use this option to quickly **restore the current snapshot** of the virtual

machine. In that case, powering off the machine will not disrupt its state, but any changes made since that snapshot was taken will be lost. The "**Discard**" button in the VirtualBox

Manager window discards a virtual machine's saved state. This has the same effect as powering it off, and the same warnings apply.

Importing and exporting virtual machines

VirtualBox can import and export virtual machines in the industry-standard Open Virtualization Format (OVF). OVF is a cross-platform standard supported by many virtualization products which allows for creating ready-made virtual machines that can then be imported into a virtualizer such as VirtualBox. VirtualBox makes OVF import and export easy to access and supports it from the Manager window as well as its command-line interface. This allows for packaging so-called **virtual appliances**: disk images together with configuration settings that can be distributed easily. This way one can offer complete ready-to-use software packages (operating systems with applications) that need no configuration or installation except for importing into VirtualBox.

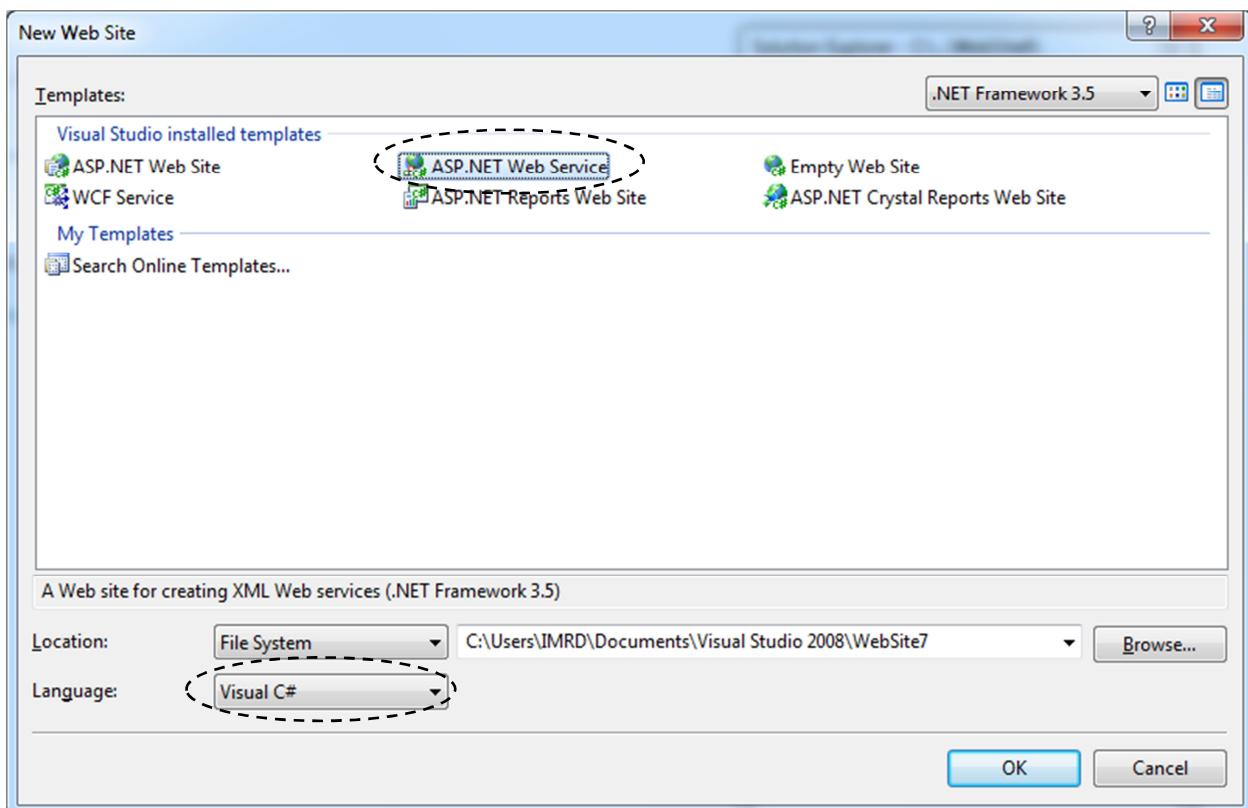
Appliances in OVF format can appear in two variants:

- They can come in several files, as one or several disk images, typically in the widely-used VMDK format (see [Section 5.2.Disk image files \(VDI, VMDK, VHD, HDD\) II](#)) and a textual description file in an XML dialect with an .ovf extension. These files must then reside in the same directory for Virtual Box to be able to import them.
- Alternatively, the above files can be packed together into a single archive file, typically with an .ova extension. (Such archive files use a variant of the TAR archive format and can therefore be unpacked outside of Virtual Box with any utility that can unpack standard TAR files.)

Select "File" -> "Export appliance". A different dialog window shows up that allows you to combine several virtual machines into an OVF appliance. Then, select the target location where the target files should be stored, and the conversion process begins. This can again take a while.

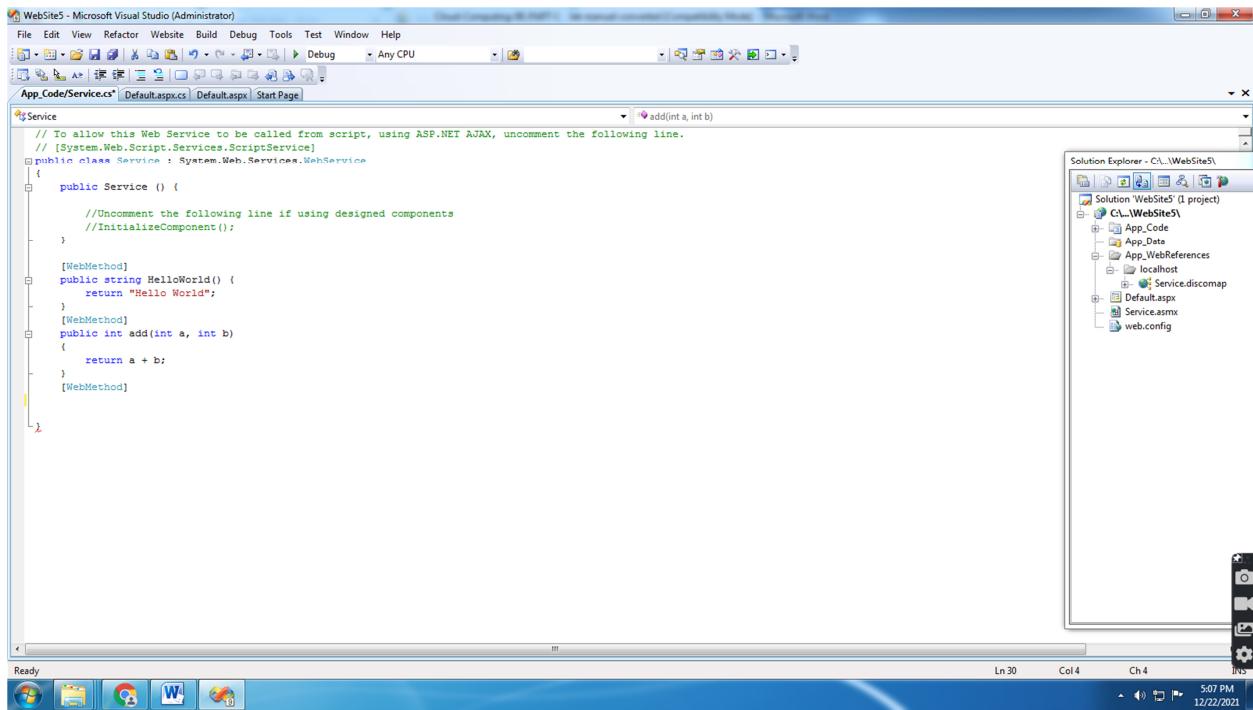
Creating and utilizing Web Service using SOAP in C#

- 1) Start visual studio
- 2) Select File ➔ New ➔ Webiste
- 3) Select Asp.net Web Services



- 4) Add following code

```
[WebMethod]  
public int add(int a, int b)  
{  
    return a + b;  
}
```



5) Test your web service by pressing F5 button on Keyboard

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [HelloWorld](#)
- [add](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "<http://microsoft.com/webservices/>".

C#

```

[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService
{
    // implementation
}

```

Visual Basic

```

<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class

```

C++

```

[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService
{
    // implementation
};

```

For more details on XML namespaces, see the W3C recommendation on [Namespaces in XML](#).

For more details on WSDL, see the [WSDL Specification](#).

For more details on URIs, see [RFC 2396](#).

6) Click on add

S Welcome to RCP Captive Portal Post Attendee - Zoom PDF to Word Converter - 100% F creating web services in asp.net Service Web Service

localhost:6156/WebSite5/Service.asmx?op=add

Service

Click [here](#) for a complete list of operations.

add

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
a:	10
b:	20

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```

POST /WebSite5/Service.asmx HTTP/1.1
Host: localhost:6156
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/add"

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
<soap:Body>
<add>
<a>10</a>
<b>20</b>
</add>
</soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

```

Date: Mon, 22 Dec 2021 18:59:46 GMT
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
<soap:Body>
<addResponse xmlns="http://tempuri.org/">
<addResult>30</addResult>
</addResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

localhost:6156/WebSite5/Service.asmx?op=add

Cloud-Computin....docx

Show all

5:09 PM 12/22/2021

S Welcome to RCP Captive Portal Post Attendee - Zoom PDF to Word Converter - 100% F creating web services in asp.net Service Web Service

localhost:6156/WebSite5/Service.asmx

This XML file does not appear to have any style information associated with it. The document tree is shown below.

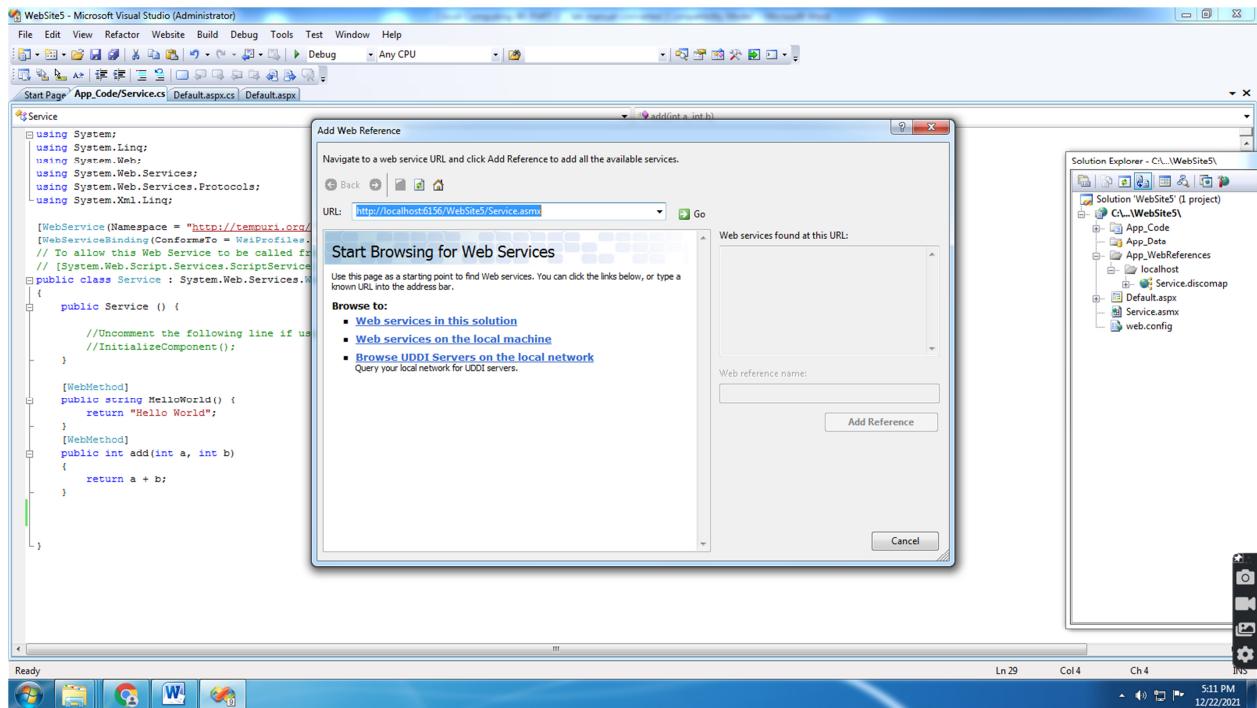
```
<int xmlns="http://tempuri.org/">30</int>
```

Cloud-Computin....docx

Show all

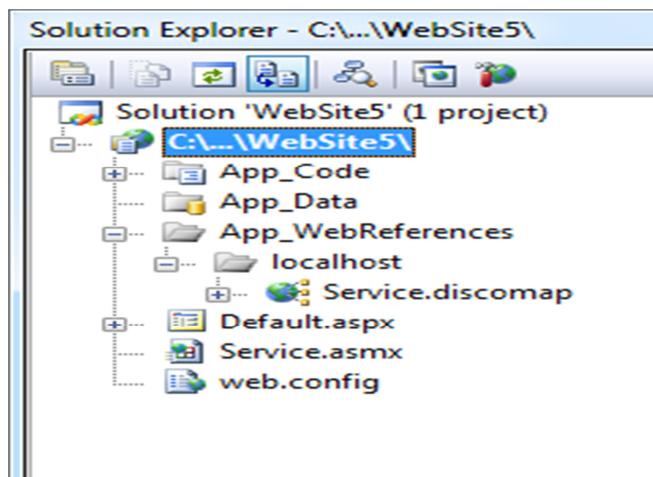
5:09 PM 12/22/2021

- 7) Copy URL of Web service (having extension .asmx)
- 8) Click on Website Menu ➔ Add Web Reference.

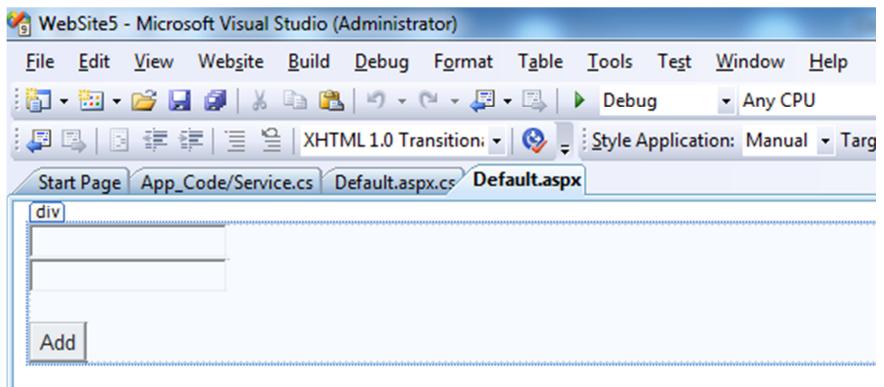


9) Paste the URL copied earlier

10) Select web service name & Click on Add reference.



Now create application for addition and add following code



```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

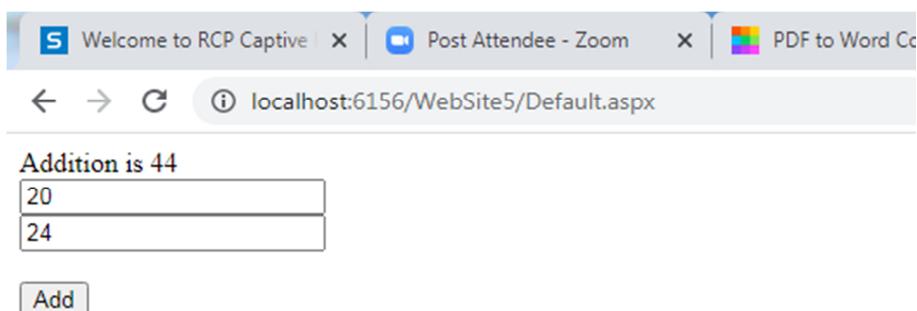
public partial class _Default : System.Web.UI.Page
{
    int n1, n2, res;
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        localhost.Service s = new localhost.Service();

        n1 = Convert.ToInt32(textBox1.Text);
        n2 = Convert.ToInt32(textBox2.Text);
        res=s.add(n1, n2);
        Response.Write("Addition is " + res);

    }
}
```

Run the Application

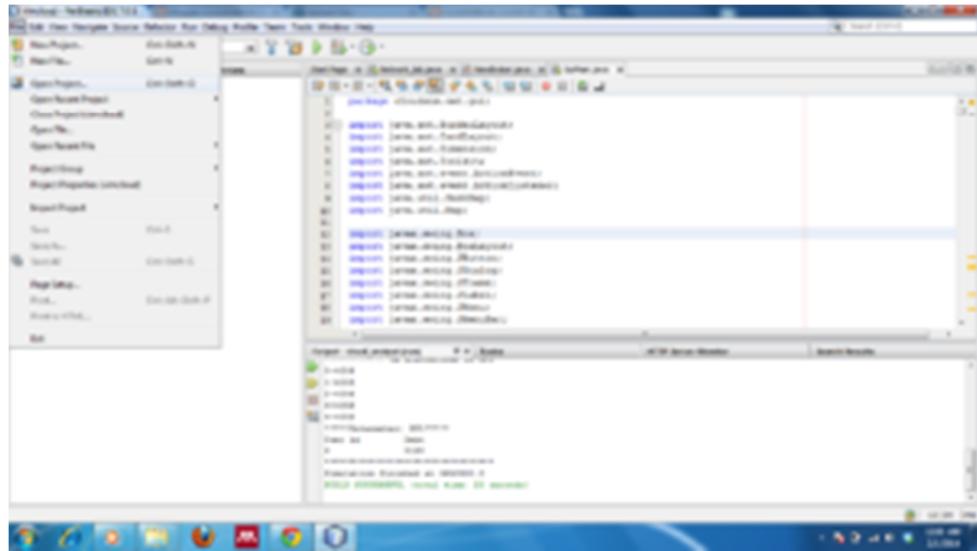


Assignment :- Downloading and Installing Cloud Analyst

Download Cloud Analyst Netbeans project from the below mentioned link

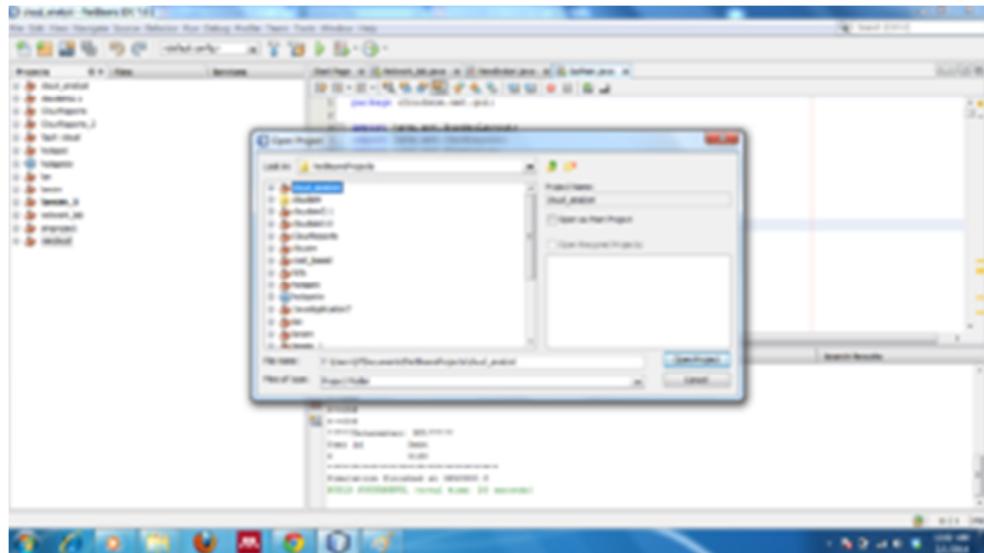
<http://www.cloudbus.org/cloudsim/>

To run the cloud analyst got to NetBeans File->open project -> browse the unzipped folder



Step2

Browse the unzipped folder that you have downloaded.



Step3

Open source package folder inside which open cloudsim.ext.gui

right click on the gui.main.java and click run

