

How to Make Boxes?

Environment Settings

1. Make your working directory.

All boxes files should be located in the hierarchy of directory *warehouse_root*.

1.1 Make directory *warehouse_root*.

1.2 Under *warehouse_root*, make three parallel directories *boxes*, *interfaces* and *datatypes*.

Figure 1 shows the directory structure.

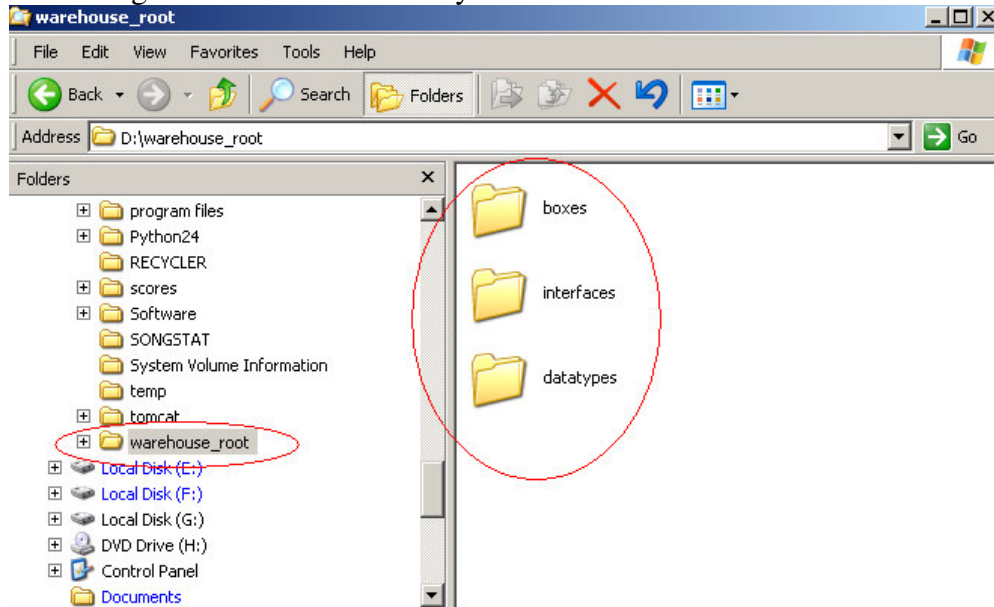


Figure 1. warehouse_root

2. Install BoxCompiler (reference “Install BoxCompiler”)

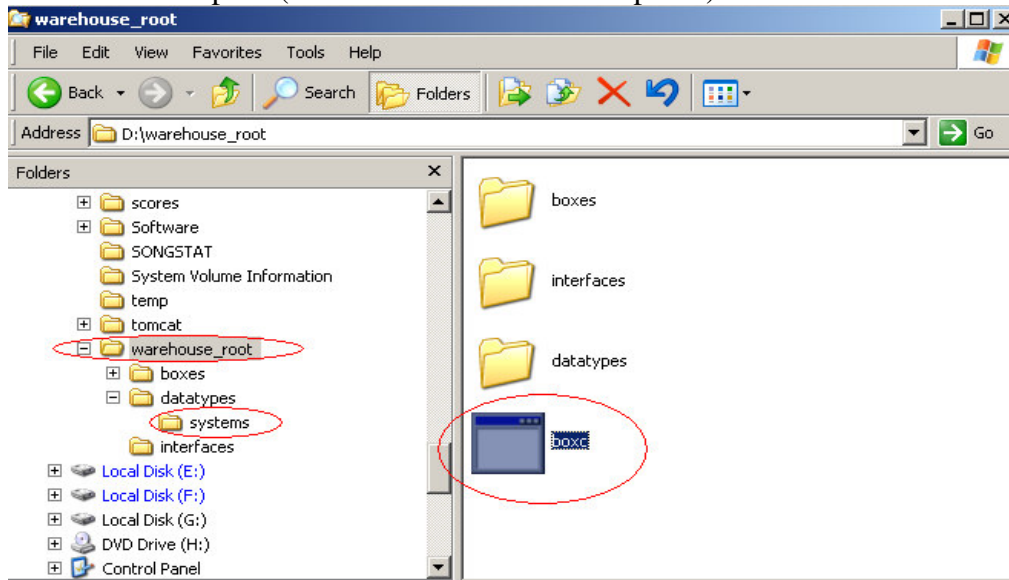


Figure 2. warehouse_root after BoxCompiler installed

2. Add warehouse_root to classpath.

My computer -> Properties -> Advanced -> Environment Variables

Make a new variable "WAREHOUSE_ROOT" (Figure 3), and add it to the classpath (Figure 4).

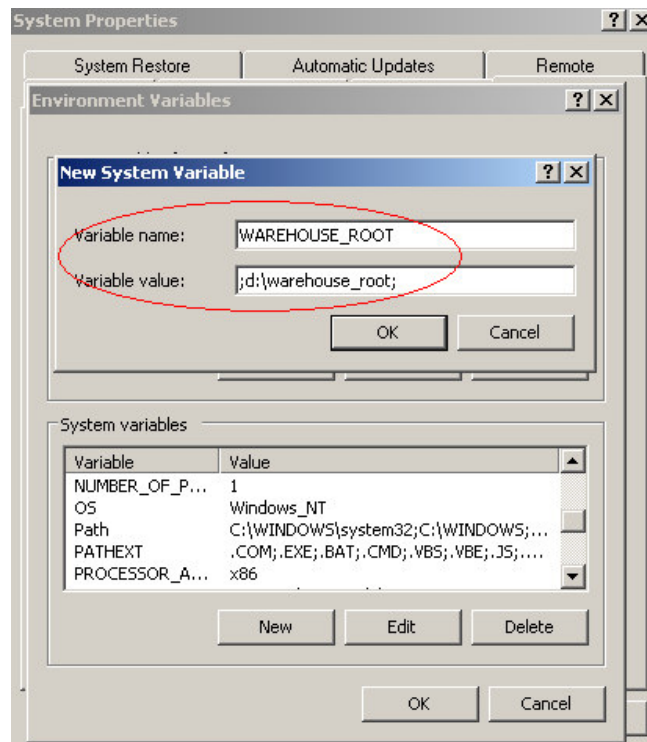


Figure 3. New variable WAREHOUSE_ROOT

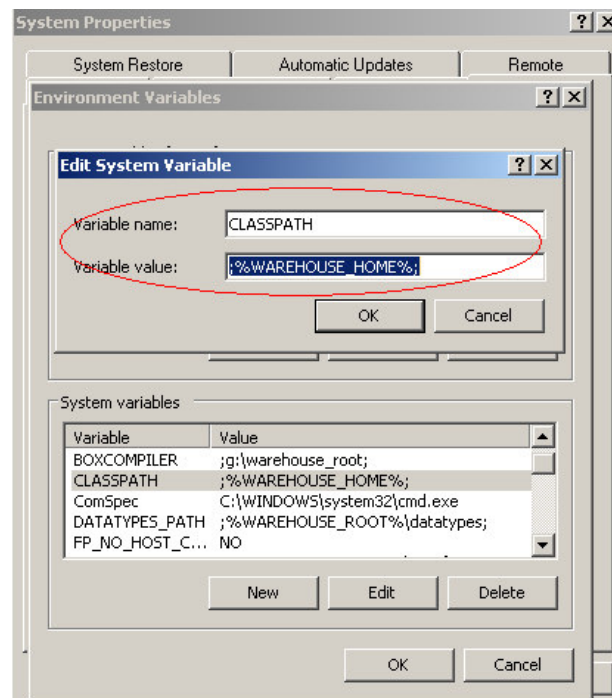


Figure 4. add warehouse_root to classpath

Make boxes

1. Make an abstract box

Suppose the abstract box is called **CalPriceAbs**.

Step 1: Make a directory *CalPriceAbs* directly below *warehouse_root* (Figure 5).

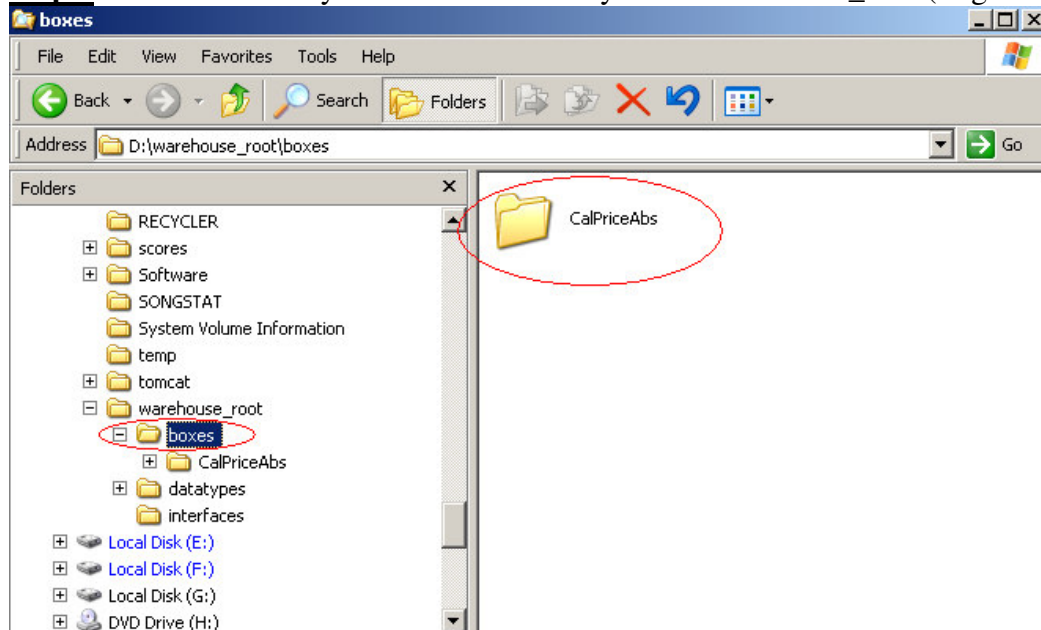


Figure 5. Make a directory for an abstract box

Step2: Write the necessary interfaces for the box, compile them and locate them into the *interfaces* directory. If user defined data types are used, locate them into *datatypes*.

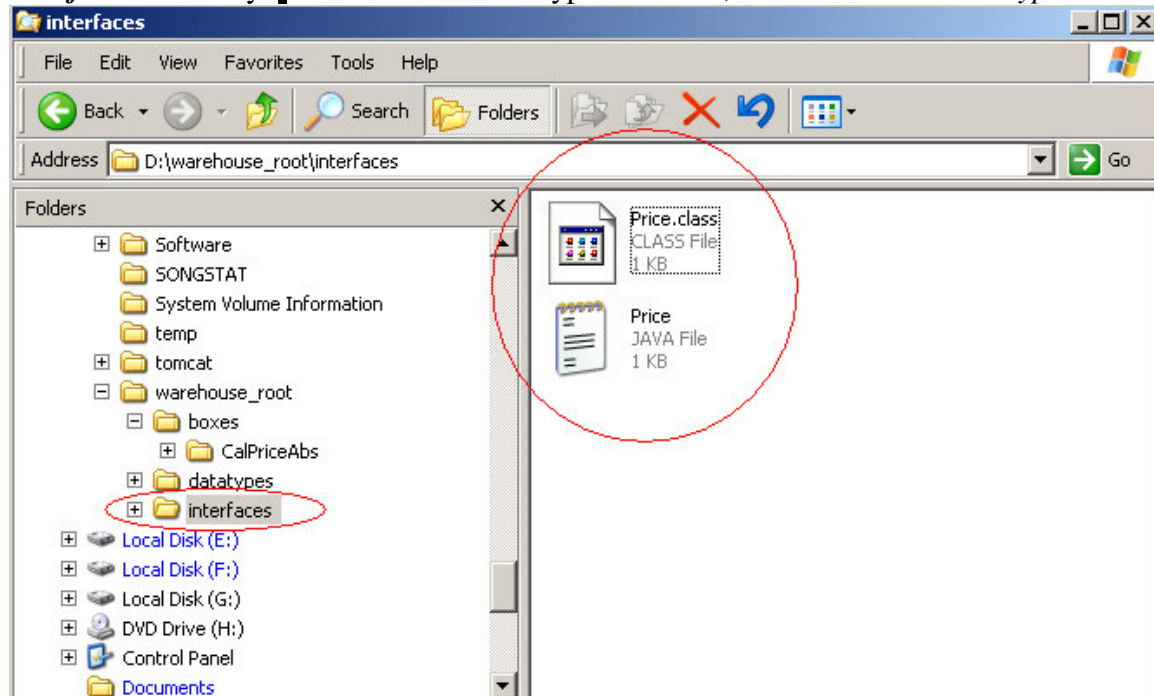


Figure 6. Locate interface files

Step3: Write a box description file CalPriceAbs.box for the abstract box and locate it in the abstract box directory CalPriceAbs.

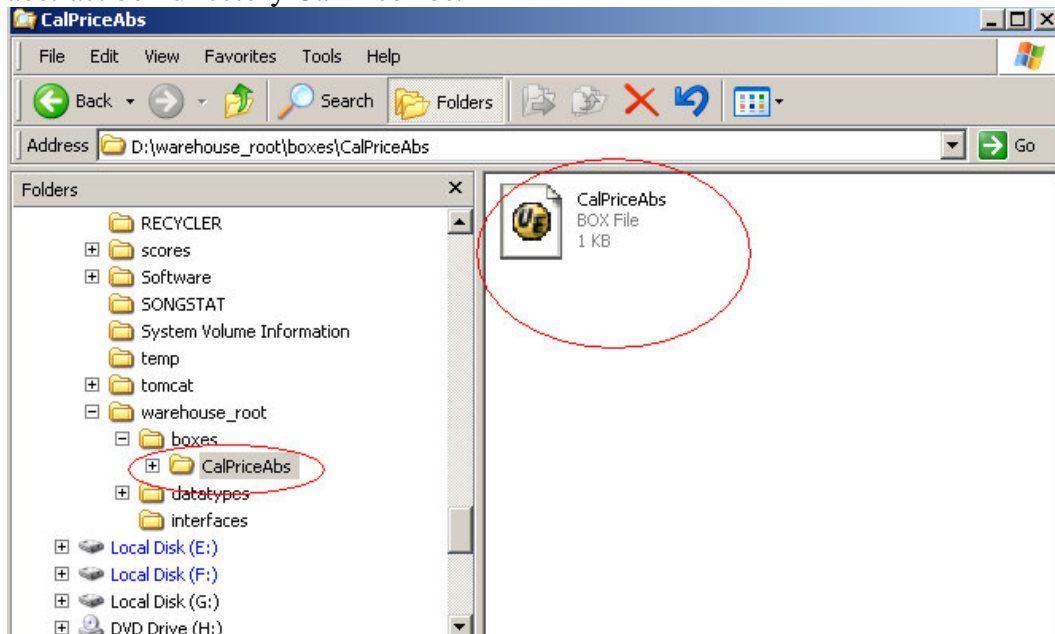


Figure 6. Locate abstract box description file into its directory

Step4: Compile CalPriceAbs.box (Figure 7) . Under CalPriceAbs, there will be a CalPriceAbs.dsc generated (Figure 8).

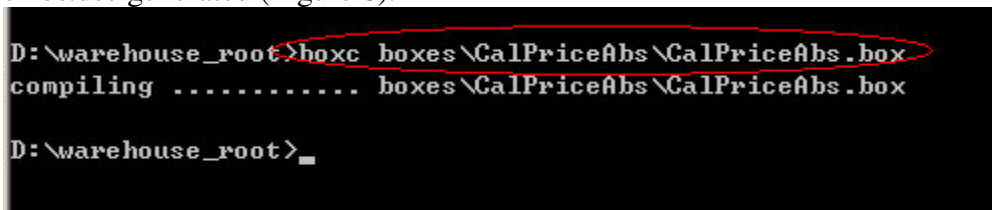


Figure 7. Compile CalPriceAbs.box

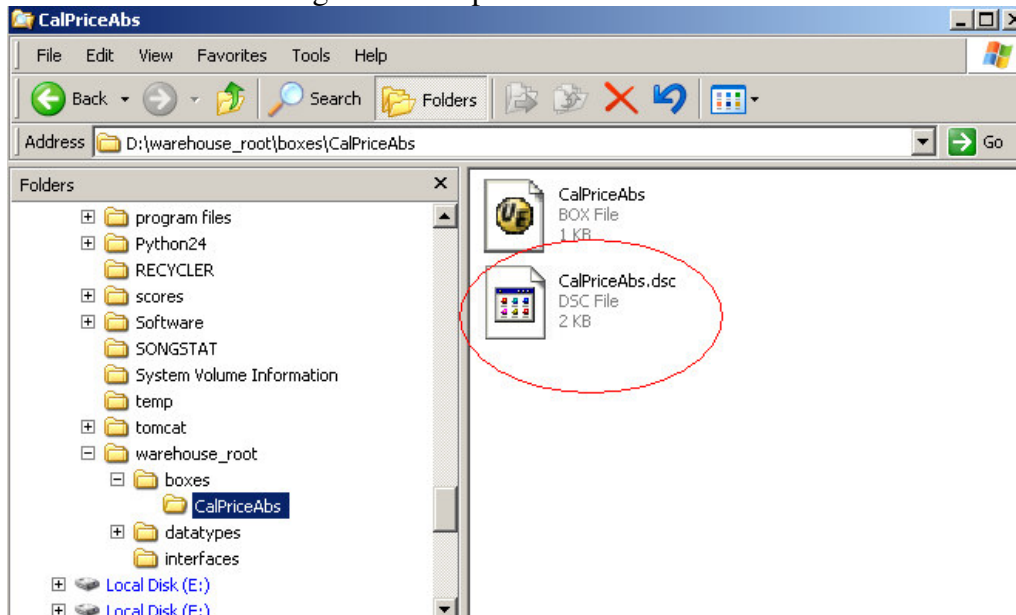


Figure 8. .dsc file generated

Abstract Box Description Syntax

File: *boxname.box*

abstract box *boxname*

```
{  
    provided interfaces interfaceType1 interfaceHandle2 [, interfaceType2  
interfaceHandle2, ..., interfaceTypen interfaceHandlen];  
    required interfaces interfaceTypea interfaceHandlea [,interfaceTypeb  
interfaceHandleb, ..., interfaceTypem interfaceHandlem];  
}
```

2. Make an atomic box

Suppose the atomic box is Pricing and it implements abstract PricingAbs. Suppose we have already made PricingAbs (Figure 9).

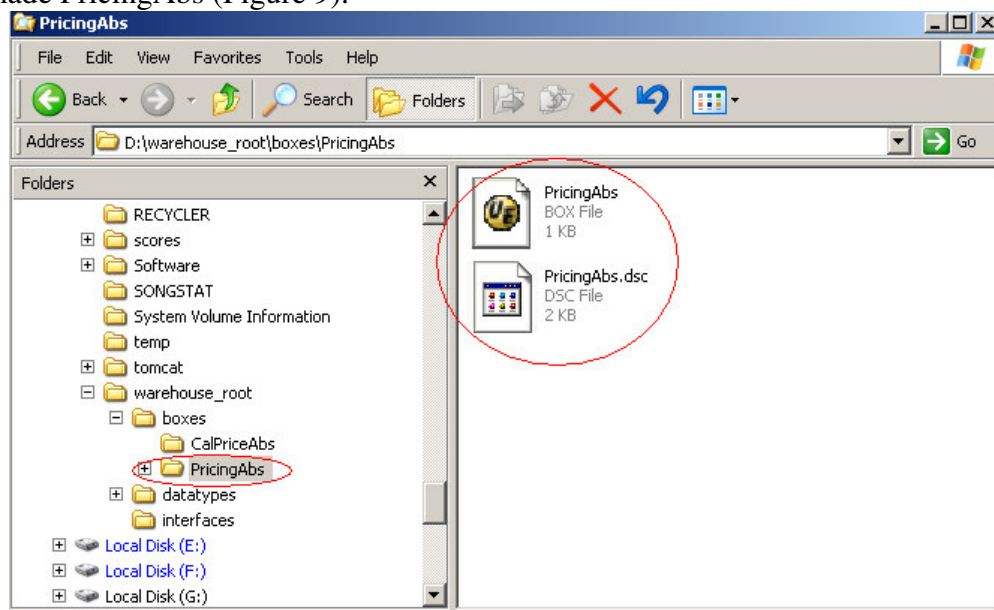


Figure 9. Abstract box PricingAbs

Step1: Make a directory *Pricing* directly below *PricingAbs* (Figure 10).

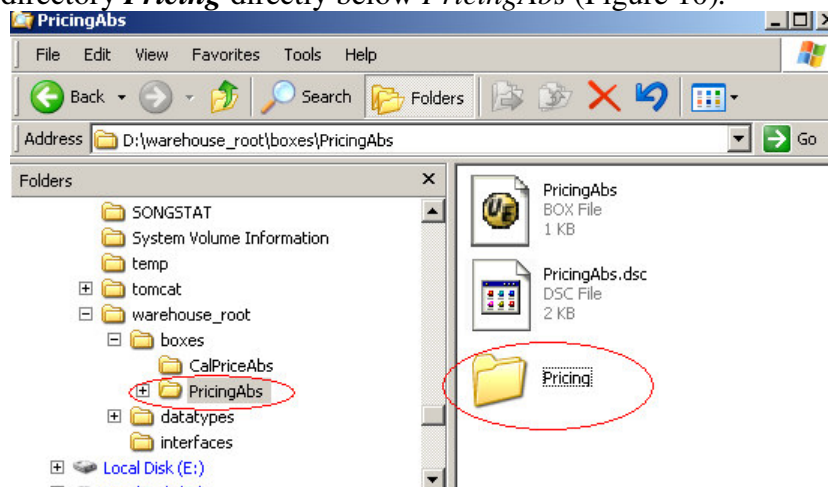


Figure 10. Make directory for atomic box

Step2: Write the necessary interfaces for the box, compile them and locate them into the *interfaces* directory (Figure 11). If user defined data types are used, locate them into *datatypes* (Figure 12).

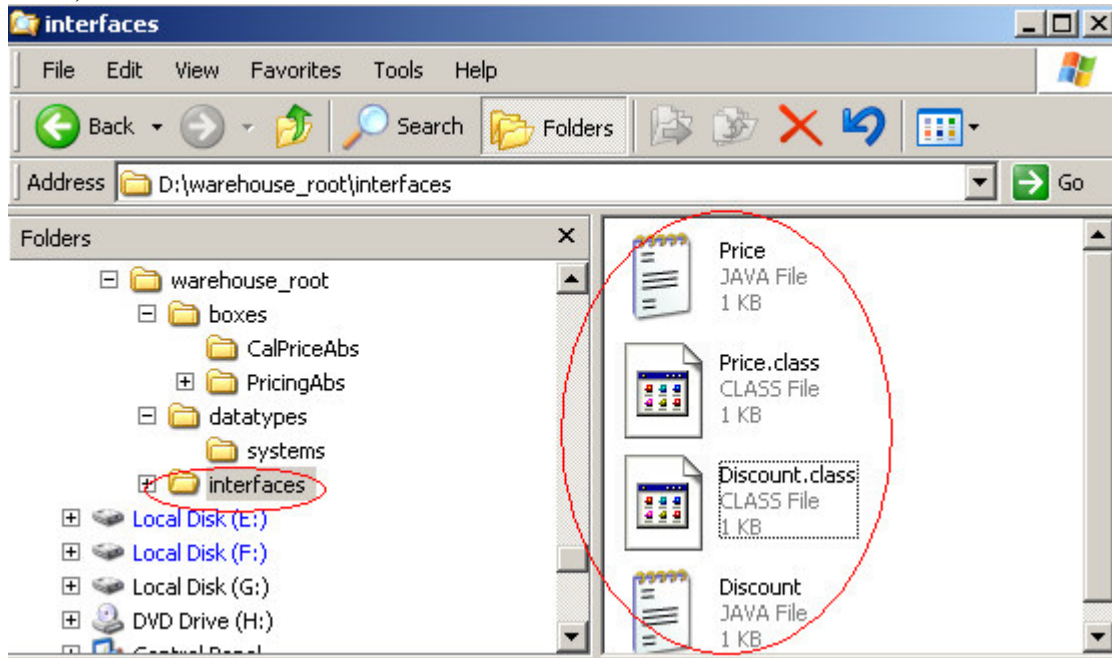


Figure 11. Locate interfaces

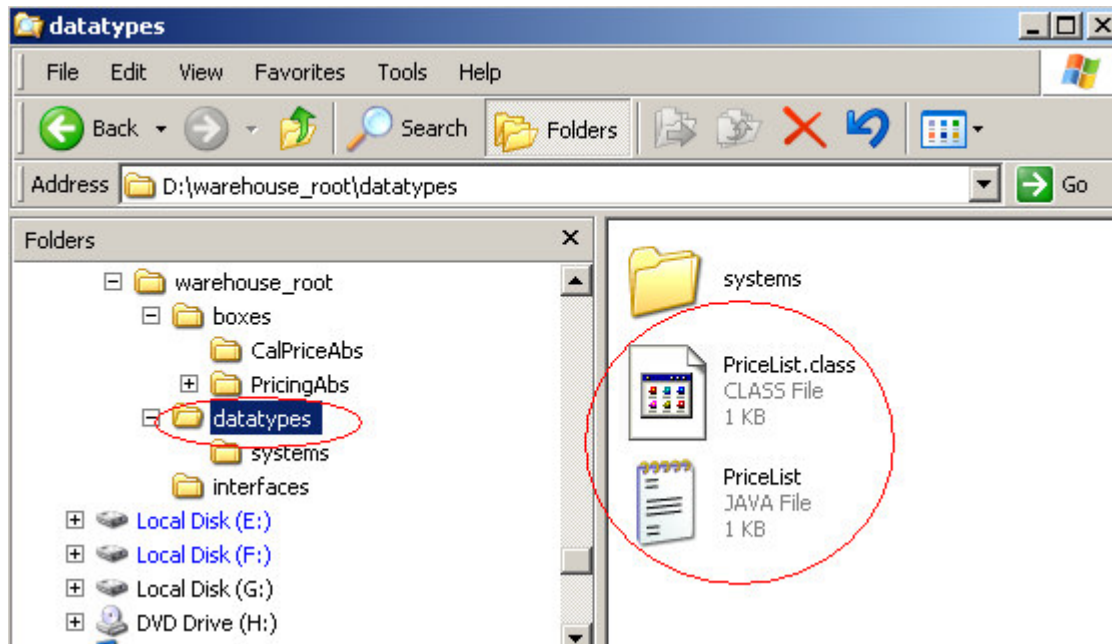


Figure 12. Locate datatypes

Step3: Write a box description file Pricing.box for the atomic box and locate it in the atomic box directory Pricing. (Figure 13)

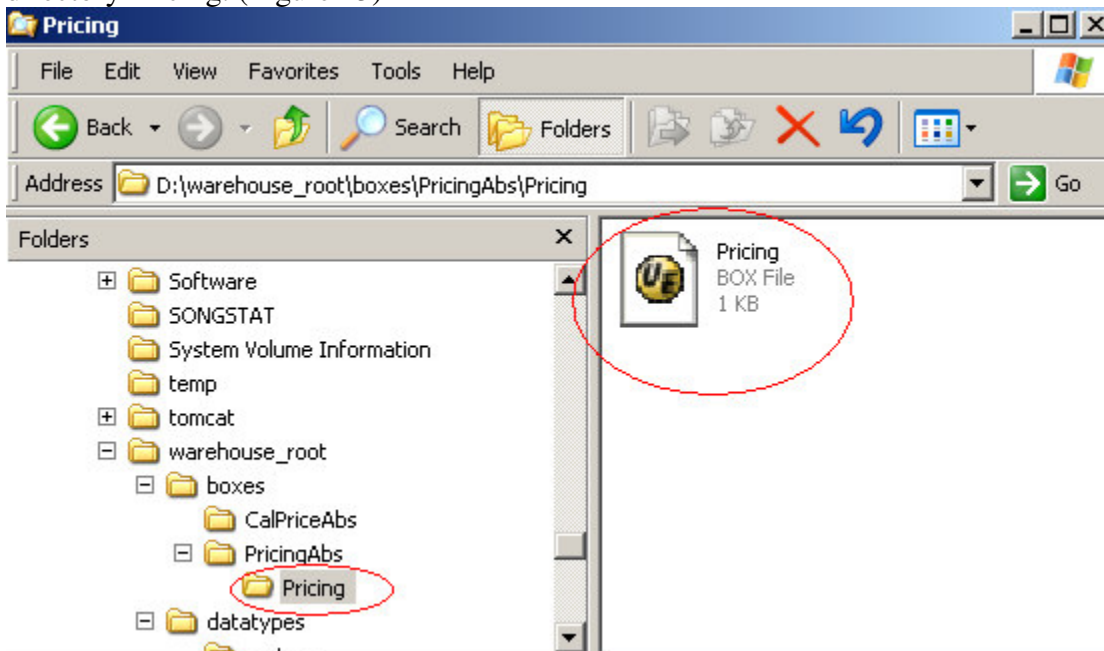


Figure 13. Locate atomic box description file into its directory

Step4: Write java files for implementing its provided interfaces, compile them and locate them in the atomic box directory Pricing. (Figure 14)

If there is a configuration file for this atomic box, locate it to this directory too.

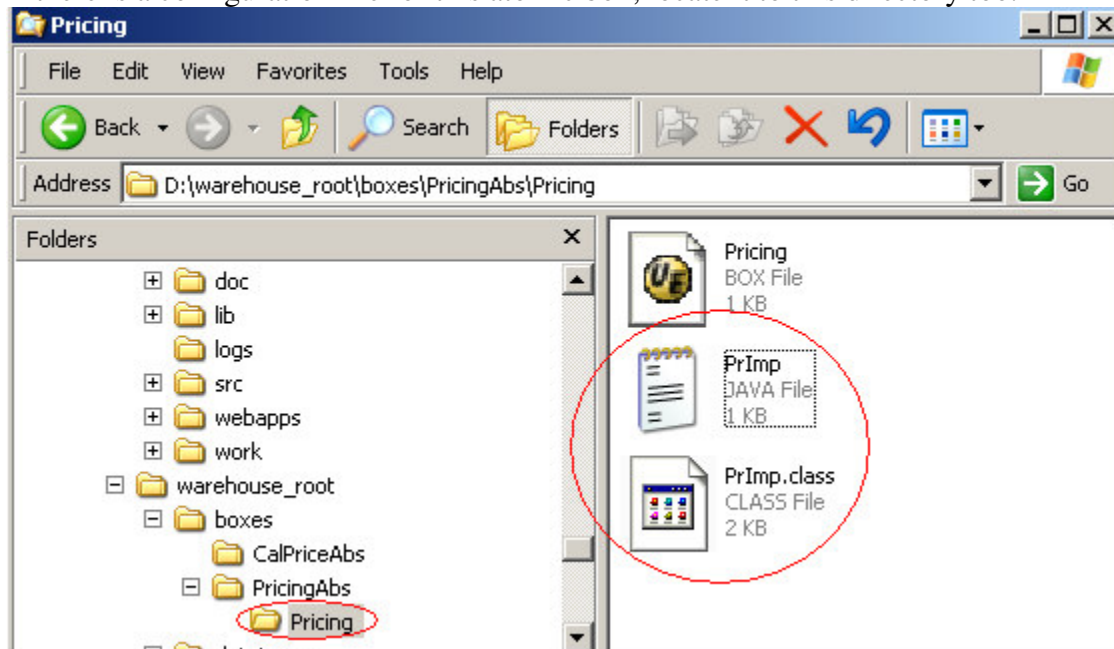


Figure 14. Locate provided interface implementations into its directory

Step5: Compile **Pricing.box** (Figure 15) . Under *Pricing*, Pricing.dsc, Pricing.java and Pricing.class will be generated (Figure 16).

```
D:\warehouse_root>boxc boxes\PricingAbs\Pricing\Pricing.box
compiling ..... boxes\PricingAbs\Pricing\Pricing.box
D:\warehouse_root>
```

Figure 15. Compile Pricing.box

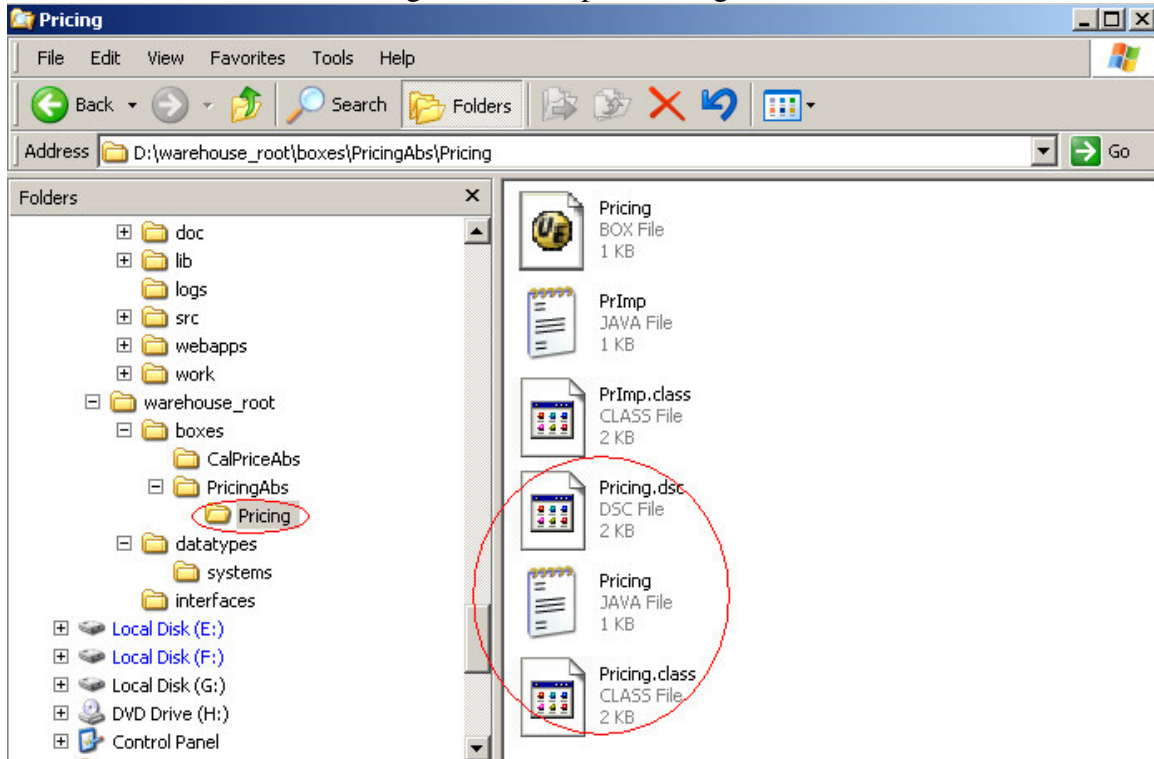


Figure 16. Generated files

Step6: Test the box.

Atomic Box Description Syntax

File: *boxname.box*

```
box boxname [implements itsAbstractBox]
{
    provided interfaces interfaceType1 interfaceHandle2 [, interfaceType2
interfaceHandle2, ..., interfaceTypen interfaceHandlen] ;
    required interfaces interfaceTypea interfaceHandlea [,interfaceTypeb
interfaceHandleb, ..., interfaceTypem interfaceHandlem];
}
```


Make a compound box

Suppose the compound box is CalPrice

Step1: If the compound box implements an abstract box, make a directory with the compound box name right below the abstract box. In this example, CalPrice implements CalPriceAbs. Make a directory CalPrice below CalPriceAbs. (Figure 17)

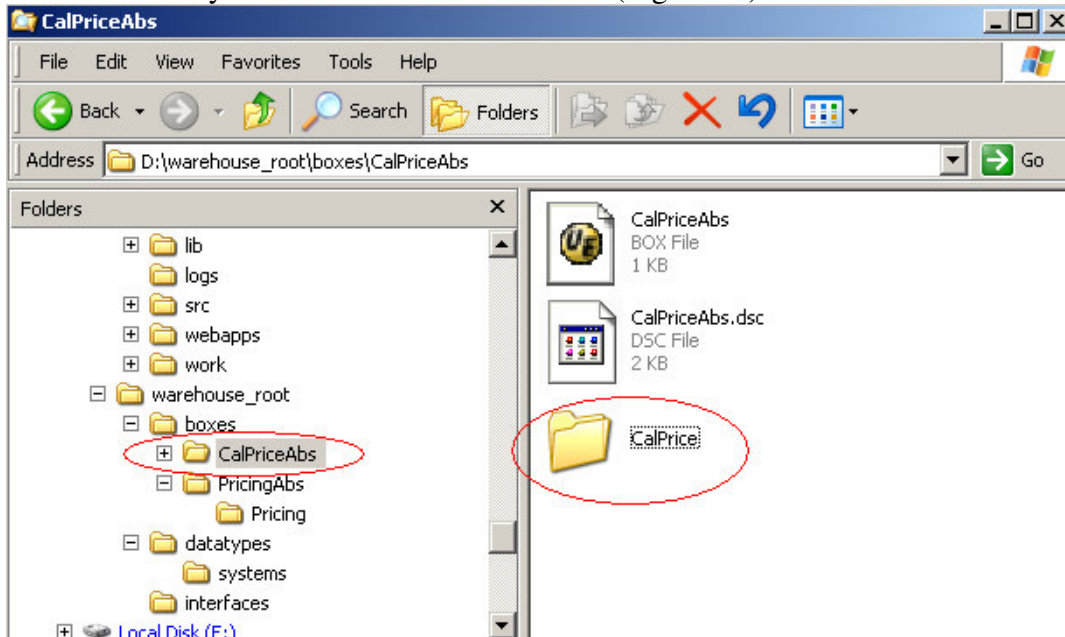


Figure 17. Make directory for compound box

Step2: Write a box description file CalPrice.box for the compound box and locate it in the directory CalPrice. (Figure 18)

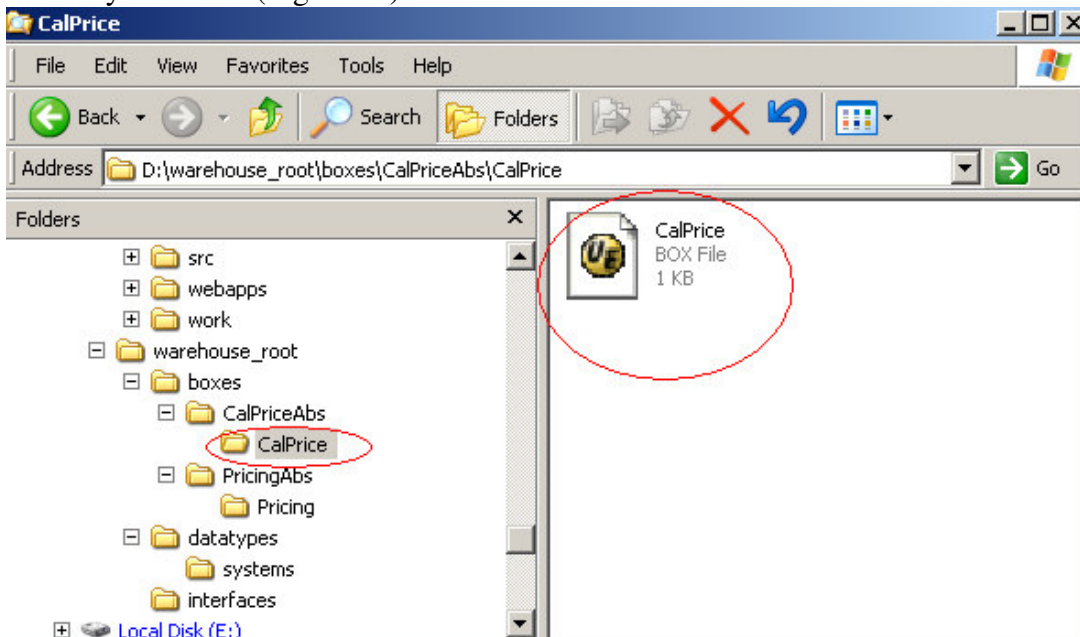


Figure 18. Locate box description file for CalPrice

Step3: Locate configuration file for the compound box. (Figure 19)

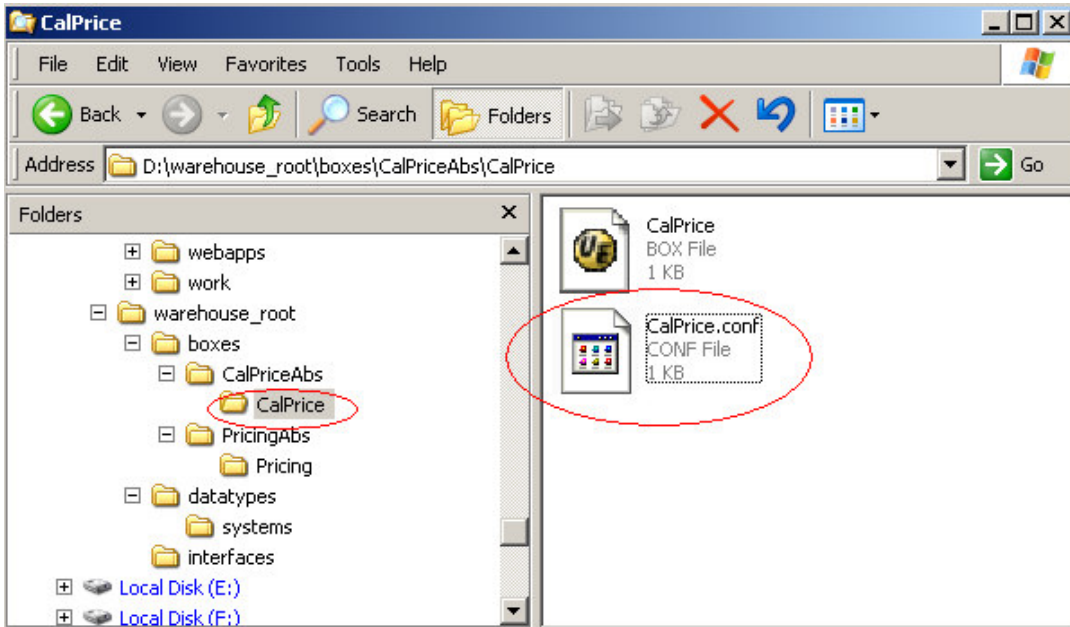


Figure 19. Locate .conf file for compound box.

Step4: Compile the box file (Figure 20) . CalPrice.dsc, CalPrice.java and CalPrice.class will be generated. (Figure 21)

```
D:\warehouse_root>boxc boxes\CalPriceAbs\CalPrice\CalPrice.box
compiling ..... boxes\CalPriceAbs\CalPrice\CalPrice.box
```

Figure 20. Compile CalPrice.box

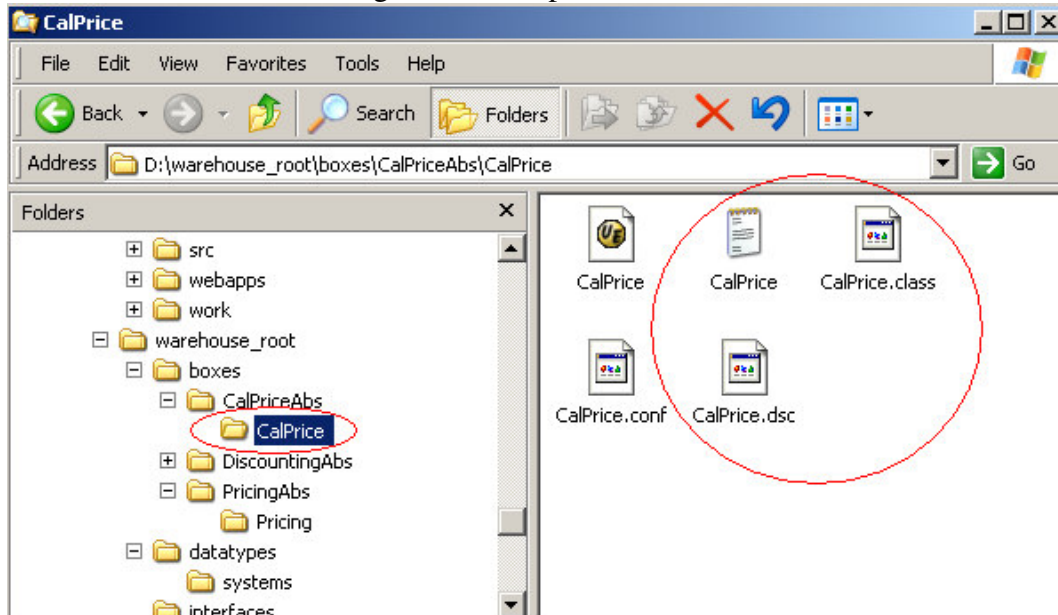


Figure 21. Generated files

Step5: Write a testing java file and test the box.

Compound Box Description Syntax

File: boxname.box

box *boxname* [*implements ItsAbstractName*]

{ composed from *boxType1 boxHandle1*, *boxType2 boxHandle2* [*boxType3 boxHandle3*, ... ,
boxTypeN boxHandleN];

provided interfaces *itfType1 itfHandle1* **from** *boxHandle1.itfhandle*
[, *itfType2 itfHandle2* **from** *boxHandle2.itfhandleA*
, ..., *itfTypeN itfHandleN* **from** *boxHandleN.itfhandleZ*];

required interfaces *ritfType1 ritfHandle1* **from** *boxHandle1.ritfhandle*
[, *ritfType2 ritfHandle2* **from** *boxHandle2.ritfhandleA*
, ..., *ritfTypeN ritfHandleN* **from** *boxHandleN.ritfhandleZ*];

connect *boxHandle1.requiredInterfaceHandleA* **to** *boxHandle2.providedInterfaceHandleB*
[, *boxHandle3.requiredInterfaceHandleC* **to** *boxHandle4.providedInterfaceHandleD*
, ..., *boxHandleM.requiredInterfaceHandleP* **to** *boxHandleN.providedInterfaceHandleQ*];

}