UniDyn--Demo-01.nb

John A. Marohn
jam99@cornell.edu
Cornell University

Abstract: This demonstration notebook loads the **UniDyn** package and executes the package's unit tests.

Set the path to the package

Check the Mathematica version number .

In[1306]:=
```
$VersionNumber
```

Out[1306]=
```
13.
```

Tell *Mathematica* the path to the directory containing the packages.

EDIT THE FOLLOWING PATH STRING:

In[1307]:=
```
$UniDynPath =
  "/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/
    unidyn";
```

YOU SHOULD NOT NEED TO EDIT ANYTHING FROM HERE ONWARDS.

Load the package

Append the package path to the system path. Before trying to load the package, ask *Mathematica* to find it. This is a test that we directed *Mathematica* to the correct directory. The output of this command should be the full system path to the Uni-Dyn.m file.

In[1308]:=
```
$Path = AppendTo[$Path, $UniDynPath];
FindFile["UniDyn`"]
```

Out[1309]=
```
/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/unidyn/UniDyn.m
```

Now that we are confident that the path is set correctly, load the package. Setting the global $VerboseLoad variable to True will print out the help strings for key commands

in the package.

In[1310]:=
```
$VerboseLoad = True;
Needs["UniDyn`"]
```

Execute the units tests in batch

Included with the package are a number of files, ending in “-tests.m”, that contain tests of the package’s functions -- so-called unit tests. Set the working directory to the package directory and pretty-print the directory name.

In[1312]:=
```
SetDirectory[$UniDynPath];
TableForm[{{$UniDynPath}}, TableHeadings → {None, {"Directory"}}]
```

Out[1313]//TableForm=

Directory
/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/unidyn

Get the names of all the unit-testing files included with the package (following my convention that the unit testing file end in “-tests.m”). Pretty-print the names of the unit-test files included with the package.

In[1314]:=
```
fn = FileNames["*-tests.m"];
TableForm[{{fn}}, TableHeadings → {None, {"Test files found"}}]
```

Out[1315]//TableForm=

Test files found
Comm-tests.m
Evolve-tests.m
Mult-tests.m
OpQ-tests.m
Osc-tests.m
Spins-tests.m

Finally, carry out the unit tests.

In[1316]:=
```
test$report = TestReport /@ fn;
TableForm[Table[test$report [[k]], {k, 1, Length[test$report]}]]
```

Out[1317]//TableForm=

TestReportObject	Title: Test Report: Comm –tests.m	Success rate: 100%	Tests run: 23
TestReportObject	Title: Test Report: Evolve –tests.m	Success rate: 100%	Tests run: 24
TestReportObject	Title: Test Report: Mult –tests.m	Success rate: 100%	Tests run: 18
TestReportObject	Title: Test Report: OpQ –tests.m	Success rate: 100%	Tests run: 21
TestReportObject	Title: Test Report: Osc –tests.m	Success rate: 100%	Tests run: 22
TestReportObject	Title: Test Report: Spins –tests.m	Success rate: 100%	Tests run: 14

Make a report.

In[1318]:=
```
tests$passed$total = Plus @@ (test$report[[#]]["TestsSucceededCount"] & /@
      List @@ Table[k, {k, 1, Length[test$report]}]);
tests$failed$total = Plus @@ (test$report[[#]]["TestsFailedCount"] & /@
      List @@ Table[k, {k, 1, Length[test$report]}]);

Print[Style[ToString[tests$passed$total ] <> " tests passed",
  FontWeight → Bold, FontSize → 18, FontColor → Blue]]
Print[Style[ToString[tests$failed$total ] <> " tests failed" ,
  FontWeight → Bold, FontSize → 18, FontColor → Red]]
```

122 tests passed

0 tests failed

Execute the units tests one-by-one

Re-execute the tests in an order determined by us. This is useful for debugging. Running the *Evolve-test.m* file takes a minute.

In[1322]:=
```
SetDirectory[$UniDynPath];
TableForm[{{$UniDynPath}}, TableHeadings → {None, {"Directory"}}]
```

Out[1323]//TableForm=

Directory
/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/unidyn

In[1324]:=
```
$VerboseLoad = False;
Needs["UniDyn`"]
```

In[1326]:=
```
TestReport[FileNames["OpQ-tests.m"][[1]]]
```

Out[1326]= TestReportObject[Title: Test Report: OpQ –tests.m Success rate: 100% Tests run: 21]

In[1327]:=
```
TestReport[FileNames["Mult-tests.m"][[1]]]
```

Out[1327]= TestReportObject[Title: Test Report: Mult –tests.m Success rate: 100% Tests run: 18]

In[1328]:=
```
TestReport[FileNames["Comm-tests.m"][[1]]]
```

Out[1328]= TestReportObject[Title: Test Report: Comm –tests.m Success rate: 100% Tests run: 23]

In[1329]:=
```
TestReport[FileNames["Spins-tests.m"][[1]]]
```

Out[1329]= TestReportObject[Title: Test Report: Spins –tests.m Success rate: 100% Tests run: 14]

In[1330]:=
```
TestReport[FileNames["Osc-tests.m"][[1]]]
```

Out[1330]= TestReportObject[Title: Test Report: Osc –tests.m Success rate: 100% Tests run: 22]

In[1331]:=
```
TestReport[FileNames["Evolve-tests.m"][[1]]]
```

Out[1331]= TestReportObject[Title: Test Report: Evolve –tests.m Success rate: 100% Tests run: 24]

Congratulations

At this point you should have

(1) loaded the UniDyn package and

(2) run the UniDyn units tests demonstrating that UniDyn is working as expected.