

## *What Is a Mathematica Package?*

---

### ■ Introduction

A *Mathematica* package is used to store *Mathematica* code so that it can be loaded into a *Mathematica* session. Typically, a package is placed in a file that has the extension ".m".

A *Mathematica* package provides one or more functions, which are placed into a context or group of *Mathematica* symbols. The code that gives the functionality is hidden in an implementation section of the package.

### ■ Structure of a Package

The contents of a sample package are shown below.

```
BeginPackage[ "Package`" ]

MainFunction::usage =
    "MainFunction[ x] computes a simple function."

Begin[ "Private`" ]

MainFunction[ x_] :=
    Module[ {y},
        y = x^2;
        y + 1
    ]

End[]

EndPackage[]
```

The context for the package is **Package`**; this is specified by the argument to **BeginPackage**. The package exports one function named **MainFunction**; this is the only function declared between the **BeginPackage** and the **Begin**. All the code used to implement the package is kept in the Private section of the package; this is between the **Begin** and the **End** statements.

### ■ Loading a Package into *Mathematica*

To load a package into *Mathematica* the package needs to be placed into a directory on the *Mathematica* **\$Path**. After this it can be loaded with either **Get** (often this entered with the '<<' syntax) or **Needs**.

```
In[1]:= << Package`

In[1]:= MyFunction[ 10]

Out[1]= 101
```

If the package cannot be loaded then *Mathematica* will print an error. This will happen if the package has not been placed on the *Mathematica* **\$Path**. In this case you might want to search for the package, for example, using the command **FindFile**.

```
In[2]:= FindFile[ "Package`" ]
```

```
Out[2]= "C:\Documents and Settings\User\My Documents\work\Eclipse_workspaces\Demo1\Package.m"
```

If you load a package with **Needs**, it will only be loaded if it has not been loaded before. If you load a package with **Get**, it will always be loaded. It is a useful optimization feature to only load a package once. However, when you are developing the package, it is often useful to load it every time you make a change, and so you should use **Get**.

## ■ Package Dependencies

One of the important techniques for building software applications is to break up your code into different components, each of which does different things. You can then set your package to load code from other packages. There are two ways that you can do this. In the first example, the extra packages are placed in the **BeginPackage** command. This makes all the functions in **Package1`** and **Package2`** available to the *Mathematica* session as well as to the functions inside **Package`**.

```
BeginPackage[ "Package`", {"Package1`", "Package2`"}]
...
```

The second technique for loading package dependencies uses **Needs** just after the **BeginPackage** command. If you do this, then the *Mathematica* session is not affected, but functions in **Package`** can use the imported commands.

```
BeginPackage[ "Package`"]
Needs[ "Package1`"]
Needs[ "Package2`"]
...
```

## ■ Summary

*Mathematica* packages are an important way to write and deliver code. Properly structured packages help to ensure that multiple packages can be installed and run within *Mathematica* without their interfering with each other. Consequently, the *Workbench* provides a lot of support for working with packages.