Importing Multiple Files

Created using Maple 14.01

Jake Bobowski

```
> restart;
  with(Statistics) :
  with(StringTools) :
   FormatTime("%m−%d−%Y, %H:%M");
                         "11-19-2013, 18:19"                              (1)
```

As discussed in the Importing Data tutorial, the *readdata* command can be used to import data from a file.

```
> data1 := readdata("G:\\UBCO\\2013-2014\\people page\\2013\\Maple\\file001data.txt",
      [float]);
                   data1 := [1.0, 1.1, 1.2, 1.3, 1.4]                     (2)
```

If you have multiple files to import, it would be hassle to write many individual *readdata* commands. If the files are named in a sensible way, we can use a for loop to complete this task more compactly.

First we'll need to understand some commands for manipulating strings. To access these tools, we need to first load the *StringTools* package. The first one that we'll look at is *Join*.

```
> Join(["Hello", "World"]);
   Join(["a", "b", "c", "d"]);
                            "Hello World"
                              "a b c d"                                   (3)
```

By default *Join* pusts a space between the strings that are being joined. That can be changed using an additional option as follows:

```
> Join(["Hello", "World"], "");
  Join(["Hello", "World"], " & ");
   Join(["a", "b", "c", "d"], "A");


                            "HelloWorld"
                           "Hello & World"
                             "aAbAcAd"                                    (4)
```

Next, let's look at *sprintf* which can be used to format numbers. I don't know in detail all of the possible output formats from this command -- you can check the Maple help file if you need to. In the example below the "08" before the "." indicates that the output string should be 8 characters long. The "3" after the "." indicates that there should be 3 characters after the decimal point. The rest of the characters should be leading zeros.

```
> sprintf("%08.3f", 3.2);
                             "0003.200"                                   (5)
```

Suppose that you have files named *001*.txt, *002*.txt, *003*.txt, ... The following *sprintf* command will produce the apprpriate format for the numbers contained in the file names.

```
> sprintf("%03.f", 1);
```

```
    sprintf("%03.f", 2);
    sprintf("%03.f", 3);
    sprintf("%03.f", 53);
                                        "001"
                                        "002"
                                        "003"
                                        "053"                                   (6)
```

Now let's import the same file that was imported using the first *readdata* command above. First, construct the appropiate sting for the path using the *Join* and *sprintf* commands.

```
>  str := Join(["G:\\UBCO\\2013-2014\\people page\\2013\\Maple\\file", sprintf("%03.f", 1),
       "data.txt"], "");
    data1 := readdata(str, [float]);
                str := "G:\UBCO\2013-2014\people page\2013\Maple\file001data.txt"
                              data1 := [1.0, 1.1, 1.2, 1.3, 1.4]                 (7)
```

Now let's attempt to use the *seq* command to import multiple files.

```
>  data := [seq(readdata(Join(["G:\\UBCO\\2013-2014\\people page\\2013\\Maple\\file",
       sprintf("%03.f", i), "data.txt"], "")), i = 1 ..5)];
    data[1];
    data[3];
data := [[1.0, 1.1, 1.2, 1.3, 1.4], [2.0, 2.1, 2.2, 2.3, 2.4], [3.0, 3.1, 3.2, 3.3, 3.4], [4.0, 4.1, 4.2,
    4.3, 4.4], [5.0, 5.1, 5.2, 5.3, 5.4]]
                                   [1.0, 1.1, 1.2, 1.3, 1.4]
                                   [3.0, 3.1, 3.2, 3.3, 3.4]                     (8)
```

Here we'll import only the files with even numbers.

```
>  data := [seq(readdata(Join(["G:\\UBCO\\2013-2014\\people page\\2013\\Maple\\file",
       sprintf("%03.f", i), "data.txt"], "")), i = 2 ..5, 2)]
                   data := [[2.0, 2.1, 2.2, 2.3, 2.4], [4.0, 4.1, 4.2, 4.3, 4.4]]  (9)
```

Finally, if your file number is not sequential, you can make a list of file numbers to import and then use *seq* as shown below. Note that we've now managed to import 7 different files using two line of code. Obviously, this can be extended to import any number of files.

```
>  k := [1, 2, 3, 4, 5, 8, 12]:
    data := [seq(readdata(Join(["G:\\UBCO\\2013-2014\\people page\\2013\\Maple\\file",
       sprintf("%03.f", k[i]), "data.txt"], "")), i = 1 ..nops(k))]

data := [[1.0, 1.1, 1.2, 1.3, 1.4], [2.0, 2.1, 2.2, 2.3, 2.4], [3.0, 3.1, 3.2, 3.3, 3.4], [4.0, 4.1, 4.2,     (10)
    4.3, 4.4], [5.0, 5.1, 5.2, 5.3, 5.4], [8.0, 8.1, 8.2, 8.3, 8.4], [12.0, 12.1, 12.2, 12.3, 12.4]]
```

Sometimes, you may not want to store the entire contents of multiple files into the memory of Maple. For example, the files may be very large (maybe images). Instead, you might want to immediately perform the same operation on the data of each file and then just store the results of that operation. Below, each data file is imported and the average of the data is found. We store only the result of the average operations.

```
>  avgData
```

```
    := [seq(Mean(readdata(Join(["G:\\UBCO\\2013-2014\\people
   page\\2013\\Maple\\file", sprintf("%03.f", k[i]), "data.txt"], ""))), i = 1 ..nops(k))]
avgData := [1.200000000, 2.200000000, 3.200000000, 4.200000000, 5.200000000,        (11)
    8.200000000, 12.20000000]
```

The **for**loop below performs the same task as the above *seq* statement.

```
> meanList := NULL :
  for i from 1 to nops(k) do:
   dataFile := readdata(Join(["G:\\UBCO\\2013-2014\\people page\\2013\\Maple\\file",
      sprintf("%03.f", k[i]), "data.txt"], ""));
   meanList := meanList, Mean(dataFile);
   end do:
   meanList := [meanList];

meanList := [1.200000000, 2.200000000, 3.200000000, 4.200000000, 5.200000000,       (12)
    8.200000000, 12.20000000]
> 
```