

Proposal: File Management Language (FML)

Cara J. Borenstein
Columbia University
cjb2182@columbia.edu

Daniel Garzon
Columbia University
dg2796@columbia.edu

Daniel L. Newman
Columbia University
dln2111@columbia.edu

Pranav Bhalla
Columbia University
pb2538@columbia.edu

Rupayan Basu
Columbia University
rb3034@columbia.edu

September 23, 2013

Abstract

This is the paper's abstract ...

1 Introduction

This is an **introduction** to our *language* ...

Language Goals The remainder of this article is organized as follows. Section 2 gives account of previous work. Our new and exciting results are described in Section 3. Finally, Section 4 gives the conclusions.

2 Syntax

2.1 Basic Data Types

Table 1: Basic Data Types

primitive	Description
<i>integer</i>	...Some text
<i>boolean</i>	...
<i>character</i>	...
<i>string</i>	...
<i>directory</i>	...
<i>file</i>	...
<i>date</i>	...
<i>ftype</i>	...
<i>tag</i>	...

2.2 Mathematical Operators

Table 2: Mathematical Operators.

operator	Description
+	...Some text
—	...
*	...
,	...

2.3 Logical Operators

Table 3: Logical Operators.

operator	Description
==	...Some text
!=	...
>	...
>=	...
<	...
<=	...

2.4 Control Statements

2.4.1 *if-then-else*

```
if <condition> then
  <expression>
else
  <expression>
end
```

2.4.2 *while*

```
while <condition> then
  <expression>
end
```

2.4.3 *for*

```
for <identifier> in <set of files> do
  <expression>
end
```

2.5 Function Definition

```
def <identifier> (<parameter list>)  
  <expression>  
end
```

2.6 Built-in Functions

3 Example Program

In this section we describe the results.

```
1  int main(int argc, char const *argv[])  
2  {  
3    printf("%s\n", "Hello World!");  
4    return 0;  
5  }
```

Listing 1: Code for ...

4 Conclusions

We worked hard, and achieved very little.