

Matrix Literals with Strings in CMAT©

Wolfgang M. Hartmann

January 2006

A new language feature, a grammar modification for the already existing matrix literals, was implemented. Inside the brackets [and] of a matrix literal, identifiers (names starting with a letter and containing only letters, digits and underscores) are treated as object names. Strings had to be embedded into quotes. This syntax is consistent with the C language and is among others similar in Matlab. For example, the input

```
a = [ 1 2 , 3 4 ];
print "A=", a;
b = [ a a , a a ];
print "B=",b;
```

leads into

A=		1	2		B=		1	2	3	4
-----					-----					
1		1	2		1		1	2	1	2
2		3	4		2		3	4	3	4
					3		1	2	1	2
					4		3	4	3	4

Now, the two new token [" and "]" can be used as brackets of a matrix literal where identifiers are treated as strings and not as object names. Such a form of matrix literal is similar to that used in SAS/IML® and will probably find lots of use for specifying small matrices of mixed, string and numeric, data type. The following two examples result in the same mixed data type matrix:

```
school = [ 1 "regular" "self" 10 , 1 "regular" "team" 17 ,
           1 "regular" "class" 26 , 1 "afternoon" "self" 5 ,
           1 "afternoon" "team" 12 , 1 "afternoon" "class" 50 ,
           2 "regular" "self" 21 , 2 "regular" "team" 17 ,
           2 "regular" "class" 26 , 2 "afternoon" "self" 16 ,
           2 "afternoon" "team" 12 , 2 "afternoon" "class" 36 ,
           3 "regular" "self" 15 , 3 "regular" "team" 15 ,
           3 "regular" "class" 16 , 3 "afternoon" "self" 12 ,
```

```
3 "afternoon" "team" 12 , 3 "afternoon" "class" 20 ];
```

```
cnam = [ "School" "Program" "Style" "Count" ];  
school = cname(school,cnam);  
print "Data 1=", school;
```

The new form does not need the so many quotes around each string for telling CMAT that the identifiers are strings. Please note, that there must not be white space separating the brackets and strings, the two symbols are recognized as one token.

```
school2 = [" 1 regular self 10 , 1 regular team 17 ,  
1 regular class 26 , 1 afternoon self 5 ,  
1 afternoon team 12 , 1 afternoon class 50 ,  
2 regular self 21 , 2 regular team 17 ,  
2 regular class 26 , 2 afternoon self 16 ,  
2 afternoon team 12 , 2 afternoon class 36 ,  
3 regular self 15 , 3 regular team 15 ,  
3 regular class 16 , 3 afternoon self 12 ,  
3 afternoon team 12 , 3 afternoon class 20 "];  
school2 = cname(school2,cnam);  
print "Data 2=", school2;
```

Using Yacc and Lex, this language extension was programmed and tested in about two hours. Until now a large data set containing many string data (without quotes) was usually stored in a separate file and later read from there using the `fread()` function.