

Most Common JSL Mistakes and How to Avoid Them

Wendy Murphrey, Sr Principal Technical Support Analyst, SAS

Justin Chilton, Sr Associate Software Development Engineer in Test, SAS

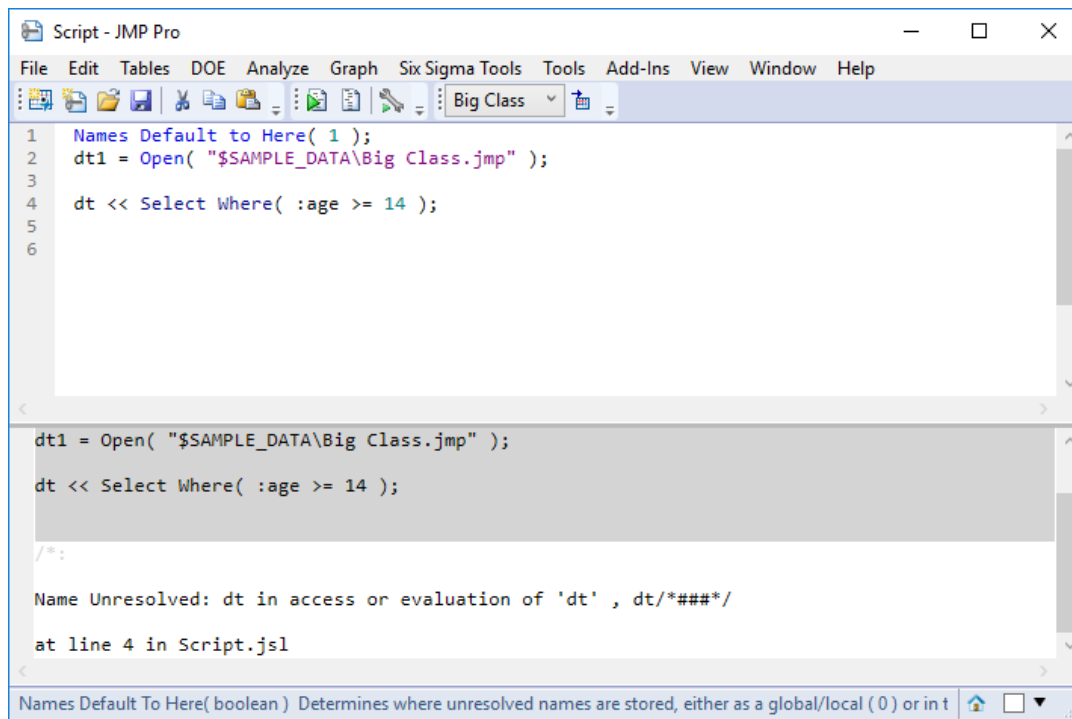


Not Checking the Log

Example 1

Example 2

The log offers important details about the script execution



```
Script - JMP Pro
File Edit Tables DOE Analyze Graph Six Sigma Tools Tools Add-Ins View Window Help
Big Class
1 Names Default to Here( 1 );
2 dt1 = Open( "$SAMPLE_DATA\Big Class.jmp" );
3
4 dt << Select Where( :age >= 14 );
5
6

dt1 = Open( "$SAMPLE_DATA\Big Class.jmp" );

dt << Select Where( :age >= 14 );

/*:
Name Unresolved: dt in access or evaluation of 'dt' , dt/****/
at line 4 in Script.jsl

Names Default To Here( boolean ) Determines where unresolved names are stored, either as a global/local ( 0 ) or in t
```

Depending on the Current Data Table

Best practice: Data table specific

- Send platform calls as messages to the desired data table reference:

```
dt1 << Bivariate( Y( :weight ), X( :height ), "Fit Line" );
```

- Create new columns using New Column() as a message instead of a function:

```
dt1 << New Column( "Mean Height by Age",  
  "Numeric",  
  Formula( Col Mean( :height, :age ) )  
)
```

Depending on the Current Data Table

Best practice: Data table specific

- Use optional data table references when referencing columns:

```
Column( dt1, "age" )  
//or  
dt1:age  
//or  
As Column( dt1, "age" )
```

- Use optional data table references in functions, where available:

```
For Each Row( dt1, <script> );  
Summarize( dt1, <options> );
```

Depending on the Current Data Table

Example

What not to do

- Refrain from assigning a data table reference using the Current Data Table() function in production scripts:

```
dt << Join(  
  With( dt2 ),  
  Select( :popcorn, :oil amt, :batch, :yield ),  
  SelectWith( :yield ),  
  By Matching Columns( :popcorn = :popcorn, :batch = :batch, :oil amt = :oil )  
);  
newDT = Current Data Table(); //Not recommended
```

Mis-using Column() In Select Where

Column() vs. As Column()

- Column() – Returns a reference to the column as a whole.

```
Column( dt, "age" )
```

```
Column( dt, "age" )[5]
```

- Column()
 - Platform calls
 - Sending messages to the column
 - Assigning variable reference to a column

Mis-using Column() In Select Where

Example

Column() vs. As Column()

- As Column() – Returns a reference to the value on the current row of the specified column.

```
As Column( dt, "age" )
```

```
:age
```

```
dt:age
```

- As Column()
 - Formulas
 - For Each Row()
 - Select Where() or Get Rows Where()

Leaving Variables in Column Formulas

Example

Solution

Replace variables with their values

- The lifespan of a JSL variable is limited to the current JMP session.
- As the value of the variable changes, so will the result of the formula.

Unless you look at the formula in the formula editor, you may not even realize there is a problem.

Incorrect Syntax for Setting Column Properties

Add Interactively First

Example

- The column...
 - is a repository for property information.
 - does not perform syntax-checking.
- The consumer of the property generally expects to receive the information in a specific format.

Manipulating Dispatch Commands

Example

- Dispatch() Command
 - Developed as an internal method for JMP to recreate graphs with customizations.
- Some arguments of a Dispatch command do not accept JSL variables

Manipulating Dispatch Commands

Solution 1

Solution 2

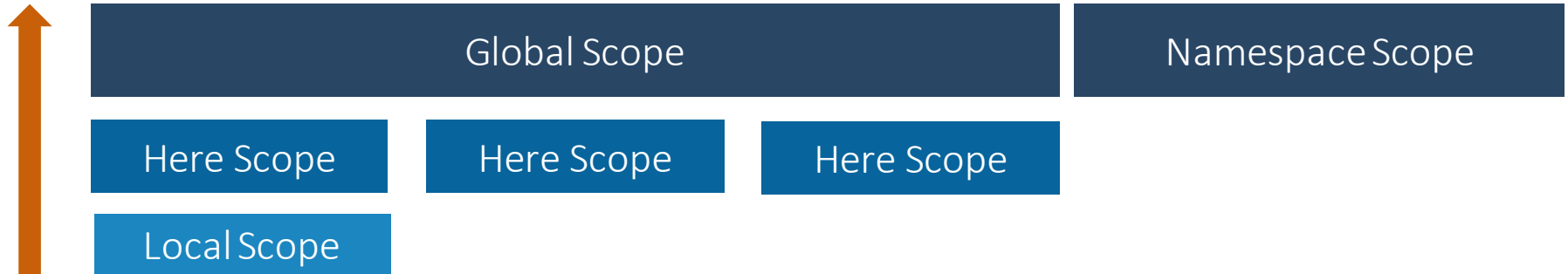
Solutions

- Use Eval/EvalExpr/Expr solution
- Use DisplayBox subscribing or XPath to send messages directly

Only Using Global Namespace

Example

- The global namespace is great for quick scripts
 - May not be suitable for larger production scripts.



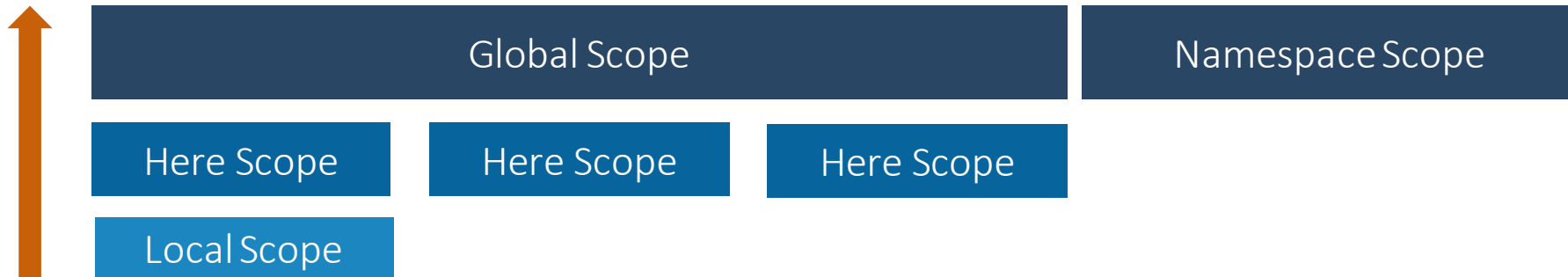
Only Using Global Namespace

Solution 1

Solution 2

Solution

- Names Default To Here(1)
- Custom Namespace



Overlooking Resources

- Scripting Index (Help > Scripting Index)
- [Online Documentation](#) (Help > JMP Help)
- [JMP User Community](#)
 - Blogs
 - Discussions
 - File Exchange
- [On-Demand Webcasts](#)
 - [Application Development](#)

Overlooking Resources

- [Technical Support](#)
 - [Report a Problem](#)
 - support@jmp.com
- [Books](#)
 - [Jump into JMP Scripting, Second Ed.](#)
 - [JSL Companion: Applications of the JMP Scripting Language, Second Ed.](#)
- [Training](#)
- [Mentoring Services](#)

Thank You

jmp.com