

ArcherCalc Guide

October 16, 2015

Contents

1	Introduction	2
2	Installation	4
3	Using ArcherCalc	5
3.1	Introduction	5
3.2	Command Reference	5
3.2.1	:browse	5
3.2.2	:emit	5
3.2.3	:load	5
3.2.4	:quit	5
3.2.5	:type	5
4	Syntax Reference	6
4.1	Comments	6
4.2	Literals	6
4.2.1	Numbers	6
4.2.2	Text	7
4.2.3	Boolean	7
4.2.4	Dates	7
4.3	Functions	7
4.4	Control Flow	7
4.5	Fields	8
4.6	Built-in Functions	8
4.7	Type Signatures	8

Chapter 1

Introduction

ArcherCalc provides a way to break down Archer calculations into re-usable and manageable pieces that can be type-checked, documented, tested, and compiled back into a format suitable for Archer.

Here is an example. The following is a file named “test.ac”:

```
//Archer fields that will be used in our calculations.
let incid_date    = field "Date of Incident" : Date as ISODate "2015-04-05";
let incid_status = field "Incident Status"  : Text as "Open";

// Calculate the number of days elapsed since a given date.
let daysAgo x = datedif x now Day;

// Given a date, return a color based on the age.
let date_color date =
  if ((daysAgo date) < 30) then
    "Green"
  else if ((daysAgo date) < 60) then
    "Yellow"
  else if ((daysAgo date) < 90) then
    "Orange"
  else
    "Red";

// Color items with a status of "Open"
let incid_color stat dt =
  if (stat == "Open") then
    (date_color dt)
  else
    "";
```

Note that we have not used the fields in the body of our functions, although it is possible to do so. This is considered a best practice for re-usability. Functions

with the field embedded can only be used on that particular field, but a function *taking* a field can work with any field of the same type.

Let's start ArcherCalc and load the file:

```

      _/ \   _ _ _ _ _ | | _ _ _ _ _ _ _ _ _ / _ _ _ | _ _ | | _ _ _
     / _ \ | ' _ _ / _ _ | ' _ \ / _ \ ' _ _ | | / _ ' | | / _ _ | Version 0.0.1.0
    / _ _ \ | | | ( _ _ | | | _ _ / | | | _ _ | ( _ _ | | ( _ _ Type :help for help
   / _ /   \ _ \ | \ _ _ | _ | \ _ _ | _ | \ _ _ \ _ _ , _ | \ _ _ |

```

ArcherCalc is provided with ABSOLUTELY NO WARRANTY.

```
ArcherCalc> :load test.ac
```

Now lets test the date_color function:

```

ArcherCalc> date_color ISODate "2015-05-05"
"Red" : Text
ArcherCalc> date_color ISODate "2015-08-05"
"Orange" : Text
ArcherCalc> date_color ISODate "2015-09-05"
"Yellow" : Text
ArcherCalc> date_color ISODate "2015-10-05"
"Green" : Text

```

We can also run it with our defined fields:

```

ArcherCalc> date_color incid_date
"Red" : Text

```

Now let's compile the previous in a format suitable for Acher:

```

ArcherCalc> :emit date_color incid_date
IF(DATEDIF([Date of Incident],NOW(),DAY)<30,"Green",IF(DATEDIF([Date of Incident],
NOW(),DAY)<60,"Yellow",IF(DATEDIF([Date of Incident],NOW(),DAY)<90,"Orange",
"Red")))

```

Chapter 2

Installation

TODO

Chapter 3

Using ArcherCalc

3.1 Introduction

3.2 Command Reference

3.2.1 :browse

3.2.2 :emit

3.2.3 :load

3.2.4 :quit

3.2.5 :type

Chapter 4

Syntax Reference

4.1 Comments

```
// Single-line comment

/* Comment
   that spans
   multiple lines. */
```

4.2 Literals

4.2.1 Numbers

Numbers can be entered verbatim, with the exception that negative numbers need parenthesis.

```
// Normal
1
3.14159

// Hexadecimal
0x90
0xAB
0xfe

// Octal
0o72
0065

// Scientific notation
1e12
1E-5
```

```
// Negative numbers
(-0.3)
(-3.14159)
```

4.2.2 Text

Text is surrounded by double quotes. Double quotes are escaped with a backslash (\).

```
"This is a test."
"a"
"They said \"All good!\""
```

4.2.3 Boolean

```
True
False
```

4.2.4 Dates

If a time is not specified, midnight UTC is assumed.

```
// ISO Style
ISODate "2015-04-02T02:00:00Z"
ISODate "2015-04-02"
ISODate "2015-04-02T04:00:00PDT"

// Archer Style
ArDate "04/02/2015 02:00:00"
ArDate "04/02/2015"
```

4.3 Functions

```
let double x = x*2;

let circ_area r = pi*r^2;

let rect_area w h = w * h;
```

4.4 Control Flow

```
if (temp > 80) then "hot" else "cold";

if (risk >= 4) then
```



```
    "red"  
else if (risk >= 3) then  
    "orange"  
else  
    "green";
```

4.5 Fields

The format for defining a field is:

```
field "Field name" : Type as Value;
```

“Field name” is the actual name of the field in Archer. The Type is the type of field in Archer (Date, Text, Num, etc.) and is required. The “as Value” is optional, but let’s one evaluate and test the field in ArcherCalc.

Here are some examples:

```
field "Number of Incidents" : Num as 132;  
field "Date of Incident" : Date as ISODate "2015-04-05";  
field "Incident Status" : Text as "Open";  
field "Risk Rating" : Text;
```

4.6 Built-in Functions

4.7 Type Signatures