

# Aarhus 2015 CONFERENCE

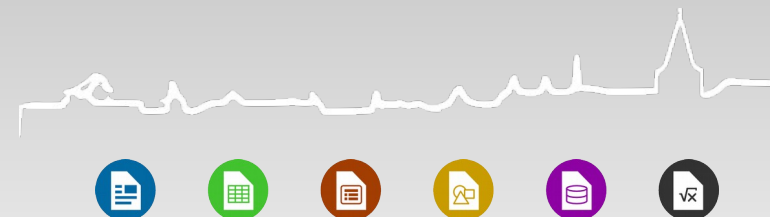


## Implementation of Table Structured References in LibreOffice Calc

- ▼ Eike Rathke, Red Hat, Inc.
- ▼ 2015-09-25



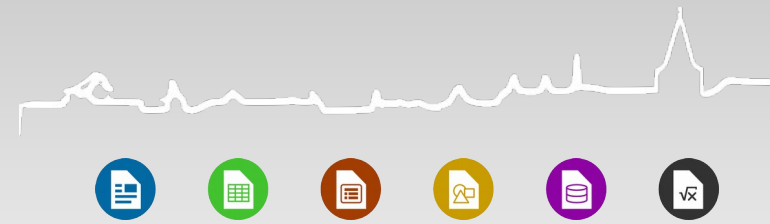
# About the Speaker



- ▼ Eike Rathke, known on the net as erAck
- ▼ Based in Hamburg, Germany
- ▼ Worked on StarOffice from 1993 to 2000 for Star Division
- ▼ Worked on OpenOffice.org from 2000 to 2011 for Sun Microsystems and one other company
- ▼ Works on LibreOffice since 2011, employed by Red Hat, Inc.
- ▼ Areas of expertise:
  - ▼ Calc core, formula compiler and interpreter
  - ▼ number formatter/scanner
  - ▼ i18n framework, locale data
- ▼ Also mentor and knowledge spreader whenever possible
- ▼ Web site <http://erack.de/>

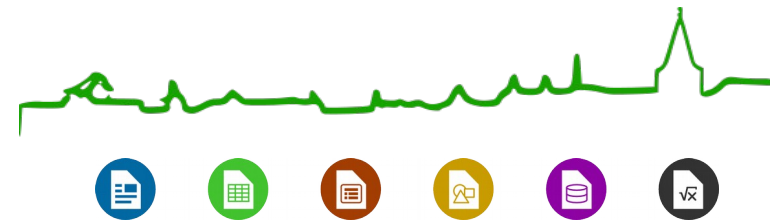


# Agenda



- ▼ Brief overview on formula compilation
- ▼ Challenges of Table structured references
- ▼ Imagine all the tokens
- ▼ Solve the puzzle





# Brief overview on formula compilation

Only what is needed to understand the later approach ...



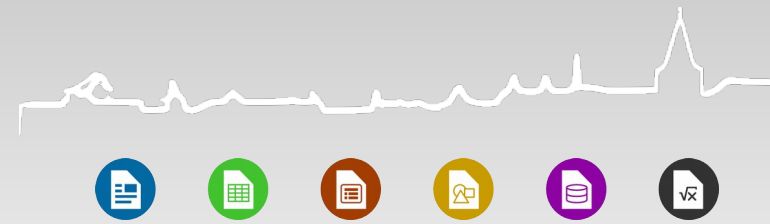
# Compiling a Formula



=SUM(A2:A4)



# Compiling a Formula



=SUM(A2:A4)

## ▼ Two passes

1. Scan text into tokens, each token with

- ▼ OpCode (enum values of functions, operators, ...)
- ▼ StackVarEnum (type of token)
- ▼ Stored in FormulaTokenArray::pCode



# Compiling a Formula



SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose



# Compiling a Formula

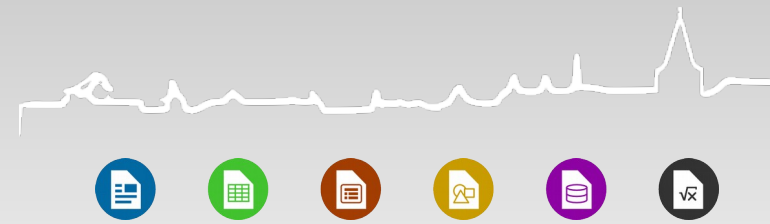


SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep





# Compiling a Formula



=SUM(A2:A4)

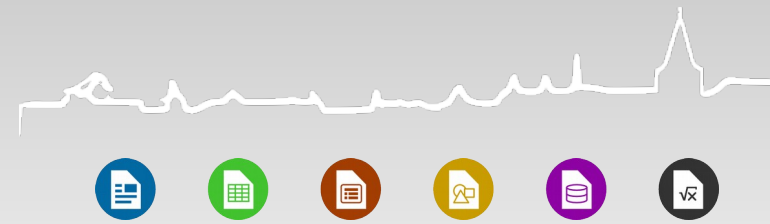
## ▼ Two passes

1. Scan text into tokens, each token with

- ▼ OpCode (enum values of functions, operators, ...)
- ▼ StackVarEnum (type of token)
- ▼ Stored in FormulaTokenArray::pCode



# Compiling a Formula



=SUM(A2:A4)

## ▼ Two passes

1. Scan text into tokens, each token with

- ▼ OpCode (enum values of functions, operators, ...)
- ▼ StackVarEnum (type of token)
- ▼ Stored in FormulaTokenArray::pCode

2. Create a sequence of interpret-able tokens

- ▼ in Reverse Polish Notation (RPN)
- ▼ through a recursive descending parser
- ▼ Stored in FormulaTokenArray::pRPN



# Compiling a Formula



SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep



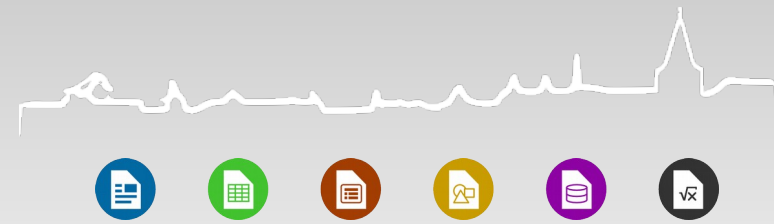
# Compiling a Formula



SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep
ocSum			



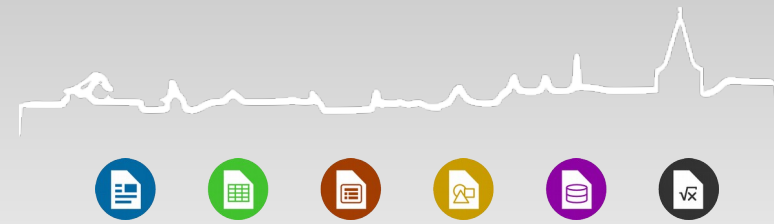
# Compiling a Formula



SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep
ocSum	┌		
	└		



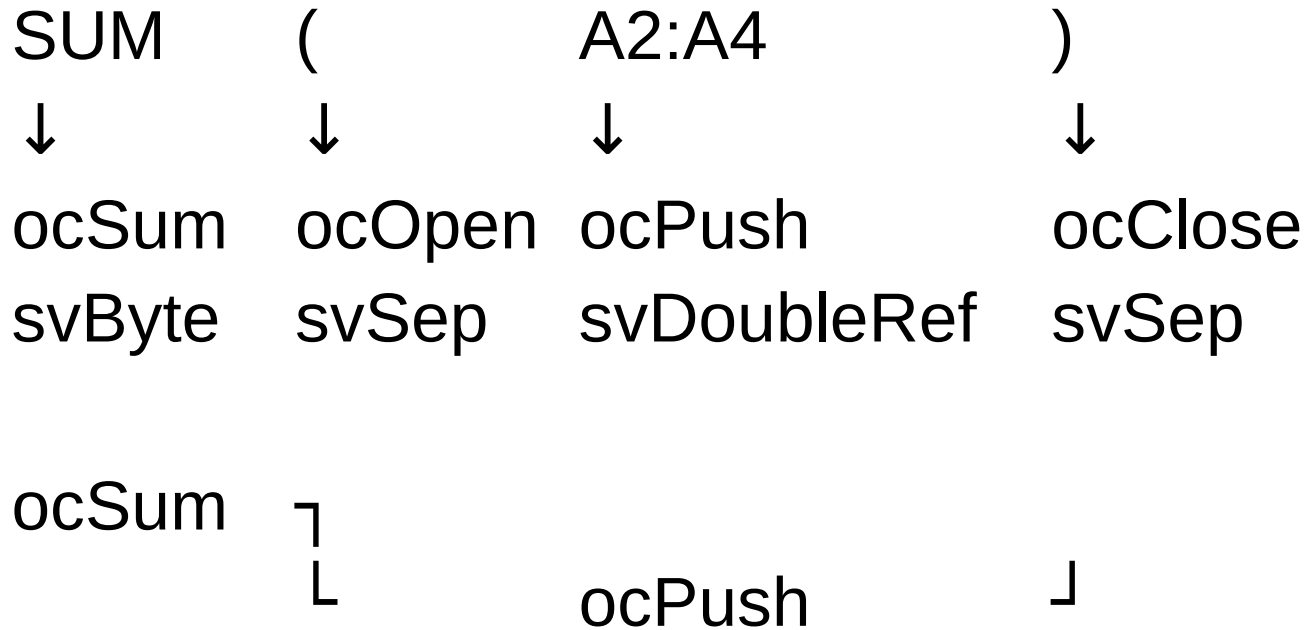
# Compiling a Formula



SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep
ocSum	┌ └	ocPush	



# Compiling a Formula



# Compiling a Formula



SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep

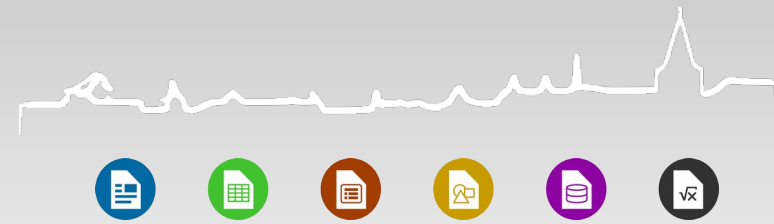
ocSum 7  
L

ocPush





# Compiling a Formula

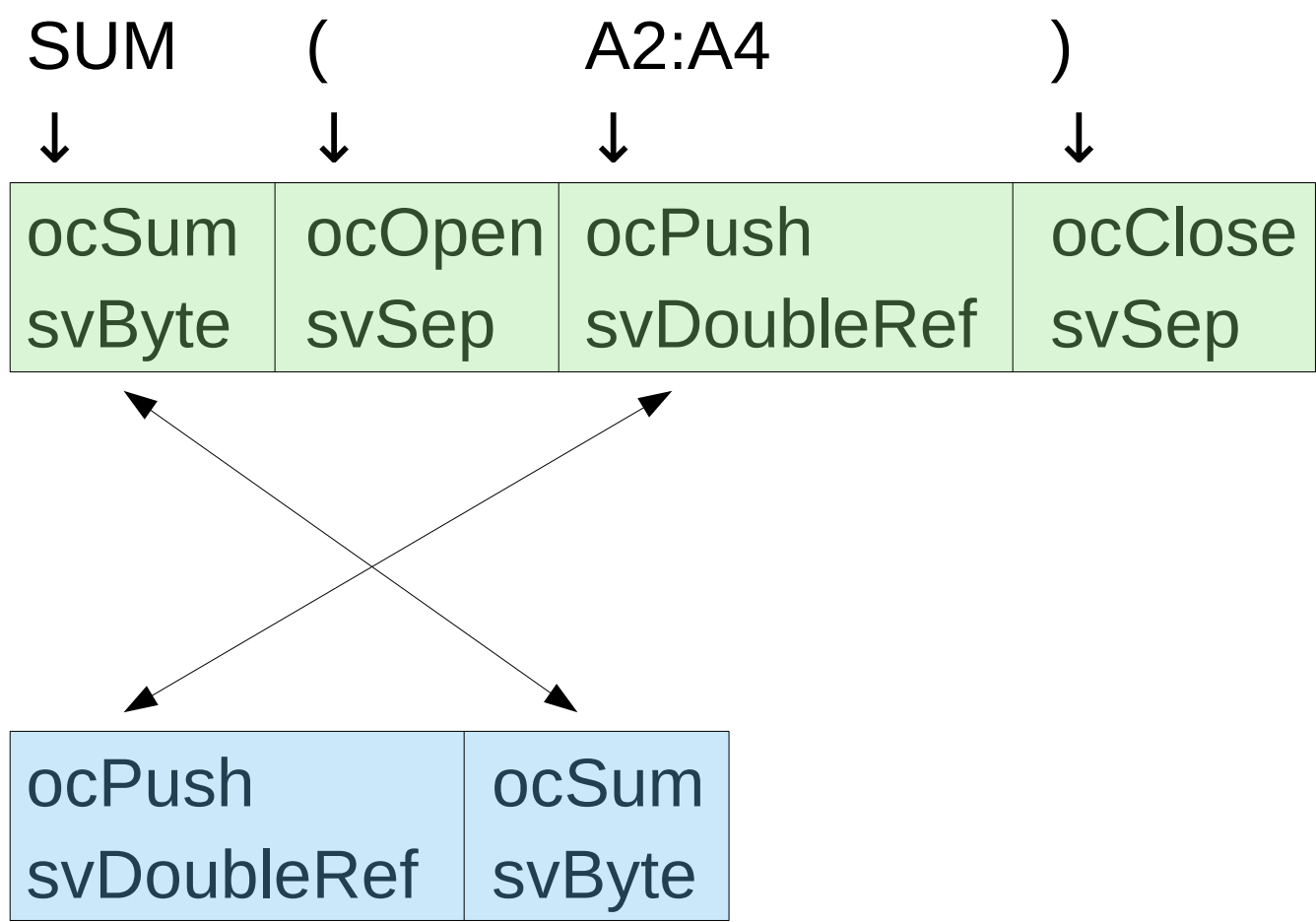


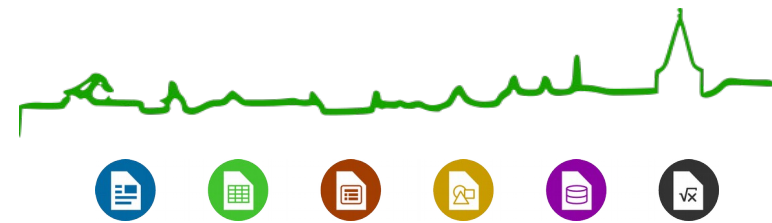
SUM	(	A2:A4	)
↓	↓	↓	↓
ocSum	ocOpen	ocPush	ocClose
svByte	svSep	svDoubleRef	svSep

ocPush            ocSum



# Compiling a Formula





# Challenges of Table structured references

It might look easy, but ...



# Challenges of TableRef



Header1	Header2
Data11	Data21
Data12	Data22
Data13	Data23
Total1	Total2

- ▼ A table is a cell range defined as a named database range
  - ▼ Optional header row
  - ▼ One or more data rows
  - ▼ Optional totals row
- 
- ▼ Table structured references provide a syntax to address different parts of such table
  - ▼ Assume this sample table to be named Table and to cover cells A1:B5 for the examples later



# Challenges of TableRef



- ▼ structure-reference = [table-identifier] intra-table-reference
- ▼ table-identifier = [book-prefix] table-name
- ▼ table-name = name
- ▼ intra-table-reference = spaced-lbracket inner-reference spaced-rbracket / keyword / ("[" [simple-column-name] ")")
- ▼ inner-reference = keyword-list / ([keyword-list spaced-comma] column-range)
- ▼ keyword = "[#All]" / "[#Data]" / "[#Headers]" / "[#Totals]" / "[#This Row]"
- ▼ keyword-list = keyword / ("[#Headers]" spaced-comma "[#Data]") / ("[#Data]" spaced-comma "[#Totals]")
- ▼ column-range = column [":" column]
- ▼ column = simple-column-name / ("[" \*space simple-column-name \*space ")")
- ▼ simple-column-name = [any-nospace-column-character \*any-column-character] any-nospace-column-character
- ▼ escape-column-character = tick / "[" / "]" / "#"
- ▼ tick = %x27
- ▼ unescaped-column-character = character ; MUST NOT match escape-column-character or space
- ▼ any-column-character = any-nospace-column-character / space
- ▼ any-nospace-column-character = unescaped-column-character / (tick escape-column-character)
- ▼ spaced-comma = [space] comma [space]
- ▼ spaced-lbracket = "[" [space]
- ▼ spaced-rbracket = [space] "]"

(Definitions excerpt of [MS-XLSX]: 2.2.2 Formulas, <https://msdn.microsoft.com/en-us/library/dd906358.aspx>)

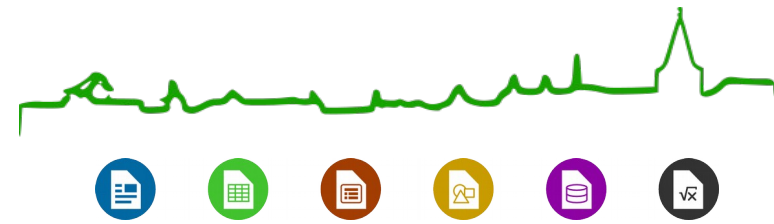


# Challenges of TableRef



Example Reference	Resulting Range
Table[#All]	A1:B5 (entire table)
Table[]	A2:B4 (data area)
Table[#Data]	A2:B4 (data area)
Table[#Headers]	A1:B1 (header row)
Table[#Totals]	A5:B5 (totals row)
Table[Header1]	A2:A4 (data of column Header1)
Table[#Data],[Header1]	A2:A4 (data of column Header1)
Table[#Totals],[Header1]	A5 (totals of column Header1)
Table[#Headers],[#Data],[Header1]	A1:A4 (header and data of column Header1)
Table[#This Row]	A1:B1,A2:B2,... (all cells of this row)
Table[#This Row],[Header1]	A1,A2,... (this row's cell of column Header1)
Table[#This Row],[Header1]:[Header2]	A1:B1,A2:B2,... (this row's cells of column range)





Imagine all the tokens

One for each...



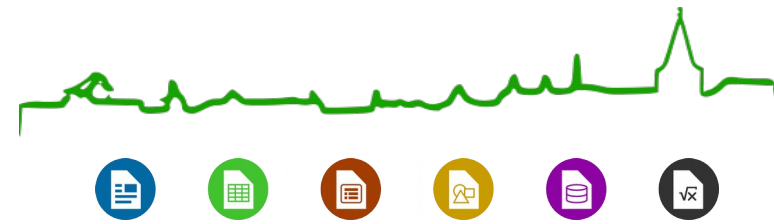
# Imagine all the tokens



Symbol	Token
Table (name)	ocDBArea, already existed
[	ocTableRefOpen
]	ocTableRefClose
#All	ocTableRefItemAll
#Headers	ocTableRefItemHeaders
#Data	ocTableRefItemData
#Totals	ocTableRefItemTotals
#This Row	ocTableRefItemThisRow
,	ocSep
:	ocRange
... resulting token / reference	ocTableRef





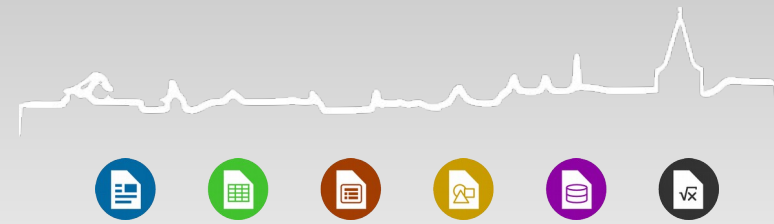


# Solving the puzzle

With yet another new token...



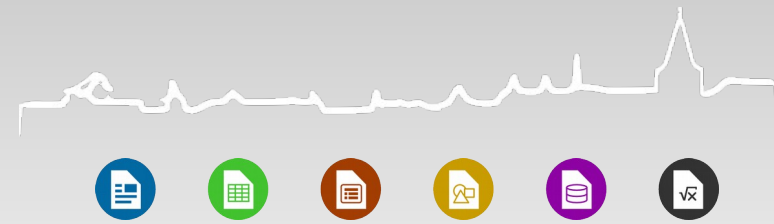
# Solving the puzzle



```
=SUM(Table[[#Data],[Header1]])
```



# Solving the puzzle



```
=SUM(Table[#Data],[Header1])  
Table[#Data],[Header1]
```



# Solving the puzzle



Table      [[#Data],[Header1]]



ocDBArea  
svIndex



# Solving the puzzle

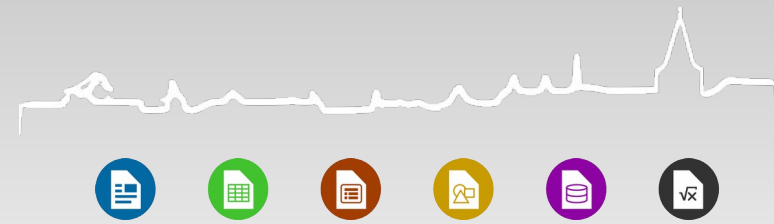


Table	[	[#Data],[Header1]]
↓	↓	
ocDBArea	ocTableRefOpen	
svIndex	svSep	





# Solving the puzzle

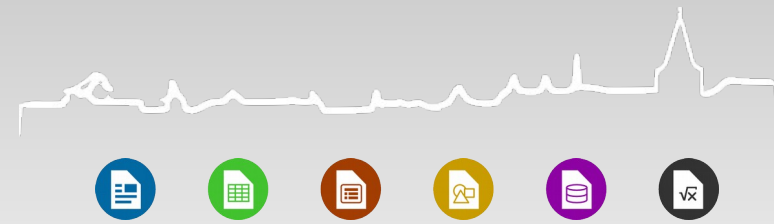


Table	[	[#Data],[Header1]]
↓	↓	
ocTableRef	ocTRO	
svIndex	svSep	



# Solving the puzzle

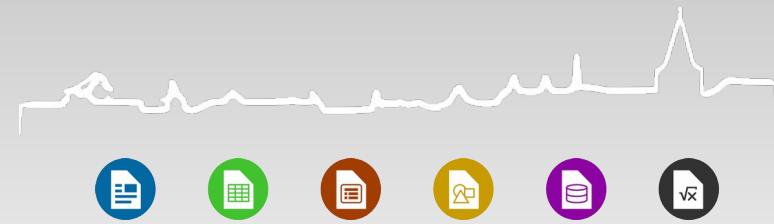


Table	[	[	#Data],[Header1]]
↓	↓	↓	
ocTableRef	ocTRO	ocTRO	
svIndex	svSep		





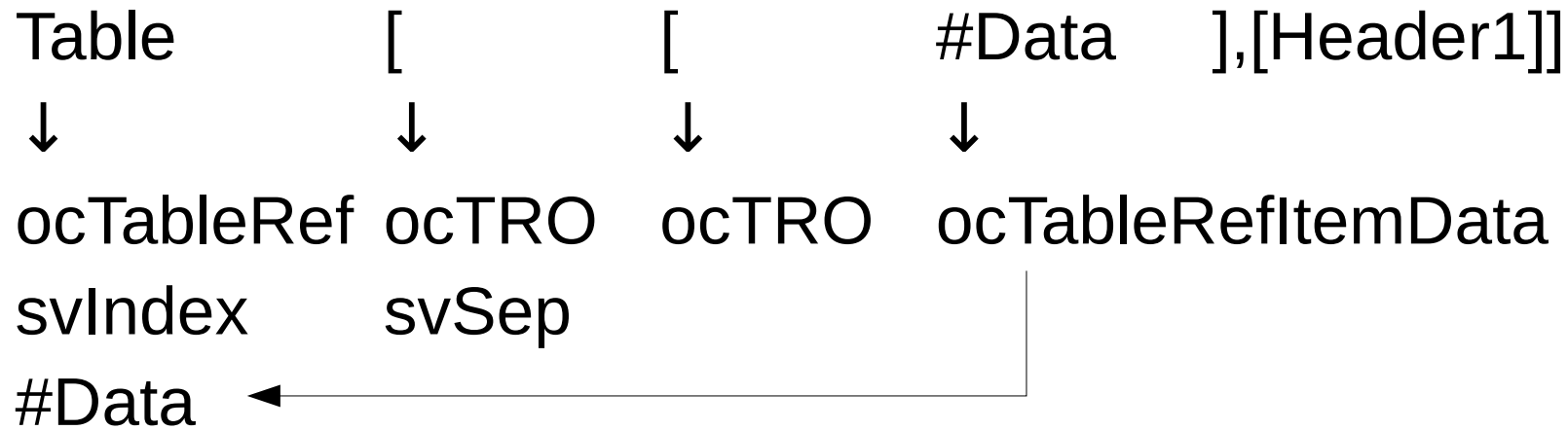
# Solving the puzzle



Table	[	[	#Data	],[Header1]]
↓	↓	↓	↓	
ocTableRef	ocTRO	ocTRO	ocTableRefItemData	
svIndex	svSep			



# Solving the puzzle



# Solving the puzzle



Table	[	[	#Data	]	,[Header1]]
↓	↓	↓	↓	↓	
ocTableRef	ocTRO	ocTRO	ocTRID	ocTRC	
svIndex	svSep				
#Data					



# Solving the puzzle



Table	[	[	#Data	]	,	[	Header1	]
↓	↓	↓	↓	↓	↓	↓	↓	
ocTableRef	ocTRO	ocTRO	ocTRID	ocTRC	ocSep	ocTRO	ocPush	
svIndex	svSep						svSingleRef	
#Data							A1	



# Solving the puzzle

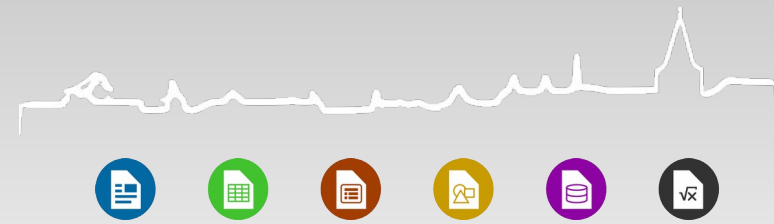


Table	[	[	#Data	]	,	[	Header1	]	]
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
ocTableRef	ocTRO	ocTRO	ocTRID	ocTRC	ocSep	ocTRO	ocPush	C	C
svIndex	svSep						svSingleRef		
#Data							A1		



# Solving the puzzle

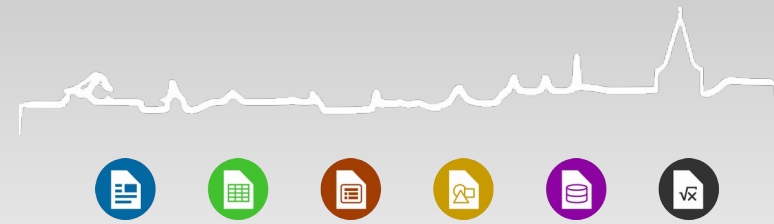


Table	[	[	#Data	]	,	[	Header1	]	]
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
ocTableRef	ocTRO	ocTRO	ocTRID	ocTRC	ocSep	ocTRO	ocPush	C	C
svIndex	svSep						svSingleRef		
#Data							A1		

ocPush  
svDoubleRef  
A2:A4



# Solving the puzzle

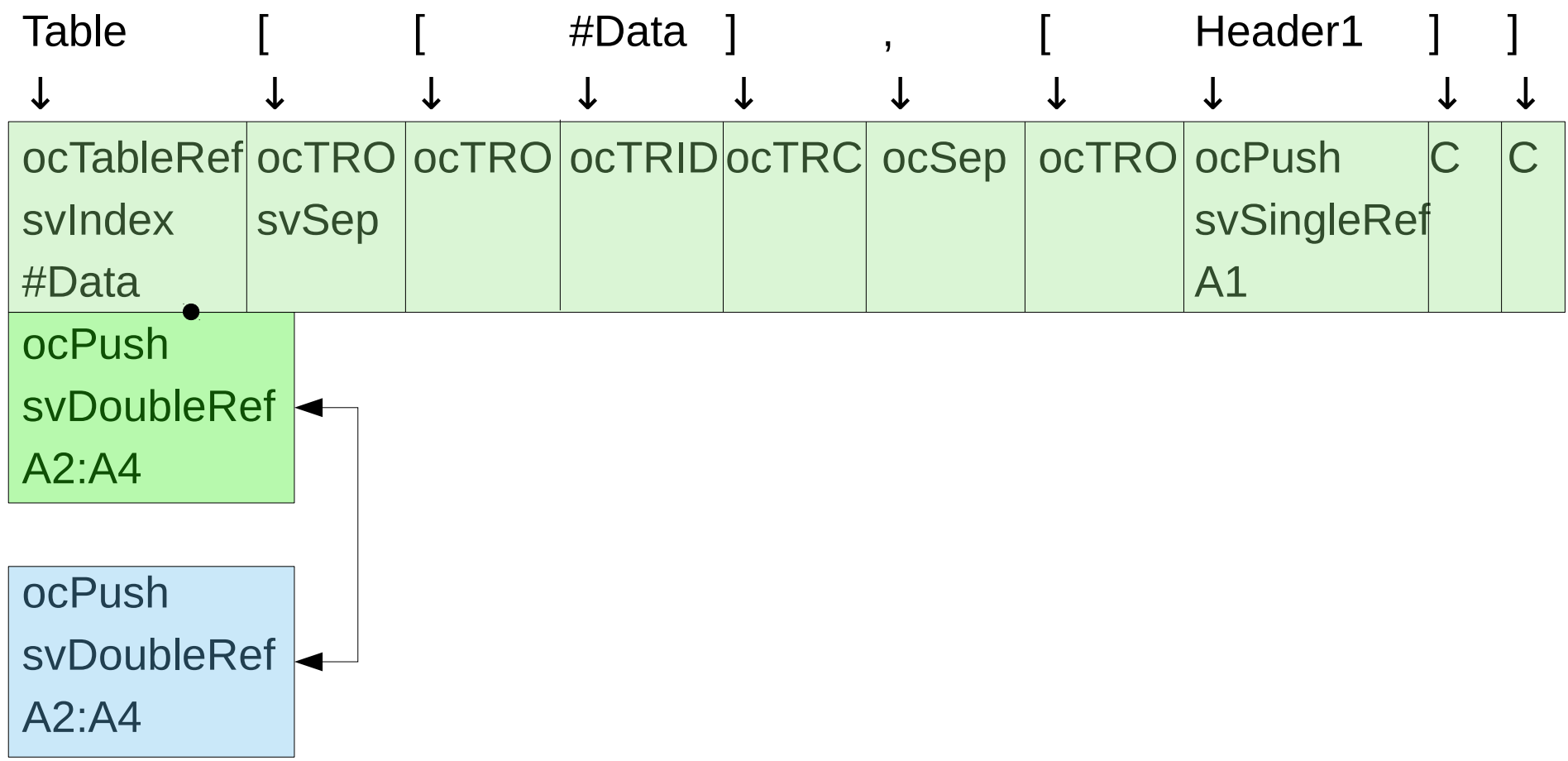


Table	[	[	#Data	]	,	[	Header1	]	]
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
ocTableRef svIndex #Data	ocTRO svSep	ocTRO	ocTRID	ocTRC	ocSep	ocTRO	ocPush svSingleRef A1	C	C

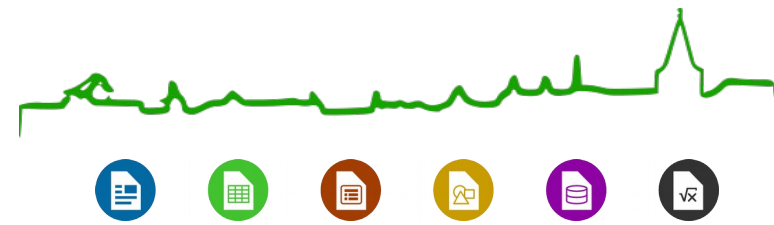
ocPush  
svDoubleRef  
A2:A4



# Solving the puzzle







# Questions?

I might be able to answer...



# Aarhus 2015 CONFERENCE



Thank you ...

- ▾ ... for using LibreOffice!
- ▾ ... for supporting LibreOffice!
- ▾ ... for hacking LibreOffice!



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).

