

Vision for Calc Development Keeping Calc Modular

Kohei Yoshida

Introduction

▼ Who (what) I am

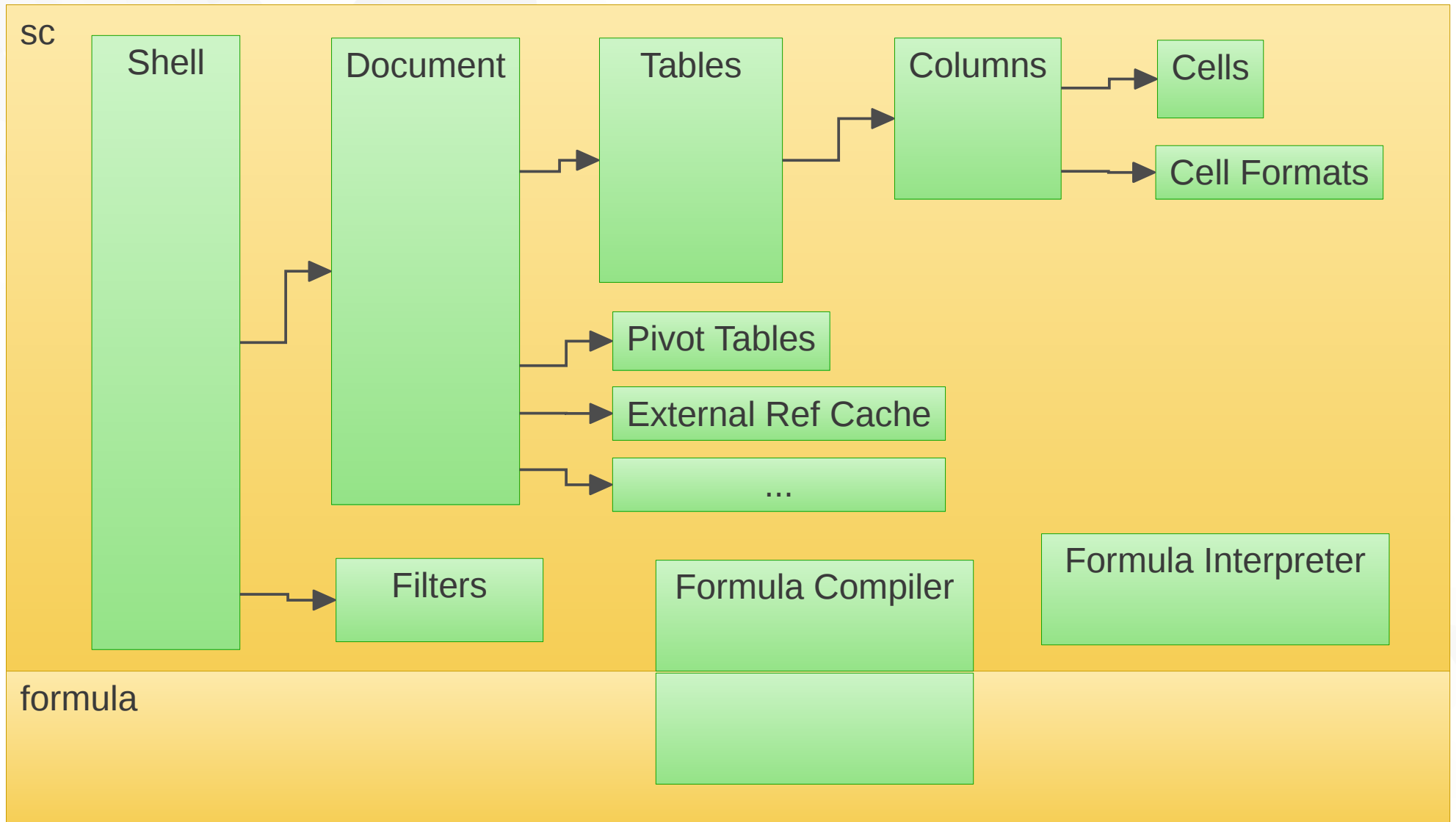
- ▼ Born in Japan, lives in Raleigh, North Carolina.
- ▼ Spare-time hacker turned full-time.
- ▼ Hacking on OOo/LibO since 2004.
- ▼ Software Engineer at Novell since 2007 (later SUSE), with emphasis on LibreOffice Calc.
- ▼ Blog: <http://kohei.us/>

My motivation for LibreOffice

- ▼ **We all have different motivations, volunteer or paid.**
- ▼ **Create a great spreadsheet application that I can be proud of.**
 - ▼ Cross-platform (at least Windows and Linux)
 - ▼ Native UI
 - ▼ Excellent performance
 - ▼ Stability
 - ▼ Ease of maintenance
 - ▼ Great support for various file formats.
- ▼ **LibreOffice Calc is certainly not there yet, but closest.**

Why make it modular

Calc today.... (1,000 feet view)



All in one place - monolithic structure

- ▼ **Almost everything is in sc, with shared bits in other modules.**
- ▼ **Ugly sc/formula separation. Increased complexity.**
- ▼ **Lots of in-house complex data structures deep within Calc's core.**
 - ▼ Cell instance storage
 - ▼ Cell format storage
 - ▼ Row / column attributes - visibility, height / width etc.
 - ▼ External reference cache
 - ▼ Pivot table cache
- ▼ **Very performance sensitive, yet not directly unit-tested.**
- ▼ **Useful code only usable in LibreOffice. Shame.**

What to extract

Extract complex data structures

Multi-dimensional data structure (mdds)

<http://code.google.com/p/multidimensionalalgorithm/>

flat_segment_tree

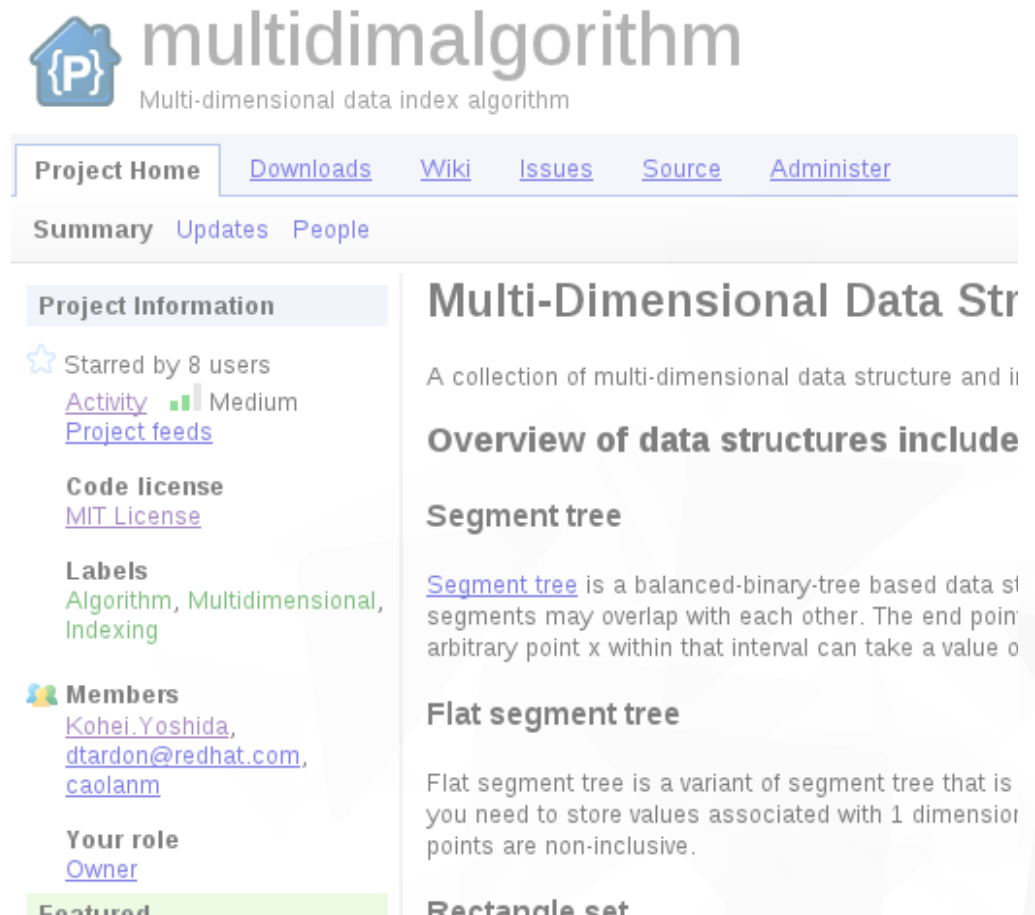
segment_tree

rectangle_set

point_quad_tree

mixed_type_matrix

more to come...



The screenshot shows the project page for 'multidimensionalalgorithm' on Code.google.com. The page title is 'Multi-dimensional data index algorithm'. The navigation menu includes 'Project Home', 'Downloads', 'Wiki', 'Issues', 'Source', and 'Administer'. Below the navigation menu, there are tabs for 'Summary', 'Updates', and 'People'. The main content area is divided into two columns. The left column contains 'Project Information' with details such as 'Starred by 8 users', 'Activity' (Medium), 'Project feeds', 'Code license' (MIT License), 'Labels' (Algorithm, Multidimensional, Indexing), 'Members' (Kohei.Yoshida, dtardon@redhat.com, caolanm), and 'Your role' (Owner). The right column contains the title 'Multi-Dimensional Data Str' and a description: 'A collection of multi-dimensional data structure and i'. Below the description, there is a section titled 'Overview of data structures include' which lists 'Segment tree' and 'Flat segment tree'. The 'Segment tree' section describes it as a balanced-binary-tree based data structure where segments may overlap. The 'Flat segment tree' section describes it as a variant of segment tree for storing values associated with 1-dimensional points.

Extract complex data structures (cont'd)

▼ Benefit

- ▼ Hides complex storage logic from Calc's code.
- ▼ No need to have LibreOffice build system. Ease of maintenance.
- ▼ Easier performance tuning and unit-testing.
- ▼ Usable outside LibreOffice.

▼ Cost

- ▼ Extra overhead when fixing bugs in mdds discovered from LibreOffice code. (Solution: comprehensive unit-testing.)
- ▼ Restricted push access. (Solution: added David and Caolan as co-maintainers.)

Extract complex data structures (cont'd)

▼ Currently used

- ▼ **flat_segment_tree** - Row visibility, row height, column width, and other misc places.
- ▼ **mixed_type_matrix** - Matrix class backend storage.

▼ Future plans

- ▼ Cell format storage to mdds. Maybe **segment_tree** will do?
- ▼ 2D grid structure for cell instance storage, pivot table cache as well as external reference cache.
- ▼ More candidate data structures lying around?

Extract formula parser & interpreter

▼ Ixion - threaded formula parser & interpreter

- ▼ <https://gitorious.org/ixion>
- ▼ Standalone C++ library usable outside LibreOffice. Easier maintenance.
- ▼ Not UNO component.
- ▼ Multi-threaded interpreter.
- ▼ Independent unit-testing framework.
- ▼ Formula parser sharing done right.
- ▼ Named after '28978 Ixion' the dwarf planet.
- ▼ My personal pet project.

Ixion - What it does

▼ Ixion provides

- ▼ Formula string lexer & tokenizer (parser).
- ▼ Formula token interpreter - multi-threaded.
- ▼ Cell dependency tracker.
- ▼ Class definitions for cells, address, tokens etc.
- ▼ Reference name resolver (A1-style).
- ▼ Interface to communicate with client code (read access to cell storage, write access to cell flags and calc results).

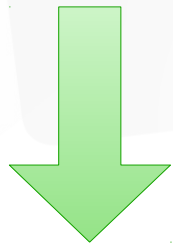
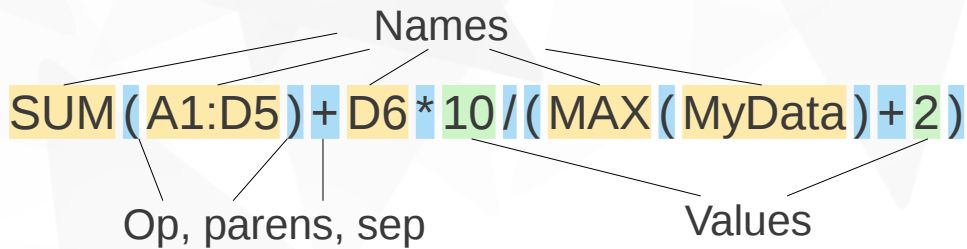
▼ Ixion will provide

- ▼ Hook for client-defined cell functions (macro functions etc).
- ▼ R1C1, ODF, ..., reference name resolvers.
- ▼ Hook for handling external references.

Ixion - How it works (formula string tokenization)

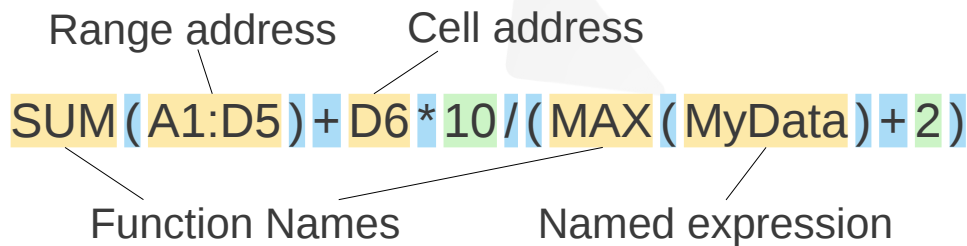
Ixion library

Break raw formula string into lexer tokens.



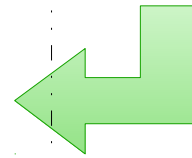
Pass the lexer tokens to the formula parser.

Convert lexer tokens into formula tokens.



Client code

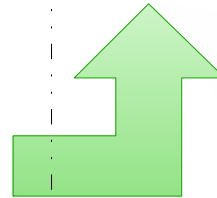
```
SUM(A1:D5)+D6*10/(MAX(MyData))+2)
```



Pass raw formula string to Ixion.

The client code creates a formula cell which stores the tokens.

Formula cell



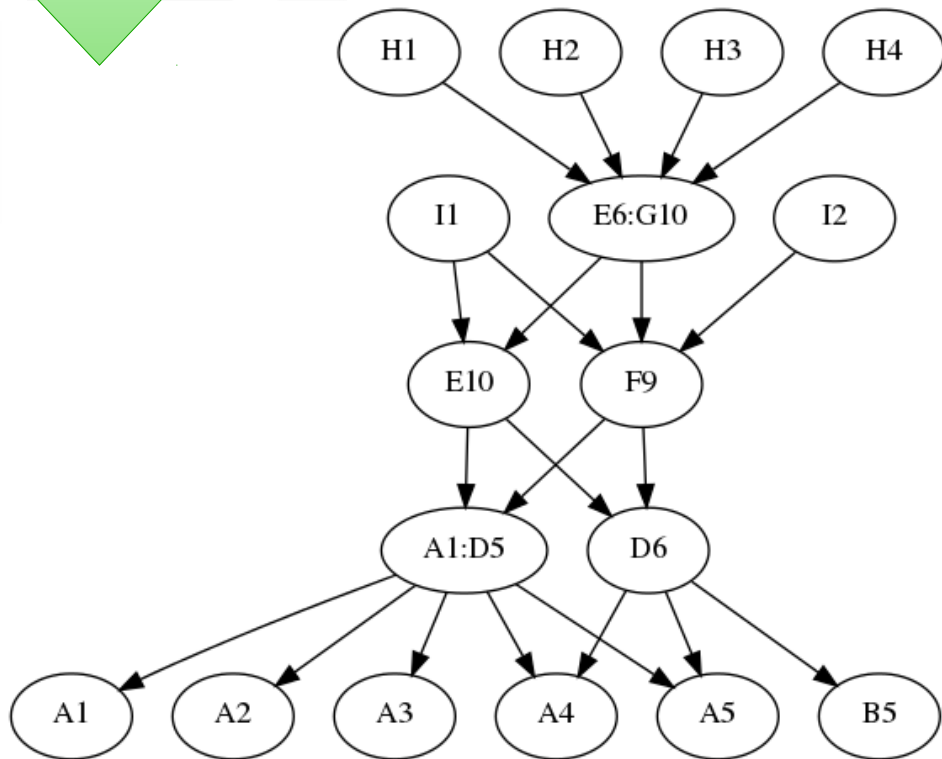
Pass the formula tokens back to the client code.

Ixion - How it works (Initial full calculation)

Ixion library

SUM(A1:D5) + D6 * 10 / (MAX(MyData) + 2)

Go through reference tokens in formula cells to build dependency graph.



Client code

Register all formula cells to Ixion.

Tell Ixion to calculate all formula cells.

Sort all formula cells by dependency.

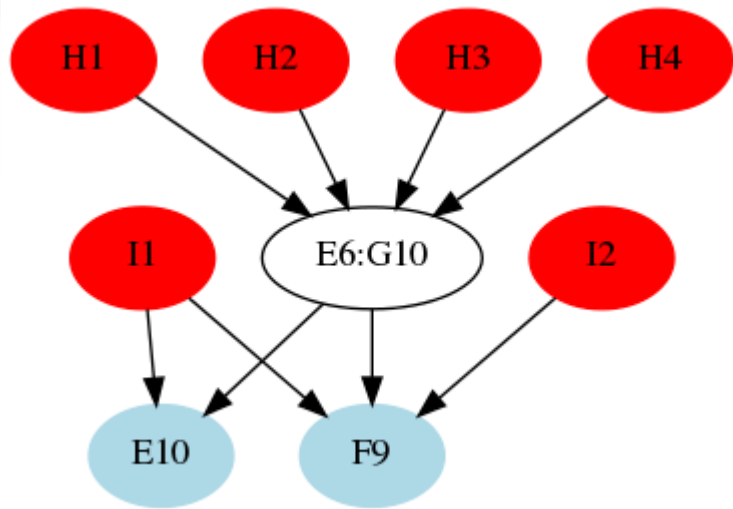
Calculate them in order using specified number of threads.

Ixion - How it works (Re-calculation)

Ixion library

SUM (A1:D5) + D6 * 10 / (MAX (MyData) + 2)

Go through references in modified cells and update dependency graph.



Sort dirty cells by dependency and calculate them in order.

E10 F9 I1 I2 H1 H2 H3 H4

Client code

Modified cells

F9 E10

Register modified cells to Ixion.

Query Ixion to get all affected (dirty) cells; cells that depends on modified cells directly or indirectly.

Ixion returns all dirty cells.

E10 H1 I1 H3
F9 H2 I2 H4

Pass all dirty cells to Ixion.

Ixion - Test framework

- ▼ **Pre-defined test cases**
 - ▼ Uses ixion-parser executable.
 - ▼ Define, calculate, and re-calculate cells and check their results.
 - ▼ Simulates run-time editing of spreadsheet document.
- ▼ **Unit test - More fine-grained tests of internal code.**
 - ▼ Reference name resolution.
 - ▼ String-to-double conversion.
 - ▼ Simple formula string tokenization.

DEMO

Ixion - Pre-integration strategy for Calc

Ixion requires	Change in Calc
String cells store IDs, not raw strings.	<ul style="list-style-type: none">* Application-wide shared strings.* Merging of ScStringCell and ScEditCell.
Formula cells store token IDs and flags.	<ul style="list-style-type: none">* Restructure ScFormulaCell, and its neighboring code.* Shared formula tokens.
Cells only store values.	<ul style="list-style-type: none">* Store other items (such as notes) outside the cells.
Use calc chain to sort cells and calculate them iteratively.	<ul style="list-style-type: none">* Defer until Ixion integration.

Actual integration must be done on a branch, and will probably take several minor release cycles. We must play it safe!

Any other stuff to extract?

Extract import filter framework.

- ▼ **Orcus - Spreadsheet document filter library.**
 - ▼ <https://gitorious.org/orcus>
 - ▼ Standalone C++ library. Not UNO component.
 - ▼ Two Layers
 - ▼ Base raw stream parsers (C++ templates) - XML, CSS, CSV.
 - ▼ Full import filters (binary) - ODS, XLSX, CSV.
 - ▼ Import filters only. Support for export filters planned.
 - ▼ Performance and maintainability.
 - ▼ Named after '90482 Orcus' the dwarf planet.
 - ▼ Personal pet project.

Orcus - Motivation

- ▼ **Unhappy with current ODS, XLSX filters.**
 - ▼ Terrible performance. Unbelievably slow.
 - ▼ Over-engineered design. Odd mixture of internal and UNO APIs.
 - ▼ Usable only in LibreOffice.
- ▼ **Simple filter design for simple format.**
 - ▼ Current CSV, HTML filters are unnecessarily complex. Hard to maintain.
 - ▼ Not optimized for performance.
 - ▼ Usable only in LibreOffice.

Orcus - What it does.

▼ Orcus provides

- ▼ Independent C++ spreadsheet filter framework.
- ▼ API designed for optimized loading & parsing performance.
- ▼ ODS, XLSX, CSV import filters.
- ▼ C++ template-based XML, CSV, CSS parsers that can be embedded in arbitrary code.

▼ Orcus will (may?) provide

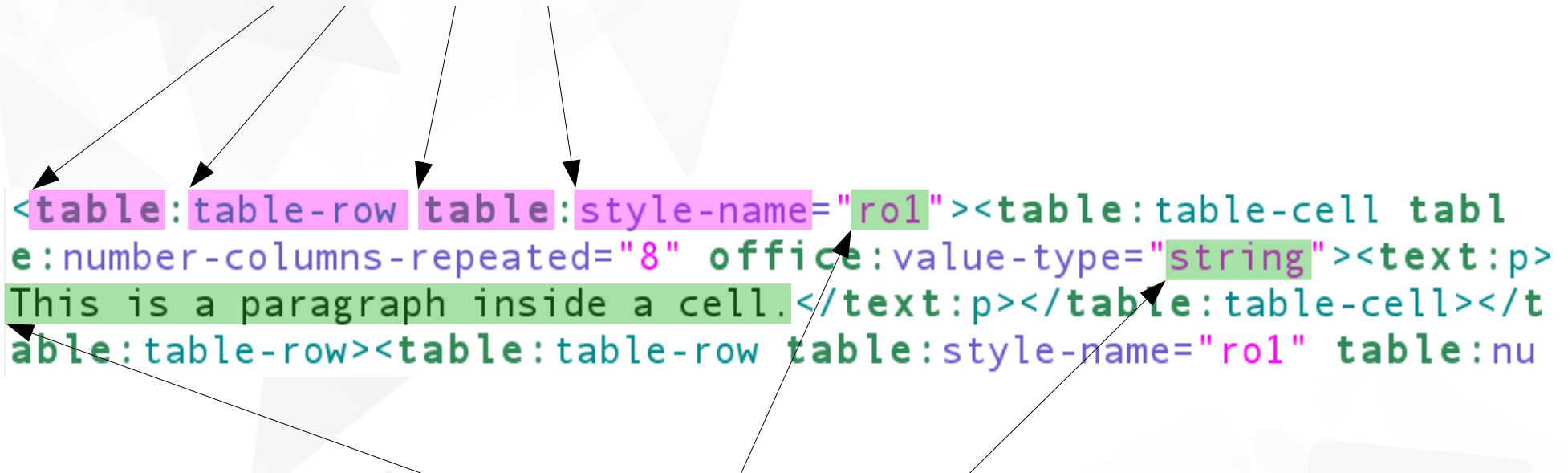
- ▼ More raw parsers - HTML
- ▼ More filters - Excel 2003 XML, HTML.
- ▼ Export filter framework.
- ▼ Any other parsers, filters as their needs come up.

Orcus - Performance bits

- ▼ **No temporary string allocations; re-use stream buffer.**
- ▼ **Tokenized XML parsing – avoid string comparisons.**
- ▼ **C++ template based parser – allow compiler optimization.**
- ▼ **API designed for performance.**
 - ▼ No temporary strings (pass pointer to first char and length).
 - ▼ Push contents to the model while parsing (to avoid intermediate storage).

Orcus - Re-use stream buffer (XML)

Tokenized to numeric IDs.



The diagram illustrates the tokenization of XML into numeric IDs. Arrows point from the text 'Tokenized to numeric IDs.' to various parts of the XML code. The XML code is as follows:

```
<table:table-row table:style-name="ro1"><table:table-cell tabl  
e:number-columns-repeated="8" office:value-type="string"><text:p>  
This is a paragraph inside a cell.</text:p></table:table-cell></t  
able:table-row><table:table-row table:style-name="ro1" table:nu
```

The tokens are highlighted in the image as follows:

- `<table:table-row` (magenta)
- `table:style-name="ro1"` (magenta)
- `>` (green)
- `<table:table-cell` (teal)
- `tabl` (teal)
- `e:number-columns-repeated="8"` (purple)
- `office:value-type="string"` (purple)
- `>` (green)
- `<text:p>` (teal)
- `This is a paragraph inside a cell.` (green)
- `</text:p>` (teal)
- `</table:table-cell>` (teal)
- `</t` (teal)
- `able:table-row>` (teal)
- `<table:table-row` (teal)
- `table:style-name="ro1"` (purple)
- `table:nu` (teal)

Only memory address and size are stored - no allocation.
Valid while the XML buffer is in memory.

Orcus – How to use it stand-alone.

```
$ orcus-ods path/to/document.ods
```

```
$ orcus-xlsx path/to/document.xlsx
```

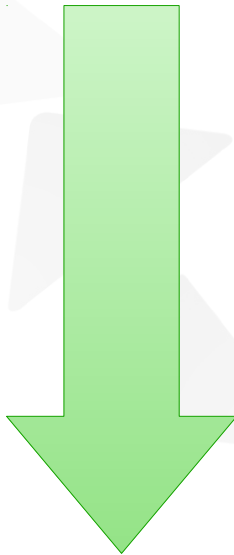
```
$ orcus-csv path/to/document.csv
```

DEMO

Orcus – How to use it as a library.

Orcus library

Open the document, unpack the package and start parsing.

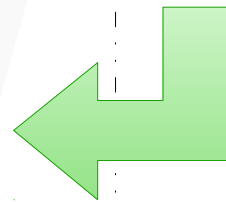


Finished parsing the document.

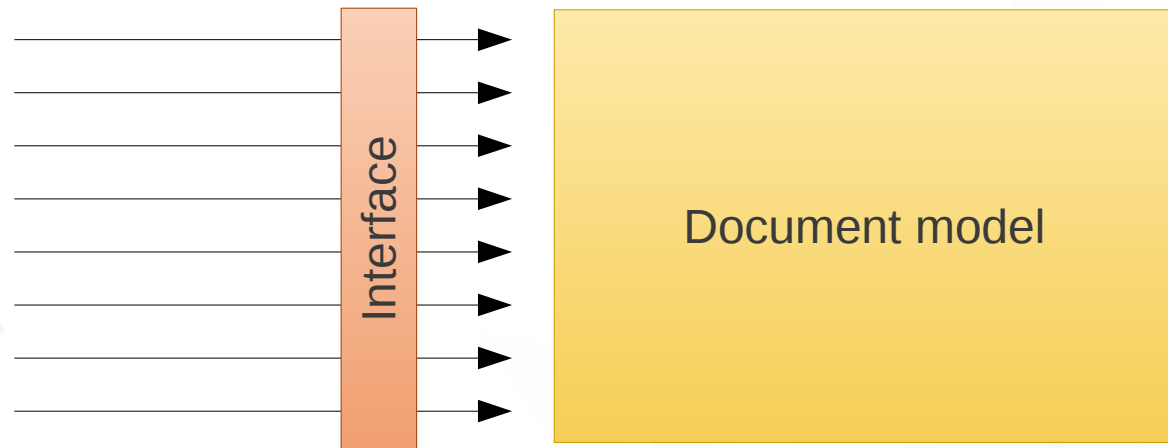
Client code

```
/path/to/financial-data.ods
```

Pass the absolute file path of the document to open.



Pass content fragments to the document model while parsing.



Orcus - preliminary results

▼ Document with unformatted text cells

- ▼ (ods) lots of text cells on 16 sheets (famous George Ou file)
 - ▼ 40 sec (current filter as of 3.4)
 - ▼ 12 sec (orcus-ods)
- ▼ (xlsx) 300,000 rows / 2 columns on 1 sheet
 - ▼ 1 min 40 sec (current filter as of 3.4)
 - ▼ 3.2 sec (orcus-xlsx)

▼ This is not final!

Orcus - Integration strategy

- ▼ **Changes required prior to integration. Largely overlaps with Ixion's integration requirements.**
 - ▼ Shared strings across Calc core.
 - ▼ Optionally bypass the normal file import path in framework.
 - ▼ Re-work cell storage and formula handling. See Ixion's integration strategy.
 - ▼ Implement the interfaces required by Orcus.
 - ▼ Perhaps lots of others I haven't thought of.
- ▼ **Likely go through several iterations.**
 - ▼ Do it in a feature branch.
 - ▼ Push changes to master in steps.
 - ▼ Coordinate with Ixion integration.

What to do short-term?

Putting it all together...

- ▼ **Re-work Calc's sheet storage and offload more code to mdds. Cell format storage and 2D cell storage.**
- ▼ **Use the 2D cell storage container in Ixion and Orcus to further improve performance of the container.**
- ▼ **Implement shared strings.**
 - ▼ rtl::OUString all the way in Calc core.
 - ▼ string cell and edit cell (rich text cell) merge.
- ▼ **Move stuff out of base cell (cell note etc).**
- ▼ **Implement shared formula tokens.**
- ▼ **... (lots-n-lots more stuff)**
- ▼ **Integrate Ixion.**
- ▼ **Integrate Orcus.**

If I still have time and energy....

What other code can be extracted?

- ▼ **Data slicer engine for pivot table.**
 - ▼ Immensely useful.
 - ▼ Performance sensitive.
 - ▼ Could use comprehensive unit-test framework.
- ▼ **Number formatter (number detection and formatting)**
 - ▼ Very complex.
 - ▼ Useful on its own.
 - ▼ Could use comprehensive unit-test framework.
- ▼ **Chart engine.**
 - ▼ Maybe useful, maybe not.
 - ▼ Abundance of data visualization software available.

Thanks for listening!



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](#).