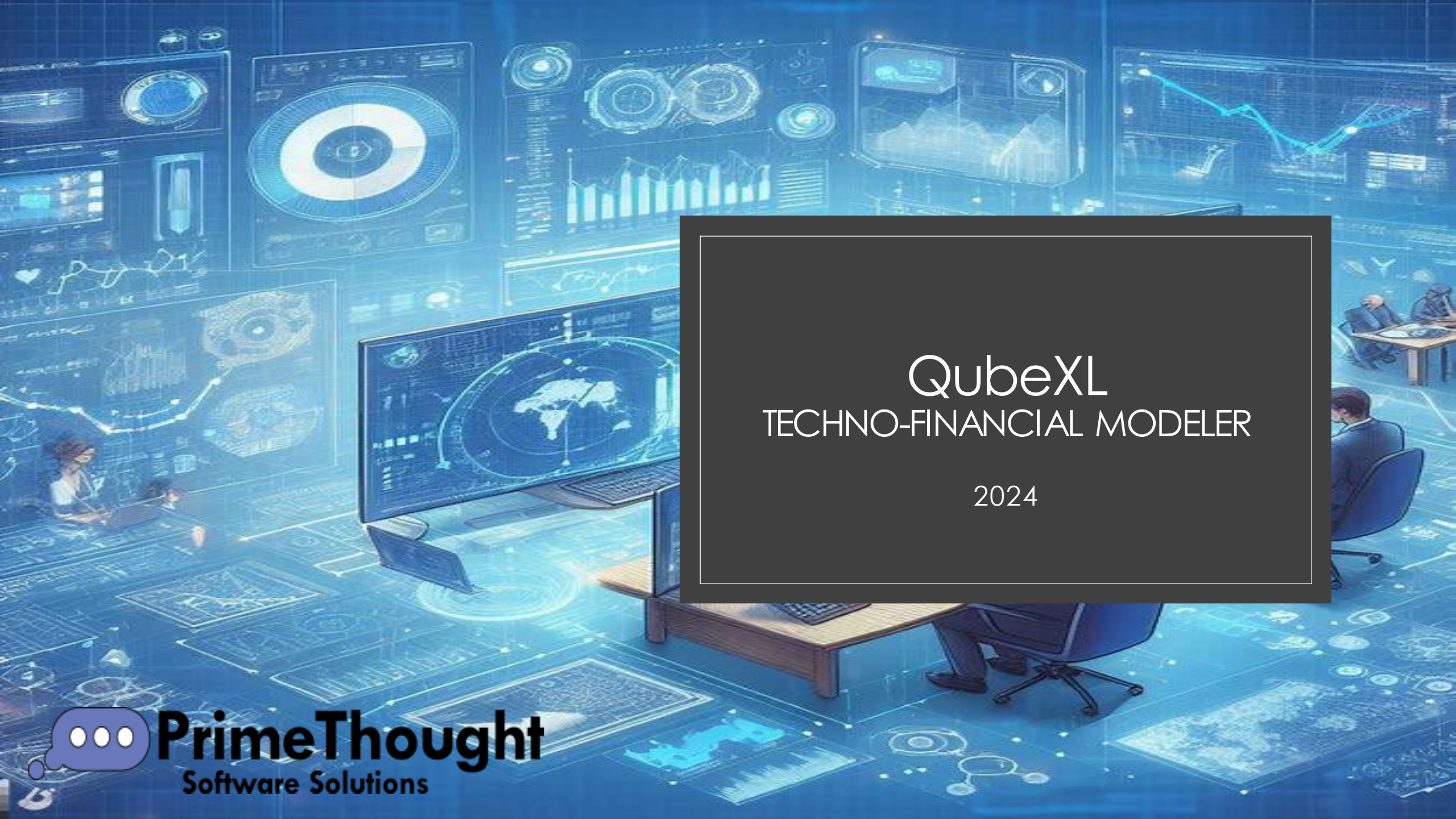


QubeXL

TECHNO-FINANCIAL MODELER

2024

PrimeThought
Software Solutions

Introduction to PrimeThought

- PrimeThought Software Solutions is a well-established South African company in operation for over a decade.
- We provide to end spatial analytic solutions

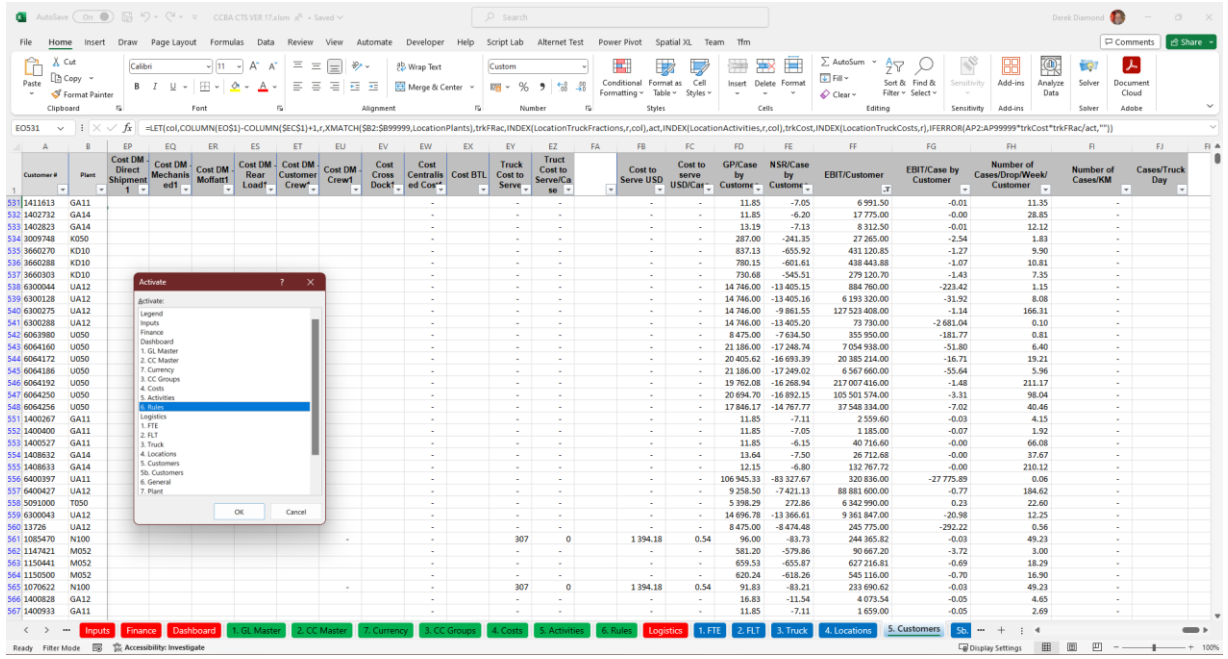
PrimeThought
Software Solutions

Our clients

PrimeThought Software Solutions

QubeXL what is it?

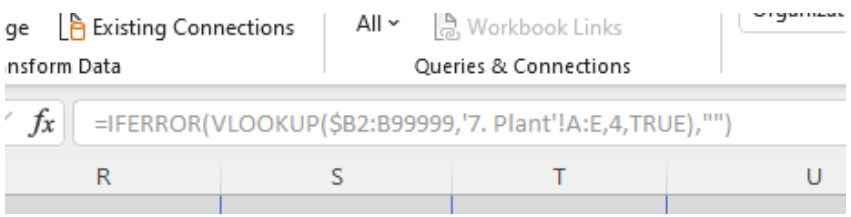
- QubeXL is a Techno-Financial modelling system.
- These models are traditionally created in Microsoft Excel.
- For small and simpler models this can be acceptable but when the models result in 100's of columns and many sheets they become unwieldy.
- This manifests in the following ways:



PrimeThought
Software Solutions

Formulae are not immediately understandable

- For example
- We can’t immediately see what is being referenced in the VLOOKUP



PrimeThought
Software Solutions

Complex formulae are almost undecipherable

```
=LET(col,COLUMN(BK$1)-COLUMN($BA$1)+1,r,XMATCH($B2:$B99999,LocationPlants),act,INDEX(LocationActivities,r,col),IFERROR(AN2:AN99999/act*INDEX(LocationDirectCosts,r,col),""))
```

- In the above this is despite using named ranges and the new LET function

PrimeThought
Software Solutions

Large workbooks can take minutes to calculate

- This is usually caused by the very inefficient VLOOKUP function and the fact that many intermediate cells need to be used to handle complicated formulae

Errors can occur in formulae that will not be detected

- All cells need to be inspected which can take ages to verify with and instances can easily be missed

PrimeThought
Software Solutions

Invalid data can be entered and not detected

- For example entering a number as text.

Lookups on more than one column

- Require additional columns
- Notoriously slow
- Crux of any complex model!

Auditing these models is very complex

- You can get dependents and precedents of a cell, but only to one level, which makes it very time consuming and error prone to trace the calculation of a cell

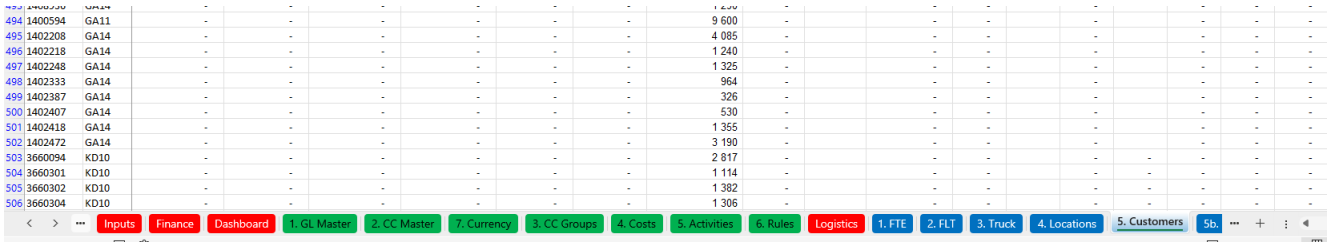
PrimeThought
Software Solutions

Complex logic is very hard to encode

- Even harder to verify and test

Can’t organize many worksheets

- Only have one large flat list of worksheets



PrimeThought
Software Solutions

Getting data into the model

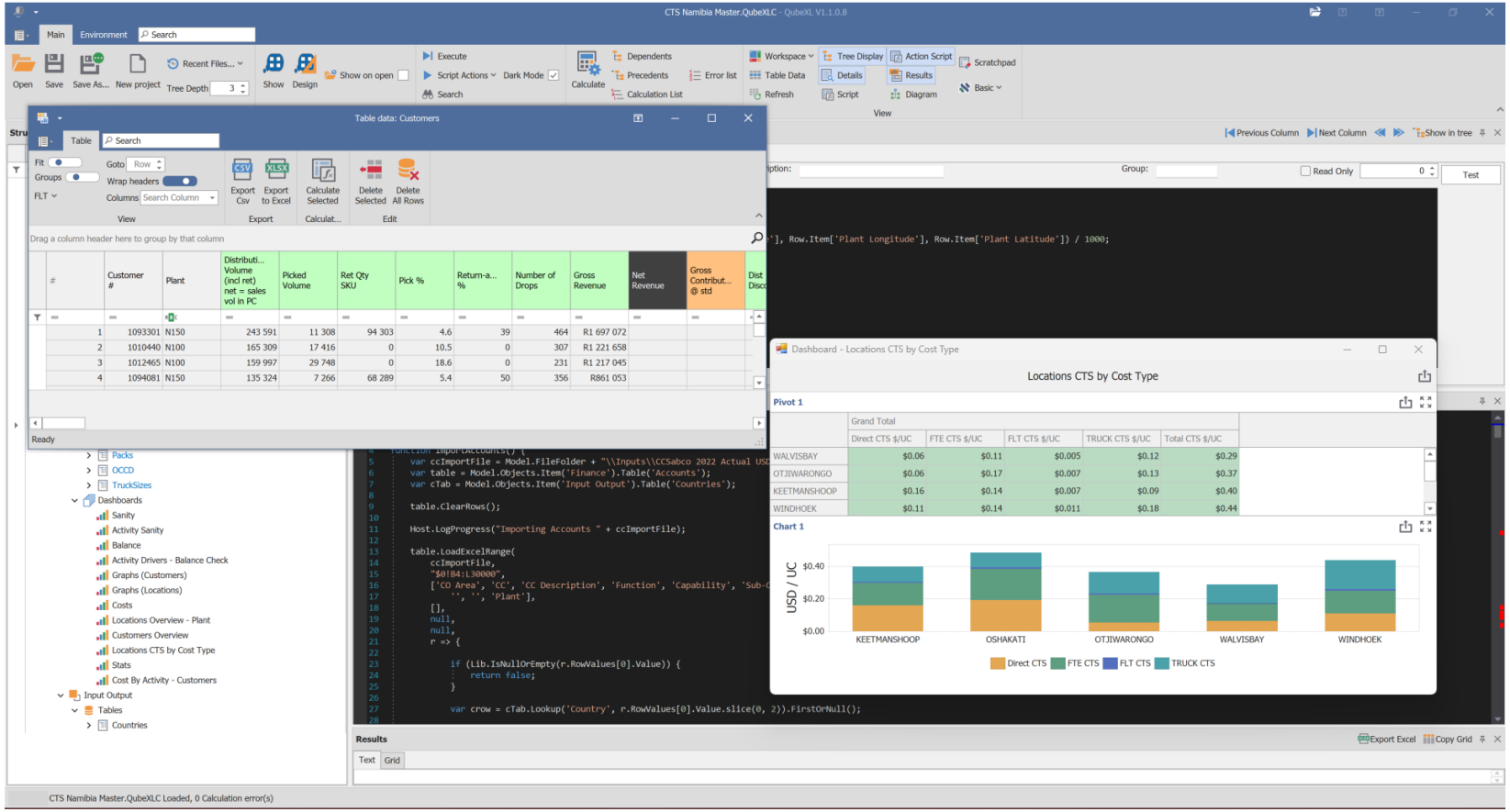
- Excel has no simple customizable way to get your data into the model
- Very manual resulting in errors

PrimeThought
Software Solutions

Getting data out of the model

- No standard way to get the data out for advanced reporting
- Manual and error prone

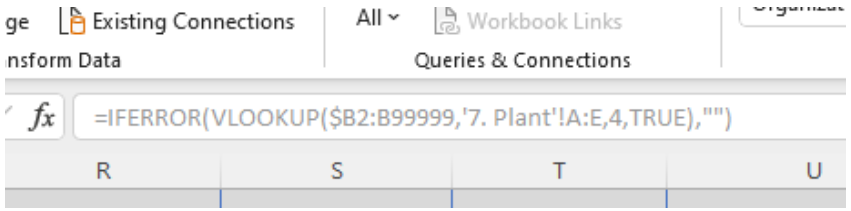
QubeXL is designed to address these issues

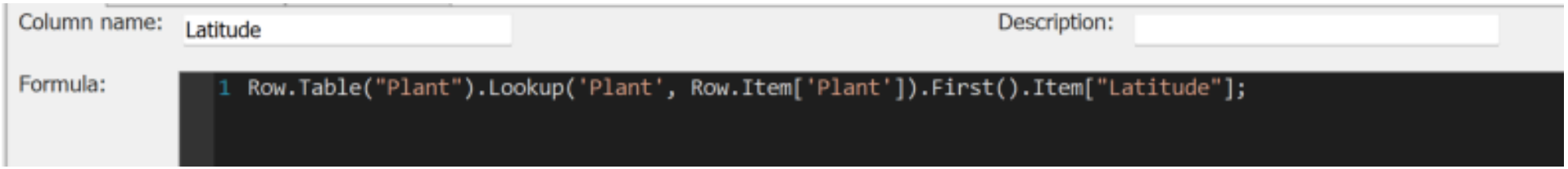


PrimeThought
Software Solutions

Formulae are immediately understandable

- For example
- In QubeXL we can immediately see we are looking up the Latitude column from the Plants table





PrimeThought
Software Solutions

Complex formulae are more understandable

```
=LET(col,COLUMN(BK$1)-COLUMN($BA$1)+1,r,XMATCH($B2:$B99999,LocationPlants),act,INDEX(LocationActivities,r,col),IFERROR(AN2:AN99999/act*INDEX(LocationDirectCosts,r,col),""))
```

```
var r = Row.Tables['Locations'].Lookup('Plant', Row.Item['Plant']).First();

Lib.Divide(Row.Item['Btl'], r.Item['Plant Btl1']) * r.Item['Direct Costs Btl1'];
```

- Here we lookup the plant of the customer in line 1 and do a divide in line two of the rows value by the plants value and multiply it by an amount in the row

PrimeThought
Software Solutions

Large models take up little space

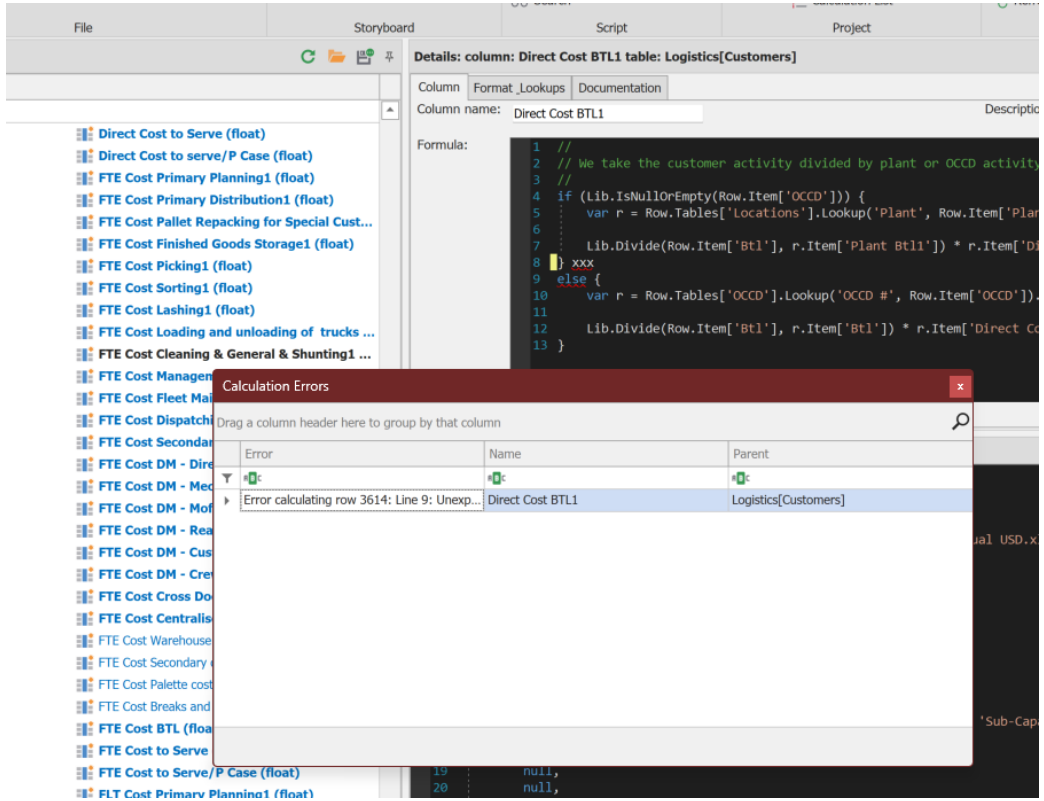
Name	Status	Date modified	Type	Size
Inputs		2023/09/11 09:30	File folder	
Templates		2023/09/11 09:30	File folder	
CCBA CTS VER 17.xlsm		2022/02/22 13:48	Microsoft Excel M...	61 715 KB
CCBA CTS VER 17.QubeXLC		2023/09/08 11:30	QubeXL Project (C...	1 449 KB

Files are 1 /20 the size of the Excel workbook

PrimeThought
Software Solutions

Errors prevent the model from calculating

- Any error in the model stops calculation



PrimeThought
Software Solutions

Data is typed per column of tables

Details: table: Logistics[Customers]

Columns | Script | Documentation

Name: Customers Read Only

Drag a column header here to group by that column

Column Name	Type	Group	Format
Customer #	**int**		
Plant	**string**		
Distribution Volume (incl ret) net = sales v...	float		N0
Picked Volume	float		N0
Ret Qty SKU	float		N0
Pick %	float		N1
Return-able %	float		N0
Number of Drops	float		N0
Gross Revenue	float		C0
Net Revenue	float		
Gross Contribution @ std	float		
Dist Discount	float		
Other Discounts	float		C2
Mode - NOW	string		
Net Revenue USD	float		C2
Gross Profit USD	float		C2
Sales Volume UC	float		N0
OCCD	string		
Scenario Pick%	float		N2
Distance from Depot (km)	float		N1
Nominal Distance Travelled	float		

PrimeThought
Software Solutions

Lookups on more than one column

Details: column: Plant table: Finance[Costs]

Column | Format _Lookups | Documentation

Column name: Plant

Description:

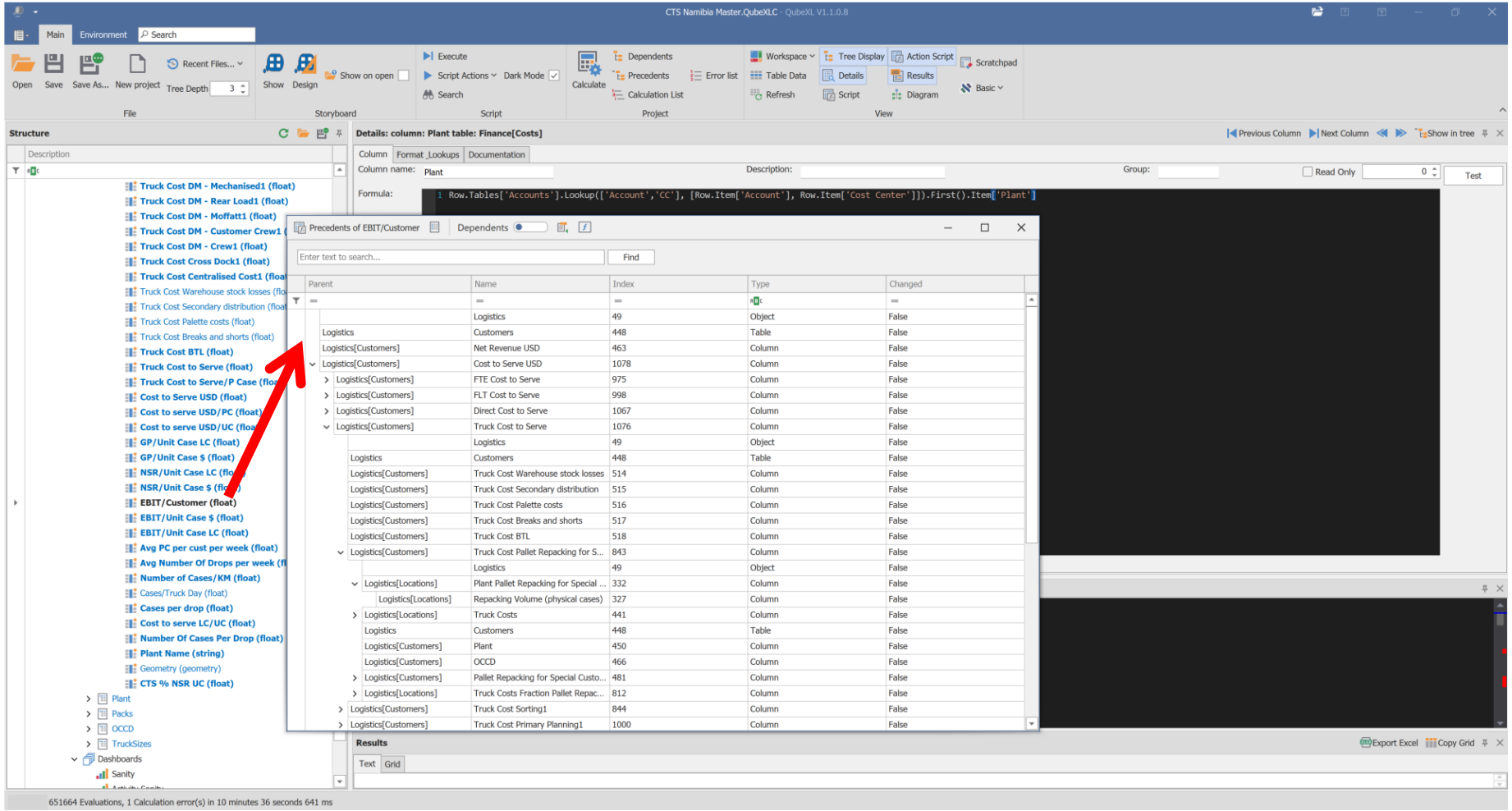
Formula:

```
1 Row.Tables['Accounts'].Lookup(['Account','CC'], [Row.Item['Account'], Row.Item['Cost Center']]).First().Item['Plant']
```

From table Accounts, lookup using columns Account and CC and values from column Account and Cost Center in this table, find first matching row and get Plant name

PrimeThought
Software Solutions

Auditing these models is easy



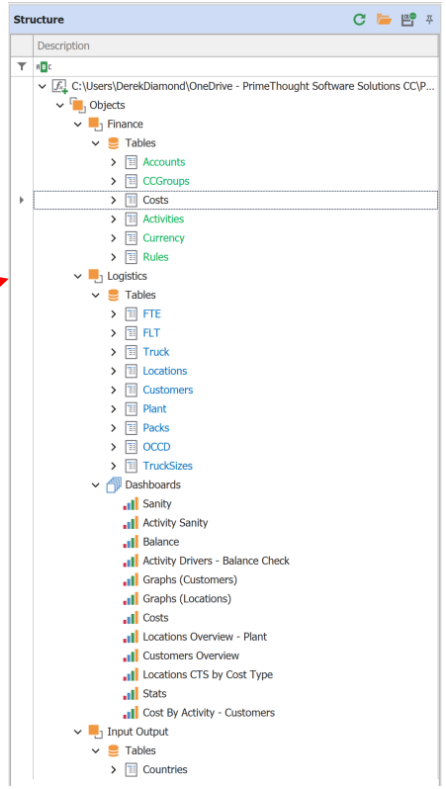
PrimeThought
Software Solutions

Complex logic can be encoded

```
//
// We take the customer activity divided by plant or OCCD activity times the plant or OCCD costs times the fraction
//
if (Lib.IsNullOrEmpty(Row.Item['OCCD'])) {
    var r = Row.Tables['Locations'].Lookup('Plant', Row.Item['Plant']).First();

    Lib.Divide(Row.Item['Picking'], r.Item['Plant Picking']) * r.Item['Truck Costs'] * r.Item['Truck Costs Fraction Picking'];
}
else {
    var r = Row.Tables['OCCD'].Lookup('OCCD #', Row.Item['OCCD']).First();

    Lib.Divide(Row.Item['Picking'], r.Item['Picking']) * r.Item['Truck Costs'] * r.Item['Truck Costs Fraction Picking'];
}

```

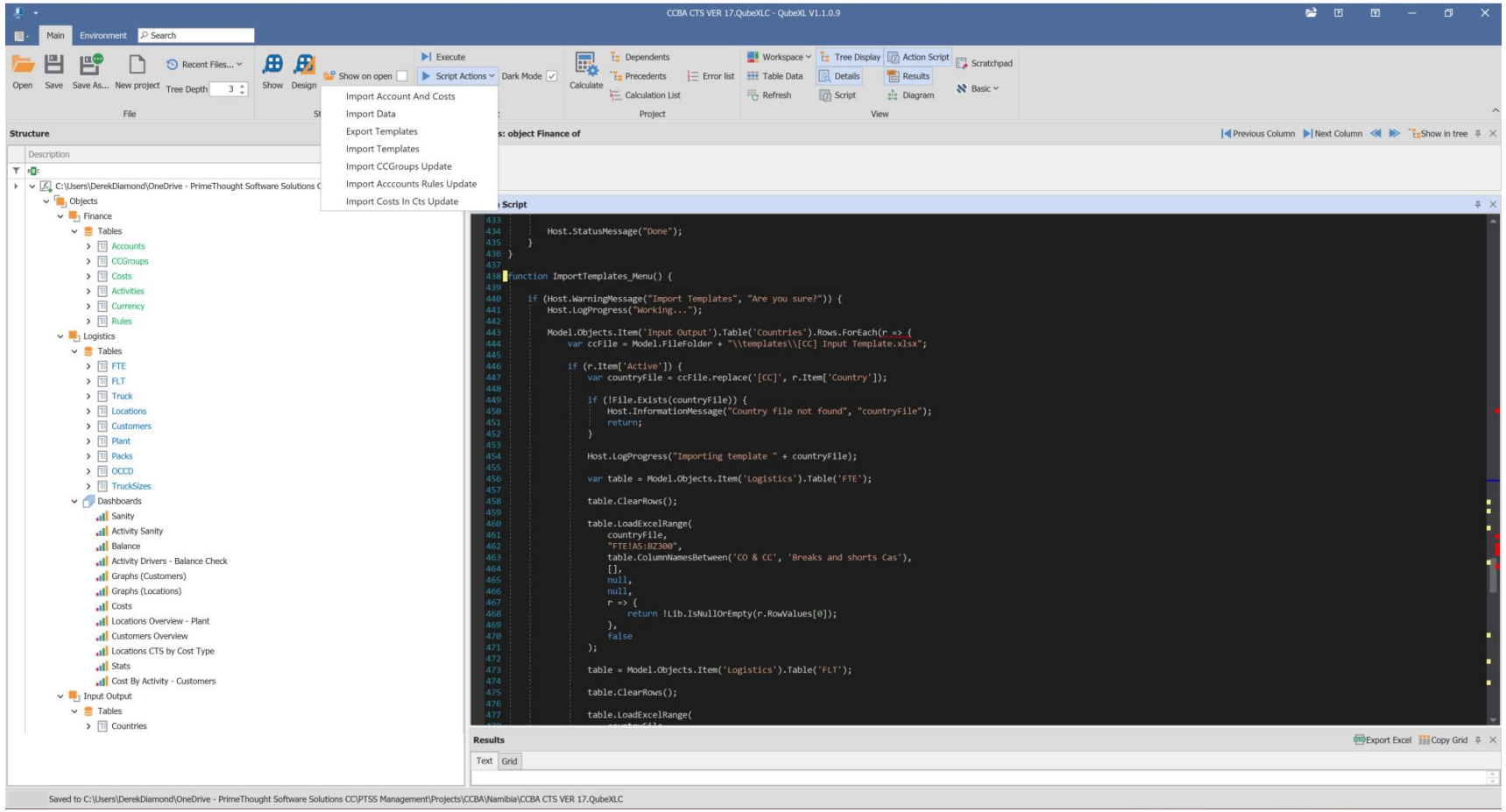
PrimeThought
Software Solutions

Tables, Dashboards, Formulae and AI Models are organized in a structure



PrimeThought
Software Solutions

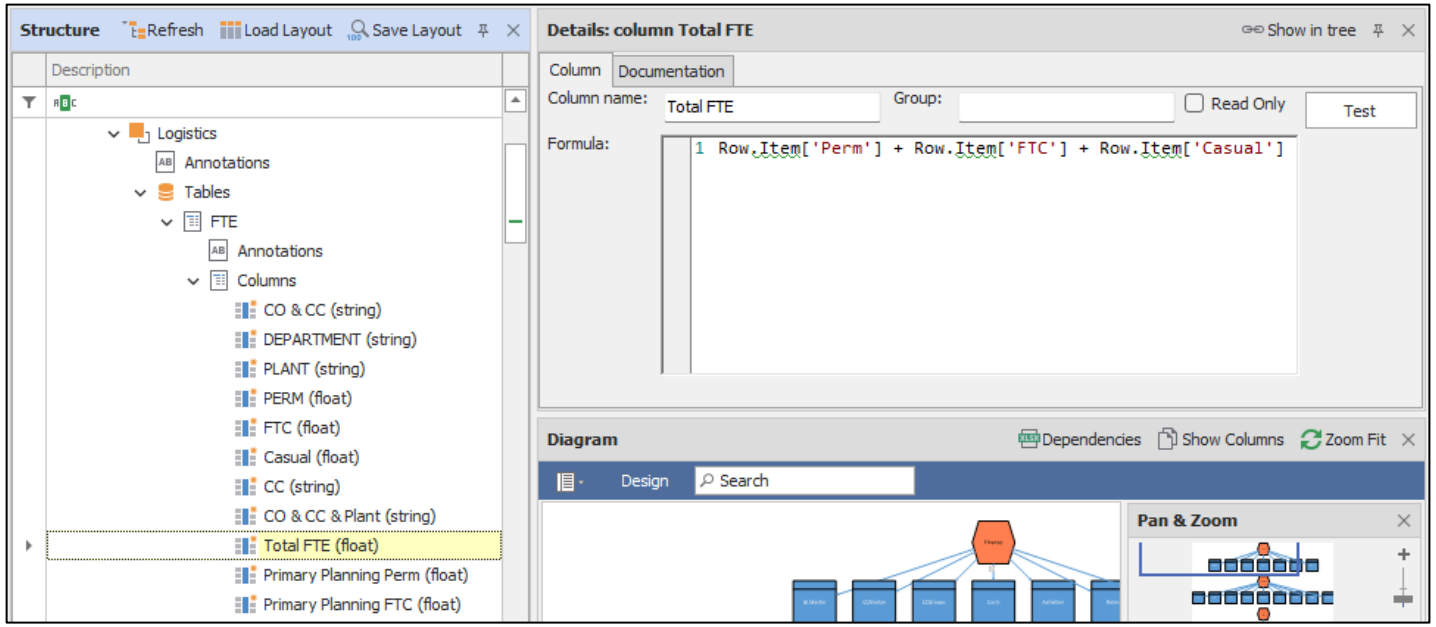
Getting data into the model fully customizable and repeatable



PrimeThought
Software Solutions

Simple and Auditable Models

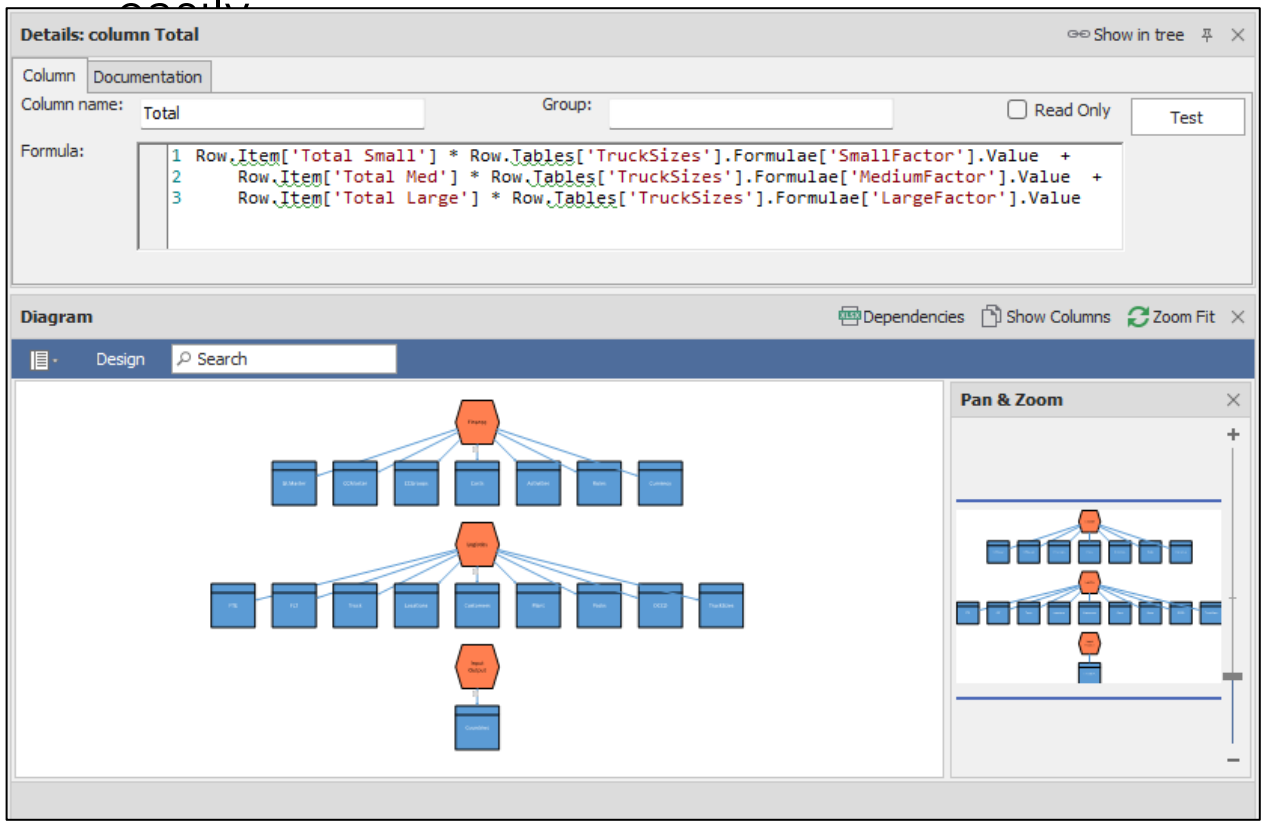
QubeXL simplifies things by having formulae and calculations per column, table and object instead of per cell as in Excel. This allows for simpler and auditable models which are easily verified and extended.

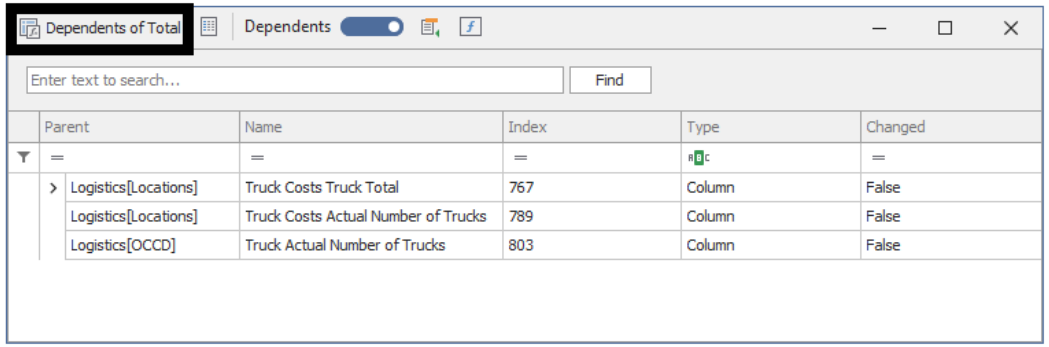


PrimeThought
Software Solutions

Simple and Powerful Scripting Language

QubeXL uses the most popular scripting language in the world: JavaScript for all its scripting and calculations. As you build your model QubeXL generates a diagram showing your model structure. You are also able to trace calculation dependencies easily.

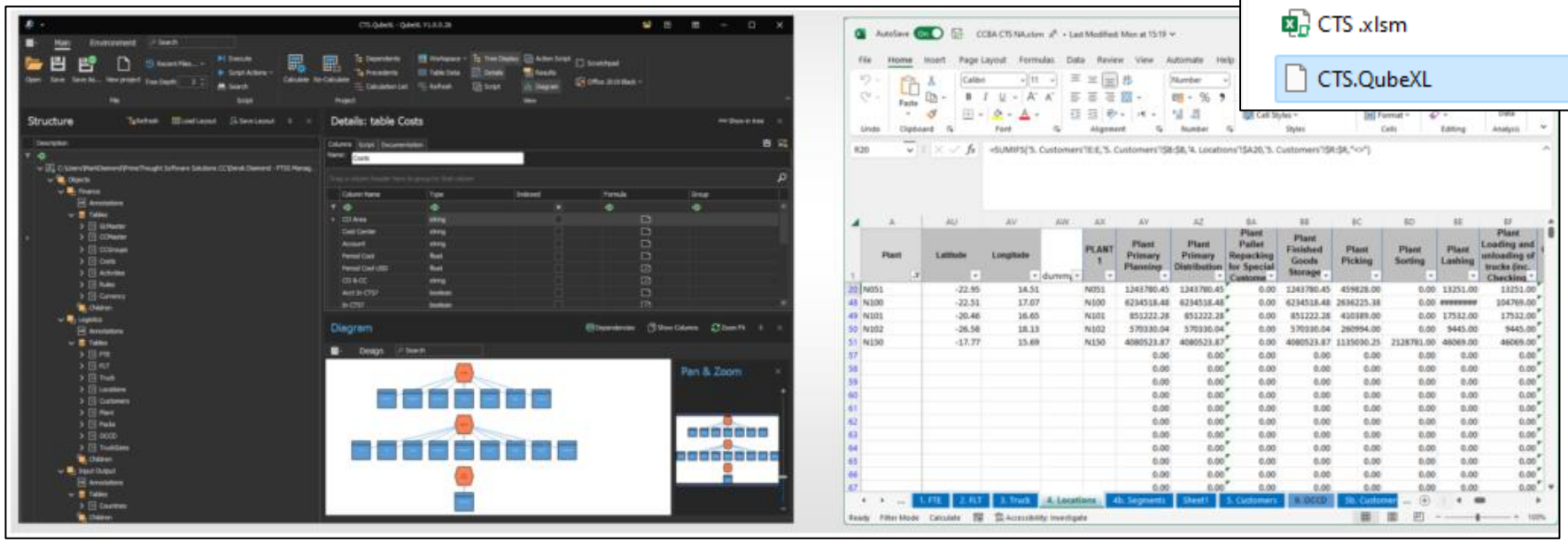




PrimeThought
Software Solutions

Powerful and fast

- Because QubeXL uses data frames borrowed from data science systems it can handle millions of rows per table and calculate these in times incomparable with traditional Excel models.
- Further it is much more compact. As an example, the exact same financial model in Excel is 73MB while in QubeXL it is only 9MB:

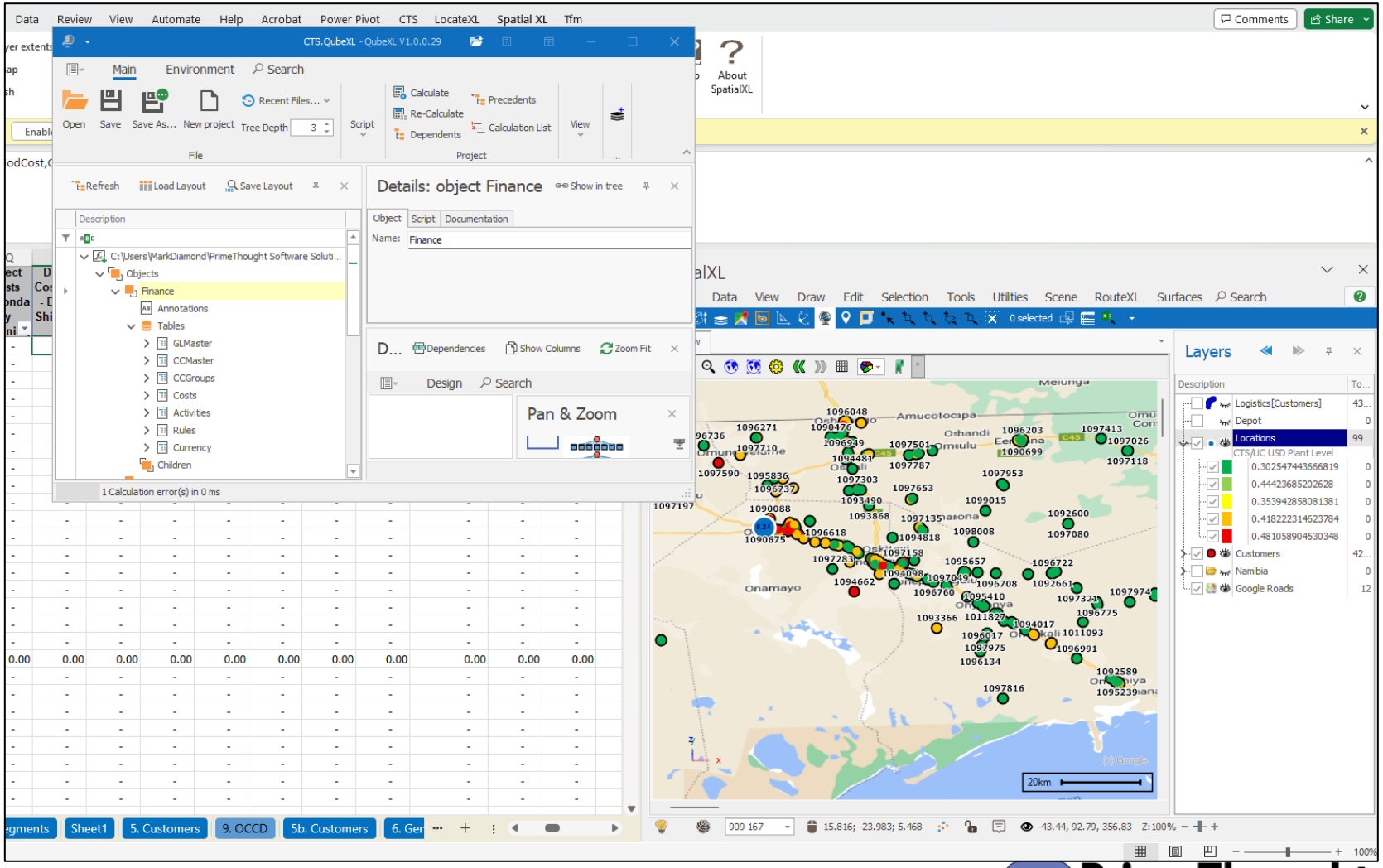


Name	Size
CTS .xlsm	73 522 KB
CTS.QubeXL	9 756 KB

PrimeThought
Software Solutions

Excel Add-In

QubeXL is able to plug in to SpatialXL which is our mapping and spatial analytics tool that integrates with Excel. Your model data can then be plotted on a map and themed etc. with full business intelligence and GIS capability offered by SpatialXL:



PrimeThought
Software Solutions

QubeXL key features

http://qubexl.com/ & https://primethought.biz/

1. Object Oriented techno-financial modeller
2. Handles millions of rows of data
3. Scriptable imports and ETL from databases and Excel
4. Self describing formulae
5. Multi views on tables
6. Scriptable model generation and extension
7. Built AI to build prediction models
8. Built in dashboarding
9. Built-in auditability and traceability
10. Built-in Storyboarding ™
11. Integration with our mapping, routing and spatial products
12. Excel Add-in

PrimeThought
Software Solutions

THANK YOU

Visit primethought.biz for more information

or email

sales@primethought.biz