

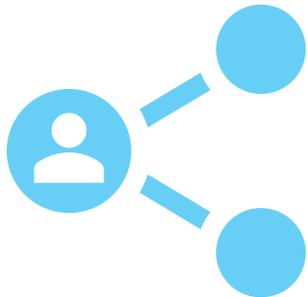


# Financial Modelling

The guide for essential professional skills



# PAYING IT FORWARD



We encourage you to share with your colleagues.

Please keep the document whole to preserve the structure, intent and spirit.

While there is no single right way in Financial Modelling, we have refined our method over 30 years to deliver confidence: championing Clarity, User Experience and Integrity – prioritising certainty and minimising errors.

You can learn the core Excel skills needed for financial modelling in a single day; or you could ask AI to explain a formula; however, robust financial model design and effective structuring skills evolve over hundreds of transactions.

We dedicate our time to a select number of client transactions, in-house training and our own project investments – so there is only so much we solve(!); however, by sharing this insight into how we work we aim to give you a boost - or a Red Bull F1 pit-stop experience if you're already racing.

Wherever you are on your journey, we're here to help you work smarter, not harder - spending less time tinkering with spreadsheets, more time confidently closing deals and powering ahead.

Enjoy our thoughts and approach, we hope it helps. If you like what we do check out [vectorHQ.co](https://vectorHQ.co)

*Nick* + *Ben*

# INTENTION

This Guide provides insight into how Vector delivers transaction-ready, bankable financial models - giving you an introduction to the foundations of how we develop elegant solutions for transactions.

Whether you develop or manage workstreams requiring robust financial models, or are just starting out, there is something here for you or your colleagues.

Our clients', and your, transactions demand analytical excellence; deceptively, this is achieved through the consistent application of seemingly small actions, with occasional innovation to solve a new situation – we lean on the same mindset and working style every time.

This Guide is not a "how-to-develop", as every model is unique; instead, here we convey the foundational principles of "how-we-work" which, when applied consistently, deliver outstanding outcomes for our clients.

If this guide resonates with you ask us about training, if you are facing a transaction and need it go to smoothly ask us about how we can support.



**MODEL  
DEVELOPMENT**



**TRAINING**

## MINDSET

Confidence

Clarity

UX

Integrity

## BEST PRACTICE ESSENTIALS

Laws

Rules

## WORKING EFFICIENTLY

Excel environment

Essential commands

Working faster

Safe is fast

Single source of truth

Output Led Design

Don't repeat yourself

Design principles

## CLARITY

Inputs

Formula

Workbook

Worksheet

Control accounts

Range names

## USER EXPERIENCE

Principle

Styles

Business logic

Model Flow

Process

## INTEGRITY

Common errors

Summing

Anchoring

Column variance

Trapping errors

Own goals

## INSPIRATION

Information display

Debt modelling

Our tool

Standard features

# MINDSET

Developing robust analytical frameworks requires a deliberate approach to deliver confidence and to ensure adaptability as the decision-making landscape evolves.

In this section we outline our priorities and focus to deliver an exceptional outcome for stakeholders whilst optimising delivery efficiency.





# THE VECTOR MINDSET

Financial modelling is decision architecture combined with consistent, detail-oriented execution. Success hinges on a deep understanding of the brief, business logic, and the needs of multiple stakeholders. We treat each engagement as if it's our own investment—this sharpens focus beyond just "pulling a model together."

90% of our calculations rely on a handful of core functions which along with our recommendation on setting up the Excel environment enables us, and you, to focus on the mechanics of the model rather than Excel.

Our primary goals are to:

- Deliver stakeholder confidence in a model— providing clarity, integrity and a high level of User Experience.
- Gather and incorporate business logic in a way that can adapt with the deal.

The above is even more critical when incorporating advanced scenario analysis, non-linear optimisation, and advanced simulation which necessitate a high-performance model.

Ultimately professional financial modelling comes down to:

- Delivering accurate, reliable, user-friendly results.
- Adapting to new information and situations.
- The model being easily used, understood and verified by multiple parties.
- Meeting stakeholder requirements on time.
- All of this under varying phases of intensity and urgency.

Challenging? Fun? Absolutely. Let's explore.



# IT'S NOT ABOUT ~~THE BIKE~~ EXCEL

Thirty years ago, Nick, then a post graduate Astrophysicist in London, was hired to explore “how project finance deals could be improved using computers”, he didn’t invent Financial Modelling, but he was certainly in the delivery room. After two decades at the forefront of the field, Nick teamed up with Ben, a software architect and expert in major project delivery. Together, after 15 years of friendship and a decade modelling together, they support select transactions and proudly train the best teams in the business and those aspiring to be so.

After three decades in the deal trenches, we still love delivering elegant solutions to complex, multi-stakeholder, fluid situations and mentoring the next generation.

Financial modelling isn’t something you can master passively by reading or following online tutorials. Unlike financial mathematics or contract law, it’s less fact-based and more akin to crafting furniture or motorcycle riding—other passions of ours. It requires practice, guided by someone with experience willing to share deep knowledge - the ‘Why’ not just ‘How’.

The essentials are laid out here, but true expertise comes with experience and hands-on collaboration. You can only develop a great model if you understand the deal and you can optimise the deal if you understand the model.

# CONFIDENCE

Confidence has two perspectives:

1. As an analyst developing a model you need to be confident that your results are as accurate as practically possible, especially when a structural change has been made, under time pressure and an evolving decision landscape.
2. As a decision maker you can rely upon a framework that is fit for purpose and be confident in the decision you will make based upon the analysis.

Confidence, our mantra, is comprised of three fundamental elements, which need to be orchestrated.

$$\text{Confidence} = \text{Clarity} * \text{UX} * \text{Integrity}$$

**Clarity** is the ability to understand and interrogate a model both at a macro (workbook) and micro level (data and calculations).

**UX (User Experience)** is the minimisation of cognitive load. Optimising the intent of data through consistency and considered presentation of information.

**Integrity** is the minimisation of errors and when they do occur, they are trapped and the user, especially the developer, is aware of them.

# CLARITY

This is without a doubt the best place to lay the foundations for the other components of Confidence. Easy wins are achieved through:

- Purposeful presentation of all cells so their intent is obvious (inputs, links, closing balances, scenario cells...)
- Clear ordering, grouping and concise naming of worksheets within the workbook
- Clean, readable formula syntax—no complex single-line “monsters”
- Precise descriptions of line items with all units clearly stated, without exception
- Quantification of any “check” into a Pass/Fail test panel
- Clearly laid out and controlled Scenario and sensitivity management
- Clear instructions for macros — when to execute them them and what are they doing when they are running.

A decision maker's initial impression significantly influences their confidence. User Experience (UX) ensures the information needed is readily available in the expected location with other supporting information clearly shown.

Focusing on UX enables a user to focus on analysis rather than searching for how something has been calculated – risking confusion and eroding confidence.

When an executive user uses a model, they expect to find the following easy:

1. What assumptions have been applied.
2. Where is the analysis, or part thereof, relevant to them.
3. Does it make sense and pass their expectations – executive users are very good at applying a “sniff-test”.
4. Make an input change, usually a Scenario or a single change such as a different set of prices.

From a practical perspective a model with high UX enables the above:

- The expected worksheet will be clearly named and accessible.
- Within the target worksheet the location of the calculation will be obviously titled (without scrolling too much).
- The calculation will be performed on localised cells with clear descriptions, units and sense checks.
- Consistent and purposeful formatting (basically use colour to reduce ‘thinking’).



UX

# UX COGNITIVE LOAD

Cognitive Load refers to the mental effort required to process information and comes in two types:

- Extraneous - how information is presented.
- Intrinsic - the complexity of the information being learned.

Cognitive load theory recognises that your working memory is limited. Information is best delivered in small, manageable amounts. This is crucial in transaction work, where speed and accuracy under pressure directly correlate to cognitive load. This is managed by:

- Providing small amounts of new information.
- Breaking up material into manageable chunks.
- Presenting information in consistent locations and formats.
- Using the optimal mix of graphical and tabular information.
- Providing guidance for complex tasks such as debt optimisation.

We ensure a lower Cognitive Load by:

1. Prioritising clarity of information with purposeful and consistent formatting. Colour is your friend in a visual, complex, environment.
2. Focusing on layout of the workbook, within worksheets and information blocks.
3. Using clear descriptions and helpful notes.

# INTEGRITY

Confidence demands integrity which has two elements:

- The minimisation of errors in the first instance.
- Making issues apparent when they exist, as soon as they exist.

A critical component of a professional modelling mindset is the constant awareness that errors are incredibly easy to make and often hide in plain sight. If this wasn't the case the global Model Audit businesses wouldn't exist. You can almost gamify this - the errors are in there how many can you find?

To maximise integrity, it is critical to:

- Work methodically following Best Practice (see our Laws and Rules).
- Create as many tests as you practically can and make sure the results of these tests are clearly presented to the user, which includes you as the developer. Whether or not a Balance Sheet balances would only be one of 20+ possible 'dynamic' tests. When creating tests, split them into categories:
  - Structural / Non-negotiable – these should always “Pass”.
  - Commercial – good things to know but it doesn't mean the model is “wrong”.
  - Debt – when optimising debt, we split these out, it's very helpful – just because debt hasn't been solved it doesn't mean it's in error - just that you will need to solve this at a later time.

Don't bury the tests! Put them in a panel and send the results through to top of most worksheets; doing this ensures that when an issue occurs it is obvious.

# MODEL DESIGN

Model design underpins the success of any financial model and is one of the interesting aspects of financial modelling. Typically, in any one sector, let's say a stand-alone single asset project financing or M&A or LBO transaction the design is a well-trodden, consistently applied path.

When a transaction involves a portfolio, this significantly changes the structure, especially when each asset or stage is financed separately. In situations involving cohorts such as meter roll-outs, advanced mine plans, SaaS, fleets of transport assets - then the approach changes again. The common theme is identifying and managing the number of dimensions of which Excel is highly limited, so the design needs to be smarter again. In situations when all the above needs to be overlaid with solving routines, simulation or updated with actuals, design considerations become even more important to deliver confidence.

A consideration in designing a model from both a user and developer's perspective, is balancing a model's computational efficiency with:

- Sufficient time resolution to capture the commercial requirements
- The number of unique formula - more so than how far they extend over time.
- The interrelation between calculations on different sheets

As complexity increases, in the case of portfolios in particular, structural consistency, or uniformity, is essential. If you have 2 or 100+ assets and one or more of them has a special consideration, then create all assets with the same structure and manage the underlying difference via inputs. This means you are only managing one master calculation which can then be replicated and automated.

Structural consistency, or uniformity, is a non-negotiable to achieve maintainability and error minimisation.

# THE BASICS

The intent of this section is to show you what we consider our essential ingredients for all tasks.

Following this recipe will help mitigate many non-commercial errors easily made when developing a model.

"If you're in a hurry, go the long way around." - Embrace the benefits of taking time to plan and make the proper steps from the outset, even if it seems slower because shortcuts lead to mistakes, wasted effort and errors.



Once you understand what you are analysing and how you will structure the model, the rest is a handful of small things performed consistently. Commercial logic tends to evolve throughout an assignment and adapting the model to accommodate these updates is also made much easier by following the below.

Everything is a balancing act but follow these and you'll be well on your way. Over 30 years we've distilled the non-negotiables to a small set of Laws, which are also highlighted in later sections.

- Single Source of Truth – don't calculate anything, ever, more than once.
- One formula, one row - beyond the freeze pane.
- Flags and counters significantly improve the clarity of subsequent formulae.
- Clearly describe each line item, everywhere and show the unit.
- Do not include numerical values, other than 1 or 0, in any formula – easier than it sounds. For example: do not type in 24 or 365 and never 1,000,000 – the value obviously won't change but using Range Names means word can be used in formula making them much easier to read and avoids errors. We search for 100,000 in third party models to spot this easy mistake-
- Use Styles for formatting (no exceptions!) - this is faster and guarantees consistency.
- Use Range Names effectively – this makes formula more readable and avoids errors.
- Never hide a row or a column – except for constraining the worksheet working area.
- Import formula components locally, as practically as possible, all values that are not range named that drive a formula. This means you can press F2 or Trace Dependents and visually sense check – there is no need for Excel Add-ins to make this easier – just create the formula to be able to be readily reviewed.
- Each time-based worksheet must have the same start column or row. Be as consistent as you can on all aspects across worksheets.
- No Circular References. Circular Logic is often inevitable, but it must be managed by a macro to reliably solve and isolate.

# LAWS



# RULES

We strictly follow and recommend these rules. Circumstances occasionally warrant breaking these; but these are our “Do 99% of the time”

- Constrain the width of each worksheet – the only time to hide anything – no exceptions
- One Column = 1 Month – unless there is a good reason not to – this provides all the flexibility you need on a financial transaction outweighing the overhead of more columns.
- Remove grid lines and use the Freeze Pane – it’s cleaner and results in a more usable screen
- Keep worksheet names short – we typically get all worksheets into the default screen view, so a user (and we) do not need to spend time scrolling to find a sheet.
- Constrain all inputs that are not ‘easy numbers’ – for example don’t rely on a user to Type “Contract1”, we make them a selectable option simply using Data Validation / List.
- Avoid functions that will not work for the end user, for example when working with large banks they are often years behind the latest Excel version. This means powerful functions like Stack() and Let() become unusable.
- Add everything - this means you can use ALT + to “auto sum everything above”.
- Keep negative numerical concepts, such as expenditure positive, all the way until their sign property is important, for example when it is needed in a Cashflow. When working with complex debt and equity instruments, especially Reserve Accounts, this approach is vital.
- Only use the Brackets that are needed for a correct mathematical outcome.
- Purposeful denomination. If a value needs to be Divided by 100 (say from cents to dollars) then divided by 1,000 to show \$’000 do not divide by 100,000 show the two steps clearly, = ( X/Hundred ) / Thousand. In this instance we would use brackets to isolate the X and show intention.
- No Live Links to other Workbooks. Use an Import Data Island to strictly control any feed and confirm the integrity. At a minimum, manually import date using Paste Special (Unicode text) into a dedicated Worksheet.

These key rules are highlighted later in this Guide and covered in our training courses.

# EXCEL FUNCTIONS

There are honestly only a handful of functions you really need to master as shown below, this may surprise you but if you know these and you know how to 'join them together' you can solve all transaction and techno-economic analysis challenges.

**BASICS**

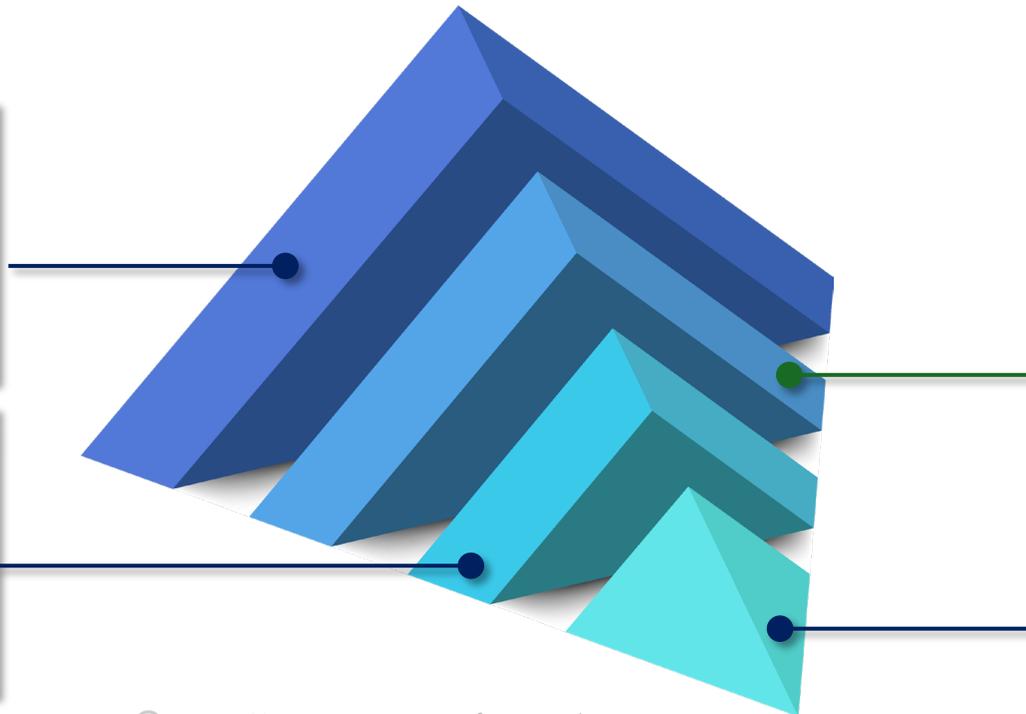
Think of these as the flour, eggs, milk, salt in your kitchen cupboard.

**SUM, IF, AND, OR, MIN, MAX, IFERROR, MONTH, YEAR, EOMONTH, EDATE, TRANSPOSE, TEXT, STRING(&), LEFT, RIGHT, MID, UNIQUE, FILTER, SORT, FIND, LET**

**POWER USER - COMPOUNDING**

Being able to transparently compound the Basic and Essential functions allows us to solve almost of all of the situations we encounter.

This approach also means other functions do not need to be utilised and you know intimately what is being performed.



**MORE ISSUES THAN BENEFITS**

Everybody works differently but if we wanted to find an error quickly, we'd look for the use of these functions. They introduce more errors than the benefits they bring.

**H/V/X Lookup, NPV, IRR, Pivot Tables, Power Query, User Defined Functions, SubTotal**

**ESSENTIALS - INDIVIDUAL**

These are the Power Tools you need to have and know how to use.

**INDEX, MATCH  
LOOKUP (Only one use)  
SUMIFS, SUMPRODUCT  
OFFSET (with care)  
XIRR (not XNPV)  
INDIRECT (with care)**

**POWER USER - VBA**

Not a 'function' but Visual Basic (VBA) is an incredibly powerful component of the Excel suite. Used properly, transparent code with a focus on making sure the user knows what is happening at all times allows you to calculations and optimisations you have no chance of doing manually.



Your secret weapon for everything else! Just make sure you ask the right question.

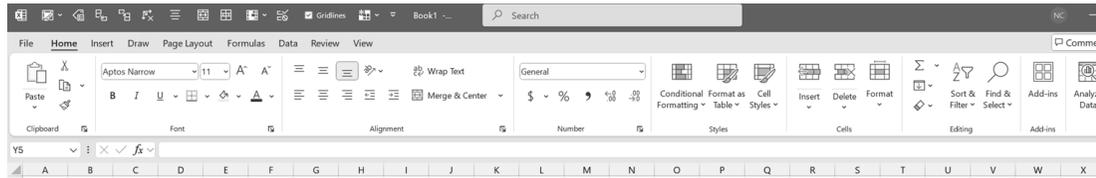
# EFFICIENCY

Even if you love modelling you don't want it to take longer than it needs to – to close the deal there are lots of other things that need to be done.

In this section you will see how we set ourselves up to conserve effort and time – maximising efficiency.



# EXCEL ENVIRONMENT



The standard Excel interface is at best distracting and at worst overwhelming compared to what you need. When working extensively in Excel your Cognitive Load is drastically reduced by using a clean workspace providing access to what you need 90% of the time – staying focused on the task at hand rather than navigating Excel.



Collapse the ribbon, set up the Quick Access Toolbar and focus on the development rather than have 25% of your screen space tied up with distracting information. Below is our standard working environment.

**RANGE NAME QUICK INTERFACE**  
Quickly create a Range Name. Highlight the relevant cell(s) and type in the name to assign. For example, **Timing.ModelStartDate** or **CostOfEquity**.  
You can't edit in this box which is why we use the Range Name Manager as our second QAT.

**QUICK ACCESS TOOLBAR**  
90% of what you need to model can be accessed by setting up the QAT; a huge driver of efficiency. ALT position number activates the functionality assigned to that position in the Toolbar. Eg ALT and+3 shows which cells are driving a formula.

**FORMULA BAR**  
Expanding the bar allows formula to be split across rows to indicate logical hierarchy. Improving readability and enabling evaluation of selective components.

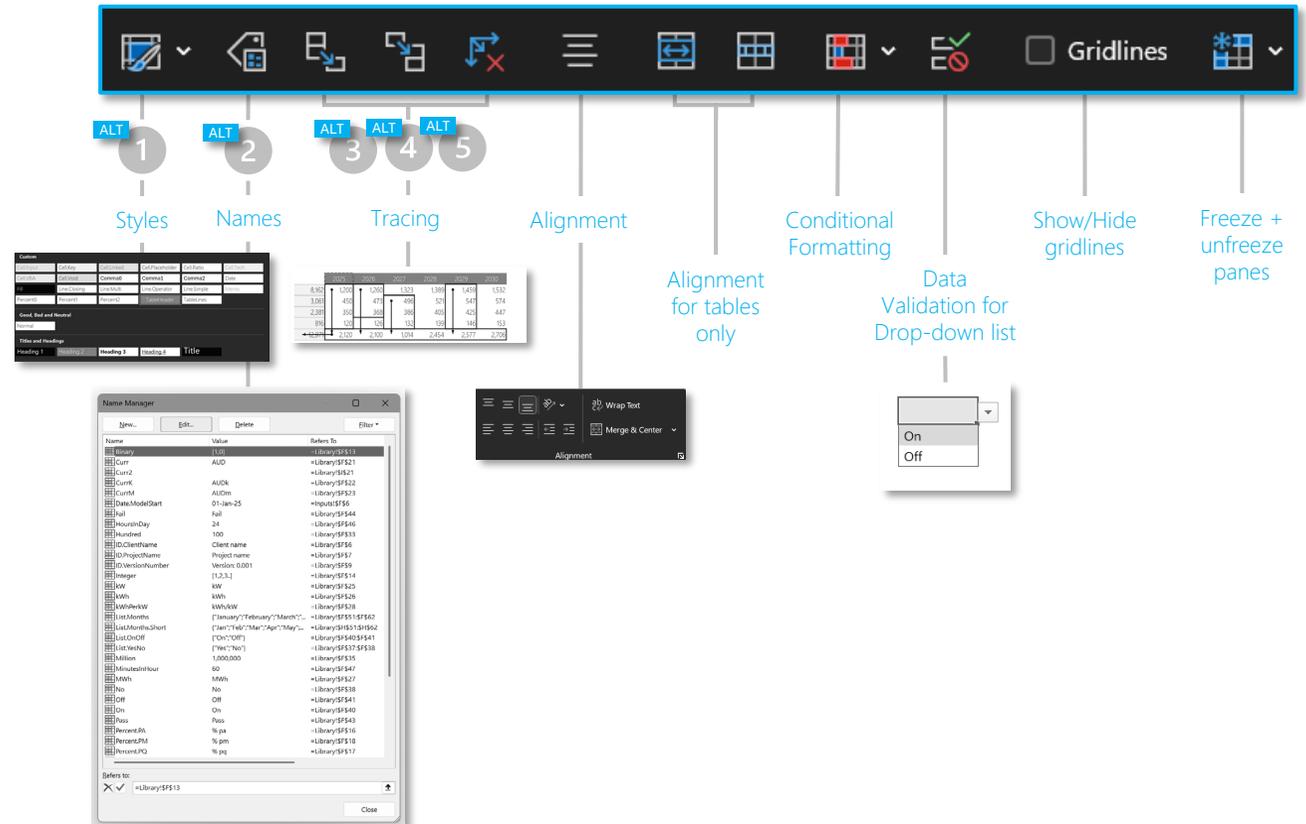
**NO ADD-INS**  
Third party add-ins are not really needed except for specialist simulation optimisation and model audit packages. Keep it lean, fast and stable.

# HOW TO WORK FASTER

Setting up and using the Quick Access Toolbar (QAT) removes the need to hunt down each tool when you need it but most importantly because it's there you will use it more.

Practically we only use the first five position numbers which mean you can fly through your work without using the mouse.

These few tools represent 90% of the Excel functionality you need.



# ESSENTIAL COMMANDS

There are so many shortcuts you could learn - some people learn a lot of them; however practically you only need to remember a handful to work very efficiently.

Here are most of the ones we use all day, every day; they are minimal because of the effectiveness of the Quick Access Toolbar.

We strike a balance between using the mouse and keyboard for the best results. If you already have the mouse in hand some tasks are quicker to do and if already using the keyboard, then the next command might be quicker to execute using keys - we minimise swapping but don't over think it - find what works for you.

## NAVIGATION

The best time savings are achieved through being able to rapidly move around the workbook, select cells and switch between formula bar and the worksheet.

It sounds simple, and it's not difficult but takes practice to lock it down without thinking consciously about it.

PURPOSE	COMMAND
Jump to the next non-blank cell	Ctrl + Arrow
Select a region cell by cell	Shift + Arrow
Select a region of cells quickly (to the first interruption)	Ctrl + Shift + Arrow
Send the first element of a selected range to the end. Quicker way of 'copying and pasting'.	Ctrl + R (to go right) Ctrl + D (to go down)
Switch between worksheets	Ctrl + PageUp/PageDown
Go to the formula bar / show dependents of a formula.	F2
Jump to the first reference of a formula	Ctrl + [
Paste only a Formula, Value of Format	Ctrl C (to copy) ALT + E + S then + F for formula V for value T for format
Completely clear a selected range of all content and formatting.	ALT + H + E + A
Group selected rows	Ctrl + Alt + → (ungroup using ←)
Insert cells	Ctrl + Shift + Plus
Delete cells	Ctrl + Shift + Minus
Find and Replace	Ctrl + H
Show all dependents within the worksheet	Ctrl + {
Force a calculation	F9



# LEARNING FROM SOFTWARE DESIGN

Let's consider what professional Financial Modelling is:

- Producing a comprehensive, robust calculation framework for complex, time constrained situations where the knowledge of everything that is needed is held by different people – and it all needs to be incorporated accurately and efficiently,
- Satisfying the analytical requirements of a range of stakeholders, each with different objectives and levels of commercial, financial and technical experience,
- A generally iterative process with information, requirements and expectations changing during development,
- Usually expected to be performed often in a very short period – schedules are generally days/weeks not months/years; and
- A process where the final product needs to be updated, used and verified by more than one person.

To accommodate these demanding requirements, we incorporate several fundamental principles of programmatic design. The acronyms are not very inspirational(!) but if followed are very effective.

- Single Source of Truth – SSOT (relates to Input data)
- Do Not Repeat Yourself – DRY (relates to Formula constructs)
- Output Led Design - OLD



# SINGLE SOURCE OF TRUTH

SSOT is a fundamental principle in software development and data management that ensures data consistency and accuracy across a system. It refers to maintaining a single, authoritative source for data, which all parts of a system reference.

Translating SSOT to day-to-day Financial Modelling means storing something once and then consistently referring to that information.

Typical examples include:

- All inputs (e.g. FX, Escalation rates, physical conversion factors);
- Names; and
- Activation switches.

By adhering to the SSOT principle, you will:

- Ensure a user can depend upon making a change in one place to flow through to all relevant sections.
- Debug faster – fixing an error at the SSOT level will automatically propagate through the model with one fix.



# DON'T REPEAT YOURSELF

A significant amount of inefficiency and error can be eliminated using the concept of Don't Repeat Yourself (DRY). DRY is a principle of software development aimed at reducing the repetition of code. It emphasises that every piece of logic should have a single, unambiguous and authoritative representation in the system.

Translating DRY to day-to-day Financial Modelling means calculating something once and then consistently referring to that calculation rather than calculating it again, either standalone or within formula.

Common examples include:

- The number of days in a period,
- Timing flags and counters (e.g., Financial Close, Construction Phases, Loan Grace Periods).

By adhering to the DRY principle, you can:

- Completely remove timing logic from formulas,
- Completely eliminate the risk of calculating something incorrectly on subsequent attempts and work faster and more accurately by simplifying your workflow,
- Debug faster – fixing an error at the SSOT level will automatically propagate through the model with one fix:
- Minimise Daisy-Chaining – making it easier for a user to find the source information.



# OUTPUT LED DESIGN

Utilising Output-Led Design (OLD), we focus from the outset on what matters most - the structure and quality of the output. The model structure is then designed by working backward to the inputs.

This approach flips the common practice of starting with inputs, building calculations, and hoping the desired output can be created. Practically, a balance must be struck since requirements often evolve during engagements. However, Output-Led Design minimises surprises, improves efficiency, and ensures that all stakeholders provide input from the beginning—not at the end.

It is very difficult, and a situation we are often called in to address, when a new piece of analysis has been started 'in Excel' or by re-using an existing file – a recipe for disaster. A clean start has never been beaten and can be surprisingly faster.

The benefits of Output Led Design are wide ranging, manifesting in better:

- **Clarity:** Developers and stakeholders gain a clear understanding of goals and direction from the outset.
- **Efficiency:** Time and resources are invested in achieving prioritised goals, avoiding unnecessary work, and ensuring more reliable timelines.
- **Planning:** Gives the project team time to compile information needed for the model rather than being asked at the last minute.
- **Alignment:** Guarantees the result satisfies the requirements of the end users.
- **Quality & Accuracy:** By focusing on what matters most from the beginning, there is an early emphasis on key objectives, ensuring the most critical aspects are not left until the last 1% of the development timetable, as is often the case with traditional approaches.



# SAFE IS FAST

Excel does crash/hang from time to time; and often at the worst time. We find it usually happens when:

- Pasting or copying large amounts of data into a file. Only bring in values, the default will bring in meta-data from the sheet which is a significant source of issues.
- Accidentally copying to the bottom of the worksheet (over 1 million rows)
- Running macros that do not have fail safe mechanisms
- There is too much use of conditional formatting
- Leaving your PC on with Excel open when Windows Updates are performed, especially true in corporate environments.

Ask yourself how much work would you be prepared to lose? Think of the development process as climbing a mountain, how far would you be happy to fall? Your fall is limited to the last peg you clipped onto. Think of a peg as the 'last file you saved with a unique file name'. Just saving over the current file is not a 'peg'. Autosave is not a 'peg'!! Our tips are:

- AutoSave – it sounds like a good idea but isn't, turn it off.
- If a file is Auto Recovered, it's as good as dead. An Auto Recovered file is a rebuilt version of the file and can include major issues. You need to go back to your last file saved with a different name. Painful lessons learned in the trenches means this is a non-negotiable.

We recommend:

- Manual save every 10-15 minutes with a unique name appending a temporary ticker to the file name. Practically re-doing 30 minutes of work will only take 10 minutes and we can live with this but it's not ideal at the end of the day or before a meeting! Our system does this automatically, but it's an easy task to do manually.
- We name files so they are ordered sequentially in a folder and avoid any potential confusion to a user. Also avoid the word "Final" in a filename but use a system that works for you – just be consistent.

**YYYYMMDD\_CompanyName\_ProjectName\_V{a.b.c}\_[save ticker]**

for example:

**@09:15 20250201\_GoldCo\_GoldHill\_V0.9\_a**

**@09:30 20250201\_GoldCo\_GoldHill\_V0.9\_b**

**@09:45 20250201\_GoldCo\_GoldHill\_V0.9\_c**

# SAFE IS FAST

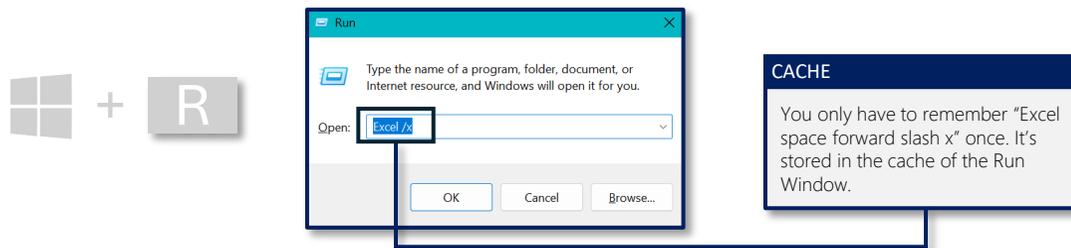


When we need to have more than one workbook open, we keep our development environment safe and stable by opening 'the other file(s)' in a new 'instance' of Excel – this is a completely new 'opening' of Excel, you can think of it as two or more files 'not knowing the others are open'. This is not the same as just opening a file within Excel and is another time and frustration saver, because if one crashes it doesn't affect the other file.

Two good reasons to do this are:

- Large, slow, buggy, linked, crash prone files from clients do not impact our live development work.
- We can run solve, scenarios / sensitivities, optimisation macros on two different live files at the same time without it impacting the other. Working in 'Parallel' not working in Series.

Avoid opening a workbook directly from email, unless you save it down first.



# CLARITY

In this section you will see how we ensure Clarity - primarily through workbook layout, sheet structure and formula composition.

Time to get your Zen on.



# SHEET STRUCTURE | INPUTS



## STATUS

It is very helpful to display the live scenario and corresponding KPIs and Test Status at the top of each sheet.

## PROJECT 1

Sunny Hill - Stage 1      Wind (MW)    450  
 Tests: Integrity    Pass      Project    11.23%  
 Tests: Commercial    Pass      Equity    16.20%  
 Scenario: Bank downside case

## CONTROLLING INPUT

Dates are not manually entered but are selected using data validation through a list tied to the main timing rows. This ensures that only valid dates can be applied.

## PROJECT SET UP

Model start date    1-mmm-yy    01-Jan-25

Projects	Status	Energy type	Installed MW	Irradiance kWh/kW
1 Sunny Hill - Stage 1	On	Solar	450	1,200
2 Sunny Hill - Stage 2	On	Solar	350	1,200
3 Solar Park 1 - Stage 1	On	Solar	180	1,200
4 Solar Park 1 - Stage 2	On	Solar	90	1,200
5 Solar Park 1 - BESS	On	BESS	500	-

Financial close
31-Aug-26
31-Oct-26
30-Nov-26
31-Aug-36
31-Aug-36

Dev Start	Development			FID	Months	Construction		COD	Operations	
	Months	Delay	Overall			Delay	Overall		Years	End
01-Mar-25	12	-	12	28-Feb-26	18	-	18	31-Jul-27	20	30-Jun-47
01-May-25	14	-	14	30-Jun-26	22	-	22	31-Mar-28	20	29-Feb-48
01-Jun-25	18	-	18	30-Nov-26	21	-	21	31-Jul-28	20	30-Jun-48
01-Mar-35	18	-	18	31-Aug-36	24	-	24	31-Jul-38	20	30-Jun-58
01-Mar-35	18	-	18	31-Aug-36	6	-	6	31-Jan-37	20	31-Dec-56

## FLOW

It is helpful to present inputs that align with the flow of the project. After the initial setup, we step through the cash flows and considerations the project will encounter as it evolves over time

## DEVELOPMENT EXPENDITURE

## CAPITAL EXPENDITURE

## GENERATION

Seasonal variance	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Check	
Sunny Hill - Stage 1		120%	105%	95%	90%	90%	85%	85%	95%	95%	105%	110%	120%	100%
Sunny Hill - Stage 2		120%	105%	95%	90%	90%	85%	85%	95%	95%	105%	110%	120%	100%
Solar Park 1 - Stage 1		120%	105%	95%	90%	90%	85%	85%	95%	95%	105%	110%	120%	100%
Solar Park 1 - Stage 2		120%	105%	95%	90%	90%	85%	85%	95%	95%	105%	110%	120%	100%
Solar Park 1 - BESS		-	-	-	-	-	-	-	-	-	-	-	-	-

## SPARKLINES

For concepts where trends are visually useful, we use sparklines to provide a subconscious appreciation of patterns. This approach works well for illustrating seasonality, mine plans, phased rollouts, price curves, and similar data

## CHECKS

Where we can test something, we do. Other examples include Capital Expenditure and manual debt repayment profiles.

If they fail, they would be shown in red using conditional formatting and also flow through to the Test Panel.

## QUASI-INPUTS

Inputs sheet need not only contain raw inputs, but all Inputs should be on dedicated Input sheets. Pink cells are linked to the Scenario manager and grey cells are calculations that will:

- 1) Be needed anyway; and
- 2) Make it easier to establish or verify the inputs.

# SHEET STRUCTURE | CALCS



Excel's calculation engine calculates sheets from left to the right, and within each sheet it calculates down. We aim to lay worksheets out so they are broadly in the order of the calculation chain however we will never compromise Useability for this consideration.

Keep worksheet names short but helpful, if they are too long you will not be able to see them all in one place. More importantly, the formula on other sheets that refer to those sheets will have unnecessarily long text in the formula.

Do not hide a worksheet. Like all rules there are occasional exceptions, for us, it's when the file opens with a Legal Disclaimer but other than that hidden worksheets are the fingerprints of a file that's harder to use. If you have information that is needed but does not need to be obvious, just put it on the right-hand side and give a short label.

Hiding sheets, including password protecting sheets is not effective – if you do not want information present, remove it.

# SHEET STRUCTURE



**LOCALISED KEY INFO**

It is helpful to see key information relating to the asset. This can be quite extensive but keep it to the left of the freeze pane as Col M will be copied across.

**DATA GROUPING**

Data grouping is an essential tool to ensure ease of navigation and lowers Cognitive Load. It has minor drawbacks when referencing a grouped row from another worksheet. Well worth it.

**DATA UNITS**

Unless obvious within the line description such as in Row 10 and 11 it is essential to show the units of every row. This forces dimensional consistency and avoids confusion.

**HEADERS**

When collapsed the worksheet should appear as a Table of Contents. This makes it easier to use and grouping calculations forces good house-keeping for efficiency.

		PROJECT 1																								
		Windy Hill - 2b		Wind (N.W)		450																				
		Tests: Integrity		Pass		Project		11.23%																		
		Tests: Commercial		Pass		Equity		16.20%																		
		From		01-Jan-25		Feb-25		01-M																		
		To		31-Dec-24		31-Jan-25		2		Feb-25		01-M														
		<b>TIMING</b>																								
		Days in month				31		28		31		30		31		30		31		31		31		31		
		Hours in month				744		672		744		720		744		720		744		744		744		744		
		<b>Flags</b>																								
		Development		[1,0]		01-Mar-25		28-Feb-26		12																
		Construction		[1,0]		01-Mar-26		31-Jul-27		6																
		Operations		[1,0]		01-Aug-27		30-Jun-47																		
		Project lifecycle		[1,0]		01-Mar-25		30-Jun-47		18																
		Financial Close		[1,0]		31-Aug-26		1																		
		<b>Counters</b>																								
		Development		[1,2,3,]						1		2		3		4		5		6		7		8		
		Construction		[1,2,3,]																						
		Operations		[1,2,3,]																						
		Project lifecycle		[1,2,3,]						1		2		3		4		5		6		7		8		
		<b>PROJECT ESTABLISHMENT</b>																								
		Development																								
		Construction																								
		<b>OPERATIONS</b>																								
		Generation																								
		Revenue																								
		Operating costs																								

**LINKED CELLS**

Importing key information drastically reduces off-sheet references and aids understanding and the intent of calculations. We highlight them with a Style.

**CONSTRAIN WIDTH**

This enables rapid copying using Ctrl R. We leave one column blank to avoid a paste accidentally going all the way out for 16,000 columns.

**ONE FORMULA PER ROW**

Without exception the first 'live' column must be the same all the way across. We routinely copy this column all the way to the end on all similar sheets for consistency.

# FORMULA STRUCTURE



- Aim to only reference cells that are local to the calculation or are Range Named. We work on the basis that the cell precedents should be visible on the screen without scrolling unless it is range named. It's the best approach 99% of the time and means you can sense check in your head.
- Only use numerical values of 1 or 0 in a formula. Any other values are best defined, named and referenced. For example: type 1,000,000 enough times, and you will make a mistake; and when dealing with large or unfamiliar physical quantities such as kWh or Ounces it's hard to pick this up visually.
- Always minimise off-sheet references. When 'going off sheet' and coming back to the source worksheet remove the automatically added source worksheet name. Not doing this makes formula unnecessarily longer, reducing Clarity.
- Use spaces, especially after comma separators – this makes it easier on the eye and more practical to evaluate an isolated component.
- Stack logical hierarchy with line breaks (ALT + ENTER) this enables more sophisticated and very transparent calculations.
- Do not perform sub-calculations within formula, such as Days in a Period or Month. Use our Single Source of Truth approach and back it up with flags / timing / counters.
- Be conscious of Daisy-Chaining. Practically minimise the number of steps that need to be followed back to the primary drivers of a calculation. This is a balance when localising cells but makes it efficient for a user to track what is being calculated and the input sources and for making changes on the fly.
- Do not squeeze too much into one formula, three rows of simple formula are much better than one complicated one. Calculating an Interest Charge is a good example.
- Use Arrays with caution, whilst powerful, they are harder to trace so use them but not as a default. You might understand it, but others may not.

# FORMULA STRUCTURE



## EXAMPLE

Calculating a CPI escalation rate. Splitting into two separate lines. 1) Import the annual rate 2) Operate on it.

 `=(1+LOOKUP(YEAR(M$3),Inputs!$G$200:$M$200,Inputs!$G$203:$M$203)^(1/Operations!M$5))-1`

 `Line1 = LOOKUP(M$4, CPI.Live, CPI.HeaderYears )`  
`Line2 = ( 1 + Line1 )^-M$5 - 1`

## EXAMPLE

A quantity conversion in a mining operation. One formula would be fine if following the rules but breaking onto three lines has at least two benefits. 1) Each new line may be useful for later calculations 2) Each step can be readily sense checked 3) Three simple formula are better than one longer one (most of the time).

 `=MinePlan!M$100/12*Index(Operations!$G$400:$M$400,Match(M$4,Operations!$G3:$M3,0))*33.1035/1000000`

 `Line1 = MinePlan!M$100 / MonthsInYear`  
`Line2 = Index( Mined.Pit1, Match(M$4,MonthEnds,0) )`  
`Line3 = Line1 * Line2 * GramPerTroyOunce / Thousand`

# FORMULA STRUCTURE



## EXAMPLE

Does a formula like this look familiar? Let's take a look at how you can avoid or transform this messy formula with only basic syntax and basic range naming. We would lay the subcomponents out locally and operate using helper-lines but a few spaces and line breaks makes the formula significantly easier to work through and debug.



```
=iferror(if(M$4 ≤ Inputs!$D$31, (MinePlan!M$100/12*Index(Operations!$G$400:$M$400, Match(M$4, Operations!$G3:$M3, 0))*33.1035/1000)+(MinePlan!M$101/12*Index(Operations!$G$400:$M$400, Match(M$4, Operations!$G3:$M3, 0))*33.1035/1000)), if(M$4 ≤ Inputs!$D$32, MinePlan!M$110/12*Index(Operations!$G$400:$M$400, Match(M$4, Operations!$G3:$M3, 0))*33.1035/1000, 0), 0)
```



```
=iferror( If( M$4 ≤ OpsEndDate,
    (Mined.Pit1 / MonthsInYear * Index( Mined.Pit1, Match( M$4, MonthEnds) +
      Mined.Pit2 / MonthsInYear * Index( Mined.Pit2, Match( M$4, MonthEnds) ),
    (Mined.Pit3 / MonthsInYear * Index( Mined.Pit3, Match( M$4, MonthEnds) ),
    If( M$4 < Pit1.DecomissioningDate,
      RehabilitationPit / MonthsInYear,
      0,
    ) * GramsPerTroyOunce / Thousand
  ), 0)
```

# CONTROL ACCOUNTS



A Control Account, sometimes referred to as a Corkscrew account, is a time-based ledger of movements that affect a closing balance. It is one of the most common structures we use to ensure Clarity and is another straight-forward construct that ensures when something unexpected happens we can readily debug or adapt.

There are two reasons to use Control Accounts:

- It is the most transparent way to create and track a balance (of anything, financial, physical, etc.)
- The Balance Sheet is readily compiled by referencing the closing balances of control accounts.

**BALANCE SHEET FEED**

The Balance Sheet is entirely composed of references to closing balances of Control Accounts. This approach takes all 'calculations' out of the Balance Sheet making it clearer, easy to debug and much easier to construct.

**1 LINE = 1 ITEM**

Each line only references one other line in the model, nothing is aggregated – this same approach as used in financial statement construction ensures when there is an issue the irregularity is easy to identify and remedy.

Control A/c		31-Jan-25	28-Feb-25	31-Mar-25	30-Apr-25	31-May-25	30-Jun-25	31-Jul-25	31-Aug-25	30-Sep-25	31-Oct-25	30-Nov-25	31-Dec-25	31-Jan-26	28-Feb-26	31-Mar-26	30-Apr-26	31-May-26	30-Jun-26
Balance b/f	AUDm			100.00	100.00	100.00	100.00	92.60	92.60	92.60	85.80	85.80	85.80	77.60	77.60	77.60			
Drawdown	AUDm			100.00															
Repayment - Scheduled	AUDm			(20.00)				(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)			
Repayment - Sweep	AUDm			(9.58)				(2.40)											
Refinance	AUDm			(71.42)												(71.42)			
Balance c/f	AUDm				100.00	100.00	100.00	92.60	92.60	92.60	85.80	85.80	85.80	77.60	77.60	77.60			

**ROW SUMS**

Summing a row, all the way to the end is a powerful way of sense checking a row. In a control a/c the "sum of row sums" of each movement usually should add to zero.

**BALANCES**

The last line of a control account is always the sum of the Opening Balance and all movements within the period.

Opening Balance aka Balance b/f  
Closing Balance aka Balance c/f

**FORMATTING**

We always format the Closing Balance (using a Style) with a bold lower line so that users are aware this quantity is "carried forward" as a balance rather than a cashflow or physical movement.



Range Names have been in Excel since day one, however we do not see them used effectively anywhere near enough. For us they are essential.

Range Names allow you give English names to cells so rather than refer to a random looking cell in a formula you can refer to a description. They can be applied to a single cell such as a date or a 2D table which makes an Index(Match) ultra transparent.

Critically, when using VBA, Range Names are the only practical way of manipulating the Spreadsheet – so when solving or simulating, it is essential.

They are one of those things that are worth spending a few seconds setting up each time, but the gains for the Developer and the User are immense. They are incredibly useful for:

- Physical conversion factors.
- Activation switches to turn calculations On/Off - essential for Portfolios and Debt Instruments.
- Commercial inputs.
- Numerical Constants – such as one million. On a recent Data Centre deal we needed Quadrillion! This isn't overkill, it's a key-way to avoid errors and increase Clarity.
- Any feed for a Drop-Down list - On,Off, Yes, No, Pass, Fail, Bank Base-Case, Down-Side

We've found over the years the key to unlocking the power of Range Names is a consistent rule for naming anything. We adopted Linnaean taxonomy, the approach used in Biology; and if we can't make it fit neatly, we use CamelCase to differentiate the words when a spacer is unnecessary. We find a full-stop cleaner and easier to read than an underscore (and takes up less space).

**{Family}. {Category}. {Anything else}**

**Debt.FinancialClose.Date**

# RANGE NAMES

We would resign if we couldn't use Range Names!

# RANGE NAMES

Range Names make formula much easier to read and they also make formulas easier to create. Faster? Yes, assuming you know the range names you have created simply start typing it in and Excel's formula inteli-sense will provide a drop-down list of matching options which you can simply select and accept. So don't think you need to type them in completely and that will take too long.

If they were not awesome, we wouldn't use them. You may have spotted the Range Name manager is position 2 on our Quick Access Toolbar. This means ALT + 2 opens the Name Manager without having to either find it or remember yet another keyboard combination (CTRL + F3). Whilst not shorter it's always ALT + [1,2,3,4] making access easier to recall - these small things make all the difference when you are working under pressure.

There are two scopes of Range Names, an aspect that's often overlooked.

- Workbook
- Worksheet

Worksheet Range Names take precedence over Workbook scoped Range Names. For most modelling Workbook scoped Range Names are fine and are the default setting when you create a Range Name in the Name Box in the top left of the toolbar. However, worksheet scoped Range Names are essential for Portfolio Modelling or when duplication is required.

Once you have created a Range Name, if you change it then the change is updated in all formula that use it. Our biggest tip here is to consider and create your Range Names as you need them the first time; if you don't, any formula you created before you named the Range Name will not pick it up.

# RANGE NAMES

This is a small selection of the range names we use. Each model has different requirements, our non-computational Range Names are all located on a dedicated worksheet. We work on the basis that we only want to type something once – remember Single Source of Truth. This an efficient way of working and maintains consistency.

When there is a home for items you spend less time typing, make less errors and formula are clearer from the outset. Consider the way components and tools are stored – think of a Porsche factory workshop compared to a local garage mechanic.

Broadly we categorise them as:

- Units
- Constants
- Lists – everything from Calendar month names, Locations, Supplier Names, Asset names
- Switches - Binary outcomes driven from user {Yes, No}, {On,Off} choices
- Really technical stuff that an executive user doesn't need to worry about

## ESSENTIAL NAMES

Whilst each new model requires its own specific Range Names there are only a handful that are always used.

When there is a home for items, like in a kitchen pantry, you spend less time looking for it, make less errors and formula are clearer from the outset.

UNITS		
Binary	[1,0]	Binary
Integer	{1,2,3..}	Integer
Percent per annum	% pa	Percent.PA
Percent per quarter	% pq	Percent.PQ
Percent per month	% pm	Percent.PM
Currency 1		
Currency	AUD	Curr
Currency k	AUDk	CurrK
Currency m	AUDm	CurrM
kW	kW	kW
kWh	kWh	kWh
MWh	MWh	MWh
kWh/kW	kWh/kW	kWhPerkW
CONSTANTS		
Ten	10	Ten
Hundred	100	Hundred
Thousand	1,000	Thousand
Million	1,000,000	Million
Yes	Yes	Yes
No	No	No
On	On	On
Off	Off	Off
Pass	Pass	Pass
Fail	Fail	Fail
Days in a year	365	DaysInYear
Hours in a day	24	HoursInDay
Minutes in hour	60	MinutesInHour

## COMPOUND NAMES

Grey cells auto update when their blue drivers change. For example, if AUD changed to A\$ then both A\$k and A\$m update here and therefore throughout the workbook.

We wouldn't set a model up to be 'any currency' unless specifically needed, as this impacts the basis of the input's values - quite risky.

# VECTOR FOCUS

We develop transaction ready models for a wide range of situations, with deep experience in

- Renewables – especially multi-technology, multi-regional, portfolios.
- Mining – all Metals, Minerals and Processing.
- Manufacturing – Chemicals and Green Fuels.
- Infrastructure and Regulated Assets.
- Corporate Business Modelling.

We deliver in-house, in-person training for your teams, we specialise in working with small groups and building capability over their careers. Our approach has been developed over decades, is 100% hands-on and we are told our passion shows. Our current courses are:

- Financial Modelling Fundamentals
- Project Finance Modelling
- Modelling Renewables Projects
- Modelling Three Way Financial Statements
- Advanced debt modelling using Visual Basic

If you'd like us on your side in a transaction or to build your teams capabilities contact us at [hello@vectorHQ.co](mailto:hello@vectorHQ.co) or check us out online at [vectorHQ.co](https://vectorHQ.co)



**MODEL  
DEVELOPMENT**



**TRAINING**

# UX

Understand how to lower a user's Cognitive Load by making sure the information needed is readily available in the expected location with other supporting information clearly shown.

Complex environments require a highly considered display of information.

This enables the user to focus on analysis rather than searching for how something has been calculated.

High UX = High confidence.



# THE PRINCIPLE OF UX

User Experience – this doesn't really take any extra time and makes all the difference for the end user but also for the developer. In a nutshell it's:

- Usability and User-centric focus.
- Functionality.
- Information architecture.
- Consistency.
- Engagement.
- Performance.
- Aesthetics.
- Time.

Generally, in the world of finance, UX isn't something that comes naturally! We don't consider it as something we have to do as a stand-alone stage once everything is finished, it's a way of working all the way through.

Adhering to good UX means:

- The user's impression is positive from the outset.
- All users cognitive load is reduced as much as is practical.
- There is an inherent trust and confidence in the model from the outset.

# STYLES



If you find yourself in the standard Formatting Menu you are wasting your time! We only format cells with these styles with very occasional custom formatting for a unique and special cell that doesn't warrant a 'style'. The default Styles in Excel are useless - so create your own.

Ours are below and used extensively in our training courses.

## EXCEL DEFAULT STYLES

Right click and Delete these but keep "Normal" and change if to suit.

## OUR STYLES

We only format cells with these Styles. They are optimised for clarity, speed and consistency. From the Quick Access Toolbar ALT+1 brings this menu up. It takes a few seconds to set-up, but you only have to do it once as they can be imported into other files.

The screenshot shows the standard Excel style menu. It is divided into several sections: 'Good, Bad and Neutral' (Normal, Bad, Good, Neutral), 'Data and Model' (Calculation, Check Cell, Explanatory T..., Input, Linked Cell, Note, Output, Warning Text), 'Titles and Headings' (Heading 1, Heading 2, Heading 3, Heading 4, Title, Total), 'Themed Cell Styles' (various percentage and accent combinations), and 'Number Format' (Comma, Comma [0], Currency, Currency [0], Percent). A red 'X' icon is overlaid on the left side of the menu.

The screenshot shows a custom style menu. It is divided into sections: 'Custom' (Cell.Input, Cell.Key, Cell.Linked, Cell.Placeholder, Cell.Ratio, Cell.Tech, Cell.VBA, Cell.Void, Comma0, Comma1, Comma2, Date, Fill, Line.Closing, Line.Multi, Line.Operator, Line.Simple, Memo, Percent0, Percent1, Percent2, TableHeader, TableLines), 'Good, Bad and Neutral' (Normal), and 'Titles and Headings' (Heading 1, Heading 2, Heading 3, Heading 4, Title). A green checkmark icon is overlaid on the left side of the menu.

# BUSINESS SCHEMATIC



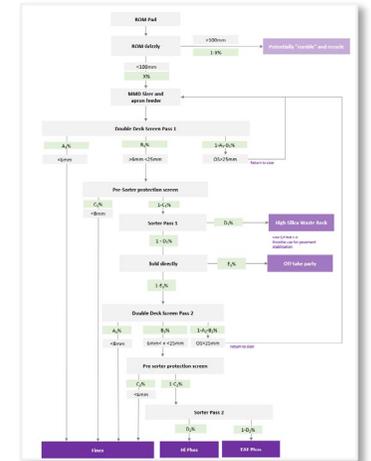
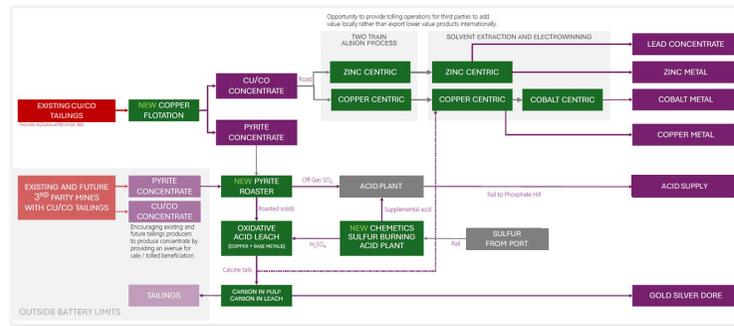
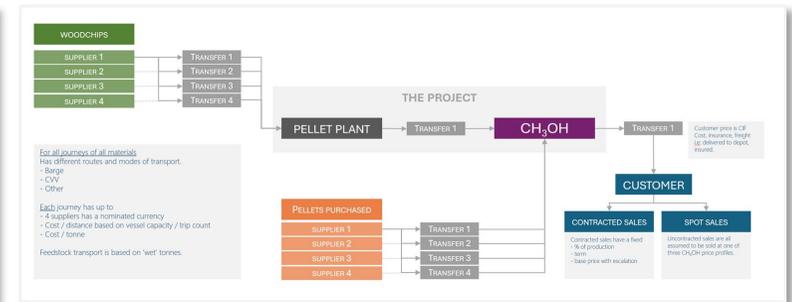
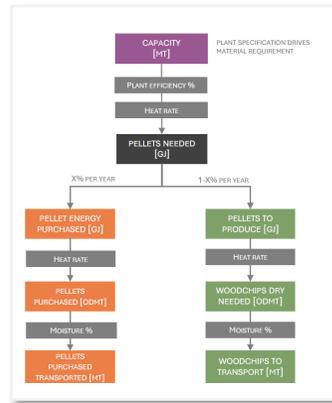
Don't start developing a model in Excel. This is a showstopper and a common source of inefficiency.

Errors especially arise from retrofitting project elements when secondary considerations are realised later. It is crucial to first lay out a clear model-logic schematic – we also document all our working notes using Notion, the best tool we've found yet for collaboration and a clear head.

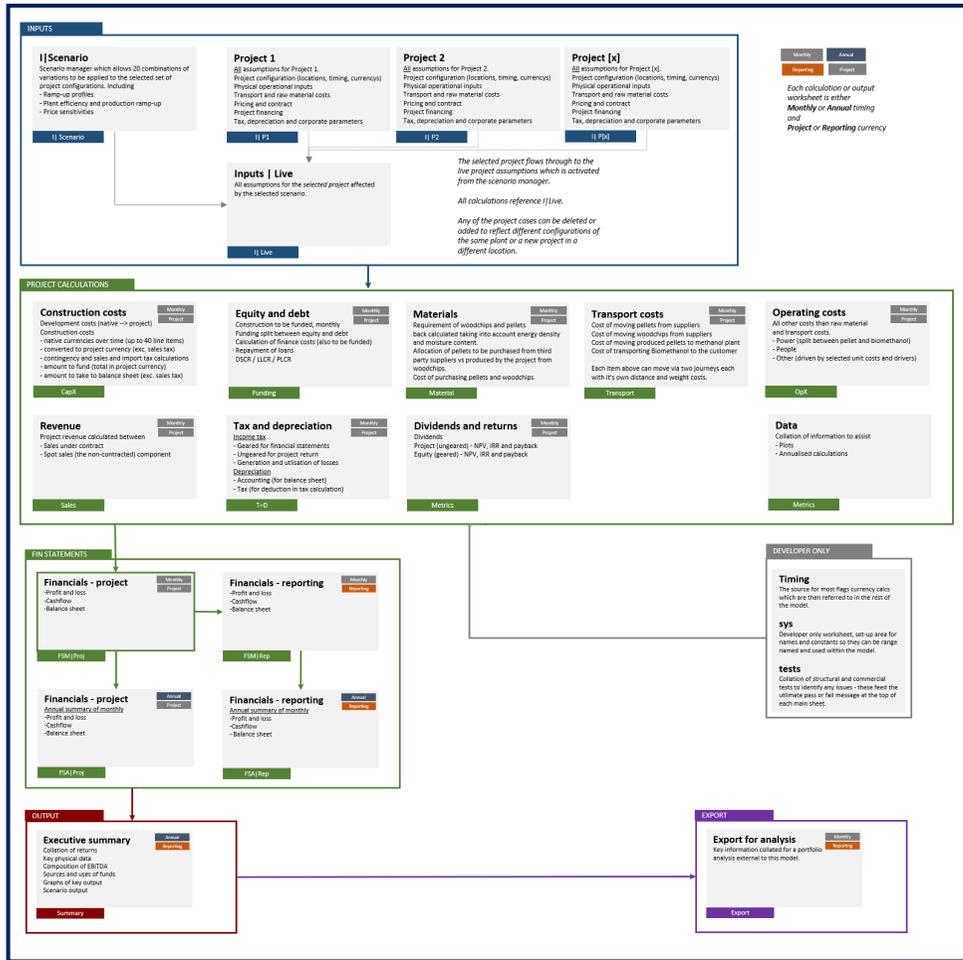
While the examples may seem simple, the clarity of business logic flow or corporate structures is achieved through deliberate and considered iterations. This exercise is not only valuable for us but also enhances the client's understanding of the level of detail and flexibility that will be delivered. It also forces the client to engage at the most important point - the start not the end....

Time spent organising this logic at the start saves significant time, minimises errors and ensures everyone is aligned. Ultimately it turns the modelling into "the easy" part.

If you think you don't have time to do this – that's exactly why we do it.



# MODEL FLOW



We include an overall flow of logic map at the start of most models. This example is not expected to be read in detail but rather to illustrate what can be achieved fairly quickly.

A user will typically only scan it but gives a helpful helicopter view of the relationship between the sheets. Live values and navigation links can be added to this but it's generally not worth the extra overhead.

A 30 second inspection of this kind of information, presented as one the first sheets also gives a user confidence that the layout has been well considered, and that care has been taken.

# A SUCCESSFUL PROCESS

## You have an

- Investment
- Divestment
- Acquisition
- Valuation
- Borrowing
- Reporting event

## Model risk considerations

- Will it be “bankable”
- Will it only be audited “at the end”
- Higher than needed model audit cost

## Modelling

- Stakeholder requirements
- Investigation of existing analysis
- Output design

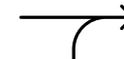
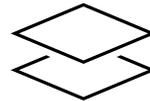
Only open Excel when the expectations of the model are agreed. Deals evolve but change is best managed via process flexibility rather than baking in too much functionality.

## Adopt an Agile workflow process

Deals evolve so the modelling process needs to be flexible to accommodate changing timelines and the “Known and Unknown Unknowns” – it keeps it interesting but requires discipline and efficiency to keep the model on track to deliver Confidence.

## Success

- Stakeholder confidence
- Very efficient model audit with no material findings.
- Model can be used internally



## You need

- Bankable outcome
- Time efficient solution
- Specific skills
- Support only through the transaction

## Constraints

- Time
- Experienced available resources

## Internal resource considerations

- Do they have required domain knowledge
- Do they have bandwidth
- Do they have the experience
- Do they have the efficiency
- Will they be consistently available

## Tips

Insufficient rigor at this stage will cause major inefficiency at later stages. This risk is mitigated by a thorough kick-off workshop.

One of the best ways we find to clarify business and structural logic are detailed flow sheets.

Within reason the model should be able to be released at any stage – no significant “tidy-ups” at any stage.

## Typical stages

- Kick-off meeting
- Weekly stand-up meeting
- As needed, daily, communications
- Staged release for concept testing
- Preparation of Output / Scenarios
- Working with stakeholders
- Model documentation
- Pre-Audit testing
- Model Audit support
- Financial Close Model execution

# INTEGRITY

Understand the most common non-commercial errors, how to avoid making them and how to trap them.

We differentiate between Excel errors and Own Goals that you should avoid!





# COMMON ERRORS

This section highlights what to avoid and some tips on how to trap the errors when they occur.

After owning and running one of the world's largest, independent model audit firms and having been on the receiving end of 100+ model audits Nick advises that overall, the most common material errors are rarely the developer's interpretation of business logic, they are the surprisingly simple things:

- SUM() not picking up the expected rows / columns (often caused by an insertion)
- Incorrect anchoring of rows or columns
- Formula variance across columns
- Incorrect conversion (think FX / cent-based tariffs, physical quantities)

This is not an extensive list but if you can address them, you will have solved 90%+ of the likely errors you will make. Errors are generally made when:

- Working too fast
- The model has not been developed following the Laws described earlier
- The model has multiple places an update needs to be made and one or more is missed.

We recommend setting up a comprehensive Test Panel – this means you have a better chance of trapping an error as soon as you've made it. We categorise Tests into three:

- Structural
- Commercial
- Debt / Macros / Other

We make the same mistakes as everybody else - that's the nature of Excel. We just make less and pick them up quickly. Other than working incredibly carefully, we have developed a proprietary tool that identifies them as we work and heals them.

# ERRORS | SUM



**INCORRECT SUMS**  
 Capturing incorrect rows within a SUM usually occurs when changes are made and often when working too fast or under pressure. It's a very common error.

OPERATIONS			
Generation			
Revenue			
<b>Summary</b>			
Nominal			
Contracted	AUDk	72%	8,162
Sold at spot	AUDk	27%	3,061
FCAS	AUDk	21%	2,381
REC	AUDk	7%	816
Total	AUDk	100%	11,283

1,200	1,260	1,323	1,389	1,459	1,532
450	473	496	521	547	574
350	368	386	405	425	447
120	126	132	139	146	153
2,120	2,100	1,014	1,065	2,431	2,553

**TRAPPING THE ERROR**  
 Select a cell which should capture everything then CTRL SHIFT {  
 Cells not included will show up as white. It's helpful to save your file before doing this.

OPERATIONS			
Generation			
Revenue			
<b>Summary</b>			
Nominal			
Contracted	AUDk	72%	8,162
Sold at spot	AUDk	27%	3,061
FCAS	AUDk	21%	2,381
REC	AUDk	7%	816
Total	AUDk	100%	11,283

1,200	1,260	1,323	1,389	1,459	1,532
450	473	496	521	547	574
350	368	386	405	425	447
120	126	132	139	146	153
2,120	2,100	1,014	1,065	2,431	2,553

No need to type =SUM(Range), just auto sum using ALT + to automatically create the sum of all cells above the live cell. This is a fast way of working but be aware that it will include all numerical values which means you run the risk of including table headers.

To avoid this issue, leave a line between the data and the header or just be aware and be careful – we do the later and Excel is improving the way this works, but it does depend on your version.

Summary	2025	2026	2027
Nominal			
Contracted	1,200	1,260	1,323
Sold at spot	450	473	496
FCAS	350	368	386
REC	120	126	132
Total	4,145		

**AVOIDING THE ERROR**  
 CTRL + may also include the header row. Depending on the magnitude of the header this may or may not be easily spotted. We find this in third party files commonly.

# ERRORS | ANCHORING



Anchors (\$) are fundamental in formula creation; they allow you to lock a reference to a row or a column such that when the formula is copied those references do not move. Anchoring is essential to minimise errors but also in the creation of blocks of formula – it’s a huge driver of efficiency.

In the example below, there are better solutions we use to avoid this issue altogether (eg Ranges Names, however these require planning as to how they will be used).

Mastering anchoring is essential for increasing efficiency, when you create blocks of code if it is correctly anchored it enables replicating existing structures. For example - you have three tranches of debt with the same structure but need each subsequent block to pick up the information associated with Tranche 2 and 3, not Tranche 1. This does require care but copying an existing asset saves a lot of time compared to creating it from scratch.

We always build a section as if we were going to replicate it – don’t anchor a reference unless you have to – this also makes formula leaner. Become comfortable using F4 to cycle modes, don’t type an anchor sign.

Escalation		2025	2026	2027	2028	2029 +
Labour	% pa	3.00%	3.20%	3.20%	3.20%	3.20%
Diesel	% pa	4.00%	4.10%	4.50%	4.90%	3.50%

Year - Calendar	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2026	2026	2026
Days in month	31	28	31	30	31	30	31	31	30	31	30	31	30	31	31	28	31
Labour	% pa	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.20%	3.20%	3.20%
Diesel	% pa	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.00%	4.10%	4.10%	4.10%

Labour	% pm	0.25%	0.23%	0.25%	0.24%	0.25%	0.24%	0.25%	0.25%	0.24%	0.25%	0.24%	0.25%	0.27%	0.24%	0.27%
Diesel	% pm	0.25%	0.23%	0.25%	0.24%	0.25%	0.24%	0.25%	0.25%	0.24%	0.25%	0.24%	0.25%	0.27%	0.24%	0.27%

**ANCHORING THE WRONG ROW**  
 Picking up the annual rate in the helper row is performed correctly – the next block ignores the Diesel fuel escalation rate. Easy to spot when the numbers are laid out simply but much harder in the heat of a transaction or in 'one line'.

Escalation		2025	2026	2027	2028	2029 +
Labour	% pa	3.00%	3.00%	3.00%	3.00%	3.20%
Diesel	% pa	4.00%	4.10%	4.50%	4.90%	3.50%

Year - Calendar	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2025	2026	2026	2026	2026	2026
Days in month	31	28	31	30	31	30	31	31	30	31	30	31	30	31	31	28	31	30	31
Labour	% pa	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.00%	3.20%	3.20%	3.20%	3.20%	3.20%
Diesel	% pa	4.00%	4.10%	4.50%	4.90%	3.50%	-	-	-	-	-	-	-	-	-	-	-	-	-

Labour	% pm	0.25%	0.23%	0.25%	0.24%	0.25%	0.24%	0.25%	0.25%	0.24%	0.25%	0.24%	0.25%	0.24%	0.25%	0.27%	0.24%	0.27%	0.26%
Diesel	% pm	0.25%	0.23%	0.25%	0.24%	0.25%	0.24%	0.25%	0.24%	0.25%	0.25%	0.24%	0.25%	0.24%	0.25%	0.27%	0.24%	0.27%	0.26%

**INCORRECT COLUMN ANCHORING**  
 Intentionally easily spotted in this example but with longer tables it’s easy to not anchor the columns of a table (or any other reference) especially when the issue is only noticeable 2-3 screen widths out to the right.

# ERRORS | COLUMN VARIANCE



Remember the Law “one row, one formula” from the earlier section? Even with best intentions it is easy to break this by accident and not know about it. We focus on the first ‘live column’ and regularly copy the entire column all the way to the end of the workings – this routinely overrides any accidental error.

Developers can use Arrays, which consistently fill the entire row with the same formula – whilst this is helpful to avoid this error, arrays are often harder to understand for everyday / executive stakeholders, so we avoid them.

**ONE FORMULA ONE ROW**

The formula in this first ‘live column’ must always go all the way to the end of workings. No exceptions, timing considerations would be managed via helper flags.

**FORMULA CHANGES**

An unintentional formula variance is easily introduced, especially when working too fast and not paying attention to the Freeze Pane. It is hard to spot so we routinely highlight the entire first column and copy to the end.

**HARDCODED VALUE**

If you have used F9 to evaluate a formula, or part of a formula, unless you press Escape to exit the value will be locked in. This error is easily made and only practically remedied by white-washing the first live formula – as below.

Summary			2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	
Nominal														
Contracted	AUDk	62%	17,048	1,200	1,260	1,320	1,380	1,450	1,532	1,600	1,669	1,773	1,862	1,955
Sold at spot	AUDk	19%	5,177	450	473	386	405	425	447	469	492	517	543	570
FCAS	AUDk	18%	4,972	350	368	386	405	425	447	469	492	517	543	570
REC	AUDk	6%	1,705	120	126	132	139	146	153	161	169	177	186	195
Total	AUDk	100%	27,454	2,120	2,100	904	2,338	2,455	2,578	2,707	2,842	2,984	3,134	3,290

Summary			L	M	N	O	P	Q	R	S	T	U	V	W
Nominal			2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	
Contracted	AUDk	57%	13,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200
Sold at spot	AUDk	21%	4,950	450	450	450	450	450	450	450	450	450	450	450
FCAS	AUDk	17%	3,850	350	350	350	350	350	350	350	350	350	350	350
REC	AUDk	6%	1,320	120	120	120	120	120	120	120	120	120	120	120
Total	AUDk	100%	23,320	2,120	2,120	2,120	2,120	2,120	2,120	2,120	2,120	2,120	2,120	2,120



Highlight the working area on the whole sheet then CTRL + R



# ERRORS | TRAPPING

The best you can achieve is that if when you make an error you are immediately aware of it. There is no 'silver bullet', but a Test Panel is a powerful tool that is easily implemented as you develop the model, not as a final stage.

As you work, build up a test panel of anything you can link together – we think at both a technical level, and from a stakeholder's perspective.

Cells that should yield the same value (Deltas that can be composed)

- Sources = Uses (Transaction costs and Funding)
- Sources = Uses (Mass Flow in physical / manufacturing)
- Total Cashflow (Monthly) = Total Cashflow (Annual)
- Balance Sheet Check = 0
- Debt drawn = Debt Repaid / Refinanced
- Asset Additions = Depreciation + Write-offs

Values that would be good to know about immediately rather than having to find out later.

- Negative cashflows
- Project return < Equity return
- A macro needs to run

## OVERVIEW

Collating the results of all tests into one master table facilitates easily identifying which type of test is triggering a failure. This can be readily linked to the top of each sheet.

	Count	Outcome
Integrity	-	Pass
Commercial	1	Fail
Debt	-	Pass
Other	-	Pass
All	1	Fail

## LIVE FEED

Create the test in the relevant worksheet and then link to this panel or create the test within the panel. Keep the test unfiltered or rounded and manage that aspect as a separate field as shown here.

On/Off	Description	Model	Rounding	Rounded	Applied
On	Balance Sheet - Local Currency (Monthly)	-	3	-	-
On	Balance Sheet - Local Currency (Quarterly)	-	3	-	-
On	Balance Sheet - Local Currency (Annual)	-	3	-	-
On	Balance Sheet - Reporting Currency (Monthly)	-	3	-	-
On	Balance Sheet - Reporting Currency (Quarterly)	-	3	-	-
On	CF(Reporting (Monthly) = CF(Reporting (Quarterly) = CF(Reporting (Annual)	-	3	-	-
On	Financial Sources = Uses (Monthly)	-	3	-	-
On	Financial Sources = Uses (Annual)	-	3	-	-
On	Mass Flow closed loop	-	3	-	-
Off					
					Pass

## SWITCHES

It is very helpful to be able to switch some tests off some of the time, especially when you are making changes, and you expect them to be failing until you are ready.

On/Off	Description	Model	Rounding	Rounded	Applied
On	Count of Negative Cashflows (Monthly)	10	1	10	1
On	Assets JOB + Additions = Depreciation + Write-Off	-	3	-	-
On	Debt   Senior   Lock-Ups   Count	-	-	-	-
Off	Debt   Senior   Defaults   Count	-	-	-	-
On	Debt   Junior   Lock-Ups   Count	-	-	-	-
On	Debt   Junior   Defaults   Count	-	-	-	-
On	Debt   DSRA   Drawn   Count	-	1	-	-
On	Tax   Thin Capitalisation Breach	-	-	-	-
On	Stockpile   High Grade   Depleted	-	-	-	-
					1
					Fail

## ISOLATION

We keep the Tests within a dedicated worksheet, this is an important aspect of the model and warrants being stand-alone.

## Tests

On/Off	Description	Model	Rounding	Rounded	Applied
On	Debt   Senior   Repaid within tenor	0	2	-	-
On	Debt   Senior   Limit fully utilised	-	-	-	-
On	Debt   Junior   Repaid within tenor	-	-	-	-
On	Debt   Junior   Limit fully utilised	-	-	-	-
On	Debt   Senior   DSRA   Fully funded	-	-	-	-
Off					

## DELTA OR COUNT

Some tests are better as a count of failures rather than a numerical delta (A-B=0). Commercial tests, such as negative cashflows are more useful to know how many times it occurs rather than an absolute value.

# ERRORS | OWN GOALS



Anyone can make these errors, you don't have to be new to modelling – but when you make them, you'll kick yourself! Here are some common ones to be vigilant of.

## Common calculation issues:

- Calculating Interest on a Closing Balance rather than Opening Balance.
- Incorrect number of days in a period – easy to spot in a monthly model but harder on a quarterly basis ~ \$250k per quarter on a \$100m loan at 10%.
- Applying an FX rate the wrong-way-around; easier to make with close currencies; same vigilance for physical conversion factors.
- Incorrect conversion of annual % rates to monthly/quarterly – decompound or simple rate conversion? Degradation and interest rates for examples.
- Applying a growth rate on the wrong time basis or escalating at the start of a period that is too early.
- Principal repayments <> Debt drawn. Often through the incorrect use of PPMT or not trapping the final period to be an exact payment of the balance due, especially important in sculpted profiles or sweeps.
- Project Return incorporating elements of gearing, the only exception being  $K_d$  within WACC.
- Dividends over-paying through incorrect restrictions on cash available – trapping is helped by negative cash balance checks and Row Summary comparisons.
- IRR or NPV being calculated at the end of the first evaluation period rather than the start. Working with XIRR() and XNPV() helps solve this but IRR() and NPV() have different rules – another reason we don't use them.

## Working practices:

- Sending the wrong version of a file.
- Sending a file to the wrong person - the only practical reason to use workbook passwords. We use User-Rights which are much more effective.
- Losing work by not saving regularly and uniquely – or saving over an important version.
- Making several individual changes in a stable important version and not recording the impact of each change or even documenting the changes made. i.e.: “here is the new version” is not helpful in the heat of a deal.
- Making a change but not testing the impact is as expected in all scenarios. Don't Set-and-Forget. Inspect all KPI's in all scenarios change are explainable.
- Typos! Nothing undermines a stakeholder's confidence than spelling – a signal that diligence has not been taken. E.g., Principal vs Principle.
- It's firmly against our working practices to recycle a model – sure it's a 'fast start' but overall, significantly less efficient. However, if you do take this approach make sure absolutely everything has been sanitised.
- Doing all the work needed but not updating Scenario Tables before release.
- Importing information from another workbook, inadvertently polluting your file with all Styles, Custom Formats and potentially links to other workbooks. Only import data as Unencoded Text. Ctrl + Z doesn't undo in this situation.

# INSPIRATION

This document has shown you the fundamentals to follow which work for us – in this section gain some further insight as to how we work in a professional development environment.

On every transaction we execute everything we've shown you, and we lean on our experience to innovate where needed to ensure a successful outcome.



# DISPLAYING INFORMATION

Every transaction is different – there is ‘no one size fits all’ when it comes to displaying information. We consider what data will be needed by different stakeholders, management, equity investors or lenders all look for different metrics and scenarios.

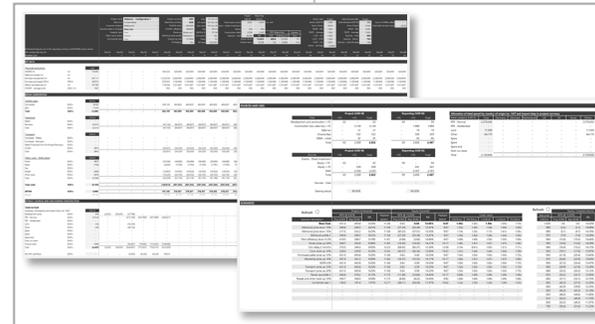
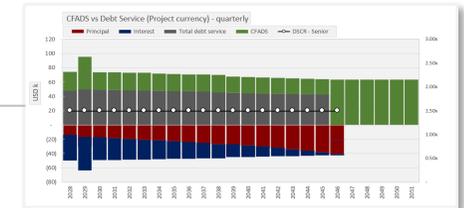
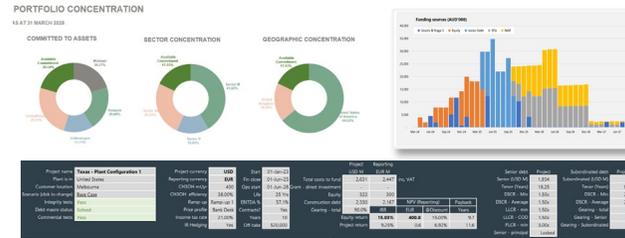
Managing the analytical requirements of equity stakeholders is sometimes the most challenging as they may have quite different assumptions and structural analysis than a lender and may not want to reveal that within a lenders model.

In preparing an executive summary, the primary model output should show clearly:

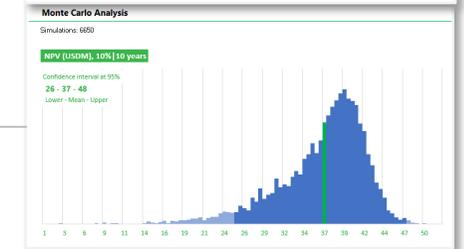
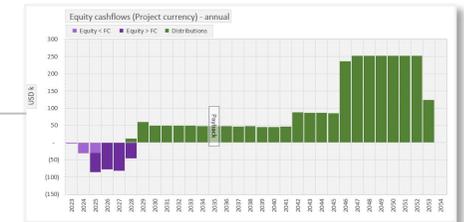
- All the key assumptions.
- Physical, commercial and financial trends.
- Key investment and integrity metrics.
- KPIs over all scenarios.

The elements to consider incorporating are:

- Tables of data.
- Time based plots (often annualised).
- Histograms of returns.
- Tornado and Value bridges.
- Dynamic pie charts – use carefully as they do distort pure information but are visually appealing.



Layout to print on 1-5 pages landscape



# DEBT IS REWARDING

One of the mathematically interesting aspects of transaction modelling is the constant innovation of the capital structuring for deals. Often these are adaptations of existing instruments but require navigation around circular logic from interrelated calculations - one small change can make things very interesting.

- Principal repayments with multiple sizing targets, constraints and covenants.
- Standby / Overrun accounts with dynamic seniority considerations.
- Reserve accounts funded at different stages by different parties with varying capital mixes.
- The full spectrum of financing fees.
- Refinancing / Regearing.
- Grants with dynamic milestones.
- Dynamic Borrowing Base.
- Debt at Project level and/or Portfolio Level.
- Managing all the above changing throughout a deal.....

Fundamentally our DNA is in advanced debt optimisation. We know any structure that can be documented can be modelled using the 'tried and trusted' tools:

- Timing Flags
- Transparent control accounts
- VBA to power automated, tightly controlled, iterative loops with transparent user dashboard and macro consoles

```
VECTOR – DEBT SOLVING CONSOL
Console cleared
18:52:02 Setting calculation mode to Manual.
18:52:04 Clearing debt and equity limits.
18:52:04 Calculating 00:00:01
18:52:05 Calculating 00:00:01
18:52:06 Calculating 00:00:01
18:52:07 Solving construction funding loop 1.
18:52:07 Calculating 00:00:01
18:52:08 Calculating 00:00:01
18:52:09 Solving construction funding loop 2.
18:52:09 Calculating 00:00:01
18:52:10 Calculating 00:00:01
18:52:11 Solving construction funding loop 3.
18:52:11 Calculating 00:00:01
18:52:12 Calculating 00:00:01
18:52:12 Solving construction funding loop 4.
18:52:12 Calculating 00:00:01
18:52:13 Calculating 00:00:01
18:52:14 Solving construction funding loop 5.
18:52:14 Calculating 00:00:01
18:52:15 Calculating 00:00:01
18:52:16 Solving construction funding loop 6.
18:52:16 Calculating 00:00:01
18:52:17 Calculating 00:00:01
18:52:18 Solving construction funding loop 7.
18:52:18 Calculating
```

## MONITORING A SOLVE ROUTINE

As debt is solved it is crucial to know what is going on. This can be achieved by updating cells on a dashboard, our approach updates the user with a wide range of status updates – very useful for a developer wanting to identify where calculation inefficiencies are. It adds no overhead to the solve speed, unlike updating a cell in a worksheet. This consol floats like a menu and is very useful when a structural change is made that may be causing solve issues, once identified it can be resolved or made more efficient via managing tolerances or changing the orders of solve loops.



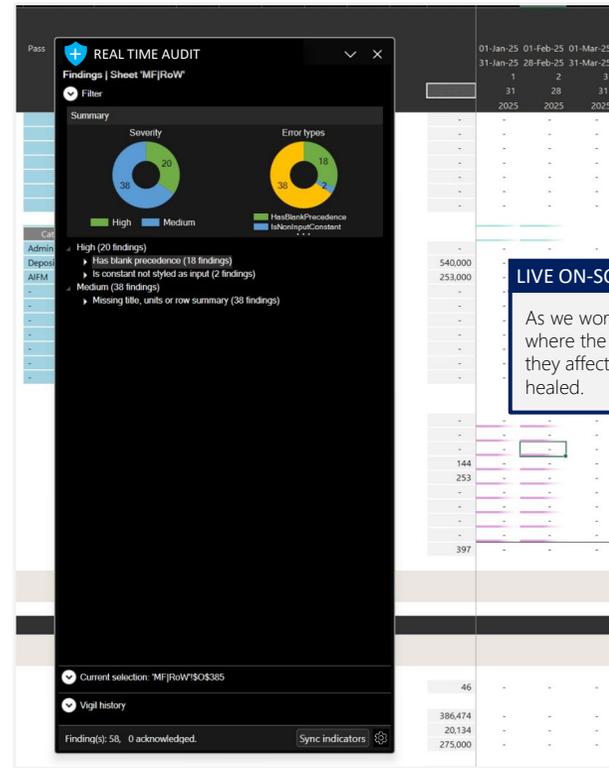
# OUR ONLY TOOL

Specific to our working environment, as we develop models our proprietary tool stands vigil over our work, highlighting and healing issues as we make them. The rules of our own methodology have been codified into a live application within Excel which works 'under the hood' in real time – without slowing anything down. This doesn't negate anything covered already it just makes it incredibly difficult for an error to slip in.

This means we can work reliably and efficiently. We and our client will always be confident in the model, from the outset solving the issue that Model Audits are performed at the end of a linear process – the last point in a deal that you want to know about an error!

Some of the functionality includes:

- Real time identification of errors via application of 30+ of our Laws and Rules.
- Real time 'healing' of an acknowledged error.
- Automatic creation of standard concepts to speed things up but also avoid making a mistake. Control Accounts, Totals, 3 Way Financial Statement outlines, Timing Flags and more.
- Identification and re-factoring (and simplification) of formula that are becoming too long or hard to understand.
- Constant tracking and awareness of remaining errors to be remedied.
- Automatic and intelligent Range Name completion and correction.





# VECTOR WHEN IT MATTERS

Our purpose is to develop elegant solutions for complex situations enabling stakeholders to make confident investment decisions, build team capabilities - as trusted advisors. We can be in for the deal but usually once the deal is done our client's ask us to work with them each subsequent time and train their teams.

- Expertise – our two principals deliver decades of hands-on experience for 100% for your engagement.
- Responsiveness – here when you need us.
- Mission focused – your project is our project.
- Knowledge sharers – no restrictions, no licenses.

If you'd like us on your side in a transaction or to build your teams capabilities contact us at [hello@vectorHQ.co](mailto:hello@vectorHQ.co) or check us outline at [vectorHQ.co](https://vectorHQ.co)



**MODEL  
DEVELOPMENT**



**TRAINING**