Modular Spreadsheet Development
Fundamentals
MODULAR SPREADSHEET DEVELOPMENT

Please check www.bestpracticemodeling.com for any updates to this document. All copyright in this document and any derivation of this document is owned by bpmToolbox Pty Ltd, bpmModules Pty Ltd, BPM Analytical Empowerment Pty Ltd and/or its associated entities.

Copyright © bpmToolbox Pty Ltd, bpmModules Pty Ltd, BPM Analytical Empowerment Pty Ltd and associated entities.

This is a Best Practice Modelling publication. To keep up to date with the ongoing evolution of best practice modeling practices and technologies, join the Best Practice Modeling Network at www.bestpracticemodeling.com/network.

Document version: 7.0.0.0

BEST PRACTICE MODELLING (BPM)

Best Practice Modelling (BPM) is a business modeling organization that specialises in the provision of best practice spreadsheet modeling resources including software, consulting and training services. BPM is the founding member of the Spreadsheet Standards Review Board (SSRB) and remains committed to overseeing the ongoing maintenance, development and adoption of the Best Practice Spreadsheet Modeling Standards. BPM can be contacted as follows:

Website: www.bestpracticemodeling.com
Email: info@bpmglobal.com

IMPORTANT NOTICES

Many of the examples provided throughout this document have been created within Microsoft Excel using bpmModules® – a best practice content management and sharing tool available from Best Practice Modelling (www.bestpracticemodeling.com). The SSRB is of the opinion that the use of bpmModules within Microsoft Excel is the most efficient and effective means of implementing modular spreadsheet development. A free trial of bpmModules may be downloaded from the Best Practice Modelling website at www.bestpracticemodeling.com/software/packages.
Table of Contents

Introduction ............................................................................................................. 3
Module Areas ......................................................................................................... 5
Modules ................................................................................................................ 6
Module Components ............................................................................................ 7
  Module Component Types ................................................................................. 9
Module Links ....................................................................................................... 10
Core Modules ..................................................................................................... 13
  Time Series Module ....................................................................................... 14
  Checks Modules ............................................................................................. 15
  Keys Modules ................................................................................................. 16
Multiple Workbooks ......................................................................................... 17
Benefits .............................................................................................................. 18
Practical Implementation ................................................................................... 19
Introduction

Modular Spreadsheet Development (‘MSD’) refers to the building of spreadsheets in an orderly piece-by-piece fashion, with each piece being linked together to form a complete model. Under this approach, any spreadsheet can be viewed as being comprised of separate units or ‘modules’ which can easily be added, modified and removed without affecting the fundamental structure or integrity of the surrounding model. This capability thereby facilitates the otherwise impossible re-use and sharing of spreadsheet content without the use of templates.

In many cases, a well-constructed best practice model will already be ‘modular’ even if the model developer has not consciously applied modular development principles during its development – i.e. its content will have been developed and laid out in separate inter-related areas, each with a different purpose. For example, consider a spreadsheet which analyzes the profitability of a business based on revenue and expense projections. The table of contents within this spreadsheet is shown below, along with images showing how the content within this table of contents could be segregated into modules:

Traditionally, a model developer would view the content within this spreadsheet as being comprised of sheets – i.e. an assumptions sheet containing the revenue and expenses assumptions, an outputs sheet containing the revenue and expenses outputs, and a second outputs sheet containing the income statement. In a modular spreadsheet development environment, the content within this spreadsheet would be viewed as being comprised of modules and module components – i.e. revenue and expenses modules, each with an assumptions and outputs module component, and an income statement module with a single outputs module component.

In this way, modular spreadsheet development recognizes that different components of any spreadsheet model can be developed as dedicated standalone modules, each with assumptions and/or outputs that are specific to one sub-section of analysis. Each of these modules can be used either in isolation to form a spreadsheet model that addresses one specific field of analysis, or linked to other modules, thus expanding the fields of analysis and functionality contained within the overall spreadsheet model.
In reality most spreadsheets will contain more than three modules, and the modules will often be grouped into areas (known as ‘module areas’) to assist other model developers in understanding similarities and relationships between them. The following diagram shows an example of how a business planning model might be sub-divided into modules grouped into module areas:

This documentation aims to provide an overview of the fundamental concepts and terminology used to implement modular spreadsheet development to facilitate efficient spreadsheet content management and sharing. It is not designed to provide detailed information or instructions regarding the development of specific modules and the content within them. For information regarding specific modules, visit the Best Practice Modeling website at www.bestpracticemodeling.com.

Note that whilst the examples throughout this document are based on business planning, modular spreadsheet development principles can be applied to any type of spreadsheet.
Module Areas

In many cases, the division of a spreadsheet into individual modules results in the creation of a large number of modules. To help manage these modules and understand the relationships between them, modules can be grouped based on similar characteristics into collections of modules referred to as ‘module areas’. Module areas provide model developers and model users with an intermediate framework of understanding between the high-level model purpose and the individual modules themselves.

The image below demonstrates how five modules within a spreadsheet might be allocated to four module areas based on common characteristics:

![Modules within Module Areas Diagram]

In this example, the module areas are represented by hollow blocks and modules are represented by solid blocks. The revenue and expenses modules have been grouped into a module area named ‘Operational’ as a result of the fact that they both provide operational analysis to the model. The income statement module has been put in a separate ‘Financial Statements’ module area to distinguish it from the time series, operational and checks content within the spreadsheet. In the event that additional modules are added to the spreadsheet, they can either be added to these existing module areas, if appropriate, or added to newly-created module areas which reflect the characteristics of the modules to be added.
Modules

A module is a self-contained portion of a spreadsheet that carries out a specific function and may be used alone or linked to other modules to carry out a broader function.

A common example of a module within a business planning model is a ‘revenue’ module – i.e. the portion of the model that collects assumptions which are used to calculate revenue and undertakes the revenue calculations. Another common business planning module is an ‘operating expenditure’ module, which is used to collect assumptions and provide operating expenditure outputs.

There are no definitive rules governing the 'modularization' of a spreadsheet model, although a reasonable guide is that it should be possible to communicate the broad content and purpose of each module within the spreadsheet using one or two short sentences. On a more practical level, the model developer should consider the implications of the way in which a spreadsheet is divided into modules for the ongoing maintenance and development of the spreadsheet model, and potentially other spreadsheet models if the modules are likely to be reused at some stage in the future.

For example, consider the spreadsheet which was discussed at the start of this chapter which analyzes the profitability of a business based on revenue and expense projections, as shown in the diagram below:

By segregating the content within this spreadsheet into three modules, the model developer effectively quarantines the revenue, expenses and income statement analysis into three areas which can therefore be modified or removed with minimal impacts on each other. Additionally, if the model developer wants to use the revenue module from within this spreadsheet in another spreadsheet, this can be done even if the other spreadsheet contains a very different income statement and does not contain an expenses module.

Hence, a primary consideration when determining the content to include within each module within a modular spreadsheet model is the implications of the resulting content segregation for the ongoing development of the model, and other potential uses of its modules within other spreadsheet models going forward.
Module Components

In a non-modular spreadsheet, content is simply added to sheets and the resulting model is effectively viewed as a collection of sheets. However, when modular spreadsheet development principles are applied to a spreadsheet, content is developed on a module-specific level rather than on a sheet-specific level because sheets may contain content from more than one module. This is done by building spreadsheet content on a ‘module component’ level rather than on a sheet level.

A module component is comprised of one or more rows within a sheet and holds content relevant to a particular module – e.g. the assumptions used to project the sales of a business. Modules may contain a single module component or multiple module components with each module component either collecting assumptions from model users (i.e. assumptions module components) or providing outputs (i.e. outputs module components).

The diagram below shows the composition of a simple example revenue module containing assumptions and outputs module components:

![Example Revenue Module Components Diagram]

The assumptions module component within this revenue module might be used to collect assumptions about the sales of a number of categories of revenue over a number of time series periods, as shown below:

![Example Revenue Module – Assumptions Module Component]

In this example, the module component is an assumptions module component and spans rows 12 – 22. The outputs module component corresponding to this assumptions module component is shown below:
This outputs module component, which calculates the sales forecasts for each revenue category, spans the rows 12 – 24. Note that this module component, like its related assumptions module component, has been positioned on a time series sheet and is located immediately below the time series period titles.

To understand how multiple module components can be located within a single sheet, consider an example model in which model users are required to enter both revenue and expenses assumptions, and separate revenue and expenses modules have been used to collect these assumptions. Furthermore, these modules have both been included in a module area named ‘Operational’ and their assumptions located within a single time series assumptions sheet due to the obvious relationship between the revenues and expenses of the business. The image below shows how this assumptions sheet might appear, and red rectangles have been used to indicate the location of the two assumptions module components within the sheet:
The development of spreadsheet content using module components in this way results in the inter-changeability of modules because the modules containing these module components can be added, removed, moved and/or replaced as required based on the purpose of the surrounding spreadsheet. For example, if the model developer decided to change the way in which revenue assumptions were collected in the above example, the revenue module (and its module components) could be deleted and replaced with another previously-built revenue module rather than editing the existing module components or building a new module from scratch. It is this ability to re-use and share modules that makes modular spreadsheet development so powerful.

**Module Component Types**

The module components within a module are differentiated based on their purpose, which in turn determines the type of sheet on which they are located – e.g. assumptions module components must be located on assumptions sheets. When built in accordance with the Best Practice Spreadsheet Modeling Standards, module components must be one of the following basic types:

<table>
<thead>
<tr>
<th>Module Component Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumptions</td>
<td>• Collects assumptions from model users.</td>
</tr>
<tr>
<td>Outputs</td>
<td>• Calculates and presents output information to module users.</td>
</tr>
<tr>
<td>Lookups</td>
<td>• Contains lookup tables for use within the module.</td>
</tr>
</tbody>
</table>

Assumptions module components are located on assumptions worksheets and outputs module components are located on outputs worksheets, with placement on blank or time series sheets dependent on the content within module components. Lookups module components must always be located on lookups sheets.

For more information on the content and purpose of different sheet types, see the General Concepts spreadsheet modeling area of the Best Practice Spreadsheet Modeling Standards.
Module Links

In most cases a spreadsheet built based on modular spreadsheet development principles will contain multiple modules, and there will be relationships between some or all of these modules. These relationships are captured by the establishment of ‘module links’ between different modules. Module links are formula links between modules which transport data between them, and are therefore crucial to developing truly dynamic modular spreadsheets. They are also the key to properly compartmentalizing analysis, which in turn quarantines risks of error and secures the structural integrity of the spreadsheet model.

For example, a revenue module could be linked into an income statement module (i.e. via a direct formula reference such as ‘=Revenue.TO!J20’), and a unit prices module could be linked into the revenue module. As a result, the impact of a change in prices on revenue could be analyzed, followed by the impact this change in revenue has on earnings, via the income statement. Hence, in this example, the linking of the three modules allows model users to understand the earnings implications of price fluctuations.

In order to fully understand these linking processes, model developers need to understand the concepts of precedent modules, dependent modules, links in and links out. As shown in the table below, there are two types of module links:

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links in</td>
<td>Formula links into a module from its precedent modules.</td>
</tr>
<tr>
<td>Links out</td>
<td>Formula links out of a module to its dependent modules.</td>
</tr>
</tbody>
</table>

Hence, when discussing the linking of modules, a module can be referred to as being one of three kinds:

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Module</td>
<td>A module that is receiving ‘links in’ from one or more ‘precedent modules’ and/or providing ‘links out’ to one or more ‘dependent modules’.</td>
</tr>
<tr>
<td>Precedent Module</td>
<td>A module that contains data that links into the active module.</td>
</tr>
<tr>
<td>Dependent Module</td>
<td>A module to which the active module links out data.</td>
</tr>
</tbody>
</table>

A simple diagrammatic representation of the concept of links in and links out between an active module and its precedent and dependent modules is provided below:

In the above example, the trade debtors module is the active module, receiving links in from the revenue module (precedent module) and also providing links out to the balance sheet and cash flow statement modules (dependent modules).
In essence, module links are simply direct formula links between the module components within different modules, but they should be viewed as arteries through which only the information necessary to relate two modules flows. Hence, module link formulas should not contain complex formulas and in most cases should contain only a direct formula link without any functions. It is also common and acceptable to include a negative sign within module link formulas when numerical data needs to be negated when being linked into a dependent module – e.g. expenses linking out of an operating expenditure module into an income statement.

The diagram below highlights the links which might exist between three related modules – i.e. revenue and operating expenditure (precedent) modules linking into a (dependent) income statement module. In this example, two module links are required – one to link revenue out of the revenue module into the income statement module, and one to link operating expenditure out of the operating expenditure module into the income statement module.

These links have been highlighted via the red connecting lines on the right side of the diagram:
Note that even though multiple categories of revenue and operating expenditure have been linked into the income statement, only two module links can be identified – revenue and operating expenditure. The following module links diagram reflects these precedent and dependent module relationships:

![Module Links Diagram](image)

The diagram below reflects the relationships between the same modules, but also includes the names of the module links for clarity:

![Example Income Statement Module Links In](image)

The above example is a very simple example. In reality, a modular spreadsheet will often contain many modules with many links between them, and understanding the difference between the different modules and the links between them is paramount to understanding the model. The diagram below shows a more realistic example of the precedent and dependent modules of an income statement module within a dynamic business planning module, and the module links which create the relationships between these modules:

![Example Income Statement Module - Precedent & Dependent Modules](image)
It is important to note that not all modules have precedent modules and/or dependent modules. For example, modules that forecast product sales volumes may not have any precedent modules, but often provide volumes outputs to dependent modules, such as revenue or operating expenditure modules, which might use volumes as a driver for forecasting.

Other modules may have precedent modules but not have dependent modules, such as an enterprise valuation module, which collects cash flow data from various modules to be used as a basis for a discounted cash flow valuation, but does not link out any data to any other modules.

The key to building a robust and flexible suite of modules which can easily be re-used and shared is determining the best links structure between related modules, as there will often be many different ways of linking them. A robust links structure is generally one that is not excessively complicated whilst facilitating the inter-changeability of modules – thereby maximizing the potential for modules to be re-used within different spreadsheets.

### Core Modules

When developing a modular spreadsheet model, there are a number of commonly-required modules which each serve a specific purpose that is fundamental to the operation and best practice nature of the surrounding workbook. These modules, which are referred to as ‘core’ modules, are listed in the following table:

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Series</td>
<td>• Contains the workbook time series assumptions and lookup tables.</td>
</tr>
<tr>
<td></td>
<td>• Facilitates the use of time series sheets.</td>
</tr>
<tr>
<td>Error Checks</td>
<td>• Contains the workbook error checks summary.</td>
</tr>
<tr>
<td>Sensitivity Checks</td>
<td>• Contains the workbook sensitivity checks summary.</td>
</tr>
<tr>
<td>Alert Checks</td>
<td>• Contains the workbook alert checks summary.</td>
</tr>
<tr>
<td>Formats &amp; Styles Key</td>
<td>• Contains the workbook formats &amp; styles key.</td>
</tr>
<tr>
<td>Sheet Naming Key</td>
<td>• Contains the workbook sheet naming key.</td>
</tr>
<tr>
<td>Range Naming Key</td>
<td>• Contains the workbook range naming key.</td>
</tr>
</tbody>
</table>

All of these modules are similar in that only one of each can be inserted into a single workbook. This is because they each serve a very specific purpose in the workbook – e.g. a time series module is required in order to undertake time series analysis, whilst the checks modules are required to implement best practice checks systems in a workbook.
**Time Series Module**

As discussed in the Time Series spreadsheet modeling area of the Best Practice Spreadsheet Modeling Standards, best practice time series models contain a single set of time series assumptions which are referenced by the period titles within all time series sheets in the underlying model. For modular spreadsheet development purposes, this time series content is contained exclusively within a single time series module, as shown below:

![Time Series Module - Composition](image)

This module composition diagram does not show the time series sheets in the underlying spreadsheet, which each contain period titles with calculations based on the time series assumptions and lookups module components within this time series module. However, it does demonstrate how the time series infrastructure within a time series module is allowed for when ‘modularizing’ a spreadsheet model.

The time series module is unique in a modular spreadsheet because without it the spreadsheet cannot contain time series sheets and, without time series sheets, many of the modules within the spreadsheet cannot be inserted. Hence, the time series module sits at the core of a modular time series model, and effectively governs the viability of the modules which can be inserted into it – e.g. an annual revenue module will not have been designed to be inserted into a time series model with only monthly time series sheets. For this reason, the time series module should be the first module inserted into a modular time series model, and care should be taken when adding modules to the spreadsheet that the modules being added are compatible with the module components within the time series module.

The image below shows how a time series module might appear within a modular spreadsheet model alongside other modules which undertake time series analysis:

![Time Series Module in a Modular Spreadsheet Model](image)
Checks Modules

The Best Practice Spreadsheet Modeling Standards require that the outcome of every error, sensitivity and alert check in a workbook be displayed in dedicated and separate error, sensitivity and alert check summaries respectively. In a modular development environment, these summaries are contained within separate error, sensitivity and alert check modules – each containing a single outputs module component as shown below:

The image below shows how these checks modules might appear within a modular spreadsheet model, located within a dedicated checks module area:

For a detailed discussion and commentary regarding the best practice modeling of checks see the Checks spreadsheet modeling area of the Best Practice Spreadsheet Modeling Standards.
**Keys Modules**

The Best Practice Spreadsheet Modeling Standards recommend that keys be included within a best practice model to communicate to model users and other model developers the formats and styles, sheet naming and range naming conventions used when developing the spreadsheet. In a modular development environment, these keys are contained within separate formats & styles key, sheet naming key and range naming key modules – each which contains a single outputs module component as shown below:

![Keys Modules - Composition](image)

The image below shows how these keys modules might appear within a modular spreadsheet model, located within a dedicated keys module area:

![Keys Modules in a Modular Spreadsheet Model](image)

For a detailed discussion and commentary regarding the inclusion of keys within best practice models see the Formats & Styles and Naming Principles spreadsheet modeling areas of the Best Practice Spreadsheet Modeling Standards.
Multiple Workbooks

The Best Practice Spreadsheet Modeling Standards require that all links between workbooks are made via dedicated and separate import and export sheets. In a modular development environment, analysis is undertaken on a module level rather than a sheet level and, as a result, links between workbooks are made via related import and export modules rather than import and export sheets.

The links between related import and export modules are highlighted by the red arrows in the below diagram. This diagram shows how import and export modules could be used to facilitate the exporting of revenue from two workbooks (i.e. Workbook 2.xlsb and Workbook 3.xlsb) into another workbook (Workbook 1.xlsb):

In this example, export modules are required within both workbooks from which revenue is being exported, and corresponding import modules are required for each of these export modules within Workbook 1. Note also that the time series module within Workbook 1 is exported into both Workbook 2 and Workbook 3, which each therefore have corresponding time series import modules, thereby ensuring the consistency of the time series components within all related workbooks.
Import and export modules, like the import and export sheets that house their module components, need only contain outputs module components because their sole purpose is to create formula links between workbooks. The composition of the export module in Workbook 2, and its related import module in Workbook 1, is shown below:

The content within the outputs module components of related export and import modules should be identical in layout and structure, with the only difference being the fact that the import module component ranges contain nothing but pure formula links to their corresponding export module component ranges.

For a detailed discussion and commentary regarding the use of multiple workbooks see the Multiple Workbooks spreadsheet modeling area of the Best Practice Spreadsheet Modeling Standards.

**Benefits**

Many benefits flow from the adoption of a modular spreadsheet development process, and these benefits increase with the size and level of detail of the spreadsheet model being developed.

The benefits of modular spreadsheet development include:

- Clarity of workbook structure;
- Significantly reduced model build time;
- Quarantining of the risk of errors to within each module;
- Reduced model audit costs;
- Facilitation of multi-tasking – i.e. by allowing the modules used in a single spreadsheet model to be developed simultaneously by different model developers;
- Efficiencies from module re-use preventing model redundancy;
- Quick and easy sharing of valuable functionality in complex modules; and
- Resulting improvements in spreadsheet model quality and sophistication.

Much of the perceived complexity surrounding spreadsheet models results from a failure to adequately compartmentalize the content. This often results in a disorganized spider web of analysis, lacking a clear structure or flow of information. The key to the successful modular development of spreadsheet models involves understanding how the model can be broken down into manageable, controllable pieces of analysis which can be independently developed, modified and/or removed from the model.
Practical Implementation

Modular spreadsheet development is a concept which can be applied to the development of any spreadsheet model within any spreadsheet application. The key to the successful practical implementation of module spreadsheet development is in the scoping and planning stages of the model development process. In this regard, the model developer should consider the many implications of the manner in which the spreadsheet is modularized by asking questions such as:

- What are the most logical and informative module areas that can be used to group the modules within the spreadsheet?
- Which modules can be used from prior modular spreadsheets, both with and without editing being required, and which modules will need to be built from scratch?
- What division of module areas and modules will best survive the ongoing development and maintenance of the spreadsheet over time?

Additionally, the model developer should consider bpmModules. bpmModules is a best practice, Microsoft Excel-based software program that facilitates the rapid development, maintenance and sharing of modules within Microsoft Excel spreadsheets. bpmModules prevents the need to manually copy, cut and paste module components within and between modular spreadsheets by automating the insertion, duplication and maintenance of modules. It also contains a large range of modular spreadsheet management and analysis tools. For more information regarding bpmModules and other best practice modeling resources, visit the Best Practice Modeling website at [www.bestpracticemodeling.com](http://www.bestpracticemodeling.com).
Other Best Practice Modelling publications...

- **Standards**
  - Tabulated Listing

- **Standards**
  - Commentary & Examples

- **bpmToolbox**
  - User Guide

- **bpmModules**
  - User Guide

- **Business Planning**
  - Modular Development

- **Time Series Analysis**
  - Fundamentals

www.bestpracticemodeling.com