MYOB Exo Business White Paper

BOMs and Stock Code Enhancements

Last modified: 27 April 2017

Contents

Introduction 1

Bills of Materials and Update Item Overview 1

Bills of Materials 1

Linked Stock Codes 2

Add-on Modules 3

Job Costing 3

Sales Order Workflow 3

Bill of Materials Changes 3

Setup 4

BOM Types 5

Pricing Mode 5

Order Templates 6

Template Pricing 7

Automatic Order Templates 7

Stock Codes 8

Hidden Cost and Sell 8

Line Types 9

StockTrans 9

New Document Fields 10

New Stock Transaction Fields 12

BOM Quantities on Invoices and Sales Orders 13

Credit Notes and BOMs 15

Crediting Kits and Templates 15

Crediting Builds 16

Stock Adjustment Method 16

Works Order Method 16

Editing BOMs within Documents 17

General Ledger Postings for BOMs 17

Postings for Kits 17

Postings for Builds 17

Postings for Order Templates 18

Introduction

Starting from version 6.18x, EXO Business has been significantly enhanced to rationalise and improve consistency in stock code reporting. Users who use Bill of Materials (BOM) functionality extensively will need to consider these changes carefully, and adjust business forms to take advantage of the new functionality. In particular there are two cases where the stock codes keyed (sometimes referred to as a PLU or Price Lookup Unit) are not necessarily the true stock keeping codes (also known as a SKU or Stock Keeping Unit). The two generic cases in EXO Business are:

- Bill of Materials
- Update Items

This change is achieved by providing all tables containing stock code related transaction lines with two stock code fields. (Prior to version 6.18x this was only partially implemented and only in Sales Orders in association with Update Items).

These tables are also populated with line types to indicate the type of line indication the nature of its origin. This can also signify to Clarity reporting the intent to hide lines on business documents. The previous practice of not inserting lines that are intended to be hidden has been retired, as Clarity provides conditional presentation abilities not previously present in the older style FMT reporting.

Bills of Materials and Update Item Overview

Bills of Materials

Bills of Materials (BOMs) comprise of a list of component stock items. They also have an output code. In some cases this is a lookup stock item and functions purely as a placeholder. The default output code for this is provided as the lookup stock item DEF BOM OUTPUT

BOMs are used to describe stock utilised in three ways:

1. As a kit, where the parent code is a shortcut to a list of items sold together. This is also sometimes referred to by terms such as list, bundle or collection. Variations include whether the price charged is set per total or independently per component. Stock transactions are written for each of the component lines as normal transactions (TRANSTYPE = 0 for Debtors or TRANSTYPE = 1 for Creditors) and contain the trading partners detail plus cost and selling prices as appropriate. Sales and purchases are therefore analysed by component in the same way as ordinary sales or purchase lines that are not part of a kit. The profit is recognised against the component.
2. During manufacture, within a discrete manufacturing process such as a Works Order or using the Bill of Materials Process or Batch Process functions. These consume the components (as negative stock adjustments, TRANSTYPE = 3) and place the designated output item into stock (as an inwards goods transaction TRANSTYPE = 1)

3. Embedded within another document type as a Works Order (e.g. inside a sales order or invoice). In this case a manufacturing process occurs to produce an output item that is then immediately sold or purchased as part of a single process. The kit header line of the document generates two stock transactions - to create the inwards goods (TRANSTYPE =1) and then sell the item (TRANSTYPE =0). The profit is recognised against the Output stock item. The DR_INVLINES for the components have zero cost and selling prices. (See also HIDDEN_COST and HIDDEN_SELL fields later in this document).

From the perspective of the stock ledger, options 2 and 3 above are the same. BOMs may therefore be one of two basic types:

- **Kit** – components are sold, the header is a shortcut code to a *collection* of components
- **Build** – components are consumed in a *transformation* to make an output item code
- In the BOM setup screen there is also a third type:
- **Order Template** – This is not a true Bill of Materials but a simple product list for use with sales orders only. The BOM code becomes a shortcut to the product list, to save keying each product into the order. Order Templates are discussed in a separate section later in this document.

Linked Stock Codes

Linked stock codes could be thought of as a form of mini BOM with only one component. These are used to sell a single stock item in different scaled packages, likely with differing pricing than the multiple of the SKU. An example of this would be 6 packs and dozens of bottles of beverages. Update items must be of type **L**ookup, and point at a shared SKU (e.g. bottles) of type **S**tocked. The target code (SKU) and ratio are defined on the stock item setup.

Add-on Modules

Both Job Costing and Sales Order Workflow have variations on these themes that allow alternate and/or grouped sets of components to be substituted or added within an instance of a BOM. Both of these support forms of "made to customer order", however their BOM handling is complementary and may form the basic decision as to which package is most suitable for any particular business.

Job Costing

Job Costing supports only Kits (collections) and is therefore more suited to a service oriented assembly process. Fewer constraints apply to the addition, substitution and removal of components processes within an instance of a kit. The components are treated as a flat list with no grouping within the kit instance.

Sales Order Workflow

Sales Order Workflow supports only Builds. The basic BOM is extended with options which provide two additional levels of grouping / substitution control within the BOM. This provides the following presentation structure:

BOM Header > Option Group > Option Type > Stock codes

All permissible alternatives must be defined within the BOM setup. One and only one Option type must be selected from within each option group. Casual component addition or substitution is not permitted. SOW therefore is more appropriate to a customised assembly type of business.

Bill of Materials Changes

The following changes are available for version 6.18x and later.

Setup

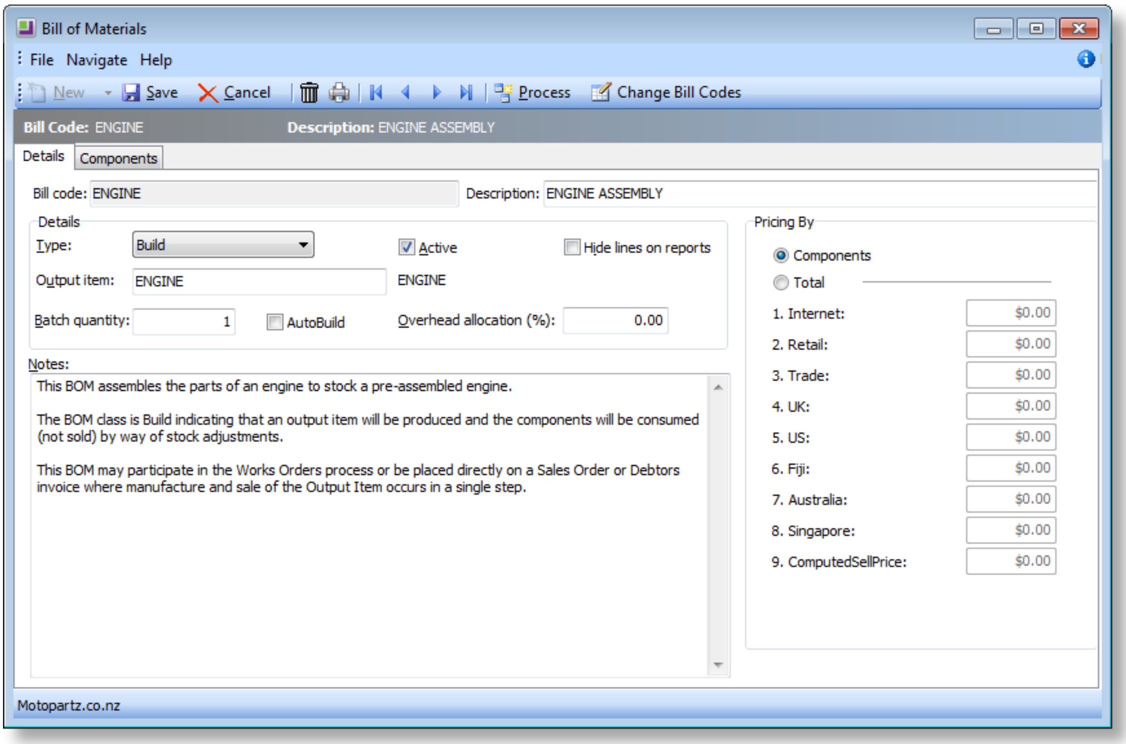
The original BOM set-up screen had two parameters of particular note:

- Pricing Mode (two options: Price by Total or Price by Components)
- Hide lines on invoices and orders

These provided four combinations or types of BOMs. In particular the **Hide Lines** property used with the **Price by Total** option did not write DR_INVLINES rows for the component, but generated STOCK_TRANS by referencing the kit header back to the BOM definition using BILLOMAT_LINES.

From 6.18x the main BOM storage in documents has been simplified from four types to two main types. All other factors are now handled via presentation.

As well as adding the new **Type** field to the BOM setup screen, the layout has also been rationalised to more clearly show the relationship of the properties.



Bill Code: ENGINE Description: ENGINE ASSEMBLY

Bill code: ENGINE Description: ENGINE ASSEMBLY

Details

Type: Build ☑ Active □ Hide lines on reports

Output item: ENGINE ENGINE

Batch quantity: 1 □ AutoBuild Overhead allocation (%): 0.00

Notes:

This BOM assembles the parts of an engine to stock a pre-assembled engine.

The BOM class is Build indicating that an output item will be produced and the components will be consumed (not sold) by way of stock adjustments.

This BOM may participate in the Works Orders process or be placed directly on a Sales Order or Debtors invoice where manufacture and sale of the Output Item occurs in a single step.

Pricing By

● Components

○ Total

1. Internet:	$0.00
2. Retail:	$0.00
3. Trade:	$0.00
4. UK:	$0.00
5. US:	$0.00
6. Fiji:	$0.00
7. Australia:	$0.00
8. Singapore:	$0.00
9. ComputedSellPrice:	$0.00

The new **Type** property designates the BOM as a Kit, Build or Order Template. The **Hide lines on invoices and orders** option has been renamed to **Hide lines on reports**, and is provided for use with Clarity to control presentation of Clarity documents.

BOM Types

As already discussed, EXO Business standardises the main BOMs into two types:

- Kits, where the components are being sold. The header is a way to identify the collection of components.
- Builds, where the output item is being sold. The components are consumed internally when the output item on the header is manipulated in the Works order process, or sold on an invoice or sales order.

Plus:

- Order Templates, where the list of items is being used on a Sales Order. The list of items on a Sales Order are treated as individual non-grouped lines.

Note: The EXO Business Works Order module, BOM and Batch BOM processes will only process Build-type BOMs. Kit-type BOMs are not valid structures for Works Order processing.

Pricing Mode

The method of pricing is controlled by the **Pricing Mode** property.

- BOMs that are **Priced by Component** derive their prices (and costs) from the STOCK_ITEMS for each component. Dynamic based on component sell prices.
- BOMs that are **Priced by Total**, the final sell price of the BOM is explicitly defined, and fixed independently of the component prices.

New changes introduced in 6.18x include:

- The existing **Pricing Mode** property has been demoted to only control the initial method of pricing when loading a BOM to a sales document.
- Previously the **Pricing Mode** also influenced how the data was stored in the database. The component prices and output item prices stay in sync, regardless of the setting of the Pricing Mode. (See “Hidden Cost and Sell” on page 8, which provides a mechanism to allow reporting analysis by either the **Output item**, or the components.).
- The editing of a BOM is now controlled by its BOM type and not by its pricing method Also note the ability to edit is controlled by the profile setting **Enforce bill of materials header and line constraints**. See “Editing BOMs within Documents”on page 17.

Note: Sales Order Workflow has additional mechanisms that are not discussed here.

Order Templates

Order Templates only may be used in Sales Orders and Debtors Invoices. They are neither Kits nor Builds; they are simply a list of products. When added to a Sales Order or Debtors Invoice:

- No BOM header line is inserted in the document.
- The usual dialog requesting number of kits does not occur, therefore no scaling of the BOM is supported.
- The component lines are inserted as ordinary lines as if individually keyed. Following insertion there is no grouping of lines; they become independent lines.

An Order Template can be created with zero quantities and used to prompt the user with a product list that the customer may wish to buy. This could also be extended to a structured product list where similar items are frequently purchased together. An example of this may be to create a template for a certain type of pen containing the stock codes of the available colours for that pen. Lines that remain zero after keying are therefore purged from the order.

As the Order Template document lines are created as ordinary lines, within the document line the following fields are set:

- LINETYPE is set to 0
- BOMTYPE is set to the letter O
- KITCODE is not populated with BILLCODE

Note: The output code of an Order Template must be a lookup item. It is not utilised unless priced by template (see Template pricing), therefore can be left as the default code DEF BOM OUTPUT

To further emphasise that Order Templates are not Kits or Builds:

- The Components tab is renamed Products.
- The **Cost Type** property at the top of the Products tab does not appear.
- The Products (Components) grid shows Sell Price 1 and its extension, rather than showing Unit and extended Total Costs.
- The **Output item** is relabelled **Template item**.

Note: Ordinary lines with zero quantity are automatically deleted on saving the order.

Template Pricing

By default Order Templates are Priced by Product. The individual product lines are priced according to the rules for ordinary lines. This concept is similar to a Kit or Build that is priced by component, however no BOM header line is inserted into the document.

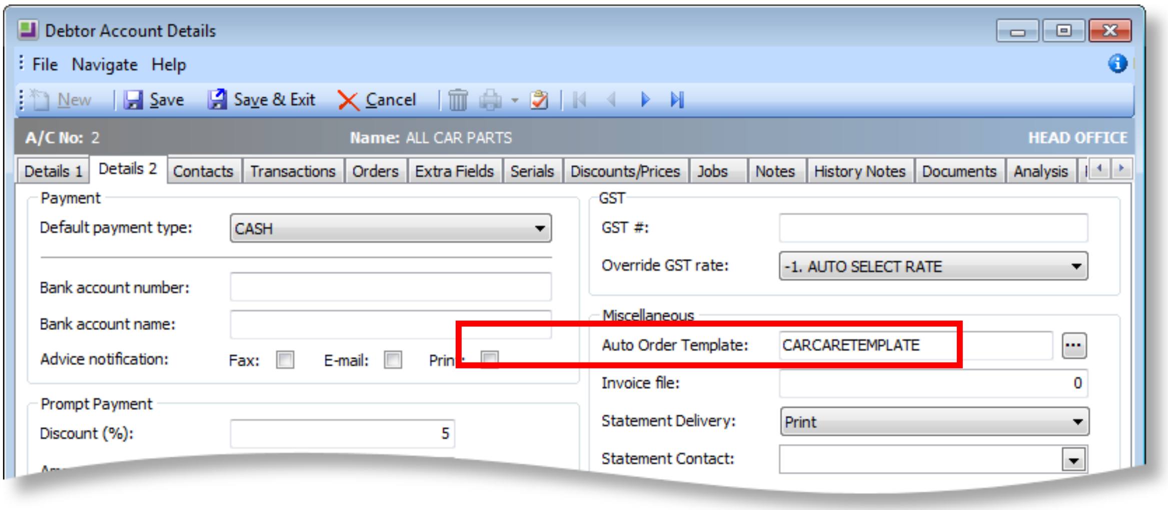
An Order Template BOM may also be Priced by Template. This effectively sets an overall price for the complete Order Template. This concept is similar to a Kit or Build that is Priced by Total, however an Order Template that is Priced by Template has the following features:

- An additional ordinary line is inserted for the **Template Item** (Output item). The price on this line is overridden with the appropriate price-level total price from the BOM setup.
- Product lines from the Product tab are inserted with the price overridden to zero
- All lines are marked as price overridden (and therefore change colour accordingly, as per normal price override operation on sales order entry).

Note: When changing a stock code for a template line within a document, the usual price override rules are suppressed and the line reverts to standard pricing. Changing a stock code in such circumstances is therefore treated as a true substitute item rather than a stock code correction.

Automatic Order Templates

An **Automatic Order Template** can be specified for a Debtor on the Details 2 tab of the Debtor Account Details window. When a new Sales Order (but not invoice) is created for the Debtor, the Order Template is automatically loaded onto the Sales Order.



A/C No: 2 Name: ALL CAR PARTS HEAD OFFICE

Payment

Default payment type: CASH

Bank account number:

Bank account name:

Advice notification: Fax: □ E-mail: □ Prin □

Prompt Payment

Discount (%): 5

GST

GST #:

Override GST rate: -1. AUTO SELECT RATE

Miscellaneous

Auto Order Template: CARCARETEMPLATE

Invoice file: 0

Statement Delivery: Print

Statement Contact:

Stock Codes

From version 6.18x onwards all tables with document lines relating to stock items contain three fields, which are always populated. This is a natural extension of the additional fields:

- STOCKCODE = the **Keyed** code or PLU.
  - For BOM header lines this will be of the form <dot>BILLCODE.
  - For Update items this will be the Lookup item (e.g. 6pack).
- LINKED_STOCKCODE = the **True** stock keeping code or SKU.
  - For BOM header lines this will be the Output code.
  - For Update items this will be the Stocked item code (e.g. bottles).
- LINKED_QUANTITY = the **Ratio** of STOCKCODE to LINKED_STOCKCODE.
  - This is now a ratio so that multiple quantity fields within the document row can be multiplied by this ratio. In previous versions it was only used for update items and was an extended quantity.
- For BOM headers and ordinary lines this will always be 1.
  - For component lines this will be the component quantity specified the BOM.
  - For update items this is the ratio of STOCKCODE to LINKED_STOCKCODE.

Note: Multi-level BOMs are not supported by any of the core EXO Business products. Complications of iteration where components point to other components therefore do not occur. All relevant stock levels and stock commitments can therefore be performed by a flat file query against the LINKED_STOCKCODE in each of the participating tables. This simplifies the STOCK_IN_LOC stored procedure.

Hidden Cost and Sell

The table DR_INVLINES now has fields for HIDDEN_COST and HIDDEN_SELL. These are only used for BOMs. In all other cases they will have zero values. These provide a way of “double dipping’, i.e. reporting profit at either a kit header or component level.

In the case of a Kit BOM where the actual components are sold, the component lines carry the real cost and selling prices. The kit header would then contain the total “rolled up” cost and sell prices in these two HIDDEN_ fields. This is also applicable in Job Costing.

In the case of a Build BOM (manufactured), the kit header represents the true cost and sell prices, as it is the output item that is sold and represents profit. The components are consumed as part of the manufacturing process and may exist on the invoice as a form of documentation of the contents of the BOM (almost a comment line). In this case the component lines use the HIDDEN_ pair of fields to carry the breakdown of their contribution to the total “real” cost and sell on the header. These can then be used as a mini “job analysis by invoice”. This is also applicable to Sales Order Workflow.

The presentation and edit-ability of prices and margins within any given function may allow modification of either or both totals or lines. Regardless of presentation, the storage of the HIDDEN data and resulting STOCK_TRANS & GLTRANS will depend on the BOM type.

> **Note:** Prior to version 6.18x the **HIDDEN_COST** and **HIDDEN_SELL** fields only existed in **DR_INVLINES** for Sales Order Workflow databases.

Line Types

In versions 6.14x through 6.16x some document lines had a LINETYPE field to designate the type of line particularly for reporting purposes. These were single digits in the range 0–4 and mainly differentiated kit headers lines and comment lines from ordinary lines. Historic lines were left with values of -1.

In 6.18x the coding structure of these was changed as follows:

- The hide and show attribute of component lines has been moved to a separate property field therefore all component lines become LINETYPE =2
- An additional LINETYPE=5 has been added to indicate that the line has a General Ledger code but no LINKED_STOCKCODE
- The 'hide lines' attribute of the BOM definition also now has no impact on processing, only reporting. Component lines are now always written to the appropriate document tables, regardless of this setting.

LINETYPE has the following values:

- -1 = Undefined (historic data)
- 1 = BOM Header
- 2 = Component Line
- 3 = No longer used (historically a hidden component)
- 4 = Narrative line
- 5 = General ledger code

StockTrans

EXO Business version 6.18x tightened the links in STOCK_TRANS for stock movements for Sales Orders.

The LINE_SEQNO links to the SALESORD_LINES SEQNO field, and populates the FROM_LEDGER field with an "o" to signify the source was a sales order. Previously when Sales Orders were supplied and invoiced simultaneously the STOCK_TRANS FROM_LEDGER was written as type "d".

New Document Fields

The following fields were added to document line tables (e.g. SALEORD_LINES) in version 6.18x:

- SHOWLINE CHAR(1) DEFAULT 'Y'
- BOMPRICING CHAR(1) DEFAULT 'N'
- BOMTYPE CHAR(1) DEFAULT 'N'
- LINKEDSTATUS CHAR(1) DEFAULT 'N'
- KITSEQNO INT

SHOWLINE

The SHOWLINE property can be used by Clarity to determine if the line should be printed. Setting this to 'N' can indicate to Clarity that the line should be suppressed at the discretion of the individual report or document. Component lines of BOMs set to hide lines will create document lines with this property set to 'N'. This line property is derived from the existing field:

BILLOMAT_HDR.HIDE_LINES

Possible values are:

- Y = Yes
- N = No

This property has no influence on data processing procedures within the Exo Business product.

BOMPRICING

This property indicates the pricing mechanism used for the BOM. It is independent of other BOM properties and purely controls the pricing method during the initial load of the BOM into the transaction. The data storage method is no longer controlled by this property. This line property is derived from the existing field:

BILLOMAT_HDR.PRICING_MODE

Possible values are:

- N = Not Applicable {Not a BOM}
- T = Priced by Total
- C = Priced by Component

BOMTYPE

This determines if the BOM is a Kit {Collection} or Build {Transformation}, and is copied from the corresponding new field within the BOM definition:

BILLOMAT_HDR.BOMTYPE

Possible values are:

- N = Not Applicable (not a BOM)
- K = Kit (collection)
- B = Build (transformation)
- O = Order template (shopping list)

Note: BOMTYPE should not be confused with the existing field KIT_TYPE, which differentiates special types of BOMs.

LINKEDSTATUS

The LINKEDSTATUS indicates whether the current stock item is Stocked or a Lookup item.

- For a standard (true) stock keeping code, or a component line of a BOM, the LINKSTATUS is copied from the stock item table.
- In the case of a BOM header, this is derived from the output code.
- For a linked stock code, this is determined from the item being linked to, (which for a linked stock code must always be Lookup). When processed, two stock transactions are generated; a Lookup item for the keyed code (PLU), and a Stocked item for the physical stock item (SKU) linked to.

Values for LINKEDSTATUS can be:

- N = Not Applicable {not a stock item – e.g. a GL only line}
- S = Stocked
- L = Lookup

KITSEQNO

All the lines (including header) of an instance of a Kit or Build BOM within a document are now allocated a sequence number that is globally unique within a database (as opposed to only unique within a document or ledger). This is stored in the field KITSEQNO, and has the following uses:

- Allows grids to keep kit lines together.
- In some functions allows the grid to hide or show component lines.
- Allows Clarity reporting at a kit level especially when multiple kit instances exist within one document (i.e. the compound string HDR_SEQNO + KITCODE may not represent a unique instance of a kit whereas KITSEQNO always does).
- Provides a unique key that can link to other custom tables to carry additional information global to the kit instance.

KITCODE

The existing KITCODE field contains the BILLCODE on all lines including components and header. Analysis by LINKED_STOCKCODE within KITCODE is therefore now possible (including KITCODE = null for ordinary lines).

> **Note:** Prior to 6.18x, only Sales Order Workflow added KITCODE to sales order lines, however this has now been added by Exo Business to various tables and is not limited to Sales Order Workflow use.
>
> **Note:** Only Sales Order Workflow forms a relationship between individual manufactured serial numbers and the serial numbers of their components. This is another product differentiator for the SOW product.

New Stock Transaction Fields

The stock transactions table STOCK_TRANS has the fields KITCODE and KITSEQNO as outlined above, plus two additional fields as follows.

PLU (Price Lookup Unit)

The field STOCKCODE in this table contains the true stock code (SKU) as opposed to the keyed code (PLU) that occurs in other tables such as DR_INVLINES. A field STOCK_TRANS.PLU has been added to indicate the keyed code.

To illustrate this, consider the following relationships:

DRINVLINES.LINKED_STOCKCODE = STOCKTRANS.STOCKCODE

DR_INVLINES.STOCKCODE = STOCKTRANS.PLU

POST_TO_GL

To allow better reconciliation between the stock transaction STOCK_TRANS and GL transaction GLTRANS tables, a new field POST_TO_GL has been added to the STOCK_TRANS table. This POST_TO_GL field indicates if posting is expected (independent of whether it has been posted or not). It is set at the time the row is created, and is not subsequently amended.

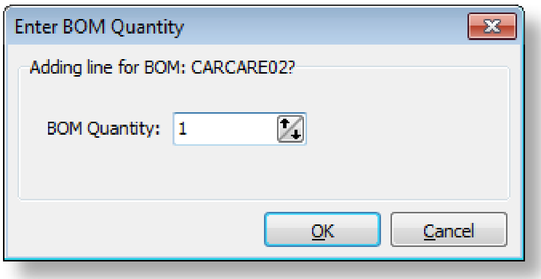
The actual posting process is still controlled by the field STOCK_TRANS.GLPOSTED.

Previously when reconciling, it was not possible to distinguish between a stock transaction that did not require posting, versus one which had been posted, when STOCK_TRANS GLPOSTED='Y'. It is therefore now possible to use the new POST_TO_GL field to determine which stock transactions are expected to have corresponding entries in the GL.

BOM Quantities on Invoices and Sales Orders

In previous versions, when a BOM (Kit or Build) was added to a Debtors Invoice or Sales Order, the quantity on the BOM header was locked to 1. When the BILLCODE is now selected an additional dialog appears to allow the quantity to be set.

BOM priced by total asks you to confirm the price when you add a price by total kit.

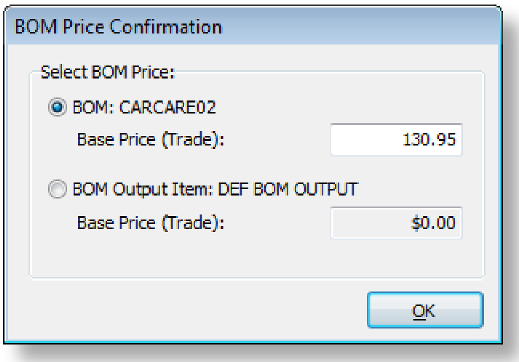


Adding line for BOM: CARCARE02?

BOM Quantity: 1

When **OK** is pressed the document will be populated with header and component lines. The BOM header will have this quantity. The quantities on the component lines will be scaled by the BOM quantity specified.

With a price by total kit the price entered here is compared to the normal price of this kit, and the difference is spread pro-rata against the components (based on the ‘best price’ sell price for each component)



Select BOM Price:

● BOM: CARCARE02

Base Price (Trade): 130.95

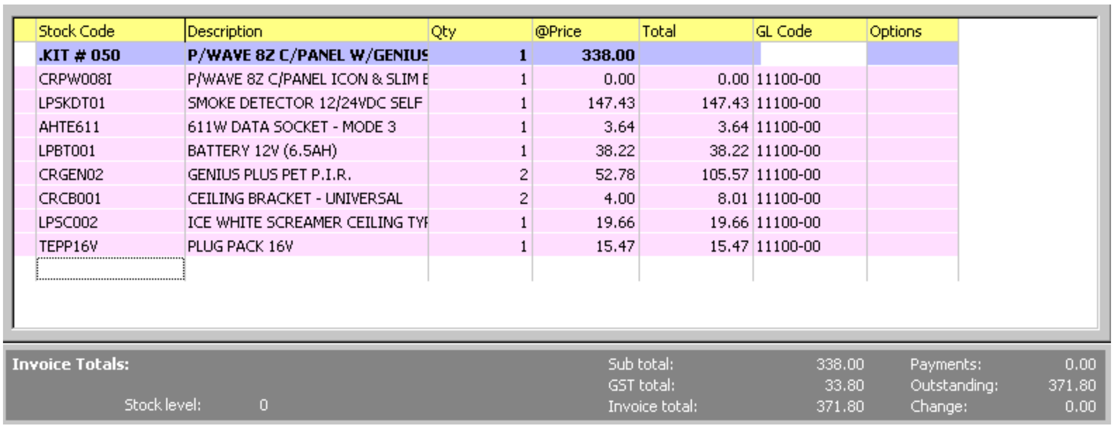
○ BOM Output Item: DEF BOM OUTPUT

Base Price (Trade): $0.00

Sometimes the sum of the tax on the component lines has a rounding that wouldn't be present if the invoice contained a single line (BOM header). The Total GST on the invoice must always add up to the total tax on the lines. To prevent this we post any GST rounding and any total rounding to the first component line on the kit with a non-zero quantity, which will bear the rounding for the kit when spreading the kit total across the lines as the kit is entered.

Example transaction:

The client wants to see the one line $338 for the Kit (priced by total) at 10% GST = $33.80. But because in reality the GST is accounted for at line level the rounding makes it $33.79.



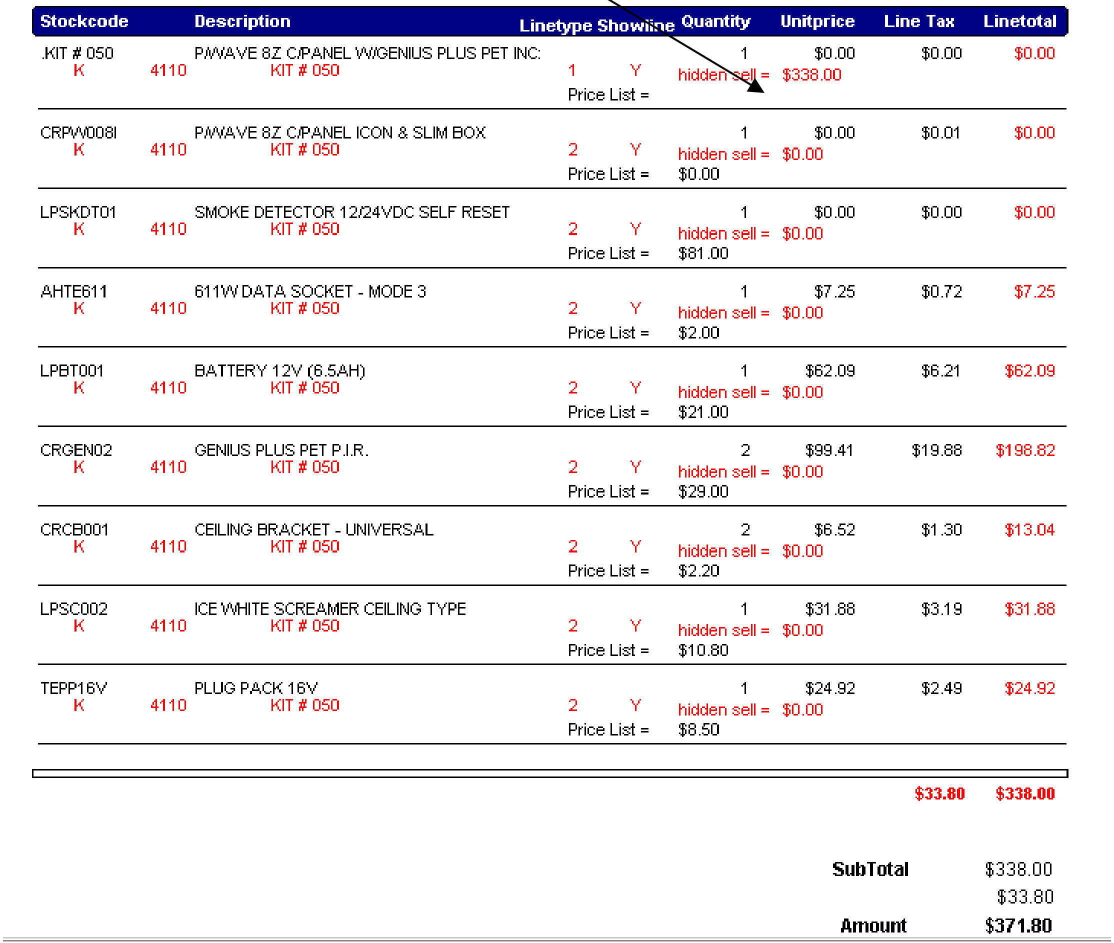
Stock Code	Description	Qty	@Price	Total	GL Code	Options
.KIT # 050	**P/WAVE 8Z C/PANEL W/GENIUS**	**1**	**338.00**			
CRPW008I	P/WAVE 8Z C/PANEL ICON & SLIM E	1	0.00	0.00	11100-00	
LPSKDT01	SMOKE DETECTOR 12/24VDC SELF	1	147.43	147.43	11100-00	
AHTE611	611W DATA SOCKET - MODE 3	1	3.64	3.64	11100-00	
LPBT001	BATTERY 12V (6.5AH)	1	38.22	38.22	11100-00	
CRGEN02	GENIUS PLUS PET P.I.R.	2	52.78	105.57	11100-00	
CRCB001	CEILING BRACKET - UNIVERSAL	2	4.00	8.01	11100-00	
LPSC002	ICE WHITE SCREAMER CEILING TYI	1	19.66	19.66	11100-00	
TEPP16V	PLUG PACK 16V	1	15.47	15.47	11100-00	

Invoice Totals:

		Sub total:	338.00	Payments:	0.00
		GST total:	33.80	Outstanding:	371.80
Stock level:	0	Invoice total:	371.80	Change:	0.00

So any price or tax rounding is placed on the first non-zero quantity component line.

Hence the Tax of .01 cent on this line below.



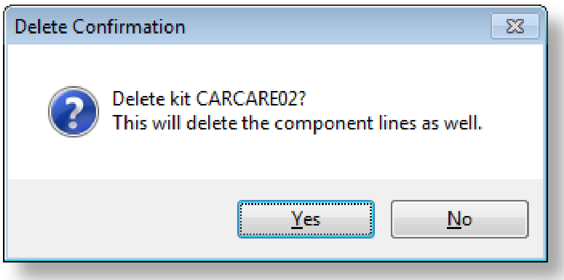
Stockcode	Description	Linetype	Showline	Quantity	Unitprice	Line Tax	Linetotal
.KIT # 050 K 4110	P/WAVE 8Z C/PANEL W/GENIUS PLUS PET INC: KIT # 050	1 Price List =	Y	1 hidden sell =	$0.00 $338.00	$0.00	$0.00
CRPW008I K 4110	P/WAVE 8Z C/PANEL ICON & SLIM BOX KIT # 050	2 Price List =	Y	1 hidden sell = $0.00	$0.00 $0.00	$0.01	$0.00
LPSKDT01 K 4110	SMOKE DETECTOR 12/24VDC SELF RESET KIT # 050	2 Price List =	Y	1 hidden sell = $81.00	$0.00 $0.00	$0.00	$0.00
AHTE611 K 4110	611W DATA SOCKET - MODE 3 KIT # 050	2 Price List =	Y	1 hidden sell = $2.00	$7.25 $0.00	$0.72	$7.25
LPBT001 K 4110	BATTERY 12V (6.5AH) KIT # 050	2 Price List =	Y	1 hidden sell = $21.00	$62.09 $0.00	$6.21	$62.09
CRGEN02 K 4110	GENIUS PLUS PET P.I.R. KIT # 050	2 Price List =	Y	2 hidden sell = $29.00	$99.41 $0.00	$19.88	$198.82
CRCB001 K 4110	CEILING BRACKET - UNIVERSAL KIT # 050	2 Price List =	Y	2 hidden sell = $2.20	$6.52 $0.00	$1.30	$13.04
LPSC002 K 4110	ICE WHITE SCREAMER CEILING TYPE KIT # 050	2 Price List =	Y	1 hidden sell = $10.80	$31.88 $0.00	$3.19	$31.88
TEPP16V K 4110	PLUG PACK 16V KIT # 050	2 Price List =	Y	1 hidden sell = $8.50	$24.92 $0.00	$2.49	$24.92
						$33.80	**$338.00**

SubTotal	$338.00
	$33.80
Amount	**$371.80**

The Clarity Invoice form needs to be configured to display the view of the kit(s) that the customer wants to see, while EXO Business retains integrity of the component items prices. Clarity form above exposes the invoice logic shows for kits (good for debugging) and is provided in the Clarity Masters variations as **InvoiceBOM.CLF**.

Note: When adding a BOM to a document, the quantity may only be negative for kit-type BOMs. For Builds, credits are not permitted and it is expected that an ordinary line of the output stock code will be keyed instead. For more details see “Credit Notes and BOMs” on page 15.

There is no provision to subsequently amend the quantity or price on the BOM header line and recomputed the component quantities. If the BOM header quantity is incorrect then the entire instance of the BOM can be deleted and re-entered. Attempting to delete the BOM header causes the following message to appear:


Delete Confirmation

Delete kit CARCARE02?
This will delete the component lines as well.

Credit Notes and BOMs

Issuing a credit note for a BOM is not necessarily just a matter of reversing the original invoice and stock transactions; it depends on the BOM type. In previous versions the BOM type was not readily identifiable so auto-generated credits (by right-click) for invoices containing a BOM were prevented.

In addition, changes to the BOM properties since the invoice was created resulted in different transaction types being generated for the reversal.

From 6.18x the BOM type is stored in all lines of the invoice (including header and components). The origins of each line is known, and therefore can be dealt with accordingly when processing a credit note, without cross-reference to the BILLOMAT_HDR table or the BOM header line within the invoice.

Crediting Kits and Templates

With Kits, the component lines are equivalent to ordinary lines. The components are sold and there was no value-add process affecting the stock value prior to sale. The header is a placeholder and is equivalent to a comment line. With Order Templates, the lines are independent ordinary lines.

Kit and Order Template BOMs may therefore be reversed like any other sales order line.

Crediting Builds

With Builds, there is effectively a Works Order within the invoice that makes (or assembles) the output code into stock before selling it.

This process could include components that are lookup items. These lookup items constitute value-add to the stock on hand prior to the sale (e.g. labour). It would be incorrect to reverse these; in fact it may be necessary to add an additional cost associated with a "disassembly". Some Builds may also be physically unable to be disassembled (e.g. un-mixing paint).

For these reasons **EXO Business products do not support computer aided disassembly processes.**

Generating a credit note from an invoice containing a Build by using the right-click method will only create a single line on the credit note for that Build entry, ready to reverse the sale of the output item and return the output item only (not the components) to stock. Additionally this line will be of type 'ordinary', with a stock code determined by the output code on the original Build header of the invoice (i.e. from LINKED_STOCKCODE on the Build header line of the original invoice).

If the disassembly of the output code is required, then this is a separate business process from the customer credit and return. Disassembly may also be considered optional, as the output code could also be on-sold to another customer as an assembled item. Either Stock Adjustments or Works Orders could be used to achieve disassembly.

Stock Adjustment Method

Stock adjustment transactions can be manually determined and entered into the Stock Movements screen. It may be necessary to designate different stock locations for different components (e.g. seconds, rework, repackage).

Stock transactions (to lookup items) or GL transactions to compensate for a change in stock value may be needed, (e.g. written down stock items or additional costs of disassembly).

Works Order Method

A true Works Order could be created to "un-manufacture" the *output code*. This will usually require elimination of any lookup items (e.g. labour add) when the BOM is imported, and substitute or add lookup items for change in stock value as outlined previously.

Note: Note that this is not an option for special BOMs such as those used in Sales Order Workflow - the Stock Adjustments Method must be used.

Editing BOMs within Documents

There is a relationship between the header quantity/price and the quantities/prices of the components. This applies when adding a BOM to an invoice or sales order. By default the quantities and prices of the header and component lines are locked from editing to ensure the header/lines relationship. This is a new constraint in the 6.18x version of the software.

Note: As Order Templates are not true BOMs, but simply shopping lists, therefore the above default constraints to not apply to them.

It is envisaged that some users may wish to be able to edit the lines independently. A Company-level profile setting is available to facilitate this:

Enforce BOM header and line constraints – If enabled, the system enforces strict BOM header and line constraints; otherwise BOM lines are treated as individual lines. The default value is true (i.e. constrained).

Note: When editing either the header or lines, there is no facility to have the system roll-up or roll-down quantities/prices.

General Ledger Postings for BOMs

General Ledger postings vary depending on the BOM type.

Postings for Kits

For Kits the component lines are equivalent to ordinary sales lines. The header line does not post to GL and the output code is not referenced.

The GL accounts for Sales and Cost of Sales arise from the component stock item as per the rules for ordinary sales lines.

Postings for Builds

For Builds:

- Component lines that are stocked items are not posted to GL (they don't affect the value of stock).
- Component lines that are lookup items are posted to GL as adjusting the value of stock (i.e. are value-add). These are posted with the GL code for the Stock On Hand account from the GL control account setup.

- The BOM header generates two entries:
    - **For inwards goods** – to the GL purchases account set on the stock item for the output code, (i.e. receipt the manufactured item into stock). The cost and sell of the stock transaction is the sum of the costs of the components (including lookups) and therefore incorporates the contra-entry for added value represented by the lookup item posting.
    - **For the sale of the output item** – with GL accounts for the sales and cost of sales being taken from the stock item of the output code.

Postings for Order Templates

For GL posting, the system treats lines of an Order Template identically to ordinary lines and is oblivious to the fact that they were derived from a BOM.