FLEXX®

Version 7.0L0

Procedures Guide

Distribution Modules

Δ Databyte

FLEXX® 7.0L0 Procedures Guide - Distribution Modules

Table of Contents

Inventory Control

1   Inventory Control Enhancements

Numerous enhancements have been made to the Inventory Control function. Following are examples of the more significant changes, with a brief description of each. *Please refer to the FLEXX Inventory Control manual for a more detailed description of all the SKU Master fields.*

1.1 Inventory Form

Quantity values can now be up to 6 decimal places long. Values with more decimals can be entered but FLEXX will round them to 6 places.

Warehouse Freeze flag and *Freeze Date*:
- used in the Stock Counting process to freeze inventory during counting,
- SKU's with *Freeze* flag set can still be ordered, but the order cannot be released or shipped until the flag is reset.
- *Freeze Date* is set by the Stock Counting process;
- both the flag and date will be reset at the completion of the Stock Counting process *(See the Stock Counting Process description for more detail).*

1.2 Pricing Form

The SKU Master Pricing Form is also referred to as the SKU Price Table or the Price Matrix.

As with all forms, Price values can now be defined with up to 6 decimal places, and larger values will be rounded to 6 decimals.

New Price Types:

- Price Type "r" - **Repair Fee** " price that will be used by FLEXX when the SKU is entered on a R/W work order, and is then specified to be "Repaired in-house" *(refer to the Repair & Warranty manual for more details).*

- Price Type "o" – **Option Price** " is the price for this SKU if it is defined as an Option item on another SKU *(see Topic 1.7 below).*

Price Description - will default to the value specified on the Price Type table for the specific Type code (e.g. Type "d" is defined "Domestic Pricing"), but can be changed to any other value as desired. This is provided to better identify the meaning of the particular price entry.

Currency code is now a required field **(6.5L2)**. FLEXX will use the setting to determine if Currency Exchange calculation will be required at the time of order entry. If the defined code is different than the Company currency setting, FLEXX will use the Foreign Exchange table to calculate the effective price in that currency.

1.3 Vendor Form

Landed Cost Factors:

Factors can be designated as to their requirements, with codes –

- N = *Not Applicable* (landed cost voucher not required);
- Y = *Allowed* (if a value is defined, it will be used, but is not required);
- O = *Required* (a Landed Cost Voucher entry for this factor is required).

These flags are only used at the time the **Landed Cost Update** procedure is run, and FLEXX will provide messages when and if any factors or vouchers are missing. *(See **Landed Cost Procedure** description for more detail on the process)*

1.4 Attributes Form

Attributes Flags changed or added:

- BOM – can have the following settings:
    - N- Not a BOM
    - B – Explodable BOM – at order entry, will explode into its component SKU's.
    - C – Non-Explodable BOM – will not be exploded at order entry.
- Order – SKU can be ordered via PO
- Sell – SKU can be sold via OP
    - Order/Sell flags can be used to either sell off old stock (Order off, Sell on), or to delay selling new stock (Order on, Sell off).
- Shippable flag has been deleted (Version 6.5). It had no function in FLEXX so was removed to avoid confusion.
- Calc. Volume – calculate volume of SKU using *Volume* fields dimensions, in units defined in Application Control *'volume'* parameter.
- Recurring – definition is a Recurring SKU

- Edition – definition is an Edition SKU
    - Edition and Recurring flags are mutually exclusive; only one or the other is allowed on. Either flag set on will then also require the Edition Cycles or Recurring fields to be defined. *(See the Edition Functions description for more detail)*
    - Edition set on results in the Show Edition tab to be lit, and access to the Editions Table is possible.

- Divergence – to identify a SKU that allows sales order (Sell UOM) quantities to be **shipped** slightly 'divergent' from the Stocking UOM quantities without affecting the actual Shipped quantity. *(See the Inventory Control - SKU Divergence description for more detail).*

- Domestic Sale – SKU is restricted to domestic sales only (sold only in same country as 'Company' country)

- RWEquipment - define this as a R/W Equipment SKU. This also requires the Serialize flag as well as the Serial Type value "se". Each item of the SKU requires a Serial Number to be predefined, either at PO receiving or manually on the Serial Number Information table (from SKU Master, Additional button).

- RWReplacement - further identify this SKU as a R/W Replacement item, meaning it can be used as a replacement item when replacing another RWEquipment SKU that has been entered on a R/W Work Order *(refer to the Repair & Warranty manual for more details).*

Units fields:

- The SKU can be defined to be Stocked in a UOM different from the Sell UOM using the **Stock UOM** and **Sell UOM Type** fields. These values need to be defined in the Unit of Measure table with their conversion factor *(See the Order Processing – Sell Unit of Measure description for more detail).*

Weight – if a value is defined for Weight, the Tangible flag will also be required to be set on.

Quantity fields:

- Max is used to specify the maximum Sales order quantity for this SKU, only if the Application Control op variable *check_qty* is set to Y.

- Otherwise, Min and Max are for user reference only, and are not used in the Stock Ordering Process in Purchasing.

1.5 Miscellaneous Form

Fields have been added to the Miscellaneous form as well as the format has been changed somewhat.

- GL Acct. Segment Value – not currently used; will be implemented on later release.

- eCommerce Attributes – these fields are used to define how the SKU is to be used in the FLEXX eCommerce module.
    - ECommerce SKU – SKU is eligible for eCommerce sales order entry.
    - Display MFR Name – display the SKU manufacturer (Vendor) on the eCommerce Product display screen.

- SKU Category fields have been enhanced:
    - Category and Sub Category fields are now both Master Type Table defined values so need to be predefined before they can be used.
    - Sub Category2 is a free-form field
    - SKU Attribute1 and Attribute2 are also Master Type Table defined values so need to be predefined before they can be used.

- Default Warehouse fields have been added (as described below in topic *SKU Level Warehouse Specification*).

1.6 SKU Level Warehouse Specification

The SKU Level Default Warehouse Specification function allows the default warehouse to be defined at the SKU level, regardless of any other default warehouse settings. This is under the control of the Application Control Table as shown below.

Application Control setting:
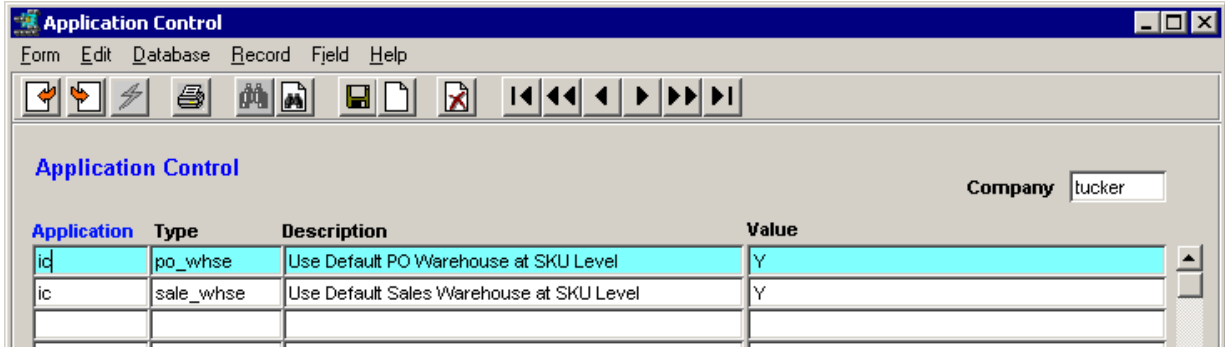
If = Y, use the Default Warehouse setting as defined on the SKU Miscellaneous form.

This setting overrides the default Warehouse defined on the Company/Division table, but only for the specific SKU. Each SKU can have a defined default, or if not defined, will use one of the other system defaults as described below.

Default warehouse settings can now be defined at the following levels:

- Company/Division Table – this is the default if no other default settings are defined;
- Application Control Table – this is the default if the Company/Division default is not defined (is null) and will be used if no other defaults are defined;
- SKU Table – this will take priority over any other default warehouse definition:
    - **Sales** – this warehouse will be used on all sales orders (OP, Quotes, WO's, Subscriptions) where this SKU is entered;
    - **Purchasing** – this warehouse will be used on all PO's where this SKU is entered.

1.7 Negative Quantity and Zero Cost Controls

It is possible to have FLEXX control whether or not to allow **negative inventory quantities** and/or **zero cost inventory values**. This is accomplished through the following Application Control settings:

Application	Type	Description	Value	Company
ic	negqty	Allow Negative Qty On Hand	Y (default) or N	default
ic	zerocost	Allow Zero Cost on items	Y (default) or N	default

- *'negqty'* is used to determine if a **negative On Hand** value is to be allowed for Inventory SKU's on the inventory table. FLEXX will report error message "Could not update inventory quantity" on the transaction that would result in a negative value to be generated (e.g. sales order overshipped).

- *'zerocost'* is used to determine if a **zero Average Cost** value is allowed for Inventory SKU's in Inventory Control. If N (no), then on Order Release, FLEXX will report message "System does not allow sku average cost to be zero cost. Please fix the sku setup on inventory". FLEXX will allow the order to be released and shipped. However, it will fail at Invoice generation with the same error message, and no invoice will be created. An average cost will need to be entered to proceed.

1.8 Inventory Generator

The FLEXX Inventory Generator Process provides a facility to generate additional warehouse entries on the SKU Inventory Table for the selected SKU's. This is useful when a new warehouse has been defined and existing SKU's also need to be assigned to this warehouse.

Either a specific SKU can be selected, or if the SKU Code field is left to default to %, all currently defined SKU's will be defined to the new warehouse. The SKU's can be further selected by any of the additional Selection criteria. The resulting entries can also be defined as to which values to copy from the existing warehouse settings.

1.9 Option SKU's

It is now possible to define a SKU to have "optional" SKU's available for ordering along with it. These are defined as Type "op" on the SKU Substitute Table.

These "option" SKU's also need to be defined on the SKU Master just as all other SKU's. However, if their 'option' price is to be different from their regular (non-option) selling price, that price needs to be defined on the SKU Pricing Table with Price Type "o".

FLEXX® *7.0L0 Procedures Guide - Distribution Modules*

When the primary SKU is entered on a sales order (in Order Processing), FLEXX will prompt with message "Access options for ……. ?"
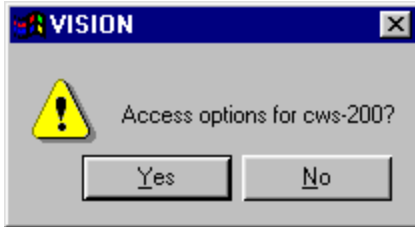
Pressing Yes will display the SKU Substitute Items form showing all of the SKU's defined with Type "op" for this ordered SKU.

 Check the Select box for those option SKU's desired and enter the quantity required. Press **Add Selected** and <<Previous Form>> to return to the order detail. On saving the Order detail entry, the options selected will be automatically entered on the order.
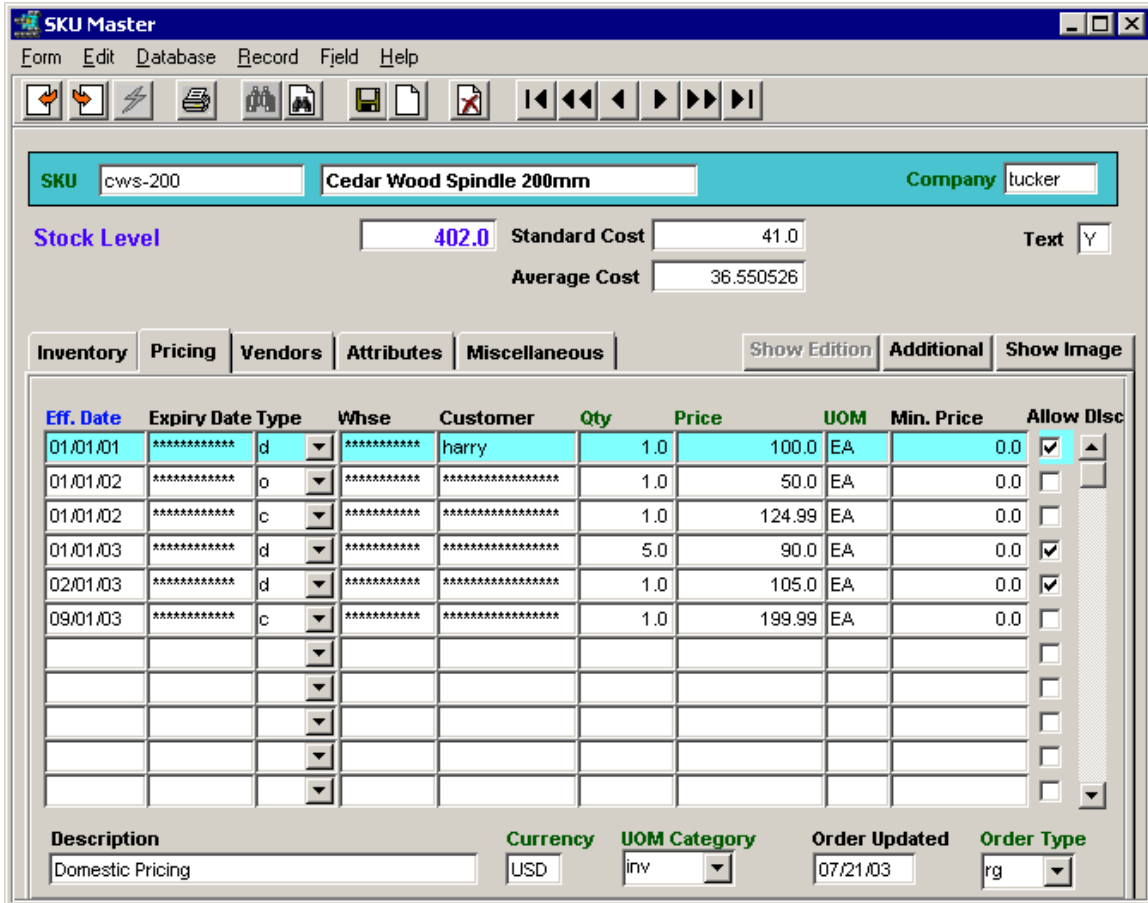
2   SKU Pricing

The individual SKU List (selling) price is defined on the SKU Master Price table *(also called the Price Matrix)*. Then, depending on whether or not discounts are allowed for the SKU, F*LEXX* will use the various specified discount codes to calculate the 'Net' price of the SKU.

2.1 Price Definitions

The SKU List Prices are determined by the values entered on the SKU Master Pricing form *(see Inventory Control manual for more detail)*. The list price may be separately defined by effective date, warehouse, price type, customer, and/or quantity break point. Any eligible discounts are calculated based on the effective Price values.

F*LEXX* will attempt to use a price defined;
- by price type for a given customer,
- by price type for a given warehouse and customer,
- by price type for a given warehouse,
- by price type.

For each search, FLEXX will also use the Currency code value to find a match with the customer's setting.

Within this hierarchy any price quoted must be consistent with the effective date of the price record (i.e. a price is valid from the specified Effective Date until the Expiry Date) and the quantity being ordered. If F*LEXX* determines that more than one price is possible, the one chosen is that for the greatest break point.

It must be noted that without careful definition of the pricing matrix, anomalies are possible (several examples follow):

- If a customer has a matching entry in the matrix, that price will be quoted even if it is more than a price for the customer's pricing group or even the normal price for the SKU. In general therefore prices should decrease as one goes through the hierarchy.

- The quantity break point is higher in the hierarchy than the date. For example, a one year old unit price for greater than quantity ten of the item will be chosen over a two day old unit price for nine units even if it has a higher price. Therefore, if a special price is being added as an incentive, it should be compared to any existing price for a higher break point to check that a conflict will not be generated. This is not an issue if the price is being added at the same break point as an existing price; the new will override the old.

- If the defined SKU price is to be used on an Order return, the Quantity field requires a negative value in the amount that will accommodate the number of items being returned. Effective date, price type, warehouse, and customer definitions will also apply so need to be considered. This is only significant on manually entered return orders. Return orders generated using the **Return** button on the Order Detail use the same price as was entered on the original order.

2.2 Customer Price Type

The SKU *Sale* price is further determined by the Customer **Price Type** and **Currency** code value defined on the Customer Master. The SKU List price will be matched with this Price Type to determine the customer's effective selling price. *See the "Order Pricing and Discounts" description for more detail on order pricing.*

2.3 SKU Pricing Example

Using the SKU Price Table below, following is a description of how FLEXX determines the effective List price:

Eff. Date	Expiry Date	Type	Whse	Customer	Qty	Price	UOM	Min. Price	Allow Disc
01/13/98	************	d	***********	*****************	1.0	50.0	EA	0.0	☐
03/16/98	************	d	main	*****************	1.0	75.0	EA	0.0	☐
09/01/98	************	d	main	*****************	1.0	90.0	EA	0.0	☐
09/01/98	************	i	***********	*****************	1.0	125.0	EA	0.0	☑
09/01/98	************	d	main	*****************	100.0	75.0	EA	0.0	☑
09/01/98	************	d	warehs	*****************	1.0	95.0	EA	0.0	☑
09/01/98	************	d	***********	*****************	-10.0	90.0	EA	0.0	☐
12/21/98	************	d	main	remco	1.0	100.0	EA	0.0	☐

Description: Domestic Pricing **Currency:** USD **UOM Category:** inv **Order Updated:** 01/14/02 **Order Type:** rg

Entry	Description
1	Effective 1/13/98, SKU price is $50. for all '*d*' type customers , all warehouses, no quantity break, no discounts.
2	Effective 3/16/98, SKU price is $75. for all '*d*' customers ordering from '*main'* warehouse, no qty break, no discounts.
3	Effective 9/1/98, SKU price is $90 for all '*d*' customers, from '*main'* warehouse, no qty break, no discounts.
4	Effective 9/1/98, SKU price is $125 for all '*i'* type customers, all warehouses, no qty break, discounts allowed.
5	Effective 9/1/98, SKU price is $75 for all '*d'* customers, ordering 100 or more items from '*main'* warehouse, discounts allowed.
6	Effective 9/1/98, SKU price is $95 for all '*d*' customers ordering from '*warehs'* warehouse, no qty break, discounts allowed.
7	Effective 9/1/98, on any returns of up to 10 items, the credit price is $90.
8	Effective 12/21/98, SKU price is $100 for customer '*remco'* only, ordering from warehouse '*main'*, no qty break, no discounts.

2.4 Minimum Pricing

FLEXX allows SKU pricing to be set to a minimum value to be used in Order Processing. The value is set on the SKU Pricing table, **Min. Price** field.

This value is used at Order Detail entry where the minimum price for this SKU on any single order will be the **Min. Price** specified value. Note that this price is not the minimum price per unit of the item, but the minimum sales amount for this item.

Using the above example, when ordering this item, although the price for one unit is $330.00, the entry will be priced at $450.00. However, if ordering more than one, the price used will be 330.00 multiplied by the quantity, and if the Extended Price is larger than 450.00, the actual price will be used. See the example below.

Order Detail Example

Notice the different Sell Unit Prices for the same SKU entered.

2.5 Changing SKU Prices

SKU prices are normally manually entered into the Price Matrix, individually for each SKU and Price Type. Price changes can also be accomplished by manually changing each price entry for each SKU. However, there are several methods whereby price changes can be performed over either a group or the complete inventory of SKU's. These are the **Price Update Maintenance** procedure and the **Generate New SKU Price Records** function of Inventory Control.

2.5.1 SKU Price Update Maintenance

The **SKU Price Update Maintenance** routine is part of the **Landed Cost Maintenance** process in that it can be used to update the List prices of all SKU's affected by Cost changes. This is described in detail in the *SKU Price Update Maintenance* description *(Topic 9.0)*.

2.5.2   SKU Price Generation from Existing Price

The SKU Price Generation from Existing Price routine can be used to update prices according to a percentage, flat rate, or a fixed value, based on the existing price definition.

SKU selection for this process can be by:
- individual SKU code
- SKU Class - Class field on SKU Attributes form
- SKU Category - Category field on SKU Miscellaneous form
- Cutoff Date – SKU price Expiry Date on SKU Price table
- Price Type - Price Type on SKU Price table
- User Attribute 3 - SKU Attributes User Defined field 3

Options selections then determines the type of price change to perform:
- Percentage requires a percent increase/decrease value in the **Percentage increase** field. A negative value will result in that percentage amount price decrease.
- Flat requires a flat amount value to be specified in the **Flat Amount** field.and that amount will be added to the SKU prices. A negative value will result in that amount price decrease.
- Fixed indicates that the value entered into the **Fixed Amount** field is the actual SKU price to be recorded for those selected. It is not an increase or decrease but an absolute price definition. So, all SKU's selected from the selection parameters entered will be defined with this price value.

Generation Criteria can also be defined as to the Effective and Expiry dates of the SKU.

2.5.3 SKU Price Generation from Average Cost

The SKU Price Generation from Average Cost function has enables the user to generate new price records for any SKU's according to the selection parameters, based on the **average cost** of the SKU. The Selection parameters allow selecting SKU's either in groups or individually. The Generation parameters are used to specify the values that are to be entered for each price record. These are described below in more detail.

Selection Criteria:

- Company – defaults to the session company code
- SKU Code – can be % (for all) or specific SKU Code
- SKU Class – can be % or specific Class code (from Attributes)
- SKU Category - can be % or specific Category code (from Miscellaneous)
- Vendor Code - can be % or specific Vendor code, as defined on SKU Vendor table – will then only select SKU's that are defined with that vendor code
- Vendor Group - can be % or specific Vendor Group code, as defined on Vendor table – will then only select SKU's that are defined with the vendors having that Group code
- Price Type - can be % or specific Price Type code (SKU Pricing) – will then only select SKU's that are defined with that type code

- Price Disc. Flag - can be % or Y or N:
    - % - select all discountable and non-discountable SKU's
    - Y – select only those SKU's that are defined for discounts
    - N – select only those SKU's that do not get discounts
- Price UOM Cat. - can be % or specific UOM Category code – will then only select SKU's that are already priced with that UOM category code (e.g. inv)
- Price UOM - can be % or specific UOM – will then only select SKU's that are already priced with that UOM code (e.g. EA)
- Avg Cost Range – defaults to 0 and 999999, and can be set to any desired cost range value; will then select SKU's with Avg Cost is in that range.
- Avg Cost Whse - can be % or specific Warehouse code – will then select the Average Cost from the specified warehouse to calculate the new price, as described below.

Generation Criteria:

- Effective Date – the starting date the on which the price will be effective
- Expiry Date – the date the price will end (field is not required).
- Options:
    - Percentage – generate the new price based on a percent increase over Avg. Cost
    - Flat – generate the new price based on adding a flat value to the Avg Cost
    - Fixed – generate the new price at a set value
- Percentage Increase – if Percentage is selected, specify the **<u>Markup</u>** percent to be used to calculate the new price, or
- Flat Amount – if Flat is selected, set the flat amount to **<u>add to</u>** the Avg Cost to generate the new price, or
- Fixed Amount – if Fixed is selected, enter the specified amount to be used as the new Fixed price
- Price Margin – if Price Margin is used as part of Price Update, enter the new Margin value to be generated in the Price Margin table for the SKU(s) (field is not required).
    - If an existing record for same SKU and Price Type, set it to status "i".
- Price Type – specify the Price Type code for the new price record;
    - Can be the same as an existing type or a new type code;
    - If the new price record is an adjustment to an existing price, use the same Type code as the Selection Type code.
    - If a price record already exists for the specified Price Type, create a new record; do not delete or overwrite old records.
- Whse – can be left null for all warehouses, or enter a specific Whse code if the new price is specific to a warehouse (field is not required).
- Customer - can be left null for all customers, or specify a specific Customer code if the new price is specific to all customers (field is not required).
- Qty – specify the sell quantity the new price is to be generated for.
- UOM Category – specify the UOM Category code the new price is for; defaults to the "default" setting on the corresponding Master Types Table (e.g. "inv").
- UOM – specify the UOM the new price is for; this can be one of the following values:
    - The specific UOM code
    - STKUOM for the defined stocking UOM (on Attributes form)
    - SELLUOM for the defined selling UOM (on Attributes form)

- If UOM specified is other than the Stock UOM, convert the sell UOM using the UOM Conversion table and calculate new price based on the number of units.
- Min. Price – specify the Minimum Price for this new price record (field is not required).
- Allow Disc. – specify Y or N if the new price is eligble for a discount.
- Currency – specify the currency the new price is to be generated in; if foreign currency specified, use the effective rate from the Exchange Table to calculate new price.
- Order Type – specify the Order Type code (SKU Pricing table) the new price s for; defaults to "rg" but can be any other predefined value.
- Description – enter the description that is to be recorded for the new price.

Generate prices:

- If Avg. Cost Whse specified and Avg Cost greater than 0, use % Increase on whse Avg Cost value to create price record;
    - o If Avg Cost = 0, use Last Cost to create price record;
    - o If Avg Cost = 0 and Last Cost = 0, define new price record = 0.00
- If Avg. Cost Whse not specified, use SKU Avg Cost - over <u>all warehouses</u>;
    - o If Avg Cost = 0, use the latest purchase price, which will be the latest Total Landed Cost value from skuven (last dated vendor adjusted cost). Note that if this value is zero (not yet defined), the generated new price will be 0.00.
- If Non-Inventory (no warehouse defined, and Avg Cost & Last Cost = 0),
    - o Use Standard Cost value (Standard Cost gets updated on PO Receive); or
    - o If Standard Cost = 0, define new price record = 0.00
- If a price record already exists for the specified Price Type, create a new record; do not delete or overwrite old records.
- If foreign currency specified, use the effective rate from Exchange Table to calculate new price.
- If UOM specified is other than Stock UOM, convert the sell UOM using the UOM Conversion table and calculate new price based on the number of units.
- If Price Margin is used, create a new Price Margin record according to the value in the Price Margin field;
    - o If an existing record for same SKU and Price Type, set it to status "i".
- All of the above apply for Flat or Fixed options; only the generated price value will be calculated differently;
    - o Flat – adds the specified flat amount to the Cost;
    - o Fixed – sets the price value to a fixed amount.

3   Bill Of Material Function

The Bill of Materials (BOM) item first needs to be defined on the FLEXX SKU Master as a SKU item. The BOM SKU can then be handled in several different ways.

- Inventoried, and Explodable or Non-explodable
- Non-inventoried, and Explodable or Non-explodable
- Serialized

The definitions to provide these BOM functions are described below.

3.1 Inventoried BOM

3.1.1   Explodable
- Attributes flags *Inventory* checked and *BOM* set to "B".
- Inventory levels will be maintained on the Inventory table.
- Will be exploded on sales order and invoice showing all component (child) SKU's.

SKU cws-500 is also a BOM, and can also be marked Explodable.

3.1.2 Non-Explodable

- Attributes flags *Inventory* checked and *BOM* set to "C".
- Inventory levels will be maintained on the Inventory table.
- Will not be exploded on sales order or invoice.
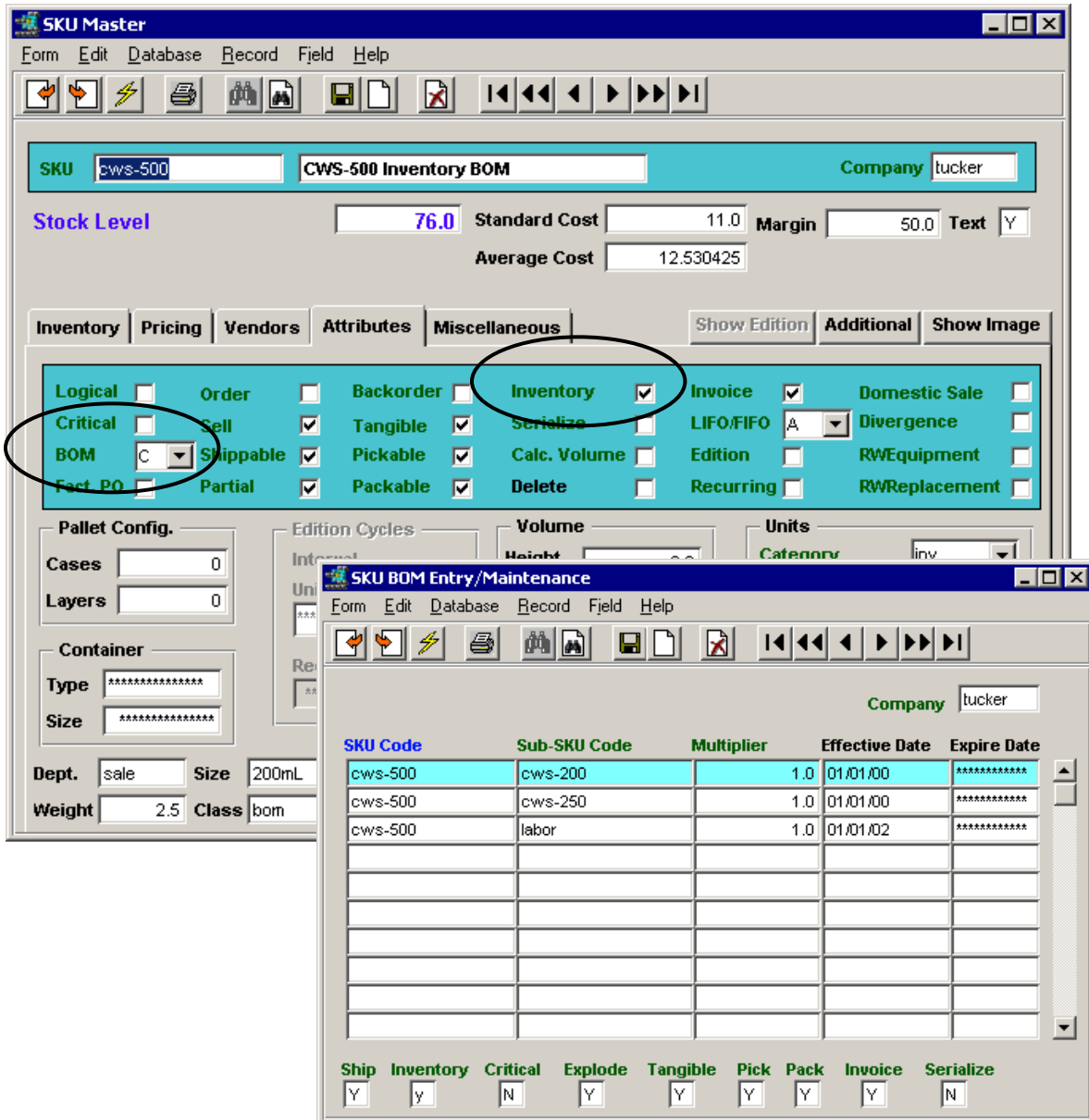
The BOM component SKU's need to be defined for the 'parent' BOM SKU on the SKU BOM Entry/Maintenance form. Then the Warehouse to Warehouse Transfer "b" (Build) process will need to be run for FLEXX to assemble (build) the BOM into a unique SKU item.

3.1.3 BOM Build Process

The Warehouse to Warehouse Transfer/Build/Receive process is used in 'building' the BOM SKU. This process will take the BOM component SKU's from inventory (decrement On Hand) and assemble them into the SKU defined as the BOM. For the process to be successful, a sufficient quantity of the 'child' SKU's must be On Hand to satisfy the Build quantity. This process will not 'Back Order' any items that are not currently available.

Selecting Warehouse to Warehouse Transfer/Build/Receive from the Inventory Control menu will display the following form:

Enter **Type** *'b'* and the BOM SKU code which will cause FLEXX to remove the required component SKU's from inventory at the specified warehouse to 'build' this item (kit). Specify the number of BOMs to build in the Quantity field, and the required component SKU's will be pulled from inventory (On Hand qty will be decreased by the required number).

Every time another BOM is to be built, this process is run, and the resulting component SKU's quantity will decrease by the number required to build each BOM SKU.

3.2 Non-Inventory BOM

A Non-inventory BOM is also referred to as a "build-on-the-fly" BOM. At order entry, FLEXX will automatically assemble (build) the 'parent' BOM using the SKU's defined on the SKU BOM Entry/Maintenance form (shown above). The Warehouse Transfer Build process is not used to build the BOM.

3.2.1 Explodable

- Attributes flags *BOM* set to "B", *Inventory* NOT checked.
- Inventory levels will be maintained for the component SKU's only.
- Will be exploded on sales order and invoice showing all component SKU's.

As for the Inventoried BOM, the Non-inventory BOM component SKU's will need to be defined on the SKU BOM Entry/Maintenance form. However, since the BOM is not an *Inventory* item, meaning the completed BOM is **not** stocked as a separate entity, the Warehouse Transfer/Build process does not need to be run, and in <u>fact should not be run</u>.

Since there is no BOM 'build' process, the component SKU's would not be removed from inventory and would still be available for separate ordering. Further, at BOM ordering, the BOM would be dynamically 'built' by withdrawing the required component SKU's from inventory as defined on the SKU BOM Entry form, and the Order Detail would show the same exploded detail of the BOM *(see example below)*. This dynamic build process **requires** that the BOM flag be set to 'B' at time of order entry.

3.2.2 Non-Explodable

- Attributes flags *BOM* set to "C", *Inventory* NOT checked.
- Will be NOT exploded on sales order and invoice.
- Should ONLY be used if component (child) SKU's are also Non-inventoried SKU's.

Although this BOM configuration is not normal, it is acceptable for standard BOM processing.

Several conditions need to be considered for this to be valid:

- BOM should also be tagged as Non-Tangible (Attributes *Tangible* flag set OFF).

- All component SKU's must also be tagged as Non-inventory and Non-tangible. If any of the 'child' SKU's are Tangible, the BOM SKU should be tagged Explodable (BOM type B).

The BOM component SKU's will still need to be defined for the 'parent' BOM SKU on the SKU BOM Entry/Maintenance form, and as on the Explodable, Non-inventoried BOM, the Warehouse Transfer/Build process is not run.

3.2.3 Order Detail - BOM Exploded.

Using the example shown above in Topic 3.1.1 (Inventoried, Explodable), at order entry of the BOM SKU, FLEXX will 'explode' the 'parent' SKU into its component parts, including any explodable component BOM SKU's.

This example shows the two BOM exploded views:

- BOM SKU 'cws-900' exploded to include SKU's 'cws-500' and 'cws-870',

- BOM SKU 'cws-500' further exploded to include 'cws-200', 'cws-250', 'labor'.

To determine which detail entry is what relative to the 'parent' SKU, display the **Group** and **Level** fields for each detail line.

- **Group** – will show the BOM group number for each group of SKU's that make up the BOM.

- **Level** – will show the group within a group if a 'child' SKU is also a BOM.

In this example, they will all be Group 1 since there is only one BOM, and the Level will be shown as follows:

- 0 – BOM 'parent' SKU

- 1 – 'child' SKU's belonging to the 'parent' cws-900 (cws-500 and cws-870)

- 2 - 'child' SKU's belonging to the 'parent' cws-500 (cws-200, cws-250, labor)

3.3 Serialized BOM

A BOM in FLEXX can also be defined as a Serialized SKU indicating it will be assigned serial numbers at time of being 'built'. However, a Serialized BOM has additional conditions that need to be considered.

1. Serialize flag set ON,

2. Defined as Serial Type "se",

3. MUST be defined as Inventory (so must be pre-built),

4. Defined as Tangible

5. Must be Explodable (type B) if any 'child' SKU's are also serialized.

3.3.1   Build Serialized BOM

The BOM Build process is similar to that described earlier for an Inventoried BOM. The serialized BOM SKU first needs to be defined on the SKU Master. Then, the component SKU's need to be defined on the SKU BOM Entry/Maintenance form.

However, there is an additional process where the new BOM SKU's are assigned serial numbers. Using the Warehouse to Warehouse Transfer/Build/Receive process, on saving the entry, FLEXX will display the Serial Number Information form requiring entry of the specific serial number.

Although the Serialized BOM item will not be purchased from a vendor, the Serial Number entry still requires a vendor code to be specified. Any valid code defined on the Vendor Master table can be used. Alternatively, FLEXX Application Control can be defined to specify a default vendor code to be used when building Serialized BOM SKU's, as follows:

Application	Type	Description	Value	Company	Parameter
ic	bombuild	Default Vendor for Serial BOM	no default	default	bom

On completion, FLEXX will exit this form only when the correct number of serial numbers have been entered and saved.

If the Serialized BOM also consists of serialized 'child' SKU's, a further process will need to be performed before the BOM Build will be completed.

3.3.2 Serialized BOM with Serialized 'Child' SKU

If any of the component SKU's of the Serialized BOM are serialized, there needs to be a sufficient quantity of available 'child' SKU's and their associated Serial Numbers.

On saving the Serial Number entry on the Serial Number Information form, FLEXX will display the BOM Serial Detail form. This form will display ALL component SKU's for the BOM being built, but only the Serialized SKU's will cause the **Serial Number** button to be active (lit).

Pressing this button will display the Serial Number List form where the desired item's Serial Number is selected.

After making the selection, return (<<Previous Form>>) to the BOM Serial Detail form, and if any other listed SKU's are serialized, repeat the serial number selection for each of them.

When finished, exit (<<Previous Form>>) both the Serial Number List and BOM Serial Detail forms, returning to the Serial Number Information form. Returning to the Warehouse Transfer form will complete the process.

If at any time of serial number entry, the number of values entered does not satisfy the 'build' quantity, FLEXX will display a prompt message and require you to make the necessary entries to satisfy the process. FLEXX will not allow you to exit the process until the selection is complete.

3.4 BOM Verification

A function is provided in FLEXX Inventory Control to verify the BOM definitions for accuracy. This function is part of the **BOM Verification Report** which should be run whenever a new BOM SKU is defined. This report will interrogate the various critical attributes definitions of the BOM SKU and report any conflicts with their settings.

Following are the conditions tested:

- BOM SKU should be Explodable if any 'child' SKU's are Serialized

- BOM SKU should be Explodable if any 'child' SKU's are Inventoried

- BOM SKU should be Serial Type "se" if any 'child' SKU's are Serialized

- BOM SKU should be Serial Type "se" if BOM is Serialized

- BOM SKU should be Explodable if any 'child' SKU's are Invoiceable

- BOM SKU should be Explodable if any 'child' SKU's are Tangible

- BOM SKU should be Invoiceable if any 'child' SKU's are Invoiceable

3.4.1   BOM Verification Report

3.5 BOM Costing and Pricing

Bill of Material SKU costing and pricing is somewhat different from ordinary SKU's. This is due to the fact that BOM's are not purchased but assembled (built) using other defined SKU's. Further, there is also a difference in costing between Inventory and Non-inventory BOM's. Pricing is the same for both.

3.5.1 Inventory BOM Costing

A BOM SKU that is defined as an Inventory item (Attributes Inventory flag set) must be 'built' before it can be used, as described in Topic 3.1.3. At the time of running the Build process, the cost of each 'child' SKU is used to determine the BOM cost. The 'child' cost value used will depend on whether or not the 'child' SKU is inventoried, as follows:

- If inventoried, the cost is the Warehouse Average (or LIFO/FIFO) cost defined on the SKU Inventory form.
- If not inventoried, the cost used will be the Standard Cost value defined on the SKU Master form.

The effective cost of the BOM is then the sum of the costs of the 'child' SKU's used that make up the BOM item. This value is then recorded in the Avg. Cost field of the Warehouse where the BOM is stocked.

Example using SKU *cws-500*, consists of one each
- *cws-200* – Inventory, Avg. Cost $2.00
- *cws-250* – Inventory, Avg. Cost $3.00
- *labor* – Non-inventory, Standard Cost $10.00
- After BOM build, Avg. Cost of SKU *cws-500* will be $15.00

Be aware that if the costs of the 'child' SKU's change after the BOM has been built, the cost of the BOM SKU will not change. The BOM is costed only at the time it is built since that is when the 'child' SKU's are withdrawn from inventory and their costs are relieved.

3.5.2 Non-Inventory BOM Costing

The costing of a Non-inventory BOM is somewhat different in that the SKU is not defined to be stocked in a Warehouse, and therefore will have no Warehouse Avg. Cost value set. Further, this BOM is not pre-built, but assembled at time of placing an order (Sales, WO, etc.) so will not have a cost value set on the SKU Master. The cost of the BOM will be determined as it is built on the order, using the 'child' SKU costs similar to the above criteria:

- If inventoried, the cost is the Warehouse Average (or LIFO/FIFO) cost defined on the SKU Inventory form.
- If not inventoried, the cost used will be the Standard Cost value defined on the SKU Master form. If a Non-inventoried BOM, will be the accumulated 'child' SKU costs.

This effective cost of the BOM used by FLEXX will be the accumulated sum of the costs of the 'child' SKU's. This cost will be recorded in the General Ledger at the time the Generate GL Transactions routine is run. The cost of the Inventory SKU's will be posted to the Inventory GL account, and the Non-inventory SKU's to the Expense GL account as defined on the SKU GL Accounts table.

3.5.3 BOM Pricing

The pricing of a BOM SKU in FLEXX is the same for Inventory and Non-inventory BOM's. They are both priced according to the price definition on the SKU Price table for the BOM 'parent' SKU. The selling price needs to be defined as with any other SKU, and must adhere to the same conditions. *Please refer to Chapter 2 for more details.*

Be aware that BOM pricing will not be automatically adjusted if its costs change. However, as of Version 6.5L2, the FLEXX *Price Update Function* can be used to re-price a BOM after it is built and its costs have changed *(see Price Update Maintenance topic 9.4 for details)*. BOM prices can also be adjusted manually as follows:
- individual adjustment on the Price table, or
- using the Generate New SKU Price Records process in Inventory Control *(see Topic 2.5.2)* .

4   Edition Function

4.1 Edition SKU Master

The *Edition Cycles* and *Recurring Interval* definitions occupy the same location on the Attributes form. Selecting the *Edition* flag will highlight the Edition Cycles fields. Selecting the *Recurring* flag will change these fields to Recurring entries. The Edition and Recurring flags cannot be selected simultaneously. One must be deselected before the other can be selected.

The Edition Cycles fields have a similar meaning as the Recurring Interval values:
- Units – day, week, month, year
- Qty – intervals between cycles
- Times – number of times to repeat the cycle

The *Safety Level* field is used in the Print Order Generation process, which is a FLEXX custom function.   A value less than one indicates a factor, and larger than one a constant quantity, but is always a safety factor. Also, the **Min.** value is used as an additional constant 'add-on' to the Safety Level factor. *For more detail*

on this process, please request a copy of the Generate Print Orders *description which is not included in this manual since the function is a* FLEXX *Custom modification.*

The **Current Edition Stock Level** field is used by FLEXX to indicate the quantity On Hand of the current edition <u>only</u> (as defined on the Miscellaneous form described below).

4.2 Miscellaneous Form

If the defined SKU is an Edition item, the Miscellaneous form will contain additional fields indicating the **Current Edition** number, and the **Next Edition Due Date**. Further, the **User Defined Fields** (User 1 to 5) will have unique meanings and uses, and their labels will be so defined in the corresponding Master Types Tables.

4.3 Editions Table

Pressing the **Show Edition** button presents the Editions table which shows all editions defined for this SKU, their status, effective date, and date that edition becomes current.

The Editions form is used to specify the various edition dates and dimensions.

Also, two User Defined fields can be labeled according to Application Control variables *ed_user_1* and *ed_user_2* (i.e. Pages, Scale, etc.).

The Editions Inventory form is used to record the inventory levels of each stocked edition for each warehouse where stock is to be maintained. This is separate from the SKU Master Inventory form, and needs to be maintained separately.

The sum of all editions for each warehouse should equal the quantity for that warehouse on the SKU Master Inventory form.

4.4 Edition Pricing

The 'Editions' SKU **Pricing** table needs to have a List Price defined for an effective date, and the **Order Type** field must be set to *'sb'* if the Edition SKU's are to be ordered using the FLEXX Subscription module.

Further, the Subscription price (Order Type "sb") must be defined with the **Whse** field undefined. For regular sales orders (Order Type "rg"), the Whse definition will be used by OP order processing as normal.

4.5 Edition Condemns

The **Warehouse to Warehouse Transfer** function is used to perform SKU 'Adjustments' and 'Condemns'.
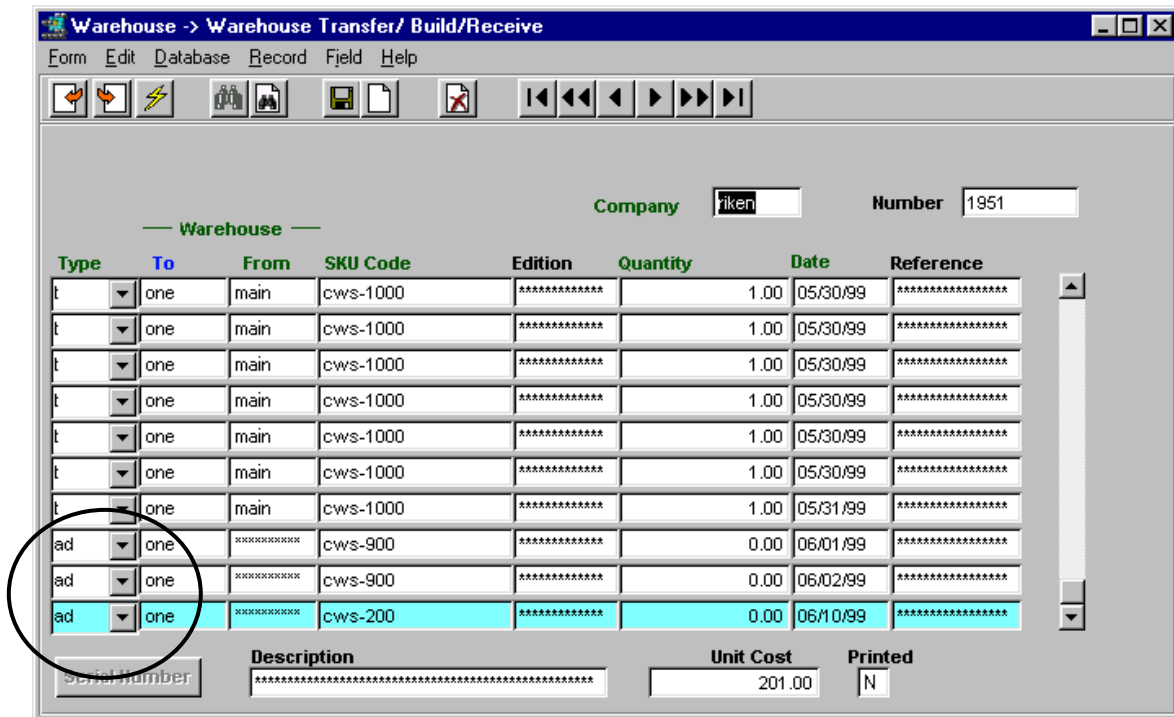
To make manual **adjustments** to stock quantities (both SKU's and Editions), use the Warehouse to Warehouse Transfer process and set the Type to *'ad',* enter the affected warehouse in the To field (From will default to the same value), the SKU code and Edition and adjustment Quantity (can be positive to increase inventory and negative to reduce inventory). The specified quantity value will now also adjust the SKU Inventory table by that amount.

To perform **condemns** of SKU Editions, set the Type to *'cd',* and as for adjustments enter the affected warehouse in both the To and From fields, the SKU code and Edition and Quantity to condemn. This will result in the quantity being made a negative, and that amount to be reduced from inventory.

Run the **Generate GL Txns (Inventory Movement)** from the Inventory Control menu to post the cost adjustments to the specified GL accounts. This routine will post only 'ad' and 'cd' type transactions as displayed on the Inventory Movement Review table.

5   SKU Container and Environmental Charges

If SKU's are containerized items (bottles, cans, etc), and there are charges to be invoiced for the containers, these charges are defined using the Container Maintenance Table *(refer to Inventory Control, Sec. 3.10 for more detail on the individual fields).*

Container and Environmental charges are generated on the Order Detail based on the specific SKU code and customer Ship To Province/ State Code. Additionally, the following Application Control variables need to be set:

Application	Type	Description	Value	Company
ic	EnvironFee	Environmental Fee	Y (N default)	default
op	container	Container Charge	Y (N default)	default

The SKU Master, Attributes form, Container fields *Type* and *Size* are used to specify the type and size of the specific SKU container which must first be defined in the Container Maintenance Table, as shown in the following example:

Assume order Ship To location is BC;

SKU Code *ojuice* is defined as Container Type TTRA and Size 946.00.
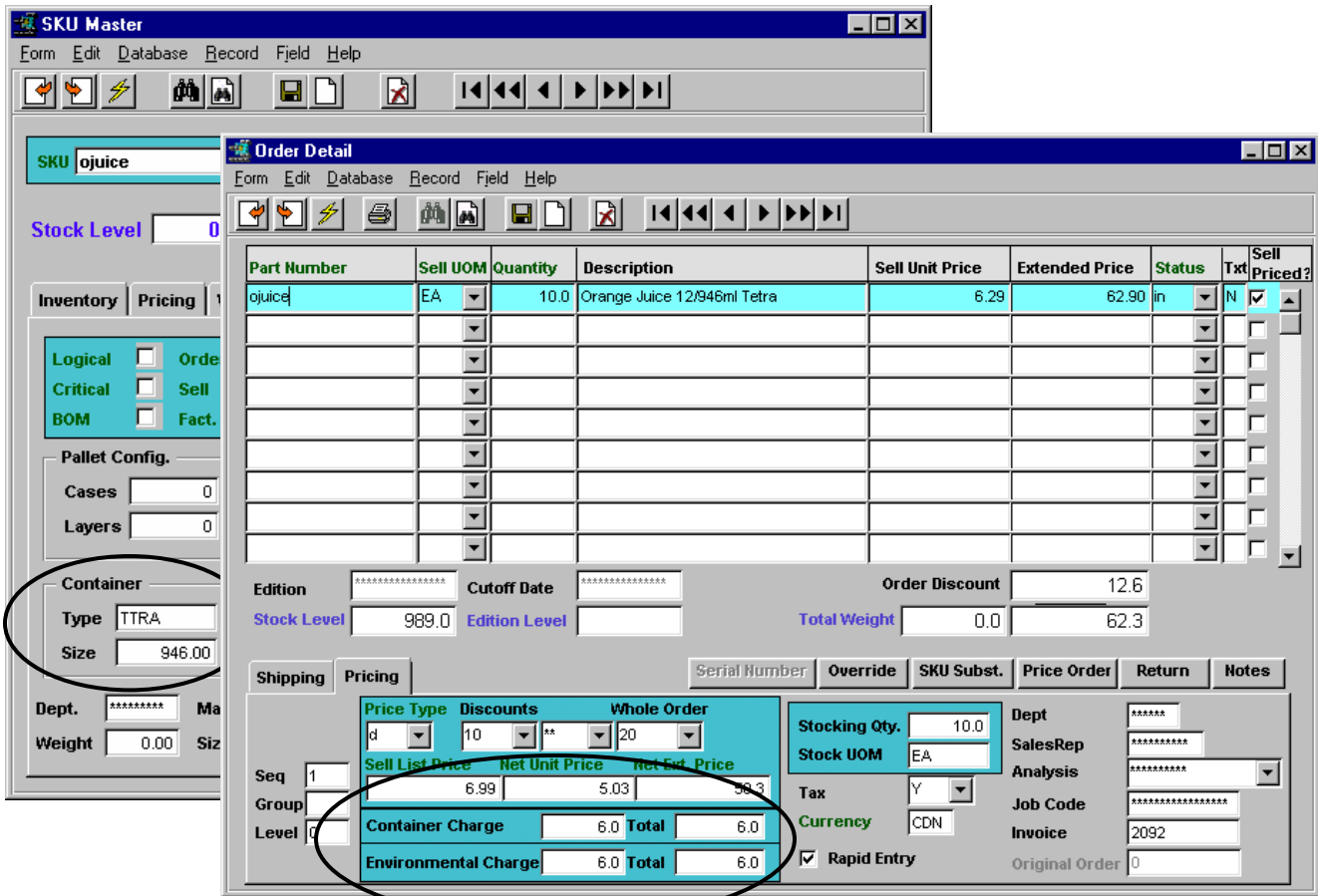
On entry of this SKU on the order detail form, FLEXX will interrogate the Container fields to determine if amounts are to be added for Container and/or Environment charges. Notice the Container Charge and Environmental Charge fields at the bottom of the Order Detail form, an amount of 6.00 for each has been

generated. This value is the result of the Quantity ordered multiplied by the Rate defined in the **Container Maintenance Table** *(see below);* i.e. 10 SKU's @ 0.60 per SKU.

This requires the Container Maintenance Table to have an entry for code *TTRA*, with Max Size to include the size specified on the SKU Master *Size* field (i.e. 946) and the State BC defined as the following display indicates.

5.1 Container Definitions

Note that the SKU consists of 12 - 946ml containers, so the Rates need to be defined for that number of individual containers (i.e. 12 containers/SKU @ 0.05/container = 0.60/SKU for TTRA).

```
Container Maintenance                                              _ □ ×
Form  Edit  Database  Record  Field  Help

 [icons]

                                              Company    prolog

Group    TTR          Group Desc.    TetraPaks


Cont.Type   State   Rate Type      Max. Size     Rate        Environmental Rate
TTRA        BC      V                999.00     0.6          0.6
TTRB        BC      V               9999.00     6.0          6.0


Cont. Desc.  >1 Litre 12 @ .50 ea.                          Text  N
Credit Account  consul    5100     Environmental Credit Account  consul   6050
```

This example confirms the resulting Container and Environmental Charges generated, in that qty 10 at 0.60 each equals 6.00.

This same table is used to define additional containers and rates; e.g.

TTRB   BC   V   9999   6.0

meaning, for containers with max size larger than 1 Litre, the charge is $6.00/SKU (or 0.50/container).

At Invoice Generation of this order, the Container/Environmental Charges are recorded as separate entries on the Invoice Detail form. However, the printed invoice will show only the Container Charges; the Environmental Charges will be included in the SKU price (i.e. hidden) since they are a non-refundable cost added to the SKU price.

5.2 Container Charge Invoicing

Using the above example, the SKU Price of 6.29 will be reported as 6.89 and the Extended Price of 62.90 will be reported as 68.90 on the Invoice, since the Environmental charge of 0.60 was added to each price amount. And, an entry of 6.00 for the Container charges is added to the total as shown on the invoice report below.

ProLogic Corp (ORACLE:TST6)
555 Bridgeport Road
PO Box 657
Vancouver, BC 58567
Phone: 604-895-7852

INVOICE
GST No.: 8758964

Invoice 2092	Page	1
Due Date	05-Apr-1999	

Bill To: Remco Distributors
234234 Halpin Street
PO Box 78897
FlinFlon BC USA
98456
Attn: Mr Thomas

Ship To: Remco Distributors
234234 Halpin Street
PO Box 78897
FlinFlon BC USA
98456

PO #	Ship Via	Order #	Salesperson	Date	Terms	Tax No.	
4545456	ups	1876	009	16-Mar-1999	Net 10 days	123456123	

Re: System generated from OP

Item	Description	Ordered	Shipped	BO/Not Fill	Unit Price	Extended
1	ojuice / Orange Juice 12/946ml Tetra	10.00	10.00	0.00	$6.8900	$68.90

Subtotal	$68.90
Total Container Charge	$6.00
Discount	($12.60)
BC7 Y	$3.94
GST7 Y	$3.94
Total Due	**$70.18**

Warning:

Be aware that errors in defining the Container/Environmental parameters are normally not reported by FLEXX until the Invoice Generation function is run. So, if you experience failures during Invoicing of 'Containerized' SKU's, you should first ensure the Container Maintenance Table is correctly defined for that SKU and a GL Account has been defined for that charge.
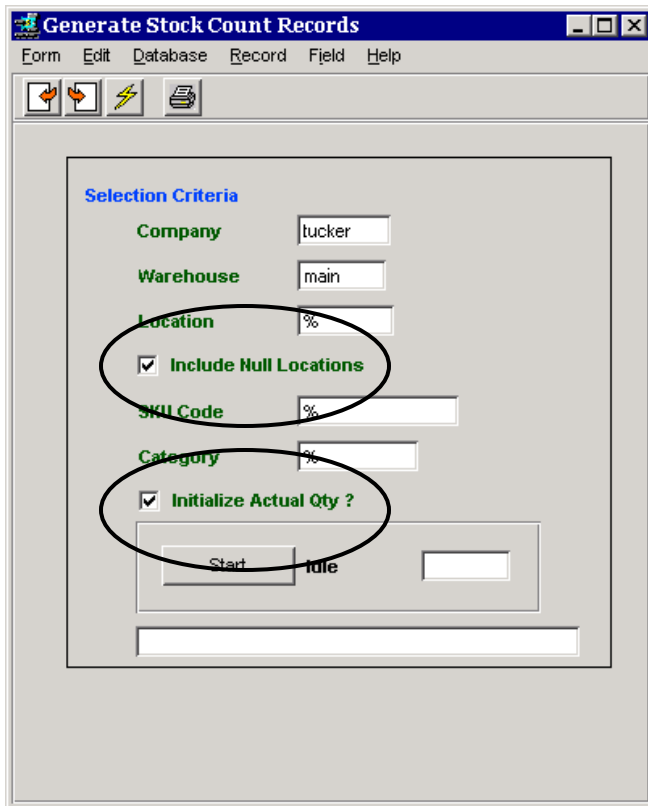
6   Stock Counting Process

The **Stock Counting Process** is used in FLEXX to automate the process of updating and posting the results of an inventory stock count. Following are the recommended steps in their proper order to perform the process.

6.1 Generate Stock Count Records

Run the **Generate Stock Count Records** function for the selected warehouse and/or SKU's.

This will generate the **Stock Count Table** of current inventory levels for the selected entries. If your SKU's do not have a Location code (Location is set to null), ensure you mark the **Include Null** box to include those SKU's.   Also, mark the **Initialize Actual Qty** box if you want the Stock Count Table and the Stock Count Report to show the current FLEXX inventory On Hand quantities.

This process will also mark the *Warehouse Freeze* flags and set the *Freeze Date* on the SKU Master Inventory form for all selected SKU's. The '*Freeze*' flag is used to freeze inventory stock during inventory stock counting. The items can still be ordered but will not be 'released', so cannot be shipped. Once the Stock Count process has been completed *(step 6.6 below)*, the '*Freeze*' flags will be reset off, and all orders that were 'frozen' can now be released and shipped.

6.2 Inventory Stock Count Report

Print the **Inventory Stock Count Report**. This is the Count Sheet used to manually record the actual stock counts, and is then used to update the actual counts into FLEXX. If any SKU's are defined as Serialized with Type "se" (SKU Attributes form), the report will also list all currently available Serial Numbers for each SKU as recorded in FLEXX, which will then also need to be individually counted.

6.3 Stock Count Table Update

6.3.1   Non-Serialized SKU's

Enter the counted values from the Inventory Stock Count report into the **Stock Count Table Update** Quantity fields. If the **Initialize Actual Qty** box was checked on the Generate Stock Count Records function, this field will already contain the system inventory value so will only need to be changed if the count value is different. Note that for serialized SKU's, the Quantity field will be locked and the count is updated using the *Serialized SKU Stock Count Update* form.

6.3.2   Serialized SKU's

If the SKU is a serialized item, the Quantity field will be locked and the **Update Serialized** button will be active. **The quantity update can only be performed by completing the serial count**.

To enter counts for serialized SKU's, press the **Update Serialized** button which is only active for serialized entries. This will display the *Serialized SKU Stock Count Update* form, listing all currently recorded serial numbers for the selected SKU.

The following count entries can now be performed:

1. Press the **Accept All** button to accept all listed serial numbers; or

2. Select the *Accept* flag to mark the highlighted serial number as counted and press <<Add/Update>>; or

3. Select the *Replace* flag if the highlighted serial number is not counted, but another serial number was counted and will replace it. Enter the replacement serial number value into the **Replaced Serial Number** field, and press Add/Update.

4. Select the *Cancel* flag if the highlighted serial number was not counted.

These steps are required to process the serialized SKU count. On pressing <<Previous Form>>, FLEXX will automatically update the Stock Count Table **Quantity** field with the results of these entries. If the process has not been completed, that is, if any entries still show *Not Processed,* FLEXX will display the following prompt:
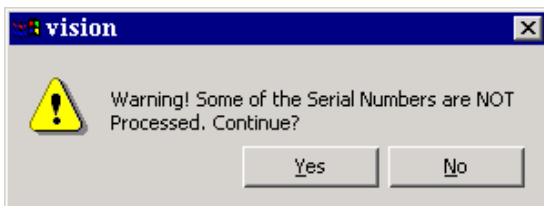
FLEXX will also not complete the subsequent *Post Stock Count Adjustment* process until all serial numbers have been processed.

6.3.3 Cancel Process

The *Cancel* function is used when one or more serial numbers are not found in the physical count. On posting, FLEXX will generate an inventory adjustment transaction to remove that item and its serial number.

6.3.4 Replace Process

When using the Replace process, a replacement serial number value needs to be entered into the **Replaced Serial Number** field. FLEXX will then prompt with several different messages depending on its records.

1. If the serial number value entered as the replacement has never been previously used in FLEXX, the following prompt will be displayed:
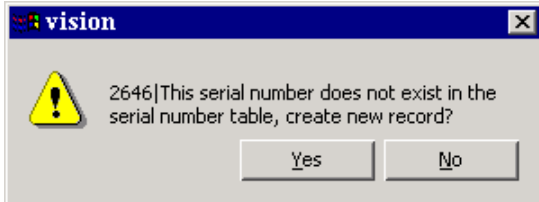
Click Yes to accept this new value as the serial number to be used to replace the listed entry. Press <<Add/Update>> to record the entry.

2. If the value entered as the replacement is currently available (according to FLEXX SKU Master records), FLEXX will prompt with:
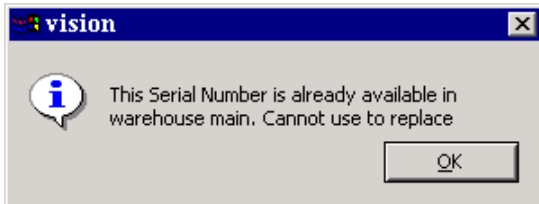
This would indicate some form of stocking error; either located in the wrong warehouse, or a duplicate serial number, etc. The serial number would not be allowed by FLEXX. Press OK to continue.

3. If the value entered was an item earlier sold (through OP, Time Billing, or Repair Warranty) indicating an error in order shipping, FLEXX will display the following message:
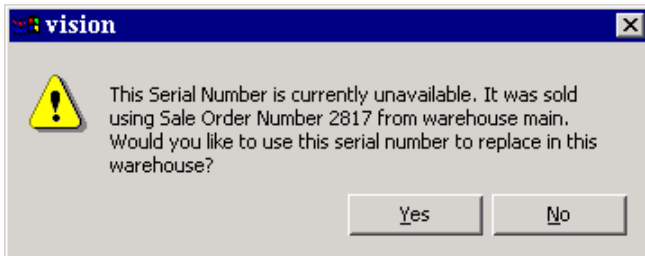
Click Yes to accept this replacement. Press <<Add/Update>> to record the entry.

On each of the replacements, FLEXX will make the corresponding changes to the SKU Serial List table in FLEXX Inventory Control. These adjustments will be completed when the *Post Stock Count Adjustment* process is run.

6.3.5   Add Item Process

If during stock counting, an additional serial number item is found, press the **Add Item** button. FLEXX will present the Warehouse Transfer screen with only Type "ad" (Adjustment) available, and with the Warehouse and SKU Code fields filled.

This will allow the operator to enter only the quantity to be added. All other fields will be locked to match the values on the *Serialized SKU Stock Count Update* form. On Saving (Add/Update), the Inventory quantity will be updated and the *Serial Number Information* form will be presented allowing the operator to enter the new serial number(s), but only as many as correspond to the 'adjusted' SKU quantity.

The additional SKU and serial number will be marked available (status "a") as soon as each entry is saved. The subsequent *Post Stock Count Adjustment* process will not affect these adjustments.

6.4 Inventory Stock Count Variance Report

Once the stock counting entries have been completed, print the **Inventory Stock Count Variance Report**. This report shows the actual count variances of all SKU's, either zero if no variance or the difference between counted and On Hand. Further adjustments can be made to the Stock Count Table as required.

6.5 Inventory Stock Count Posting and Audit Report

Print the **Inventory Stock Count Posting and Audit Report**. This will print the updated stock count listing with the values that FLEXX will use when performing the posting process.

6.6 Post Stock Count Adjustment

Run the Post Stock Count Adjustment routine.

This routine will post any adjustments to the Inventory Movement Table as well as update the SKU Master Inventory table with the new quantity values. This process also resets the *Warehouse Freeze* flags and the *Freeze Date* allowing the affected SKU's to again be shipped. Ensure you again mark the ***Include Null*** box to include those items with a null Location code.

6.7 Remove Stock Count Records

Once the Stock Counting process has been completed, run the **Remove Stock Count Records** process to clear the Stock Count Table.

Entries in either Posted (p) or Entered (e) status can be cleared. Again, ensure you mark the *Include Null Location* box to include those items with a null Location code.

6.8 Generate GL Txns (Inventory Movement)

Adjustments can now be posted to the GL using the **Generate GL Txns (Inventory Movement)** routine from the Inventory Control menu. This routine will post all inventory cost 'adjustment' activity to the GL.

6.9 Examples of Reports

6.9.1 Inventory Stock Count Report

PRO LOGIC CORPORATION
Inventory Stock Count Report

Company: prolog   Warehouse: main

Location	SKU	Description	Serial Numbers	Category / Edition Code	Quantity	Unit	Counted
50-05-W1B2	cws-200	CWS-200			108.0000	CASE	
60-05-W1B2XX	cws-500	CWS-500		ne	5.0000	EA	
	cws-5000	Julians sku			0.0000	EA	
	cws-900	cws-900 is a BOM		ne	10.0000	EA	
			2				
			3				
	cws-linda				-98,877.0000	EA	

6.9.2 Inventory Stock Count Variance Report

PRO LOGIC CORPORATION
Inventory Stock Count Variance Report

Company: prolog   Warehouse: main

Location	SKU Code	Description	Serial Numbers	Category / Edition Code	Unit	Expected	Actual	Variance
50-05-W1B2	cws-200	CWS-200			CASE	108.0000	108.0000	NOT COUNTED
60-05-W1B2XX	cws-500	CWS-500		ne	EA	5.0000	5.0000	NOT COUNTED
	cws-5000	Julians sku			EA	0.0000	0.0000	NOT COUNTED
	cws-900	cws-900 is a BOM		ne	EA	10.0000	10.0000	NOT COUNTED
			2					
			3					
	cws-linda				EA	-98,877.0000	100.0000	NOT COUNTED

6.9.3   Inventory Stock Count Posting and Audit Report

PRO LOGIC CORPORATION
Inventory Stock Count Posting and Audit Report

Company: prolog	Warehouse: main

Location	SKU	Description	Serial Numbers	Edition Code	Category	Quantity	Unit	Actual Quantity
50-05-W1B2	cws-200	CWS-200				108.0000	CASE	108.0000
60-05-W1B2XX	cws-500	CWS-500			ne	5.0000	EA	5.0000
	cws-5000	Julians sku				0.0000	EA	0.0000
	cws-900	cws-900 is a BOM			ne	10.0000	EA	0.0000
			2					
			3					
	cws-linda					-98,877.0000	EA	100.0000

7   SKU Divergence

The FLEXX SKU **Divergence** feature allows sales order (Sell) quantities to be **shipped** slightly 'divergent' from the Stocking UOM quantities. The function is enabled by setting the *Divergence* flag on the SKU Master Attributes form for the affected SKU's *(see example).*

7.1 SKU Definition

Using the above SKU example, *cheese* is stocked by the KG unit (Stock UOM) and is sold by the CSWT (Sell UOM). The Units of Measure table is defined so that a CASE consists of 5 units of KG, or 5 kg per case of product. Note that the Stocking UOM (i.e. KG) should be defined with a Factor of 1.0 for the Sell UOM conversion to work properly.

7.2 Divergent SKU Order Entry

The process described here is a variation on the process described in the Order Processing section, Sell Unit of Measure description. At order entry, the normal sale quantity is entered with the specified Sell UOM (i.e. CASE as defined on SKU Master).

In this example, the Stocking Qty for 1.0 CSWT is 5.0 KG, and therefore the Extended Price is 5 times the selling price, resulting in 5 times the Stock UOM price (1.90) since no Sell UOM price was defined (indicated by the *Sell Priced* flag off). This process is described in more detail in the document entitled *Order Processing – Sell Unit of Measure*.

7.3 Order Shipping

After the order has been Released, pressing the Ship button to ship the order will display the Shipment Maintenance form, and changes can be made to it as required. If the Ship quantity is the same as the Order quantity, press the **Ship** button to ship the order. If the quantity to be shipped is different, enter the actual value in the **Qty Shipped** field and either press Add/Update to update the one detail line or press **Ship** to ship the order.

7.3.1   Shipped Quantity different than Sell Quantity

Shipping a SKU that is defined with *Divergence,* it is possible to make a slight change to the Stock Qty Shipped value, and not affect the actual Sell quantity being shipped. As shown above, **increasing** the Stock Qty Shipped to 5.5 KG (originally 5.0) still results in the actual shipment of 1.0 CSWT, although the CSWT UOM is defined to consist of 5.0 KG units *(Factor value in UOM table)*.   Since the SKU is stocked in KG units, the Inventory levels will be adjusted by this modified amount so that rather than the 5.0 KG that had been Committed at order Release, 5.5 KG will be deducted from the On Hand quantity.

If the Qty Shipped value is changed to an amount equal to or greater than one additional SELL unit (which in this example would be an additional 5.0 KG, or a value of 10 or greater), FLEXX will adjust the SELL Qty Shipped to the corresponding UOM increase. Using the above example, had 10.2 been shipped, FLEXX would increase the Sell Qty Shipped value to 2.0 CSWT units.

The same rationale holds true for under-shipping an item, and a 'bo' (backorder) or 'cf' (cannot fill) entry will generated for the balance of the order (depending on the setting of the Backorder attributes flag). FLEXX will also present the prompt message: "**Were the non-shipped items put back into inventory? Y/N?**".   Responding NO will result in the inventory level not being adjusted for the 'divergence', and the Sell Quantity left at its original value. If YES, the inventory On Hand level will be adjusted for the 'divergent' quantity.

7.3.2 Divergence Order Invoicing

The invoice generated for a 'divergent' sku order will use the value for the actual quantity shipped to calculate the invoiced prices.

Using the previous example where 5.5 KG was shipped and 1.0 CASE was ordered:

The invoice will show the Shipped UOM and quantity in the detail entry line (e.g. 5.50 KG) and the Ordered quantity in the Sell UOM and Units fields (e.g. 1.04 CSWT). The invoiced amount is the shipped quantity (5.50) multiplied by the Rate value as defined in the SKU Master Price table.

Additionally, if the SKU UOM price used to determine the invoice price was defined on the SKU master, the *'Sell Priced'* flag would be set. In this example, since the Sell UOM used was CSWT which was not priced and the Stock UOM price was used, the Sell Priced flag is off.

+

8   Landed Cost Procedure

The **Landed Cost Procedure** is used in the FLEXX Purchasing/Receiving process to distribute the various shipping 'landed costs' (i.e. freight, duty, etc.) to each SKU included on that shipment. The SKU needs to be defined with the eligible Landed Cost Factors if it is to be recognized by the Landed Cost Routine.

8.1 Landed Cost Factors

The procedure uses the SKU purchase price from the Purchase Order and subsequent Voucher process to calculate the new 'Total Landed Cost' of the SKU;

- if a foreign Vendor, Local Cost amount on the SKU Master Vendor form will be converted to the local currency using the exchange rate defined for that date in the Exchange Rate Table.

- It also uses the landed cost factors defined on the SKU Master Vendor form to determine which costs are to be used to calculate the new cost. This means any SKU that is to have the landed costs factored into its purchase price will need to have those factors specified *(see example below)*

Factors:

- o   N = *Not Applicable* (landed cost voucher not required);
- o   O = *Required* (a Landed Cost Voucher entry for this factor is required).
- o   Y = *Allowed* (if a value is defined, it will be used, but is not required);

8.2 Processing Landed Costs

Create a New Purchase Order for the SKU's;

Receive the PO (either directly from the PO or at the Transfer PO to Voucher routine).

Create a Voucher for this Vendor, and if Landed Costs are to be added to SKU costs;

- set Landed Cost box to 'l' (lower-case L) to allow this voucher to be used by the Landed Cost routine. Note that **Landed Cost** button is now active (lit) allowing the routine to be run directly from the voucher.

- Transfer the PO to the voucher:
  - press the Transfer PO button which opens the Transfer PO Detail form; either enter the PO number to be transferred or do a Find and select the PO to be transferred,
  - change values as desired,
  - <<Add/Update>> will perform the PO transfer (will also perform the Receiving if not received on PO),
  - Voucher will now be in 'o' status.

Original PO will now be in 'c' (Closed) status if all entries have been transferred;

- PO Detail lines will be in 'o' status (Open Voucher/Received)

Enter the Landed Costs on a voucher, as follows:

- Either create a new Voucher for the Vendor of each Landed Cost item, or enter the Landed Costs to the existing PO voucher if they are to be paid to the PO vendor.

- If a NEW voucher is required, select Voucher Detail (Non-Inventory),

- enter specific non-inventory 'landed cost' item (duty, freight, etc.)

- enter **Landed Resource Code** value selected from the list,

- enter **Reference Voucher** number (original PO voucher e.g. 10534), which will be used to link the Landed Cost vouchers to the original voucher during the Landed Cost calculation. If landed costs are entered on the PO voucher, enter that voucher number in the Reference Voucher field.

To distribute the Landed Cost charges to the individual PO items,

- Find/display the original Reference voucher (e.g.10534),

- Press the **Landed Cost** button on the Voucher header; this will present the Landed Cost Maintenance form *(see below)*,

- do a <<Find>> of original voucher (e.g.10534),

- Note that the displayed Cost values can be changed if desired, but they would then no longer reflect the actual 'vouchered' landed costs,

- press 'Start' button to run the routine.
- Costs will be distributed to the purchased/received SKU's based on the following Application Control parameters:

Application	Type	Description	Value	Company
ic	bank_dist	Distribution type for Bank	Dollar (default), Fixed or Weight	default
ic	ins_dist	Distribution type for Insurance	Dollar (default), Fixed or Weight	default
ic	rmd_dist	Distribution type for RMD	Dollar (default), Fixed or Weight	default
ic	duty_dist	Distribution type for Duty	Dollar (default), Fixed or Weight	default
ic	frght_dist	Distribution type for Freight	Weight (default) Dollar or Fixed	default
ic	brok_dist	Distribution type for Brokering	Dollar (default), Fixed or Weight	default
ic	misc_dist	Distribution type for Misc.	Dollar (default), Fixed or Weight	default

- If all required vouchers have not been created, or if there is a problem with an entry, either the Start button will be disabled, or FLEXX will report a message as to the reason.
- The Landed Cost flag on original Voucher header will now be changed to 'c' (completed), and
- The Landed Cost routine cannot be run again for the same voucher.

On the SKU Master,

- Standard Cost and Average Cost are updated,
- Total Landed Cost will be updated on the SKU Vendor form,
- Est Factor fields will be updated so future Landed Cost values can be calculated for this SKU when actual Landed Cost amounts aren't yet known, and Total Landed Cost is to be calculated using this factor;
- these factors can be manually modified to recalculate Landed Cost prices for the SKU.

The **Price Update Process** (described in the next topic) can now be run to update the SKU Price table, to change the recorded prices based on the newly established Landed Costs of each SKU. *Please refer to the Price Update Maintenance description for more detail*.

9   SKU Price Update Maintenance Routine

The SKU Price Update Maintenance Routine works in conjunction with the Landed Cost Routine (see previous topic) to update the prices of **Inventory** SKU items whose landed costs have changed. Functionality exists through the use of Application Control, to determine which SKU items will need their prices updated. The Application Control definition type *"pricevar"* allows you to determine up to what percentage the costs of a SKU must change for it to be eligible for price updating. If the Average cost change equals or exceeds this base minimum percentage, the SKU item will appear on the Price Update form.

The Application Control can be defined as follows:

Application	Type	Description	Value	Company	Parameter
po	pricevar	Pricing Variance Margin %	-1 (default) or % value e.g. 2.0	Default	cost margin margincost

A value of -1 indicates to FLEXX that the Price Update function will not be used, so no cost change calculation will be performed.

The new price can be calculated by using either the actual cost change percentage or the cost margin percentage as defined on the Price Margin Table *(See topic 9.3)*, or both. This is also determined by the "pricevar" variable with the Parameter setting of "cost", "margin", or "margincost". The formulas for each Parameter used will be:

- **cost** - Projected Price = Base Price + (Base Price * Cost increase %)

where Base Price is the SKU Price with qty=1.0, Currency=company currency, Customer code=null, and the UOM is the Sell or Stock UOM.

- **margin** - Projected Price = New Avg. Cost/(1-Margin)

where Margin = the margin value defined in the **Price Margin** Table *(see Topic 9.3)*.

- **margincost** - both of the above formulas will be used to calculate new prices.

The Price Update Routine can be used to calculate new prices at three different times when purchasing a SKU; at PO Receiving, at PO Transfer to Voucher, or at running the Landed Cost Maintenance Routine *(See topic 8.0)*. It can be based just on estimate cost components (Landed Cost Maintenance has not been run) or on actual cost components (after the Landed Cost Maintenance has been run).

The Price Update Routine cannot be used to re-price **Non-inventory** SKU's. It is designed to work only with Inventory SKU's (those that have an Average Cost value).

If the SKU is a **BOM**, its price can be updated using the Price Update Routine after it has been 'built' using the Warehouse Transfer Routine in Inventory Control. The Update routine will use the new Avg. Cost with the existing Base price to calculate the new price using the above formulas.

Additionally, if it is desired to have the BOM price be updateable as each component (child) SKU is received/vouchered at a higher cost, the following Application Control setting can be used:

Application	Type	Description	Value	Company
ic	pricebom	Price BOM when cost changes?	Y	default

When Price Update is run and the SKU's to be re-priced are BOM "child" components, the Price Update Header will also display the BOM "parent" SKU allowing it to be re-priced as well.

Note that this Price Update procedure can be used after PO receiving, after PO Transfer to Voucher, and after Landed Cost is run. If the cost has changed by the *pricevar* value at any of these stages, the SKU will be listed on the SKU Price Update Maintenance form after each event.

Following is an explanation of the procedure.

9.1 SKU Price Update Maintenance

After the PO has been received or vouchered or Landed Cost has been run, the SKU Price Update Maintenance routine is run. Selecting the function from the Inventory Control menu will display the **SKU Price Update Maintenance** form. If any SKU's have been received or vouchered at an increased cost equal to or greater than the *pricevar* percentage (in Application Control), they will be listed on the Update form. Each SKU can then be re-priced as desired. The search can be further limited to only those SKU's that have price records of a specific Price Type.

For this example, the Application Control *pricevar* value is set at 2.0% with a *margincost* parameter.

After the PO was received (and/or vouchered), the Price Update Header will display the SKU's whose average cost has increased by 2% or more.

The **Request Type** will indicate the function that recorded the cost increase. At PO receiving or PO Transfer to Voucher, it will be "poreceive". If the cost change occurred by running the Landed Cost routine, the Type will be "landedcost".

The **Previous** and **Current** average costs are shown as well as the calculated cost change percentage.

The **Priced** field will be either "Y" or "N"; N indicating the record has not been updated, and Y that it has. FLEXX will automatically update this flag to "Y" after the update has been completed. It can be manually set to either value if desired. The table will by default only display the "N" records. If the "Y" records are also to be shown, do a FIND with the field cleared.

To update the price of a listed SKU, select the SKU record and press the **Price Details** button. This will display the **Price Update Details** form.

9.2 Price Update Details

In this example, the **Price Update Details** form will show the **Current Average Cost** as a result of the PO Receive just performed as well as the **% Cost Change** calculated. This value will be used to calculate the new price if Application Control *pricevar* parameter is "cost" or "margincost".

It will also list all **effective** prices for the SKU. These will include all Price Types, Customer specific prices, Quantity break prices, and Warehouse specific prices.

If the SKU is to be priced by cost *margin* (Application Control *pricevar* parameter is "margin" or "margincost"), the **Target Margin** field will show the margin value for that SKU as defined on the Price Margin table *(see Topic 9.3)*. Each SKU can be defined with a unique margin percentage.

The **Current** fields show the current price as defined on the SKU Price table, and once it has been re-priced, the margin of the new price.

Projected by Margin fields show the projected price and margin for the SKU to be re-priced at based on the margin percentage. The reason for the **Mrgn** value to be slightly different from the Target Margin value is due to rounding of the new price value. Note that if the Price Update is using the *cost* parameter, this column will be stippled. Using the above example, the projected price by margin calculation for cws-220 Type "c" is:

New Price = Current Cost/(1-Margin) = 32.455443/(1-0.02) = 32.455443/0.98 = 33.12

Projected by Cost fields show the projected price and margin for the SKU to be re-priced at based on the cost increase percentage. Note that if the Price Update is using the *margin* parameter, this column will be stippled. Using the above example, the projected price by cost calculation for the same SKU is:

New Price = Old Price + (Old Price * cost increase %) = 250 + (250 * 0.20546) = 250 + 51.365 = 301.37

The re-pricing process can be performed by individual price record or all records at one time.

9.2.1   SKU Re-Pricing

To re-price each price record **individually**, select the record and click either the **margin** price field (e.g. click on 33.12) or the **cost** price field (e.g. click on 301.37) depending on which price is to be used. This will copy the corresponding Projected value into the **New** price field. This New price value can also be manually entered or adjusted as desired.

To re-price **all listed** price records, click either the **Select All by Margin** or the **Select All by Cost** button depending on which price is to be used. This will transfer the corresponding Projected values into the **New** price field.

In this example, the **Select All by Cost** has been selected.

Once all prices have been re-priced, the actual update process can also be performed individually or in total.

To update the price record **individually**, select the re-priced record and press the **Process Price** button. To perform the update process on all listed re-price records, press the **Process All Lines** button. This will result in a new price record being created on the SKU Price Table at the re-priced value. Each re-price line will then also have the **Proc** flag then checked.

When the process has been completed, press <<Previous Form>> to return to the Price Update Header form to perform any additional updates of other listed SKU's. If all records on the Price Update Details form have not been re-priced and updated, or it is desired not to do the updating at this time, FLEXX will display prompt message: "Not all prices have been processed. Continue to exit? Y/N". Pressing No will leave the Details form displayed allowing the operator to complete the process. Pressing Yes will display another prompt message: "Mark the SKU's pricing as completed? Y/N". Pressing No will display the Update Header form but leave the selected record Priced status at N. Pressing Yes will display the Update Header form and mark the selected record Priced status Y. This may be the desired status since the user did not want to perform the update at this time (the cost change occurred at PO Receiving), but is to be performed at a later process; either at voucher entry or at Landed Cost completion. If the cost will again change at either of those two processes, the Update Header will display the SKU record for that cost change. There will always only be one entry for a particular SKU at any one time, which will be the latest cost change entry.

See the following example.

In this example, the average costs of each SKU have changed again; cws-201 at voucher entry and cws-220 at Landed Cost routine completion. Prices can now be updated using these values rather than the earlier PO Receive values.

9.3 Price Margin Table

The Price Margin Table is used to define the Margin percentages to be used in calculating the new SKU price after its cost has changed. It is only used and is required when the *pricevar* parameter is defined "margin" or "margincost". If Price Update is to be run using "cost" only, the Margin table is not used and does not need to be defined.

The table can be defined to have a unique **Price Margin** rate for each SKU (e.g. cws-200 Type "c" set at 25.0%), or it can be defined as general as desired. The only values required are Type and Price Margin. Note that all Price Types used on the SKU Price Table need to have a Margin defined, even if no updating is to be performed on a particular price. In that case, set the Margin to 0.0 for that price.

Using the above example:

- All SKU's defined Price Type "c" and not specifically coded will be priced at a margin of 50.0%

- All SKU's defined Price Type "d" and not specifically coded will be priced at a margin of 25.0%

- All SKU's defined Price Type "i" and not specifically coded will be priced at a margin of 60.0%

- All SKU's defined Price Type "o" and not specifically coded will be priced at a margin of 5.0%

- All SKU's defined Price Type "p" and not specifically coded will be priced at a margin of 20.0%

- Any SKU specifically coded will be priced at the margin specified; e.g.

    - cws-200 Type "d", Sale Type "sale" and quantity 10 will be priced at a margin of 30.0%

    - cws-200 Type "c" and Sale Type "sale" and qty 0.0 will be priced at a margin of 60.0%

Be aware that if the Price Update routine is run using the Price Margin calculation and the Price Margin table has not been defined for that price, FLEXX will fail the process.

9.4 BOM Price Updating

The Price Update function can also be used to update the price of a BOM (parent) SKU. Normally, the BOM SKU is "costed" at the time it is built, using the average costs of each of the "child" SKU's. If the BOM average cost has increased after the build completion (as a result of a 'child' SKU cost increase), the Price Update Header will show that SKU with its associated values.

Example:

- BOM cws-2000, parent of cws-201 & cws-220
- cost increase of 3.55322% from "bombuild" function

The Price Update Details screen can then be used to re-price the BOM SKU based on the *pricevar* setting and the percent cost change.
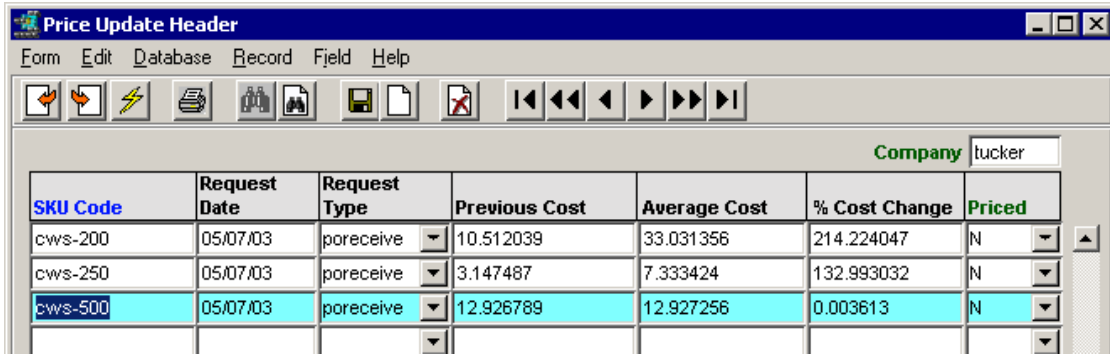
It is also possible through an additional Application Control definition to have the BOM price be update-able as each component (child) SKU is received/Vouchered at a higher cost (rather than just at BOM build). This requires the following Application Control definition:

Application	Type	Description	Value	Company
ic	pricebom	Price BOM when cost changes?	Y	default

With this setting, when any of the BOM component SKU costs increase at time of PO receiving, vouchering, or the Landed Cost routine is run, the Price Update Header screen will display those component SKU's as well as the parent BOM SKU. However, since the BOM has not yet been built using

these higher cost SKU's, it will not be re-costed but will only show its original Average Cost value. See the example below.

Example:

- BOM cws-500, parent of cws-200 & cws-250,
- 'child' SKU costs have increased at "poreceive" process,
- 'parent' BOM SKU shows marginal increase, virtually zero.

The Price Update Details form can then be used to manually increase the BOM price to the desired value. Then, when the BOM is built and the Average Cost increases, the Update Header will again display the BOM record with the cost change percentage. The Details form will then display the readjusted recommended update price based on the actual cost and the *pricevar* settings.

10 Inventory Adjustments

The **Warehouse to Warehouse Transfer** function of Inventory Control is used to perform manual stock Adjustments, along with the Transfer and BOM Build functions.

To make manual **adjustments** to stock quantities, enter Type *'ad'*, enter the affected warehouse in the To field (From will default to the same value), the SKU code (and Edition if an Edition SKU), and the adjustment Quantity (can be positive to increase inventory and negative to reduce inventory). Any reason can be entered into the Description field.

On saving the entry, the specified quantity will also adjust the SKU Inventory On Hand value by that amount.   The GL account adjustments will be made when the **Generate GL Tran (Inventory Movement)** from Inventory Control is run. FLEXX will use the GL SKU Code Table specified for the SKU to determine the accounts that need to be adjusted. It will use the Sale Type "adj" values if defined, or if not defined, the default Sale Type settings.

If the SKU is an Inventory item (Inventory flag checked), the GL adjustment will be made to the account defined on the GL SKU Code table as the *Inventory* account. If it is a non-inventory item, the adjustment will be made to the GL account defined as the *Expense* account.

11 Inventory Quantity Change Logging

It is possible to log all inventory quantity changes as they are performed in the various modules. The function is enabled by setting the following Application Control variable:

Application	Type	Description	Value	Company
ic	invntrylog	Inventory Log of Qty Changes	Y (N default)	default

With this setting, FLEXX will make an entry in the **Inventory Log of Quantity Changes** Table as inventory items are being used. This table will show the changes to each of the Inventory Quantity values, and the process that made the change. The quantity fields recorded will be On Hand, On Order, Committed, and Transfer. This process is useful in tracking inventory when there are concerns with quantity values being changed in error.
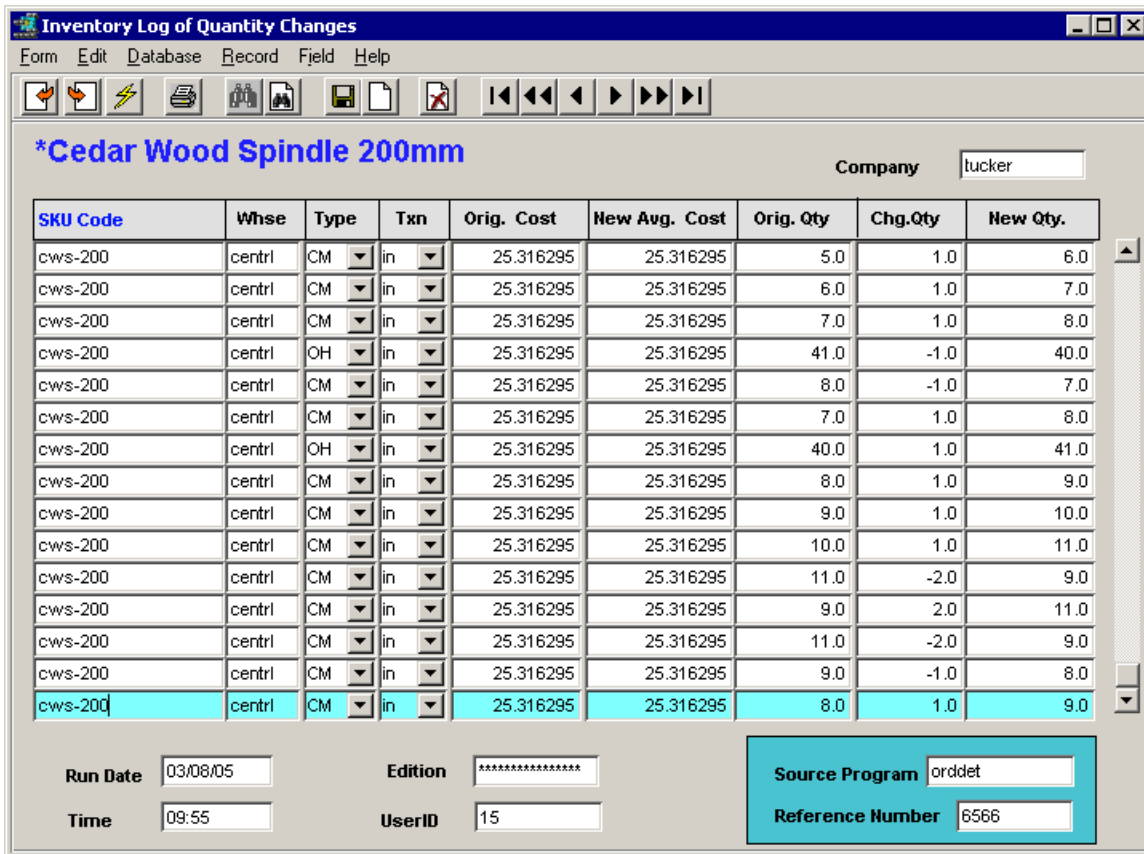
Type is the Inventory quantity field that was changed:

- CM – Committed
- NG - Negative Quantity
- OH – On Hand
- OO – On Order
- TR – In Transit

The **Cost** fields show the Average Cost of the SKU that was processed.
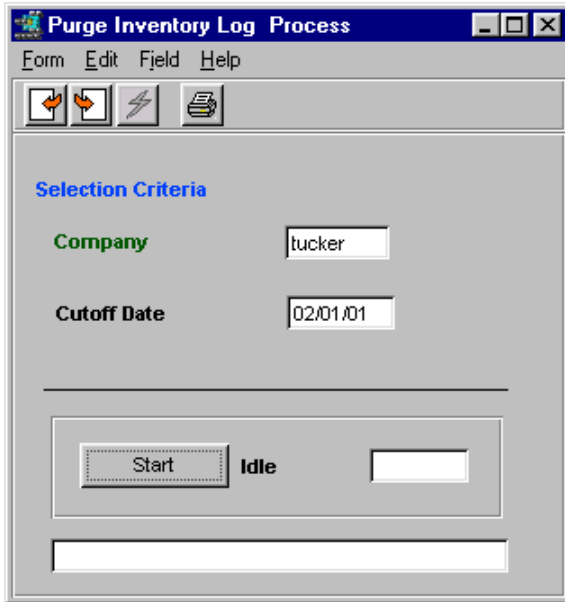
The **Quantity** fields indicate the values of the changes and results.

Source Program is the function that was performed to cause the change.

Reference Number is the source program transaction number, either Order, Purchase Order, Transfer, etc that caused the quantity change.

Be aware that this table can become very large in a fairly short period, and could thus result in system performance degradation. So it is recommended the **Purge Inventory Log Process** be run regularly.

It is also recommended to deactivate the function (set the *invntrylog* variable to N) if logging is no longer required. The function will require additional system resources when enabled and will impact system performance somewhat.

12 Warehouse Transfer via Inventory Control

The **Warehouse to Warehouse Transfer/Build/Receive** function can be used for the following warehouse management processes:

- Stock transfers between warehouses,
- BOM build process *(see Topic 3, Bill of Materials Function for details)*,
- Stock quantity adjustments for specific warehouses *(see Topic 10, Inventory Adjustments)*,
- Edition SKU condemns (deletes) from specific warehouses *(see Topic 4.5)*.

Warehouse **Transfers** are performed by specifying Type 't' and then entering the To and From warehouse codes along with the SKU code and quantity to be transferred. This decreases inventory levels at the From warehouse and increases it at the To warehouse by the entered quantity value. No other processing is required, and no cost values are modified.

FLEXX also allows for manual stock Adjustments and Edition Condemns by using this form. Adjustments to stock quantities of individual SKU's can be done manually using this function *(see Topic 10)*, or through the Stock Counting process, described in detail in Chapter 6.

13 SKU Copy Routine

The SKU Copy Routine is used to copy a SKU definition either within the same company or to another company defined to FLEXX. Depending on the selection parameters specified, the SKU can be copied to have all the definitions of the original SKU, or only certain selected definitions. It will perform any or all of the following functions:

- Copy an existing SKU to a new SKU;

- Copy various details of existing SKU to another existing SKU, including the Attributes and Miscellaneous forms;

- Copy the Inventory table to another existing SKU (note that the quantity and cost values are not copied);

- Copy the SKU Pricing table details to another existing SKU;

- Copy the SKU Vendor table details to another existing SKU.

Each of these processes allows further selections to enable more specific data to be copied. All functions can be run within the same company as well as across company. Be aware that when running cross-company, certain Master Type Table definitions will first need to be defined in the "To" company before the copy will be successful. FLEXX will display a prompt if the definition can't be found, and the copy will not be performed.

Be aware that the copy routine allows only a one-for-one copy to be created. A wildcard (%) cannot be entered into the SKU Code field.

Copy SKU?

- Attributes – copy all the settings of the Attributes form to the new SKU. This includes all the Attributes flags as well as the various field definitions on that form.

- Misc. Details – copy all the settings of the Miscellaneous form to the new SKU.

- BOM - copy all details of the SKU BOM Entry/Maintenance form This is only valid if the SKU is defined as a BOM and will complete only if all the defined "children" SKU's are already defined in the "To" company.

- Price Margin – copy all details of the Price Margin table.

- Substitutions – copy all details of the SKU Substitute Numbers table. This will complete only if all the defined "substitute" SKU's are already defined in the "To" company.

- Text – copy all Text data to the SKU.

Copy Inventory?

- Whse Code - the Warehouse data to be copied. If left %, all warehouses will be copied. Any single specific warehouse can be copied.

- Details – copy all the details of the specified warehouse.

Copy Vendors?

- Vendor Code - the Vendor data to be copied. If left %, all vendors will be copied. Any single specific vendor can be copied.

- Details – copy all the details of the specified vendor.

- L.C. Factors – also copy all the Landed Cost settings. This flag will only be in effect if the Details flag is also set. The L.C. settings cannot be copied to the new Vendor form by themselves.

Copy Price?

- Start Effective Date – specify the Starting effective date of the price records to be selected. This must be a valid date and cannot be %.

- End Effective Date – specify the Ending effective date of the price records to be selected. This must be a valid date and cannot be %.

- Type – specify the Price Type records to be copied. If left %, all Price Types will be copied.

- Whse – specify the Warehouse records to be copied. If left %, all warehouse specific entries will be copied.

- Customer – specify the Customer records to be copied. If left %, all customer specific entries will be copied.

- Currency – specify the Currency records to be copied. If left %, all currency specific entries will be copied.

- UOM Cat. – specify the UOM Category records to be copied. If left %, all UOM specific entries will be copied.

- Details – copy all the details of the specified Pricing Table.

Copy Edition?

- Edition - the Edition code for the data to be copied. If left %, all editions will be copied. Any single specific edition can be copied. This is only valid if the SKU is defined as an Edition SKU.

- Details – copy all the details of the specified edition.

14 View Image Function

FLEXX Inventory Control provides a process to store and display SKU images (pictures of SKU). This function requires a Web Server be installed and operating on the host system network since FLEXX uses the system web browser to handle SKU images. The function will support either Netscape or MS Internet Explorer as the browser.

 This also requires the following Application Control definition:

Application	Type	Description	Value	Company
all	webserver	Internet Web Server name.	web server name	default

The Show Image function on the SKU Master from will display the FLEXX Image Master form. This form is used to store as well as display the SKU image.

Multiple images can be stored for the same SKU. Images can be of the following file types only:

- jpeg – jpg or jpeg or bmp formats
- gif – gif format
- pdf – Adobe PDF format
- word – MS Word doc format
- excel – MS Excel xls format
- text – txt format

To store a new image, press <<Clear to Add>>, enter the File Type and Description and press <<Add/Update>>. Then enter the file path name of where the image is stored into the **Image Stored** at field and press Enter or Tab. This will activate the **Load New Image** button. Press the button to store the new image in the Image Master Table.

To display a stored image, select the image to be viewed (if more than one is stored) and press the **View Image** button. This will retrieve that image and display it on the user's default web browser.

Order Processing

15 Order Credit Hold

FLEXX automatically performs credit checking in OP as the order is entered according to the following criteria:

- **Credit Status** flag on Customer Master, Financial form (can be 0, 1, 2, or 3);
- **Credit Limit** amount on Customer Master;
- Application Control *creditchk* settings and Customer's **outstanding balance**.

An order will be put on **Credit Hold** (status 'cr') under the following conditions:

1. Credit Status flag = **0** (zero) to perform <u>normal credit checking</u>. This will result in the following controls:

   - If the customer has an **outstanding balance beyond the <u>Credit Check 1</u> value** (in number of days) as defined on the Application Control Table *(shown below)*, any new order **may** be put on Credit Hold in response to the prompt *"Customer has $xx.xx 'level1' days & $0 ' level2' days overdue. Put this order on HOLD?"*. This Credit Hold is set with Override, so can be Released from Credit Hold by an authorized FLEXX user. Authority is determined by the '*Credit Hold Release*' user authorization parameter and requires a password entry to be functional.

Application	Type	Description	Value	Company	Parameter
ar	creditchk	Credit check level 1 (days)	15 (default)	default	level1

   - If the customer has an **outstanding balance beyond the <u>Credit Check 2</u> value** as defined on the Application Control Table *(shown below),* any new order **will automatically** be put on Credit Hold (status 'cr') with the message *"Customer has $xx.xx 'level1' days & $xx.xx 'level2' days overdue"* displayed. This Credit Hold is set with Override, so can be Released from Credit Hold by an authorized FLEXX user.

Application	Type	Description	Value	Company	Parameter
ar	creditchk	Credit check level 2 (days)	30 (default)	default	level2

   - If the customer has Credit Limit value set and the value of the order together with any Outstanding balance exceeds that limit, any new order **will automatically** be put on Credit

2. Credit Status flag = **1** will put the customer <u>on Credit Hold</u> **with NO override** capability, resulting in the customer not being allowed to place any new orders.

3. Credit Status flag = **2** will put the customer on <u>Credit Hold</u> **with override** capability. Any new orders will now be automatically put on Credit Hold, but can be Released by an authorized FLEXX user (requires '*Credit Hold Release*' authorization).

4. If the customer has exceeded his **Credit Limit** as defined on the Customer Master Financial form, Credit Limit field, any new order **will automatically** be put on Credit Hold (status 'cr'), with message *"Customer balance exceeds credit limit; status change to credit hold"* displayed. This Credit Hold is set with Override, so can be Released from Credit Hold by an authorized FLEXX user.

5. Credit Status flag = **3** will <u>bypass all credit checking</u> for the particular customer. The Credit Limit value as well as the customer's outstanding balance will not be considered on any new orders being entered.

16 Order Pricing and Discounts

Order pricing is based on several factors:

1. The selling price of the item (SKU);

2. The specified discounts applied to this selling price;

3. The specified discount applied to the whole order;

4. The currency defined for the customer and the SKU.

The individual SKU selling price is defined on the SKU Master, Price table. Then, depending on whether or not discounts are allowed for the SKU, FLEXX will use the various specified discount codes to calculate the 'Net' price of the SKU as well as the 'Net' price of the order.

16.1   SKU Pricing

Individual SKU List Prices are determined by the values entered on the SKU Master Price form *(see Inventory Control Sec. 3.4 for more detail).* The list price may be separately defined by product, warehouse, order type, quantity break and/or customer. Any eligible discounts are calculated based on the effective Price values *(see Sec. 4.2 for more details on Discount processing).*

The SKU price is calculated based upon the following hierarchy:

FLEXX will attempt to use a price defined;

-   by price type for a given customer,
-   by price type for a given warehouse and customer,
-   by price type for a given warehouse,
-   by price type.

For each search, FLEXX will also use the Currency code value to find a match with the customer's currency setting.

Within this hierarchy any price quoted must be consistent with the effective date of the price record (i.e. a price is valid from the specified Effective Date until the Expiry Date) and the quantity being ordered. If FLEXX determines that more than one price is possible, the one chosen is that for the greatest break point.

It must be noted that without careful definition of the pricing matrix, anomalies are possible (several examples):

1. If a customer has a matching entry in the matrix, that price will be quoted even if it is more than a price for the customer's pricing group or even the normal price for the SKU. In general therefore prices should decrease as one goes through the hierarchy.

2. The quantity break point is higher in the hierarchy than the date. For example, a one year old unit price for greater than quantity ten of the item will be chosen over a two day old unit price for nine units even if it has a higher price. Therefore, if a special price is being added as an incentive, it should be compared to any existing price for a higher break point to check that a conflict will not be generated. This is not an issue if the price is being added at the same break point as an existing price; the new will override the old.

3. If the defined SKU price is to be used on an Order return, the Quantity field requires a negative value in the amount that will accommodate the number of items being returned. Effective date, price type, warehouse, and customer definitions will also apply so need to be considered. This is only significant on manually entered return orders. Return orders generated using the **Return** button on the Order Detail use the same price that was charged on the original order

16.1.1 Customer Price Type

The Customer **Price Type** is defined on the Customer Master Main form, Price Type field. This value will be the default used on the sales order for this customer, and can be changed on either the Order Financial form or each detail entry if required.

16.1.2 SKU Pricing Example:

Using the SKU Price Table below, following is a description of how FLEXX determines the effective List price:

Eff. Date	Expiry Date	Type	Whse	Customer	Qty	Price	UOM	Min. Price	Allow Disc
01/13/98	************	d	***********	*****************	1.0	50.0	EA	0.0	☐
03/16/98	************	d	main	*****************	1.0	75.0	EA	0.0	☐
09/01/98	************	d	main	*****************	1.0	90.0	EA	0.0	☐
09/01/98	************	i	***********	*****************	1.0	125.0	EA	0.0	☑
09/01/98	************	d	main	*****************	100.0	75.0	EA	0.0	☑
09/01/98	************	d	warehs	*****************	1.0	95.0	EA	0.0	☑
09/01/98	************	d	***********	*****************	-10.0	90.0	EA	0.0	☐
12/21/98	************	d	main	remco	1.0	100.0	EA	0.0	☐

Description: Domestic Pricing   Currency: USD   UOM Category: inv   Order Updated: 01/14/02   Order Type: rg

Entry	Description
1	Effective 1/13/98, SKU price is $50. for all '*d*' type customers , all warehouses, no quantity break, no discounts.
2	Effective 3/16/98, SKU price is $75. for all '*d*' customers ordering from '*main'* warehouse, no qty break, no discounts.
3	Effective 9/1/98, SKU price is $90 for all '*d*' customers, from '*main'* warehouse, no qty break, no discounts.
4	Effective 9/1/98, SKU price is $125 for all *'i'* type customers, all warehouses, no qty break, discounts allowed.
5	Effective 9/1/98, SKU price is $75 for all '*d'* customers, ordering 100 or more items from *'main'* warehouse, discounts allowed.
6	Effective 9/1/98, SKU price is $95 for all '*d*' customers ordering from *'warehs'* warehouse, no qty break, discounts allowed.
7	Effective 9/1/98, on any returns of up to 10 items, the credit price is $90.
8	Effective 12/21/98, SKU price is $100 for customer *'remco'* only, ordering from warehouse '*main'*, no qty break, no discounts.

16.2 Minimum Pricing

FLEXX allows SKU pricing to be set to a minimum value to be used in Order Processing. The value is set on the SKU Pricing table, **Min. Price** field.

This value is used at Order Detail entry where the minimum price for this SKU on any single order will be the **Min. Price** specified value. Note that this price is not the minimum price per unit of the item, but the minimum sales amount for this item.

Using the above example, when ordering this item, although the price for one unit is $300.00, the entry will be priced at $450.00. However, if ordering more than one, the price used will be 300.00 multiplied by the quantity, and if the Extended Price is larger than 450.00, the actual price will be used. See the example below.

Order Detail Example – Minimum Price SKU

Notice the different Extended Prices for the same SKU entered.

16.3   Discount Set Up and Processing

FLEXX OP discounting can provide virtually unlimited potential for customer/inventory/date and volume specific price adjustments. In order to qualify the Sales Order for a discount, certain criteria must be met.

16.3.1 Customer Discount Definition

To qualify a Customer for a discount, the Customer Master/ Main form must contain the eligible Discount codes, (i.e. Discount Type for item discounts and Order Discount for whole order discounts). These discount codes need to be defined in the Customer Discounts Table as described in detail later in this section. These two values are used as defaults on the Order Header/Financial form when a new order is created, and can be modified on the order if required.
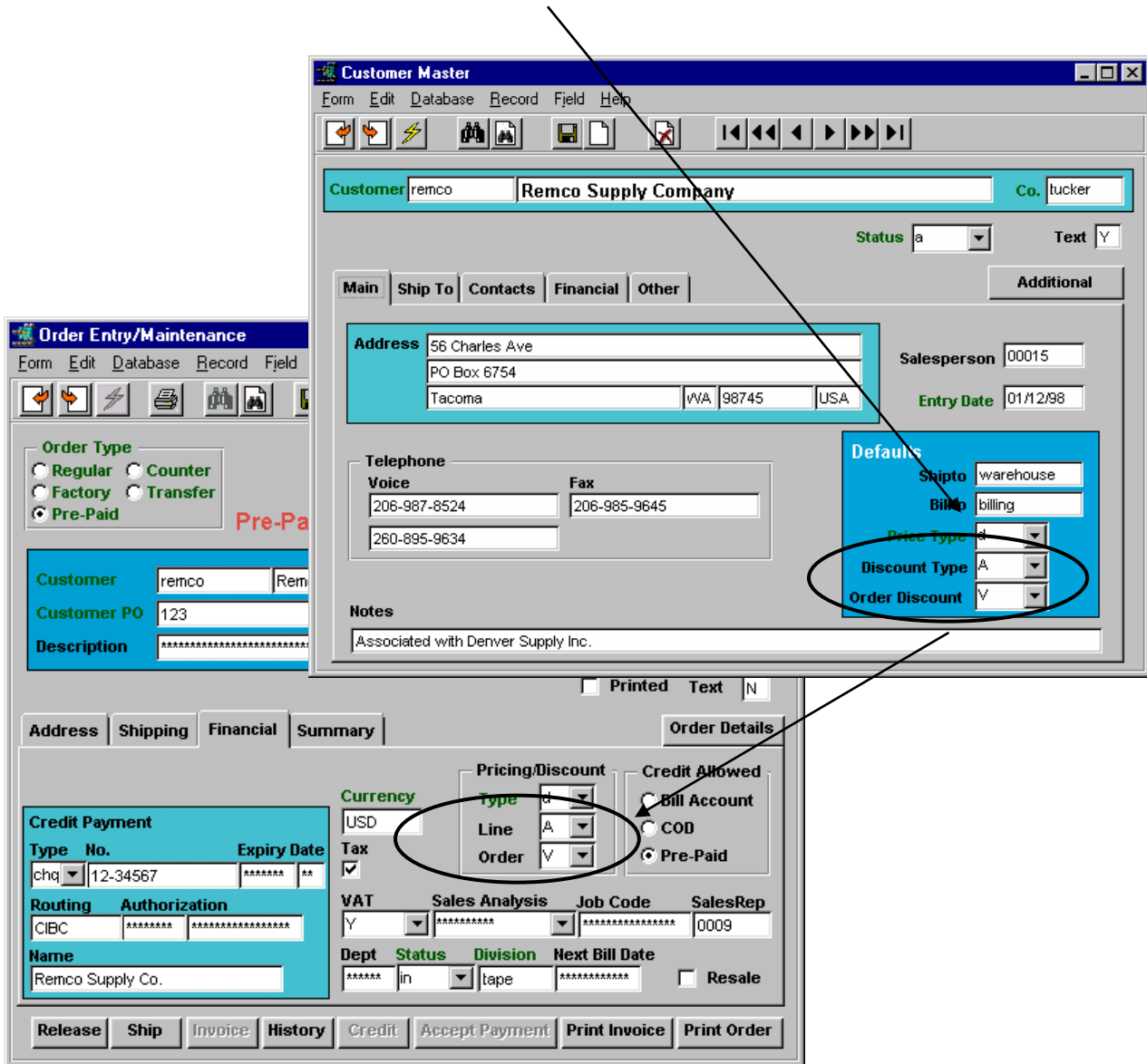
The above example indicates customer *'remco'* is eligible for **Item** discounts according to the Discount Type *'A'* rate, and **Whole Order** discounts according to the Type *'V'* rate as defined on the Customer Discounts Table.

16.3.2 Allow SKU Discounts

To enable discounts on an individual SKU item, the **Allow Disc** flag must be checked on the SKU Master/Pricing form. Additional discounts based upon the SKU are available on the SKU/Miscellaneous form via the **SKU Disc Cd** and **Disc. Attrib.** fields (though not required if only Customer Discount Type is to be used). These additional fields are then used on the Customer Discounts Table as additional qualifiers in defining discounts.

Note the specified **SKU Disc Cd** value of *P* and **Disc. Attrib** value of *99*.

16.3.3 Customer Discount Table

When the SKU in the above example (e.g. cws-200) is entered into an Order (Detail line item entry), Flexx will try to calculate a discounted price because of the Allow Disc. flag on the SKU Master entry. However, for FLEXX to know the Discount rate to use in the calculation, it will use the codes specified on the Customer Master (copied to the Order Header) and defined in the Customer Discount Code Table as described below.

1. Customer Master/Main form, Discount Type value (e.g. 'A' as shown on page 1 above) specifies the Type of discount this customer is eligible for;

2. Discount Type is defined in the Customer Discounts Table (Customer Master, Additional button, select Customer Discount Table);

3. The specific Code needs to be defined here with the assigned Rate (Multiplier or Constant).

This table now defines the following:

- entry 1: *remco* (defined as A) and on orders for SKU's with Disc. Attrib. 99 get a 50% discount (Multiplier = 0.50) per SKU item (described in more detail later), effective 02/01/98;
- entry 2: all 'A' orders of SKU *cws-500* get 30% discount (M = 0.70);
- entry 3: all 'A' customers ordering *'P'* coded SKU's get 10% on those SKU's;
- entry 4: *remco* (defined as A) will get a standard 20% discount;
- entry 5: all 'A' type customers will get a 5% discount (M = 0.95);
- entry 6 top form: all 'V' customers get a Whole Order Volume discount of 25% (M = 0.75) on defined Quantity and/or Volume values;
- entry 6 bottom form: all 'V' customers get a Whole Order Volume discount of $100.00 (Constant = 100) on a purchase Volume of $1000. or more.

Using the initial example of SKU *cws-200* and customer *'remco'* with the above Discounts Table*:*

- *remco* is defined with Discount Type A;
- *cws-200* is defined with Disc. Cd. P and Attribute 99;
- an order by *remco* for *cws-200* will result in a 50% discount for each item,
- plus, if the total order exceeds $1000, *remco* will get an additional $100. Whole Order discount.

16.3.4 Further Discount Considerations

1. Each Order item entry supports a second detail line item discount value that is manually specified on the Order Detail form (not copied in as a default) as per the following screen. This code also needs to be defined in the Customer Discounts Table.

The two values A and V are copied from the Order header Financial form defaults.

2. If the first discount field of the Order Detail (e.g. *'A'*) is specified (and defined), F*LEXX* further checks for *'SP'* discount codes defined in the Customer Discounts Table. This definition will instead be used, and will override all other discount definitions if defined *at a better rate* .

3. Whole Order Discounts can be shown as an embedded Net price on each Detail line entry, or as a Whole order total at the bottom of the order report by coding the *'disc_line'* parameter with a value of Y or N (*See Administration Guide, Appendix A for a description*).

4. Individual Line entries can be excluded from Whole Order Discounts by setting their individual Whole Order codes to Null. However, these items will still be used to calculate Order Volume for Volume discounting.

5. For Volume discounts, FLEXX uses several methods to determine the value:

    Volumes and Quantities are determined for:

    - the whole order
    - every SKU item
    - every SKU Attribute

    the appropriate choice is made for the type of discount record being analyzed.

6. When prices are recalculated for a discount, the number of decimal places in the price returned is set to the number of positions specified by the "decimal_pr" parameter as defined in the Application Control table. Values can be 0 to 5 *(See FLEXX Implementation Guide, Sec. 3 for more detail)*.

16.3.5 Discount Attributes function

The Discount Attributes option is another level of discount beyond the regular discount selections, as follows:

In the above table, all *remco* orders for SKU's defined with a *'99'* Discount Attribute, will get 50%.

So, the Discount Attribute can be used to give Special discounts at specific times. All SKU's that might be eligible for the Special discount could be assigned a unique Attribute, and then at the time of the Special discount, the Discount Table could be redefined with that Attribute's rate. However, be aware the SKU Disc. Code takes precedence over the SKU Disc. Attribute.

Using the above example, you would have two 'A' entries; one for all normal 'A' discounts with no Attribute discount, and another for the SKU's that also have the Attribute defined. Normal 'A' sales (non-special discount time) would use the non-Attribute discount rate and both entries would have the same rate defined. Then at Special time, the entry for 'Attribute' SKU's would only need to be changed, and all others still remain at the original setting.

16.3.6 Discount by Weight

This is a new function in FLEXX (Rel. 6.6L0) that allows the user to provide a discount by SKU weight. This is functional in both standard FLEXX as well FLEXX eCommerce.

The **Discount** table has been modified to enable this function, as shown:

All *weight-based* discounting is defined on the table with a Type = **W** and requires a **Weight Ratio Per** value entered. The **Weight** field is only used if there is a *minimum* weight specified.

Using the above entries, the "e" entry represents a $2.00 discount per 100 wt units (**Weight Ratio Per = 100)** with no minimum (**Weight = 0)**. Additionally, all SKUs that get this discount can be defined with a **SKU Disc Cd** set to the same value, and that code is specified on this table (e.g. "X"). This way the SKUs can be grouped by discount eligibility and do not need to be defined individually. Also, these SKUs need to have a Weight value defined on the SKU Master, Attributes form.

Whole Order discounting can also be defined by weight, as per the second Discount Table entry "v".

The "v" setting represents $1.00 per 100 wt (**Weight Ratio Per = 100**) but would only apply if the entire order was 1000# or more **(Weight = 1000)**. Again, the SKUs eligible for this discount are specified by the SKU Disc Cd setting matching the discount definition, and would be selected in this manner. And the customers eligible for Whole Order discounts would be specified by defining the **Order Discount** field with the corresponding code (e.g. "v").

SKU Definition Requirements

There are some specific requirements for SKU definitions on the SKU Master when using weight based discounting, as follows:

- Need to have a **Weight** defined,
- Must have the **Allow Discount** flag set on the Pricing table.

BOM SKU's;
- Invoiceable BOM, whether Inventoried or Non-inventoried;
    - **Invoice** flag On,
    - Needs to have a **Discountable** price defined,
    - Needs to have a **weight** value; FLEXX will not use the weight of the 'child' SKUs to determine the 'parent' weight,
    - Needs to have the **Tangible** flag On.
- Non-Inventoried and Non-Invoiceable BOM (i.e. BOM will be priced from the 'children' prices);
    - **Inventory** flag Off,
    - **Invoice** flag Off,
    - No price defined on 'parent',
        - Prices defined for all 'children',
    - No weight value; FLEXX will use the weight of the 'child' SKUs to determine the order weight,
        - Weight defined for 'children',
    - Needs to have the **Tangible** flag Off

16.3.7 Discount Method - Type M or L

Additionally, FLEXX allows one of two different discount *methods* to be specified. FLEXX will calculate the available discount for any eligible SKU using two different algorithms, M (Matching criteria) or L (Lowest possible price). The algorithm to be used is defined in the Application Control table as follows:

Application	Type	Description	Value	Company
op	*discount*	Discount Type M/L	**M** (default) or **L**	default

Where:

M - FLEXX will take the highest priority discount Matching the Discount Table parameters. This is the default and also the most common method used for calculating discounts.

L – FLEXX will use the available discount that will result in the Lowest sale price (greatest amount of discount) within the criteria.

This is described in more detail below.

Using the above definitions:

Discount Type = M - Following is the priority order for selecting the discount definition:

1. Customer –the customer specific definition for the particular discount;

2. SKU – the SKU specific definition;

3. SKU Disc Cd – the SKU Discount Code definition

4. Disc. Attrib. – the Disc. Attribute definition.

With the above settings, and assuming customer "harry" is defined with Disc. Type "A":

- if customer "harry" ordered "coke", the discount would be 3% (0.97 Multiplier), in fact, that would be the available discount to "harry" for any discountable SKU;

- any other customer ordering "coke", discount would be 8% (M = 0.92);

- any other customer ordering any other SKU;

  i. if SKU Disc. Cd. matches the SKU ordered, discount would be 7% (M = 0.93,) or

  ii. if Disc. Attrib. matches the SKU ordered, discount would be 5% (M = 0.95), or

  iii. if no SKU Disc. Cd. and no Disc. Attrib., discount available would be 2% (M = 0.98).

Discount Type = L - Following is the order for selecting the discount definition:

Using the same Discount settings;

- if customer "harry" ordered "coke", discount available would be 8% (Multiplier 0.92); even though the customer definition has a higher priority than the SKU setting and would only be a discount of 3%, the best discount available is 8%;

- if "harry" ordered any other SKU, and no SKU Disc. Cd. and no Disc. Attrib., discount available would be 3% (M = 0.97);

- if "harry" ordered any other SKU and

      i.   SKU Disc. Cd. matches the SKU ordered, discount would be 7% (M = 0.93);

      ii.   Disc. Attrib. matches the SKU ordered (and not SKU Disc. Cd.), discount would be 5% (M = 0.95);

      iii.   SKU Disc. Cd. and Disc. Attrib. match, discount available would still be 7% (M = 0.93);

-    if any other customer ordered "coke", discount available would be 8% (Multiplier 0.92) regardless of the SKU Disc. Cd. and Disc. Attrib. settings, since it is the highest discount value available;

If any other customer ordered any other SKU, the available discount would follow the same selection criteria as above for "harry", except the specific definition for "harry" would not be applicable (M = 0.97 would never be used).

16.3.8 Discount Hierarchy

The following chart shows the discount hierarchy FLEXX uses when Type M is specified.

Priority	Customer Disc. Type	SKU Disc. Cd	SKU Code	Customer Code	SKU Disc Attribute
1	*	*	*	*	*
2	*	*		*	*
3	*			*	*
4	*			*	
5	*		*		
6	*	*			*
7	*	*			
8	*				*
9	*				

Where:

-    * means SKU or Customer value equals the value in the Discount Table;

-    *blank* means that field must be left blank or null;

For the Discount rate to be selected, the specified discount **Effective Date** must be before or equal to the Order date, and the quantities and volumes (total dollar values) on the order must be equal or greater than that of the **Quantity** and **Volume** values specified on the Discount Table.

Records of the Same priority are further selected:

    First by Date – the line with the latest date is chosen;

    Second by Quantity – for records of the same date, the record with the highest quantity is chosen;

    Third by Volume – for records with the same date and quantity, the record with the highest Volume is chosen.

16.4   Order Pricing and Foreign Exchange (6.5L2)

FLEXX Order Processing also allows for foreign currency exchange calculation during SKU pricing. Both the customer's currency definition and the SKU currency setting are used to select the SKU price on the Price Table and to calculate the customer's price in his currency.

The SKU can be defined in either the company currency or the customer currency. If it is defined in the customer currency, no exchange calculation will occur at order entry and invoicing. However, the GL Transaction record generated from the sale will be converted to the company currency.

If the SKU is not defined in the customer currency but only the company currency, at time of order entry, the selling price will be converted to the customer's currency using the Foreign Exchange Table. See below for an example.

Note that this function REQUIRES the Foreign Exchange Table be defined with a rate for the different currencies being used. If not defined, FLEXX will fail this process.

16.4.1 Currency Conversion Example

1. SKU is defined in customer's currency

Using SKU cws-200 and Customer "cdnco", where

- company is defined USD currency, and
- customer "cdnco" is defined Price Type "c" and "CDN" currency, and
- SKU "cws-200" is defined with CDN at a value of 150.00 (no discount)

SKU Price Table

At order entry, FLEXX will select the SKU price based on the Customer Price Type and Currency (along with the criteria described earlier; e.g. warehouse, customer code, etc.).

Order Detail with Customer Currency

Notice the price selected was 200.00, the same value as is defined on the Price Table. This is because the price was defined for currency code CDN, the same as the customer's currency.

If the SKU price is defined with only the company currency, following are the results.

2. SKU is defined in company's currency only

Using SKU cws-201 and Customer "cdnco", where

- company is defined USD currency,
- "cdnco" is defined Price Type "c" and "CDN" currency,
- "cws-201" is defined with USD at a value of 40.00 (no discount)

SKU Price Table

At order entry, FLEXX will again select the SKU price based on the Customer Price Type and Currency (along with all previously described criteria). Since there is no definition at the customer's currency setting, FLEXX will default to the price defined for the customer's price type ("c") only *(see description below)*.

Order Detail with Company Currency Only

The SKU price for Type "c" is defined at 30.00 USD.

The Foreign Exchange Table is defined with a rate of 1.25.

The Selling price of the SKU in CDN currency will be 30.00 * 1.25 = 37.50

This will also be the value invoiced to the customer. The invoice will be generated in the customer's currency. However, at time of GL Transaction Generation of the invoice, FLEXX will convert the value back to the company's currency so the sale value will be posted to the GL in company currency numbers.

17 Sales Orders and Tax Calculations

FLEXX is designed to allow the user the ability to define numerous different tax rates that will be used when calculating the tax charges to be added to Sales orders. These taxes can be a Provincial/State Sales tax (P/ST) and/or a Value Added Tax (VAT).

The P/ST is defined at the customer level, based on their Ship To location. The VAT is defined at the SKU level only, implying that regardless of the Jurisdiction, this tax will be added..

17.1   Customer Tax Definitions

The Customer Master table is used to define the customer to FLEXX. It contains such data as the customer's normal mailing address, their various shipping addresses, customer contacts, as well as their financial information, as shown below. The Financial tab accesses the customer's financial definitions where the different taxes the customer is subject to are specified; **P/ST Payable** for Province/State Tax, and **VAT Type** for VAT. P/ST Payable is a **Yes/No** flag, and VAT Type is a tax **Type** code in that it can be any value other than Y, and where N is a No.

FLEXX can also be set up to default these settings at customer entry by using the following Application Control Table settings:

Application	Type	Description	Value	Company
customer	PST flag	Default PST flag	Y (N default)	Default
customer	GST flag	Default GST flag	Y (N default)	Default

17.2   Tax Jurisdiction Set Up

There are numerous tax considerations in FLEXX that require the correct definitions of the Tax Table together with the Tax Code Maintenance table described below.

If the customer is defined to be eligible for P/ST *(Tax Payable = Y)*, then each customer Ship To location (as defined on the Customer Ship To form) needs to be defined with the applicable **Tax Jurisdiction** codes. These codes refer to the values defined in the Tax Table where the different tax rates are specified for those ship to '*jurisdictions*'.   Each location can have more than one jurisdiction assigned to accommodate multiple sales taxes for that jurisdiction (e.g. Provincial, County, City, etc.). The rate set for each code will be used to calculate the tax charges for each item ordered. Each code will generate a separate tax detail entry on the invoice form *(described in Topic. 15.6)*.

This example shows customer *remco* is subject to taxation at the WA and TA defined rates for any sales made to the *JobSite* ship to location. Additional ship to locations can be defined, and each can be defined to use different Tax Jurisdictions.

17.3   SKU Tax Definitions

All items to be sold are defined in the SKU Master Table. For the purposes of this discussion, only the tax definitions of the SKU Master will be described.

As shown in the example below, the SKU tax eligibility is defined on the Miscellaneous form, **Taxation** fields. State Tax is used to define the P/ST **Type** code and VAT Code specifies the code defined in the Tax Table for the VAT rate to be charged on this SKU.

In this example, the sku *cws-200* is eligible for P/ST, and VAT at the GST7 defined rate. If the customer is also defined to be eligible for P/ST and VAT, tax at the rate defined for their Ship To location as well as VAT at the rate defined for GST7 will be charged on this SKU.

Note that the **P/ST** code is more than just a Yes/No representation. It is a **Type** code that can be any one-digit alphanumeric value and is used to further define the different rates of each Tax Jurisdiction specified for the customer's ship to location.

Example: State Tax code Y for Jurisdiction NY = 5.0%

State Tax code A for Jurisdiction NY = 5.5%

State Tax code B for Jurisdiction NY = 6.0%

Code N will however still define a No value. An explanation of the Tax Table definitions follows.

17.4   Tax Definitions

17.4.1 Tax Code Definitions

Each Tax Code to be used in FLEXX needs to be defined in the **Tax Code Maintenance Table**, as shown below. To access the table, either <<zoom>> on any Tax Code field, or select it from the Administration Menu.

Enter and Save each Tax Code as required. This table will need to contain the definitions of **all** Tax Jurisdiction codes used in FLEXX.

Once all required tax codes have been defined, click Next Form to display the Tax Table Information form (described in the next topic).

17.4.2 Tax Table Information Definitions

The **Tax Table Information** screen is used to define the rates for the various tax codes to be used in FLEXX, both P/ST and VAT. Each **Tax Code** can be defined with any number of **Type** codes for a specific rate. Note that the Type code is not necessarily a Y/N (Yes/No) code, but an alphanumeric code that can be used to assign multiple rates to the same Tax Code *(as shown in the previous example)*.

 In the above example, tax code WA is defined at 8.25% for type Y, and 0% for type N, and type A has a value of 10%. Similarly, GST7 (the VAT code) is defined at 7.0% and 0% for the Y and N types respectively. Each code is also assigned the GL account that FLEXX is to use to record the tax amounts in. If the tax is refundable, a Credit account can also be specified. Further, the VAT column is used to indicate to FLEXX whether or not VAT is to be charged on the P/ST amount. The tax amounts are recorded in the GL when the **Generate GL Transactions (Invoices)** process is run.

Please refer to the *Specialty Procedures Guide, Tax Functionality* for a description of the **Deductibles** function shown at the lower part of the Tax Table form.

17.5   Tax Jurisdiction Example

Using the above definitions as an example, following is a description of how FLEXX computes the taxes on a sales order.

An order is entered for customer *remco*, to be shipped to location *JobSite*. The Customer Master definition *(Topic 15.2)* for *remco* specified both P/ST and VAT eligibility, and Tax Jurisdictions of **WA** and **NY** for Ship To location *JobSite*. The order header will appear as shown below.

a.   Order Header, Address form

The Address form will show the specified Bill To and Ship To locations as taken from the Customer Master definitions. Only the Ship To is used for tax calculations.

b. Order Header, Financial form

The Financial form will indicate the customer's Tax eligibility, also taken from the Customer Master Table. Note that these values can be changed at time of order entry, before it is invoiced, if they need to be different for this particular order.

P/ST is flagged and VAT Type = Y indicates this order is eligible for both P/ST and VAT charges. The P/ST flag is a Yes/No flag, where the VAT code is a tax Type code as described earlier *(Topic 15.4)*.

Using the Jurisdictions specified on the customer's Ship To form for location *JobSite* (WA and TA)*,* and the rates defined in the Tax Table for WA and TA, the P/ST tax rates to be charged are 8.25% and 1.5% on any item (SKU) that is eligible for taxation (as mentioned in topic 15.3).

c. Order Detail entry

Entering SKU cws-200 (described earlier) which is defined P/ST = Y and VAT = GST7, both P/ST and VAT will be calculated for this item at the rates defined in the Tax Table. These tax charges will be added to the order cost at the time of invoice generation *(shown below).*

d. Order Invoicing, Invoice Detail form

After the order has been Released and Shipped, it can be invoiced. The Invoice Generation routine will calculate the tax charges to be added to the order, based on the defined rates for the various Ship To jurisdictions used.

Notice the tax charges added to the order amount:

GST7 – 7% on $100.   based on the VAT jurisdiction specified for the SKU;

WA – 8.25% on $100.   based on the P/ST jurisdictions …

TA – 1.5% on $100.   … specified for the Ship To location

Following is a display of the printed invoice P6372 showing only the tax charges:

18 Cash (Pre-Paid) Sales Orders

Any order can be defined Pre-Paid/Cash. This will require the order to be paid before if can be released and shipped. Customers defined as 'Pre-Paid' are **only** eligible to submit Pre-Paid/Cash orders. A Regular (Bill Account) customer can submit all order types; Bill Account, COD and Pre-Paid. A COD customer can submit only COD or Pre-Paid orders.

18.1   Order Entry

At Order entry, the order will either be predefined as Pre-Paid (as a result of the customer being defined Pre-Paid) or it can be designated Pre-Paid by selecting the *Pre-Paid/Cash* flag. This will also result in the Pre-Paid flag being set in the Credit Allowed fields on the Financial form. The Pre-Paid order will be in **Check Hold** status until it has been paid. Detail records can be entered as required, but the order cannot be released/shipped until it has been paid (i.e. is in "in" status).

On Pre-paid orders, the **Accept Payment** button will be active, and pressing it will present the Payment Entry form to enable the payment to be processed.

It is also possible to change a Pre-Paid/Cash order to Standard (Bill Account) order IF the customer is defined as a Bill Account customer. If the customer is defined Pre-Paid, he will only be allowed to place Pre-Paid/Cash orders which cannot be changed to any other type.

18.2 Accepting Payment

Upon presentation of the Payment Entry form, FLEXX will copy the financial data from the Customer Ship To form, which can be changed if required, and then automatically generate an invoice (e.g. P5335). The default Bank defined in the Application Control *"ppbank"* parameter will be shown in the **Bank** field, and can be changed if the payment is to be deposited in another defined bank. If a default Bank is not being used, a validly defined Bank code will be required before the payment can be Completed.

Press the **Complete Payment** button to process the displayed entry, or if changes are made, press Add/Update to save the changes. If the Balance Amount is zero, the payment will be processed and the invoice will be marked Paid.

The payment can also be split over several different types by entering the amount for each desired type (e.g. Visa, Cash, MC, etc.) and Saving that entry, as shown below:

Saving the final entry (when Balance Amount goes to 0.00) will complete the payment process, mark the Order as **Invoiced** (status will be 'in') and mark the corresponding invoice Paid. The invoice number will be displayed in the Invoice # field (e.g. P5335) and can be accessed by zooming on the number.

The **Cancel Payment** button can be used at any time BEFORE the final entry has been saved or Complete Payment is pressed, to cancel the displayed entry. Pressing <<Previous Form>> will then result in FLEXX prompt message: "Not all amounts are paid, Do you want to add a new payment record? Y/N". Clicking No will return you to the Order Entry form.

18.3   Invoice Printing

It is now also possible to print the invoice directly from the Cash order. Where the **Print Invoice** button was previously stippled, once the order has been paid (on return from the Payment Entry form) it will be lit, and on pressing it, will result in the specified 'auto-invoice' report to be printed.

The invoice can be printed at any time in the ordering process, after the payment has been processed; i.e. before or after Releasing or Shipping.

18.4   Pre-Paid Order Invoice

Invoice P5335 (below) has been generated from Order 5335. At invoice generation, Flexx will match the invoice number with the Order number prefixed with P if *'matchinvno'* is set to Y in the Application Control table.

 Also, the invoice will be marked as Paid (status *'p'*). The order can now be released and shipped as per normal OP order processing.

18.5   Accept Payment Automation

The Accept Payment process can also be automated using the Application Control *'sp_prepaid'* variable set to Y. This will then automatically perform all the order completion processes once the Complete Payment button has been pressed. This includes the following:

- Generate an invoice (AR),
- Mark the order Invoiced,
- Generate a receipt (AR), and mark the invoice Paid,

- Release and Ship the order, and
- Print the invoice (using the "auto-invform" coded report).

This eliminates numerous operational steps and greatly speeds the cash sales process for the operator.

19 Sales Order Returns

The FLEXX **Order Return** process is only functional on Invoiced orders, since it is possible to cancel the order before it has been invoiced. Order Returns can be processed in one of three methods:

- The original order is known.
- The original order is not known
- Bulk returns using the Order Return function

19.1   Original Order is Known

Access (FIND) the original order, select the Detail form and press the **Return** button for the selected detail entry. The following Return Order form is displayed.

Enter the **Return Quantity** if it is different from the displayed value, which will be the original order quantity.

Enter the **Return Warehouse** if different from the original shipping warehouse.

Press the **OK** button. This will automatically generate the return order, which will have as the return price the value of the item on the original sales order.

After the Return is completed, the Order Detail **Text** button will be lit indicating a text message has been generated. This text message will contain the order number of the generated Return order.

Order Text
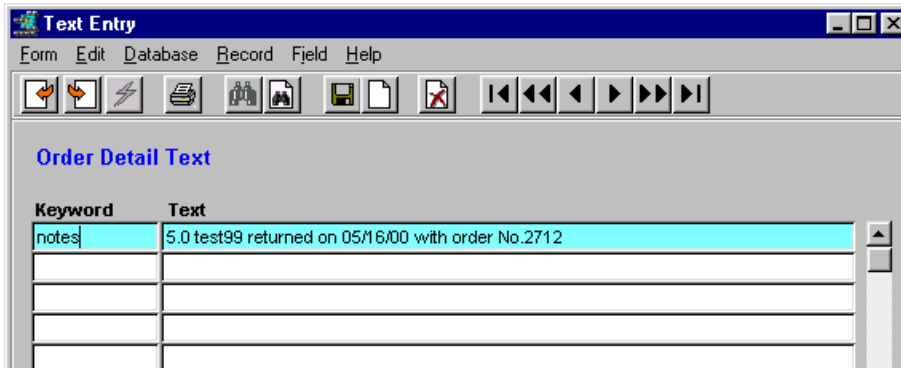
The generated return order will then need to be accessed and processed as per normal return order processing. As per above example, Order 2712 is the return order which now will need to be processed as described in Sec. 19.5 below.

19.2   Original Order is Not Known

If original order is not known, a new order with a negative (minus) quantity for the returned item will need to be entered. Create a new sales order for that customer, and on the Details form enter the minus (-) quantity for the item. If the SKU Master Price table has been properly defined to accommodate negative order quantities, that price will be used for the return price calculation. If this is not set up, then a return price will need to be manually entered into the order List Price field, and this is only possible by an authorized user (has *'Order Discount Price'* authorization defined in the User Authorization). When complete, this order will now need to be processed as per normal return order processing *(Topic 17.5)*.

19.3   Using the Order Return Function

The **Order Return** function selected from the OP menu can be used to make 'bulk' returns. By selecting the orders by customer and/or SKU codes will display all orders for that selection.
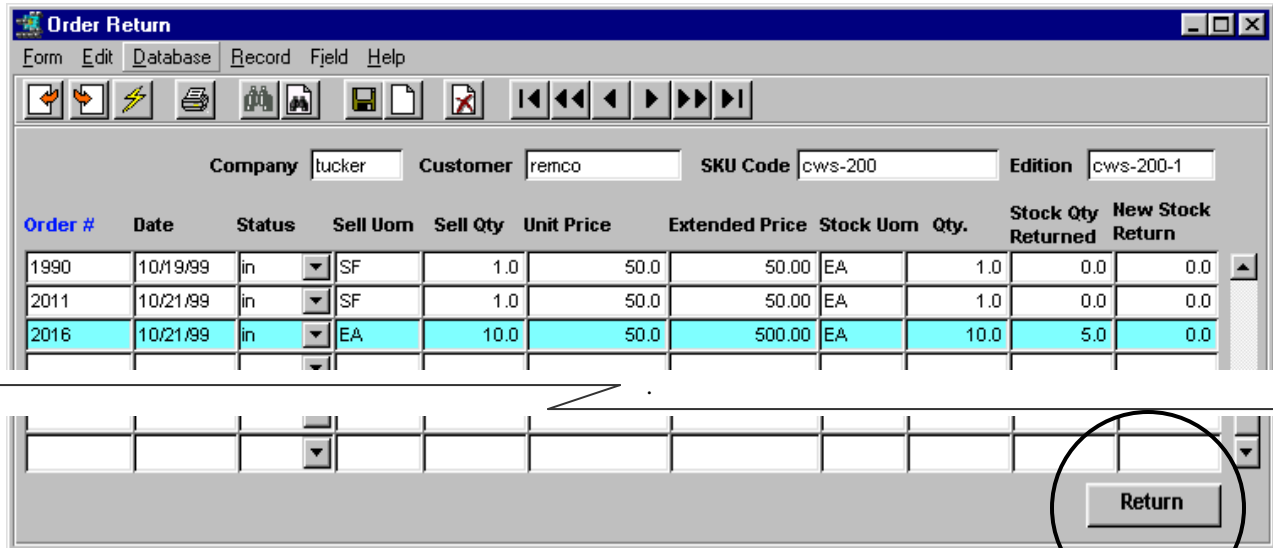
The specific order can then be selected, and if it has not already been returned, the **Return** button will be lit. Enter the return quantity in the **New Stock Return** field and press the **Return** button to return that SKU.   The Return Order process form (as shown previously) will then be displayed, and pressing OK will generate the return order, which will then need to be processed as per normal return order processing. All returns for the same customer will be entered onto the same return order.

19.4   Serialized SKU's

If the returned item is a Serialized SKU with serial Type "se", it can ONLY be returned using the Return Order function (either via the Return button on order detail or the Order Return function). This includes SKU's defined as RWEquipment on the SKU Master Attributes form.

When initiating a return for a serialized SKU, the Return Order screen will display a list of serial numbers originally ordered.

Enter the Return Quantity and mark the **Return Selected** box for each serial number to be returned and press **OK** to process the return. FLEXX will automatically generate the return OP order and on processing the return order, will update the Serial List table with the returned serial numbers being marked "ra" (Replaced/Available).

If the serial number being returned is different than those listed, implying a possible shipping error (wrong serial number shipped), and the error was earlier discovered and corrected in the Stock Counting Process *(see Stock Counting Process description, Replace process)*, enter the actual returned serial number in the **Replaced Serial Number** field and mark the **Return Selected** flag for the entry this is to replace. Press **OK** and FLEXX will generate the return order as well adjust the Serial List table with the correct values.

This function will only be available if the serial number discrepancy was earlier corrected in the Stock Counting Process. Otherwise, the Replaced Serial Number field will not be active and will not allow an entry.

19.5   Return Order Processing

The newly created return order now needs to be processed as follows:

a. **Release/Ship** – order must be released and shipped.

Since the order is a negative quantity, the release process will also automatically perform the Ship process, with the items being put into *'sh'* status. Release can be selected from the particular order, or by using the Order Release Process function from the OP menu.

b. **Invoice** – the order must be invoiced, whether or not it is for a Bill Account or Pre-Paid (cash) customer, either using the Invoice button on the order, or by using the Invoice Generation process from the OP menu.

The invoice process will produce a Credit Memo for this customer. This Credit Memo can then be processed as per normal Credit Memo processing described in the Accounting Modules Procedures Guide.

20 Sales Order Rush Shipment

If an order is to be rush shipped, the **Rush** flag needs to be set on the Order header Shipping form.

This will result in a RUSH line entry to be generated on the order Detail form.

A non-inventory '***RUSH***' SKU will need to be predefined on the SKU Master to allow for this entry, and the price defined for this SKU will be copied into the price field. Be aware that the RUSH flag will serve no other purpose than to cause FLEXX to create the RUSH order entry.

21 Sales Orders - Sell Unit Of Measure

FLEXX accommodates a <u>Selling</u> Unit of Measure different than the standard <u>Stocking</u> UOM. This allows the user to sell SKU items at different units than are being stocked, and to also Price those items at both the Selling and Stocking units.

21.1   SKU Master Definitions

The SKU Master Attributes form identifies the UOM values to be used in the **Units** fields (*see example*).

The values specified and their conversion factors need to be defined in the **Units of Measure** table *(zoom on UOM field to access)*, with the Category they are applicable to *(see example)*. The Stocking UOM (i.e. EA) should be defined with a Factor of 1.0 for the Sell UOM conversion to work properly. Be aware that each defined SKU Category can have its own unique Factor for the same UOM code thereby allowing for different conversion factors for the same UOM.

Then, the Price table should be defined for the different UOM values, Sell and/or Stock. Otherwise, FLEXX will use the Conversion Factor defined for that UOM (in the Units of Measure table) as a multiplier of the defined Stocking UOM price.

Be aware that if an undefined UOM is used at time of sale, FLEXX will be unable to assign a price to that item.

21.2   OP – Sales Order Pricing

The Order Processing module has also been modified to accommodate the **Sell UOM** function. Each Detail entry needs to have the Sell UOM value specified. The Sell UOM value defined on the Attributes form will be the default value used, and can be changed to another valid value for that particular SKU. If the UOM value entered does not have a specific price defined, FLEXX will use the defined Stock UOM price value as the List Price adjusted by the conversion factor for the order *(described below)*.

21.2.1 Order Detail Entries

Using the above example of SKU 'coke-2' *(as defined on the previous pages)*, the price for one Sell UOM (CASE) is defined as 150.00, and for one Stock UOM (EA) as 20.00 *(as shown on the SKU Master Price table earlier)*. If the UOM entered has a price defined, its price will be used, and the **Sell Priced** flag will be set. If, however, some other UOM is entered (i.e. *DOZ* which is not priced for this SKU), FLEXX will use the defined Stock price (i.e. 20.00) adjusted by the Sell UOM factor (12 for DOZ), and use that value to calculate the Extended Price, and not mark the **Sell Priced** flag. So, the flag can be used to identify whether or not the Sell UOM used has a price defined in the Price Table.

Be aware that if a non-priced UOM is used, FLEXX will use the Stocking UOM defined price with the Factor specified in the UOM table to calculate the Extended Price for that item. However, if there is no Stock UOM price defined, FLEXX will report the following error message, and a price will first need to be entered.
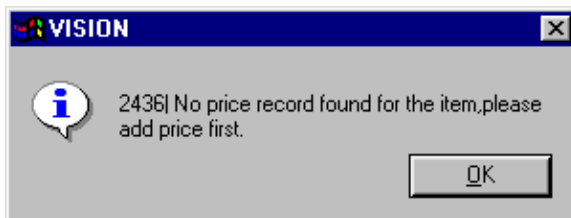
Additionally, the Stocking quantity making up the Sell UOM is indicated on the Order Detail form. This is taken from the UOM table, using the Factor defined for that UOM. Using the same example, the Sell UOM of CASE is stocked in EA units, and 10 units will make up a CASE.

Upon Release of the order, the resulting Shipment form will show the released (status 's') SKU's in their Stocking UOM quantities *(see example below).*

21.3   Shipping

The Shipment form shows on line 1 the 10.0 items of UOM EA as ordered, then 50.0 items of coke-2 where the order was entered as 5.0 CASEs, and 24.0 items ordered as 2.0 DOZ. This will also result in a shipping Piece Count of 84.0 items rather than actual quantities ordered.
 If desired, the Piece Count can be **Repacked** using the **Piece Count** button on the Shipment tab *(see below)*.

21.3.1 Auto Back Order Shipping

A function has been provided in Order Processing to allow for **automatic shipping of backordered quantities**. This feature can be used when actual stock on hand is such that there is sufficient quantity to ship the items even though FLEXX Inventory shows it as 'Insufficient quantity' (i.e. stock has arrived into the warehouse but not yet received into FLEXX Inventory). The Shipment form has an **Auto Ship** flag for each detail line. This flag is used to indicate to FLEXX whether or not to automatically ship the backordered quantities. *Please refer to the following example.*

The Auto Ship flag is set according to the following Application Control parameter:

Application	Type	Description	Value	Company
op	shipbo	Auto Ship bo, cf, and bp items	N (default) or Y	default

Note that this function will by default not be enabled
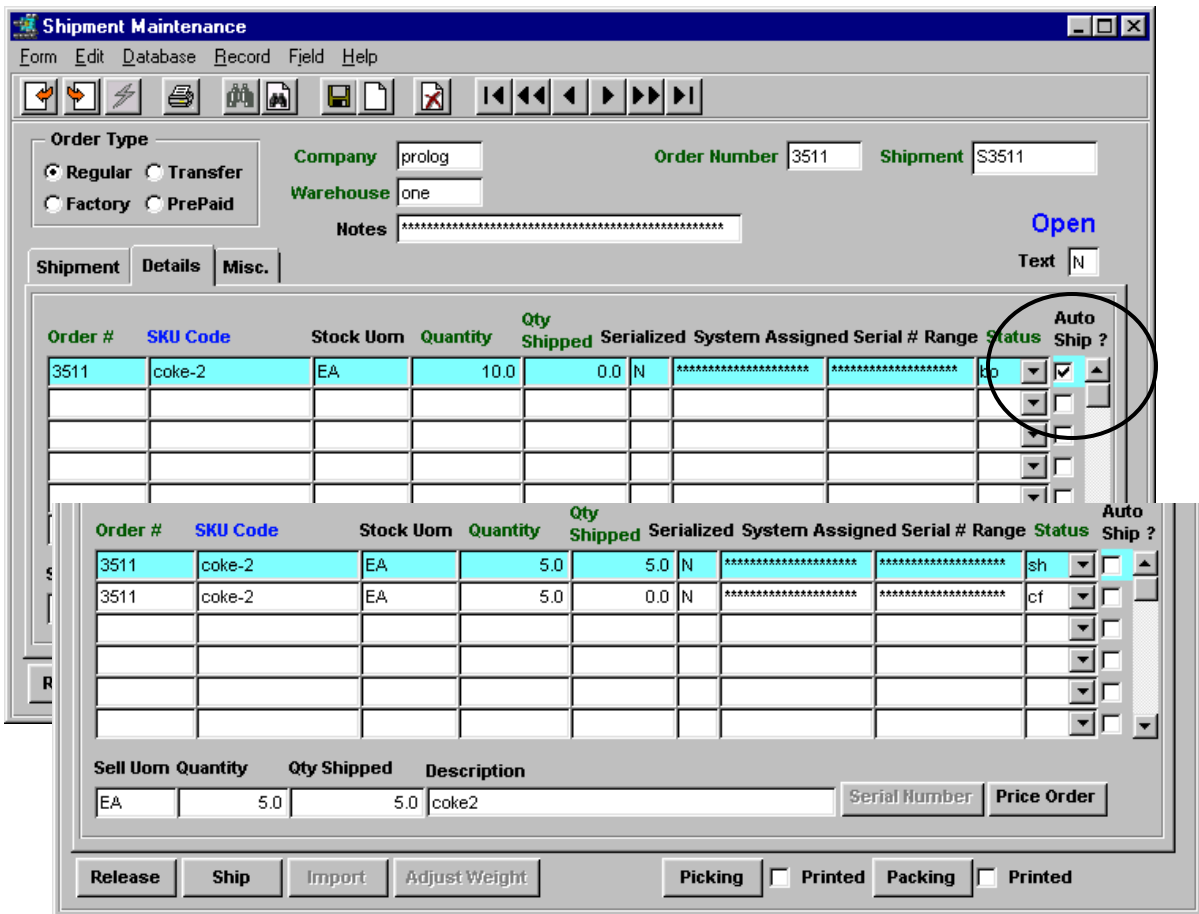
The above shipment, with the **Auto Ship** flag set, will automatically ship all the 'bo' items when pressing the **Ship** button. Any changes made to the Qty Shipped value will also result in the Auto Ship flag being turned off after saving the entry. If the 'bo' items are not to be shipped, the user will only need to turn off the Auto Ship flag and Save that entry.

Note that the function is applicable to all items in status 'bo', 'bp', and 'cf' and will automatically ship those items if the flag is set.

21.4   Invoicing

The resulting Invoice Detail will now also indicate the Sell UOM along with the Shipped quantity (Units) for each item being invoiced.   It will also have the **Sell Priced** flag to indicate whether or not the price used was a defined price.

In the above example, line 2, notice the invoice is for 50.00 units of coke-2 which compares with the order shipped quantity, and the **Sell** units is 5.0 CASE as was ordered. Also, the **Sell Priced** flag set indicates the price for the item was defined in the SKU Price table. Be aware, though, that the **Rate** value for the item is still that for the Sell UOM (CASE) and not for 50.0 units. This again compares with the defined price of the SKU in that UOM.

22 Order Shipment Adjustments

The shipping process in FLEXX Order Processing provides a means of adjusting the quantity shipped after the shipment has been completed (is in "sh"- Shipped status), but not yet invoiced. The **Shipment Adjustment** screen is provided to enter the quantity adjustment.

This screen can be accessed from either the OP Management Menu or by the use of the **Adjust Ship** button on the Shipment Maintenance form. Normally the process would be entered from the Shipment Maintenance form since that is the record the user would most likely be working from.

22.1 Shipment Maintenance Changes

Before the shipment is shipped (still Open – status "o"), the Adjust Ship button will be inactive (unlit).
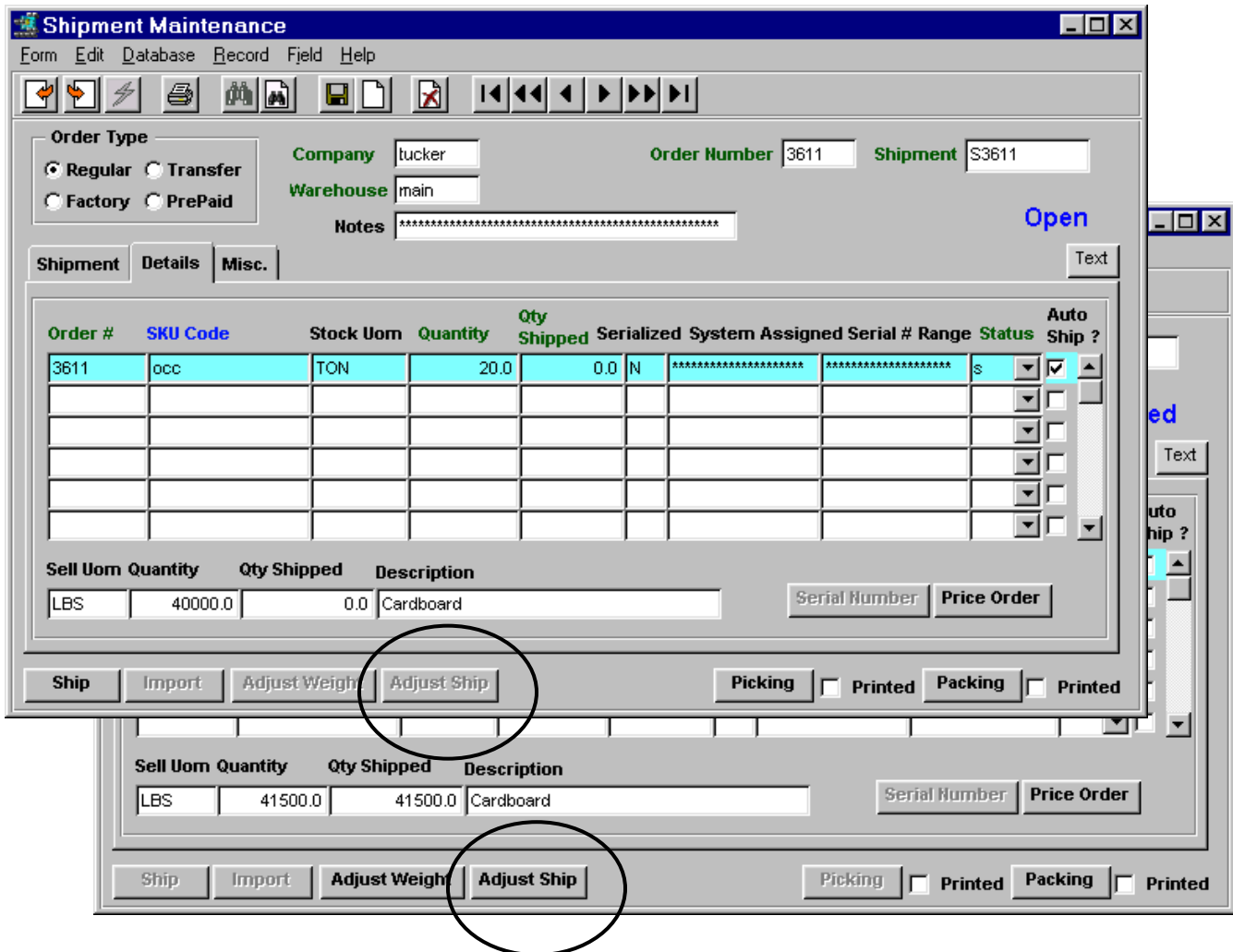
While in this status ("o"), quantity changes can be made to either the Stock or Sell **Qty Shipped** values. This is normal FLEXX operation. However, once the shipment has been shipped and is in Shipped ("sh") status, no further changes can be made on this screen. Also, the Adjust Ship button will now be lit, and pressing the button will present the **Shipment Adjustment** screen.

22.2   Shipment Adjustment Screen

The Shipment Adjustment screen can be used to make shipped quantity changes after the shipment has been shipped (status "sh"). You can change either the Stocking quantity or the Selling quantity values.

To change the Stock UOM quantity shipped, enter the new value in the Stock UOM **New Ship Qty** field. To change the sell quantity, enter the shipped value in the Sell UOM **Qty Shipped** field. Save the entry and that value will now be displayed in the Shipped Qty fields. The original shipped quantity will be maintained in the **Orig Ship Qty** field.

These values can be changed as long as the shipment is not invoiced. Once invoiced, the screen can still be displayed but will not allow any changes to be made.

If the quantity shipped is increased from that originally shipped, FLEXX will make a corresponding adjustment to the order detail entry showing the order quantity to be the quantity shipped. If the ship quantity is decreased from the original shipped, FLEXX will adjust the order detail entry showing the quantity ordered to be the new shipped quantity. There will be no "bo" or "cf" entry made to account for the quantity that was eliminated *(see example below)*.

FLEXX will also record the user name and id making the changes in the **Audit by** fields and the date the last change was made in the **Audit Date** field. This can be useful in tracking shipping adjustments.

Adjustment Example

Shipment adjusted *(decreased)* to 39000 LBS from 41500 LBS (Original qty 20.75 TON)

Resulting Order Detail form:

23 Factory Direct Order Processing

The Factory Direct function in FLEXX OP is used to order a SKU from the vendor and have it shipped directly to the customer. This is normally used for out-of-stock items, but can also be used for in-stock items; the process is the same. Following is a description of the process.

1. Create a new Sales Order, but check the **Factory Direct** flag <u>BEFORE</u> saving (Add/Update) the order.

2. On saving the order header, the Detail screen will display. Enter the SKU items required. The detail entries will be in 'fo' status, whether or not the items are currently in stock.

3. Once all items required are entered and all data is as desired, <<Next Form>> select **Factory Direct PO Generation** to display the PO generation procedure form.

Factory Direct PO Generation

Enter the required values (Order #, Buyer) and press Start. A PO will be automatically generated. Its number will be shown in the PO Number box.

It will also be shown on the Sales Order, Detail form - in the PO Number field of the Shipping tab.

And the just generated PO Detail form will show the number of the original sales order in the Order # field.

The generated PO will use the Vendor recorded in the SKU Master with the highest priority, and the Ship To location will be the sales order customer location.

4. If desired or necessary, it is possible to access the Factory Direct generated PO by either 'zooming' on the PO Number field, or exiting OP and entering Purchasing and selecting (FIND) the PO. Certain changes can be made to the PO as desired as described below.

If no changes need to be made to the PO, you can skip the following description and go directly to Step 5.

Factory Direct PO

Vendor changes:

Enter the desired vendor code in the Vendor Code field, and the vendor's data will overwrite the generated vendor data in the Mail To fields as well as any other vendor specific fields (Dates, Terms, etc.) transferred from the Vendor Master. Any vendor code entered must be predefined on the Vendor Master.

Be aware that any Vendor changes will likely result in the Detail cost values to be inaccurate and needing to be manually corrected. Further, any changes to the PO header form will also require a valid LOCAL Warehouse code to be entered in the Ship To field (more detail follows).

Customer changes:

By default, the Factory Direct PO **Ship To** address will be that of the <u>Customer's Ship To location</u> as defined on the Customer Master. If necessary, it is possible to enter a local warehouse Ship To code (as defined in the company Warehouse Table) if this order is NOT to be shipped to the customer but to the Company's warehouse. It is not possible to manually change only the Ship To address. The location can either be the generated customer Ship To location or a predefined Company Warehouse location. As stated above, if the Vendor Code has been changed, it will be necessary to also enter a valid predefined local Ship To code, implying this PO will no longer ship to the originally generated location but to a Company defined Warehouse location.

If these changes are required, and for the PO to still be shipped to the customer Ship To location, it is possible to define the customer Ship To location as a local Warehouse location, and then specified as the Ship To on the PO.

5. Return to the Factory Direct menu in OP and select **Factory Direct Receiving**, enter the PO number to be received, and press Start. On completion, the PO should now show the Detail entries in 'r' (Received) status.

6. On successful completion of Receiving, select **Factory Direct Shipping** and enter the PO number to be shipped, and press Start. The FD Sales Order will now show the Detail entries in 'fs' status (Factory Order – shipped). This will complete the FD order shipping and receiving process.

7. If desired, the Sales Order and PO can now be printed.

8. The PO and Sales Order are now handled by accounting the same as any other PO and Order would be.
    -   **Voucher** is created for the PO,
    -   voucher Marked for payment,
    -   Payment is generated for voucher,
    -   Check is printed for this payment.
    -   **Invoice** is generated, either from OP Invoice Generation function, or Invoice button on the FD order, or via AR manual Invoice Entry.
    -   Invoice is then processed via AR as normal.

9. The GL accounts used to record the sale and purchase amounts are those as defined for the Vendor and Customer, or if not defined, as set up in the Company or Co/Div Table defaults.

    To record the transactions in the GL, the various Generate GL Transactions functions would need to be run from both AR and AP.

24 Multiple Salespersons Per Customer

FLEXX supports the definition of multiple salespersons for each customer. The function is enabled through the following Application Control setting:

Application	Type	Description	Value	Company	Parameter
op	salesman	Salesperson Tracking Method	1 (default) or 4	default	method

(Note the **Parameter** field with a value of *method*)

The Value field can only be defined 1 or 4. A value of 1 indicates there is only one salesperson assigned and 4 indicates multiple salespersons can be assigned to any customers. The function is enabled when the value is set to 4.

Each salesperson defined in the Salesperson table will also need to be assigned using the **Salesperson Allocation Table**. The normal **Salesperson** field on the Customer Master will then not be used.

Each Salesperson is then also defined to a particular department (as defined in the Department table), thereby controlling which salesperson will be credited for the specific product sales. The **Dept.** code will be selected from the Dept. code defined on the SKU Master for each SKU as it is entered on a sales order.

FLEXX® 7.0L0 Procedures Guide - Distribution Modules

The salesperson for the item ordered will then be identified on the Order Detail form, as shown:

Each detail line will be individually identified as to the Salesperson and Dept. for that SKU.

Example:

- SKU cws-500 is assigned to Dept *sale*;
- salesperson *00015* is assigned to Dept. *sale;*
- customer *remco* (placed order) has salesperson *00015* assigned
- order for cws-500 will record Dept *sale* and SalesRep *00015*.

This information is now available for reporting. FLEXX does not use it in any other processes.

25 Warehouse Transfer Procedure via Order Processing

The Warehouse to Warehouse Transfer process in Order Processing is used to transfer SKU's between a company's defined warehouses when it <u>is required to track</u> the actual transfer items. The SKU(s) being transferred will be recorded on the SKU Master, Inventory table <u>In Transit</u> fields.

Further, if Picking and/or Packing slips will also need to be printed for the transfer orders, the following settings will need to be predefined (before the transfer order is created).

25.1   FLEXX Set Up

Application Control Table

Application	Type	Description	Value	Company
op	xfrcust	Transfer Order Customer	xfrcust (default) or any value (e.g. "transfer")	default

This setting indicates to FLEXX the customer definition to be used when creating the Picking and Packing slip reports.

Customer Master

The Customer Master Table will also need to be defined with an entry for the code specified on the Application Control "xfrcust" value (e.g. "transfer"). This customer definition requires no additional data, and can be set to status 'i' – Inactive to prevent it from being used for other functions.
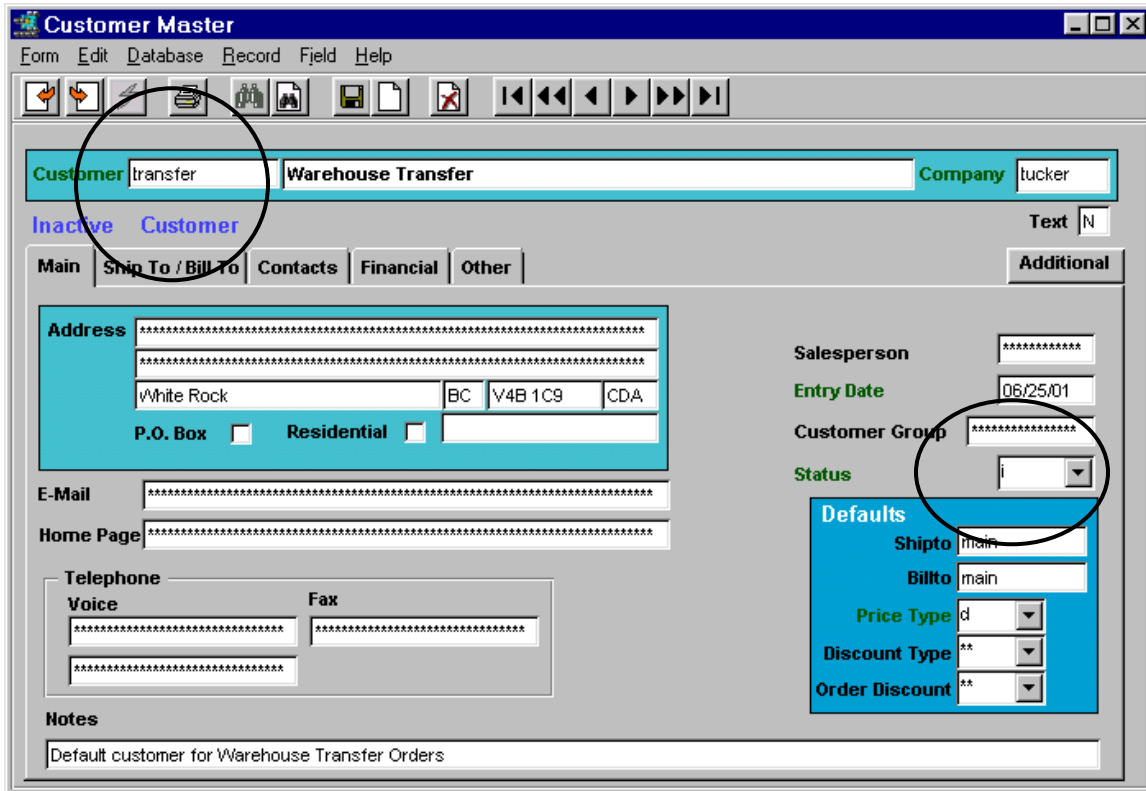
25.2   Shipping Warehouse

1. Create the transfer order using the **Transfer Order Entry/Maintenance** form (selected from the Order Processing menu);
    -   enter the From and To warehouse codes (must be predefined in the Warehouse table).
    -   make any other additions or changes as required

2. On saving the header, the Transfer Order Detail form will be presented;
   - enter the SKU code(s) and Quantities to transfer,

   Note: a warehouse entry must already be defined in the SKU Master Inventory table for this SKU to be transferred to the desired warehouse.

   - Save (Add/Update) the data
   - press <<Previous Form>> to return to header form.

3. At this time, the Order may need to be **Authorized.** This is performed by pressing the Authorize button (which will be highlighted). Authorization can only be performed by an 'authorized' user:
   - User Authorization *SKU Transfer Override* parameter, and User Password defined.

4. After authorization, press <<Next Form>> and select **Product Release Process** from the menu. The Product Release Procedure form will be displayed.

- **Start** the release process (will release only the just created T/O),
- On completion, make note of the Shipment Number generated (required in next step).

5. Ship the Transfer Order:
    - select the **Shipping Process** from the OP menu; this will present the Shipment form.

- enter the Shipment Number in the Shipment field and do a Find; this will display the T/O shipment record.
- Print the Picking slip (if required)
- Ship the order (press Ship button)
- Print the Packing slip (if required).
- Shipment is now completed at the 'shipping' warehouse.

The 'shipping' warehouse now ships the order together with the Packing Slip which shows the Shipment Number needed by the 'receiving' warehouse to process the order receipt.

25.3 Receiving Warehouse

6. At the receiving warehouse, on receipt of the transfer order, select **Transfer Order Receiving** from the OP menu,
   - enter the Shipment code of the T/O (from Packing slip)
   - do a FIND; will display the records from the Order Shipment form,
   - make any changes if required,
   - Save (Add/Update) – completes the receiving process.

The Transfer is now complete. The SKU Inventory table for this item will show the On Hand quantities adjusted; reduced at the 'shipping' warehouse and increased at the 'receiving' warehouse.

The **Inventory Movement Review** table will also record this transfer and show it on two lines; one entry will show the 'shipping' warehouse with a negative value in the Quantity field, and the second entry showing the 'receiving' warehouse with the corresponding positive quantity.

26 Zip Code Maintenance, Zip to Zone, and Carrier Rate Tables

These three tables are used in the order shipping process to determine the shipping rates for the different customer Ship To locations. With the following Application Control setting, FLEXX will use the Carrier Rate Table to determine the shipping rate for the specific carrier and Ship To location.

Application	Type	Description	Value	Company
op	courrate	Use Zip, Courier Rate for freight	Y (N default)	default

Set to N, FLEXX will not use the Carrier Rate Table, and method "best" will have no effect, and any freight code other than 'collect' and prepaid' will allow a freight charge to be entered on the Shipment form, Freight Amount field *(see OP manual for more detail)*.

26.1   Zip Code Maintenance Table

Select: Administration => System File Maint => Set Up File Maint => Zip Code Maint

   (or zoom on any address Zip Code field)

If Zip Code validity checking is to be performed, set the Application Control parameter *'zipcode'* to Y. All Zip codes used in address fields will then be verified against the **Zip Code Maintenance** table.

The Zip Code Maintenance table can be maintained manually (manual data entry) or electronically using the **Zip Code File Loading Process**. This table is also used by FLEXX , together with the **Zip to Zone** Table to determine the shipping Zone when specifying the *'best'* method for order shipping (as defined on the Customer Master or Order header).

The individual fields are self explanatory. If manually entered, they should be valid for the specified city. Electronically uploaded values will be defined according to the postal service zip code table.

26.2   Zip Code File Loading Process

This process is used to load the Zip Code flat file into the FLEXX Zip Code table. If all existing codes are to be deleted and replaced, check the **Delete Existing Codes** box. Normally, to update the table with additional new codes, the flag can be left off and any duplicate codes will be ignored.

26.3   Zip to Zone Table

Select: Administration => System File Maint => Set Up File Maint => Zip to Zone

The Zip to Zone table is set up to match the carrier's shipping zone with the 'Ship To' warehouse zip code.

Fields:

Whse – Company warehouse code as defined on the Warehouse table.

Courier Code – code defined on the Carrier Code table.

Zip Code – as defined on the Zip Code table.

Zone – zone code to correlate to this zip code.

26.4   Carrier Rate Table

The **Carrier Rate** table is used to define the Carrier <u>shipping rates</u> for the different shipping zones. It, together with the Zone table is then used to determine the shipping rates for the specific order. **It is also used to select the most economical shipping method if '*best*' is specified as the Freight code on the order header Shipping form** (copied from Customer Master Ship To form for this location).

Select: Administration => System File Maint => Set Up File Maint => Carrier Rate

Fields:

- **Courier Code** – user defined designation of the carrier (ups, fedx, etc.)
- Delivery flags:
    - **Commercial** – carrier will deliver to Commercial address (not flagged Residential)
    - **Residencial** – carrier will deliver to address flagged as Residential
- **Zone** – shipping zone as defined in the Zip to Zone table.
- **Weight Base** – Base minimum weight the defined cost is applicable for.
- **Cost Base** – Carrier Cost for base weight.
- **Weight Unit** – weight value for additional cost.
- **Cost Unit** – cost for the additional Weight Unit value.
- Cost Method flags :
    - **Flat amount** – flat rate Cost Base amount, cost is the value for that weight range for that Carrier and Zone.
    - **Cost per unit** – shipping cost uses both the Base and Unit rates.

Each carrier defined on this table must first be defined in the Carrier Code Master Type table (see next page).

26.5   Carrier Code Definition

All carriers to be used in Order Processing need to be defined in the **Carrier Code Master Type** table.

Select: <<zoom>> on Carrier code   => <<Next Form>> to Master Type Detail form.

Value	Description	Sys Status	Seq	Default	
fedx	Federal Express	fedx	5	***	***
oak	Oak Harbor	oak	10	***	***
puro	Purolator Courier	puro	15	***	***
Ryd	Ryder Truck Lines	ryd	20	***	***
Turr	Turner Freight	tur	30	***	***
ups	United Parcel Service	ups	50	***	***
usps	US Post	usps	60	***	***

Company: riken   Type: 1620

Weight Conversion Factor: 16

Suspend: ☐
Delivers to P.O.Boxes: ☐

Each carrier can be defined with additional operating parameters as follows:

Weight Conversion Factor – used to define a factor to convert the stocking/selling weight unit to the Carrier's shipping weight pricing unit; i.e. items stock in ounces, and shipped in lbs., factor would be 16 (16 oz./lb.). It is recommended that a value be entered for each carrier. Carriers that then deliver in the same weight units as the items are sold would require a value of 1.

Suspend - used to suspend the carrier from any shipping selection for all customers.

Delivers to P.O.Boxes – identifies carriers that will deliver to PO Boxes as defined on the order Ship To address; normally only the Postal service will do this.

27 RMA Processing

The RMA Process in FLEXX Order Processing is used to create an RMA (Return Merchandise Authorization) record when a SKU item is to be returned from a customer. The SKU can be returned to inventory on an OP **Return** Order or returned to be repaired or replaced on a Repair & Warranty **Repair** Order. This will then allow for added tracking and control of the returned SKU.

The RMA Process is not neccessary to return SKU's but can be used if so desired by the company. The normal Order Return process can be used if an RMA number is not required with the returned item.

A new RMA record is created as follows.

27.1 Create RMA Record

From the Order Processing menu, select the RMA Process entry. This will display a new RMA Process form in Find mode.

Press <<Clear to Add>> to put the screen into Update mode and enter the Customer code if known, and any other known values into the Search By fields.

Press the **Initiate Search** button. Records fitting the search parameters will be displayed on the Search Results subform (see example below).

- **Standard SKU**

Select the **Reason for Return** code from the drop-down list and add any data into the **Comments** field as desired. The **Return Whse** value will be the company default warehouse. This can be changed to any other defined value if the return is to be directed to another warehouse.

Select the specific item that is to be returned and enter the return quantity into the **New Stk Return** field. If it is a serialized SKU, check the **Select** box for the Serial Number to be returned.

Press the desired Return button (**Create Return Order** or **Create Repair Order**). Note that only SKU's defined as "RWEquipment" (on SKU Attributes form) can be returned on a Repair order.

An OP Return order will then be automatically generated, and the RMA will be assigned a number corresponding to the Return order number (e.g. if Order # 1234 was created, RMA will be RO1234).

27.1.1   Completed RMA

Item Receiving

When the item is received from the customer, do a FIND of the associated RMA. This can be done by doing a FIND on the RMA number if known, or FINDing all RMA's for a specific customer code. If these values aren't known, the user can use the **RMA History** function to list all RMA's entered for a specific customer (see Topic 27.3). Once the RMA is found, it will display the previously entered RMA record with

the Return Order Details subform (tab) displayed (see below).

Email Confirmation

The **Email** button can be used to send an RMA Request confirmation to the customer. The address used will be the value defined in the E-Mail field of the Customer Master for the RMA customer. Also, for the Email function to be operational, the header and signature text first needs to be defined on the Email Template Table *(Please refer to the FLEXX Getting Started manual, Topic Email Template for details)*.

Printing RMA Request Form

To print the RMA, press the **Print** button. This will print the RMA Request Form report which can be used to accompany the returned item during further handling.

27.1.2   Return Order Details

- **Standard SKU**

- **Serialized SKU**

Return Order Details shows the RMA SKU information with a status of "rh" – RMA Hold. The OP return order details will also be in "rh" status. If the SKU is serialized, select the **Select to Receive** box. Press the

Receive button to receive the item. This will update the RMA status to "r" – Received, and the RMA Detail and Order Detail status to "o" - Open. The associated order (Return or Repair) can then be processed according to normal procedures (e.g. Release, Ship, Invoice for OP).

If the return was through a Repair Order, the Repair Order Details subform would show similar data, but related to the RW Equipment SKU item. The Repair & Warranty process would then need to be completed.

This completes the RMA processing for a SKU return. Additional functions may need to be performed to complete the return process; e.g. create a replacement order to replace the returned SKU, complete the credit function if item was returned for credit or refund, complete the repair process if item was returned for repair, etc.

Following is an example of the automatically generated OP return order.

27.2   OP Return Order

The Return Order generated in OP will be tagged with the RMA number to allow it to be referenced to the RMA.

After the RMA has been received, the order will need to be processed the same as a manually entered sales order;

- Release
- Ship
- Invoice

27.3   RMA History

The RMA History function is used to display RMA records previously created. Pressing the RMA History button will display the **Customer RMA Review** screen. This form can be used to find an existing RMA request, or to find all RMA's that fit the search criteria entered.

The RMA can be searched by using any of the fields. The form is display-only and no entries or changes can be made.

27.4   RMA Reports

The following reports are available with the RMA Process:

- RMA Request Form – A standard RMA Request Form that can be used as a record of the item return.

- RMA Report – A report to list RMA's according to the entered selection parameters;
    - Open RMA's,
    - RMA's for a specific customer,
    - RMA's as of a certain date range,
    - etc.

28 Rapid Order Entry Process

The Rapid Order Entry function has been provided in Order Processing to enable the user to create an order or quote directly from doing a SKU price lookup. The screen is accessed from the Order Processing main menu by selecting the **Rapid Order Entry** menu item.

The fields that are required are:

- **Customer** – the customer code for the customer requesting the order; if not known, you can zoom this filed to display the Customer Search screen allowing a search for the desired code.
- **SKU** – the code or description of the SKU to be ordered.

Once these fields have been entered, press the **View Price/Inventory** button and the price and inventory records for that SKU will be shown, effective for that customer. The prices shown will include the following:

- All defined 'customer specific' prices effective for the current date, and
- All non-specific other than the customer's Price Type, but with the customer's currency code, and
- All warehouse specific prices effective for the current date.
- All prices will include the customer's defined discount if the SKU is flagged discountable.

If the prices to be displayed are to be only those defined for the customer's Price Type, enter that code into the **Price Type** field (e.g. "d" rather than the "%"), and only those prices will be listed.

After pressing the **View Price** button, the following SKU Price/Inventory Details screen will be displayed.

FLEXX® 7.0L0 Procedures Guide - Distribution Modules

28.1 SKU Price/Inventory Details Screen

The screen can be used as a price lookup as well as an order entry screen. The prices listed will be all those effective for the selected customer as defined on the SKU Price table.

The field **Last Price Paid** shows the price this customer paid on the latest previous order. This is taken from the Customer History table.

The operator will select the specific entry to be ordered and use the selection as shown or he/she can make changes to the **Qty** and **Discount Price** values as required for the order, as shown in the example below (e.g. Qty changed to 4 and Discount Price to 75.00).

Other than Qty and Discount Price, all fields are display only and will show the current status of the SKU.

The highlighted entry can then be selected for either order or quote entry by pressing the corresponding **Select For Quote** or **Order** button. In the same manner, additional SKU's can be selected for ordering. Both order and quote selections can be made concurrently. Notice the **View Price** button now indicates **Clear Price/Inventory.** Press the button to clear the screen and allow entry of another SKU selection. The Customer code can also be overwritten with a new code to list SKU prices effective for that customer.

Pressing either **Select** button will record the specific selection (together with any changes) on the **Rapid Order Details** screen. If the selection is to be for a Quote, a valid **Quote Section** will also need to be entered. All section codes that are to be used need to be predefined on the Quote Section Master Type table (Table # 2315).

Additional functions that can be performed on this screen are:

- **Customer Order History** – display the Customer Order Review form showing all previous orders by this customer for the specific SKU code.

- **Customer Quote History** - display the Quote Overview form showing all previously entered quotes for this customer.

- **Show SKU Image** – display the FLEXX Image Master screen to allow viewing the loaded SKU image

28.2 Rapid Order Details Screen

The Details screen will list all items selected for either Order or Quote generation. The **Rapid Order Type** field will show the type of selection the highlighted entry is for (Order or Quote). This can be changed if required.

Order Selection

Quote Selection

Further changes can be made on this screen to the **Qty** and **Unit Price** values as required. If the selected SKU is defined with a *minimum* price (SKU Master Price table), that value will be honored at time of order generation. This means that if the operator entered a special price for this customer, and that item had a minimum price that was higher, the minimum price would be used on the actual order.

The **Shipto Location** value is required and is copied from the default value defined on the Customer Master, but can be changed to another valid value.

Again, for a quote selection, or the selected entry has been changed from an order to a quote selection, a valid **Quote Section** code is required. All predefined codes will be displayed in a drop-down box list.

Once entries are complete, press the **Create Quote** or **Create Order** button to transfer the selections to a new quote in Quotation Manager or sales order in OP. If more than one detail entry is displayed, all entries will be transferred to the newly created quote/order.

If any of the listed records are not to be ordered, the user should delete those before pressing the Create Order button. Or, if the selected records are not to be ordered (i.e. customer has refused the order), the records should be deleted; otherwise they will remain on this table and may cause confusion in subsequent order requests.

If the operator made selections for multiple customers before creating the corresponding orders, the Details form will display a button labeled "Warning: Multiple Customer Orders – View All Orders", as shown below.

Multiple Orders

Pressing the "View All Orders" button will list all items that were previously 'selected for ordering' but have not yet been ordered (the "Create" button has not been pressed). The following screens show all the items selected.

- Customer "Remco" items

- Customer "Harry's" items

Note that these items will remain on the Details form until either an order/quote has been created or they have been manually deleted.

The operator can still create orders for any of the listed items by selecting (highlighting) that item and pressing the Create Quote/Order button. Or, if any of the additional entries are not required, they should be deleted to avoid further confusion with new order selections.

29 Print Shipping Labels

The Print Shipping Labels function in Order Processing enables the user to print multiple copies of shipping labels for any order Shipments. The function can be called from either the individual shipment or from the Report Selection List. The labels can be printed using either a specific label printer (e.g. Datamax E-4203) or a standard printer using label paper. If using a label printer, there will be some additional setup procedures required to make it function correctly with the FLEXX generated labels. This will be described in greater detail later in this topic.

Labels can be printed as often as required, and can be printed from shipments in any status.

From Shipment Screen

The Shipment Maintenance form, Shipment tab has been modified by providing a **Shipping Labels** button and a field labeled **# of Copies**.

The **# of Copies** value will default to 1, and can be changed to any desired value.

Pressing the **Shipping Labels** button will present the Report Scheduler screen. Use this screen to select the printer to be used for printing the labels. *See Topic 29.1 below, "FLEXX Setup for Label Printing" for defining the label printer.*

From Report Select List

Labels can also be printed using Report Select list in OP by selecting the Shipping Labels report *ship_labelc* from the list. The Report Parameters screen will be displayed.

Enter the Shipment number, the number of copies and specify whether or not the company logo is to be printed. If it is, review the required settings described in *Topic 29.1* below.

Further, if the logo is not required, you can specify a "comment" to be printed on the label using the **Label Comment** field. Note that if the logo is specified, the Comment field is not used.

Proceed as per normal report printing.

Report Scheduler Screen

As mentioned earlier, if this is a unique label printer, additional setup definitions will be required, and are described later.

Be aware that the **# of Copies** value specified on this form will not determine the number of labels to be printed. That value must be set on the Shipment form BEFORE pressing the **Shipping Labels** button. Otherwise it will default to the number of pieces on the shipment which is most likely not the number of labels desired.

29.1 FLEXX Setup for Label Printing

The following definitions can be set if required.

1. To print the Company logo on the labels, the logo file needs to be uploaded into FLEXX. This can be a bmp, jpeg, or gif file. This can be the same logo that is used for printing invoices, or it can be a logo unique to order shipment labels. This upload is performed from the Company Master form as follows:

    The Company Master has been provided with a **Company Logo** button.

Press this button to present the FLEXX Image Master form where the path and name of the logo image file is defined.

Enter the **File Type** from the dropdown list (jpeg or gif) and Save (Add/Update) the record.

Enter the full path and filename of the logo image in the **Image Stored at** field, and then press the **Load New Image** button. This will upload the image into FLEXX.

2. Defining the Default label printer for users who will be printing labels. If only certain FLEXX users will be printing labels on a specific printer, that printer can be pre-defined as the default label printer for those users, as follows:

   From the Administration Menu, Report Services/Control Menu, select Default User Report Printers. This presents the Default User Report Printers form.

Set the **User ID** to the users who will be printing labels, and the **Report Code** as shown. Note that both values should be defined. Define the **Printer** value to the value used in Master Table # 892 for the printer that is to be used for printing labels (described below). Note that this value can be changed at time of printing if desired on the Report Scheduler form, to any other validly defined printer.

29.2   Label Printer Setup

If the printer to be used for printing labels is a unique label printer (e.g. Datamax E-4203), it needs to be defined to FLEXX as a unique printer, other than the standard 'default' printer. This is done in the Master Type table # 892 as follows:

Access the table by either zooming on the **Print** field of the Report Scheduler form (shown earlier) or via the Administration Menu, System File, Internal File Maintenance and select Master Type Header. Do a FIND for Table # 892 and go to the detail form by clicking Next Form.

- Enter a code for the printer in the **Value** field.

- The **Description** needs to be set to the exact description of the printer from the system Printer Properties screen.

- The **Printer Port Name** needs to be set to the Port setting on the Printer Properties "Port" screen.
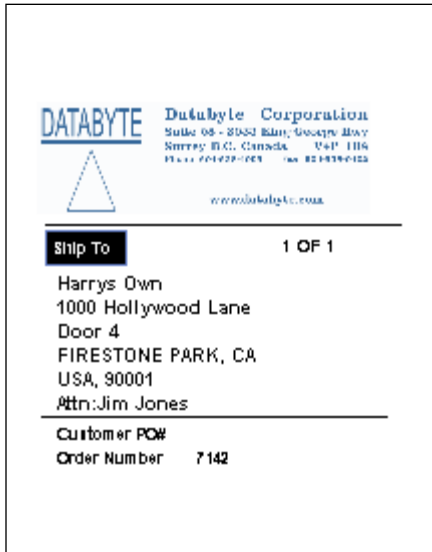
Note:
Be aware that these settings are ONLY required if the printer to be used for printing labels is a unique label printer, and it doesn't format the labels properly if printed using the standard 'default' printer settings.

29.3   Example of Printed Label

This is an example of a label where the Company Logo was specified to be printed.

DATABYTE Databyte Corporation
Suite 65 - 8863 Elm; Geogre Hwy
Surrey B.C. Canada   V4P 1H6
www.databyte.com

Ship To	1 OF 1

Harrys Own
1000 Hollywood Lane
Door 4
FIRESTONE PARK, CA
USA, 90001
Attn:Jim Jones

Customer PO#
Order Number   7142

Purchasing/Receiving

30 Purchase Order – Buy Unit of Measure

It is possible using the Purchase Order function, to buy in units different from the units the SKU is stocked in. This is especially useful when creating PO's for SKU's defined as '*divergence*', though the 'divergence' function has no effect on purchase orders.

30.1   SKU Definition

The buying Unit of Measure is defined on the SKU Master, Vendor form **Buy UOM** field (e.g. CSWT). Then, the conversion factor between the Buy UOM and Stock UOM is specified in the **UOM Conv. Factor** field. The Stock UOM value is that value defined on the Attributes form for Stock UOM Type (e.g. KG).

Using the above example, SKU 'test999' is normally purchased from vendor 'acme' in CSWT units, at 50.00 each. It is stocked in inventory in KG units, and each CSWT normally consists of 5.0 KG units; thus the **Conv. Factor** of 5.0, and the SUOM (Stocking UOM) Price of 10.00 (or 10.00 per KG).

Also, note the **Weight** field; this is the Buying weight of the SKU in Buy UOM units (e.g. 5.0 kg per CSWT) and is used by FLEXX in the Landed Cost calculation process (*described in the Landed Cost description of Inventory Control*).

30.2   SKU Purchase

The PO is completed as usual, using the defined **Buy UOM** in the UOM field (e.g. CSWT).

In this example, using the previous SKU definitions, where each CSWT is defined as being 5.0 KG, purchasing 4.0 CSWT results in the order also showing 20.0 **Stocking Units** (e.g. KG) being ordered. The PO Total **Weight** is the computed value using the order quantity and the weight value defined on the SKU Master Vendor form (e.g. 5.0). The vendor will then ship the 4.0 CSWT's of the sku, but they may not total the expected ordered Stocking Units (e.g. 20.0 KG). This is accounted for at the Receiving process *(more detail following)*.

30.3   PO Receiving

When the order is being Received (on pressing the **Receive** button), the **Order Receiving** form is presented showing the **Ordered** quantities.

If all quantities received are as ordered, pressing Add/Update for each detail entry or pressing the **Receive All** button will 'receive' those quantities into FLEXX inventory. However, any item can be received at quantities different from those ordered, either Buy UOM or Stock UOM quantities. This is accomplished by entering the actual received value into the respective **Received Qty** field.

Using the above example, 4.0 CSWT of SKU 'test999' was ordered, which equates to 20.0 KG of the item. In this example, we will receive the order by the **Stocking UOM** (KG) since each CSWT may not exactly consist of 5.0 KG.

We will receive 19.5 KG so will need to enter this value into the **Stocking UOM Received** field.

FLEXX will then present the following prompts:

Responding "No"

Responding "Yes" results in the following values to be generated on the Order Receiving form:

Notice the Stocking Qty received is 19.5 KG, and this value will be added to the Inventory On Hand value. Also notice the Buying Ordered and Received quantities are recalculated by FLEXX to correspond to the Received quantities. These values are for reference only and are not used by FLEXX in any further processing. Inventory levels are maintained in **Stocking UOM** values only and so the Buying values are not significant to the operation.

This will also result in the PO Detail to be modified to show the actual quantities received, as follows:

Notice the additional entry showing the non-received (cancelled) quantity of the order. This is normal FLEXX processing of cancelled PO items.

Also, the **PO Total** values (**Weight** and **Units)** will be recalculated to show the actual received quantities of the order.

30.4 PO Voucher

The voucher that is generated will be a normal voucher, showing the 'received' quantities. On creating the voucher, at the **Transfer PO to Voucher**, the Transfer form will be displayed as follows:

The displayed values will be the same as were generated on the PO Receiving form. Either each individual entry can be transferred by pressing <<Add/Update>> on each line, or the whole order can be transferred using the **Transfer All** button. The cancelled entry will not be transferred and will produce the FLEXX prompt "Cannot transfer a cancelled entry". The resulting Voucher Detail will show and price the 'received' items in **Stock UOM** units (e.g. 19.6 KG).

31 Stock Ordering Process

The Stock Ordering Process can be used to replenish inventory stock and allows you to automatically generate a Purchase Order or append to an existing Purchase Order, for stock items that are below their designated Minimum Stock Levels.

The Stock Ordering Process can be selected from a PO, either existing Open or newly created, or directly from the Purchasing menu, and functions somewhat differently depending on how it was selected.

Enter the required data according to the following criteria:

1. *Show Zero Qty* – deselect this flag if zero quantities are not to be displayed. This box will default to selected and will display all items that meet the following criteria and suggested order qty is 0.0.

2. *Stock Level* values can be specified as to which items are to be retrieved, and are determined by the SKU status values on the SKU Master <u>Inventory</u> form:

    A – Out of Stock, Orders Pending

    B – Out of Stock, not on order

    C – Stock below minimum level

    N – Normal (Stock at or above minimum)

Enter the selections and press the **Retrieve Items** button. The *Suggested* and *Order Now* values will be populated on the table according to the following criteria.

3. *Suggested* value is the quantity required to bring the stock level for the specified Warehouse to the defined maximum level (Max Qty value on SKU Master Inventory form).

4. *Order Now* quantity is the quantity to be ordered and cannot be larger than the Suggested value since that would bring the stock level above the defined maximum. The order quantity can be the displayed and suggested value, any smaller value, or a value of 0 (zero). Press Add/Update (or Enter if Rapid Entry is selected) to process the selection (described below).

To order more than the Suggested amount, several options are available:

    a. Enter the Suggested value into the Order Now field and then once the order has been completed, go into the PO Detail form and increase the order quantity as desired;

    b. Bypass the particular SKU to be ordered over max, and then once the Stock Order is complete, go into the PO Detail form of this order and manually enter the order for that SKU item;

    c. go to the SKU Master Inventory form for that SKU (zoom on SKU code) and change the Max Qty value to the amount desired, and then on return to Stock Order, rerun the Retrieve Items function.

31.1   Selection from PO

1. On an existing or new PO, press the **Stock Order** button.

2. The Vendor code on the Stock Order form will show the defined vendor of the PO and be stippled out so it cannot be changed at this point. The vendor can only be changed by creating a new PO for that vendor.

3. Since all items will be ordered from the same vendor, each selection can be Saved (or Entered if Rapid Entry set on) in succession (no FLEXX intervention), and those items will be entered into the selected PO Detail form.

4. Once the Stock Order process is complete, it is still possible to go into the PO Detail and make additions/modifications as desired.

31.2   Selection from Purchasing Menu

- All Selection Criteria and Stock Level fields can be entered as desired, and the items retrieved will be based accordingly.

- Rapid Entry can be flagged so as to complete each item selection when the Enter key is pressed.

- Upon Add/Update (or Enter if Rapid Entry set), FLEXX will present the message

      You must create/append a PO.

- On pressing **OK** on this message, FLEXX will present the PO header in *Find* mode;

  o If there is already an Open PO for the particular vendor, that PO will be displayed. Press <<Previous Form>> and FLEXX will use that PO for the selected item (PO number will show in the Purchase Order field of the Stock Order form).

  o If no currently Open PO for this vendor, create a new PO. Press <<Clear to Add>>, enter all relevant/required data including Vendor code for this item. Press <<Add/Update>> and FLEXX will return to the Stock Order form.

  o To use another Open PO for this vendor, do a <<Find>> for the specific PO. Press <<Previous Form>> and FLEXX will use that PO for the selected item (PO number will show in the Purchase Order field of the Stock Order form).

- The item just selected will have the **New** flag turned off indicating it has been ordered (entered on PO Detail which can be verified by zooming on the Purchase Order field).

- Select additional items as desired.

- On each subsequent selection, FLEXX will present the message **"You must create/append a PO"** and you will need to select the PO to be used for this order. At each return to this Stock Order form, before you make your next selection, you can enter the PO number being used for this order into the Purchase Order field and that PO will be selected. This process only works this way when selecting Stock Order from the Purchasing menu, and does so since Stock can be ordered from various different vendors and so POs need to be selected for the specific vendor.

- Again, once the Stock Order process is complete, it is still possible to go into the PO Detail and make additions/modifications as desired.

31.3   Stock Order Completion

Although the PO may still be in 'o' (Open) status, once it has been Printed (PO Printed flag on, and Print button stippled), it is no longer possible to use this PO for any further Stock Ordering process. This is also true for adding/modifying PO Detail entries. The PO is considered 'closed' when it is printed. If changes are still to be made to the PO, you would first need to turn off the Printed flag which is accomplished by running the **PO Print Status Update** routine.

The PO must now be processed as per normal PO handling as described in the FLEXX Purchasing/ Receiving manual.

32 PO Accruals

FLEXX provides a function whereby accrued Purchase costs can be posted to the GL.   To implement this function, the Application Control table needs to be defined for your company as follows:

Application	Type	Description	Value	Company
pur	poaccrual	Enable PO Accrual to GL (Y/N)	Y (default N)	default

Then, to indicate to FLEXX which GL accounts the funds are to be posted to, the Credit and Debit accounts need to be defined for each SKU as follows.

32.1   Accrual GL Accounts

32.1.1 Accrued Debit Account

In Inventory Control, access the SKU Master, Miscellaneous form, and zoom on the GL SKU Code to display the following form:

The defined **Inventory Account** will be the Accrued Debit account at time of posting. Be aware that FLEXX will only accrue costs on Inventory items, so only the Inventory account is used.

32.1.2   Accrued Credit Account

A new Account field named **PO Accrual** needs to be defined on the Company Master. This will be used by FLEXX as the PO Accrued **Credit** account at time of posting *(see example below).*

32.2   Posting Accruals

The Accruals will not be automatically posted to the GL at time of PO Receiving. To generate the accruals in the GL, the **Generate GL Transactions from PO** function needs to be run whenever the accruals are to be posted. This can be performed as often as desired, i.e. daily, after each PO, or as desired. Be aware that only after running this process and subsequent **GL Posting** will the costs actually be accrued to the GL.

32.2.1 Purchase Order Accruals

The PO Detail form, **Financial tab** will now also show the Accrual GL accounts that FLEXX will use to post to (*per example*).

These account numbers are automatically entered by FLEXX at PO creation using the defined values as described earlier. However, if necessary, these values can be changed at any time prior to running the Generate GL procedure.

The PO must first be Received before the Generate Transactions procedure will do any posting, since costs really aren't accrued until the purchased item is received.

Once the Accruals have been posted, the Period it was posted in will show in the GL Period field, and the transaction number will be recorded in the GL Trans field, and Post will be 'p'.

32.2.2 GL Transaction Posting

After running the **Generate GL Transactions from PO** procedure, the GL will contain the generated transactions as follows:

GL Transaction Detail

Note: don't be concerned with the dollar amounts of this example; the test used a purchase with CDN and USD currency, and the exchange rate was defined as 2.0.

33 Purchase Order Complex Shipping

The Complex Shipping function in PO involves either an Intermediate ship to location before ultimate delivery to the 'Company', or an alternate Supplier (Vendor) pick up location of the order, other than the Vendor's main address. Both alternate addresses can be used on the same PO, and FLEXX will assign both to the PO report generated.

33.1   Intermediate Ship To Location

The intermediate location is defined as an Alternate Address on the particular Vendor Master form using the location code *'interaddr'*. See the following example:
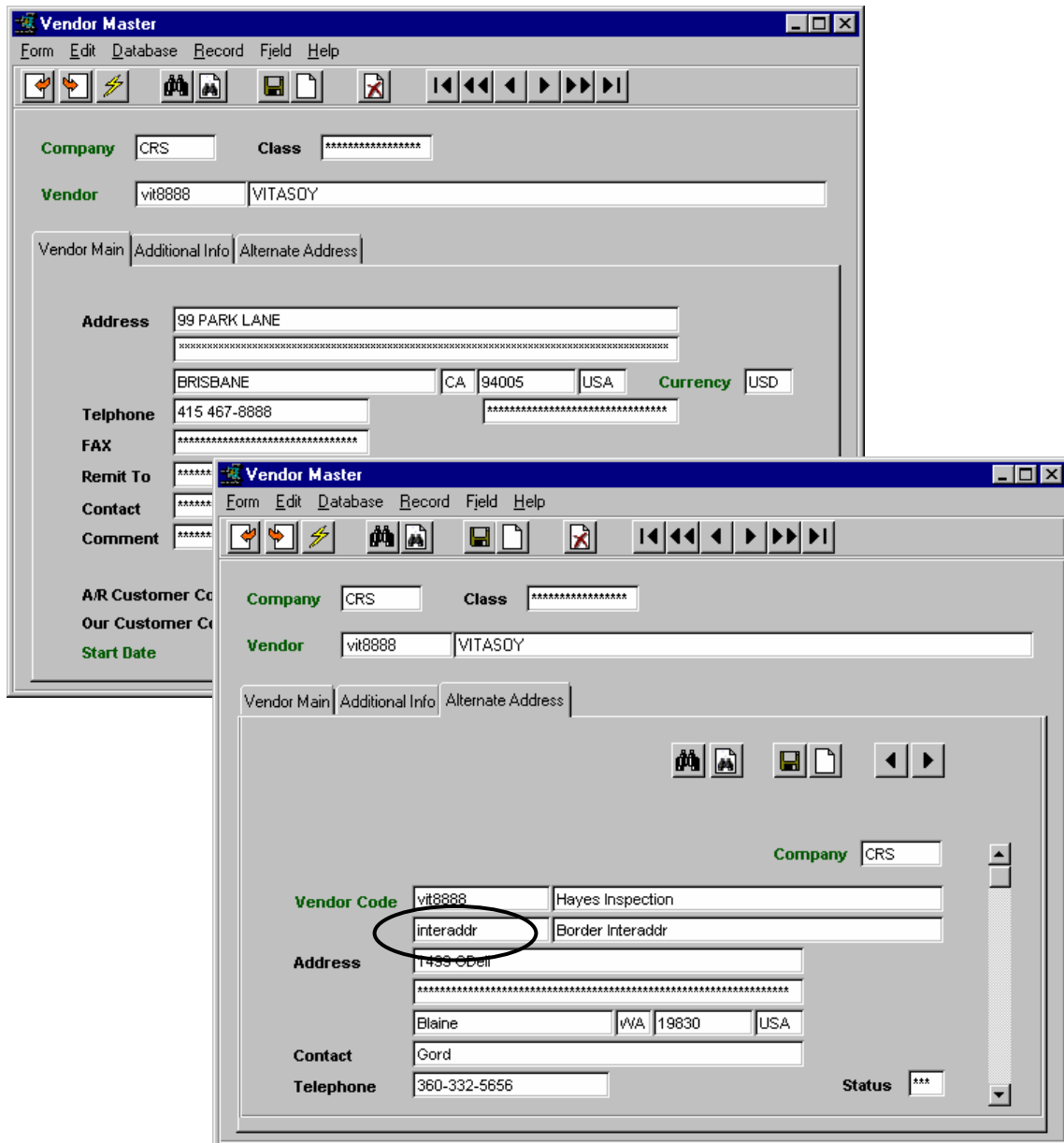
Notice the Alternate location code 'interaddr'. If this code is defined in the Vendor's Alternate Address form, FLEXX will recognize that as an alternate Ship To location. When a new PO is created for that Vendor, FLEXX will copy that Alternate Address data into the PO Text form using keyword *'interaddr'*.

This Text data will then be used as the Ship To address on the Supplier and Shipper copies of Report *'poformc'*, the multi-page PO report, which can also be defined as the 'auto-po' (PO Print button) default report.

See examples of the Purchase Order report below.

PURCHASE ORDER - Supplier Copy

Purchase Order No: **1022**	Page: 1	Vendor Code: vit8888

VITASOY 99 PARK LANE BRISBANE CA 94005 415 467-8888	Horizon Distributors Ltd. 8335 Winston St. Burnaby BC V5A 2H3 (604) 420-6751

Currency:	USD	FOB:	SOUTH SAN	Ship Via:	
Date:	05-Feb-1999	Request No:		Terms Days:	30
Buyer:	harry	Date Required:	05-Feb-1999	Date Promised:	05-Mar-1999
Division:	horizn	Comment:	test interaddr		

Units UOM Pallets Supplier Code Description	Special Cost/Unit Price	Total Cost
Pallet Config. Internal Code		
Ship To: Hayes Inspection 1499 ODell Blaine, WA 19830		

PURCHASE ORDER - Shipping Copy

Purchase Order No: **1022**	Page: 1	Vendor Code: vit8888

Hayes Inspection 1499 ODell Blaine, WA 19830	Horizon Distributors Ltd. 8335 Winston St. Burnaby BC V5A 2H3 (604) 420-6751

Currency:	USD			Ship Via:	
Date:	05-Feb-1999	Request No:			
Buyer:	harry	Date Required:	05-Feb-1999	Date Promised:	05-Mar-1999
Division:	horizn	Comment:	test interaddr		

SKU Code	Description	Total Cost

PURCHASE ORDER - Warehouse Copy

Purchase Order No: **1022**	Page: 1	Vendor Code: vit8888

VITASOY 99 PARK LANE BRISBANE CA 94005 415 467-8888	Horizon Distributors Ltd. 8335 Winston St. Burnaby BC V5A 2H3 (604) 420-6751

Currency:	USD	FOB:	SOUTH SAN	Ship Via:	
Date:	05-Feb-1999	Request No:		Terms Days:	30
Buyer:	harry	Date Required:	05-Feb-1999	Date Promised:	05-Mar-1999
Division:	horizn	Comment:	test interaddr	Arrival Date:	02-Apr-1999

SKU Code	Description		UOM	Units	Warehouse	Loc.	Pallets
					Units Rec'd		Int. Pallet
	Ship To: Horizon Distributors Ltd.						
06640	*VITASOY Original L-12/946ml		CASE	1.00	main	001-15-W1A1	.02
							15X4
		30	Exp. Best Before:	04/05/1999			
		Actual Best Before: _____					
06645	*VITASOY Vanilla Light		CASE	1.00	main	001-00-H1A1	.02

33.2   Alternate Pick Up location

The same general procedure is required to define an alternate Pick Up location for the PO. The alternate pick up location is defined as an Alternate Address on the particular Vendor Master form using the location code *'pickup'*. See the following example:

This will now function the same as the 'interaddr' location code in generating a PO Text entry with 'pickup' keyword and the pickup address data being copied into it.

Rather than printing the Ship To: heading on the PO Reports, it will print Pick Up: with the defined address.

34 Return PO Function

The FLEXX PO Return function has been provided to simplify the process of creating a PO to return a previously purchased SKU to the vendor. The PO Detail form has buttons **Return** and **View Return** added.

To return an item, FIND the original PO (or any previous PO) that the item was purchased on. Go to the Details form and select the particular detail line. Press the **Return** button. The **PO Return** screen will be displayed (see below).

PO Return

The PO Return screen is used to specify the actual return quantity as well as the Warehouse the SKU is to be returned from. The return quantity will default to the whole amount of the original PO but can be changed as desired.

Pressing **OK** will result in a new PO being created using the following default values:

- Current system date and GL Period

- SKU code and quantity as selected,

- original Purchase price (can be changed on the PO detail if desired),

- UOM as on original PO,

- if PO Accruals is used, the same GL Accrual Account numbers as on original PO.

When the PO is complete, press the **Create Voucher** button on the PO header form. This will automatically generate a new voucher (Debit Memo) with a negative value. This DM can then be used to pay down another open voucher by pressing the **Apply Debit Memo** button on the voucher header.

View Return Button

The **View Return** button on the PO Detail form can be used to display the Return PO created for the selected detail line. If the selected line has not been returned, FLEXX will display prompt message: "Unable to select order detail records."

35 Purchasing and Taxes

FLEXX will calculate the VAT amount of a SKU purchase based on purchases made from vendors that are defined to be eligible for VAT, and SKU's defined VAT eligible.

35.1   SKU Tax Definitions

All items to be purchased (and sold) are defined in the SKU Master Table. For the purposes of this discussion, only the tax definitions of the SKU Master will be described.

As shown in the example below, the SKU tax eligibility is defined on the Miscellaneous form, **Taxation** fields. State Tax is used in Sales orders only, and VAT Code specifies the code defined in the Tax Table for the VAT rate to be charged on this SKU.

In this example, the sku *cws-200* is eligible for P/ST *(State Tax = Y),* and VAT at the GST7 defined rate. If the vendor is also defined to be eligible to charge VAT, tax at the rate defined for GST7 will be charged on this SKU.

Note that only the **VAT Code** is applicable to Purchase Order processing.

35.2   Vendor Tax Definitions

The Vendor Master form is used to define each vendor, including whether or not VAT is to be charged on purchases from that vendor.

The Tax Information fields are used to define the various tax related values.

The VAT Flag together with the VAT Code determine the VAT rate the vendor is eligible to charge on purchases of VAT eligible SKU's. The rate at which VAT/ will be charged is determined by the definition of the VAT code in the Tax Table described below.

35.3   Tax Table Definitions

The **Tax Table Information** table is used to define the rates for the various tax codes to be used in FLEXX, both P/ST and VAT. Each Tax Code can also be defined with any number of **Type** codes for a specific rate. Note that the Type code is not necessarily a Y/N (Yes/No) code, but an alphanumeric code that can be used to assign multiple rates to the same Tax Code.

In the above example, GST7 (the VAT code) with Type Y is defined at 7.0% and Type N at 0. Additional Type codes could be defined for the same VAT code to allow for multi-level VAT tax calculations for the same GST7 code, as per the following example:

     GST7 code H = 15.0%

     GST7 code N = 0.0%

Each code is also assigned the GL account that FLEXX is to use to record the tax amounts in. If the tax is refundable, a Credit account can also be specified. Further, the VAT column is used to indicate to FLEXX whether or not VAT is to be charged on the P/ST amount. The tax amounts are recorded in the GL when the different **Generate GL Transactions** processes are run.

35.4   VAT Usage Example

Using the above definitions as an example, following is a description of how FLEXX computes the taxes on a purchase order.

A purchase order is entered for vendor *acme.* The Vendor Master definition for *acme* specified VAT eligibility with a jurisdiction code of GST7. The order will appear as shown below.

PO Header form

There are no tax definition fields on the PO header form.

PO Detail form

SKU *cws-200* is entered on the PO Detail form, where the Internal Rate is the Cost price defined on the SKU Vendor table for this vendor.

Note that there is no indication on the PO of any taxes to be charged. VAT determination and calculation is performed only at the voucher generation process, as shown below.

Voucher generation

After the PO has been received, a voucher can be generated using the data from the original PO. This is accomplished with the Transfer PO to Voucher process.

At voucher creation, the VAT flag definition is copied from the Vendor Master table. Thus, as shown on the voucher header above, vendor *acme* is eligible to apply VAT to the cost of all VAT eligible purchases.

Voucher Detail form

FLEXX generates the voucher detail entries when the **Transfer PO to Voucher** process is run.

The detail entries will be the actual SKU items ordered plus the VAT amount to be charged for all VAT eligible SKU's. Since SKU *cws-200* was defined to be eligible for VAT at the GST7 rate, and vendor *acme* was eligible to charge VAT, FLEXX added the defined rate for GST7 to the cost of the order.

Each PO transferred to the voucher will result in a separate tax calculation and entry. Therefore, if multiple PO's have been transferred to a common voucher, there will be multiple tax entries made.

As mentioned earlier under Tax Table description, the tax amounts will be recorded in the GL at the time the Generate GL Transactions from Vouchers process is run. On purchases, the GL account used will be that defined for **Tax Credit** on the Tax Table for the tax code used (e.g. GST7 will use account 8600 - *see sec. 28.3*).