Getting Started with Pricing

Release 8.6.3 (Eterm)

Legal Notices

Activant® Eclipse™ 8.6.3 (Eterm) Online Help System

Publication Date: May 12, 2008

Table of Contents

Pricing Overview

Eclipse Pricing Management helps you maintain consistent costing and pricing information. From updating price sheets to reporting on sales outcomes and commissions earned, Pricing Management provides a reliable and accurate way of costing and pricing merchandise.

Price Updates

Vendors supply their price information to you through price sheets or price update files. You can then update your pricing information manually or automatically. The system can also automatically add product records for new products during a price update.

Price Lines

When new products are added to the product file they are assigned to price lines. Price lines are groups of products used for sales performance reporting, unit of measure descriptors, and commission groups. Price lines provide default information for the products within a price line.

Price Sheet Entry

Vendors provide basis names associated with a dollar amount on the vendor's price sheet. Each vendor may use different basis names to define their pricing, so cross-reference vendor basis names with Eclipse basis names to create a standard pricing scheme for each price line.

Pricing Matrix

The system prices items using a pricing matrix. A sell matrix defines the pricing rules for your sales, branch transfers, and adjustments. A buy matrix defines pricing rules involving the costs for products on purchase orders.

Within each matrix cell, a formula and a basis name calculation defines the cost or price on an order. You can include the following price- or cost-determining factors in matrix cells:

- Buy and sell groups that share the same pricing rules.
- Quantity break pricing that offer discounts for buying quantity.
- Combination groups that offer quantity break discounts on the combined total of items.
- Rebate pricing that offers customers discounts directly from your vendors.

Commissions

Set up your salespeople with commission plans that regulate how the system calculates commissions for each salesperson. Set up commission plans to calculate commissions based on one of the following:

- Gross profit dollars.
- Sales dollars.
- Net sales dollars.

- Items sold as members of a product commission group.

Quotes

Use Quote Maintenance to offer special pricing to customers during limited periods.

See Also:

Assigning Products to Buy or Sell Groups

Pricing Basis Fundamentals

Pricing Matrix Hierarchy Details

Buy and Sell Group Overview

Setup Requirements for Price Lines

The following control maintenance records and authorization keys are used for Price Line Maintenance.

Control Maintenance Records

Set the following control maintenance records:

- Base Currency For Exchange Rates
- Base Minimum GP% Price Check Off COMM-COST
- Copy Default Price Sheet To Blank Sheets In Product Maint
- Cost Of Goods Sold Basis Name
- Display Products Within A Customer's Product Zones
- Enable Branch Specific Products For
- Global Buy/Sell Basis Names
- New Nonstock Price Line Default
- Notify User When COGS is Updated
- Include Price Lines With Sell Groups For Pricing
- Valid Foreign Currencies
- Valid Customer Points Programs
- Valid Product Zones

Authorization Keys

Assign the following authorization keys:

- OE.NSTK.UM.EDIT
- OE.PRICE.VIEW.LEVEL
- RF.PICK.QTY.INCREASE
- PRD.ZONE
- SOE.MIN.GP

See Also:

Price Line Overview

Creating Price Lines

Setup Requirements for Product Pricing

Following are the control maintenance records and authorization keys used for Product Pricing.

Control Maintenance Records

Set the following control maintenance records:

- Display All Quantity Breaks
- Number Of Days After Which Users Can Edit Old Prices
- Quantity Break Display Percentage
- Should Credit Sales Order Update Avg/Last Cost
- Valid Product Commission Groups

Authorization Keys

Assign the following authorization keys:

- BR.COST.ACTIVITY.VIEW
- NONSTOCK
- PRICE.SHEET.MAINT
- PRODUCT.MAINT
- SOE.OVRD.NO.PRC.CHANGE
- SOE.PACKAGE.QTY

See Also:

Product Pricing Overview

Assigning Pricing Criteria to Products

Setup Requirements for Price Update

The following are control maintenance records and authorization keys used for Price Update.

Control Maintenance Records

Set the following control maintenance records:

- User Defined Product Notes
- Minimum Days Before Report Purge

Authorization Keys

Assign the following authorization keys:

- COGS.VIEW
- COST.VIEW

See Also:

Auto Price Updating Overview

Defining Auto Price Update Parameters

Setup Requirements for Price Sheet Entry

Following are the authorization keys used for Price Sheet Entry.

Authorization Keys

Assign the following authorization keys:

- PRICE.SHEET.UPDATE
- COGS.VIEW
- COST.VIEW

See Also:

Price Sheet Entry Overview

Generating Price Sheets

Setup Requirements for Buy and Sell Matrix Maintenance

Following are the control maintenance records and authorization keys used for Buy and Sell Matrix Maintenance.

Control Maintenance Records

ACCT (Account Management)

Set the following control maintenance records:

- Cost Of Goods Sold Basis Name
- Global Buy/Sell Basis Names
- Should Inventory Adjustment Update Avg/Last Cost
- Update Landed Avg Cost Using Landed Cost

General

Set the following control maintenance records:

- Valid Customer Price Classes

INVM (Inventory Management)

Set the following control maintenance record:

- Enable Exclusion of Matrix Cells From Demand Calculations

Maint (Maintenance)

Set the following control maintenance records:

- Check For Rebate Information For Matrix Cell Cost Overrides
- Default Best Price Check In Sell Matrix Maint To No
- Default Customer Price Class
- Default Cost View on Matrix Maint Cost Override to COMM-COST
- Prohibit Quantity Break Repricing On Closed Orders
- Rebate Detail Setup Information

SOE (Sales Order Entry)

Set the following control maintenance records:

- Best Cost Check Through All Matrix Cells For Sales Orders
- Best Price Check Through Matrix Cells
- Change Both COGS and Commission Cost On Override

- Check For Rebate Information For Matrix Cell Cost Overrides
- Display All Quantity Breaks
- Number of Digits of Accuracy for Pricing
- Prompt For Cost Code On Cost Override In Matrix Maintenance
- Prompt For Cost Code On Cost Override In SOE
- Quantity Break Display Percentage
- Search for Cost Overrides During Pricing
- Stop Best Price Check At First Valid Sell Group
- Valid Cost Override Codes

Velocity Pricing

- Default Rank for Velocity Pricing
- Use Central Warehouse Branch Rank If No Pricing Branch Rank

Authorization Keys

Assign the following authorization keys:

- BMATRIX.MAINT
- OE.PRICE.CLASS.LEVEL
- PRICE.CHANGE.OVRD
- SELL.GROUP.REBATE.MAINT
- SMATRIX.MAINT
- SMATRIX.MAINT.CUS.CLASS
- SOE.SPLIT.PRICING

See Also:

Pricing Matrix Hierarchy Details

Pricing Overview

Pricing Rebate Overview

Setup Requirements for Buy and Sell Groups

Following are the control maintenance records and authorization keys used for Buy and Sell Group Maintenance.

Control Maintenance Records

Set the following control maintenance records:

- Apply Rebates From Sell Group Rebate Table
- Default Customer Price Class
- Include Price Line With Sell Groups For Pricing
- Valid Buy/Sell Group Types
- Valid Customer Price Classes

Authorization Keys

Assign the following authorization keys:

- SELL.GROUP.REBATE.MAINT

See Also:

Buy and Sell Group Overview

Creating Buy and Sell Groups

Setup Requirements for Commission Plan

Following are the control maintenance records and authorization keys used for Commission Plan program.

Control Maintenance Records

Set the following control maintenance records:

- Assign Commission Plan At Time Of Invoice Process
- Calculate Negative Commission When GP Negative
- Maximum Collection Days Date
- Minimum Credit Order GP$ For Commission
- Minimum Sales Order GP$ To Earn Commission
- Valid Product Commission Groups

Authorization Keys

Set the following control maintenance records:

- COMMISSION.USER.AUTH
- COGS.VIEW
- COST.VIEW
- COMM.PLAN. MAINT

See Also:

Commission Plan Overview

Creating Commission Plans

Pricing Basis Fundamentals

Your vendors use pricing references called basis names for pricing their products. Eclipse refers to these as *local basis names* and they can vary from vendor to vendor, therefore they are *local* to each vendor. For example, one vendor may use LIST as your highest cost, while another vendor uses DFLT-LST (default list).

The following can determine your local basis names:

- Common marketplace names.
- Vendor pricing.
- The companies who supply price update information, such as Trade Service or PlumLee.

Global basis names are defined in the system during Eclipse implementation, and the system uses them to calculate prices, sort information for reports, and select data sets for mass updating. You cross-reference the vendor's local basis names with global basis names to standardize basis names for the system. This creates consistent pricing and costing references for all transactions. Many areas of the system, such as Sales Order Entry, Purchase Order Entry, and Reporting rely on global basis names to calculate pricing and costing.

Pricing formulas are paired with basis names to determine costs and prices for all price-related transactions throughout the system. Use formulas to add, subtract, multiply, or divide pricing basis.

System-Defined Pricing and Costing

Your system includes basis names with specific uses for pricing, reporting, and purchasing. These basis names are required for internal system calculations.

Average Cost and Last Cost

The AVG-COST (average cost) and LASTCOST (last cost) basis names are system-defined measurements used for reporting and analysis.

The system calculates AVG-COST as follows:

$$\frac{\text{[(Current on-hand quantity) x (Current AVG-COST)] + [(Incoming quantity) x (Incoming cost)]}}{\text{[Current on-hand quantity + Incoming quantity]}}$$

The LASTCOST is the last incoming cost on a purchase order.

To include an inventory adjustment in the average cost calculation, change the setting in the **Should Inventory Adjustment Update Avg/Last Cost** control maintenance record.

Landed average cost is the total cost required to get a product to the distributor's warehouse, which includes the vendor cost, duty charges, freight charges, and miscellaneous additional charges.

System-Defined Global Basis Names

Many areas of the system, such as Sales Order Entry, Purchase Order Entry, and Reporting rely on system-defined global basis names to calculate pricing.

You can rename basis names or add user-defined global basis names in the **Global Buy/Sell Basis Names** control maintenance record, if you map them to local basis names in each price line. For more information, see Assigning Local Bases to Eclipse Global Bases.

The following table describes the system-defined global basis names.

Global Basis Names	Description	Example
DFLT-LIST	Default List Prompt	Use if no value is assigned to list price.
DFLT-COST	Default Cost Prompt	Use if no value is assigned to list cost.
COGS-COST	COGS (cost of goods sold) Cost Prompt	Assigned in the **COGS Basis Name** control maintenance record.
COMM-COST	Commission (cost) Prompt	Use when calculating sales commissions.
REBAT-COST	Rebate Cost Prompt	Use when vendors offer rebates.
SELL-BREAK	$ (dollar) Break Sell Prompt	Use when offering quantity breaks to customers.
PURC-BREAK	$ (dollar) Break Purchase Prompt	Use when vendors offer quantity breaks to you.
DISP-COST	Display Cost Prompt	Use to openly display a vendor's special costs.

Access Eclipse pricing through the following areas in the system:

- Sales orders
- Purchase orders
- Customer records for velocity pricing
- Product records for velocity pricing
- Price sheets
- Matrix cells

See Also:

Pricing Basis and Formula Guidelines

Pricing Overview

Unit of Measure Guidelines in Pricing

Pricing Basis and Formula Guidelines

Prices in the system are directly related to basis names. Basis names combined with formulas create the pricing scheme used to map customer and vendor pricing.

Formulas can add, subtract, multiply, and divide a price or cost basis. You can combine formulas with basis names at different levels in the system to control pricing.

Enter a pricing formula using the following formats:

Formula...	Description
+n.nnnn	Specifies the basis name plus the percent that follows (for example, +1.123).
-n.nnn	Specifies the basis name minus the percent that follows (for example, -1.123).
-n.nn/n.nn/...	Specifies a chain discount (for example, -1.23/2.34/...).
***n.nnn**	Specifies a multiplier, for example, *1.25 increases a basis by 25 percent.
dn.nn	Specifies a divisor. Divide the amount following *d* into the basis, for example, d1.123 means to divide 1.123 into the basis.
gpn.n	Gross profit (GP) followed by a number (99.99 max.) arrives at the defined margin. The formula to calculate gross profit is: $\frac{\text{Amount for basis}}{(1.0 - \text{GM\%})}$ For example, the formula GP25, on basis REP-COST = $100 is $100 / (1.0 - .25) = $133.33. This formula produces a gross margin percent of 25 percent or a mark up of 33.33 percent (*1.33).
$nn.nn	*Net Pricing* uses the unit of measure for the most recent effective date with a non-zero price in the price sheet, regardless of how the basis is set up. For example, an entry of $15.75 ea would result in a price of $15.75 each; if the unit of measure changes for a more recent price sheet to per 6, if you do not change the price in the $ formula, the resulting price is $15.75 for 6.
Field left blank	Specifies the face value of the basis, or the basis times 1 (*1).
B	Specifies a basis code that applies a multiplier in the price line to the formula.

The following examples describe some of the ways to apply formulas in Eclipse pricing:

- Updating LIST in Price Sheet Entry
- Changing a Formula in a Sell Matrix Cell
- Applying formulas to quantity breaks
- Applying the Gross Profit (GP) Margin Formula
- Applying Chain Discounts

Note: In Sales Order Entry you can run a report on a line item that shows how the system determined the selling price of that item.

Updating LIST in Price Sheet Entry

You have decided to increase LIST price for a group of products by 35 percent. This example contains three products. The product's replacement costs (REP-COST) and formulas derive the new LIST price is shown below.

Product #	REP-COST	Formula	New LIST price
1	$5.00	*1.35	$6.75
2	$10.00	*1.35	$13.50
3	$15.00	*1.35	$20.25

To display the Price Sheet Entry screen, select the price line, price sheet, and discount class.

The **Basis Names** field displays all the local basis names assigned to the price line.

In the **CalcBase*** field adjacent to LIST, enter the basis **REP-COST**.

In the **Formula** field, enter *1.35 (times 35 percent). This creates the formula REP-COST x 1.35.

The result is a new LIST price for each item as shown in the table above.

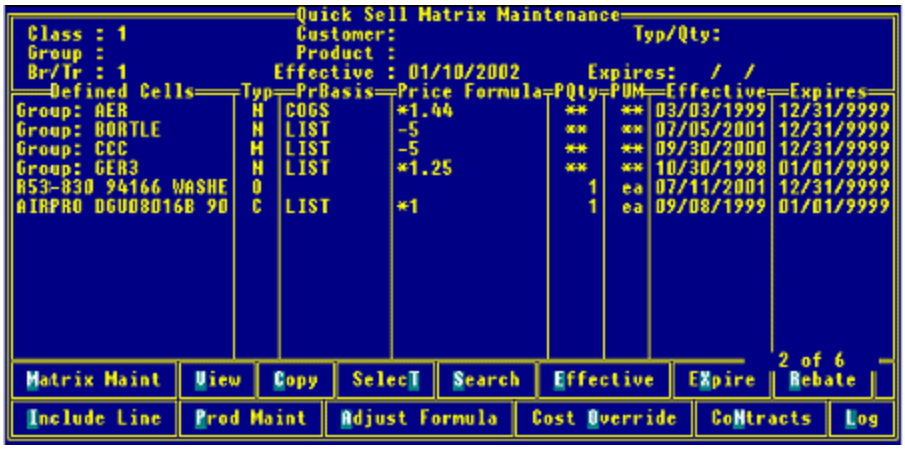
See the following topics for more information:

- Selecting a Price Sheet
- Manually Updating Prices

Changing a Formula in a Sell Matrix Cell

You have found that a sell matrix cell for group CCC has an incorrect formula of LIST *1.2. The sell group contains three products. In this example, change the formula from LIST *1.2 (plus 20 percent) to LIST -5 (minus 5 percent) as follows:

- Display the Quick Sell Matrix Maintenance screen.



- Enter a customer class, a branch, and an effective and expiration date.

  The **Defined Cells** column displays all of the group and product matrix cells assigned to that class and branch whose effective and expiration dates fall within the range.

  In the **PrBasis** column for Group: CCC, find the erroneous cell with a basis of LIST and price formula of *1.2 (plus 20 percent).

- In the **Price Formula** field, enter **-5** (minus 5 percent) in place of *1.2.

  The sell matrix cell for the sell group, customer class, and date range affects prices of products ordered using this matrix cell.

  Note: This process also applies to buy matrix cells.

  The following table shows how prices for each product in sell group CCC are affected by changing the formula from LIST *1.2 to LIST -5.

Product #	Previous formula and price using LIST *1.2	Changed formula LIST -5
1	$10.00 x 1.2 = $12.00	$10.00 -$.50 = $9.50
2	$20.34 x 1.2 = $24.41	$20.34 -$1.02 = $19.32
3	$40.33 x 1.2 = $48.40	$40.33 -$2.02 = $38.31

For more information, see the following topics:

- Creating Matrix Cells in Quick Matrix Maintenance
- Buy and Sell Group Overview

Applying Formulas to Quantity Breaks

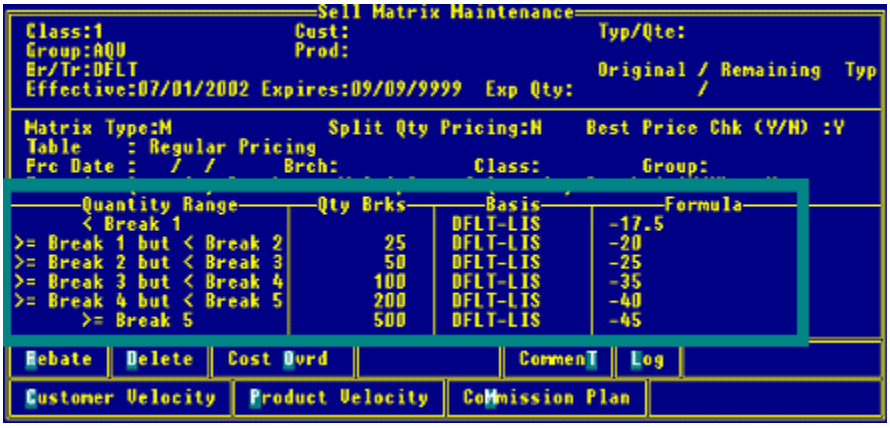
Set up quantity breaks for your customers to encourage them to buy larger quantities from you at greater savings. Keep track of quantity breaks from your vendors on the buy matrix, and assign quantity breaks to your customers on the sell matrix. In the following example, define quantity breaks for class 1 customers and the AQU sell group for all branches as follows:

1. Display the Sell Matrix Maintenance screen.

2. Enter the customer class 1.
3. Enter the AQU product group.
4. Enter DFLT to make this set up apply to all branches.
5. Enter the effective and expire dates.
6. Enter a matrix type of M. Quantity break pricing requires a matrix type of **M** (matrix), **G** (group), **P** (product), or **C** (combo). For more information, see Comparing Matrix Types.

   The **Quantity Range** column lists quantity breaks from < (less than) Break 1 through > (more than) Break 5. Set up five all possible quantity breaks available.
7. In the columns adjacent to **< Break 1**, enter the following information:
   - Nothing in the **Qty Breaks** column. Less than break 1 is the price for this matrix cell before offering quantity breaks, so no quantity is needed.
   - The **DFLT-LIS** (default LIST) basis in the **Basis** column. An entry is required whether you are setting up quantity breaks or not.
   - The formula to calculate the prices with this matrix cell in the **Formula** column. An entry is required whether you are setting up quantity breaks or not.
8. In the columns adjacent to >=**Break 1-4 but <Break 2-5** through **Break 5**, enter the following information as shown on the screen below:
   - In the **Qty Breaks** column, the number that defines the first through fifth quantity breaks.
   - In the **Basis** column, the basis that determines prices on this matrix cell for each quantity break.
   - In the **Formula** column, the formulas to calculate prices for each quantity break.

   The following screen shows the quantity breaks setup for this example.



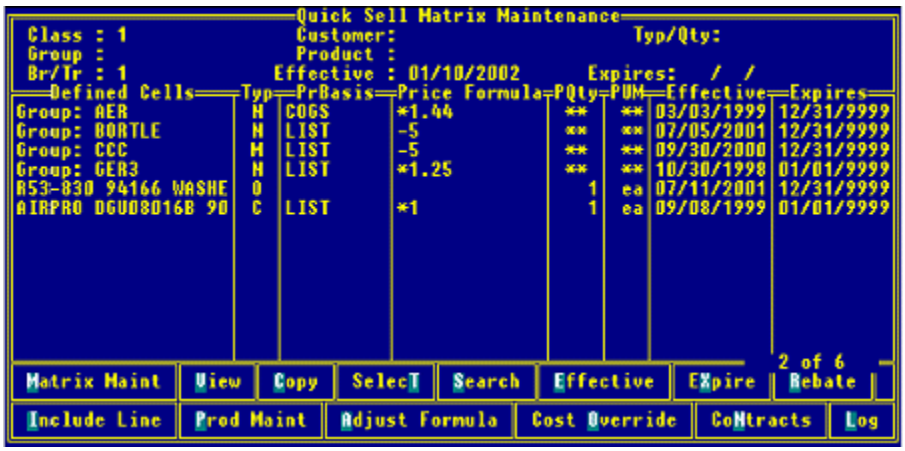
- Break 1 defines a price break of 20 percent for buying 25 to 49 items.
- Break 2 defines a price break of 25 percent for buying 50 to 99 items.
- Break 3 defines a price break of 35 percent for buying 100 to 199 items.
- Break 4 defines a price break of 40 percent for buying 200 to 499 items.
- Break 5 defines a price break of 45 percent for buying more than 500 items.

For more information about quantity breaks, see Working With Quantity Breaks.

Applying the Gross Profit (GP) Margin Formula

You decide to produce a gross profit margin percent of 25 percent for the product P34 Digital Clock that applies to all of your electrical customers. Do the following:

1. Display the Quick Sell Matrix Maintenance screen.



2. Enter **electrical** in the **Typ/Qty** field.
3. Scroll to P34 digital clock in the **Defined Cells** column.
4. Enter **COGS** in the **PrBasis** field. COGS for the P34 Digital Clock = $80.

   Note: If you use **Order COGS** as the price basis, the matrix cell uses the COGS override on the order to calculate the gross profit.

5. Enter **GP25** in the **Formula** field.

   This produces the formula: COGS/(1.0 - GP%).

   The price equals $80/ (1.0 - .25) = $106.67, which is a gross profit percent of 25 percent and a mark up of $26.67.

Applying Chain Discounts

Use chain discounts to determine a selling price by combining formulas to be calculated in a chain. Chain discounts start with the full price of a basis and add formulas to decrease or increase each resulting price in the formula.

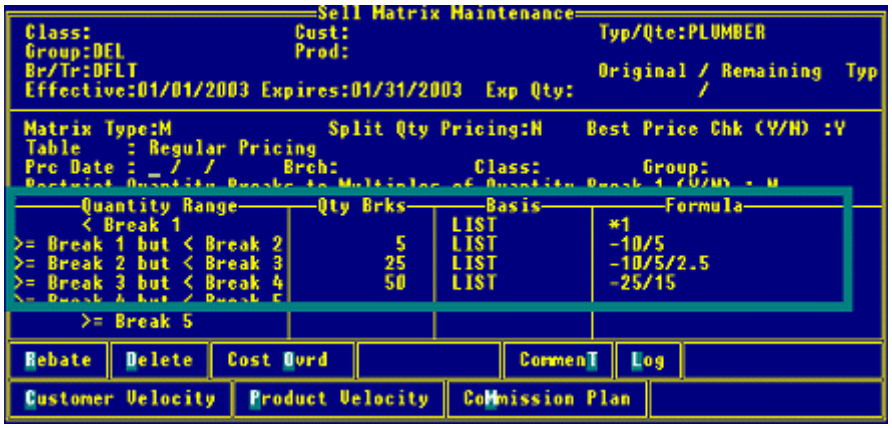
For example, the chain discount LIST - 20/10/5/5 is illustrated in the following table for three products priced at LIST = $20, LIST = $130, and LIST = $200.

Basis	Full price	Price after subtracting 20%	Price after subtracting an additional 10%	Price after subtracting an additional 5%	Final price after subtracting an additional 5%
LIST	$20.00	$16.00	$14.40	$13.68	$13.00
LIST	$130.00	$104.00	$93.60	$88.92	$84.47
LIST	$200.00	$160.00	$144.00	$136.80	$129.96

You cannot mix constants and multipliers in chain discounts unless the constant is the last item in the chain. For example, the system does not support the formula -$10 / *.98 / D.8. The system calculates this equation like *.98 / D.8 / -$10.

The system does support the formula *.5 / +15 / -$8.5. Here the base price is multiplied by 5 percent, then, 15 percent is added to the price, and finally, $8.50 is subtracted from that amount.

Additionally, you can apply chain discounts when pricing quantity breaks. For example, for your plumbing customers, you offer higher discounts for buying quantity in the DEL (Delta) sell group. This matrix cell is shown below.



You can also combine calculation methods within a chain discount. For example, LIST -10/+ $0.50 is illustrated in the following table.

Basis	Full price	Price after subtracting 10%	Price after adding $0.50
LIST	$20.00	$18.00	$18.50
LIST	$130.00	$117.00	$117.50
LIST	$200.00	$180.00	$180.50

Note: Define the selling price in multiple ways. For example, to reduce five percent from the list price, you can use either of the formulas LIST - 5 or LIST * .95.

See Also

Pricing Basis Fundamentals

Basis Code Formula Guidelines

Quantity Break Guidelines

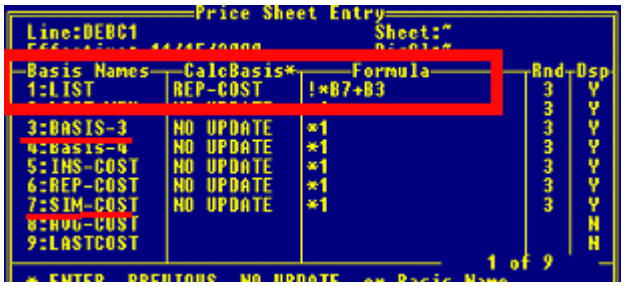
Basis Code Formula Guidelines

Use the basis code formula to calculate a manually updated product price using multipliers or dollar amounts directly from the product record. The basis code is the variable "B" used with a formula on the Price Sheet Entry screen. You can use multiple basis codes in a formula.

The following rules apply to using the basis code (B) in a formula:

- Start the formula with an exclamation mark (!).
- All basis used must be in the same price line.
- Precede B with one or more arithmetic characters (+, -, *, or /).
- Follow the basis code with the number of the basis in the **Basis Names** field in Price Sheet Entry that you want to include in the formula.

For example, the diagram below shows the expression !*B7+B3 for #1:LIST. The formula multiplies basis name number 7 by REP-COST and adds basis name number 3.

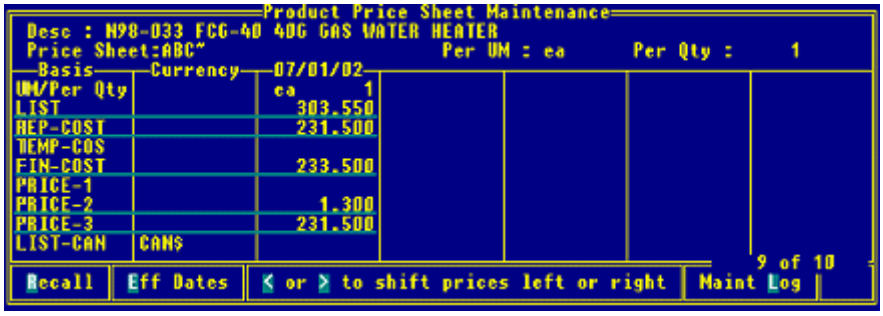


The amount or multiplier for the numbered basis in Price Sheet Entry is defined in Product Price Sheet Maintenance.

The following example sets up a multiplier in Product Price Sheet Maintenance and a basis code in Price Sheet Entry to price a 40-gallon water heater, so the profit margin is always 30 percent plus $2.00 each time the product sells, regardless of what the vendor charges.

The price sheet, below, dated 07/01/02 for the 40G Gas Water Heater shows the following:

- LIST = 303.55. This is the final price for customers.
- Basis REP-COST (replacement cost) = $231.50.
- Basis FIN-COST ($233.50) shows a $2 increase from REP-COST.
- Basis PRICE-2 has a multiplier of 1.300 (30 percent when multiplied).
- Basis PRICE-3 is the same as REP-COST ($231.50).



To price this water heater for future updates, the goal is to add two dollars to the cost, and add an additional 30 percent, or 1.3 x [REP-COST +$2]

On the Price Sheet Entry screen dated 07/01/02, set up the formulas and basis codes as shown, and described below.

1. Enter **PRICE-3** in the **CalcBasis*** field for FIN-COST.
2. Enter **+$2.00** in the **Formula** field for FIN-COST.

   This is FIN-COST = PRICE-3 + $2.00, or $233.50 = $231.50 + $2
3. Enter **PRICE-2** in the **CalcBasis*** field for LIST.
4. Enter **!*B4** in the **Formula** field for LIST.

   This is LIST = PRICE-2 x [PRICE-3 + $2], or 303.55 = 1.3 x 233.50
5. Type **enter** in the **CalcBasis*** field for PRICE-3.

   This allows you to use the **Enter Prices** hot key to manually enter your cost for the water heater on the Price Sheet Entry/Update screen.
6. Enter ***1** in the **Formula** field for PRICE-3.

The result each time you receive a price update on this water heater, you can enter the cost on the Price Sheet Entry/Update screen, and ensure the final price is 30 percent more than your cost plus $2.

See Also:

Pricing Basis Fundamentals

Units of Measure Guidelines in Pricing

Define units of measure (UM) for the following areas of Eclipse:

- **Price line** – Defines the pricing unit of measure. When you assign a price line to a product, the units of measure defined for that price line display in the **UM** column on the product record.
- **Product record** – Defines the product units of measure used for selling, purchasing, transferring, and storing.
- **Product price sheet –** Calculates the unit cost of a product.

We recommend using all uppercase or all lowercase letters when defining units of measure.

Because products in a price line can have varying units of measure for different selling, buying, and storage purposes, set up the price line unit of measure at the lowest level, such as each, and use it as a place holder for the default unit of measure. Then set up each product unit of measure with the detail needed for all transactions.

The following table describes the different areas in Eclipse pricing where units of measure are used, where they are set up, and their purpose.

Unit of Measure (UM)	Eclipse screen	Field and Entry	Purpose
Physical UM	Price Line Maintenance	Enter **ea** in the **UoM Desc** field as the default unit of measure. Enter an asterisk in each row for the default unit of measure for each transaction type: • **S** (Sales Orders) • **P** (Purchase Orders) • **T** (Transfer Orders) • **A** (Inventory Adjustments) • **I** (Inquiry/Inventory)	Used as a default for the units of measure. These entries display on the Product Maintenance screen when you assign a price line to the product. You can override these settings for each product.

Unit of Measure (UM)	Eclipse screen	Field and Entry	Purpose
Product UM	Product Maintenance	In the **UoM Desc** field, enter all of the units of measure you use for storing, selling, and buying the product. If you change a Product UM, you must also change the Pricing UM. In the **Quant** field, enter the quantity for the unit of measure. The quantity for the lowest unit of measure must be 1. Enter an asterisk in each row for the default unit of measure for each transaction type: • **S** (Sales Orders) • **P** (Purchase Orders) • **T** (Transfer Orders) • **A** (Inventory Adjustments) • **I** (Inquiry/Inventory)	Used to set up units of measure for products. Set up quantities for each unit of measure for various transaction types. **Caution!** Do not change the quantity on the lowest unit of measure to be anything other than 1. This causes serious quantity and pricing or costing errors on all history, on-hands, and open orders.

Unit of Measure (UM)	Eclipse screen	Field and Entry	Purpose
Pricing UM	Product Price Sheet Maintenance	**Per UM** and **Per Qty** are read-only. The system calculates these fields from the Physical UM and Product UM. Enter the nit of measure and quantity in the **UM/Per Qty** field for the price sheet to calculate the price, per unit of measure and per quantity of a different um. Click here for more information about determining the Price Per. The unit of measure is normally determined by how the product is priced on the price sheet from the vendor. The values in the Per Up and **Per Qty** fields and the value in the **Basis** field corresponding to the pricing matrix cell for the customer work together to determine the selling price for one unit of the product, referred to as the *price per*. **Example:** the LIST price of a product (defined on this screen) is $20.00 per box (Per UM), with 4 units per box (Per Qty). A customer, who pays LIST * 1 (defined in Sell Matrix Maintenance) for the product, buys 2 of the product. The system divides $20.00 by 4 to determine the unit price of $5.00, and then multiplies that by 2 for a selling price of $10.00.	Used to calculate the unit cost of a product. **Per UM** displays the unit of measure used to price the product. For example, per *each*, per *box*, per *case*, per *c* (hundred), per *m* (thousand), and so forth. **Note:** There is no connection between the value assigned in the **Per UM** field and any of the values defined as units of measure for the Product UM. **Per Qty** displays the number of units contained in the unit of measure entered in the **Per UM** field. For example, if the Per UM is box, the Per Qty might be 4 per box. This value also displays in order entry's View Pricing view in the **UM** column to inform the salesperson how the system defined the unit price.

See Also:

Creating Price Lines

Creating Product Records

Effective, Expiration, and Price Date Guidelines

The matrix cell effective and expiration dates work with the price date on an order to determine which matrix cell the system uses to calculate the cost or price of a product. You can set up multiple matrix cells for the same vendor or customer and product combination, but each must have different effective dates.

If two matrix cells are set up for the same customer and product classifications, and one has an effective date that overlaps the other's current expire date, the system chooses the cell whose effective date is closer to the price date on the order. For example:

- Two sell matrix cells set up for a class 5 customer and sell group FITTINGS:
  - One is dated 10/15/03 (effective) and 10/31/03 (expire).
  - The other is dated 10/15/03 (effective) and 10/31/03 (expire).
- A class 5 customer orders a product from group FITTINGS.

  The price date entered on this order is 10/20/03.

  The system uses the sell matrix cell with effective date 10/15/03 because the effective date is closest to, but not after, the price date entered on the order.

  Note: If you attempt to create a sell matrix cell identical to another matrix cell, and the new matrix cell's effective date is within the existing cell's effective and expire range, the system informs you that an identical cell exists and displays the dates associated with that matrix cell.

For information on how to override the normal matrix selection routine in Sales Order Entry, see Changing Prices Using Price/Cost Override.

See Also:

Changing Price Sheet Effective Dates

Changing Matrix Cell Effective and Expiration Dates

Index

