VOFM ROUTINES IN
SALES & DISTRIBUTION

PART 4: FORMULAS

PRODUCT REQUIREMENTS PLANNING

Version 1.0
April 1999

Table of Contents

1. Introduction

Transaction VOFM is a tool that was developed in R/3 to facilitate the definition of both SAP delivered as well as customer defined routines / rules used in the system during various business processes. VOFM routines are ABAP code written in Forms. VOFM provides the user with the benefit of choosing from one of the standard delivered R/3 routines or writing their own. VOFM is intended for the implementation team when configuring the system. It is not intended for the end user.

VOFM is divided up into four main areas. These include copying requirements, data transfer, requirements, and formulas. This paper will focus on formulas that were delivered by SAP to support the Sales & Distribution (SD) and Logistics Execution (LES) applications. At a high level, formulas are routines that define how a value should be calculated or determined in the R/3 system. For example, a pricing condition value formula allows the user to define how a specific pricing value should be calculated. Similarly, a rounding rule formula defines how rounding should take place when pricing mass maintenance is performed. Formulas are a powerful tool allowing the user to adjust the way in which the system determines certain values. They can also be used to set certain variables in the coding, or even to call other programs.

Section 2 defines how to create a new VOFM routine, in the event that the standard routines do not match the user needs. In the remaining sections, the SAP standard delivered VOFM routines related to Sales & Distribution and Logistics Execution are documented. Starting with Release 4.6A, this documentation is part of the standard product.

This paper assumes that the reader has a general understanding of the SD and LES modules in R/3 as well as a working understanding of the condition technique.

2. Creating a New VOFM Formula

In each area of VOFM, with the exception of Structure of Group Key Formulas, SAP delivers routines using the name space from 1 to 599. SAP customers can create their own VOFM routines using the name space from 600 to 999. With Structure of Group Key Formulas, the SAP name space is 1 to 49, and the customer name space is 50 to 99. To create a new routine, follow these steps:

1. First check to see whether you can use one of the formulas delivered in the standard system.
2. Either overwrite an existing formula or enter a new number on a new line from the customer name space 600 to 999. Also enter a short description of your formula.
3. Program your formula in the ABAP editor.
4. Activate the program.
5. Enter the application if you want to use the formula in a specific application area.
6. Enter your new formula in the appropriate area in customizing. For example, a new pricing condition value formula would be assigned to a condition type or value line in a pricing procedure. As another example, a free goods calculation rule would be assigned to a free goods condition record.

New VOFM routines created at a customer site are not overwritten by a software upgrade.

3. Scale Base

A scale base formula can be used to alter the value that the system uses to read the scales in a condition record during pricing. In standard pricing, the system will read the scale in a pricing condition record using the quantity, value, weight, etc. of the document depending on the type of scale that has been defined.

Example:

Price Product A	
From 0 cases	$50 per case
From 100 cases	$45 per case
From 500 cases	$40 per case

If the customer orders 100 cases of Product A, the system will read the pricing scale using 100 cases and determines the appropriate price, $45 per case. Using a scale base formula, it is possible to alter the value, in our example 100, prior to the scale being read. Scale base formulas are assigned to pricing condition types in R/3 configuration.

When looking at the code for the standard delivered scale base formulas or when writing your own, XKWERT is the field name that the scale base value should be assigned to.

Following is a description of the scale base formulas delivered in the standard system.

SCALE BASE FORMULA 1: FREE

This is an example of a scale basis formula. It is not currently used.

SCALE BASE FORMULA 23: PARTIAL QUANTITY

The formula '23' sets the whole number part of the value to zero. For example, the value 203.559 would be changed to 0.559. Formula '23' was delivered to support the R/3 delivered condition type KP03 to support mixed pallet surcharges.

Example:
A company sells their products in cases. Each of their materials has a conversion factor to pallets. When an order is placed by a customer, the user would like the system to add up the quantities across items and compute the number of full pallets. If the customer does not order in full pallets, the user would like to charge a fixed surcharge of 20 USD. The user sets up condition type KP03 in the pricing procedure. In customizing for the condition type KP03, the scale base formula '23' is assigned as well as the group condition flag so that the quantities across order items can be considered. Within the condition records for condition type KP03, the user maintains a rate of "from 0.001 PAL" a fixed charge of 20 USD. If an order is placed, for example, that is equal to 10.35 pallets, the formula '23' will alter the value to 0.35 and then read the condition record scale. A surcharge of 20 USD would then be applied to the overall sales order.

SCALE BASE FORMULA 43: TAX LICENSE FRANCE

Formula '43' was delivered to support tax exemption licenses in France (R/3 delivered condition type LCFR) where the tax exemption is granted up to a certain invoiced amount. For additional information on tax exemption licenses, please refer to Note 72040.

SCALE BASE FORMULA 202: PRICE-BOOK SCALE

This requirement is used together with the feature 'Data determination in the access' (delivered with Release 4.5). In the R/3 delivered system, condition type PBUD can be used to determine a special scale basis in the pricing procedure. If scale basis formula '202' has been assigned to subsequent condition types that appear in the pricing procedure, the system uses the special scale basis versus the normal scale basis from the sales document line item. Further information about 'Data determination in the access' can be found in the R/3 Sales & Distribution documentation under the section titled 'Special pricing functions'.

Example:
A company wishes to define a pricing agreement that specifies that all customers in a certain customer group receive the price associated with ordering 100 cases of product regardless of how many cases are ordered. This involves fixing the scale basis for the customer group for a certain period of time. This can be accomplished by using the 'Data determination in the access". A condition type, such as PBUD can be used to define the fixed scale value of 100 cases. Subsequent condition types that should use this fixed value are then assigned scale base formula '202'.

4. Condition Base Value

Condition base value formulas can be used to influence the condition basis to which the pricing condition rate will be applied. In standard pricing, the system will apply the condition rate to the quantity in the sales document. For example, the price determined by the system is $45 per case for Product A. 100 cases of Product A have been ordered. The system would then multiply $45 times the number of cases of Product A ordered. Using a condition base value formula, it is possible to alter the condition base, in our example 100, prior to the calculation taking place. A condition base value formula is commonly used to determine the base for distributing header discounts / surcharges to the sales document line items. A condition base value formula is assigned to a condition type in the pricing procedure.

When looking at the code for the standard delivered condition base value formulas or when writing your own, XKWERT is the field name that the condition base value should be assigned to.

Following is a description of the condition base value formulas delivered in the standard system.

CONDITION BASE VALUE FORMULA 1: VOLUME

Formula '1' uses the volume of the sales document line item as the condition base value.

Example:
A company regularly applies fixed amount header discounts to a sales order. For example, the user may apply a fixed discount of 500 USD to the header of the sales order. Fixed header conditions are always distributed across the line items in the document. In this case, the company would like to distribute the fixed amount based on the volume of the line items. To accomplish this, the user

would assign condition base value formula '1' to the header discount condition type in the pricing procedure.

CONDITION BASE VALUE FORMULA 2: NET VALUE

Formula '2' uses the net value of the sales document line item as the condition base value.

Example:
Reference example for formula 1.

CONDITION BASE VALUE FORMULA 3: NET PRICE

Formula '3' was delivered to support net price processing. For additional information on the use of this formula along with examples, please refer to Note 80183.

CONDITION BASE VALUE FORMULA 4: NET VALUE PLUS TAX

Formula '4' uses the net value plus tax of the sales document line item as the condition base value.

Example:
Reference example for formula 1.

CONDITION BASE VALUE FORMULA 5: KZWI1

Formula '5' uses the value determined for subtotal '1' in the pricing procedure as the condition base value.

Example:
A company regularly applies fixed amount header discounts to a sales order. For example, the user may apply a fixed discount of 500 USD to the header of the sales order. Fixed header conditions are always distributed across the line items in the document. In this case, the company would like to distribute the fixed amount based on a subtotal derived based on certain values in the pricing procedure. The user has assigned subtotal '1' to the relevant pricing procedure value(s). In addition, the user would assign condition base value formula '5' to the header discount condition type in the pricing procedure.

CONDITION BASE VALUE FORMULA 6: KZWI2

Formula '5' uses the value determined for subtotal '2' in the pricing procedure as the condition base value.

Example:
Reference example for formula 5.

CONDITION BASE VALUE FORMULA 7: KZWI3

Formula '5' uses the value determined for subtotal '3' in the pricing procedure as the condition base value.

Example:

Reference example for formula 5.

CONDITION BASE VALUE FORMULA 11: CASH DISCOUNT BASE

Formula '11' can be used with the cash discount condition type. This formula reads the indicator for the company code to determine if the cash discount is based on the net value of the item. If it is not, then the system bases it off of the net value plus the tax.

CONDITION BASE VALUE FORMULA 12: GROSS WEIGHT

Formula '12' uses the gross weight of the sales document line item as the condition base value.

Example:
Reference example for formula 1.

CONDITION BASE VALUE FORMULA 13: NET WEIGHT

Formula '13' uses the net weight of the sales document line item as the condition base value.

Example:
Reference example for formula 1.

CONDITION BASE VALUE FORMULA 14: SET EXCLUSION INDICATOR

Formula '14' is an example of how a programmer can dynamically set the condition exclusion indicator based on certain values. This condition exclusion indicator can then be checked in subsequent condition base value formulas (reference formula '15') to exclude certain condition types.

CONDITION BASE VALUE FORMULA 15: CHECK EXCLUSION INDICATOR

Formula '15' is an example of how a programmer can check the value of the condition exclusion indicator and then influence certain values. In this example, if the exclusion field is set to '$', the condition base value is set to zero and the condition is marked inactive. This example is provided in combination with formula '14' which shows how to dynamically set the condition exclusion indicator.

CONDITION BASE VALUE FORMULA 16: NET VALUE MINUS CASH DISCOUNT

Formula '16' uses the net value minus the cash discount value as the base value. This can, for example, be applied to a tax condition type that should use this value as a basis.

CONDITION BASE VALUE FORMULA 17: NET PRICE

Formula '17' was delivered to support net price processing. For additional information on the use of this formula along with examples, please refer to Note 80183.

CONDITION BASE VALUE FORMULA 18: NO QUANTITY CONVERSION

Formula '18' is used to avoid rounding errors that can occur for quantity dependent condition types where the sales unit and the pricing unit are one unit of measure and the base unit of measure is another. Using formula '18', no conversion is done to base unit of measure when the sales unit of measure and pricing unit of measure are identical.

NOTE: As of Release 4.0, this formula is no longer necessary. The user can achieve the same result by setting the "quantity conversion" indicator in condition type configuration.

Example:
A company uses KG as their base unit of measure, but chooses to quote prices and sell products in pieces (PC). A quantity conversion has been maintained to indicate that 3 PC correspond to 1 KG. A pricing record has been created to price 1 PC of product at 1 USD. If a customer orders, for example, 1 PC of product, it is possible that the system returns back an item value of 0.999 USD. To avoid this, the user assigns condition base value formula '18' to the price condition type as well as other quantity dependent condition types in the pricing procedure.

CONDITION BASE VALUE FORMULA 19: KZWI4

Formula '19' uses the value determined for subtotal '4' in the pricing procedure as the condition base value.

Example:
Reference example for formula 5.

CONDITION BASE VALUE FORMULA 20: KZWI5

Formula '20' uses the value determined for subtotal '5' in the pricing procedure as the condition base value.

Example:
Reference example for formula 5.

CONDITION BASE VALUE FORMULA 21: KZWI6

Formula '21' uses the value determined for subtotal '6' in the pricing procedure as the condition base value.

Example:
Reference example for formula 5.

CONDITION BASE VALUE FORMULA 22: WHOLE NUMBER

Formula '22' is used to convert the basis to a whole number. For example, a basis of 300.153 would be converted to 300. Formula '22' is delivered in R/3 along with the condition type KP00 which can be used to compute pallet discounts.

Example:
A company sells their products in cases. Each of their materials has a conversion factor to pallets. When an order is placed by a customer, the user would like the system to calculate the number of full pallets for each line and to offer a 5 USD discount per full pallet ordered. The user sets up condition type KP00 in the pricing procedure and assigns condition base value formula '22'. Within the condition records for condition type KP00, the user maintains the 5 USD per pallet discount rate. If

an order line item is placed that contains 5.5 pallets, the system will adjust the base value to 5 and compute a discount of 25 USD for the sales line item.

CONDITION BASE VALUE FORMULA 24: 1 IF PARTIAL QUANTITY

Formula '24' is used to convert the basis to a quantity of 1 if a partial quantity is involved, otherwise the basis is set to zero. For example, a basis of 300.153 would be converted to 1. As a second example, a basis of 300 would be converted to zero. Formula '24' is delivered in R/3 along with the condition type KP01 which can be used to calculate an incomplete pallet surcharge.

Example:
A company sells their products in cases. Each of their materials has a conversion factor to pallets. When an order is placed by a customer, the user would like the system to calculate the number of full pallets for each line and to charge a 5 USD surcharge to the item if a full pallet quantity is not ordered. The user sets up condition type KP01 in the pricing procedure and assigns condition base value formula '24'. Within the condition records for condition type KP01, the user maintains the 5 USD surcharge. If an order line item is placed that contains 5.5 pallets, the system will adjust the base value to 1 and compute a surcharge of 5 USD for the sales line item.

CONDITION BASE VALUE FORMULA 26: BOLLO IN FATTURA

Formula '26' was provided to support the Italian Bollo in Fattura. Using this formula will set the condition to inactive if the VAT value is not zero. This formula can be assigned to the R/3 delivered condition types BOLL and MWBO in the pricing procedure.

CONDITION BASE VALUE FORMULA 27: XWORKK – DEACTIVATE CONDITION

This formula was delivered to support tax exemption licenses in Italy. The condition base value formula '27' is assigned to the tax condition type in the pricing procedure. The formula is used to deactivate the tax when a valid license exists. For additional information on tax exemption licenses, please refer to Note 72040.

CONDITION BASE VALUE FORMULA 28: 100% DISCOUNT

Formula '28' sets the rate of the condition to a 100% discount. This formula was delivered with condition type R100 to support Release 4.0 Free Goods functionality.

Example:
A company has a free goods agreement with their customers. For every 10 cases of Product A that the customer buys, the customer receives 2 cases of Product B for free. From a pricing perspective, the user wants to track both revenue and sales deductions for the free items since Product B is also sold by itself sometimes in the sales process. To do this, the user flags the free goods item category with the pricing indicator 'B'. In addition, the user adds the R/3 delivered condition type R100 to the pricing procedure at the point at which the 100% discount should be applied. Condition base value formula '28' is assigned to condition type R100 in the pricing procedure to apply the 100% discount rate.

CONDITION BASE VALUE FORMULA 29: FREE GOODS / INCLUSIVE

Formula '29' was delivered along with condition type NRAB to support Release 4.5 Inclusive Free Goods agreements where the user would prefer to have a discount applied to the ordered item rather than having a sub-item generated for the free quantity.

Example:
A company has a free goods agreement with their customers. If the customer orders 100 cases of Product A, they receive 10 of these cases free of charge. Instead of having a free sub-item generated by the system to represent the free 10 cases, the user would like a discount applied to the 100 case line item equal to the value of 10 cases. To accomplish this, the user assigns the NRAB condition type to the pricing procedure and assigns the condition base value formula '29'.

CONDITION BASE VALUE FORMULA 44: VAT FRANCE

This formula was delivered to support tax exemption licenses in France. The condition base value formula '44' is assigned to the tax condition type in the pricing procedure. The formula is used to deactivate the tax when a valid license exists. For additional information on tax exemption licenses, please refer to Note 72040.

CONDITION BASE VALUE FORMULA 50: QUANTITY ACTIVE INGREDIANT

Condition base value formula '50' was delivered to support pricing based on active ingredient quantities. This formula is assigned to the condition types in the pricing procedure for which you work with proportion quantities. Using this formula, the system can carry out billing at the main item level where the system cumulates the quantities from the batch split items. For additional information on Active Ingredient Management, please refer to the Batch Management Guide in the R/3 Library.

CONDITION BASE VALUE FORMULA 51: SHIPMENT COSTING -TAXES

Formula '51' was delivered along with shipment costing in the shipment cost document. This formula assigns the tax indicator from the tax condition record to the shipment cost document item. This condition base value formula should be assigned to tax condition types in the shipment cost document pricing procedure.

CONDITION BASE VALUE FORMULA 61: PREFERENCE

Condition base value formula '61' was delivered to solve a field overflow problem that can occur when working with preference determination. This can occur due to the quantity dependency. Formula '61' is assigned to the preference condition type in the pricing procedure (R/3 delivered condition type PREF). Also reference condition value formula '61'. For additional information, please reference Note 92321.

CONDITION BASE VALUE FORMULA 202: PRICE BOOK FACTOR

Formula '202' can be used to offer a price that is a percentage of a predefined price. For additional information on this Release 4.5 feature, please refer to the R/3 Library documentation on Price Book, which can be found in the Special Pricing Functions (Data Determination in the Access) section of the Sales & Distribution Pricing Guide.

Example:

A company would like to define a pricing agreement with their customer whereby the customer should pay 90% of the material price from the previous year. This agreement can be stored using R/3 Price Book functionality and the pre-delivered condition type PBUD. 90% is entered as the condition value in the PBUD condition record and the pricing date is defined using the data determination in access functionality. In the Price Book pricing procedure, condition type PBBS is then used to determine the base price from last year. Next in the pricing procedure is condition type PBUP which is used to determine the gross price which should be 90% of last year's price. To perform this calculation, condition basis formula '202' is assigned to condition type PBUP in the pricing procedure.

5. Condition Value

Condition value formulas are available to influence the condition value that is displayed for a particular condition type or value line in the pricing procedure. In standard pricing, the system will calculate the condition value by multiplying the condition rate by the quantity. For example, 100 cases times $45 per case equals $4,500. Using a condition value formula, it is possible to alter the condition value, in our example $4,500. Condition value formulas can also be used to compute values that should appear as value lines in the pricing procedure. As an example, standard delivered condition value formulas can be used to compute profit margin and item net value. A condition value formula is assigned to a condition type in the pricing procedure.

When looking at the code for the standard delivered condition value formulas or when writing your own, XKWERT is the field name that the condition value should be assigned to.

Following is a description of the condition value formulas delivered in the standard system related to SD.

CONDITION VALUE FORMULA 1: PROFIT MARGIN WITH REBATE

Formula '1' sets the value equal to the pricing subtotal '3' minus the cost of the line item.

Example:
A company offers rebate agreements to their customers that are paid out at the end of the year based on cumulative sales. On a sales order, rebate agreement accruals show up as a statistical amount and do not effect the net value of the line item. The company, however, would like to see the profit margin in the line item reflect the expected rebate payments. In order to accomplish this, the user assigns the subtotal field '3' to the relevant condition types in the pricing procedure so it's value equals the net value as well as the value of any rebate accruals. The condition value formula '1' is then assigned to the ‘Profit Margin' line in the pricing procedure. Note that subtotal '3' is just an example supplied by the system. Another subtotal could be used and the user would then copy this formula and create a new one that uses the other subtotal value. If rebate accruals should not be included in the profit margin calculation, the user would select condition value formula '11'.

CONDITION VALUE FORMULA 2: NET VALUE

Formula '2' sets the value equal to net value that has been calculated so far for the item in the pricing procedure. It contains the amount excluding taxes.

Example:
A company would like to show subtotals in their pricing screen that would represent the gross value, net value, and net value 2. These are all value lines in the pricing procedure that do not correspond to a specific condition type. To determine the value for these value lines, the user assigns the condition value formula '2'. The system then shows in these subtotal lines the net value of the line item up until that point in the pricing procedure.

CONDITION VALUE FORMULA 3: CASH DISCOUNT MINUS TAX

Formula '3' sets the value equal to the amount eligible for cash discount minus tax.

CONDITION VALUE FORMULA 4: NET VALUE PLUS TAX

Formula '4' sets the value equal to the net value plus tax. This can be assigned, for example, to a value line at the end of the pricing procedure that should show the final value of the line item.

CONDITION VALUE FORMULA 5: PROFIT MARGIN DIALOGUE

Formula '5' provides an example of how a message can be displayed if the profit margin for a particular item is below a predefined percentage. To use this feature, the user would copy formula '5' into a new formula and assign the relevant percentage to the field MINDEST_PROZ.

CONDITION VALUE FORMULA 6: INITIAL PRICE

Formula '6' was delivered to support net price processing. For additional information on the use of this formula along with examples, please refer to Note 80183.

CONDITION VALUE FORMULA 8: EXPECTED VALUE

Formula '8' provides an example of how the system can compare the customer expected value of a line item with the net value computed by the system. If the system net value is not within a value range of 1 of the customer expected value, than the item is blocked. Formula '8' was delivered along with the condition type EDI2 for customer expected value. Users can copy this formula and specify their own value range tolerance in the field MAXIMUM.

CONDITION VALUE FORMULA 9: EXPECTED PRICE

Formula '9' provides an example of how the system can compare the customer expected price of a line item with the net price computed by the system. If the system net price is not within a value range of 0.05 of the customer expected price, than the item is blocked. Formula '9' was delivered along with the condition type EDI1 for customer expected price. Users can copy this formula and specify their own value range tolerance in the field MAXIMUM.

CONDITION VALUE FORMULA 11: PROFIT MARGIN

Formula '11' computes the profit margin for a line item. This is done by subtracting the cost of the item from the net value. This formula can be assigned to a value line at the end of a pricing procedure to show the profit margin for the item.

CONDITION VALUE FORMULA 13: MINIMUM VALUE SURCHARGE

Formula '13' computes the applicable surcharge when the order value (before taxes) falls below the predefined minimum order value. This formula was delivered with condition types AMIW (used to define the minimum order value) and AMIZ (used to compute the surcharge if the minimum is not met). Within the pricing procedure, formula '13' should be assigned to condition type AMIZ. In addition, the subtotal 'D' must be assigned to condition type AMIW in the pricing procedure.

Example:
A company would like to define minimum order values for their customers. As an example, a minimum order value of 200 USD is defined for Customer A. If Customer A places an order for anything less than 200 USD (before taxes), the system should automatically compute a surcharge equal to the difference and apply it to the order. To accomplish this, the user would configure pre-delivered condition types AMIW and AMIZ in their pricing procedure as defined above and maintain a condition record for AMIW and Customer A equal to 200 USD.

CONDITION VALUE FORMULA 14: BEST PRICE

Formula '14' can be used to select the best price in a pricing procedure when more than one condition type to determine the price has been configured.

Example:
A company quotes prices based on weight as well as number of cases. In the pricing procedure, they have maintained two condition types for the price. Condition type ZWGT is listed first and computes the price based on the weight. Condition type ZCSE is listed second and computes the price based on the number of cases. When pricing is done, the system should automatically select the best price for the customer. To accomplish this, the user assigns condition value formula '14' to condition type ZCSE in the pricing procedure. Using this formula, the system compares the two. If the case price is more expensive, then the ZCSE condition line is set to inactive. If the case price is less expensive, then the ZCSE condition line remains active causing the ZWGT condition line to become inactive (active subsequent price in the pricing procedure).

CONDITION VALUE FORMULA 15: MINIMUM PRICE

Formula '15' was delivered along with condition type PMIN to define minimum prices.

Example:
A company has defined minimum prices for materials. When a material is sold, it should not be sold for a price below the predefined minimum price. When pricing is done for a sales document line item, if the net price of the item falls below the minimum, the system should automatically compute a surcharge to bring the price up to the minimum price. To accomplish this, the user would define the minimum prices using the condition type PMIN. PMIN would be defined in the pricing procedure and condition value formula '15' would be assigned. Using the formula, the system compares the minimum price with the net price calculated to that point in the pricing procedure. If the minimum price is not met, the system computes the necessary surcharge and assigns it to the PMIN condition line.

CONDITION VALUE FORMULA 16: ROUNDING THE TOTAL

Formula '16' was delivered along with condition type DIFF to support the rounding unit rules that can be defined in T001R for company code / currency combinations. Condition type DIFF was delivered to perform the rounding at the end of the pricing procedure with the total value. Using formula '16', the system computes the rounded value and assigns the difference to the condition type DIFF.

CONDITION VALUE FORMULA 17: ROUNDING AS PER T001R

Formula '17' was delivered so that a condition value could be rounded off according to the rounding unit rules that can be defined in T001R for company code / currency combinations. When formula '17' is assigned to a condition type, the condition value will always be rounded using T001R.

CONDITION VALUE FORMULA 18: PERCENT CONTRIBUTION MARGIN

Formula '18' computes the percent contribution margin for the line item comparing the cost and net value of the item. This formula can be assigned to a value line at the end of a pricing procedure to show the percent contribution margin for the item.

CONDITION VALUE FORMULA 19: P-VARIANT RITTER

Formula ‘19’ is used to support the calculation of the net price. For additional information on the use of this formula along with examples, please refer to Note 80183.

CONDITION VALUE FORMULA 20: P VARIANTS DISCOUNT

Formula ‘20’ is used to support the calculation of the net price. For additional information on the use of this formula along with examples, please refer to Note 80183.

CONDITION VALUE FORMULA 25: KZWI1 MINUS TAX

Formula '25' computes the value of the condition line to be equal to the value in subtotal '1' minus tax. Formula '25' was delivered along with condition type NETW to compute the value of the goods when tax is part of the price. Reference standard delivered pricing procedure RVAB01 'Tax included in price'.

CONDITION VALUE FORMULA 36: CUMULATION CONDITION

Formula '36' enables the user to display the total of the net values of an item and all the sub-items belonging to that item. This formula was delivered along with condition type KUMU. This condition type can be assigned to the pricing procedure along with formula '36' to display cumulative values when main and sub-items are used.

CONDITION VALUE FORMULA 37: TAX EXEMPTION LICENSE

Formula '37' was delivered to support tax exemption licenses in Italy. This formula should be assigned to the condition type for tax exemption licenses in Italy (R/3 delivered condition type LCIT) in the pricing procedure. For additional information on tax exemption licenses, please refer to Note 72040.

CONDITION VALUE FORMULA 38: EXCLUSION WITH VALUE ZERO

Formula '38' sets the value of the field AUSSCHLUSSWERTNULL to 'X'. This formula was delivered in order to support condition exclusion where conditions with a value of zero should be considered in the exclusion. For additional information, refer to Note 39641.

Example:
A company has two condition types in their pricing procedure that represent surcharges. A condition exclusion group has been defined with these two condition types indicating that the lowest of the two should be applied. In some cases, one of the surcharges may be zero. This could be due to a condition record that is found or a manual entry. In order to have the system consider zero as the lowest surcharge for the customer, condition value formula '38' must be assigned to one of the condition types in the pricing procedure.

CONDITION VALUE FORMULA 47: VAT FRANCE

Formula ‘47’ was delivered to support tax exemption licenses in France. This formula should be assigned to the condition type for tax exemption licenses in France (R/3 delivered condition type LCFR) in the pricing procedure to support this functionality. For additional information on tax exemption licenses, please refer to Note 72040.

CONDITION VALUE FORMULA 48: CHECK DOWN PAYMENTS

Formula '48' was delivered to ensure that the down payment amount the user offsets in a billing document does not exceed the actual down payment value. Condition value formula '48' is assigned to the condition type in the pricing procedure representing down payments (R/3 delivered condition type AZWR). Down payment functionality was delivered with R/3 Release 4.0.

CONDITION VALUE FORMULA 61: PREFERENCE MAX VALUE

Condition value formula '61' was delivered to solve a field overflow problem that can occur when working with preference determination. This can occur due to the quantity dependency. Formula '61' is assigned to the preference condition type in the pricing procedure (R/3 delivered condition type PREF). Also reference condition base value formula '61'. For additional information, please reference Note 92321.

6. Structure of Group Key

A structure of group key formula can be used to influence the basis the system uses when reading the scale of a group condition. Group conditions are used to cumulate quantities from more than one sales document line item to read pricing scales. For example, when pricing a particular sales document line item, the user would like the system to not just consider the quantity of the current line item, but the sum of the quantities of all line items that share the same material pricing group as the current line item. The formula is assigned to a group condition type in customizing.

When looking at the code for the standard delivered structure of group key formulas or when writing your own, XVAKEY is the field name that the structure of the group key should be assigned to.

Following is a description of the structure of group key formulas delivered in the standard system.

STRUCTURE OF GROUP KEY FORMULA 1: OVERALL DOCUMENT

Formula '1' adds up the quantities / values of all of the line items in the sales document that have the same condition type as the group condition currently being processed.

Example:
A company defines a particular discount (condition type Z001) with scales based on weight. When a sales order line item is priced that is eligible for the Z001 discount, the user would like the system to read the scale with not just the weight of the current line item, but the combined weight of all items in the sales document where the Z001 discount applies. To accomplish this, the user defines condition type Z001 as a group condition and assigns structure of group key formula '1' to it in customizing.

STRUCTURE OF GROUP KEY FORMULA 2: ACROSS ALL CONDITION TYPES

Formula '2' adds up the quantities / values of all of the line items in the sales document independent of which condition types have been applied.

Example:
A company defines their prices with scales based on weight. When a sales order line item is priced, the user would like the system to read the scale with not just the weight of the current line item, but the combined weight of all items in the sales document. To accomplish this, the user defines their price condition types as group conditions and assigns structure of group key formula '2' to them in customizing.

STRUCTURE OF GROUP KEY FORMULA 3: MATERIAL PRICING GROUP

Formula '3' adds up the quantities / values of all of the line items in the sales document that have the same material pricing group (field KONDM) as the current sales document line item.

Example:
A company defines a particular discount (condition type Z001) with scales based on weight. When a sales order line item is priced that is eligible for the Z001 discount, the user would like the system to read the scale with not just the weight of the current line item, but the combined weight of all items in the sales document that have the same material pricing group as the current line item. To accomplish this, the user defines condition type Z001 as a group condition and assigns structure of group key formula '3' to it in customizing.

STRUCTURE OF GROUP KEY FORMULA 10: SHIPPING MATERIAL

Formula '10' is only relevant for shipment costing in the shipment cost document. Formula '10' adds up the quantities / values of all of the sub-items in a shipment cost document item that have the same shipping material as the sub-item currently being processed.

7. Rounding Rule

Rounding rule formulas are provided to determine the way in which rounding should take place when a mass change of pricing records takes place. For example, the user may be carrying out a mass price increase of 10% to all materials in a particular material group. The user would like the system to round the new prices to .99 after the calculation. This can be accomplished with a rounding rule formula. A rounding rule formula is assigned when a mass pricing change is carried out.

When looking at the code for the standard delivered rounding rule formulas or when writing your own, fields RV13A-KBETR and RV13A-CURCONV are used to assign the rounded value to.

Following is a description of the rounding rule formulas delivered in the standard system.

ROUNDING RULE FORMULA 1: ROUND TO 9 DECIMAL PLACE

The amount is rounded down by one place and the decimal places of the rounded down amount are rounded up to 9.

Example:
Price $698.45 increased by 1%

Without rounding rule:	$705.43
With rounding rule:	$704.99

ROUNDING RULE FORMULA 2: LAST DIGIT ROUNDED TO 9

The last decimal point of the changed amount is rounded up to 9.

Example:
Price $777.03 increased by 1%

Without rounding rule:	$784.80
With rounding rule:	$784.89

ROUNDING RULE FORMULA 3: 5 RAPPEN ROUNDING

5 Rappen rounding. This is only possible if the currency is CHF (Swiss Franc)

Example:
Price 12.10 CHF increased by 1%

Without rounding rule:	12.22 CHF
With rounding rule:	12.20 CHF

Price 12.13 CHF increased by 1 CHF

Without rounding rule:	13.13 CHF
With rounding rule:	13.15 CHF

ROUNDING RULE FORMULA 4: 2 DIGITS AFTER COMMA

The system rounds down to 0 as of the second decimal place

Example:

Price $555.55 increased by 1%

Without rounding rule:	$561.11
With rounding rule:	$561.10

8. Calculation Rule – Free Goods

Free goods calculation formulas can be used to define the way in which free goods quantities are computed within a free goods agreement. The formula is assigned to a free goods condition record.

When looking at the code for the standard delivered free goods calculation formulas or when writing your own, work area L_FRM is filled with the calculated values.

Following is a description of the free goods calculation rules delivered in the standard system.

CALCULATION RULE – FREE GOODS 1: PROPORTIONAL

Formula '1' interprets the free goods quantities as a proportional agreement.

Example:
A company offers free goods to their customers when they order certain materials. For example, when a customer orders 100 cases of Material A, the customer receives an additional 20 cases of Material A for free. The company would like to have the system treat the 'Buy 100, Get 20 Free' agreement proportionately. For example, if the customer orders 162 cases, the system should automatically grant 32 for free [162 x (20/100)]. To accomplish this, the user would assign free goods calculation rule '1' to the free goods condition record.

CALCULATION RULE – FREE GOODS 2: UNIT REFERENCE

Formula '2' interprets the free goods quantities as being related to number of units.

Example:
A company offers free goods to their customers when they order certain materials. For example, when a customer orders 100 cases of Material A, the customer receives an additional 20 cases of Material A for free. The company would like to have the system interpret the 'Buy 100, Get 20 Free' agreement by granting 20 free for every full 100 purchased. For example, if the customer orders 162 cases, the system should automatically grant 20 for free [100 x (20/100)]. To accomplish this, the user would assign free goods calculation rule '2' to the free goods condition record.

CALCULATION RULE – FREE GOODS 3: WHOLE UNITS

Formula '3' grants to free goods only if whole units are ordered.

Example:
A company offers free goods to their customers when they order certain materials. For example, when a customer orders 100 cases of Material A, the customer receives an additional 20 cases of Material A for free. The company would like to have the system apply the 'Buy 100, Get 20 Free' agreement only if the customer orders in increments of 100. For example, if the customer orders 162

cases, the system should grant 0 for free since 162 is not a multiple of 100. To accomplish this, the user would assign free goods calculation rule '3' to the free goods condition record.