OmniPrice

Cavallo Support - 2025-06-11 - Sales

SalesPad Desktop offers a stored procedure for running OmniPrice.

Please note that this spcp is provided as is, and modifications may require development assistance.

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[spcpGetItemPriceNonExt] @Customer_Num AS VARCHAR(50) = NULL
,@Item_Number AS VARCHAR(50) = NULL
,@Currency AS VARCHAR(50) = NULL
,@UOfM AS VARCHAR(50) = NULL
,@Qty AS NUMERIC(19, 5) = NULL
,@Date AS DATETIME = NULL
,@Price_Level AS VARCHAR(50) = NULL
,@Warehouse_Code AS VARCHAR(50) = NULL
,@CalcMissingUOfMPrice AS BIT = 0
,@Address_Code AS VARCHAR(50) = NULL
,@Priority AS VARCHAR(100) = 'SEQUENCE'
,@Sales_Doc_Type AS VARCHAR(15) = NULL
,@Sales_Doc_ID AS VARCHAR(15) = NULL
,@Generic_Value AS VARCHAR(MAX) = NULL
,@SerialLotBin AS VARCHAR(50) = NULL
AS
SET NOCOUNT ON

DECLARE @Cust_Price_Level AS VARCHAR(25)
,@Item_Price_Level AS VARCHAR(25)

SET @Cust_Price_Level = NULL
SET @Item_Price_Level = NULL

IF (@Qty = 0)
SELECT @Qty = 1

IF (
(@Price_Level IS NULL)
AND (@Customer_Num IS NOT NULL)
)
BEGIN
--print 'Using customer''s default price level.'
SELECT @Price_Level = PRCLEVEL
FROM RM00101(NOLOCK)
WHERE CUSTNMBR = @Customer_Num
END

IF (
(@Customer_Num IS NOT NULL)
AND (@Item_Number IS NOT NULL)
)
BEGIN
IF (
(
SELECT count(*)
```

```
FROM IV00108 AS p(NOLOCK)
WHERE p.ITEMNMBR = @Item_Number
AND (
@Currency IS NULL
OR p.CURNCYID = @Currency
OR p.CURNCYID = ''
)
AND (
@Price_Level IS NULL
OR p.PRCLEVEL = @Price_Level
)
AND (
@UOfM IS NULL
OR @UOfM = p.UOFM
)
AND (
@Qty IS NULL
OR (
@Qty >= FROMQTY
AND @Qty <= TOQTY
)
)
) = 0
)
BEGIN
--print 'Pulling item''s default price level; Customer''s or Document''s price level has not been setup for the item.'
SELECT @Cust_Price_Level = isnull(PRCLEVEL, '')
FROM RM00101 AS c(NOLOCK)
WHERE CUSTNMBR = @Customer_Num
AND (
(
SELECT count(*)
FROM IV00108 AS p(NOLOCK)
WHERE p.ITEMNMBR = @Item_Number
AND (
@Currency IS NULL
OR p.CURNCYID = @Currency
OR p.CURNCYID = ''
)
AND (
@Price_Level IS NULL
OR p.PRCLEVEL = c.PRCLEVEL
)
AND (
@UOfM IS NULL
OR @UOfM = p.UOFM
)
AND (
@Qty IS NULL
OR (
@Qty >= FROMQTY
AND @Qty <= TOQTY
)
)
) > 0
)

SELECT @Item_Price_Level = isnull(PRCLEVEL, '')
FROM IV00101 AS i(NOLOCK)
WHERE ITEMNMBR = @Item_Number
AND (
(
SELECT count(*)
FROM IV00108 AS p(NOLOCK)
WHERE p.ITEMNMBR = @Item_Number
AND (
@Currency IS NULL
```

```
OR p.CURNCYID = @Currency
OR p.CURNCYID = ''
)
AND (
@Price_Level IS NULL
OR p.PRCLEVEL = i.PRCLEVEL
)
AND (
@UOfM IS NULL
OR @UOfM = p.UOFM
)
AND (
@Qty IS NULL
OR (
@Qty >= FROMQTY
AND @Qty <= TOQTY
)
)
) > 0
)

/* Set Price Level */
IF @Cust_Price_Level <> ''
BEGIN
SET @Price_Level = @Cust_Price_Level --Using Customer Price Level
END
ELSE IF @Item_Price_Level <> ''
BEGIN
SET @Price_Level = @Item_Price_Level --Using Item Price Level
END
END
END

DECLARE @multiplier AS NUMERIC(19, 5)

SET @multiplier = 0

IF (
(
SELECT count(*)
FROM IV00108(NOLOCK)
WHERE ITEMNMBR = @Item_Number
AND UOFM = @UOfM
AND PRCLEVEL = @Price_Level
) = 0
AND @CalcMissingUOfMPrice = 1
)
BEGIN
--get @mutliplier from baseuofm conversion factor
DECLARE @UOfMBase AS VARCHAR(50)
DECLARE @prevUOfM AS VARCHAR(50)

SELECT @UOfMBase = SELNGUOM
FROM IV00101(NOLOCK)
WHERE itemnmbr = @Item_Number

SET @prevUOfM = @UOfM
SET @UOfM = @UOfMBase
SET @multiplier = 1
END

------------------------------------------------------------------------
--Get Great Plains Standard Pricing
------------------------------------------------------------------------
DECLARE @ItemType INT

SET @ItemType = (
```

```
SELECT ITEMTYPE
FROM IV00101(NOLOCK)
WHERE ITEMNMBR = @Item_Number
)

IF (@ItemType = 3)
BEGIN
DECLARE @KitCost AS DECIMAL(19, 5)

SET @KitCost = ISNULL((
SELECT SUM((
CASE i.PRICMTHD
WHEN 3
THEN i.CURRCOST
WHEN 4
THEN i.CURRCOST
WHEN 5
THEN i.CURRCOST
ELSE i.STNDCOST
END
) * kit.CMPITQTY * uofmd.QTYBSUOM)
FROM IV00104 AS kit(NOLOCK)
INNER JOIN IV00101 AS i(NOLOCK) ON kit.CMPTITNM = i.ITEMNMBR
INNER JOIN IV40202 AS uofmd(NOLOCK) ON uofmd.UOMSCHDL = i.UOMSCHDL
AND uofmd.UOFM = kit.CMPITUOM
WHERE kit.ITEMNMBR = @Item_Number
GROUP BY kit.ITEMNMBR
), 0)
END

SELECT @Item_Number AS ITEMNMBR
,i.DECPLCUR - 1 AS Item_Curr_Dec
,@Customer_Num AS CUSTNMBR
,p.CURNCYID
,Price_Source = cast('STANDARD' AS NVARCHAR(30))
--need to cast it because other strings for Price_Source could be longer
,p.UOFM
,FROMQTY
,TOQTY
,lp.LISTPRCE
,UOMPRICE
,CURRCOST
,STNDCOST
,p.QTYBSUOM
,CAST((
CASE i.PRICMTHD
WHEN 1
THEN UOMPRICE --amount
WHEN 2
THEN UOMPRICE / 100 * lp.LISTPRCE * p.QTYBSUOM --% of list
WHEN 3
THEN CASE i.ITEMTYPE
WHEN 3
THEN (UOMPRICE / 100 * @KitCost * p.QTYBSUOM) + (@KitCost * p.QTYBSUOM)
ELSE (UOMPRICE / 100 * i.CURRCOST * p.QTYBSUOM) + (i.CURRCOST * p.QTYBSUOM) --% markup of
currcost
END
WHEN 4
THEN CASE i.ITEMTYPE
WHEN 3
THEN (UOMPRICE / 100 * @KitCost * p.QTYBSUOM) + (@KitCost * p.QTYBSUOM)
ELSE (UOMPRICE / 100 * i.STNDCOST * p.QTYBSUOM) + (i.STNDCOST * p.QTYBSUOM) --% markup of
standard cost
END
WHEN 5
THEN CASE i.ITEMTYPE
WHEN 3
```

```
THEN (@KitCost * p.QTYBSUOM) / (1 - UOMPRICE / 100)
ELSE (i.CURRCOST * p.QTYBSUOM) / (1 - UOMPRICE / 100)
END
--% margin of current cost
WHEN 6
THEN CASE i.ITEMTYPE
WHEN 3
THEN (@KitCost * p.QTYBSUOM) / (1 - UOMPRICE / 100)
ELSE (i.STNDCOST * p.QTYBSUOM) / (1 - UOMPRICE / 100)
END
--% margin of standard cost
END
) AS NUMERIC(19, 5)) AS PRICE
,ROUNDTO
,ROUNDHOW
,RNDGAMNT
,UOMPRICE AS UPPRICE
,UOMPRICE AS DOWNPRICE
,UOMPRICE AS WORKVAR
--,PRICESEQ = CASE p.PRCLEVEL
-- WHEN @Price_Level
-- THEN 1
-- WHEN @Cust_Price_Level
-- THEN 2
-- WHEN @Item_Price_Level
-- THEN 3
-- ELSE 4
-- END
,PRICESEQ = 1 --[Great Plains Standard Pricing]
,Discount_Percent = cast(0.0 AS NUMERIC(19, 5))
,Price_Engine_ID = 0
,p.PRCLEVEL AS Price_Level
INTO #tmp
FROM IV00108 AS p(NOLOCK)
INNER JOIN IV00101 AS i(NOLOCK) ON i.ITEMNMBR = p.ITEMNMBR
INNER JOIN IV00105 AS lp(NOLOCK) ON lp.ITEMNMBR = i.ITEMNMBR
AND lp.CURNCYID = p.CURNCYID
INNER JOIN IV00107 AS o(NOLOCK) ON o.ITEMNMBR = p.ITEMNMBR
AND o.CURNCYID = p.CURNCYID
AND o.PRCLEVEL = p.PRCLEVEL
AND o.UOFM = p.UOFM
LEFT JOIN MC40000 AS currSetup(NOLOCK) ON 1 = 1
LEFT JOIN DYNAMICS..MC40200 AS funcCurr(NOLOCK) ON funcCurr.CURNCYID = currSetup.FUNLCURR
WHERE p.ITEMNMBR = @Item_Number
AND (
@Currency IS NULL
OR p.CURNCYID = @Currency
OR p.CURNCYID = ''
)
AND (
@Price_Level IS NULL
OR p.PRCLEVEL = @Price_Level
)
AND (
@UOfM IS NULL
OR @UOfM = p.UOFM
)
AND (
@Qty IS NULL
OR (
@Qty >= FROMQTY
AND @Qty <= TOQTY
)
)

DECLARE @c AS INT
,@p AS DECIMAL
```

```
SET @c = (
SELECT count(*)
FROM #tmp
)

IF (@multiplier > 0)
BEGIN
--update @tmp
DECLARE @newPrice AS NUMERIC(19, 5)
DECLARE @UofMQty AS NUMERIC(19, 5)
DECLARE @UofMSchedule AS VARCHAR(50)

SELECT @UofMSchedule = UOMSCHDL
FROM IV00101(NOLOCK)
WHERE itemnmbr = @Item_Number

SELECT @newPrice = UOMPRICE
FROM IV00108(NOLOCK)
WHERE itemnmbr = @Item_Number
AND UOFM = @UOfMBase

SELECT @UofMQty = QTYBSUOM
FROM IV40202 AS u(NOLOCK)
WHERE u.UOFM = @prevUOfM
AND u.UOMSCHDL = @UofMSchedule
AND EQUIVUOM = @UOfMBase

PRINT @UofMQty
PRINT @newPrice

SET @newPrice = @newPrice * @UofMQty

UPDATE #tmp
SET PRICE = @newPrice
END

------------------------------------------------------------------------
--Multiple Of
------------------------------------------------------------------------
UPDATE #tmp
SET UPPRICE = cast(PRICE / RNDGAMNT AS INT) * RNDGAMNT + RNDGAMNT
,--multiple of , up the price
DOWNPRICE = cast(PRICE / RNDGAMNT AS INT) * RNDGAMNT --multiple of , lower the price
WHERE ROUNDHOW = 1 --multiple of
AND ROUNDTO != 1 --none no adjustment(none, up, down, nearest)

------------------------------------------------------------------------
--Ends With
------------------------------------------------------------------------
UPDATE #tmp
SET WORKVAR = CASE
WHEN cast(RNDGAMNT AS INT) = 0
THEN 1
ELSE POWER(10, len(cast(cast(RNDGAMNT AS INT) AS VARCHAR(25))))
END
WHERE ROUNDHOW = 2 --ends with
AND ROUNDTO != 1 --none no adjustment(none, up, down, nearest)

UPDATE #tmp
SET UPPRICE = cast(PRICE / WORKVAR AS INT) * WORKVAR + RNDGAMNT
,--ends with , up the price
DOWNPRICE = cast(PRICE / WORKVAR AS INT) * WORKVAR - (WORKVAR - RNDGAMNT) --ends with , lower
the price
WHERE ROUNDHOW = 2 --multiple of
AND ROUNDTO != 1 --none no adjustment(none, up, down, nearest)
```

```
UPDATE #tmp
SET UPPRICE = UPPRICE + WORKVAR
,DOWNPRICE = UPPRICE
WHERE ROUNDHOW = 2 --multiple of
AND ROUNDTO != 1 --none no adjustment(none, up, down, nearest)
AND UPPRICE < PRICE

---------------------------------------------------------------------
--Update the Price based on the RoundTo (2/up, 3/down, or 4/nearest)
---------------------------------------------------------------------
UPDATE #tmp
SET PRICE = CASE ROUNDTO
WHEN 2
THEN UPPRICE
WHEN 3
THEN DOWNPRICE
WHEN 4
THEN CASE
WHEN (PRICE - DOWNPRICE) < (UPPRICE - PRICE)
THEN DOWNPRICE
ELSE UPPRICE
END
END
WHERE ROUNDTO != 1

UPDATE #tmp
SET PRICE = round(PRICE, Item_Curr_Dec)

-----------------------------------------------------------------------------------------------------------------------------------
-- Omni Price
-----------------------------------------------------------------------------------------------------------------------------------
IF (isnull(@Customer_Num, '') <> '')
BEGIN
DECLARE @omniPrice DECIMAL
,@gpPrice NUMERIC(19, 5)
,@CalcUnitPrice NUMERIC(19, 5)
,@StartingPrice NUMERIC(19, 5)
,@ContractID CHAR(15)
,@MarkDownAmount NUMERIC(19, 5)
,@MarkDownPercent NUMERIC(19, 5)
,@MarkDownMode SMALLINT
,@PricingMode SMALLINT
,@ContractLineDexRowID INT

SELECT @gpPrice = PRICE
FROM #tmp

--Grab defaults to pass in
DECLARE @SalespersonID CHAR(15)
DECLARE @SalesTerritory CHAR(15)
DECLARE @CurrencyID CHAR(15)
DECLARE @ShippingMethod CHAR(15)
DECLARE @PaymentTermsID CHAR(21)
DECLARE @LocationCode CHAR(11)
--DECLARE @UofMSchedule       CHAR(30)
DECLARE @BillToAddressCode CHAR(15)
DECLARE @ShipToAddressCode CHAR(15)

SELECT @SalespersonID = ISNULL(SLPRSNID, '')
,@SalesTerritory = ISNULL(SALSTERR, '')
,@CurrencyID = ISNULL(CURNCYID, '')
,@ShippingMethod = ISNULL(SHIPMTHD, '')
,@PaymentTermsID = ISNULL(PYMTRMID, '')
,@BillToAddressCode = ISNULL(PRBTADCD, '')
,@ShipToAddressCode = ISNULL(PRSTADCD, '')
FROM RM00101(NOLOCK)
WHERE CUSTNMBR = @Customer_Num
```

```
SELECT @LocationCode = ISNULL(@Warehouse_Code, '')
,@UofMSchedule = ISNULL(UOMSCHDL, '')
FROM IV00101(NOLOCK)
WHERE ITEMNMBR = @Item_Number

--RETURNS
-- @[OPCalculateUnitPrice2] TABLE (
-- [CalcUnitPrice] NUMERIC(19,5),
-- [StartingPrice] NUMERIC(19,5),
-- [ContractID] CHAR(15),
-- [MarkDownAmount] NUMERIC(19,5),
-- [MarkDownPercent] NUMERIC(19,5),
-- [MarkDownMode] SMALLINT,
-- [PricingMode] SMALLINT,
-- [ContractLineDexRowID] INT
-- )
SELECT TOP 1 @CalcUnitPrice = [CalcUnitPrice]
,@StartingPrice = [StartingPrice]
,@ContractID = [ContractID]
,@MarkDownAmount = [MarkDownAmount]
,@MarkDownPercent = [MarkDownPercent]
,@MarkDownMode = [MarkDownMode]
,@PricingMode = [PricingMode]
,@ContractLineDexRowID = [ContractLineDexRowID]
FROM [dbo].[OPCalculateUnitPrice2](@gpPrice, @Customer_Num, @Item_Number, @Date, @SalespersonID --
SalesPerson
, @SalesTerritory --SalesTerritory
, @Price_Level --PriceLevel
, @Currency --CurrencyID
, @ShippingMethod --ShippingMethod
, @PaymentTermsID --PaymentTerms
, @LocationCode --Location Code
, @UofM --UofM
, @Qty --Qty
, @BillToAddressCode --BillToAdrscode
, @ShipToAddressCode --ShipToAdrsCode
, '', '', '', '', '', '', '', '')
ORDER BY [CalcUnitPrice] ASC

IF (
@MarkDownMode = 4
OR @MarkDownMode = 5
)
BEGIN
UPDATE #tmp
SET Discount_Percent = @MarkDownPercent

--MODIFIED 12/18/2014
--CHANGING THIS WILL SUPPORT Discount_Percent for OmniPrice
UPDATE #tmp
SET Price_Source = 'SPCUSTOMERSPECIAL'

--END MODIFIED
UPDATE #tmp
SET PRICE = @gpPrice

UPDATE #tmp
SET Price_Level = 'SALESPAD'
END
ELSE
BEGIN
UPDATE #tmp
SET PRICE = @CalcUnitPrice --, Price_Description = @ContractID
END
--if(@omniPrice < @gpPrice) begin
-- update #tmp set Discount_Percent = (1-round((@omniprice / @gpPrice), Item_Curr_Dec))
```

```
--end
END

-----------------------------------------------------------------------------------------------------------------------------------------------------
--Customer Special Pricing   (for SalesPad ---spGroupPricing table)
-----------------------------------------------------------------------------------------------------------------------------------------------------
/* Always use default currency for custom pricing  */
DECLARE @ci AS VARCHAR(25)
,@numDecimals AS INT

SELECT TOP 1 @ci = CURNCYID
,@numDecimals = DECPLCUR - 1
FROM DYNAMICS..MC40200 AS a(NOLOCK)
INNER JOIN MC40000 AS b(NOLOCK) ON a.CURRNIDX = b.FUNCRIDX

IF (NOT @Customer_Num IS NULL)
BEGIN
INSERT INTO #tmp
SELECT TOP 1 @Item_Number AS ITEMNMBR
,DECPLCUR = 0
,@Customer_Num AS CUSTNMBR
,isnull(@ci, '') AS CURNCYID
,'SPCUSTOMERSPECIAL' AS Price_Source
,UOfM AS UOFM
,FROMQTY = 0
,TOQTY = 999999999999.00000
,LISTPRCE = 0
,UOMPRICE = 0
,CURRCOST = 0
,STNDCOST = 0
,QTYBSUOM = 0
,Price
,ROUNDTO = 0
,ROUNDHOW = 0
,RNDGAMNT = 0
,UPPRICE = 0
,DOWNPRICE = 0
,WORKVAR = 0
,PRICESEQ = 3 --[Customer Special Pricing]
,cp.Discount_Percent
,Price_Engine_ID = cp.Custom_Price_ID
,Price_Level = @Price_Level
FROM spCustomPrice AS cp(NOLOCK)
WHERE cp.Item_Number = @Item_Number
AND cp.Customer_Num = @Customer_Num
AND (
UOfM = @UOfM
OR UOfM = ''
)
AND (
(
isnull(cp.Begin_Date, '1/1/1900') = '1/1/1900'
AND isnull(cp.End_Date, '1/1/1900') = '1/1/1900'
)
OR (
isnull(@Date, '1/1/1900') BETWEEN DATEADD(dd, DATEDIFF(dd, 0, cp.Begin_Date), 0)
AND DATEADD(dd, DATEDIFF(dd, 0, cp.End_Date), 0)
)
)
AND (
(
isnull(Qty_To, 0) = 0
AND isnull(Qty_From, 0) = 0
)
OR (
@Qty >= Qty_From
AND @Qty <= Qty_To
```

```
)
)
ORDER BY Begin_Date
,End_Date
,UOfM ASC
END

-----------------------------------------------------------------------------------------------------------------------------------------------------------
--Group Pricing   (for SalesPad ---spGroupPricing table)
-----------------------------------------------------------------------------------------------------------------------------------------------------------
INSERT INTO #tmp
SELECT TOP 1 @Item_Number AS ITEMNMBR
,a.DECPLCUR - 1 AS Item_Curr_Dec
,@Customer_Num AS CUSTNMBR
,isnull(@ci, '') AS CURNCYID
,Price_Source = 'SPGROUP'
,gpi.UOfM
,FROMQTY = gpi.Qty_Start
,TOQTY = gpi.Qty_End
,LISTPRCE = isnull(lp.LISTPRCE, 0)
,UOMPRICE = 0
,CURRCOST = 0
,STNDCOST = 0
,QTYBSUOM = 0
,Price = CASE gpi.Price_Type
WHEN 'FixedAmount'
THEN gpi.Price_Value
WHEN 'PercentAdjustment' --list price * percentage
THEN lp.LISTPRCE * p.QTYBSUOM * gpi.Price_Value / 100
WHEN 'AmountAdjustment' --list price + adjustment
THEN lp.LISTPRCE + gpi.Price_Value
WHEN 'MarkupPercent' --cost + (cost * markup percentage)
THEN (
CASE gpi.Price_Source
WHEN 'CurrentCost'
THEN a.CURRCOST
ELSE a.STNDCOST
END
) * p.QTYBSUOM + (
(
CASE gpi.Price_Source
WHEN 'CurrentCost'
THEN a.CURRCOST
ELSE a.STNDCOST
END
) * p.QTYBSUOM * gpi.Price_Value / 100
)
WHEN 'MarginPercent' --cost / (1 -margin percentage)
THEN (
CASE gpi.Price_Source
WHEN 'CurrentCost'
THEN a.CURRCOST
ELSE a.STNDCOST
END
) * p.QTYBSUOM / (1 - gpi.Price_Value / 100)
ELSE 0
END
,ROUNDTO = 0
,ROUNDHOW = 0
,RNDGAMNT = 0
,UPPRICE = 0
,DOWNPRICE = 0
,WORKVAR = 0
,PRICESEQ = 2 --[Group Pricing]
,[Discount_Percent] = 0 --not used
,Price_Engine_ID = gpi.ItemDetail_ID
,Price_Level = @Price_Level
```

```
FROM IV00101 AS a(NOLOCK)
INNER JOIN IV40201 AS b(NOLOCK) ON b.UOMSCHDL = a.UOMSCHDL
INNER JOIN spGroupPricingItemDetail AS gpi(NOLOCK) ON
--Based On Item Number
(
(
gpi.Field_Name = 'Item Number'
OR gpi.Field_Name = ''
OR gpi.Field_Name IS NULL
)
AND a.ITEMNMBR = gpi.Item_Number
)
OR
--Based on Item Class
(
gpi.Field_Name = 'Item Class'
AND a.ITMCLSCD = gpi.Item_Number
)
INNER JOIN spGroupPricing AS gp(NOLOCK) ON gp.Group_ID = gpi.Group_ID
LEFT JOIN spGroupPricingCustomerDetail AS gpc(NOLOCK) ON gpi.Group_ID = gpc.Group_ID
AND (
gp.Apply_To_All_Customers = 1
OR gpc.Customer_Num = @Customer_Num
)
LEFT JOIN IV00105 AS lp(NOLOCK) ON lp.ITEMNMBR = a.ITEMNMBR
AND (
lp.CURNCYID = @ci
OR lp.CURNCYID = ''
) --the null check is for when running non-multicurrency
LEFT JOIN IV00108 AS p(NOLOCK) ON p.ITEMNMBR = a.ITEMNMBR
AND p.CURNCYID = lp.CURNCYID
AND p.UOFM = gpi.UofM
WHERE a.ITEMNMBR = @Item_Number
AND gpi.UOfM = @UOfM
AND (
(
isnull(gp.Begin_Date, '1/1/1900') = '1/1/1900'
AND isnull(gp.End_Date, '1/1/1900') = '1/1/1900'
)
OR (
isnull(@Date, '1/1/1900') BETWEEN DATEADD(dd, DATEDIFF(dd, 0, gp.Begin_Date), 0)
AND DATEADD(dd, DATEDIFF(dd, 0, gp.End_Date), 0)
)
)
AND (
(
isnull(gpi.Qty_Start, 0) = 0
AND isnull(gpi.Qty_End, 0) = 0
)
OR (
@Qty >= gpi.Qty_Start
AND @Qty <= gpi.Qty_End
)
)
AND gp.Is_Active = 1
AND gp.Is_Deleted = 0
AND (
(
gpc.Customer_Num IS NULL
AND gp.Apply_To_All_Customers = 1
)
OR gpc.Customer_Num = @Customer_Num
) --return Group Pricing w/ matching Customer Number or Pricing is set for all customers
AND (
gp.Apply_To_All_Customers = 1
OR gpc.Address_Code = @Address_Code
OR gpc.Address_Code = ''
```

```
OR gpc.Address_Code IS NULL
) --return Group Pricing w/ matching address code or w/ blank address
AND (
gp.Sales_Doc_Type = @Sales_Doc_Type
OR gp.Sales_Doc_Type = ''
OR gp.Sales_Doc_Type IS NULL
) --return Group Pricing w/ matching Sales_Doc_Type or w/ blank Sales_Doc_Type
AND (
gp.Sales_Doc_ID = @Sales_Doc_ID
OR gp.Sales_Doc_ID = ''
OR gp.Sales_Doc_ID IS NULL
) --return Group Pricing w/ matching Sales_Doc_ID or w/ blank Sales_Doc_ID
AND (
gp.Warehouse_Code = @Warehouse_Code
OR gp.Warehouse_Code = ''
OR gp.Warehouse_Code IS NULL
) --return Group Pricing w/ matching Warehouse_Code or w/ blank Warehouse_Code
ORDER BY gpc.Address_Code DESC
,gp.Sales_Doc_Type DESC
,gp.Sales_Doc_ID DESC
,gp.Warehouse_Code DESC
,gp.Sequence
,gp.Begin_Date
,gp.End_Date
,UOfM ASC --Priority: 1. Group Pricing w/ matching addr, 2. Group Pricing w/ lower Sequence #, 3. dates, 4.
UofM

-----------------------------------------------------------------------------------------------------------------------------------------
--Contract Pricing   (for SalesPad ---spvCustomerContract table)
-----------------------------------------------------------------------------------------------------------------------------------------
IF (NOT @Customer_Num IS NULL)
BEGIN
INSERT INTO #tmp
SELECT TOP 1 @Item_Number AS ITEMNMBR
,DECPLCUR = 0
,@Customer_Num AS CUSTNMBR
,isnull(@ci, '') AS CURNCYID
,Price_Source = 'SPCONTRACT'
,cd.UOfM
,FROMQTY = 1
,TOQTY = CASE Contract_Type
WHEN 'Detail'
THEN (cd.Contract_Qty - cd.Purchased_Qty)
WHEN 'Grouped'
THEN (c.Contract_Qty - c.Purchased_Qty)
ELSE 0
END
,LISTPRCE = 0
,UOMPRICE = 0
,CURRCOST = 0
,STNDCOST = 0
,QTYBSUOM = 0
,Price = ISNULL(cd.Contract_Price, 0)
,ROUNDTO = 0
,ROUNDHOW = 0
,RNDGAMNT = 0
,UPPRICE = 0
,DOWNPRICE = 0
,WORKVAR = 0
,PRICESEQ = 4 --[Contract Pricing]
,0
,Price_Engine_ID = cd.Detail_Number
,Price_Level = @Price_Level
FROM spCustomerContract(NOLOCK) AS c
INNER JOIN spCustomerContractDetail(NOLOCK) AS cd ON cd.Contract_Number = c.Contract_Number
WHERE cd.Item_Number = @Item_Number
AND c.Customer_Number = @Customer_Num
```

```
AND (
(
isnull(c.Begin_Date, '1/1/1900') = '1/1/1900'
AND isnull(c.End_Date, '1/1/1900') = '1/1/1900'
)
OR CAST(ISNULL(@Date, '1/1/1900') AS DATETIME) BETWEEN CAST(c.Begin_Date AS DATETIME)
AND CAST(c.End_Date AS DATETIME)
)
AND (
(isnull(c.Contract_Qty, 0) - isnull(c.Purchased_Qty, 0)) > 0
OR (isnull(cd.Contract_Qty, 0) - isnull(cd.Purchased_Qty, 0)) > 0
)
AND c.[Enabled] = 1
AND c.[Deleted] = 0
AND cd.[Deleted] = 0
END

-----------------------------------------------------------------------
--Output
-----------------------------------------------------------------------
SELECT Customer_Num = isnull(@Customer_Num, '')
,Currency = CURNCYID
,Price_Description = ''
,Start_Date = @Date
,End_Date = @Date
,Item_Number = ITEMNMBR
,UOfM = UOFM
,From_Qty = FROMQTY
,To_Qty = TOQTY
,Price = PRICE
,Qty = @Qty
,DATE = @Date
,Price_Level = ISNULL(@Price_Level, Price_Level)
,Price_Source
,PRICESEQ
,Discount_Percent = isnull(Discount_Percent, 0.0)
,Is_Promo = 0
,Markdown = 0
,Price_Engine_ID
,SerialLotBin = ''
FROM #tmp
ORDER BY CASE
WHEN @Priority = 'LOWEST PRICE'
THEN Price
END ASC
,CASE
WHEN @Priority = 'HIGHEST PRICE'
THEN Price
END DESC
,CASE
WHEN @Priority = 'SEQUENCE'
THEN PRICESEQ
END DESC
```