**Business One in Action – Tracking Price Changes for an Item**

**Applies to:**
SAP Business One, Logistics, Stock, General Logistics

**Summary:**
This document details how to display the different prices an item used to have throughout its history in SAP Business One. The beauty of this report is that it only shows the price changes for the item and nothing else.

**Author:** Noreen Armstrong, Senior Support Consultant, GSC, SAP Business One

**Company:** SAP

**Created on:** August 2010

**Author Bio:**
Noreen Armstrong joined SAP in 2004 and is currently active as Senior Support Consultant for SAP Business One. She has worked as both technical and business consultant.
# Table of Content

- Tracking Price Changes in Business One ................................................................. 3
- Related Contents ...................................................................................................... 14
- Copyright .................................................................................................................. 15
Tracking Price Changes in Business One

It is frequently required to review the changes to the price of an item. There are many reasons why this may happen; it may be to judge how often the price has changed, or to analyze the changes in the price over a period with a view to forecasting and determining the future profitability of the item.

In SAP Business One it is possible to view the changes to the item’s prices through the Item Master Data record of the item. To do this:

1. Open the Item Master Data record of the item
2. Go to Tools -> Show History
   a. Highlight the first row and then press Show Differences. The list of all changes which has occurred to this item is now displayed.
3. Click on the Filter, icon in the upper menu.
4. In the Filter Table window in the row 'Changed Field', set Rule to Equal and Value to Price list.

Result:
The Differences window now shows the history where the price changed only. All other changes are excluded.
An alternative option to view the changes to item prices is to use SQL queries. The OITM table records all the Item Master Data information. It is linked to the ITM1 table which records the current price for the item in all price lists where a price is defined. The AITM table records the history of the OITM and the AIT1 records the history of the ITM1. Using this information it is possible to track the changes to an items price.

The following is a sample query which displays only those records, where an item price has changed. The query will show if there has been a change in currency or if the current price has not yet been recorded to the history table.

Regarding the current price display, the AITM table records a row for each change that happens to the item in its Item Master Data record. The AITM will also record a new row if the price of an item is updated; this change is recorded in the AIT1. However, both tables are updated if either the price or an element of the item master data record is changed. The number of changes made to an item or its price is counted in the AITM and AIT1 by the field Log Instance. Each item has its own Log Instance count.

This query will show the current price, in the situation where the price has changed but no further update has been made to that item’s master data record. In these cases, the field ‘LogInstanc’ in the query below will display the Value ‘0’.

Please note:

In the query below, the writing in green is an explanation of what happens in each section of the query. This should help you to understand what information each section of the query is pulling from the database.
/*This query uses Common Table Expression (CTE) to return only those rows
where there has been a price change per item per price list.*/

With PriceTrack (Itemcode, pricelist, PriceListName, currency, price, updatedate, loginstanc) as

/* Price Changes History:
Select each item's price, per log instance, in a specific price
list and return those rows where the price is different */

Select distinct T0.itemcode, T1.PriceList, T7.listname, t1.currency, 
T1.Price, T0.UpdateDate, T1.loginstanc

FROM

AITM T0
  Inner Join AIT1 T1 on T0.itemcode = T1.Itemcode
    and t0.loginstanc = t1.loginstanc
  left outer join AIT1 T3 on T1.itemcode = T3.itemcode
    and t1.pricelist = T3.pricelist
    and t1.loginstanc > t3.loginstanc
  Inner join OPLN T7 on T1.PriceList = T7.ListNum and T7.ListNum = 
  T3.pricelist

where (t1.price <> t3.price )

/* Currency Change:
select price changes where only the currency changes*/

union

Select distinct T0.itemcode, T1.PriceList,t7.listname, t1.currency,
T1.Price, T0.UpdateDate, T1.loginstanc

FROM

AITM T0
Inner Join AIT1 T1 on T0.itemcode = T1.Itemcode
and t0.loginstanc = t1.loginstanc
left outer join AIT1 T3 on T1.itemcode = T3.itemcode
and t1.pricelist = T3.pricelist
and t1.loginstanc > T3.loginstanc

Inner join OPLN T7 on T1.PriceList = T7.ListNum and T7.ListNum = T3.pricelist

where (t1.price = t3.price and T1.currency <> T3.currency)

/*Current Price:
Select the current Price of the item. This also shows items where the
factor has changed or where the item price has been manually overwritten.
This query will only return results if the current price has not been
updated to the AIT1 table. Remember the AIT1 will also get updated.
If the item master data is updated (OITM and AITM), the log instance in
AITM and AIT1 must always be the same. Consequently, the log instance field
for this section will always show 0.*/

union

select t0.itemcode, t0.pricelist, T7.listname, T0.Currency, T0.Price, T2.UpdateDate, ''
from

OITM T2
Inner Join ITM1 T0 on T0.Itemcode = T2.itemcode
inner join AIT1 t1 on T0.ItemCode = t1.ItemCode and T0.PriceList = t1.pricelist
Inner join OPLN T7 on T0.PriceList = T7.ListNum

Where

T0.Price not in( T1.price )
and t1.Loginstanc in
(Select Top 1 t9.LogInstanc from AIT1 T9 where t9.ItemCode =

Group By
t0.itemcode, t0.pricelist, t0.price, T7.ListName, T0.currency, T2.UpdateDate

/* Exclude rows with no price change:
Exclude all those incidents where the price has not changed (That is a change has happened to the item master data but not to the price of the item in that price list).

e.g. log instance 5 shows a change to the price of $20 but there is no subsequent price change until log instance 10. This section removes all the irrelevant rows from the result.

set */

except

Select distinct T0.itemcode, T1.PriceList, T7.ListName, tl.currency, T1.Price, T0.UpdateDate, T1.loginstanc

FROM

AITM T0

Inner Join AIT1 T1 on T0.itemcode = T1.Itemcode

and t0.loginstanc = t1.loginstanc

left outer join AIT1 T3 on T1.itemcode = T3.itemcode

and t1.pricelist = T3.pricelist

and t1.loginstanc > t3.loginstanc

Inner join OPLN T7 on T1.PriceList = T7.ListNum and T7.ListNum = T3.pricelist

where t1.price = t3.price and t1.currency = T3.currency
/*Select all data from the common table expression. To filter the results by item or price list use a 'where' condition in the following statement.*/

select * from pricetrack

To filter the query results, include a WHERE clause in the final select query (select * from pricetrack). The exact fields to use in the 'where' clause comes from the very beginning of the query in the section:

“With PriceTrack (Itemcode, pricelist, PriceListName, currency, price, updatedate, loginstanc)"

For example, to filter the results based on ItemCode and PriceList:

select * from pricetrack

becomes

select * from pricetrack where itemcode = 'ItemcodeX' and pricelist = 1

To filter this query for the item '001_Brownie', as shown in the above screenshots, in Price List 1 the last section of the query would look as follows:

select * from pricetrack where itemcode = '001_Brownies' and pricelist = 1.

The results of this example query are:

<table>
<thead>
<tr>
<th>Item Code</th>
<th>Price List</th>
<th>Price List Name</th>
<th>Currency</th>
<th>Price</th>
<th>Update date</th>
<th>Log Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>001_brownie</td>
<td>1</td>
<td>Base Price</td>
<td>GBP</td>
<td>5</td>
<td>2010-08-12 00:00:00.000</td>
<td>5</td>
</tr>
<tr>
<td>001_brownie</td>
<td>1</td>
<td>Base Price</td>
<td>GBP</td>
<td>6</td>
<td>2010-08-12 00:00:00.000</td>
<td>0</td>
</tr>
<tr>
<td>001_brownie</td>
<td>1</td>
<td>Base Price</td>
<td>GBP</td>
<td>6.25</td>
<td>2010-08-12 00:00:00.000</td>
<td>6</td>
</tr>
</tbody>
</table>

If more columns are needed in the query, the new field needs to be added into each section of the query. The CTE expression in MS SQL dictates that each section of the query should have the same number of fields.

For example, if the Item description is required, alongside the Item Code, modify the query as highlighted in orange below:
/*This query uses Common Table Expression (CTE) to return only those rows where there is a price change per item per price list.*/

With PriceTrack (Itemcode, ItemName, pricelist, PriceListName, currency, price, updatedate, loginstanc) as

( /* Price Changes History: */
  Select distinct T0.itemcode, T0.ItemName, T1.PriceList, T7.listname, t1.currency, T1.Price, T0.UpdateDate, T1.loginstanc
  FROM
    AITM T0
    Inner Join AIT1 T1 on T0.itemcode = T1.Itemcode
    and t0.loginstanc = t1.loginstanc
    left outer join AIT1 T3 on T1.itemcode = T3.itemcode
    and t1.pricelist = T3.pricelist
    and t1.loginstanc > t3.loginstanc
    Inner join OPLN T7 on T1.PriceList = T7.ListNum and T7.ListNum = T3.pricelist

  where (t1.price <> t3.price )

  /* Currency Change: */
  select price changes where only the currency changes*/
  union

  Select distinct T0.itemcode, T0.ItemName, T1.PriceList,t7.listname, t1.currency, T1.Price, T0.UpdateDate, T1.loginstanc
FROM

AITM T0
  Inner Join AIT1 T1 on T0.itemcode = T1.ItemCode
    and t0.logInstanc = t1.logInstanc
  left outer join AIT1 T3 on T1.itemcode = T3.itemcode
    and t1.pricelist = T3.pricelist
    and t1.logInstanc > t3.logInstanc
  Inner join OPLN T7 on T1.PriceList = T7.ListNum and T7.ListNum = T3.pricelist

where (t1.price = t3.price and T1.currency <> T3.currency )

/****************** Current Price:  
Select current Price of the item. Also shows items where the factor has changed or where the item price has been manually overwritten. 

This query will only return results if the current price has not been updated to the AIT1 table. Remember the AIT1 will also get updated if the item master data is updated (OITM & AITM). The log instance in the AITM and the AIT1 must always be the same. */

union

select t0.itemcode, T2.ItemName, t0.pricelist, T7.listname, T0.Currency, T0.Price, T2.UpdateDate, ''
from

OITM T2
  Inner Join ITM1 T0 on T0.ItemCode = T2.itemcode
inner join AIT1 t1 on T0.ItemCode = t1.ItemCode and T0.PriceList = t1.pricelist
  Inner join OPLN T7 on T0.PriceList = T7.ListNum
Where

T0.Price not in( T1.price )
and t1.LogInstanc in

(Select Top 1 t9.LogInstanc from AIT1 T9 where t9.ItemCode = t1.ItemCode
Group By
t0.itemcode, t0.pricelist, t0.price, T7.ListName, T0.currency, T2.UpdateDate, T2.ItemName

/* Exclude rows with no price change: 
Exclude all those incidents where the price has not changed. 
e.g. log instance 5 shows a change to the price of $20 but there is no 
subsequent price change until log instance 10. This section removes all the 
irrelevant rows from the result 
set. */
except

Select distinct T0.itemcode, T0.ItemName, T1.PriceList, T7.ListName, t1.currency, T1.Price, T0.UpdateDate, T1.loginstanc

FROM

AITM T0
    Inner Join AIT1 T1 on T0.itemcode = T1.Itemcode
                    and t0.loginstanc = t1.loginstanc

left outer join AIT1 T3 on T1.itemcode = T3.itemcode
                    and t1.pricelist = T3.pricelist
                    and t1.loginstanc > t3.loginstanc

    Inner join OPLN T7 on T1.PriceList = T7.ListNum and T7.ListNum = T3.pricelist

where t1.price = t3.price and t1.currency = T3.currency

/* Select all data from the common table expression */
Notice that in the “Current Price” section of the query the Item Name field has been entered twice. This is necessary because this section contains a GROUP BY part and all new fields also need to be added in here.

Again, using the example of the item 001_brownie, if the above query is run then the results show the following:

<table>
<thead>
<tr>
<th>Item Code</th>
<th>Item Name</th>
<th>Price List</th>
<th>Price List Name</th>
<th>Currency</th>
<th>Price</th>
<th>Update date</th>
<th>Log Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>001_brownie</td>
<td>Dark Chocolate Brownie 74%</td>
<td>1</td>
<td>Base Price</td>
<td>GBP</td>
<td>5</td>
<td>2010-08-12 00:00:00.000</td>
<td>5</td>
</tr>
<tr>
<td>001_brownie</td>
<td>Dark Chocolate Brownie 74%</td>
<td>1</td>
<td>Base Price</td>
<td>GBP</td>
<td>6</td>
<td>2010-08-12 00:00:00.000</td>
<td>0</td>
</tr>
<tr>
<td>001_brownie</td>
<td>Dark Chocolate Brownie 74%</td>
<td>1</td>
<td>Base Price</td>
<td>GBP</td>
<td>6.25</td>
<td>2010-08-12 00:00:00.000</td>
<td>6</td>
</tr>
</tbody>
</table>

Note that in these results we see that one row has a Log Instance of 0. This indicates that GBP 6.00 is the current price of the item 001_brownie in Price List 1 and that there were no updates to the item master data subsequent to the price change.

This query can be saved to the Query Manager and run when needed. To do this, open the Query Generator and proceed to the Query Preview window. Open this window in edit mode (click on the pencil icon to the left of the window) and paste the query in.

Next, press Save to open the Query Manager and save the query in the appropriate category.

In order to add variables to the report when executing it in SAP Business One, please follow the instructions in note 730960.
Related Contents

- [SAP Business One in Action Catalog Page](#)
- [Link to DRC documentation](#)
- Note [730960](#)
- Note [1165947](#)
- For more information, visit the [Business One homepage](#).