

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.

The screenshot shows the 'Item Master Data' window for item '001_brownie'. The 'Show History...' dialog box is open, displaying a table of history records. The first row is highlighted, and the 'Show Differences' button is highlighted with a red box.

#	Instance	Object Code	Updated	User Name
1	1	001_brownie	12.08.2010	manager
2	2	001_brownie	12.08.2010	manager
3	3	001_brownie	12.08.2010	manager
4	4	001_brownie	12.08.2010	manager
5	5	001_brownie	12.08.2010	manager
6	6	001_brownie	12.08.2010	manager
7	7	001_brownie	12.08.2010	manager
8	8	001_brownie	12.08.2010	manager
9	9	001_brownie	12.08.2010	manager
10	10	001_brownie	12.08.2010	manager

#	Date	Changed Field	Old Value	New Value	User Name
2	12.08.2010	Item Description	Dark Chocolate Brownie	Dark Chocolate Brownie 74%	manager
3	12.08.2010	Item Description	Dark Chocolate Brownie	Dark Chocolate Brownie 74%	manager
4	12.08.2010	Item Description	Dark Chocolate Brownie	Dark Chocolate Brownie 74%	manager
4	12.08.2010	Item Group	Items	Bakery	manager
4	12.08.2010	Planning Method	MRP	None	manager
5	12.08.2010	Item Description	Dark Chocolate Brownie	Dark Chocolate Brownie 74%	manager
5	12.08.2010	Item Group	Items	Bakery	manager
5	12.08.2010	Property 1	No	Yes	manager
5	12.08.2010	Property 2	No	Yes	manager
5	12.08.2010	Planning Method	MRP	None	manager
5	12.08.2010	Price List	0.00	5.00	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	10.00	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	3.75	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	6.25	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	10.00	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	8.75	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	15.00	manager
5	12.08.2010	Price List Currency		GBP	manager
5	12.08.2010	Price List	0.00	11.25	manager
5	12.08.2010	Price List Currency		GBP	manager

- Click on the *Filter*, , icon in the upper menu.
- In the *Filter Table* window in the row 'Changed Field', set *Rule* to *Equal* and *Value* to *Price list*.

#	Field	Rule	Value	To Value
1	#			
2	Date			
3	Changed Field	Equal	Price List	
4	Old Value			
5	New Value			
6	User Name			

Result:

The *Differences* window now shows the history where the price changed only. All other changes are excluded.

#	Date	Changed Field	Old Value	New Value	User Name
5	12.08.2010	Price List	0.00	5.00	manager
5	12.08.2010	Price List	0.00	10.00	manager
5	12.08.2010	Price List	0.00	3.75	manager
5	12.08.2010	Price List	0.00	6.25	manager
5	12.08.2010	Price List	0.00	10.00	manager
5	12.08.2010	Price List	0.00	8.75	manager
5	12.08.2010	Price List	0.00	15.00	manager
5	12.08.2010	Price List	0.00	11.25	manager
6	12.08.2010	Price List	0.00	6.25	manager
6	12.08.2010	Price List	0.00	12.50	manager
6	12.08.2010	Price List	0.00	4.69	manager
6	12.08.2010	Price List	0.00	7.81	manager
6	12.08.2010	Price List	0.00	12.50	manager
6	12.08.2010	Price List	0.00	10.94	manager
6	12.08.2010	Price List	0.00	18.75	manager
6	12.08.2010	Price List	0.00	14.06	manager
7	12.08.2010	Price List	0.00	6.25	manager
7	12.08.2010	Price List	0.00	12.50	manager
7	12.08.2010	Price List	0.00	4.69	manager
7	12.08.2010	Price List	0.00	7.81	manager
7	12.08.2010	Price List	0.00	12.50	manager
7	12.08.2010	Price List	0.00	10.94	manager
7	12.08.2010	Price List	0.00	18.75	manager
7	12.08.2010	Price List	0.00	14.06	manager
8	12.08.2010	Price List	0.00	6.25	manager
8	12.08.2010	Price List	0.00	12.50	manager
8	12.08.2010	Price List	0.00	4.69	manager
8	12.08.2010	Price List	0.00	7.81	manager
8	12.08.2010	Price List	0.00	12.50	manager
8	12.08.2010	Price List	0.00	10.94	manager

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.

Query

```

/*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 =

```

```
t1.ItemCode and t1.PriceList = T9.PriceList order by T9.LogInstanc desc)
```

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, 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. 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:

Item Code	Price List	Price List Name	Currency	Price	Update date	Log Instance
001_brownie	1	Base Price	GBP	5	2010-08-12 00:00:00.000	5
001_brownie	1	Base Price	GBP	6	2010-08-12 00:00:00.000	0
001_brownie	1	Base Price	GBP	6.25	2010-08-12 00:00:00.000	6

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 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, 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
and t1.PriceList = T9.PriceList order by T9.LogInstanc desc)

```

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*/
```

```
select * from pricetrack where Itemcode = '001_brownie' and pricelist = 1
```

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:

Item Code	Item Name	Price List	Price List Name	Currency	Price	Update date	Log Instance
001_brownie	Dark Chocolate Brownie 74%	1	Base Price	GBP	5	2010-08-12 00:00:00.000	5
001_brownie	Dark Chocolate Brownie 74%	1	Base Price	GBP	6	2010-08-12 00:00:00.000	0
001_brownie	Dark Chocolate Brownie 74%	1	Base Price	GBP	6.25	2010-08-12 00:00:00.000	6

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.

The screenshot shows the SAP Query Preview window. At the top, there is a pencil icon in a red box, indicating edit mode. Below it, the query text is displayed: `/*Select all data from the common table expression*/` followed by `select * from pricetrack |`. Below the query text is a table with 21 rows and 8 columns: #, Itemcode, ItemName, pricelist, PriceListName, currency, price, updatedate, and loginstanc. The table shows various price lists for the item 001_brownie, including Base Price, Discount Purchase Price, Regular Purchase Price, and Distributor Sales Price. At the bottom of the window, there are buttons for Execute, Cancel, Reverse Table, Save (highlighted with a red box), and Open.

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](#).

Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.