Search query examples V2

Purpose

Please note that the information in this presentation is NOT aimed to replace educations in IES. This is presentation is only compiled to give an insight in usable search queries. It does not cover any setup, how to think with more advanced searches etc

It is highly recommended that separate IES educations are attended and at a bare minimum the IES “M3 how to” videos should have been watched. These are accessible through the M3 homepage on the sales portal (link Quick studies) or through the Infor education portal. As an example of M3 IES educations please see Infor educations and a course like M3: v13x Configuring ION Enterprise Search

Version dependency

Please note that this information is based on current IES version (v11.0) and M3 version (13.3/13.4) at the time of writing. Some of the described features are not available in earlier M3 or IES versions. For more information about version dependency please see published NCR documents.

Topics

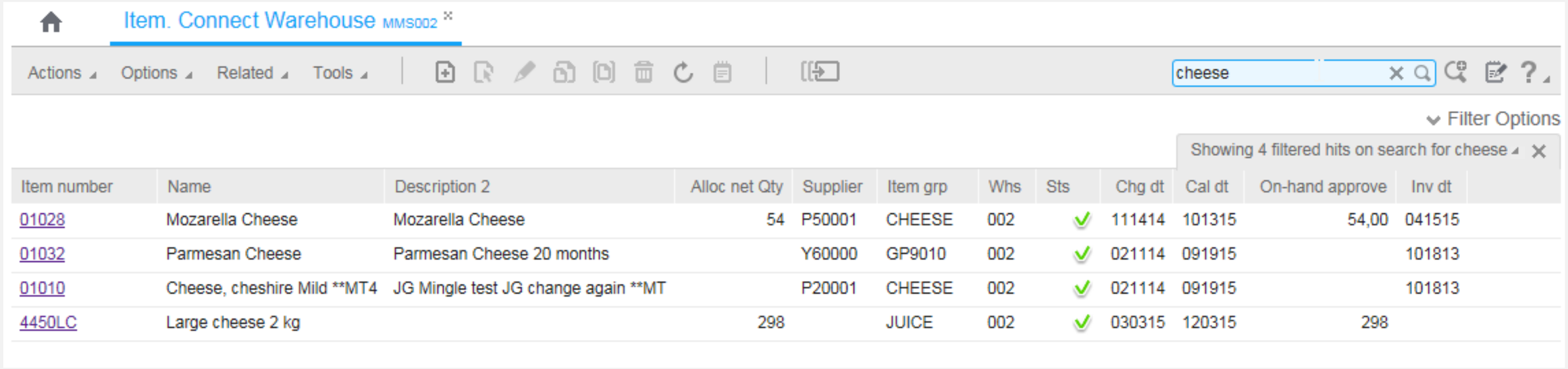
- Simple search
- Advanced search
- Search including reserved words
- M3 IES usage through API calls
- Multi level key search
- Very advanced search queries
- Lucene references

Simple search

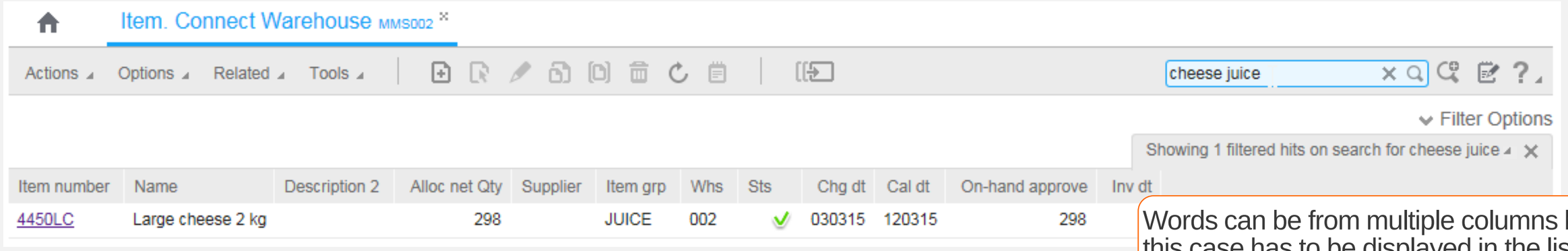
Simple search

This is the simplest form of search with entry of full words.

Note that search is always case insensitive.



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
01028	Mozarella Cheese	Mozarella Cheese	54	P50001	CHEESE	002	✓	111414	101315	54,00	041515
01032	Parmesan Cheese	Parmesan Cheese 20 months		Y60000	GP9010	002	✓	021114	091915		101813
01010	Cheese, cheshire Mild **MT4	JG Mingle test JG change again **MT		P20001	CHEESE	002	✓	021114	091915		101813
4450LC	Large cheese 2 kg		298		JUICE	002	✓	030315	120315	298	

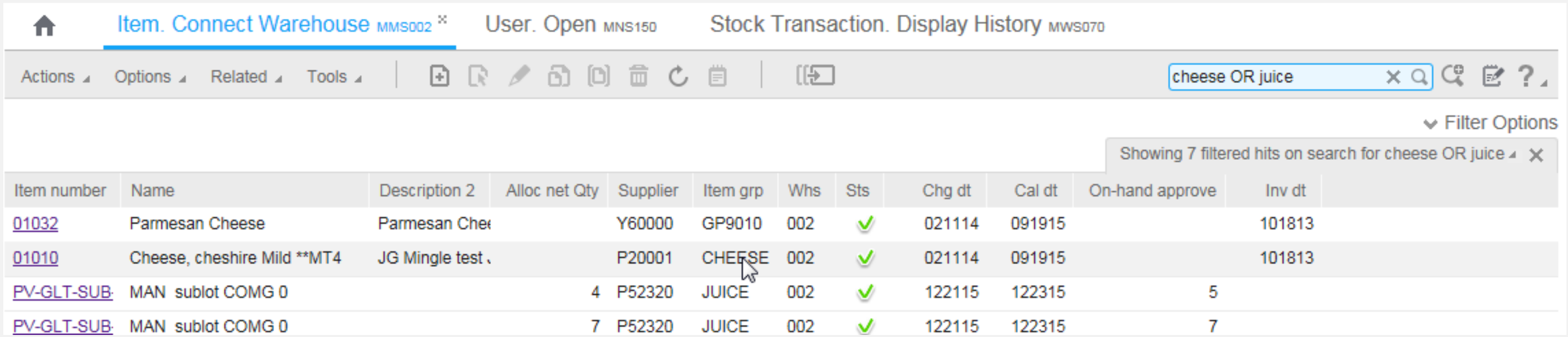


Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
4450LC	Large cheese 2 kg		298		JUICE	002	✓	030315	120315	298	

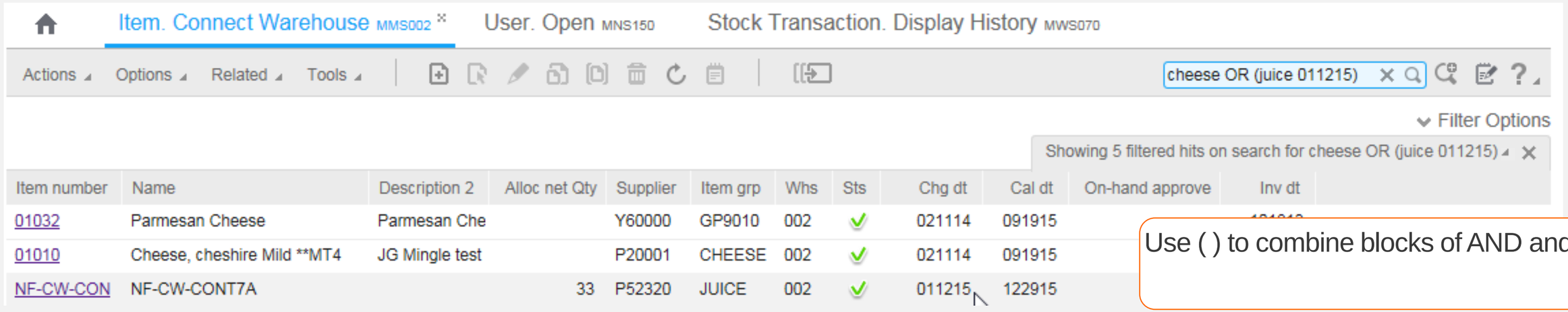
Words can be from multiple columns but in this case has to be displayed in the list

Simple search

Default if using multiple words in search is AND meaning all words has to exist in one or more columns. Using OR means one of them must exist in one of the columns



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
01032	Parmesan Cheese	Parmesan Che		Y60000	GP9010	002	✓	021114	091915		101813
01010	Cheese, cheshire Mild **MT4	JG Mingle test		P20001	CHEESE	002	✓	021114	091915		101813
PV-GLT-SUB	MAN sublot COMG 0		4	P52320	JUICE	002	✓	122115	122315	5	
PV-GLT-SUB	MAN sublot COMG 0		7	P52320	JUICE	002	✓	122115	122315	7	

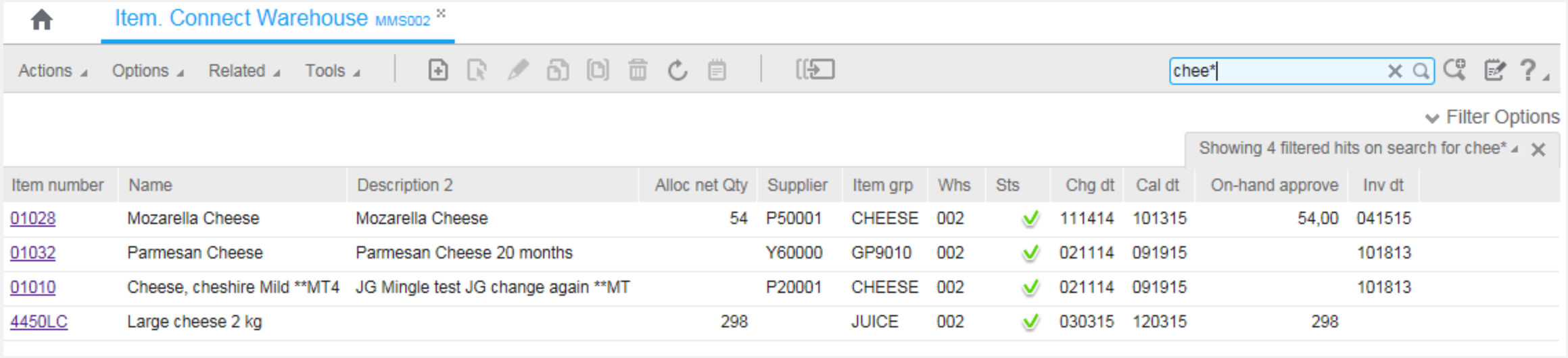


Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
01032	Parmesan Cheese	Parmesan Che		Y60000	GP9010	002	✓	021114	091915		
01010	Cheese, cheshire Mild **MT4	JG Mingle test		P20001	CHEESE	002	✓	021114	091915		
NF-CW-CON	NF-CW-CONT7A		33	P52320	JUICE	002	✓	011215	122915		

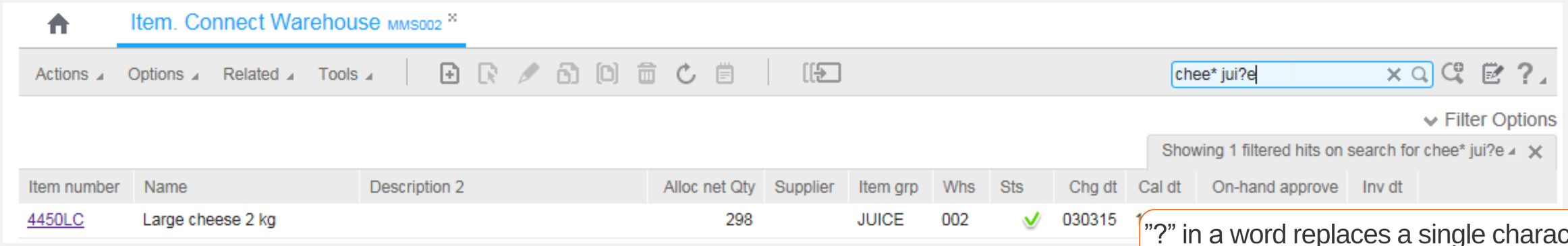
Use ( ) to combine blocks of AND and OR

Simple search

Using Wildcard let you only enter the first part of a word. Risk is of course that you get other hits that you didn't want.



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
01028	Mozarella Cheese	Mozarella Cheese	54	P50001	CHEESE	002	✓	111414	101315	54,00	041515
01032	Parmesan Cheese	Parmesan Cheese 20 months		Y60000	GP9010	002	✓	021114	091915		101813
01010	Cheese, cheshire Mild **MT4	JG Mingle test JG change again **MT		P20001	CHEESE	002	✓	021114	091915		101813
4450LC	Large cheese 2 kg		298		JUICE	002	✓	030315	120315	298	

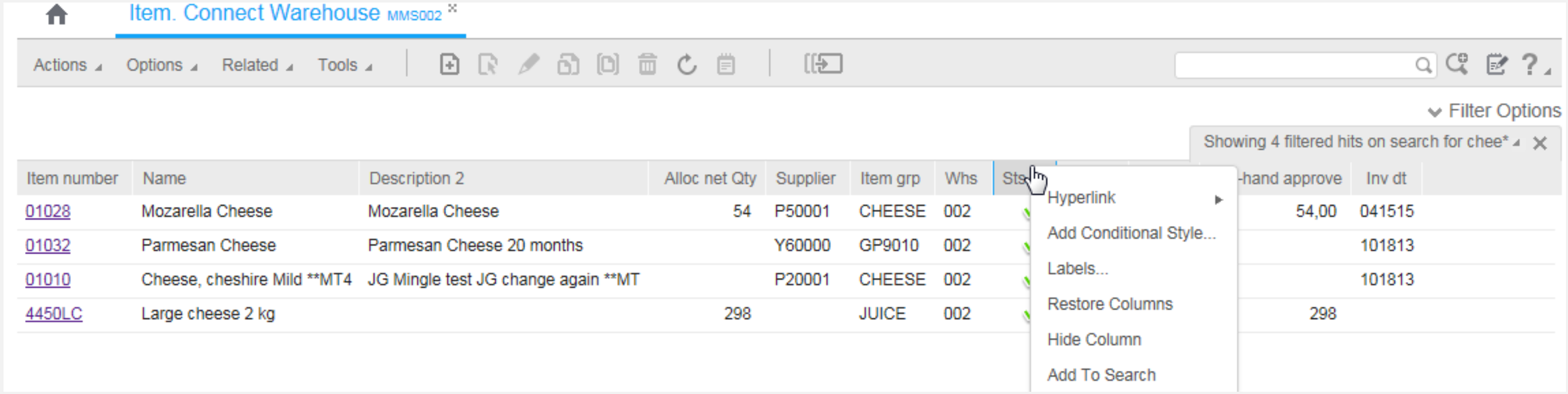


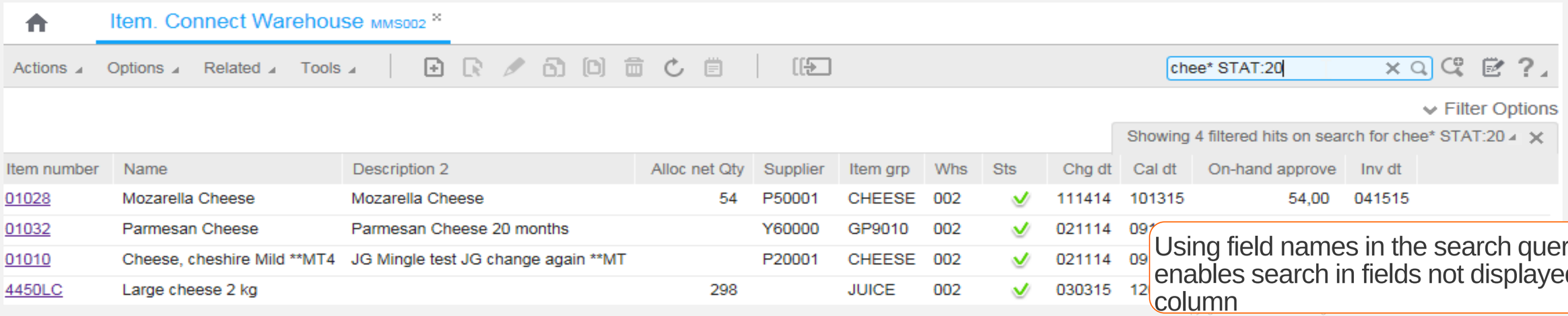
Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
4450LC	Large cheese 2 kg		298		JUICE	002	✓	030315			

"?" in a word replaces a single character (not blank space)

Simple search

To limit search to a single column right click on column header and select ”Add to search” (or write field name manually if known). Write search word without any blank space

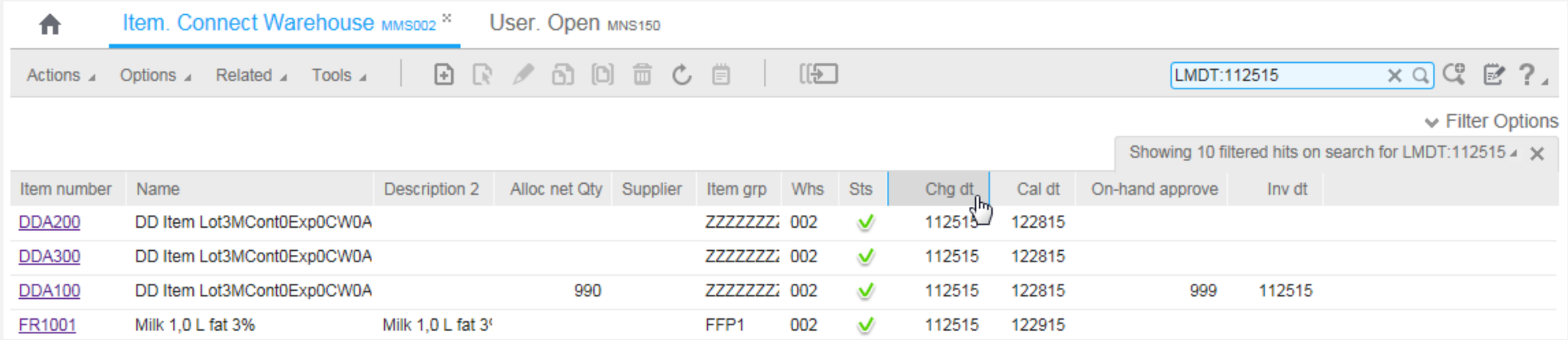




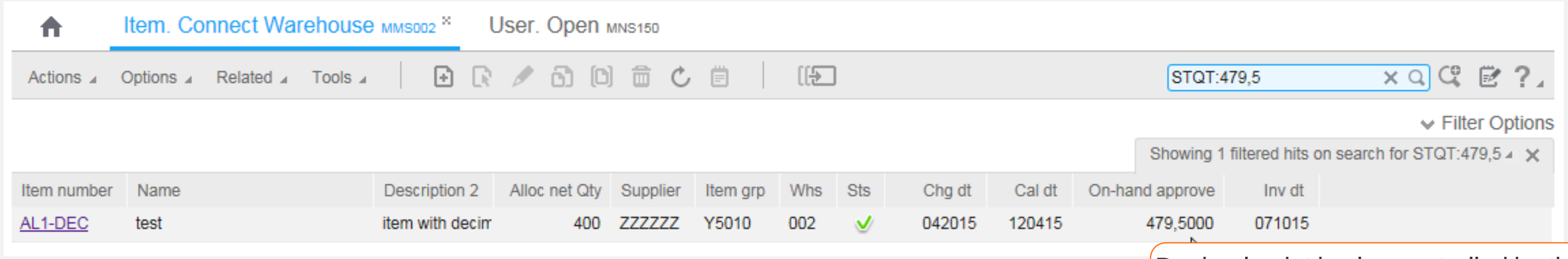
Using field names in the search query enables search in fields not displayed as a column

Simple search

Date is entered in same date format as is set on the user in the user master (MNS150). Normally same as used in the list. Date can be entered with 6 or 8 digits



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
DDA200	DD Item Lot3MCont0Exp0CW0A				ZZZZZZZ	002	✓	112515	122815		
DDA300	DD Item Lot3MCont0Exp0CW0A				ZZZZZZZ	002	✓	112515	122815		
DDA100	DD Item Lot3MCont0Exp0CW0A		990		ZZZZZZZ	002	✓	112515	122815	999	112515
FR1001	Milk 1,0 L fat 3%	Milk 1,0 L fat 3			FFP1	002	✓	112515	122915		



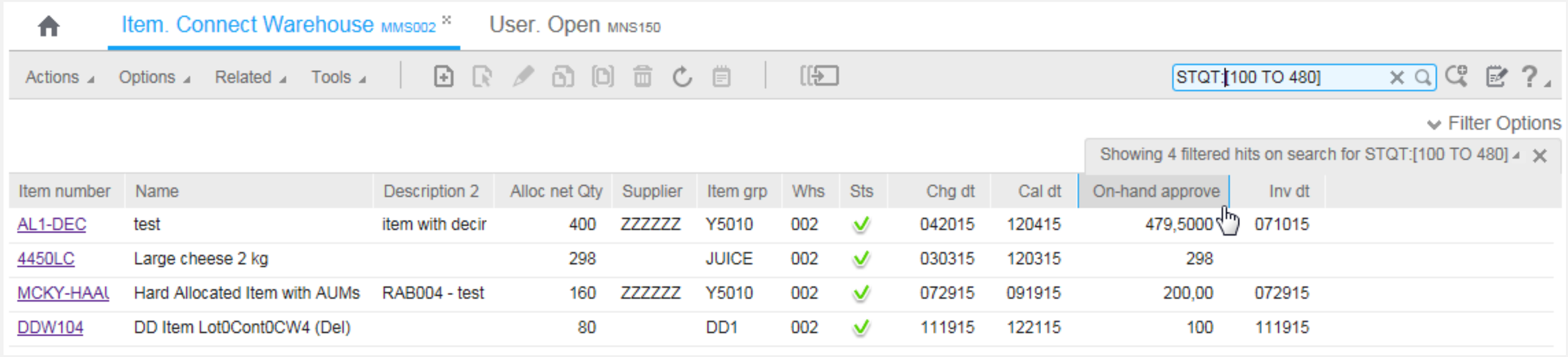
Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
AL1-DEC	test	item with decim	400	ZZZZZZ	Y5010	002	✓	042015	120415	479,5000	071015

Decimal point is also controlled by the setting on the user master

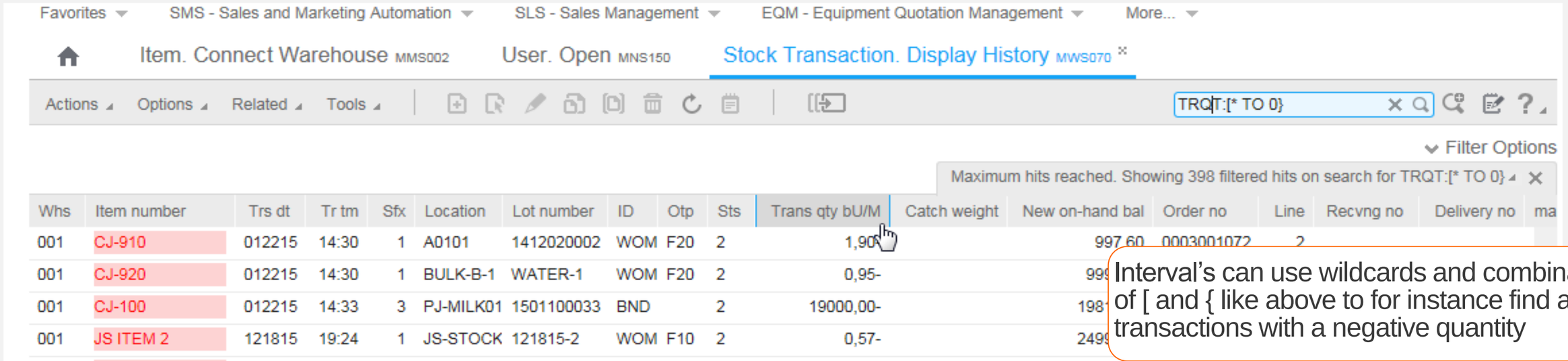
Advanced search

Advanced search

Interval is done with format [x TO y] or { x TO y } or a combination of them. Usage of [ ] means the value is included. Usage of { } means value is excluded



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
AL1-DEC	test	item with decir	400	ZZZZZZ	Y5010	002	✓	042015	120415	479,5000	071015
4450LC	Large cheese 2 kg		298		JUICE	002	✓	030315	120315	298	
MCKY-HAAL	Hard Allocated Item with AUMs	RAB004 - test	160	ZZZZZZ	Y5010	002	✓	072915	091915	200,00	072915
DDW104	DD Item Lot0Cont0CW4 (Del)		80		DD1	002	✓	111915	122115	100	111915

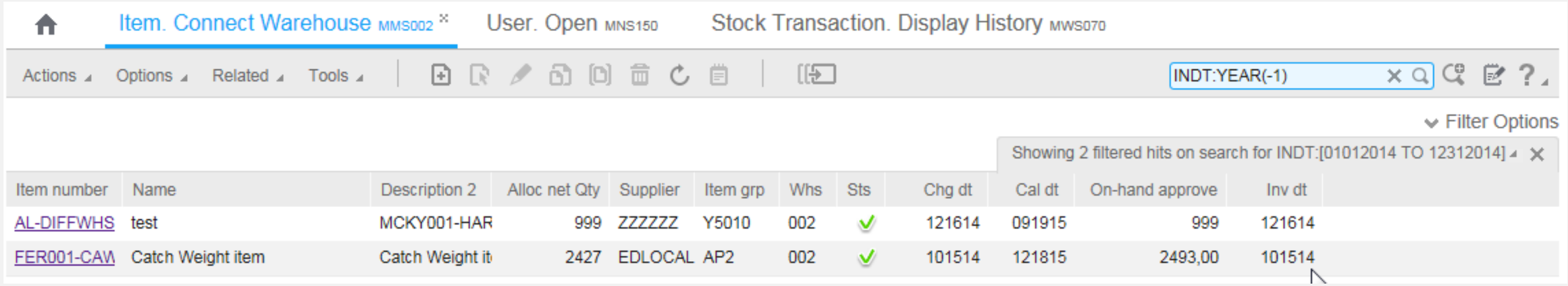


Whs	Item number	Trs dt	Tr tm	Sfx	Location	Lot number	ID	Otp	Sts	Trans qty bU/M	Catch weight	New on-hand bal	Order no	Line	Recvng no	Delivery no	ma
001	CJ-910	012215	14:30	1	A0101	1412020002	WOM	F20	2	1,90		997,60	0003001072	2			
001	CJ-920	012215	14:30	1	BULK-B-1	WATER-1	WOM	F20	2	0,95-		999					
001	CJ-100	012215	14:33	3	PJ-MILK01	1501100033	BND		2	19000,00-		198					
001	JS ITEM 2	121815	19:24	1	JS-STOCK	121815-2	WOM	F10	2	0,57-		249					

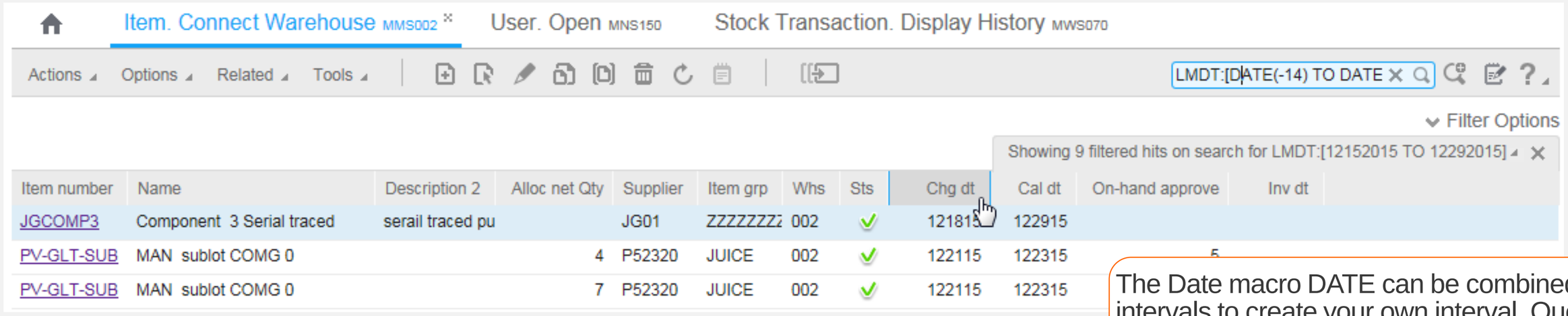
Interval's can use wildcards and combinations of [and { like above to for instance find all transactions with a negative quantity

Advanced search

Date can be entered through date macro's. Following macro's exists: YEAR(x), MONTH(x), WEEK(x) and DATE(x) where x is a relative number meaning DATE(0) = current date



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
AL-DIFFWHS	test	MCKY001-HAR	999	ZZZZZZ	Y5010	002	✓	121614	091915	999	121614
FER001-CAW	Catch Weight item	Catch Weight it	2427	EDLOCAL	AP2	002	✓	101514	121815	2493,00	101514

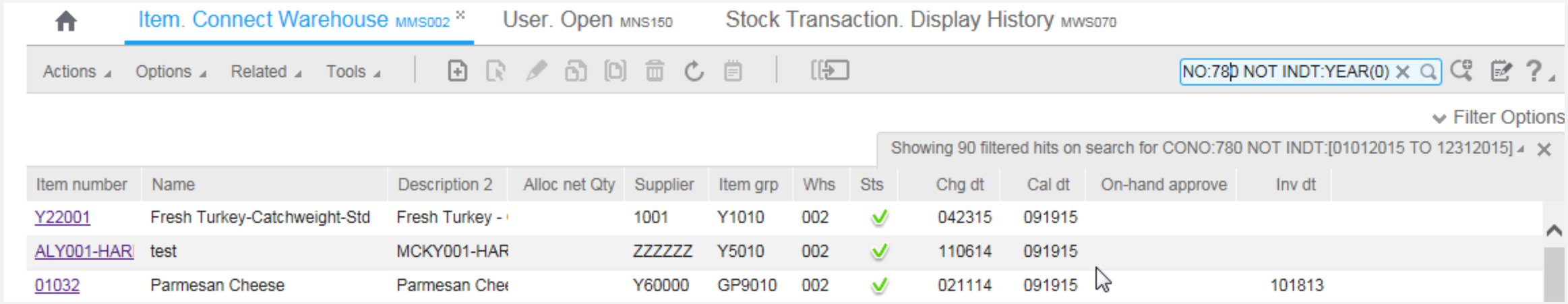


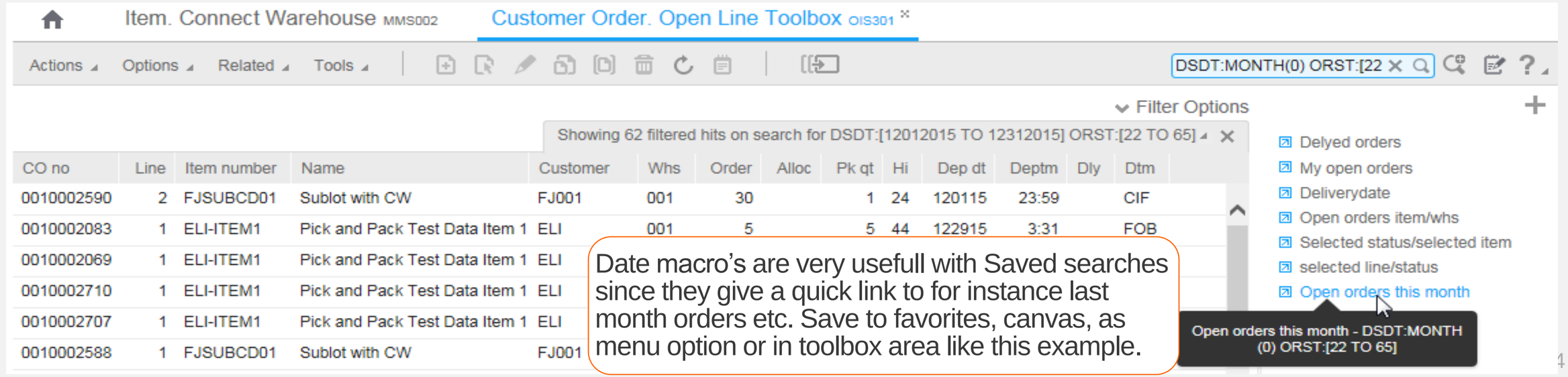
Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
JGCOMP3	Component 3 Serial traced	serail traced pu		JG01	ZZZZZZZ	002	✓	121815	122915		
PV-GLT-SUB	MAN sublot COMG 0		4	P52320	JUICE	002	✓	122115	122315		
PV-GLT-SUB	MAN sublot COMG 0		7	P52320	JUICE	002	✓	122115	122315		

The Date macro DATE can be combined with intervals to create your own interval. Query LMDT:[DATE(-14) TO DATE(0)] results in last two weeks

Advanced search

Searching with NOT must start with other query. Hint is to use company number as search if the whole table should be searched. Here are all items not been part of a stock take 2015 displayed





Date macro's are very usefull with Saved searches since they give a quick link to for instance last month orders etc. Save to favorites, canvas, as menu option or in toolbox area like this example.

Advanced search

Creative usage of the ? allows a simple search for all records with supplier number. This since ? only finds records with a character in the positon. Blanks are omitted. Query SUNO:?* gives you all records with a supplier number in this case. For numeric fields you use the NOT statement like NOT STQT :0

Item. Connect Warehouse MMS002 | Customer Order. Open Line Toolbox OIS301

Actions | Options | Related | Tools | SUNO:?*

Filter Options

Showing 86 filtered hits on search for SUNO:?*

Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
01032	Parmesan Cheese	Parmesan Cheese 20 months		Y60000	GP9010	002	✓	021114	091915		101813
JGCOMP3	Component 3 Serial traced	serail traced purchased item		JG01	ZZZZZZZZ	002	✓	121815	123015		
01028	Mozarella Cheese	Mozarella Cheese	54	P50001	CHEESE	002	✓	111414	101315	54,00	041515
02027	Oregano	Oregano		P50012	CHEMICAL	002	✓	021114	091915		101913

Item. Connect Warehouse MMS002 | Customer Order. Open Line Toolbox OIS301

Actions | Options | Related | Tools | min*~

Filter Options

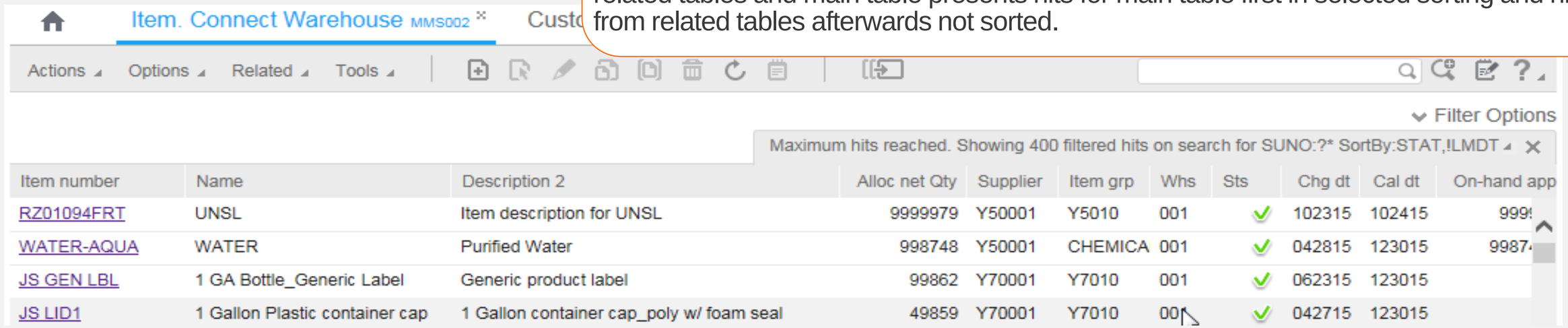
Showing 3 filtered hits on search for min*~

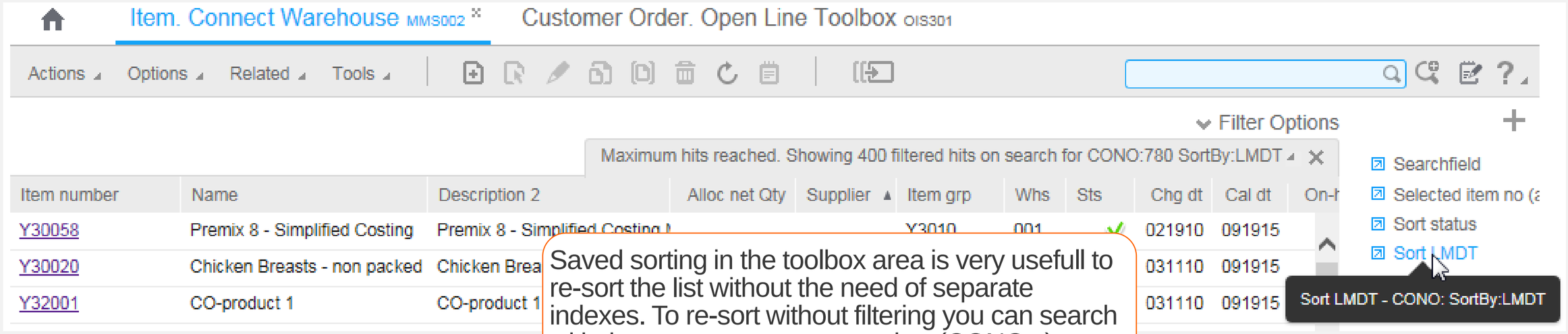
Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
807001	Minced meat (BIN)	Minced meat (BIN)			AP2	002					
807501	Minced meat (consumer)	Minced meat (consumer)			AP2	002					
01010	Cheese, cheshire Mild **MT4	JG Mingle test JG change again **MT		P20001	CHEESE	002					

The ~ (like query min*~ above) introduces a fuzzy search which returns similar hits like mild in this example. Usefull to manage spelling errors etc. Note! Never use it for numeric values.

Advanced search

Sorting of the search result is a very useful feature since it allows a free sorting of any field in the main table of the function (exception with fields containing mulitple words like a description). The sorting is done by key word SortBy:x,z,!y where ! perform a decending sorting. Please note that sorting is only done on searches on main table. A search finding hits in a related table (setup in CMS022) are not sorted. A Search finding hits in both related tables and main table presents hits for main table first in selected sorting and hits from related tables afterwards not sorted.

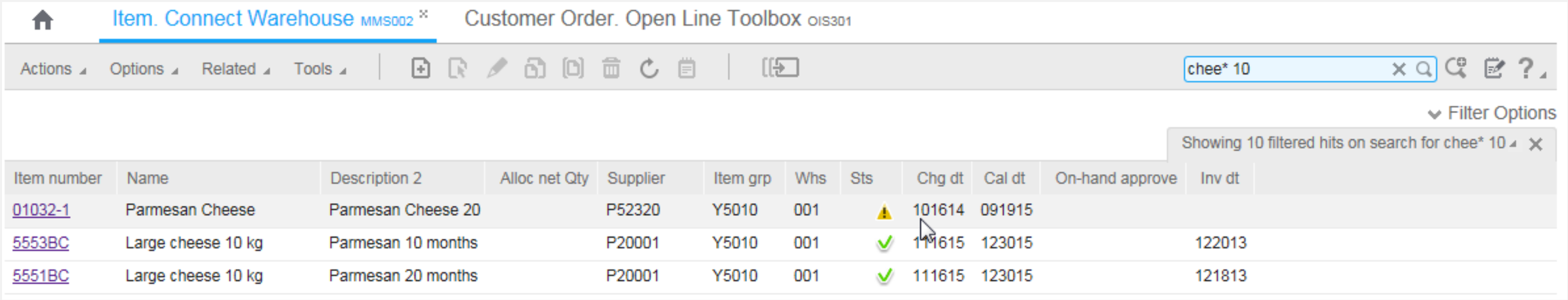




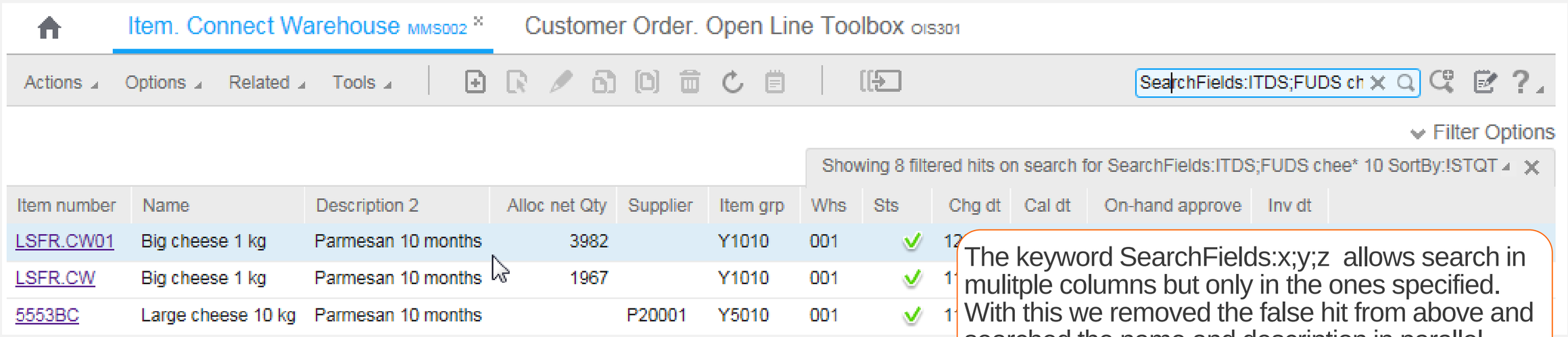
Saved sorting in the toolbox area is very usefull to re-sort the list without the need of separate indexes. To re-sort without filtering you can search with the current company number (CONO:x)

Advanced search

Searching with multiple words existing in several columns can easily give false hits (like first line below where it found it becase the status = 10). For this you can't use specific fields either since you do not know in which column the word exist.



Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
01032-1	Parmesan Cheese	Parmesan Cheese 20		P52320	Y5010	001	⚠	101614	091915		
5553BC	Large cheese 10 kg	Parmesan 10 months		P20001	Y5010	001	✓	111615	123015		122013
5551BC	Large cheese 10 kg	Parmesan 20 months		P20001	Y5010	001	✓	111615	123015		121813

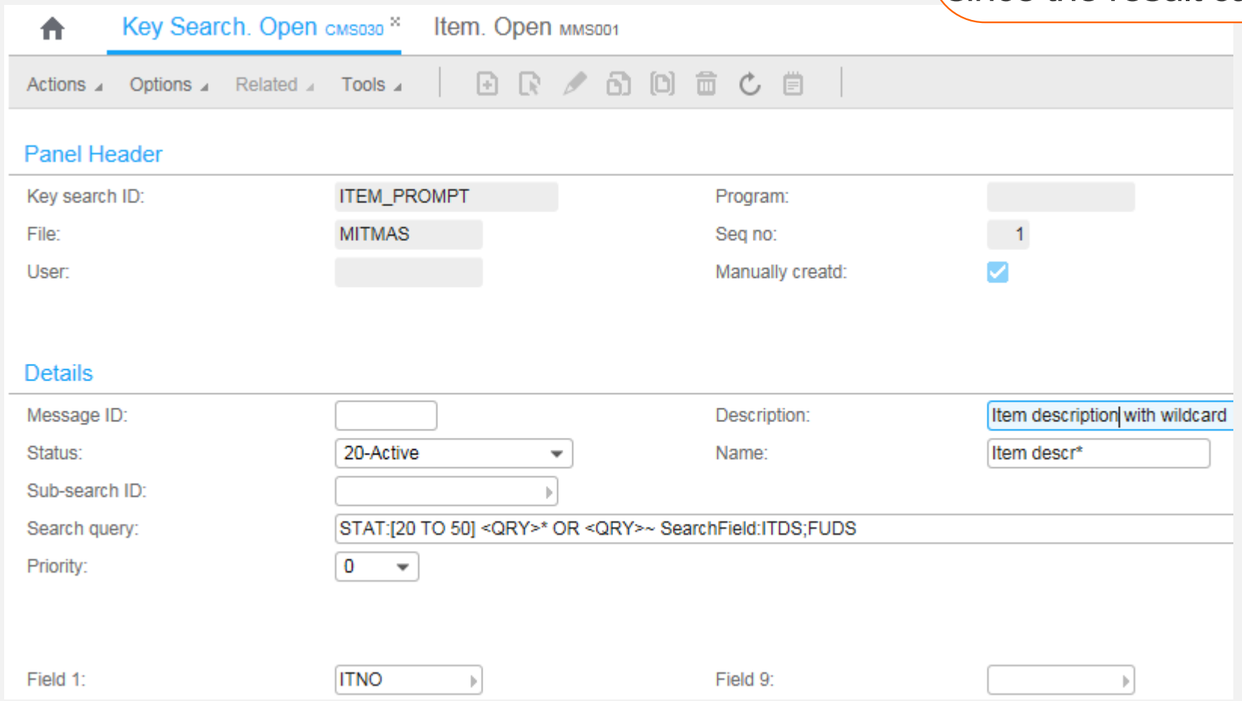


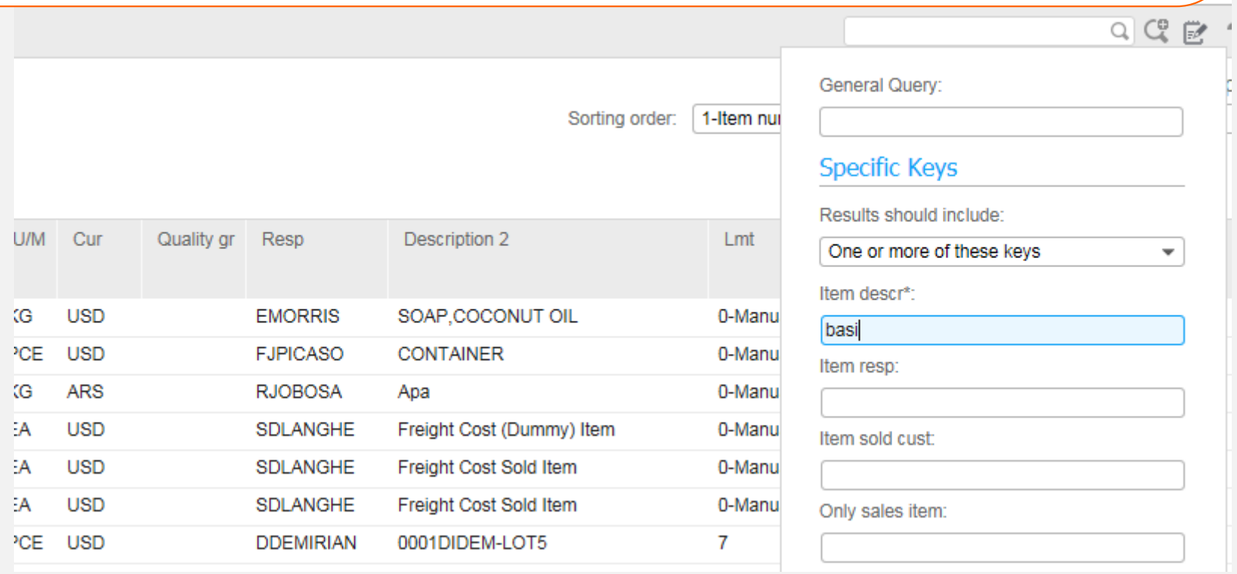
Item number	Name	Description 2	Alloc net Qty	Supplier	Item grp	Whs	Sts	Chg dt	Cal dt	On-hand approve	Inv dt
LSFR.CW01	Big cheese 1 kg	Parmesan 10 months	3982		Y1010	001	✓	12			
LSFR.CW	Big cheese 1 kg	Parmesan 10 months	1967		Y1010	001	✓	1			
5553BC	Large cheese 10 kg	Parmesan 10 months		P20001	Y5010	001	✓	1			

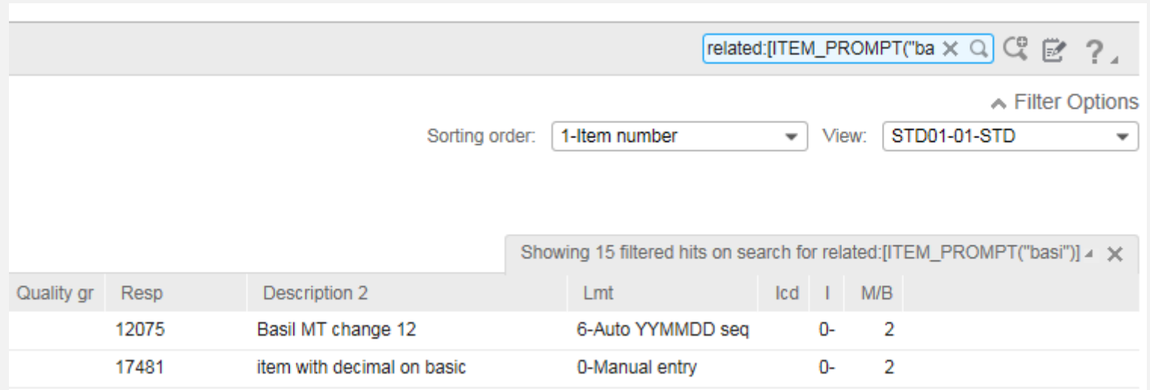
The keyword SearchFields:x;y;z allows search in mulitple columns but only in the ones specified. With this we removed the false hit from above and searched the name and description in parallel.

Advanced search

Searching with help of key search simplifies often more complex queries for the end user. It also allows search in other tables without the need for matching keys (as related search setup in CMS022 requires). In the below example the end user just enters the query and the pre-set part in the key search is automatically appended. Availiable Key search is displayed by Ctrl+f. Please note that a key search is performed on both main table and related tables (in CMS022). It is not recommended to use related tables and key search since the result can be confusing



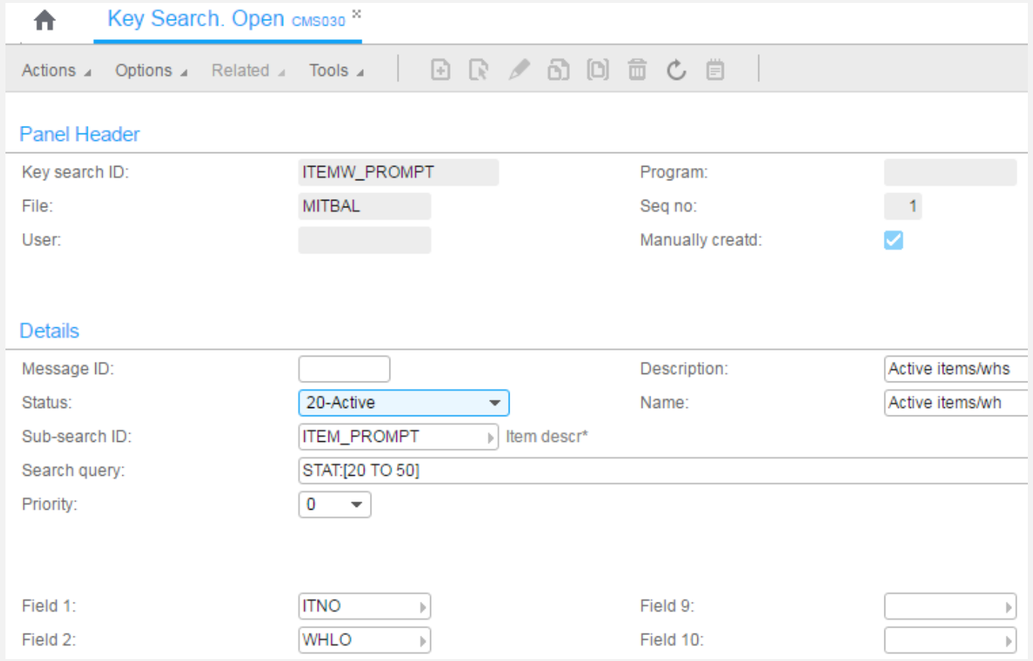




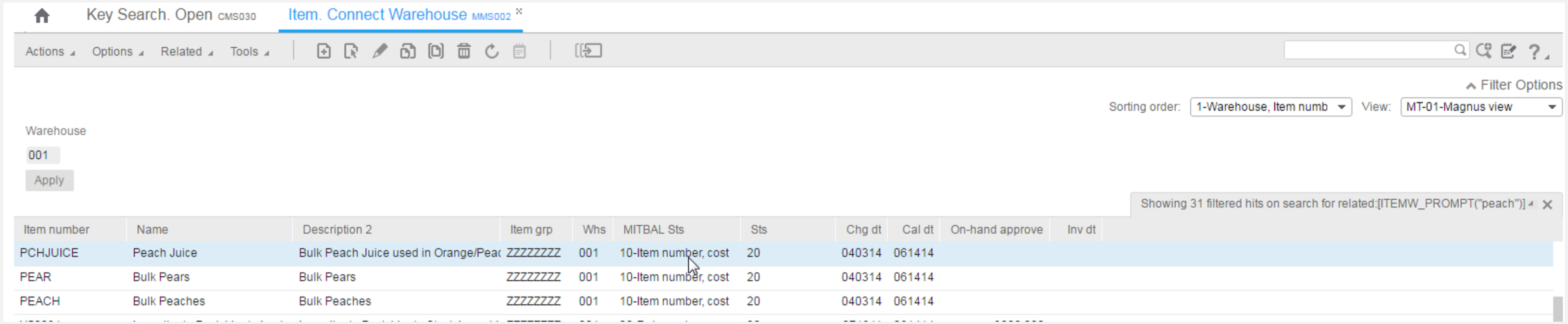
Key search can be used in many scenarios. In above case a key search is also used to search for responsible by name (searches User master), items sold to a specific customer (searches customer order lines as well as an automatic limit of search to only include items set as sales items.

Advanced search

Please note that a key search is performed on both main table and related tables (defined in CMS022 or related tables through CMS011 when using custom list, CMS100). It is not recommended to use related tables and key search in combination since the result can be confusing



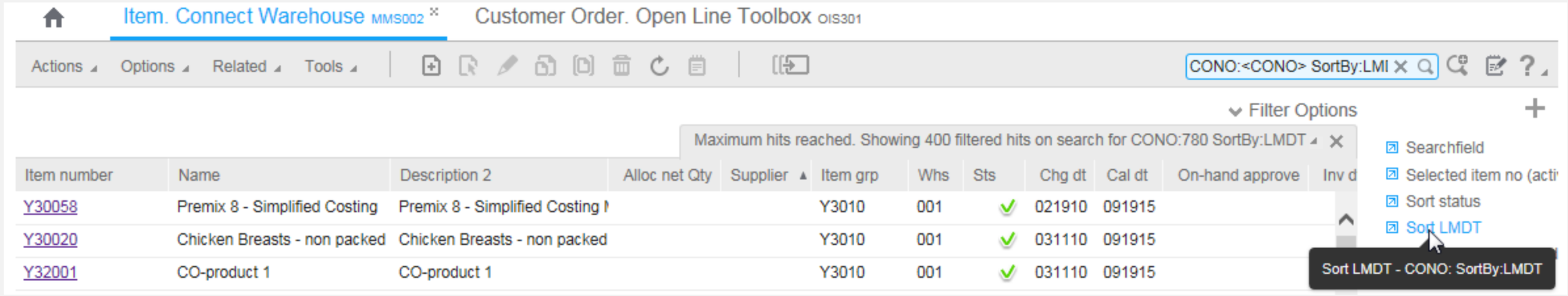
In the below example the key search is used including the status validation of MITBAL records. Still records with MITBAL status = 10 are displayed. The reason for this is that the key search is also performed on related table MITMAS and that search only returns item number. This means that if there is any MITBAL record for the item with status 20 to 50 then it is a valid hit. When finally displayed in MMS002 it will display the MITBAL record for the selected warehouse which happens to have status 10. Basically ending with the same result as if the key search only returned field Item number.

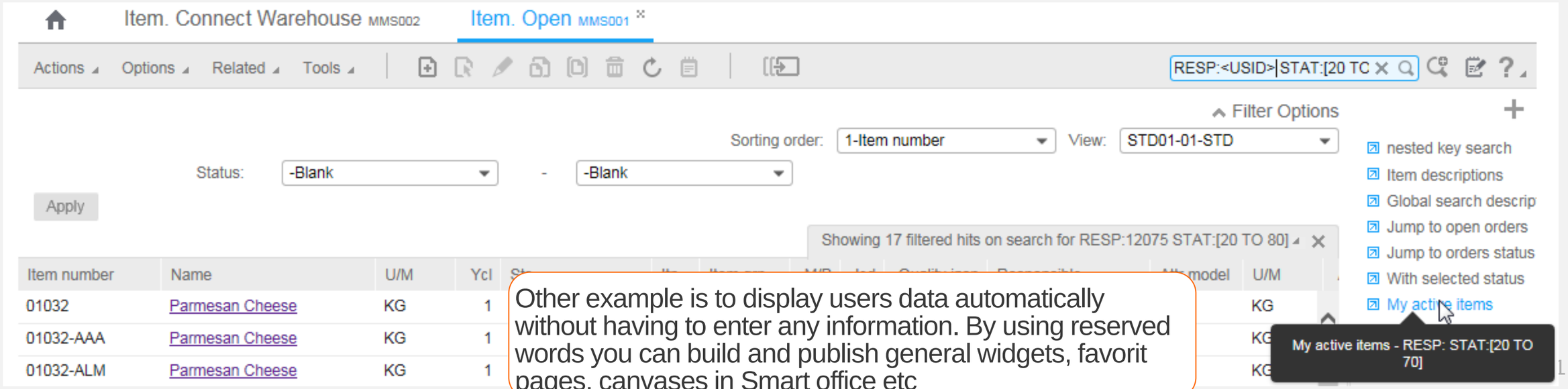


Search including reserved words

Advanced search

M3 IES reserved words are very useful in stored and published search queries since they automatically retrieve information from the context they belong to. In example below company number is retrived from current session making the query independent of user and context.

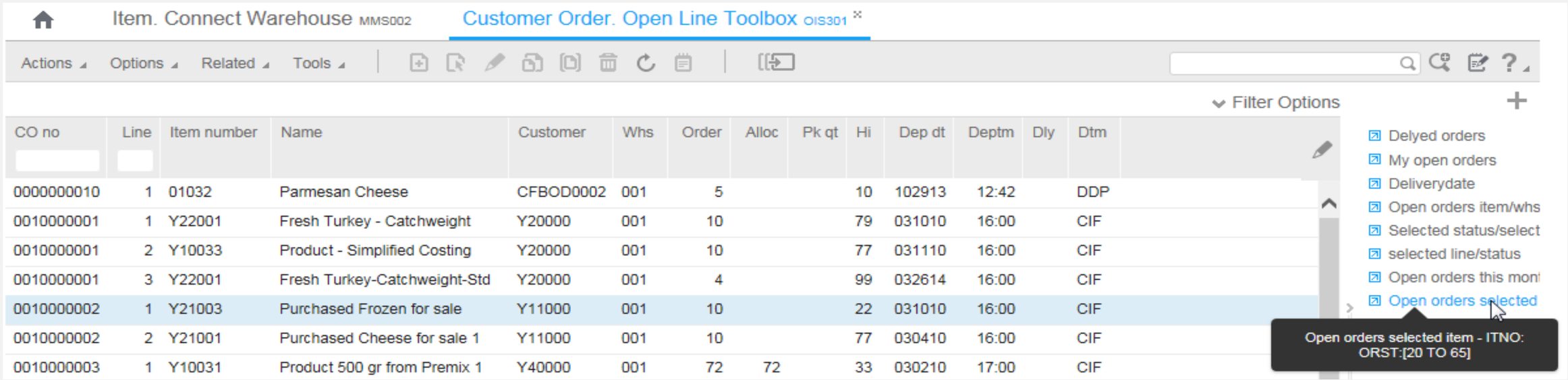




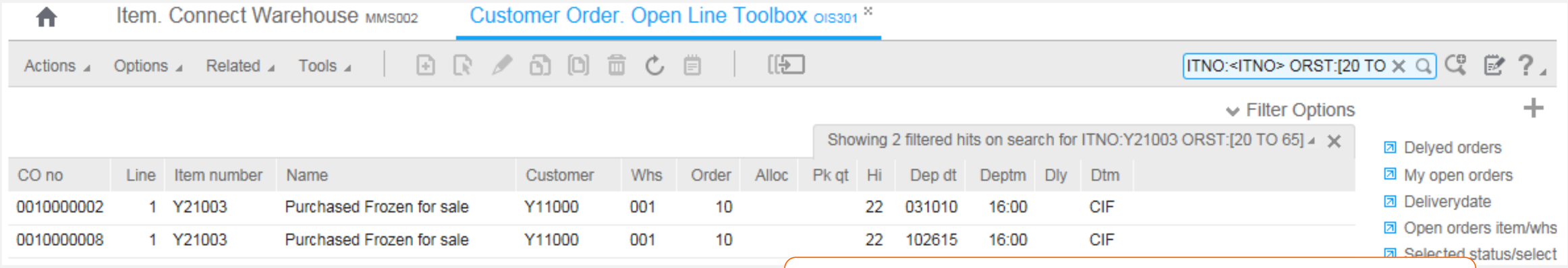
Other example is to display users data automatically without having to enter any information. By using reserved words you can build and publish general widgets, favorit pages, canvases in Smart office etc

Advanced search

Reserved words can also replace information in query based on selected line in a list or from panel header. Below selected item number is added to query to create a simple drill around among the data.



CO no	Line	Item number	Name	Customer	Whs	Order	Alloc	Pk qt	Hi	Dep dt	Deptm	Dly	Dtm
0000000010	1	01032	Parmesan Cheese	CFBOD0002	001	5			10	102913	12:42		DDP
0010000001	1	Y22001	Fresh Turkey - Catchweight	Y20000	001	10			79	031010	16:00		CIF
0010000001	2	Y10033	Product - Simplified Costing	Y20000	001	10			77	031110	16:00		CIF
0010000001	3	Y22001	Fresh Turkey-Catchweight-Std	Y20000	001	4			99	032614	16:00		CIF
0010000002	1	Y21003	Purchased Frozen for sale	Y11000	001	10			22	031010	16:00		CIF
0010000002	2	Y21001	Purchased Cheese for sale 1	Y11000	001	10			77	030410	16:00		CIF
0010000003	1	Y10031	Product 500 gr from Premix 1	Y40000	001	72	72		33	030210	17:00		CIF

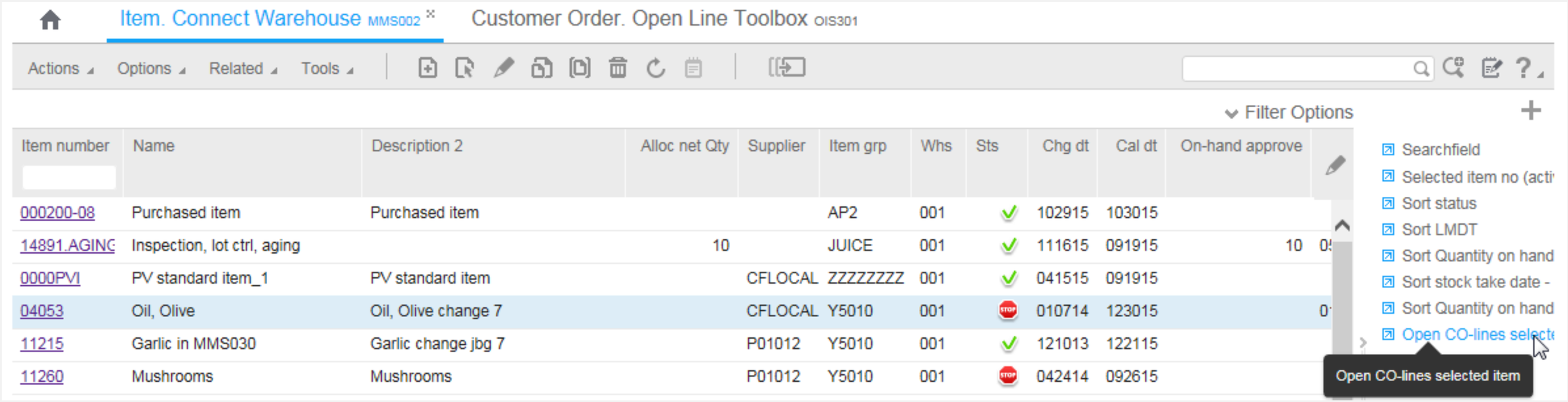


CO no	Line	Item number	Name	Customer	Whs	Order	Alloc	Pk qt	Hi	Dep dt	Deptm	Dly	Dtm
0010000002	1	Y21003	Purchased Frozen for sale	Y11000	001	10			22	031010	16:00		CIF
0010000008	1	Y21003	Purchased Frozen for sale	Y11000	001	10			22	102615	16:00		CIF

The result is a list with all open orders for the selected item

Advanced search

Saved search can also be used to add links to other programs where the search query is passed to the next program. In below example we jump to Customer order lines (OIS301) and included the item number from the selected line



M3 IES reserved words

Reserved words

- SortBy:x,y,!z — Sorts search result in x, y z hierarchy. The ! Prior field name perform a descending sorting (NCR 6670 and 6671)
- SearchFields:x;y — Limits search to listed fields x, y etc. Increases performance and can eliminate false hits
- YEAR(x) — Replaces with date range of year relative to current year x=-1 equals last year etc
- MONTH(x) — Same as YEAR but for month
- WEEK(x) — Same as YEAR but for week
- DATE(x) — Similar to YEAR but replaces with specific date
- <xxxx> — List & list header based replacements to query string. (NCR 6813)

All below are described in NCR 6220

- <USID> — Replaced by logged on user id
- <CONO> — Replaced by logged on company number
- <DIVI> — Replaced by logged on Division. Same as <CurrentDivison>
- <FACI> — Replaced by default facility from user master (MNS150)
- <CUNO> — Replaced by customer no from user master
- <LNCD> — Replaced by language code from user master
- <CurrentLanguage> — Replaced by currently used language in M3 (in case of switched language during logged in session)
- <TIZO> — Replaced by time zone from user master
- <WHLO> — Replaced by warehouse from user master
- <DEPT> — Replaced by department from user master
- <MNVR> — Replaced by menu version from user master
- <DFMN> — Replaced by menu name from user master

M3 IES reserved words

Reserved words for usage with key search only:

- <QRY> Replaced by entered query in the search string
- MAXJOIN:x Used to change number of records used in the key search join (default set in IES local management pages) (NCR 4351)
- NOOPT Used when multiple key searches done to same table (like find items with attribute x, y but NOT z or items existing in warehouse a, b and c (NCR 4349)

M3 IES stop words

Stop words in IES has the meaning of words that are not indexed and are automatically removed from the search query. Examples of stop words are “and”, “or” “etc” and other similar types. IES, by default has a set of English based stop words which apart from above examples also includes “a”, “by”, “is”, “as” plus about 20 other words. Unfortunately in many cases when searching ERP type of data you might want to search for “factor a”, the Swedish word for ice “is” etc.

The work around for this is to add a new stop word in the IES admin pages. Make sure it is something never search for like “lksdhja7789”. By doing that the default stop words are not used anymore. Please note that affected tables like MITMAS, MITLAD, OCUSMA, CIDMAS etc need to be re-indexed after this has been done.

M3 IES search query and key search

The following rules are used to set * and ~ in a key search:

- <QRY>* — Automatically appends a wild card to each word in the user entered query. As an example the query joh smit is set to joh* smit* before search is performed. Please see exceptions below.
- <QRY>~ — Automatically appends a tilde (fuzzy search) to each word in the user entered query. As an example the query joh smit is set to joh~ smit~ before search is performed. Please see exceptions below. NOTE it is not recommended to use fuzzy search for numeric values. This is due to the unpredictable result of this type of search.

Exceptions

- Query inside “” — When entering a query enclosed by “” like “joh smit” wildcards or tilde is not added to the words. The main reason for this is that the search engine treats all characters inside “” as normal characters meaning “joh*” only returns result named joh* and not john etc.
- Short words — From M3 IES v11 patch 5. Short words containing less than the number of characters set in the IES Local admin page (Default is 3 characters meaning jo, 10, xl) are not appended with * or ~ . The reason for this is the very heavy load short words with wild cards places on the IES server. Very often such a search also ends in an error message stating “To many possible hits”. In short it means that a query like joh smit 10 ends up like joh* smit* 10

M3 IES usage through API calls

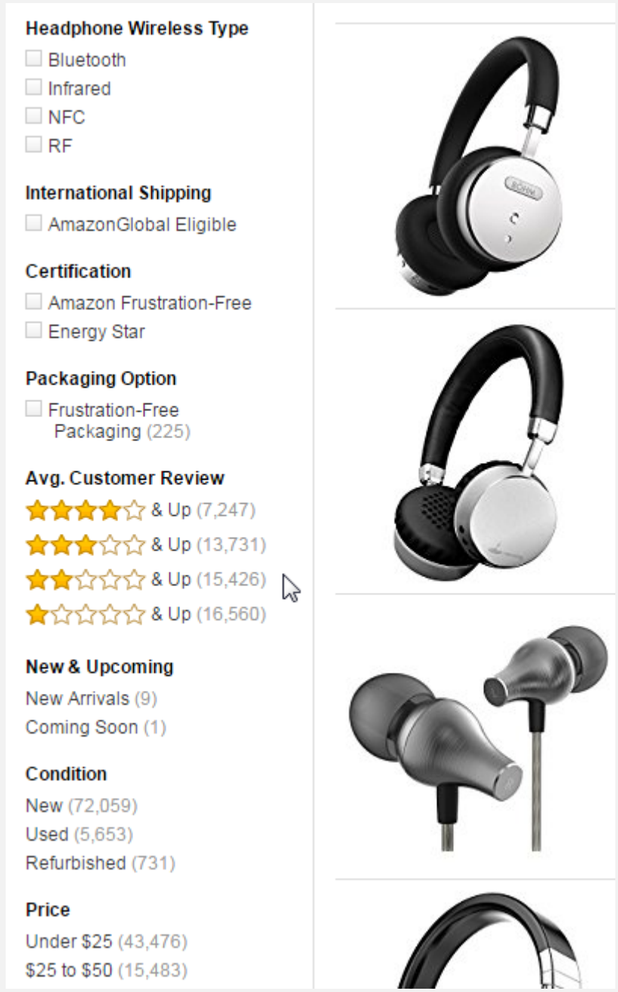
M3 IES usage through API calls

M3 API fully supports the usage of IES. There are some standard API transactions that are search enabled (like MMS200MI Trans SearchItem) where the normal filter input is replaced by a search query. The search through API supports all normal search queries except the UI related ones like Date macros and field value replacements (like <WHLO>).

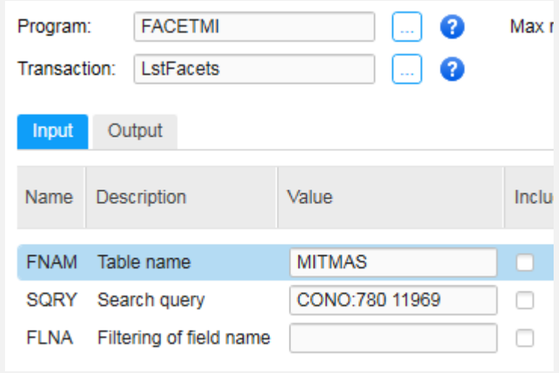
The M3 API tools (MDBREADMI and CMS100MI) are fully search enabled.

Please note that there is no fixed upper limit of number records possible to return through an search in an API. However there is an practical limit due to that the API fails due to a timeout. The number of successful returned records due to this is highly dependable by the environment. Tests have pointed to a reasonable maximum around 3000-6000 records.

M3 IES usage through API calls - Facets



A special part of the IES through API calls are the facet searches. In the most simplistic form facets can be explained as a count of the number of records tht matches a certain value in a specific field. As a start, facets where used to simplify the filtering of information by aggregating number of existing records into various groups. M3 Sales hub as well as many web shops etc uses this technique. To the left is a screenshot from Amazon where examples of facets are displayed.



The M3 facets are retrieved by using the FACETMI. The input to the API is the table name for which the facets are retrieved and secondly the search query for filtering the facets and their count. The above gives this result. As example based on the query there are 161 records with 0 as purchase price and 1 with sales price = 20.

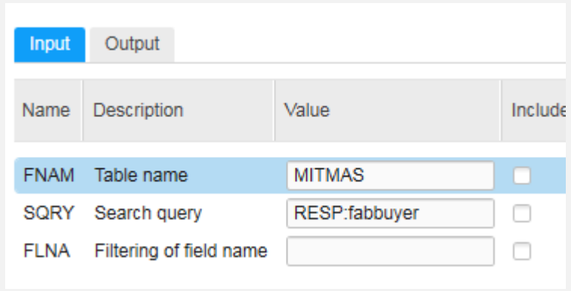
Program: FACETMI Max records:
Transaction: LstFacets
Input Output

Facet name	Language constant	Field name	Facet value	Facet count
Sales price	WSA02AA	SAPR	20	1
Sales price	WSA02AA	SAPR	40	1
Sales price	WSA02AA	SAPR	45	1
Responsible	WRE01AA	RESP	11969	472
Responsible	WRE01AA	RESP	FABBUYER	36
Responsible	WRE01AA	RESP	FABPRODUCT	18
Responsible	WRE01AA	RESP	FABPLANNER	9
Responsible	WRE01AA	RESP	DSUBRAMA	8
Responsible	WRE01AA	RESP	10546	2
Responsible	WRE01AA	RESP	SRAJARAO	1
Company	WCO01AA	CONO	780	546
Purchase price	WPU01AA	PUPR	0	175
Purchase price	WPU01AA	PUPR	0.0	161

M3 IES usage through API calls - Facets

The strength with facets is that is gives very high performance regardless of the number of records in the table and secondly that the filter of the facets and their count is based on a search query. This gives endless possibility to drill around in the information.

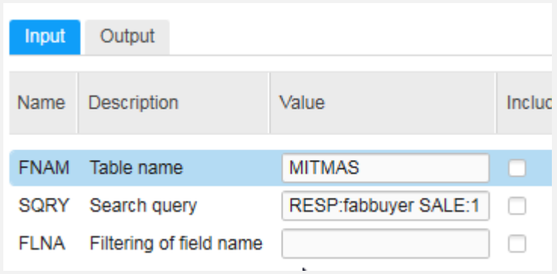
As a limitation it can only do a count of the number of records.



As an example start with investigating some statistics for the responsible user ”fabbuyer”

Input | Output

Facet name	Language constant	Field name	Facet value	Facet count
Item group	WIT02AA	ITGR	POTAT	1
Status	WST03AA	STAT	20	107
Status	WST03AA	STAT	05	6
Status	WST03AA	STAT	10	3
Item category	WTPCDAA	TPCD	0	116
Quality group	WQMGPAA	QMGP	POTATO	1
Quality group	WQMGPAA	QMGP	PULP_JUICE	1
Quality group	WQMGPAA	QMGP	SG-Q1	1
Product group	WIT03AA	ITCL	ZZZZZ	115
Product group	WIT03AA	ITCL	SB01	1
Sales item	WSA31AA	SALE	0	103
Sales item	WSA31AA	SALE	1	13
Supplier	WSU01AA	SUNO	Y50001	30



Further refining the query to only see facets for sales items but still for same responsible person

Input | Output

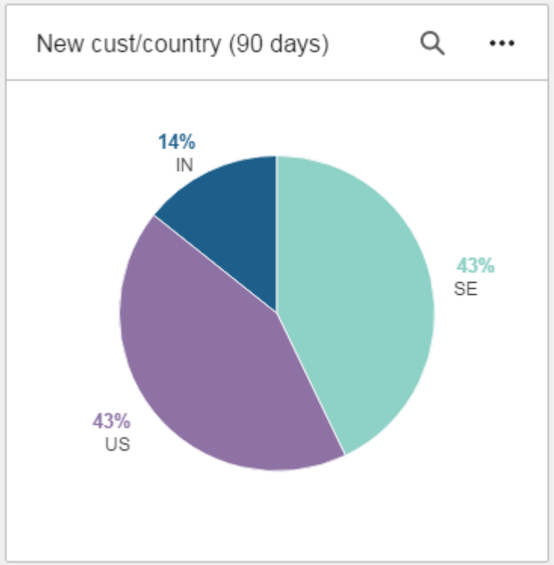
Facet name	Language constant	Field name	Facet value	Facet count
Purchase pric	WPU01AA	PUPR	9.0	1
Purchase pric	WPU01AA	PUPR	15.0	1
Purchase pric	WPU01AA	PUPR	20.0	1
Item group	WIT02AA	ITGR	Y5010	9
Item group	WIT02AA	ITGR	ZZZZZZZZ	3
Item group	WIT02AA	ITGR	Y7010	1
Status	WST03AA	STAT	20	13
Item category	WTPCDAA	TPCD	0	13
Quality group	WQMGPAA	QMGP	SG-Q1	1
Product group	WIT03AA	ITCL	ZZZZZ	12
Product group	WIT03AA	ITCL	SB01	1
Sales item	WSA31AA	SALE	1	13
Supplier	WSU01AA	SUNO	Y50001	2

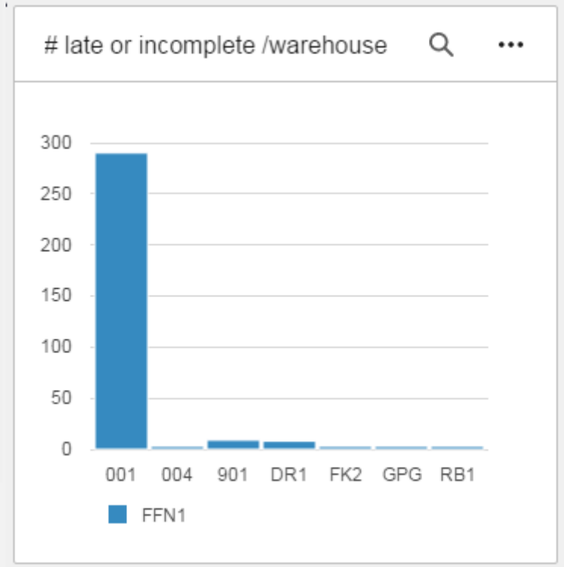
M3 IES usage through API calls - Facets

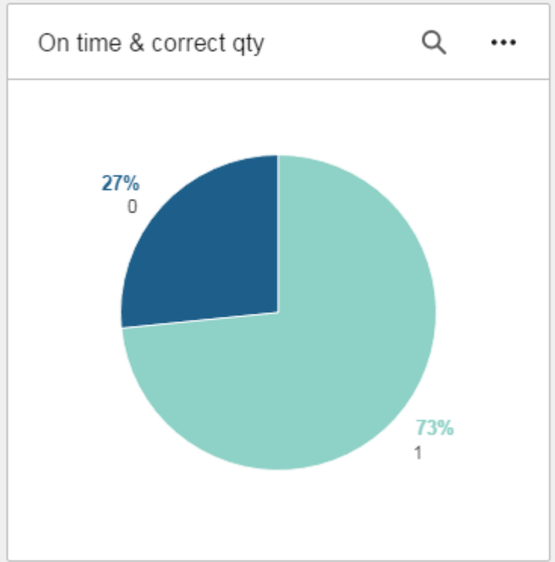
The facets in M3 can facilitate many statistical measurements like the below examples taken from the ”M3 Sales Manager homepage template”

Apart from using facets as a selection criteria is can also be used as an on-line statistics.

To facilitate this facets has been added, by default, to many M3 tables including the transactional and statistical tables.



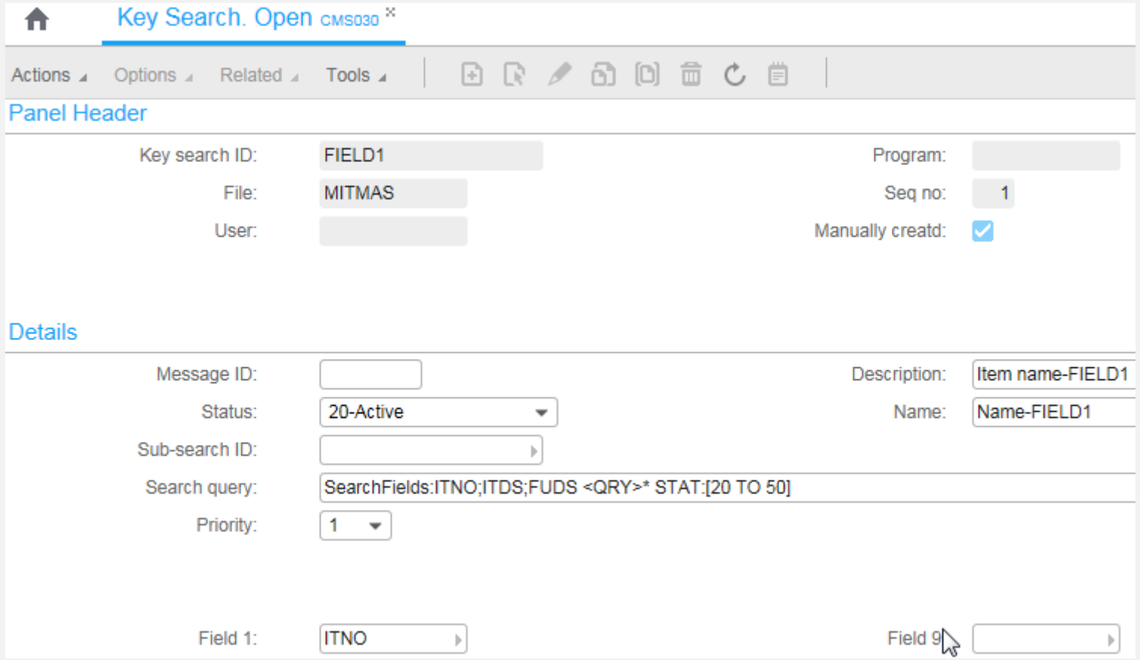


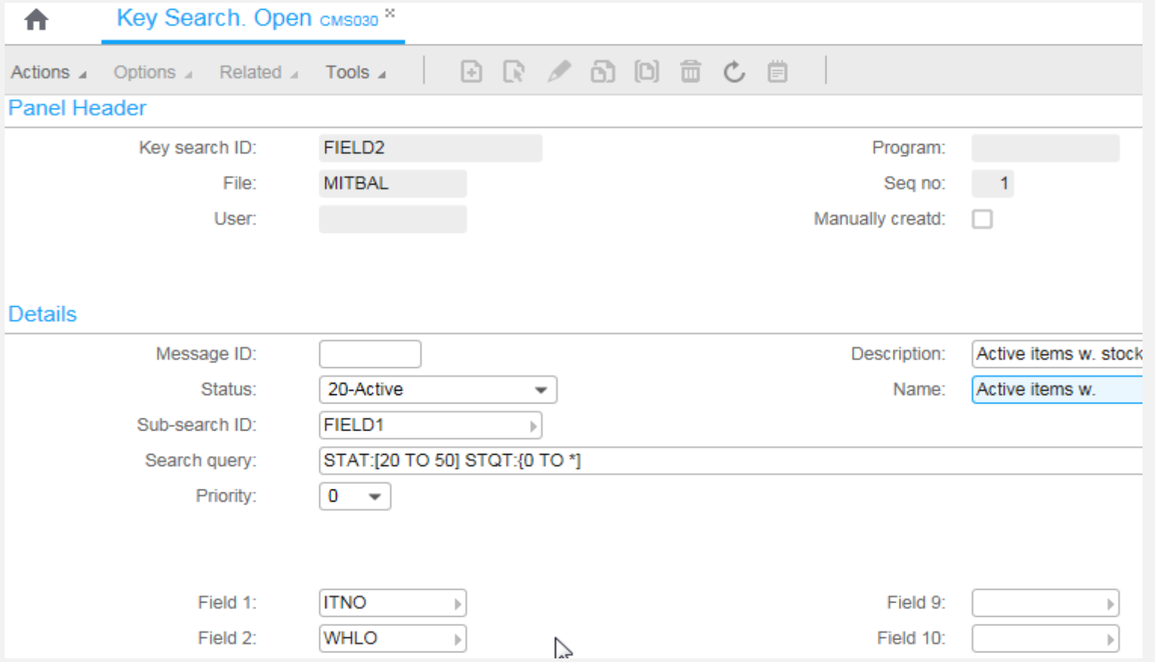


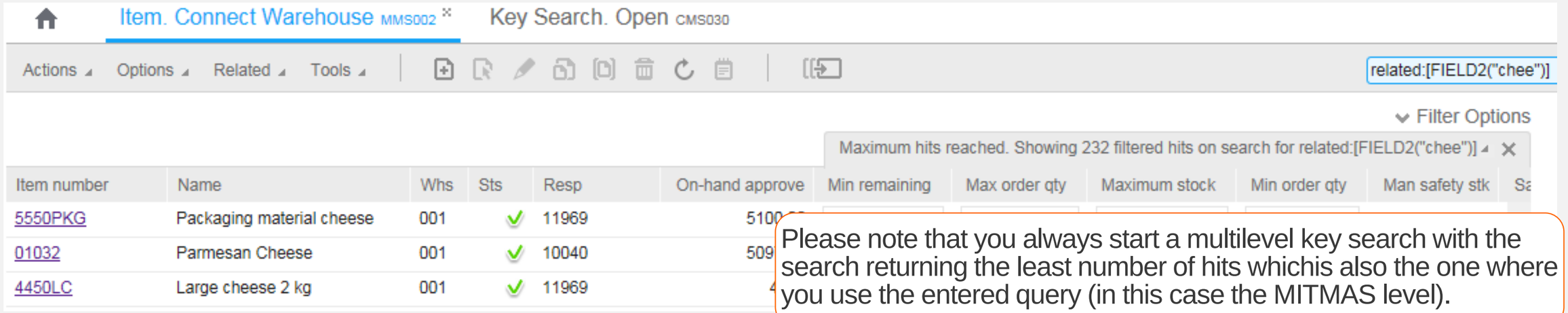
Multi level Key search

Multi level Key search

Multi level key search is a very useful feature and in this example it is used to find items from the name, description or item number but automatically limits it to items with stock.



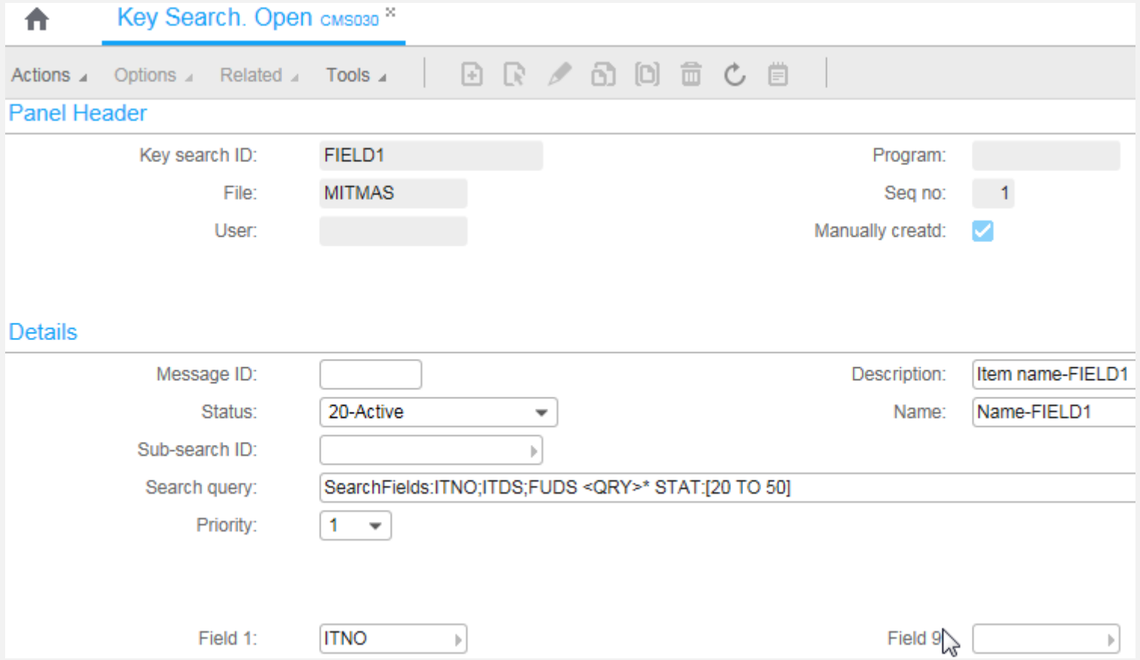


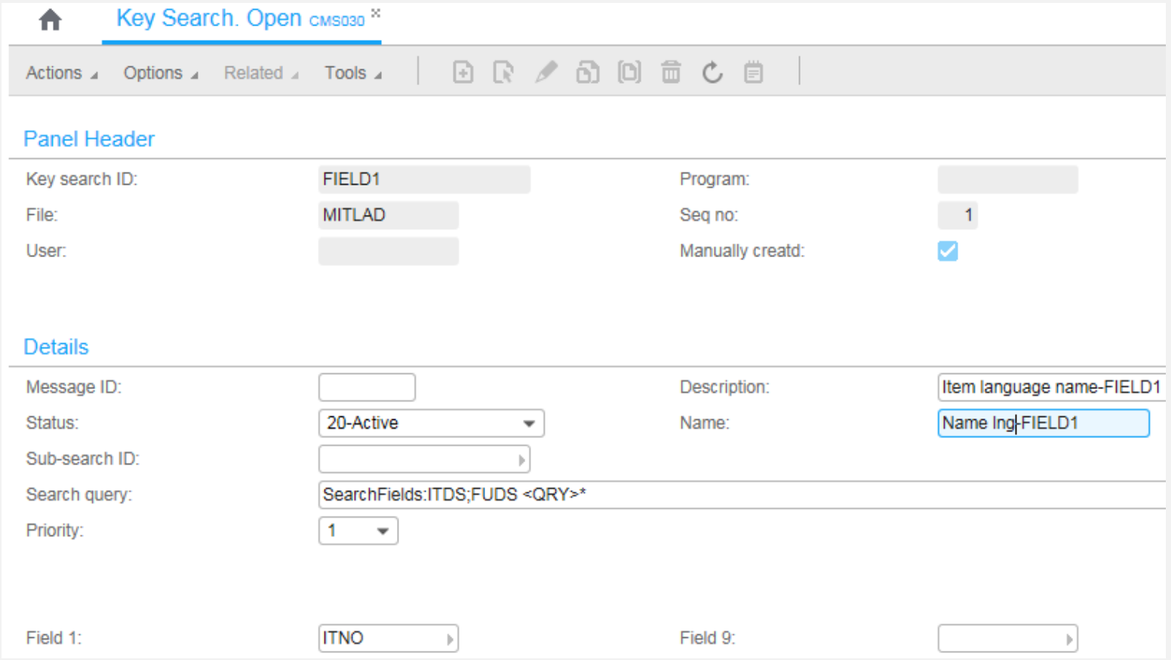


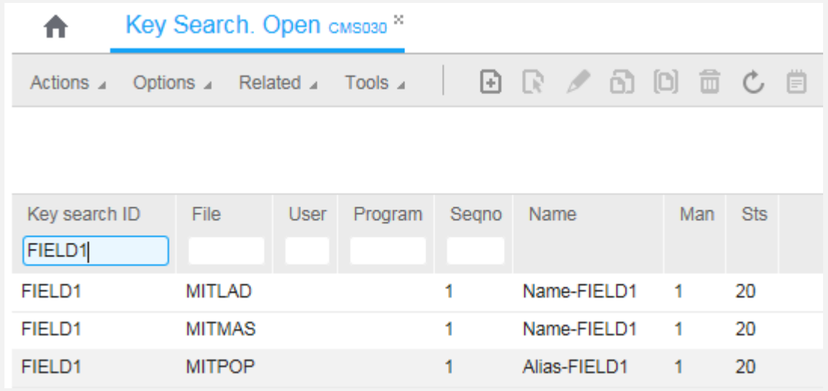
Please note that you always start a multilevel key search with the search returning the least number of hits whichis also the one where you use the entered query (in this case the MITMAS level).

Multi level Key search

Key search can expand the search to multiple tables to be searched in parallel. To add for instance MITLAD (Item language description) to the search in previous example is done by adding a keysearch with same ID but for MITLAD (see below). The search is then done in both MITMAS and MITLAD in parallel and results from both key searches are used





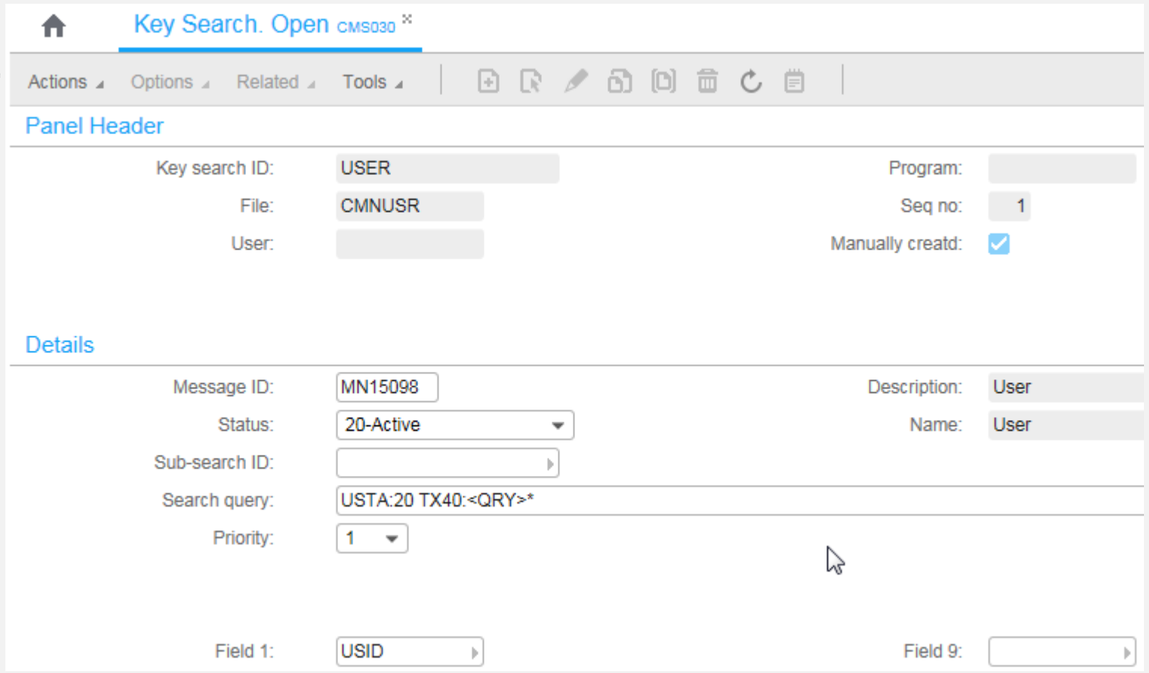


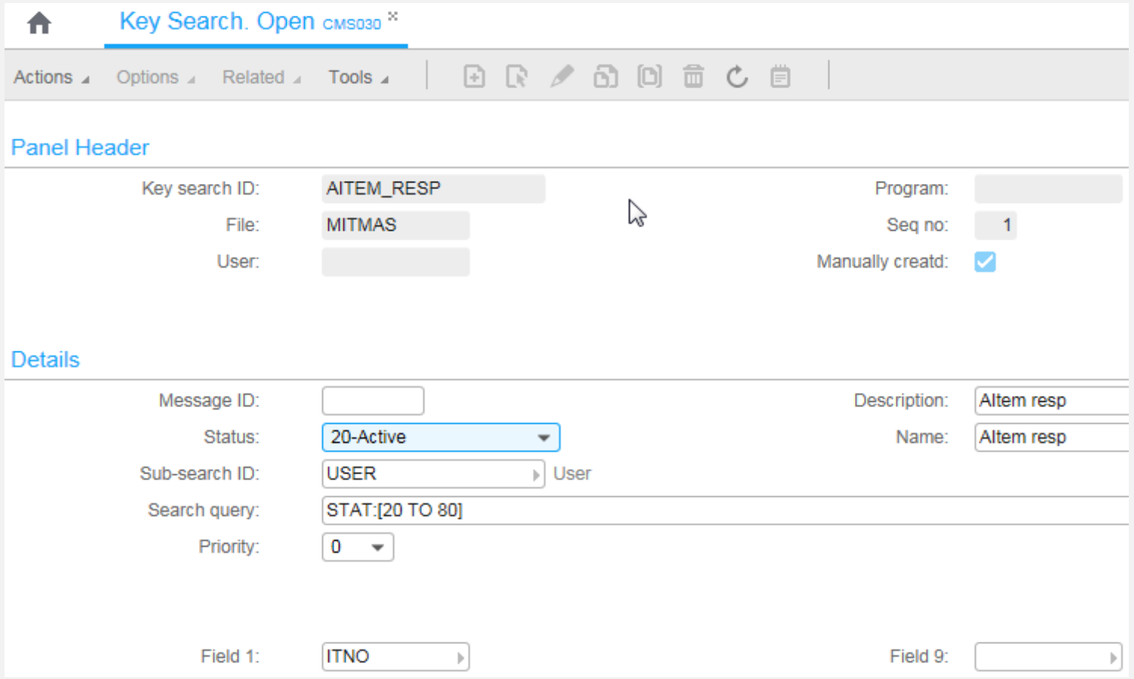
Adding several key searches like this makes it possible with same query from end user to search descriptions for different languages as well as aliases. Note that you connect keysearch to higher level in same way as if there was only a single key search

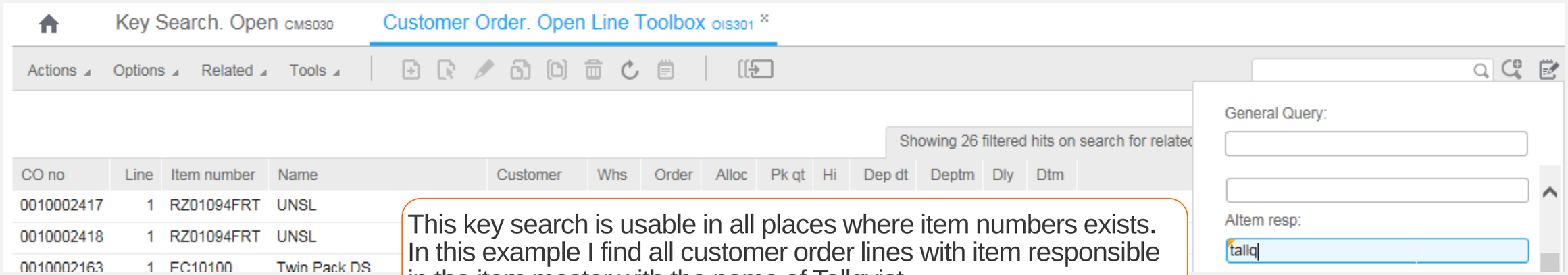
Multi level Key search

Another useful example is to search for items based on responsible persons name rather than he's user id.

Note that the below example requires that the field aliases (CMS031) has been generated with default values.

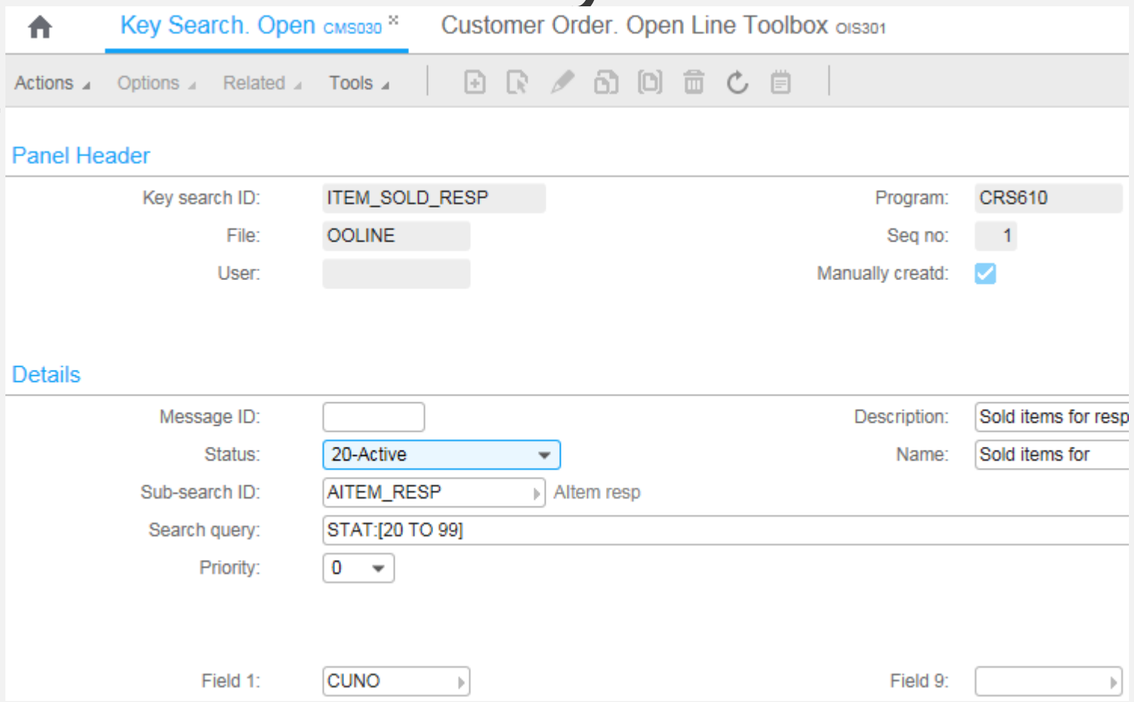




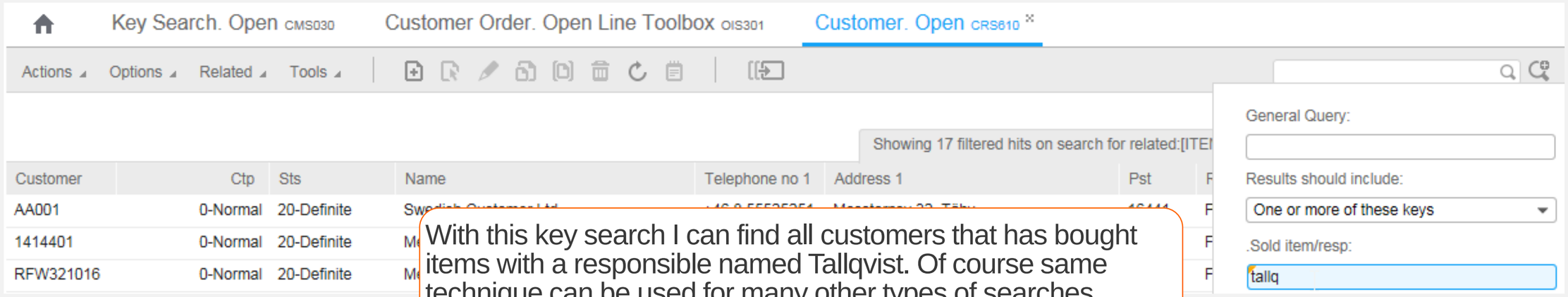


This key search is usable in all places where item numbers exists. In this example I find all customer order lines with item responsible in the item master with the name of Tallqvist.

Multi level Key search



The key search can be expanded to more than a single level. In the below example I add one more level on top of the item search and is instead finding all customer order lines for items with a certain responsible. Since the key search is returning customer number it is only of use in functions like the customer master so it has been restricted to only be used in the function for customers (CRS610)



With this key search I can find all customers that has bought items with a responsible named Tallqvist. Of course same technique can be used for many other types of searches.

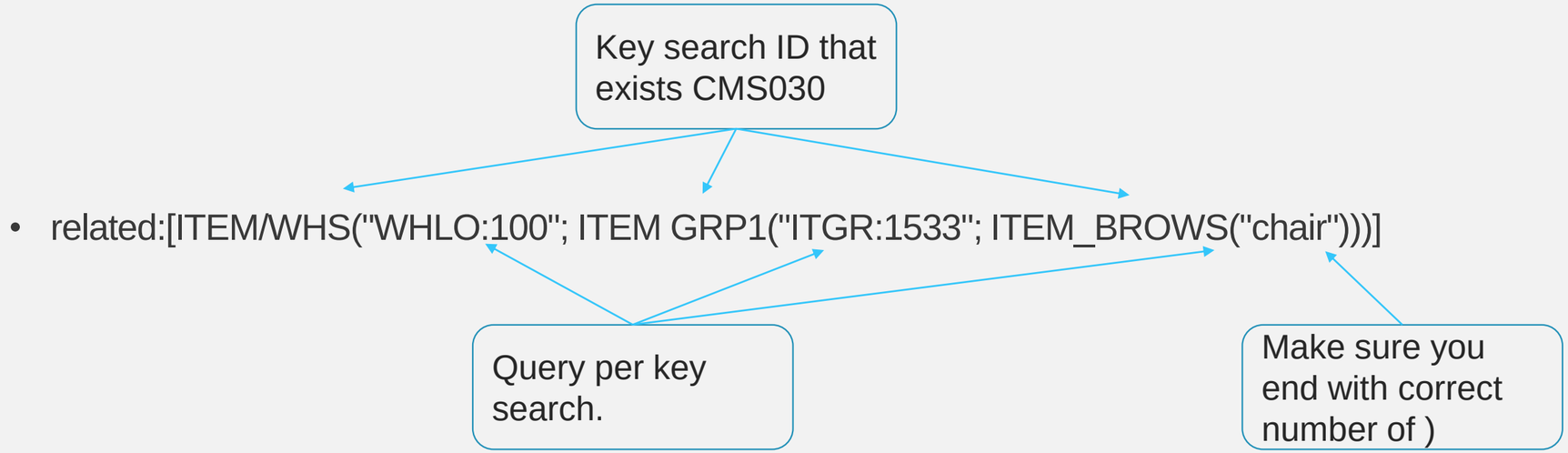
Very advanced search queries

Very advanced search queries

Please note that this section is mostly targeted to advanced IES users wanting to run IES search queries through code. No one is expected to write these queries by hand.

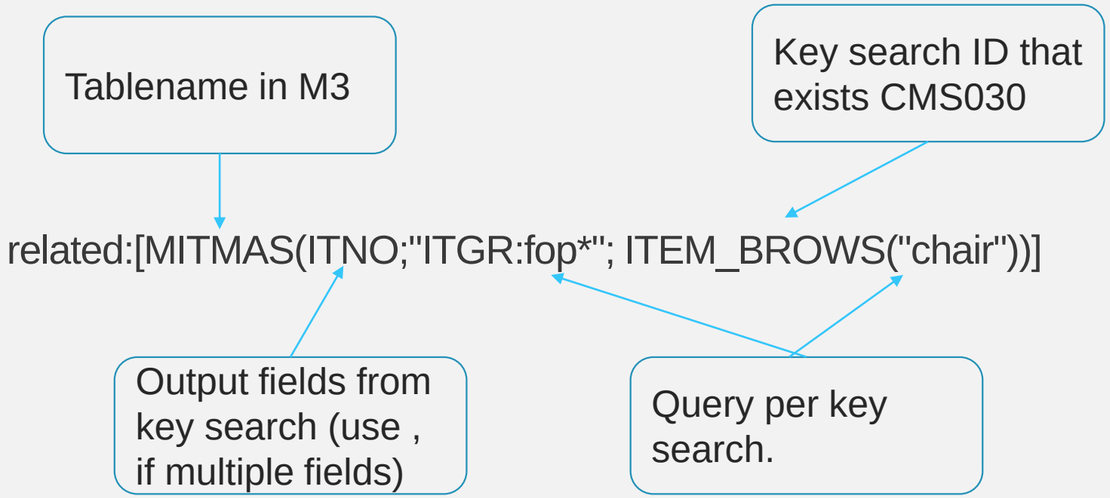
Key search without Ctrl+f UI/CMS030

Syntax used when defined in CMS030:



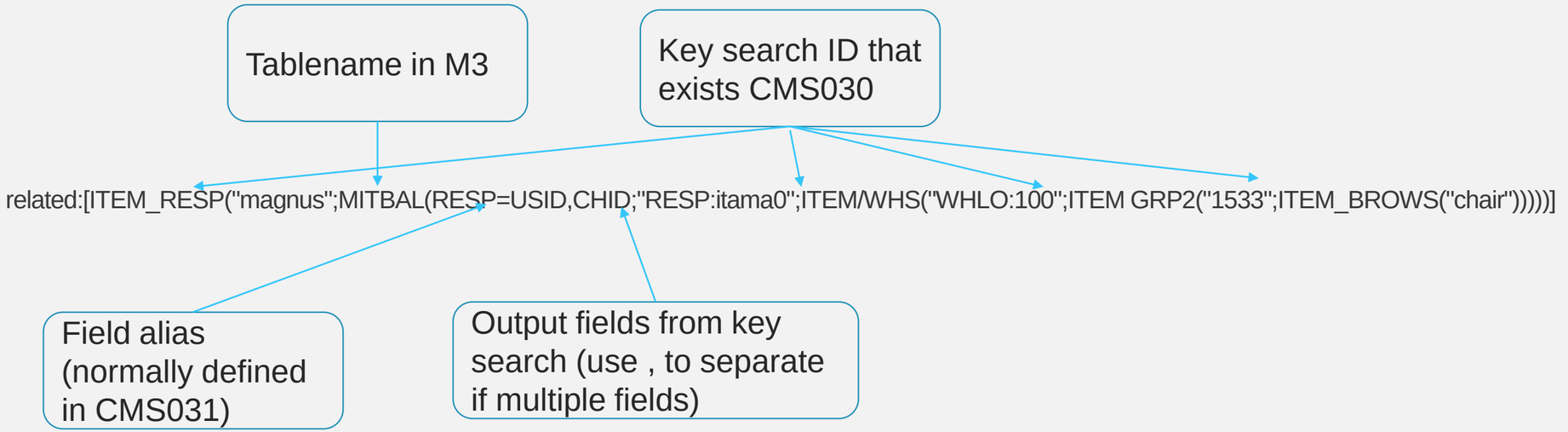
Key search without UI/CMS030

Syntax used when not defined in CMS030:



Key search without UI/CMS030

Syntax used when not defined in CMS030 and alias not in CMS031:



Various examples using key search

- related:[ORDEREDGT0_1(MITMAS(ITNO;"ITNO:cherry OR ITDS:cherry~ OR FUDS:cherry~ OR ITDS:cherry* OR FUDS:cherry*"))]
- related:[MITMAS(ITNO;"ITGR:fop*"; ITEM_BROWS("chair"))]
- related:[ITEM/WHS("WHLO:100"; ITEM GRP1("ITGR:1533"; ITEM_BROWS("chair")))]
- related:[MITBAL(ITNO;"RESP:itama0";ITEM/WHS("WHLO:100";ITEM GRP2("1533";ITEM_BROWS("chair"))))]
- related:[ITEM_RESP("magnus";MITBAL(RESP=USID,CHID;"RESP:itama0";ITEM/WHS("WHLO:100";ITEM GRP2("1533";ITEM_BROWS("chair"))))]
- related:[MITMAS(ITNO;"ITGR:fop*"; ITEM_BROWS("chair"))] NOT related:[ITEM/WHS("WHLO:100)] (separate key search. You can use AND (default if nothing entered), OR and NOT. In the above example it will list hits existing in the first part as long as they do not hits in the second key search
- MAXJOIN:2000 related:[MITMAS(ITNO;"ITGR:fop*"; ITEM_BROWS("chair"))] (Note MAXJOIN increases the number of hits internally in the key search. This overrides the setting in the IES local management pages. For more information please see NCR 4351).

Lucene references

Lucene Query syntax

IES uses Lucene search engine as a base. The documentation of lucene search syntax can be found here:

https://lucene.apache.org/core/2_9_4/queryparsersyntax.html


infor
TM