Description

Purpose: The purpose of this audit is to run a query to return the ***last person*** who modified a table.

Details: This audit can essentially be run from almost every single table because every table contains the appropriate fields to join to the person table (updt_id and updt_dt_tm). The idea is that to link to the tables, you set the primary key of that table (person_id, product_id, task_assay_cd, order_catalog_cd etc. equal to the updt_id.

In query builder, you can tell what your primary key is by a red key.

> If multiple people have modified the item you are looking at there is pretty much no way of telling the first person who modified it because the data gets overwritten after the last person modifies it.

Dates

All of the audit examples listed below contain a date range qualification that looks similar to:

curdate = current date

2300 = 24hr time (11pm)

Since curdate is the current date curdate-1 means yesterday, curdate-2 would be two days ago etc. By modifying the dates accordingly, it is easy to quality your audit during the correct time range. A lot more on dates can be found here.

Orderables

The following audit returns all orderables that have been modified within the defined date range and who modified them.

UPDT_ID	ORDERABLE	PERSON_ID	NAME_FULL_FORMATTED	UPDT_DT_TM	UPDT_APPLCTX
744099.00	acetaminophen	744099.00	Francisco, Tony	August 27, 2013 09:09:00	2923153.00
589923.00	linezolid	589923.00	TEST, DBA CERNER	August 28, 2013 23:28:47	2925553.00
589923.00	albuterol	589923.00	TEST, DBA CERNER	September 03, 2013 12:22:57	2933282.00

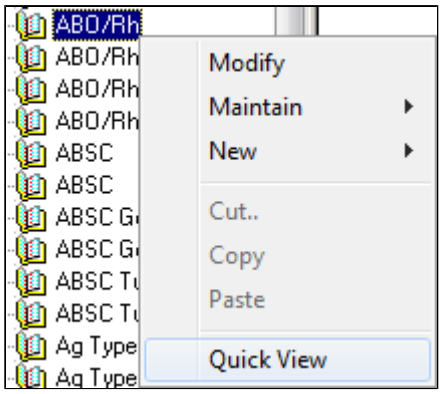
As you can see in the screen shot, there is no qualification on the activity_type_cd, meaning that the audit returns every single order within the given time range. To qualify on activity_type_cd insert the following code after the date range qualification

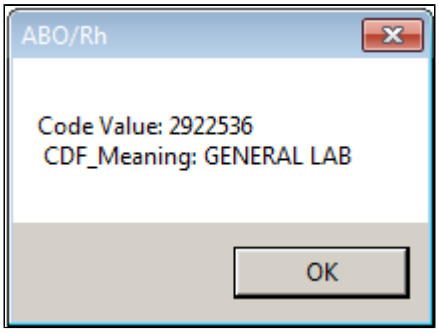
where *code_value* is the appropriate code_value. To find the appropriate code_value run the following script:

> Most of the time returning data from an entire data range won't be really useful. The following audit shows how to look at a specific order.

UPDT_ID	ORDERABLE	PERSON_ID	NAME_FULL_FORMATTED	UPDT_DT_TM	UPDT_APPLCTX
907962.00	ABO/Rh	907962.00	Townsend, Christin	July 19, 2012 08:44:49	2141707.00

The catalog_cd of an order can be found in DORC.





Products

The following audit is an example of querying the product_index table to return products that have been modified within the given time range.

> The product_index table is the product reference table (products built in Bedrock). Products received in inventory are located on the Products table.

UPDT_ID	PRODUCT	PERSON_ID	NAME_FULL_FORMATTED	UPDT_DT_TM	UPDT_APPLCTX
589923.00	RBC AS1 LR Irr	589923.00	TEST, DBA CERNER	August 29, 2013 16:24:24	2926516.00
589923.00	RBC AS1 LR Irr	589923.00	TEST, DBA CERNER	August 29, 2013 16:24:29	2926516.00
589923.00	PLT ACDA Irr	589923.00	TEST, DBA CERNER	August 29, 2013 15:38:22	2926437.00

To add a qualification to find a specific product remove the date code and add the following code:

where the product_cd can be found by running the following audit:

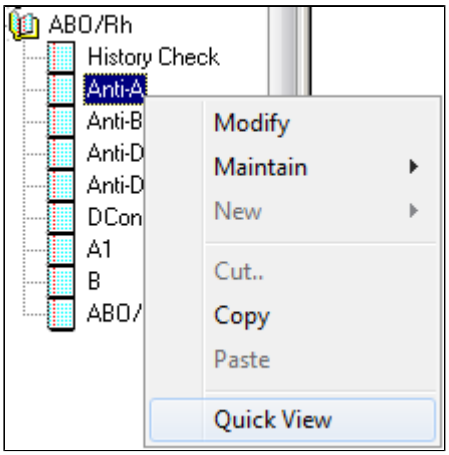
Assays

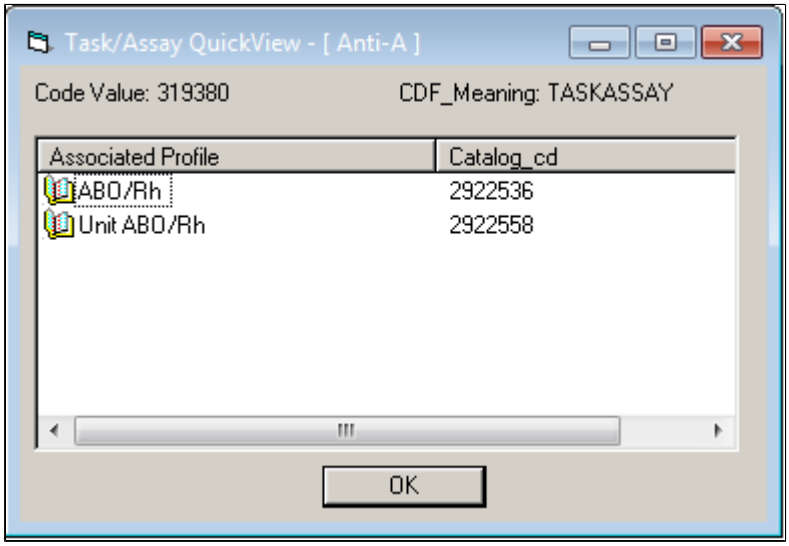
The following audit queries the discrete_task_assay table to return who modified an assay.

> The modification of an assay refers to modifying the actual assay - changing the name, result type, result processing type etc. The following examples show how to query who changed what order an assay was associated to.

UPDT_ID	DTA	PERSON_ID	NAME_FULL_FORMATTED	UPDT_DT_TM	UPDT_APPLCTX
589923.00	Anti-A	589923.00	TEST, DBA CERNER	May 15, 2013 13:16:55	2772088.00

The task_assay_cd can be found in DORC.





Assay Association

The following audit queries the profile_task_r table to determine who modifiied a task assay association between an assay and an orderable.

UPDT_ID	DTA	ORDERABLE	PERSON_ID	NAME_FULL_FORMATTED	UPDT_DT_TM	UPDT_APPLCTX
589923.00	Anti-A	ABO/Rh	589923.00	TEST, DBA CERNER	May 15, 2013 08:03:53	2771594.00

The *task_assay_cd* and the *catalog_cd* can be found in DORC as shown by the previous screen shots.

Digging Deeper...

There are ways in which you can find even more information that just who the last person to modify a table was. By using a front end security application and by querying the security tables, a number of useful pieces of information can be found: the application the update was made with, the date/time range in which the user was logged into the application for, the username of the individual who made the modification, the position or role of the individual and even additional modifications they made during that single session.

This can be done in two ways as stated above: querying the security table Application_Context or by using the front end application appctnxt.exe.

Querying the Application_Context Table

Notice in all of the previous screen shots the *updt_applctx* field was included on all of the audits. It was included because this is the field used to query the application_context table.

Using the assay association example above, we can take the *updt_applctx* and use it in the qualification of the following audit:

The following screenshots show the information returned by that audit.

APP_CTX_ID	APPLCTX	PERSON_ID	NAME	USERNAME	APPLICATION_NUMBER	POSITION_CD	START_DT_TM	END_DT_TM	APPLICATION_IMAGE
402017.00	2771594.00	589923.00	Cerner, DBA	CERDBA	15000	441.00	05/15/13	05/15/13	deptorcwizard

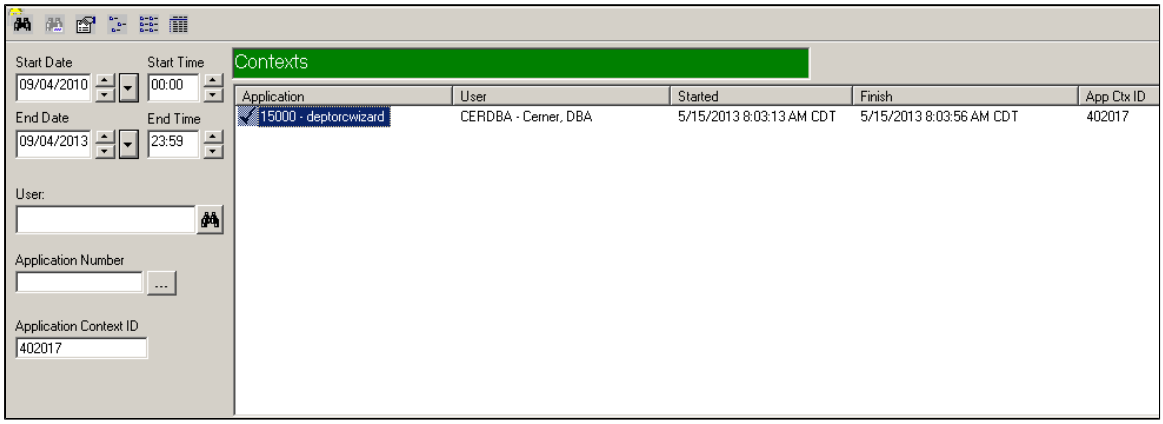
APPLICATION_DIR	APPLICATION_STATUS	PARMS_FLAG	DEVICE_LOCATION	DEVICE_ADDRESS	AUTHORIZATION_IND	APPLICATION_VERSION
C:\Program Files\CERNER\deptorcwizard.exe	0.00	0.00	Device Location Not Found	Device Address Not Found	1	2012.07.1.3

DEFAULT_LOCATION	CLIENT_START_DT_TM	CLIENT_NODE_NAME	TCPIP_ADDRESS	LOGDIRECTORY	UPDT_APPLCTX	UPDT_DT_TM	UPDT_ID	UPDT_CNT	UPDT_TASK	CLIENT_TZ
Default Location Not Found	05/15/13	SCTESTCTX2	10.190.130.180	C:\TEMP\ym025465	2771594.00	05/15/13	589923.00	1	15000	75

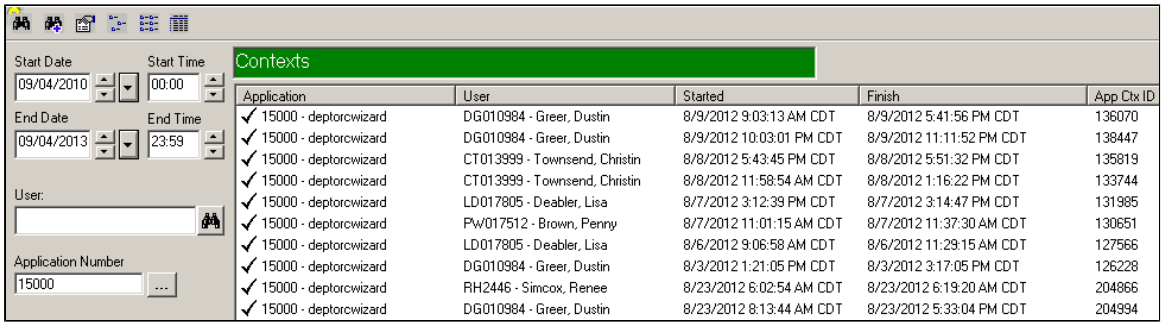
All information made during the same session will have the same updt_applctx. Therefore, if you were querying the product_index table for example and you created two-hundred products at one time, all two-hundred would pull in the previous audit.

Integrating with Appctnxt.exe

Again, the *applctx* field in the screen shot above is the *updt_applctx*. The useful piece of information in regards to inserting into the front-end application is the *app_ctx_id*. Entering it into the application provides some of the same information the audit of the application_context table showed.



Because we have the application used to make the update and the time range (Started and Finished) the user was logged into the application, we can see all changes made in that time range using this application.



It is possible to view all modifications by a user in a given session. Take for instance the first item on our list with an *app_ctx_id = 136070.* We can use this to query the application_context table again.

APP_CTX_ID	APPLCTX	PERSON_ID	NAME	USERNAME	APPLICATION_NUMBER	POSITION_CD	START_DT_TM	END_DT_TM	APPLICATION_IMAGE
136070.00	2305851.00	939935.00	Greer, Dustin	DG010984	15000	441.00	08/09/12	08/09/12	deptorcwizard

By running this query we get our *applctx* value.

Remember, the applctx is equivalent to the updt_applctx on any table

The *updt_applctx* can now be used to determine exactly what updates were made to the system. The next step is a bit of a process of elimination because you have to search a table for the *updt_applctx*. But, which table do you search? We can use a process of elimination - the only tables that can be updated are the ones in which modifications can be made via deptorcwizard (application ID 1500).

Several of the modifications:

orderables - order_catalog table
assays - discrete_task_r table

assay associations - profile_task_r table
reference range associations - reference_range_factor_id
routing - several tables

By going through the list of tables available the following query pulls data:

REFERENCE_RANGE_FACTOR_ID	TASK_ASSAY_CD	UPDT_DT_TM	UPDT_ID	UPDT_TASK	UPDT_CNT	UPDT_APPLCTX
21812551.00	21777066.00	08/09/12	939935.00	15001	0	2305851.00
21812552.00	21777066.00	08/09/12	939935.00	15001	0	2305851.00
21812553.00	21777066.00	08/09/12	939935.00	15001	0	2305851.00
21812554.00	21777066.00	08/09/12	939935.00	15001	0	2305851.00
21812555.00	21777066.00	08/09/12	939935.00	15001	0	2305851.00
21812568.00	21777068.00	08/09/12	939935.00	15001	0	2305851.00
21812569.00	21777068.00	08/09/12	939935.00	15001	0	2305851.00
21812570.00	21777068.00	08/09/12	939935.00	15001	0	2305851.00
21812571.00	21777068.00	08/09/12	939935.00	15001	0	2305851.00
21812572.00	21777068.00	08/09/12	939935.00	15001	0	2305851.00
21777072.00	21777066.00	08/09/12	939935.00	15001	1	2305851.00
21777074.00	21777066.00	08/09/12	939935.00	15001	1	2305851.00
21777076.00	21777066.00	08/09/12	939935.00	15001	1	2305851.00
21777080.00	21777068.00	08/09/12	939935.00	15001	1	2305851.00
21777082.00	21777068.00	08/09/12	939935.00	15001	1	2305851.00
21777088.00	21777066.00	08/09/12	939935.00	15001	1	2305851.00
21777090.00	21777068.00	08/09/12	939935.00	15001	1	2305851.00
21777078.00	21777068.00	08/09/12	939935.00	15001	1	2305851.00
21777086.00	21777068.00	08/09/12	939935.00	15001	1	2305851.00
21777084.00	21777066.00	08/09/12	939935.00	15001	1	2305851.00

This screen shot shows all of the data that was added/modified during a given session in the application (same *updt_applctx* value)

Page History

Version	Date	Comment
Current Version (v. 3)	**Sep 19, 2013 15:57**	**Magoon,Yitzhak**: Included a new version of the following page: Digging Deeper
v. 2	Sep 19, 2013 15:50	**Magoon,Yitzhak**: Included the Digging Deeper page
v. 1	Sep 19, 2013 14:53	**Magoon,Yitzhak**