- 1 Introduction
  - 1.1 Issues With Existing Rules
  - 1.2 BB_CALC_SPEC_EXPIRE Overview
- 2 Setting up Program
  - 2.1 Import the Program
  - 2.2 Defining OPT_PARAM Parameters
  - 2.3 Creating a Code Set Extension
  - 2.4 Linking Other Templates
  - 2.5 Page History

Introduction

The BB_CALC_SPEC_EXPIRE custom rule was created to help improve the blood ordering process from both a physician standpoint and a Blood Bank standpoint. Often times clients will implement rules that will either automate the ordering process or prompt the physician for the necessary orders he or she needs to place. For those clients who have their specimen expiration set to "Days" in the preference tool and not "Hours", this can pose a lot of confusion from both parties and create unnecessary phone calls to the lab.

Issues With Existing Rules

An example of a rule that attempts to automate the ordering process is "BB_VALID_SPEC_EXPIRE" which can be found on the "Blood Bank Rules" page under standard rules. The issue at hand is the fact that specimen validity is based on a set of rules that look back a number of hours (not days) for a set of orders. If a client has their preferences set to hours, the number of scenarios which the rules will not work decreases (neonate and pre-admit specimen extensions still will not). For clients who have the parameter say as days, the issue becomes more frequent. The problem can be explained in more detail by the excerpt below.

Specimen validity is based on a rule we have set up to look back four days for TS/XM orders. If a specimen is collected 6/4 @ 0800 it will be valid until 6/7 @ 2359 – a total length of 3 days 15:59 hours. The rule looks back 4 days or 48 hours (from 6/4 @ 0800 to 6/8 @ 0800), so between 6/8 @ 0:00 and 6/8 @0759 the rule will not add the TS/XM orders even though the specimen is expired. The techs will need to determine if a new specimen is needed and add the test manually on a new accession if this is the case.

Regardless of whether clients have their preference as hours or days, the rule will not function for any specimen that was extended. Because the rule only looks back 4 days and these specimens can be valid from 14-120 (depending on the type of specimen neonate, pre-admit etc.) there is no way they can function properly.

In addition to the aforementioned issues, another issue that causes extremely complex are orders on collection lists. Because the EKS_ORDER_FIND_L templates only filter orders based on their "Order Status" and not "Department Status", they find all orders that are: Scheduled, Dispatched, Collected, In-Lab, Pending, and Complete. This can cause great difficulty when you're trying to figure out whether to add orders to an existing specimen or to add orders to an existing encounter.

BB_CALC_SPEC_EXPIRE Overview

BB_CALC_SPEC_EXPIRE is a custom program called by the EKS_EXEC_CCL_L template that determines whether a patient has any valid specimens. Valid specimen is defined as a specimen that is current/non-expired. The program was designed to be used with the ADDTOSCRATCHPAD evoke, but it can be used with virtually any evoke including ORDER_EVENT, COLLECTION_EVENT and BB_RESULT_EVENT.

The custom program has three main functions:

- Determine if a patient has a valid specimen using a standard specimen validity script (the same one called when viewing available specimens in PPC).
- Filter orders defined to invalidate the specimen.
- Filter orders according as defined by the user for orders on collection lists.

Clients that have historical orders, product orders or any other communication orders built as Blood Bank tests will have those specimens returned as valid specimens in the Patient Product Inquiry available specimens alert icon. Because a 'true' specimen is not actually collected for these orders, they should be excluded from the rule.

The custom program allows you to filter those orders that the program should disqualify through the use of a code set extension off of the order catalog code set (200). This gives you the ability to ensure specimens that only have product orders etc. on them are not considered valid. More information on setting this up under the 'Creating a Code Set Extension' section below.

Another method of filtering that generally trumps code set filtering is parameter filtering. The program can accept parameters that are used to ensure certain orders are on the specimens for them to be valid. The program accepts four parameters corresponding to ABORh, Antibody Screen and Crossmatch orders as well as a status parameter. This allows the user to control the number of action groups and additional orders added via action templates with the fewest possible EKS_EXEC_CCL_L templates.

For example, if a user wants a rule to determine if a patient has a valid specimen *that contains* an ABORh and Antibody Screen order to determine whether to prompt the physician with a message, they can control that by including the parameters corresponding to ABORh and Antibody Screen. For more information on parameter filtering see the 'Defining OPT_PARAM Parameters' section below.

Setting up Program

It is highly recommended a user with backend group 0 access perform the steps to include the custom program.

Import the Program

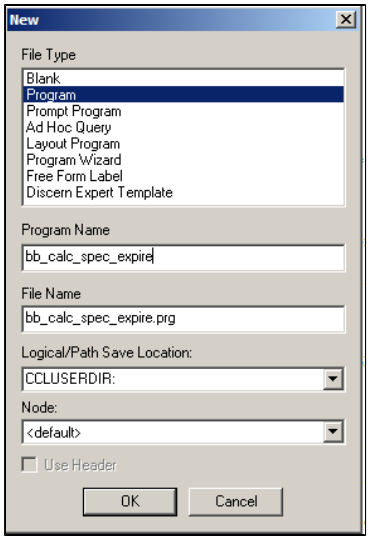
Create a new program and name it according to the screen shot below. Copy and paste the code from the BB_CALC_SPEC_EXPIRE program on the "Blood Bank Rules" page into DVD.

Do not copy

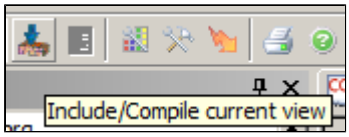
drop program bb_calc_spec_expire go
create program bb_calc_spec_expire

or

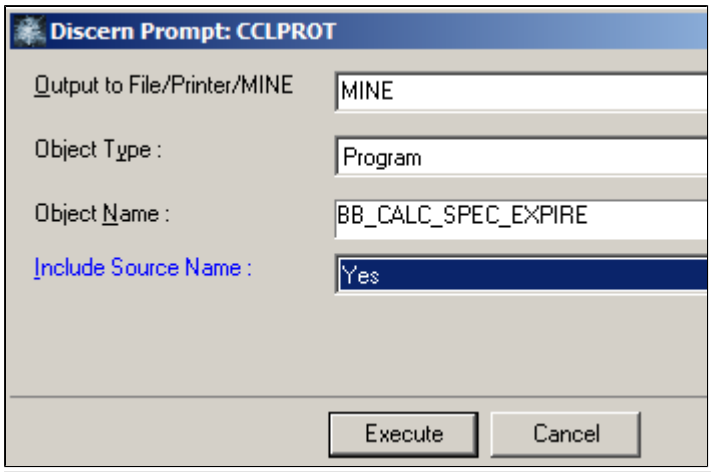
end
go

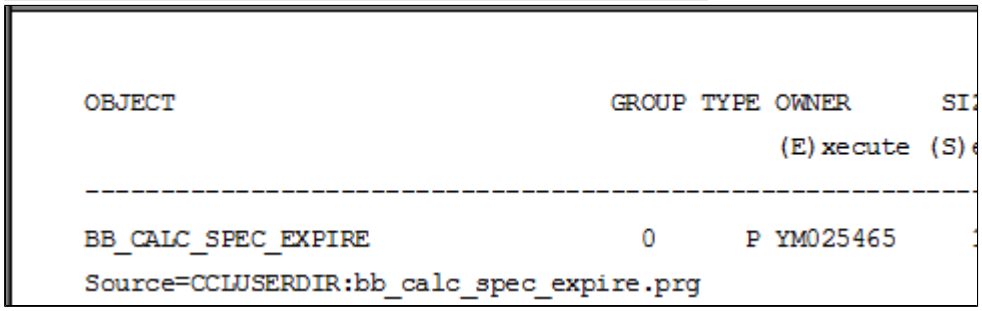


and include it into the domain.

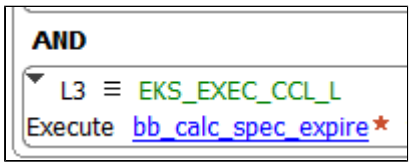


Open CCLPROT and search for program BB_CALC_SPEC_EXPIRE to verify if it was included as group 0 or as group 1. The group type controls how you need to name the program in the EKS_EXEC_CCL_L template.





Create the 'shell' of your rule and add the EKS_EXEC_CCL_L template. If the program was included as group 0, use the program name "bb_calc_spec_expire" otherwise use the program name "bb_calc_spec_expire:GROUP1".



Defining OPT_PARAM Parameters

The OPT_PARAM field on the EKS_EXEC_CCL_L template is required. The field requires four parameters separated by commans. The first field corresponds to ABORh, the second field corresponds to Antibody Screen and the third field corresponds to Crossmatch. The valid parameters are "ABORH" / "NULL", "ABSC" / "NULL", "XM" / "NULL" and "CURORDER" / "FUTUREORDER" where null indicates the order is not required on the specimen. **Quotes are required.**

The acceptable patterns are listed below.

ABORh Parameter	Antibody Screen Parameter	Crossmatch Parameter	Outcome
"NULL"	"NULL"	"NULL"	Any order regardless of their procedure type is considered valid.
"ABORH"	"NULL"	"NULL"	If an order with the procedure type "Patient ABORh" is not on the specimen, then it is not considered valid.
"NULL"	"ABSC"	"NULL"	If an order with the procedure type "Antibody Screen" *or* "Antibody Screen Comp Interp" is not on the specimen, then it is not considered valid.
"ABORH"	"ABSC"	"NULL"	If an order with the procedure types "Patient ABORh" and "Antibody Screen" *or* "Antibody Screen Comp Interp" are not on the specimen then it is not considered valid.
"ABORH"	"ABSC"	"XM"	If an order with the procedure types "Patient ABORh", "Crossmatch" and "Antibody Screen" or "Antibody Screen Comp Interp" are not on the specimen then it is not considered valid.

Department Status Parameter	Outcome
"CURORDER"	Only orders in a status of Collected, In-Lab, Pending and Completed will be used to determine if a specimen is valid (not expired).

"FUTUREORDER"	Only orders in a status of Scheduled and Dispatched will be used to determine if a specimen is valid.

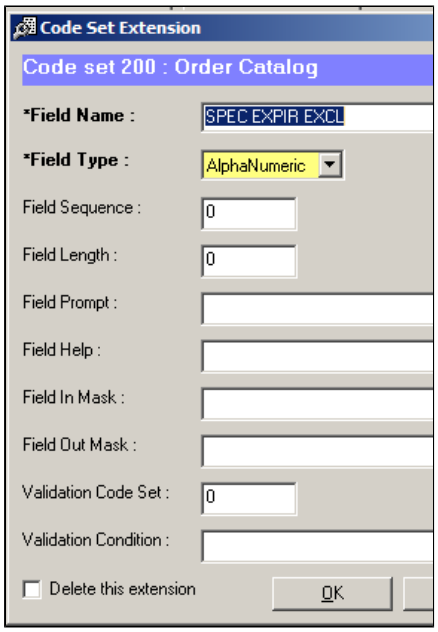
It is important to note that these are the **minimum** requirements. For instance, if the parameters are defined as ("ABORH","NULL","NULL") and the specimen contains an order for "Patient ABORh", "Antibody Screen Comp Interp" and "No Special Processing" the system still considers the specimen valid.

The following example shows how the parameters can be used. The rule would fire when a RBC order is placed - the bb_calc_spec_expire program will determine if a patient has a valid specimen and if there is a Type and Screen on it. If that scenario is true then the template will return 'true' otherwise it will return 'false'.

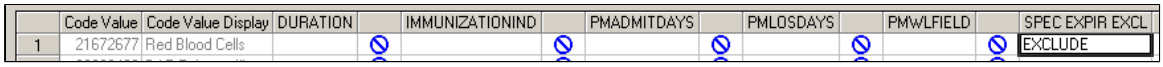
Creating a Code Set Extension

Creating the code set extension is required. Without creating it, regardless of whether any values are inputted, the script will not function properly. The code set extension allows you to further filter specimens that the standard script considers valid, but really aren't. Product orders built as "No Special Processing" on their own accession are prime examples of orders that do not have a true specimen drawn for them. These should be removed.

In CoreCodeBuilder.exe search for code set 200. This corresponds to the order catalog code set. Once it loads, click on the 'Code Value Extension' tab on the far right then click on 'Code Set Extension...". The field name must be "SPEC EXPIR EXCL" and is case sensitive. Fill out the remaining pieces of information shown in the screen shot below.

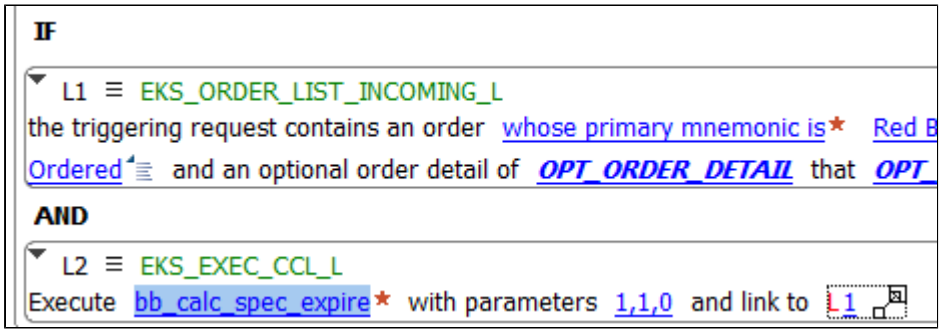


Once the code set extension is created, a list of code value's of the orders being excluded is needed. The easiest way to find this is in DeptOrderCatalog --> quick view. Once the list is compiled, scroll through the list of orders and enter "EXCLUDE" in the newly created SPEC EXPIR EXCL field. "EXCLUDE", just like "SPEC EXPIR EXCL" is case sensitive.



Linking Other Templates

Specimen validity is all determined based on a person's encounter. The custom program obtains the person's encounter_id by linking to another logic template. Therefore, defining L? is required or the template will always fail. Every single logic template stores a person's encounter_id, (even EKS_LOGIC_TRUE and EKS_STOP_LOGIC). It is good rule writing practice to repeat the evoke in the logic section anyway, so this should not be an issue.



The custom program stores four parameters that can be used by other logic or action templates: order_id, accession_id, person_id and encounter_id. Person_id and encounter_id are pretty straightforward as there are only one per rule execution. Order_id and accession_id differ, however, because a person can have multiple valid specimens (accession numbers) and each specimen can have more than one valid order.

The accession number accessible by other templates is determined by the drawn_dt_tm of the specimen. In other words, the specimen that is collected first, regardless of it's expiration date, is the specimen returned when linked by another template.

Page History

Version	Date	Comment
Current Version (v. 5)	**Jul 10, 2014 22:43**	**Magoon,Yitzhak**: rewrote entire program and fixed all of the optimization points
v. 4	Jun 27, 2014 11:05	**Magoon,Yitzhak**: Potential optimization - 1. allow rule to qualify even if there is no code_set_extension 2. Make OPT_PARAM optional if not entered default to (0,0,0) 3. Re-evaluate how to return order_id's and potentially incorporate container information 4. Incorporate collection list information into the rule design
v. 3	Jun 27, 2014 10:06	**Magoon,Yitzhak**
v. 2	Jun 27, 2014 09:23	**Magoon,Yitzhak**
v. 1	Jun 26, 2014 22:28	**Magoon,Yitzhak**