

## Rule Types other than IF/THEN/ELSE

If you have the Professional or Advanced versions of CRM Rules, you have the ability to create rules using some special Rule Types we have developed over the years in response to customer need. The other rules types are listed in the table below, and explained in this chapter.

### Count Children

The Count Children rule type enables you to count the number of records linked (1:N) to any CRM record. For example, an Account can have many Contacts. If you create a Count Children rule on the Account entity, targeting Contacts as the child entity, you can store the number of Contacts that are linked to each Account. (First, you would create a custom field to capture this number.)

This rule type is slightly unique in that no JavaScript is produced. Instead, a plug-in is activated against the Create, Update and Delete events on the child entity. The data you supply on the form is read by these plug-ins, and the appropriate calculation is performed.

So, to test, deploy the Count Children rule, then add a new Contact to the Account. After the plug-in runs asynchronously, (you may have to refresh the parent record a few times), you will see the count of Contacts linked to this Account appear in your custom field.

Tip: If you want to run this on production data that already exists, create a Workflow that updates a (perhaps new, or benign, unused OOB) field to some value. Run this on all child records. That will run the plug-in that populate the field on the parent, with the count of children linked to each parent.

As an example, let's suppose we want to count the number of Products linked to an Opportunity. Before you can start writing the rule, you have some prep work to do:

#### Prepare to Create the Rule

- 1) First, I create a custom field on the Opportunity called "Number of Opportunity Products" (whole number), and place it on the Opportunity form. This field will hold the result of the calculation of the count of the opportunity product records, and show that on each opportunity record.
- 2) Publish all Customizations in CRM (wait until finished).
- 3) Anytime you do that, you should also go into CRM Rules and "Create/Update" the entity that you modified. In this case, I'm opening the Opportunity entity, and clicking "Create/Update".
- 4) You actually need to make sure that you have run the "Create/Update Entity" process on both the PARENT and the CHILD entity! You must wait until one job finishes before starting the next job. (Tip: in 2013, you can use the "Back" button to go back to the Entity view...)

As you can see on the following page, when you "Create/Update" entity, we do a lot of work, reading the metadata in your CRM system, and populating tables used by our interface to capture the details behind your business rules. Each line in Deployment History is a system job. We do this to try and avoid the two minute timeout for all CRM system jobs.

## Result of clicking the “Create/Update” button for an entity, e.g. Opportunity

### ➤ Active Deployment History ▾

✓	Name	Status	Entity
	Create More Actions for opportunity	Success	Opportunity
	Create More Actions for opportunity	Success	Opportunity
	Create Conditions for regarding fields for opportunity	Success	Opportunity
	Create Lookup Conditions for opportunity	Success	Opportunity
	Create More Lookup Conditions for opportunity	Success	Opportunity
	Create Conditions for option set fields for opportunity	Success	Opportunity
	Create More Conditions for option set fields for oppo...	Success	Opportunity
	Create Conditions for text fields for opportunity	Success	Opportunity
	Create Conditions for DateTime fields for opportunity	Success	Opportunity
	Create Conditions for boolean fields for opportunity	Success	Opportunity
	Create Conditions for customer fields for opportunity	Success	Opportunity
	Create Conditions for partylist fields for opportunity	Success	Opportunity
	Create Conditions for int fields for opportunity	Success	Opportunity
	Create Conditions for owner fields for opportunity	Success	Opportunity
	Create option sets for opportunity	Success	Opportunity
	Create lookups sets for opportunity	Success	Opportunity
	Create Actions for opportunity	Success	Opportunity
	Create tabs and sections for opportunity	Success	Opportunity

Note that only if you have to create a lot of actions, will you see the “More Actions” line at the top. (But, as I ran this against an OOB opportunity entity online, you should see the same jobs appear on your system. Note that the Opportunity Product did NOT show these More Actions, Create Conditions for Text fields was the last line there.) As you can see we create conditions for every field. (Each job is divided by field type. These conditions vary by field type. For example, a Date field will have conditions like “After”, while a Number field will have “Less Than”.)

After waiting one to possible several minutes, until about 30 newly created records in the deployment history entity all show a status of success, (for both parent and child entities), I am then ready to build the rule.

### Create the Count Children Rule

Note that you need to “prep” both the parent and child entity before you can successfully build a “Count Children” rule. (But, you can prep several entities, for example, Account, Contact, and Opportunity, and write multiple rules at once.) We’re going to assume you have prepped the Opportunity and Opportunity Product entities for the remainder of this example.

- 5) Go to the Rule Definition entity, and click New.
- 6) Fill out the General tab with the “Count Children” Rule Type, as shown below.

The common mistake is to build this on the child entity.

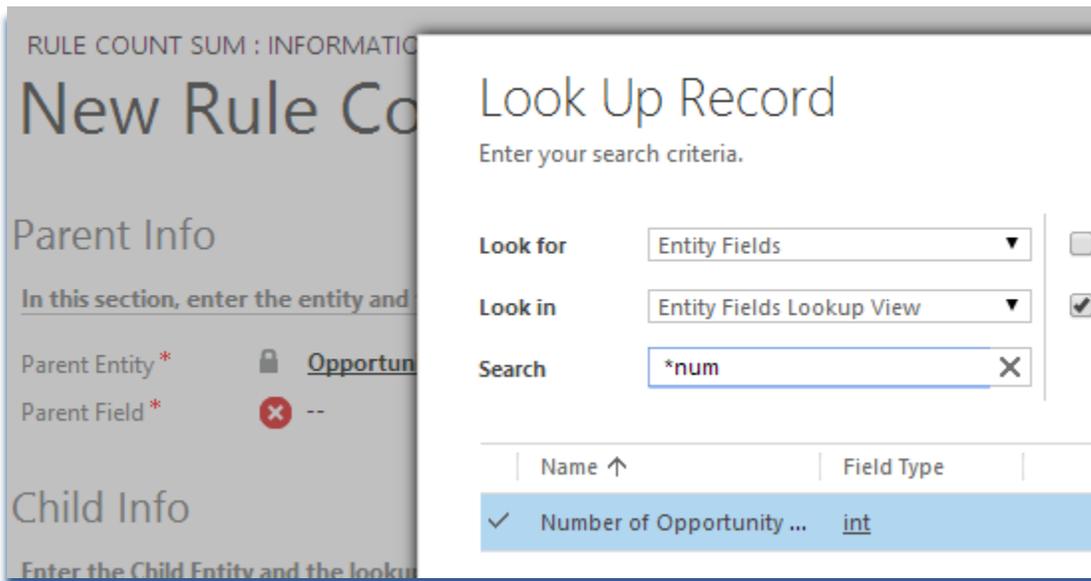
**The Entity should be set to the Parent entity**, where you will store and show the field that counts the number of children linked to each parent record.

- 7) You may see a Reload form dialog, if so, click Reload this page. (Part of auto save fun, I suspect). If so, just exit the form, the rule has been saved, just open it again...
- 8) Click on the Plus sign on the far right edge of the grid to create a new Rule Count Sum Detail record.

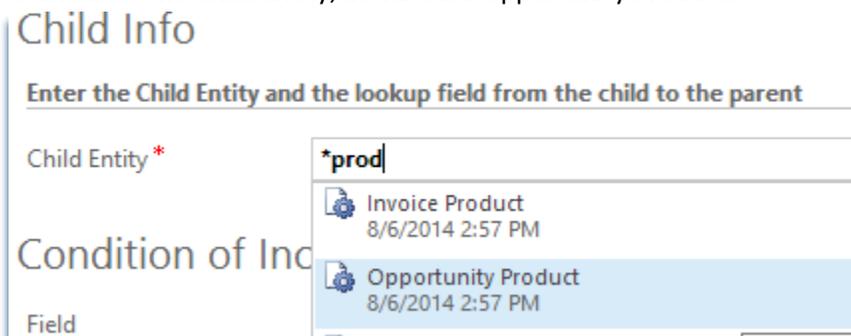
9) Then you have to click on the Lookup icon, then the New button that appears...



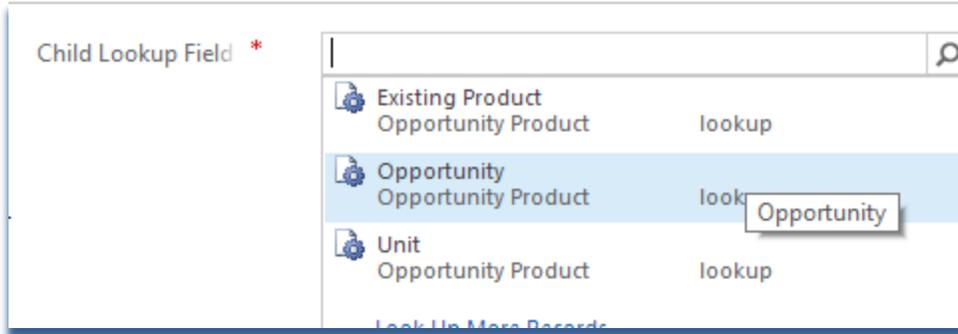
10) Finally, you get to the New Rule Count Sum page... First step is to identify the field on the Parent where the number of children linked to this parent will be stored. We created the new “Number of Opportunity Products” field for this purpose, so I navigate the lookup and select that value. (Click Add to select that field.)



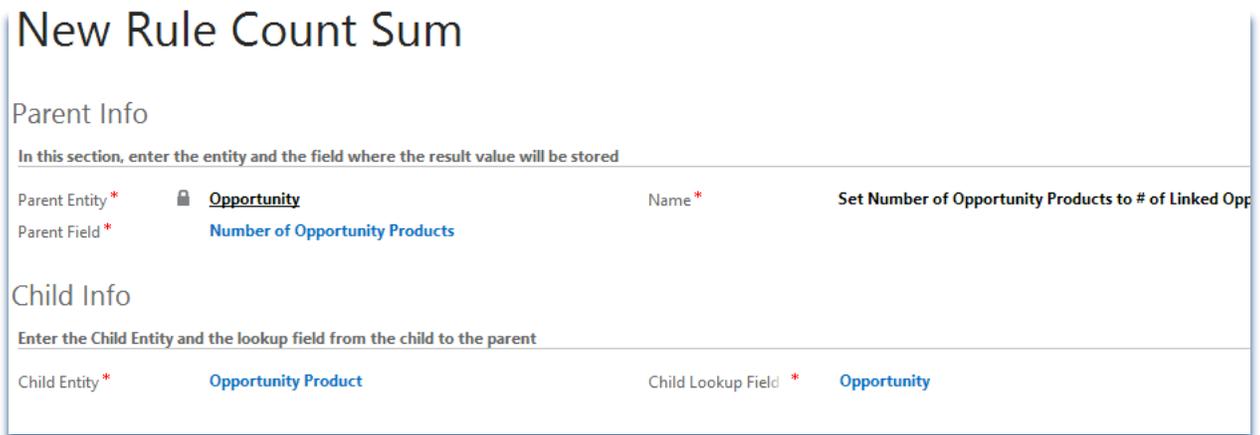
11) I then select the Child entity, in this case Opportunity Product.



12) I now select the Child Lookup field. This is the field on the Child record that is a lookup field, back to the Parent record we defined above. In this example, I select “Opportunity”.



13) My screen now looks as follows, as this is enough to define a rule that counts the opportunity products that are linked to each product. Note that the Name now shows a reasonable explanation of the rule. Edit if you wish.



14) At this point, we could save and close this new rule count sum definition, and this would count ALL of the opportunity product records that are linked to this opportunity.

15) BUT, there is one other option, new in 2014. You can now specify one conditional expression that will be used when counting children. If the data on the child record meet the condition you select, it will be counted. If it does not, it will not. In our example, Opportunity Product records do not have a state; they are either active, or deleted. So our rule, as specified, without a condition, will work.

But for example, if I am counting the Contacts linked to an Account, I can require that the STATE of the Contact be active. That way, if somebody deactivates a Contact record, I will NOT count that in my total for this linked Account.

- 16) Now that we have finished our counting rule, Save and Close the New Count Sum rule. You should now see this in the grid on the Rule Definition page.
- 17) Click the Validate button to Validate the rule. The status should be set to Ready to Deploy.

RULE DEFINITION | ADVANCED RULES ▾

## Count Opportunity Products

Step 1: Define the Rule Header

Count Opportunity Products

Rule Name\* **Count Opportunity Products**

Entity\* **Opportunity**

Rule Type\* **Count Children**

Function Name\* **CountOpportunityProducts**

Comments --

Rule Number(s) --

Rule Status **Ready to Deploy**

User Defined Status --

### Rule Count Sum Details

Name ↑	Child Entity	Child Field Nam...	Child Lookup Fl...	Parent Entity	Parent Field	Run Rule when ...	Created On	
Set Number of Opportunity Products to # of Linked ...	Opportunity...			Opportunity	Opportunity	Number of O...	No	8/10/2014 12:33...

- 18) Click the Deploy button. Click OK on the message box.
- 19) Open the Deployment History grid. Your last two entries should show that you successfully validated this rule, and then deploy the rules for the opportunity entity.

Active Deployment History ▾

✓ Name	Status	Entity	Deployment Start ↑	Deployment End	Deployment Type	Rule
Deploy for Opportunity	Success	Opportunity	8/10/2014 12:36 PM	8/10/2014 12:37 PM	Deploy and Publish for Current Entity	
Validate	Success		8/10/2014 12:33 PM	8/10/2014 12:33 PM	Validate Rule	Count Opportunity Prod...
Create Conditions for test fields for opportunityproduct	Success	Opportunity Product	8/10/2014 11:40 AM	8/10/2014 11:40 AM	Create/Update Conditions for Current Entity - Test Bas...	

### To test:

- 1) Open up an Opportunity record. Verify that the Number of Opportunity Products field is blank, or null.
- 2) Go through the process of adding a product line item to this opportunity. (I had to set up a price list, then a product), and save that product line item record.

OPPORTUNITY : MODIFIED OPPORTUNITY ▾

# second try at a sale

**Qualify (Active)**

- Identify Contact *click to enter*
- ✓ Identify Account **A. Datum Corporation (S**
- Purchase Timeframe *click to enter*

## Summary

NewDemoField --

Number of Opportur

Topic\* **second try at a sale**

## Product Line Items

Price List **Default**

Revenue **User Provided**

Product Name	Price Per Unit	Quantity	Discount	Extended Amount
sample	\$100.00	1.00000	\$0.00	\$100.00

After waiting for a few seconds, and refreshing the opportunity, I could see that the number of opportunities field now shows the number '1'.

Lower on that form, the Product Line Items grid also shows that 1 item.

- 3) Refresh that opportunity, and you can see the value appear in the Number of Opportunity Products field.

## Updating Parent Counts

Note that there is no way to “batch update” that parent field value... except for the following workaround...

You could write a workflow that updates a benign field on the child entity form (unused system field?), which would then trigger the update plug-in, which would then update each child’s parent record with that count. (Just have to run that workflow on each record. If you have millions, there are batch workflow utilities out there as well.)

Note that when you have multiple records attached to one parent, this workflow would run each time the child was saved. Each time, it would calculate the same number of children, and update the parent with that same number, one time for each child record. So you may want to turn off auditing on the parent, if doing this in production.