

## Steps to Successful Data Validation Rules

Data Validation rules released in Winter '07 provide a powerful mechanism to enforce process adherence and data quality. Data Validation rules allow administrators to leverage the power of the Apex platform and configure rules via “clicks not code”. For example, the VP of Sales of a newly acquired security solutions division could stipulate that we must have a competitive status field filled in for all deals over 100K

With this powerful new capability released our experts have defined a 7 step process to make sure you are successful deploying them.

Step 1: Understand what data validation rules are and where you can find more help about them.

- Release Notes (PDF) in the salesforce.com application
- [Getting Started with Validation Rules](#) (Breeze)
- [Useful Sample Validation Rules](#) (PDF)
- [New Feature Workshop](#) (Live Virtual Classroom)
- Help and Quick links in the Formula Wizard!

Step 2: In business terms, describe your business case. Break down your business case into steps. Understand the scenarios in which you will need a different message based on the error etc.

Let's build on the initial example: Our current sales organization has never distinguished their sales process in the various divisions. They always required our users to fill in the competitive status (Ahead of, Even, Behind). The security solutions division has a lengthier sales cycle (more stages), completely different competitors and only wants to require the competitive status when the deal is over 100K. They also want those that are over 500K to be approved by senior management. For new leads coming through the website, they want to enforce the “number of employees” are filled in for all customers interested in security solutions.

Step 3: Identify the complete list of features that help you build out the solution to address your business case

In this example we have 4 specific functional needs

- A record type to distinguish between the sales processes for the different divisions
- A dependent picklist for the relationship between the record type and the competitor picklist



- **A validation rule requiring the competitive status to be entered when the deal is over a 100K for the new division**
- An approval process for the deal over 500K to be reviewed by senior management

Focusing on my validation rule (since this is the reason for the blog posting) and breaking it down into steps:

Error message should occur informing the user that a competitive status is required when:

- 1) The sales division is anything other than new division
- 2) Sales division is new and the deal is greater than 100K

Dissecting the rule into a logical statement I'm left with:

Competitive Status is blank **AND** (sales division is not equal to "Security Solutions" **OR** opportunity amount is greater than 100K)

Final rule for entry:

```
AND(ISPICKVAL( Competitive_Status__c , ""), OR(NOT(ISPICKVAL( $User.Sales_Division__c , "Security Solutions")), Amount >100000))
```

Step 4: Make sure you haven't contradicted any other rules or any of the standard validations

Previously a required field existed for all sales division on the "competitive status" field; however the new sales division only wants rule for deals over a 100K. The previous need creates a contradiction here and my new business rule requires me to remove the required attribute for "Competitive Status" and use data validation for all opportunities for the old division and for those over 100K on the new division. My rule above handled this.

Step 5: Understand how this new rule impacts your integration or other client applications that are inserting and updating data in salesforce.com

Example: Previously we had an integration job that was auto creating new closed opportunities for "orders" place directly on our website. This was opened up for low dollar transactions to select customers. Since we incorporated this validation rule and the API respects this, we want to make sure that we put a competitive status of "Ahead of" for any of these opportunities.

Step 6: Understand areas that will be enforced without any return of an error (w2c,w2l)



Example: Leads created directly through the application will enforce the validation rule for users of salesforce.com (i.e. intra-department lead generation) however the web2lead forms will need to be modified to incorporate client side script to mimic the functionality of the validation as this case is similar to step 5 and the validation will be enforced through the API.

Step 7: Implement and Test your solution. With these types of rules it's important for you test your solution thoroughly.

Example: It's important to test your solution out thoroughly. Thus, you should deploy one of the following options.

Option 1 (using a test field on user or qualifying based on profile/role etc)

- 1) Set up a custom user field like "test validation" as a checkbox. Then include this field in all of your validation rules
- 2) Update the test user or a select few users in your company to have this field set.
- 3) Test out your solution
- 4) Once you are happy remove the setting on the users and revert to the old rule

Example opportunity rule in this case would have looked something like this:

```
AND( $User.test_validation__c, AND(ISPICKVAL( Competitive_Status__c , ""), OR(NOT(ISPICKVAL( $User.Sales_Division__c , "New")) , Amount >100000)))
```

Option 2 (leverage the sandbox edition)

If you have the sandbox edition it is advisable to test out your rules first against this edition and implement the tested solution in your production org.

