# If you can set enforcement for a rule, you can set up lack of enforcement

devblogs.microsoft.com/oldnewthing/20150521-00

May 21, 2015

Raymond Chen

One of the things you can do with an internal tool I've been calling *Program Q* is run a program any time somebody wants to add or modify a record. The program has wide latitude in what it can do. It can inspect the record being added/modified, maybe record side information in another table, and one of the things it can decide to do is to reject the operation.

> We have set up a validator in our main table to ensure that the widget being added or modified is priced within the approver's limit. But sometimes, there is an urgent widget request and we want to be able to bypass the validation temporarily. Is there a way to disable the validator just for a specific record, or to disable it for all records temporarily?

If you can set up a program to validate a record, you can also set up a program to *not* validate a record.

Suppose your current validator for adding a widget goes like this:

```
if (record.approver.limit < record.price) {
 record.Reject("Price exceeds approver's limit");
 return;
}
... other tests go here ...
```

And say you want to be able to allow emergency requests to go through even though, say, all approvers are unavailable. Because, maybe, the widget is on fire.

You could decide that a widget whose description begins with the word EMERGENCY is exempt from all validation, but it generates email to a special mailing list.

```
if (record.description.beginsWith("EMERGENCY"))
{
 // emergency override: send email
 // and bypass the rest of validation
 generateNotificationEmail(record);
 return;
}
if (record.approver.limit < record.price) {
 record.Reject("Price exceeds approver's limit");
 return;
}
... other tests go here ...
```

Of course, the EMERGENCY rule was completely arbitrary. You can come up with whatever rules you like. The point is: If you wrote the rules, you can also write the rules so that they have exceptions.

Raymond Chen

**Follow**