**Redacted**  
Salesforce Content Filtering Engine

Redacted is a Salesforce app for filtering content posted within a Salesforce organization by means of creating rules that apply to objects and their fields. The rules are written as regular expressions and various configuration options may be set beyond that, such as email alerts different configurable behaviors when a match is found. While it comes with out of the box sample configurations for chatter, it is easy to add filtering to any text fields on any object by follow the guide below (see extending functionality). It also comes with basic logging abilities, and administrative reporting to help isolate users who may be making undesirable posts.

**Features**

1. Apply filtering to any object, including chatter posts and comments\*
2. Customizable error messages for each rule (if error is enabled)
3. Email alerts when a rule is activated
4. Ability to remove matched text and replace with another message or block the save/update action entirely. Can also silently log data and take no additional action.
5. Ability to activate tracking to monitor how many times a rule has been activated.

*\*Comes with support for chatter objects, to support other object types see the ‘Extending Functionality’ section.*

**Installation**

1. Install the app from <https://login.salesforce.com/packaging/installPackage.apexp?p0=04tE0000000PcqZ>
2. Copy the code from the SRC folder (the class and triggers) and save them into your org.

*\*Due to what seems to be a bug in package deployment, the code could not be included in the package. Normally code and metadata would be included in a single package.*

**Usage**

Follow this guide to set up a basic filter that will disallow users from posting product numbers (3 letters, followed by 6 numbers, then another letter) in our org via chatter.

1. Install the application from the link
2. Change to the ‘Redacted’ app from the app chooser.
3. Click on the Redacted Filters Tab
4. Click the new button
5. Enter a name for your filter. Tick the active checkbox and the track matches checkbox.
6. Enter [a-zA-Z]{3}[0-9]{6}[a-zA-Z] in the pattern string field.
7. Select all the available options in the Applies To field.
8. Tick the Replace pattern checkbox.
9. Enter \*\*REDACTED\*\* in the replacement pattern box.
10. Tick the send email on match box.
11. Enter your email address in the Notification Email Address box.
12. Save

This is all that is required to enable filtering of our imaginary product numbers. Go into chatter and attempt to post ‘Man I love ABC123456D it is so awesome’. Upon posting you should see that the product number is replaced by the word ‘\*\*REDACTED\*\*’. You should also receive an email informing you of the match, and if you visit the rule configuration again you should see a tracking object attached to it.

You may also optionally choose to block posting entirely if the pattern is matched. In this case tracking objects are not created, and the post is not inserted. The text specified in the error message field is returned to the user.

**Extending Functionality**

You can use redacted to filter any text fields on any object. Enabling other objects requires only minimal coding and configuration changes.

1. Create an Apex trigger on the object to be filtered.
   1. Decide which fields will be filtered
   2. Find the ID field that contains the user Id of the creator (usually createdBy)
   3. Call the redacted.filterObject(list <sObject>, list<string> textFields, Id creatorId) method
2. Add the object type in the ‘Applies To’ multiple select list (only required for custom objects).
3. Create your new rule that applies to the desired object.

**Example:**

To create a filter for the a custom object called ‘Challenge\_\_c’ where we want to filter ‘Teaser\_\_c’ field and the ‘Requirements\_\_c’ field we would create a trigger that looks like this.  
  
trigger filterChallenge on Challenge\_\_c (before insert, before update)   
{  
 String[] filterFields = new String[] {‘Teaser\_\_c’,’ Requirements\_\_c’}; filterChatter.filterChatterPost(trigger.new,filterFields,'parentId');   
}

Save the trigger.  
Now in your Salesforce org, navigate to   
setup->create->objects->redacted filter->Applies to

Click New under picklist values.  
Type ‘Challenge\_\_c’ (without quotes) and save.  
  
You can now create a filter rule for Challenge objects that filters on the Challenge\_\_c object and the given fields specified in the trigger.

Please note that for any of the logging/reporting functionality of a rule to work, you must enable the ‘Track matches’ option. Otherwise the rule will run, however it will not keep any records of matches to run analytics on.

You can also combine rules, and have multiple rules per object. Do note that every rule requires calculation and if you being to have many rules in your org you may end causing performance issues or hitting governor limits.