



THE CUSTOMER SUCCESS PLATFORM  
SALES SERVICE MARKETING COMMUNITY ANALYTICS APPS

# Custom settings the Essential tool in your APEX utility belt

Martin Haagen  
Systems Architect  
@MartinHaagen

dreamforce

# Safe Harbor

Safe harbor statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.



# Salesforce Speaker

Speaker Title



# Martin Haagen

Systems Architect



FORCE.COM  
DEVELOPER



FORCE.COM  
ADVANCED DEVELOPER

## Session outline

- Writing the trigger as a inexperienced Salesforce developer
- Improving techniques to use best practices
- Custom Settings – what's that?
- Rebuilding using custom settings
- Using custom settings in a maintainable way – fourth time around.

## Business Case – what we are trying to do

When Opportunity gets “Closed Won” and

- The Referrer (custom lookup field, User) is a marketing user (Profile: “Custom: Marketing Profile”)
- Amount is greater than \$50k

Then Opportunity should get flagged:

- Update the field “Qualified for compensation”, checkbox field.



# First attempt to write the trigger

Let's look at some code

dreamforce

salesforce



# First attempt to write the trigger

Don't hard code IDs!

- Best practices:  
[https://developer.salesforce.com/page/Apex\\_Code\\_Best\\_Practices](https://developer.salesforce.com/page/Apex_Code_Best_Practices)
- Moving code between different environments / sandboxes
- Readability

```
// Check conditions and set field value
for (Opportunity Opp : closedWonOpps) {

    if (referrers.get(Opp.Referrer__c).ProfileId == '00eU0000000XVFB' &&
        Opp.Amount > 50000) {
        Opp.Qualified_for_Compensation__c = true;
    }

}
```



# Time to improve the trigger

Let's look at some code

dreamforce

salesforce

## Second round – querying for ID

- No hard coded IDs
- Code more portable
- Trigger is now more expensive – 1 extra SOQL query

```
// Check conditions and set field value
for (Opportunity Opp : closedWonOpps) {
    if (Profiles.contains(referrers.get(Opp.Referrer__c).ProfileId) &&
        Opp.Amount > 50000) {
        Opp.Qualified_for_Compensation__c = true;
    }
}
```

## Custom settings - where

- Can be found in setup under App Setup > Develop
- Created similar to sObjects with fields
- Loaded into the application cache
  - Can be accessed without needing to spend SOQL queries
  - Limited to 2MB
- Two types
  - List – used for application settings.
  - Hierarchy – used for personalized data. Can be overridden on a organization, profile or user level.



## Custom Settings

# Demo

Let's investigate a custom setting

dreamforce

salesforce



## How to access Custom Settings

- SOQL – but this will count against governor limits
- Custom Settings can be used in:
  - APEX
    - Custom settings is available as custom objects
    - `yourObject.getValue('name')` can be used to get a specific record / row
    - `yourObject.getAll()` can be used to get all records / row
  - Formula Fields and Validation Rules
    - Only hierarchy Custom Settings
    - `$Setup.yourObject.Field__c`
  - SOAP API
    - SOQL Query
  - AJAX API
    - SOQL Query

## Custom Settings – Writing Tests

- Running tests and SeeAllData=false
  - Create Custom Settings as they where sObjects
  - All test methods that run code Settings
  - Do not set SeeAllData=true
- test.loadData() – to load data
  - Easy to maintain test data



# Improving using custom settings

Let's look at some code

dreamforce

salesforce

## Third round – introducing custom settings

- Less SOQL queries
- Profile ID is not hard coded
- Profiles can be changed and added without changing code
- Is the structure of the Custom Setting maintainable?
- Text type Custom Settings are limited to 255 character in length

```
// Load Profiles
Set<string> Profiles = new Set<string>();
string QualificationProfiles = MyTriggerSettings__c.getValues('default').QualificationProfiles__c;
if (QualificationProfiles != null) {
    Profiles = new Set<string>(QualificationProfiles.split(',', 0));
}
```



## Settings a wrapper

- Small wrapper component for Custom Settings
- Available as
  - Source via GitHub  
<https://github.com/haagen/Settings>
  - Package  
<https://login.salesforce.com/packaging/installPackage.apexp?p0=04tU00000005aMM>
- Example implementation
  - Multiple field types / data types
  - Single values
  - List values
- 100% test coverage
- Documentation
  - Blog: <http://www.webaholic.se>

# Custom Settings in a maintainable way

Let's look at some code

dreamforce

salesforce

## Fourth round – Custom Settings in a maintainable way

- No added SOQL queries
- Profiles and Amount is easily changeable
- Limitation of text field (255 character) avoided
- Format of Custom Setting is maintainable

```
Set<string> Profiles = new Set<string>();  
for (object o : Settings.getList('QualificationProfiles')) {  
    Profiles.add((string) o);  
}
```

```
// Check conditions and set field value  
for (Opportunity Opp : closedWonOpps) {  
    if (Profiles.contains(referrers.get(Opp.Referrer__c).ProfileId) &&  
        Opp.Amount > (decimal) Settings.get('QualificationAmount')) {  
        Opp.Qualified_for_Compensation__c = true;  
    }  
}
```

## Summary

- Custom Settings is a way to store configuration data
- No SOQL needed to retrieve data
- Test methods needs to load data into the Custom Settings
  - Avoid using SeeAllData = true
- Move constants from code out to configuration
- Simplify sandbox management



# Q & A

Martin Haagen  
@MartinHaagen

dreamforce

salesforce



Thank You