

Overload Apex Class to be Controller AND Extension

July 24, 2009 by [David Schach](#) 2 Comments

Share

Tweet

Share

Share

Wow - today brought an interesting discovery. Here's the situation:

Coding the new premium version of Mass Update Contacts (details to come), I replaced the two parts of the page with Apex Components. This will allow the app to support custom address fields and international address formats.

I didn't want to write one ControllerExtension for the main page, a CustomController for the view section component, and another CustomController for the pageblocktable component. So here is the overloaded class constructor. Note that this works because an extension passes the StandardController to the constructor, and a CustomController passes nothing:

```
1 public with sharing class VersatileClass {
2
3     private Account account;
4
5     public VersatileClass(){
6         system.debug('OPERATING AS CONTROLLER');
7         if(System.currentPageReference().getParameters().get('id')==null){
8             //Include error checking here
9         } else{
10             string AId = System.currentPageReference().getParameters().get('id');
11             account = [select id, name from Account where id = :AId];
12             //And whatever else you want to do
13         }
14     }
15
16     public VersatileClass(ApexPages.StandardController controller) {
17         system.debug('OPERATING AS EXTENSION');
18         if(System.currentPageReference().getParameters().get('id')==null){
19             //Include error checking here
20         } else{
21             this.account = (Account)controller.getRecord();
22             //And whatever else you want to do
23         }
24     }
25 }
```

Enjoy! This should save people a lot of time.