

ProcessComposer

Table of Contents

1. Introduction to ProcessComposer	1
2. ProcessComposer Screens in Salesforce.....	2
2.1 Process Definition Detail Screen	2
2.2 Business Process related list on a Contact	2
2.2.1 List of Steps in a Process Definition.....	3
2.2.2 Visualization.....	3
2.2.3 Activities	3
3. Installation and Basic Setup	5
3.1 Installing ProcessComposer.....	5
3.2 Security Profile Configuration.....	7
3.2.1 End User Profile Security Configuration.....	7
3.3 Triggers, Classes, and Testing	11
3.3.1 Creating an Apex Trigger	11
3.3.2 Creating a Test Class	14
Index	18

1. Introduction to ProcessComposer

ProcessComposer™ empowers you to define and execute real-world processes in a Salesforce environment. A process is a series of Steps. At each Step, a Task or Event placeholder is created. The container holding process Steps is called a process Definition (PCE - Definition).

First, you add a new Record Type and Page Layout* to the object you want to use with your process. Next, you name a new process Definition and (optionally) add criteria to determine when the process should start. Then, you add Steps to the Definition, specifying for each Step what happens, who is responsible, and when it is due. Steps can progress one at a time, in Stages, or based on other criteria. Once you have added Steps to the Definition, you have a process. Defined processes are associated with your Contacts and Accounts. [See how it looks.](#)

Processes can be *defined* on a Case, Campaign, Opportunity, Business Process (a custom object added by ProcessComposer) as well as any custom object. When you create or edit records in the system, ProcessComposer monitors these actions and executes processes when your Definitions match the record.

Here is how to get started:

1. Follow the [installation](#) and [setup](#) instructions.
2. Watch our [ProcessComposer videos](#) and proceed to the [tutorials](#).
3. Learn how to [create](#) and [use](#) your own process definitions.

Ready to get started? First, some initial setup is required, which you will find in [Basic Setup](#). Once the setup is complete, we recommend you explore our [Tutorials](#), which illustrate setting up processes that utilize various pieces of the ProcessComposer functionality.

✔ **Note:** Refer to [version and pricing information](#) for a list of features that apply to each version of ProcessComposer.

* Not always required

2. ProcessComposer Screens in Salesforce

When you are up and running with ProcessComposer, here is what you will see in Salesforce.

With ProcessComposer, you define your Business Processes and add Steps to them. At each Step, a Task is created. When initiated on a Contact, a Business Process automatically creates Tasks as previous Tasks are completed, reflecting the flow of a real-world process.

2.1 Process Definition Detail Screen

The screenshot shows the 'PCE - Definition' page for 'Address Change'. At the top, there is a breadcrumb trail: '< Back to PCE - Definitions List'. Below this, there are links for 'PCE - Steps [5]', 'PCE - Stages [0]', 'Notes & Attachments [0]', and 'PCE - Definition History'. The main section is titled 'PCE - Definition Detail' and includes buttons for 'Edit', 'Delete', 'Visual', 'Clone', and 'Export'. Below this, there is a table with the following information:

PCE - Definition Name	Address Change	Is Active	✓
Categories		Object Name	Business Process
		Object Record Type	Address Change

2.2 Business Process related list on a Contact

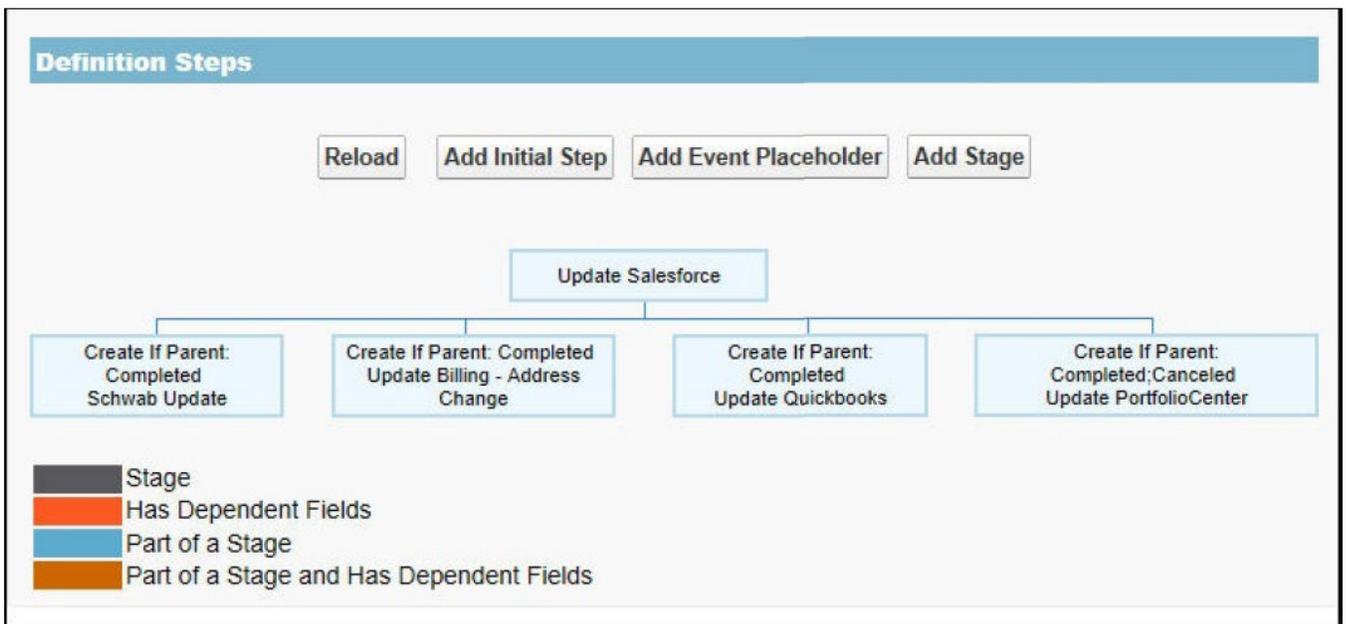
The screenshot shows the 'Business Processes' related list on a Contact page for 'Mrs. Tawni Plath'. The page includes a search bar, navigation tabs (Home, Accounts, Contacts, PCE - Definitions, ProcessComposer Help, Recurring Business Processes), and a 'Create New...' button. The 'Business Processes' list is currently empty, showing 'No records to display'. A 'New Business Process' button is visible at the bottom of the list. The 'Business Processes' link in the breadcrumb trail is circled in red.

2.2.1 List of Steps in a Process Definition

PCE - Steps							
Action	PCE - Step Name	Parent	Create if Parent	Due Date Adjustment	Assignee	Step Object	Field Deps
Edit Del New Child	Update Salesforce			Parent Object + 0 day(s)	User - Sayer Martin	Task - Master	0
Edit Del New Child	Update PortfolioCenter	Update Salesforce (Step)	Completed	Parent Object + 0 day(s)	User - Sayer Martin	Task - Master	0
Edit Del New Child	Update Quickbooks	Update Salesforce (Step)	Completed	Parent Object + 0 day(s)	User - Erica Martin	Task - Master	0
Edit Del New Child	Custodian Update	Update Salesforce (Step)	Completed	Parent Object + 0 day(s)	User - Sayer Martin	Task - Master	0
Edit Del New Child	Update Billing - Address Change	Update Salesforce (Step)	Completed	Parent Object + 0 day(s)	User - Erica Martin	Task - Master	0

2.2.2 Visualization

A visual depiction of how the Steps will execute. Here, four Steps will occur after the “Update Salesforce” Step has been completed.



2.2.3 Activities

A list of activities created by ProcessComposer. Here, after the ‘Update Salesforce’ Task was completed, four additional open Tasks were created on this Contact by ProcessComposer.

Open Activities

[New Task \(BP\)](#)[New Event \(BP\)](#)

Action	Subject	Name	Task	Due Date	Status	Priority	Assigned To
Edit Cls	Update PortfolioCenter	John Baker	✓	8/22/2012	Not Started	Normal	Saver Martin
Edit Cls	Update Quickbooks	John Baker	✓	8/22/2012	Not Started	Normal	Erica Martin
Edit Cls	Custodian Update	John Baker	✓	8/22/2012	Not Started	Normal	Saver Martin
Edit Cls	Update Billing - Address Change	John Baker	✓	8/22/2012	Not Started	Normal	Erica Martin

Activity History

[Mail Merge](#)[View All](#)[Log a Call \(BP\)](#)[Send Email \(BP\)](#)

Action	Subject	Name	Task	Due Date	Assigned To	Last Modified Date/Time
Edit Del	Update Salesforce	John Baker	✓	8/22/2012	Saver Martin	8/22/2012 2:05 PM

3. Installation and Basic Setup

In order to fully integrate ProcessComposer with your Salesforce organization, you must install the application and perform some setup. The instructions that follow take you through changing security profiles, editing layouts and adding a Picklist value, as well as overriding the Task and Event pages with the custom pages included with ProcessComposer. Once these steps are complete, you will be ready to automate your Business Processes.

3.1 Installing ProcessComposer

You must first have an account with www.salesforce.com before installing the ProcessComposer module.

1. Click the **ProcessComposer** installation link sent to you in an e-mail by Orchestrate,LLC.
2. Click **Continue**.
3. Log in to your **www.salesforce.com** account.

Package Installation Details Help for this Page ?

Package Name	ProcessComposer
Version Name	ProcessComposer Debut
Version Number	1.0 (Beta 27)
Publisher	Foster Group
Description	

Continue
Cancel

4. Click **Continue**.
5. Select the **Yes, access to these third-party web sites** checkbox to use this web service to manage access to ProcessComposer features. Orchestrate LLC can enable features based on the license level you purchase. In addition, this web service reports user counts to Orchestrate LLC for license auditing.

Approve Third-Party Access x


This package may send or receive data from third-party websites. Make sure you trust these websites. [What if you are unsure?](#)

Website	SSL Encrypted
fostergrp.secure.force.com	✓

Yes, grant access to these third-party web sites

Continue
Cancel

6. Click **Continue**.

Package Installer
Help for this Page ?

ProcessComposer

Step 1. Approve Package API Access
Step 1 of 3

These settings control the access that s-controls and other components in this package have to standard objects via the API. The access will still be constrained by the user's profile. You can view and edit the package API access to standard objects after the package is installed from the package detail page. [Tell me more](#)

Package Custom Objects

This Package will have the user's access (via the API) to all Custom Objects in your Organization.

Extended Object Permissions

	Read	Create	Edit	Delete		Read	Create	Edit	Delete
Accounts	✓	✓	✓	✓	Ideas	✓	✓	✓	✓
Assets	✓	✓	✓	✓	Leads	✓	✓	✓	✓
Campaigns	✓	✓	✓	✓	Opportunities	✓	✓	✓	✓
Cases	✓	✓	✓	✓	Price Books	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	Products	✓	✓	✓	✓
Contracts	✓	✓	✓	✓	Solutions	✓	✓	✓	✓
Documents	✓	✓	✓	✓					

General User Permissions

This Package will be able to use all of the General User Permissions from the user's Profile.

Administrative Permissions

This Package will be able to use all of the Administrative Privileges from the user's Profile.

Next
Cancel

7. **Grant access to all users** and click **Next**.

Package Installer
Help for this Page ?

ProcessComposer

Step 2. Choose security level
Step 2 of 3

Select security settings:

- Grant access to admins only Users with your profile get full access (best for limited deployments)
- Grant access to all users All internal custom profiles get full access
- Select security settings User access set by profile (recommended for most packages)

Previous
Next
Cancel

8. Click the **Install** button.



3.2 Security Profile Configuration

After you install ProcessComposer, configure [End User Security](#) access and [Administrator Security](#). An **End User** executes or participates in a process. An **Administrator** creates process definitions. Whether you chose to Grant Access to All Users or Grant Access to Administrators when installing ProcessComposer, follow the steps below and configure your security profiles to match. Apply these changes to the appropriate custom profiles in your organization. When you upgrade ProcessComposer, your security settings will not be overridden. Therefore, it is safe to choose Grant Access to All Users when upgrading ProcessComposer.

3.2.1 End User Profile Security Configuration

1. Select **Setup** from your account's user name menu **Administration Setup > Manage Users > Profiles**. Select the profile that applies to “End Users” as defined above. The screen shots were taken using the standard Profile Editor, if you use the Enhanced Profile Editor ensure that the settings match the screen shots below. You can enable and disable the Enhanced Profile Editor from Setup > Customize > User Interface > Check/Un-Check Enable Enhanced Profile User Interface. If you are unsure how to match up these settings with the Enhanced editor temporarily disable it while configuring ProcessComposer.
2. Click **Edit**. Make sure that your settings match the ones below.
 - Select the **ProcessComposer Visible** checkbox.



- Select **Tab Hidden** from the **PCE - Definitions** drop-down list box.
- Select **Tab Hidden** from the **ProcessComposer Help** drop-down list box.
- Select **Default Off** from the **Recurring Business Processes** drop-down list box.



- Select the **Read, Create, Edit** and **Delete** checkboxes for:
 - **Business Processes.**
 - **PCE – Team Members.**
 - **Recurring Business Processes.**
- Select the **Read** checkboxes for:
 - PCE – Definitions.
 - PCE – Definition Criteria.
 - PCE – Stages.
 - PCE – Steps.
 - PCE – Step Field Dependencies.

Custom Object Permissions													
	Basic Access				Data Administration			Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All		Read	Create	Edit	Delete	View All	Modify All
Business Processes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PCE - Step Field Dependencies	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Client Service Trades	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PCE - Team Members	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Google Campaigns	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Recurring Business Processes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Keywords	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Search Phrases	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PCE - Definitions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SFGA Version	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PCE - Definition Criteria	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Test PCEs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PCE - Stages	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Test PCE Steps	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PCE - Steps	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Text Ads	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Click **Save**.
4. Navigate to the **Selected Record Type** section.
5. Click **Edit** next to Cases.

Record Type Settings			
Standard Record Type Settings			
Accounts			Events
Campaigns	Standard (Default), FSTR_Test [Edit]		Leads
Campaign Members			Opportunities
Cases	Standard (Default), FSTR_Test [Edit]		Standard (Default), FSTR_Test [Edit]
Contacts			Products
Content			Solutions
Contracts			Tasks

 **Note:** If this is a profile for Platform Users; Cases, Campaigns and Opportunities will not be available, skip to step 10.

6. Remove the **FSTR_Test** Record Type from the Selected Record Types list.
7. If there are no other Record Types selected, add – **Master** -- as the only Selected Record Type.

Record Type Settings
Edit Save Cancel

User Profile High Volume Customer Portal User
Record Type Case

Selected Record Types

Select the record types for this user profile. You need to add the record type field associated with this profile to display it on record detail and edit pages.

Available Record Types **Selected Record Types**

Standard
FSTR_Test

--Master--

Add
▶
◀
Remove

Default Record Type

Select the default record type for this user profile. The default record type is used on the record detail page.

Default | --Master-- ▼

Save Cancel

8. Click **Save**.
9. Repeat for **Campaigns and Opportunities**.
10. Repeat for **Business Processes** but remove the **Sample Process** Record Type instead of FSTR_Test.
11. Navigate to the **Enabled Visualforce Page Access** section and click **Edit**.

Enabled Visualforce Page Access	
Visualforce Page Name	AppExchange Package Name
AnswersHome	
BandwidthExceeded	
ChangePassword	
Exception	
FSTR.DefCriteriaList	ProcessComposer
FSTR.DefinitionClone	ProcessComposer
FSTR.DefinitionExport	ProcessComposer
FSTR.DefinitionImport	ProcessComposer
FSTR.DefinitionList	ProcessComposer
FSTR.DefinitionListExport	ProcessComposer
FSTR.DefinitionVisual	ProcessComposer
FSTR.Event	ProcessComposer
FSTR.Field_Dependency_Edit	ProcessComposer
FSTR.LicenseSync	ProcessComposer
FSTR.PCEFeatureEditor	ProcessComposer

12. **Enable Visualforce Page Access** to the pages specified in the following screen shot.

Enable Visualforce Page Access

Available Visualforce Pages		Enabled Visualforce Pages
<div style="border: 1px solid gray; padding: 5px; min-height: 200px;"> AnswersHome BandwidthExceeded ChangePassword Exception FileNotFound ForgotPassword ForgotPasswordConfirm IdeasHome InMaintenance MyProfilePage SiteLogin SiteRegister SiteRegisterConfirm SiteTemplate </div>	Add <input type="button" value="▶"/> <input type="button" value="◀"/> Remove	<div style="border: 1px solid gray; padding: 5px; min-height: 200px;"> FSTR.DefCriteriaList FSTR.DefinitionList FSTR.DefinitionVisual FSTR.Event FSTR.PCE_Definition_View FSTR.PCE_Step_View FSTR.Recurring_Business_Process FSTR.StageView FSTR.StepView FSTR.Task </div>

13. Click **Save**.

14. Navigate to the **Enable Apex Class Access** section and click **Edit**.

Enabled Apex Class Access	
Edit	
Apex Class Name	AppExchange Package Name
FSTR.ProcessComposerExecution	ProcessComposer
FSTR.ProcessRBP	ProcessComposer
FSTR.RBPEExecution	ProcessComposer
FSTR.RecurringBusinessProcessController	ProcessComposer
FSTR.ScheduleRBPEExecution	ProcessComposer

15. **Enable Apex Class Access** as specified in the following screen shot.

Enable Apex Class Access

Available Apex Classes		Enabled Apex Classes
ChangePasswordController ForgotPasswordController MyProfilePageController SiteLoginController SiteRegisterController FSTR.ProcessRBP FSTR.ScheduleRBPEExecution	Add <input type="button" value="▶"/> <input type="button" value="◀"/> Remove	FSTR.ProcessComposerExecution FSTR.RBPEExecution FSTR.RecurringBusinessProcessController

16. Click **Save**.

3.3 Triggers, Classes, and Testing

3.3.1 Creating an Apex Trigger

A trigger is Apex code that executes before or after specific data manipulation language (DML) events occur, such as before object records are inserted into the database, or after records have been deleted. A list of all triggers in your organization is located at Your Name > Setup > Develop > Apex Triggers. You can invoke Apex through the use of triggers. Apex triggers are stored as metadata in the application under the object with which they are associated.

A trigger is Apex code that executes before or after the following types of operations:

- Insert.
- Update.

- Delete.
- Merge.
- Upsert.
- Undelete.

✓ **Note:** You must create a Page Layout and a Record Type for each process that you desire to automate.

1. Select **Your Name > Setup > Create > Objects.**
2. Click the custom object **Label.**

Custom Objects [Help for this Page](#)

Custom objects are database tables that allow you to store data specific to your organization in salesforce.com. You can use custom objects to extend salesforce.com functionality or to build new application functionality.

Once you have created a custom object, you can create a custom tab, custom related lists, reports, and dashboards for users to interact with the custom object data. You can also access custom object data through the Force.com API.

New Custom Object

Action	Label	Installed Package	Master Object	Deployed	Description
Edit	Business Process	ProcessComposer		<input type="checkbox"/>	This object can be customized to support a company's business processes. It can also be used as a template to guide in create your own custom object.
Edit Del	Documentation			<input checked="" type="checkbox"/>	
Edit	PCE - Definition	ProcessComposer		<input type="checkbox"/>	Container to define business processes for an organization.

3. Scroll down to the **Triggers** frame and click the **New** button.

Triggers [Triggers Help](#)

New

Action	Name	Api Version	Is Valid	Status	Size Without Comments	Last Modified By
Edit Del	Documentation	22.0	✓	Active	923	Tawni Plath , 8/22/2011 4:24 PM

4. Copy the following code:

```
trigger TRIGGERNAME on YOUR_CUSTOM_OBJECT_API_NAME (after insert,
before insert, before update, after update, before delete) {
    //after triggers
    if (Trigger.isAfter) {
        /*
            This part creates first group tasks when new object
            is created
        */
        if(Trigger.isInsert)
            FSTR.ProcessComposerExecution.CreateSteps(Trigger.new
    );
        if(Trigger.isUpdate){
```

```

        FSTR.RBPExecution.AfterObjectUpdate(Tri
Trigger.new);
        FSTR.ProcessComposerExecution.CreateStepsOnUpdate(Tri
gger.old, Trigger.newMap);
    }
}

/*
The trigger is fired before creation of new object
*/
if (Trigger.isBefore) {
    if (Trigger.isInsert || Trigger.isUpdate){
        if(Trigger.isInsert){
            FSTR.ProcessComposerExecution.DetermineBusiness(T
rigger.new);
            for(YOUR_CUSTOM_OBJECT_API_NAME newRec :
Trigger.new) {
                //set status to open
                newRec.Status__c = 'Open';
            }
        }
        FSTR.ProcessComposerExecution.DetermineDefinition(Tri
gger.new);
    }
    else if(Trigger.isDelete){
        for(YOUR_CUSTOM_OBJECT_API_NAME delRec : Trigger.Old)
        {
            if(delRec.Recurring_Business_Process__c != null)
                delRec.addError('You cannot delete a YOUR
CUSTOM OBJECT LABEL HERE that was generated by a Recurring
Business Process.');
```

5. Paste the **code** in the Apex Trigger frame.

Apex Trigger

Help for this Page ?

Apex Trigger Edit Save Quick Save Cancel

Apex Trigger Version Settings

Is Active

```

1 trigger TRIGGERNAME YOUR_CUSTOM_OBJECT_API_NAME (after insert, before insert,
2 before update, after update, before delete) {
3     //after triggers
4     if (Trigger.isAfter) {
5         /*
6          * This part creates first group tasks when new object is created
7          */
8         if (Trigger.isInsert)
9             FSTR.ProcessComposerExecution.CreateSteps(Trigger.new);
10        if (Trigger.isUpdate) {
11            FSTR.RBPExecution.AfterObjectUpdate(Trigger.old, Trigger.new);
12            FSTR.ProcessComposerExecution.CreateStepsOnUpdate(Trigger.old, Trigger.newMap);
13        }
14    }
15    /*
16    The trigger is fired before creation of new object
17    */
18    if (Trigger.isBefore) {
19        if (Trigger.isInsert || Trigger.isUpdate) {
20            if (Trigger.isInsert) {
21                FSTR.ProcessComposerExecution.DetermineBusiness(Trigger.new);
22                for (YOUR_CUSTOM_OBJECT_API_NAME newRec : Trigger.new) {
23                    //set status to open
24                    newRec.Status__c = 'Open';
25                }
26            }
27            FSTR.ProcessComposerExecution.DetermineDefinition(Trigger.new);
28        }
29        else if (Trigger.isDelete) {
30            for (YOUR_CUSTOM_OBJECT_API_NAME delRec : Trigger.Old) {
31                if (delRec.Recurring_business_Process__c != null)
32                    delRec.addError('You cannot delete a YOUR_CUSTOM_OBJECT LABEL
33                HERE that was generated by a Recurring Business Process.');
```

Position: Ln 37, Ch 2 Total: Ln 37, Ch 1444

6. Replace line 1, **TRIGGERNAME**, with the name of your new trigger. Use underscores instead of spaces.
7. Replace **YOUR_CUSTOM_OBJECT_API_NAME** as in lines 1, 22, 30, and 32 with the name of your custom object.
8. Click the **Save** button.

3.3.2 Creating a Test Class

Create the following Class in order to deploy the trigger to the production environment.

1. Select **Your Name > Setup > Develop > Apex Class**.
2. Click the **New** button.
3. Copy the following code:

```

@isTest
private class CustomWorkflowObject {
    private static testMethod void Test_CustomWorkflowObject ()
    {
        CustomWorkflowObject__c fw = new
CustomWorkflowObject__c();
        insert fw;
        System.assertNotEquals(null, fw.id);
    }
}

```

4. Paste the **code** in the frame.

Apex Class [Help for this Page](#)

Apex Class Edit Save Quick Save Cancel

Apex Class Version Settings

A A

```

1 @isTest
2
3
4 private class CustomWorkflowObject
5
6
7     private static testMethod void Test_CustomWorkflowObject () {
8
9
10        CustomWorkflowObject__c fw = new CustomWorkflowObject__c();
11
12
13        insert fw;
14
15
16        System.assertNotEquals(null, fw.id);
17
18
19    }
20
21
22 }

```

Position: Ln 22, Ch 2 Total: Ln 22, Ch 243

5. Replace lines containing **CustomWorkflowObject** with the name of your Object. There may be multiple occurrences in a line.
6. Click the **Save** button.
7. Click the **Run Test** button.

Apex Class [Help for this Page](#)

TestObject

[« Back to List: Apex Classes](#)

Apex Class Detail [Edit](#) [Delete](#) [Download](#) [Run Test](#) [Security](#) [Show Dependencies](#)

Name	TestObject	Status	Active
Namespace Prefix		Is Valid	✓
Created By	Tawni Plath , 8/1/2011 12:35 PM	Last Modified By	Tawni Plath , 8/8/2011 3:30 PM

Class Body [Class Summary](#) [Version Settings](#) [Log Filters](#)

```
1 @isTest
2 private class TestObject {
3
4     private static testMethod void Test_CustomWorkflowObject () {
5         Test__c fw = new Test__c();
6         insert fw;
7     }
8 }
```

8. Verify your results.

Apex Test Result

[Help for this Page](#) 

Summary

Test Class	TestObject
Tests Run	1
Test Failures	0
Code Coverage Total %	0
Total Time (ms)	83.0

Test Successes

Method Name	Total Time (ms)
TestObject.Test CustomWorkflowObject	83.0

Code Coverage

 Class Code Coverage

Class Name	Coverage %
ActivityController	0
DefinitionExtension	0
DefinitionVisualController	0
DependentFieldExtension	0
GoogleViz	0
JSONObject	0
ProcessComposerExecution	0
StepExtension	0
startHereController	0
XMLDom	0

Test Coverage Warnings

Message
Average test coverage across all Apex Classes and Triggers is 0%, at least 75% test coverage is required.

Index

A

Apex Trigger 11

C

Class 14

Configuration

 Security Profile 7

Creating

 Apex Trigger 11

 Test Class 14

G

Grant Access to All Users 7

I

Installation

 Installing ProcessComposer 5

P

Profile

 Security Configuration 7

S

Security

 Grant Access to All Users 7

 Security Profile Configuration 7

T

Test

 Test Class 14

 Trigger 11

U

Updating

 Grant Access to All Users 7