

The Eclipse IDE for Salesforce Admins

1. Introduction
 - a. How It Works
2. Installation
 - a. Java Installation
 - b. Eclipse Installation
 - c. Windows installation
 - d. Mac installation
 - e. Starting Eclipse for the First Time
 - f. Force.com Plug-In Installation
3. Creating a Salesforce Project In Eclipse
4. Searching the Project
 - a. Use Case: Where is a custom field used?
5. Move Reports to Another Folder
6. Copying Fields to Another Salesforce Instance
7. Refreshing the Project
8. What Eclipse Cannot Do
 - a. Cannot Delete Items
 - b. Cannot Change API Names
9. Appendix A: Identifying Salesforce API Names
 - a. Custom Object
 - b. Custom Field
 - c. Report
 - d. Custom Report Type
 - e. Workflow
 - f. Email Template
 - g. Page Layout
10. Appendix B: Editing the Eclipse.ini File

1. Introduction

The Eclipse Integrated Development Environment (IDE) is an application used by developers to write Apex / Visualforce / Lightning code in Salesforce. However, it can be also used by Salesforce administrators for various tasks such as:

- Find out all the places where a custom field / object is used: Apex code, reports, custom report types, list views, page layouts, workflow rules and actions, and email templates
- Search for all reports that use a specific custom report type
- Mass-move reports from one folder to another

- Copy an object's custom fields from one organization to another

Eclipse runs on Windows, Mac and Linux machines. This document will explain how to install and configure Eclipse on Windows and Mac computers and is intended for experienced Windows / Mac users who are comfortable with Windows Explorer or OS X Finder, installing applications, and working with Zip or DMG files. Also, you should be able to understand how to use Salesforce sandboxes and how to obtain Salesforce security tokens. No programming experience is required.

a. How It Works

The IDE is basically a text editor that allows you to view and edit Salesforce metadata. Metadata is a text representation of the Salesforce item, whether it is a field, object, report, etc. It is formatted in Extensible Markup Language (XML) format.

XML looks similar to HTML in its use of tags, which are keywords enclosed in angled brackets. For example, here is a custom field in XML format. Note the various tags like `<fields>`, `<fullName>`, etc.

```
<fields>
  <fullName>Custom_Field_01__c</fullName>
  <externalId>>false</externalId>
  <inlineHelpText>Custom field 01</inlineHelpText>
  <label>Custom Field 01</label>
  <precision>18</precision>
  <required>>false</required>
  <scale>0</scale>
  <trackFeedHistory>>false</trackFeedHistory>
  <trackTrending>>false</trackTrending>
  <type>Number</type>
  <unique>>false</unique>
</fields>
```

2. Installation

Go to the Salesforce Developer website to review the latest information about the installation process.

https://developer.salesforce.com/page/Force.com_IDE_Installation

a. Java Installation

Before installing Eclipse, you must install Java Standard Edition Development Kit (Java SE JDK) from Oracle Corporation. As of September 2015, the Salesforce web page advises using Java SE JDK 7.x. However, Oracle no longer supports this version, so you must use the current Java 8.x version. Go to this page to download and install this package.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

If you're running 64-bit Windows, select the Windows X64 download file. If you're not sure, select the Windows x86 download file.

Java Platform, Standard Edition

Java SE 8u60
This releases includes support for ARMv8 processors, Nashorn enhancements, and improvements to Deployment Rule Set functionality.
JDK for ARM releases are now available on the same page as the downloads for other platforms.
[Learn more](#) ▶

- Installation Instructions
- Release Notes

JDK
DOWNLOAD ↓

Download and install the Java SE JDK on your computer.

Java SE Development Kit 8u60		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE ; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u60-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.64 MB	jdk-8u60-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.66 MB	jdk-8u60-linux-i586.rpm
Linux x86	174.83 MB	jdk-8u60-linux-i586.tar.gz
Linux x64	152.67 MB	jdk-8u60-linux-x64.rpm
Linux x64	172.84 MB	jdk-8u60-linux-x64.tar.gz
Mac OS X x64	227.07 MB	jdk-8u60-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.67 MB	jdk-8u60-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.02 MB	jdk-8u60-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.18 MB	jdk-8u60-solaris-x64.tar.Z
Solaris x64	96.71 MB	jdk-8u60-solaris-x64.tar.gz
Windows x86	180.82 MB	jdk-8u60-windows-i586.exe
Windows x64	186.16 MB	jdk-8u60-windows-x64.exe

b. Eclipse Installation

The Salesforce developer page will state the latest version of Eclipse to download and install. As of September 2015, use the Kepler 4.3 version. On the Eclipse website, select Eclipse IDE for Java Developers. Your 32/64 bit choice of Eclipse must match the 32/64 bit version of Java that you just installed.

If you have 64-bit Windows and you installed 64-bit Java SE JDK, you must select the 64-bit version of Eclipse. If you're not sure and you installed 32-bit Java, select 32 bit Eclipse.

If you have Mac OS X, select Mac Cocoa 64-bit:



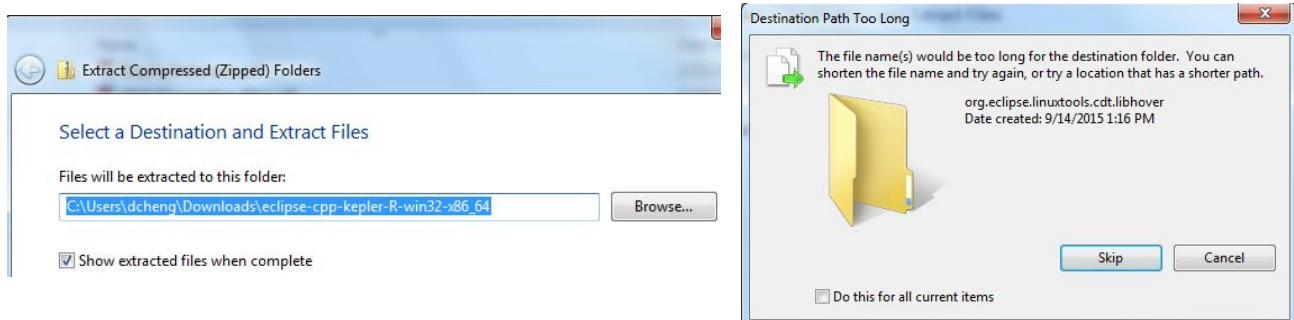
Windows **32-bit 64-bit**
Mac Cocoa **32-bit 64-bit**
Linux **32-bit 64-bit**

c. Windows installation

Eclipse does not have a setup program or process. Instead, you can open the zip file and drag the "eclipse" folder to the desired location on your computer. It does not have to be placed in the Program Files folder, and it will not appear in the Control Panel > Programs and Features list.

You can also extract the "eclipse" folder from the Zip file by right-clicking on the Zip file and choosing Extract All. However, you may run into an error "the file name(s) would be too long for the destination folder" due to the length of the base folder

Example: a base folder of this length will result in an error during extraction.

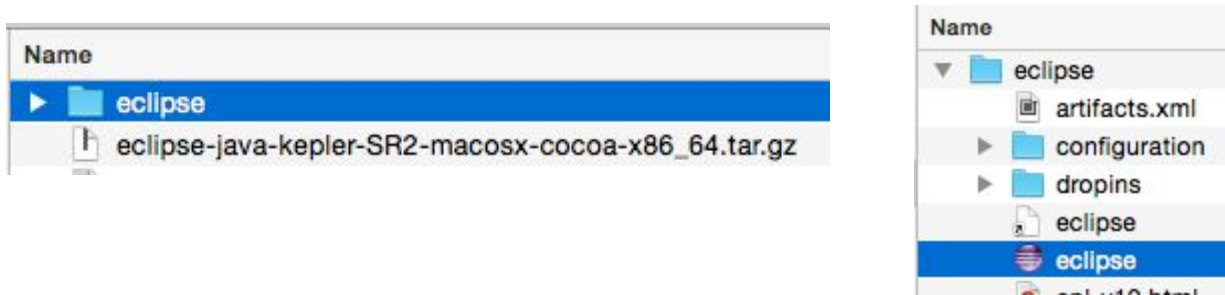


Select a shorter base folder. Once you extracted the "eclipse" folder, you can double-click on the Eclipse.exe to run the application.

d. Mac installation

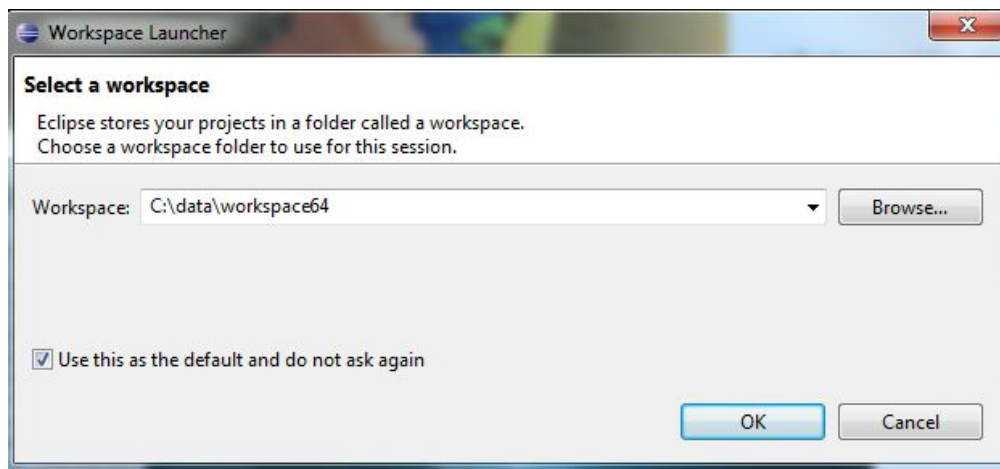
The Eclipse file for OS X is in "tar" format and ends in "tar.gz". Double-click the file to extract the "eclipse" folder from it.

You can then drag the "eclipse" folder into your Applications folder. To run Eclipse, double-click the eclipse application (with the globe icon).



e. Starting Eclipse for the First Time

When you start Eclipse for the first time, you'll be prompted for your workspace. This is a folder on your computer where Eclipse will store your project files. Use the default folder or select another folder. Be sure to tick the "use this as the default and do not ask again" checkbox. Click the OK button when done.



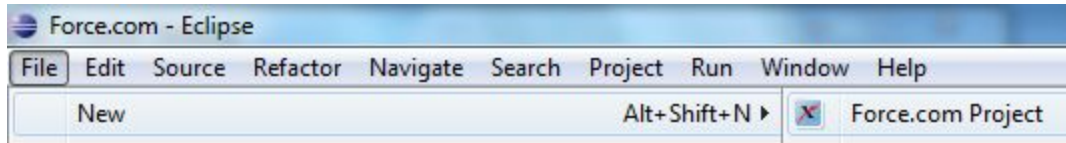
f. Force.com Plug-In Installation

Once you've started Eclipse the first time, follow the instructions on the Salesforce developer page (there's an Installation Steps section) to install the Force.com plug-in.

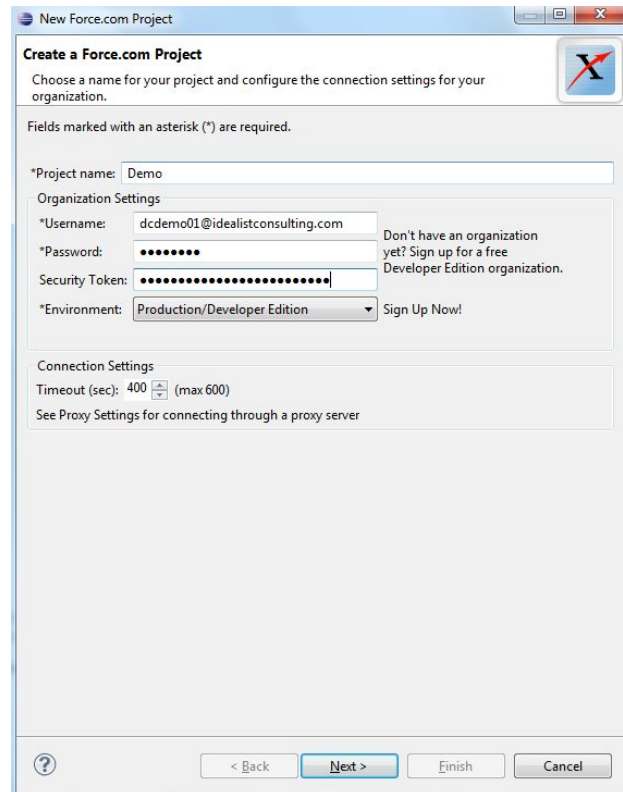
https://developer.salesforce.com/page/Force.com_IDE_Installation

3. Creating a Salesforce Project In Eclipse

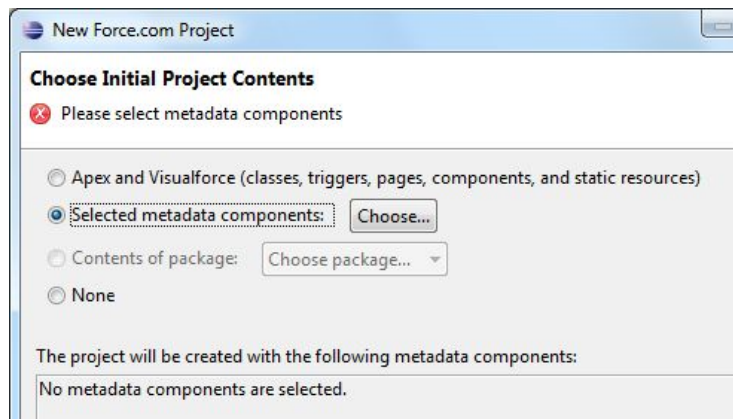
Once the Force.com plug-in is installed, you can create your Eclipse project for your Salesforce instance. On the main menu, choose File > New > Force.com Project.



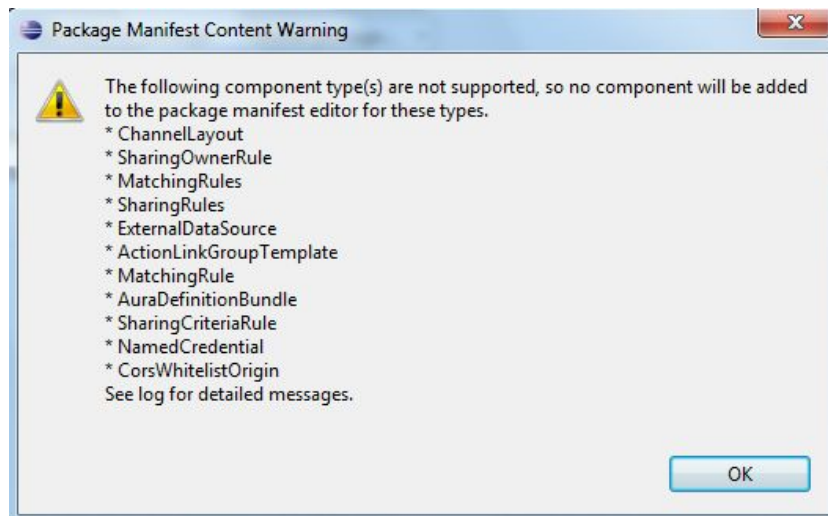
You'll be prompted for your Salesforce credentials. Generally you should connect to your Salesforce sandbox, not to your Salesforce production instance. (For this document, we are using a Developer Edition instance, so the Environment is set to Production/Developer Edition.) Click the Next button to continue.



On the Choose Initial Project Contents screen, choose "Selected metadata components" and click the Choose button.

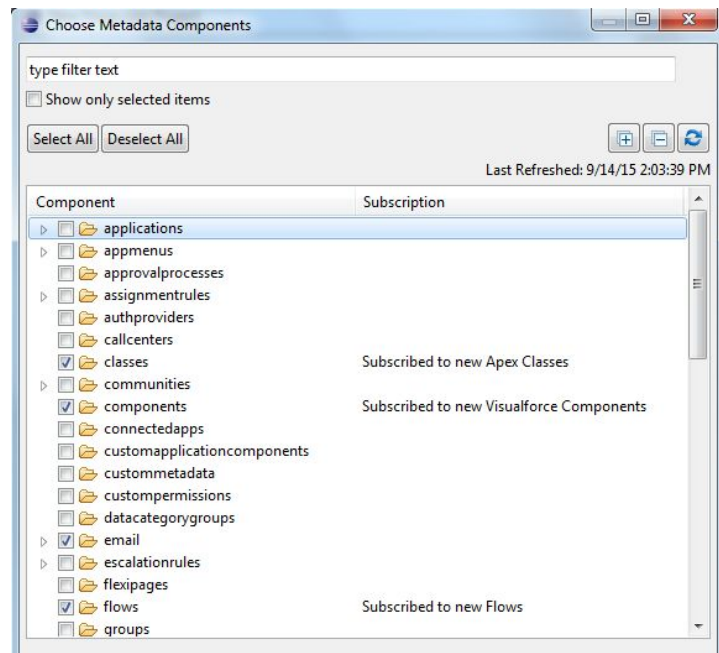


You may see a Package Manifest Content Warning about component types that are not supported. In most cases you can ignore this message. Click the OK button.

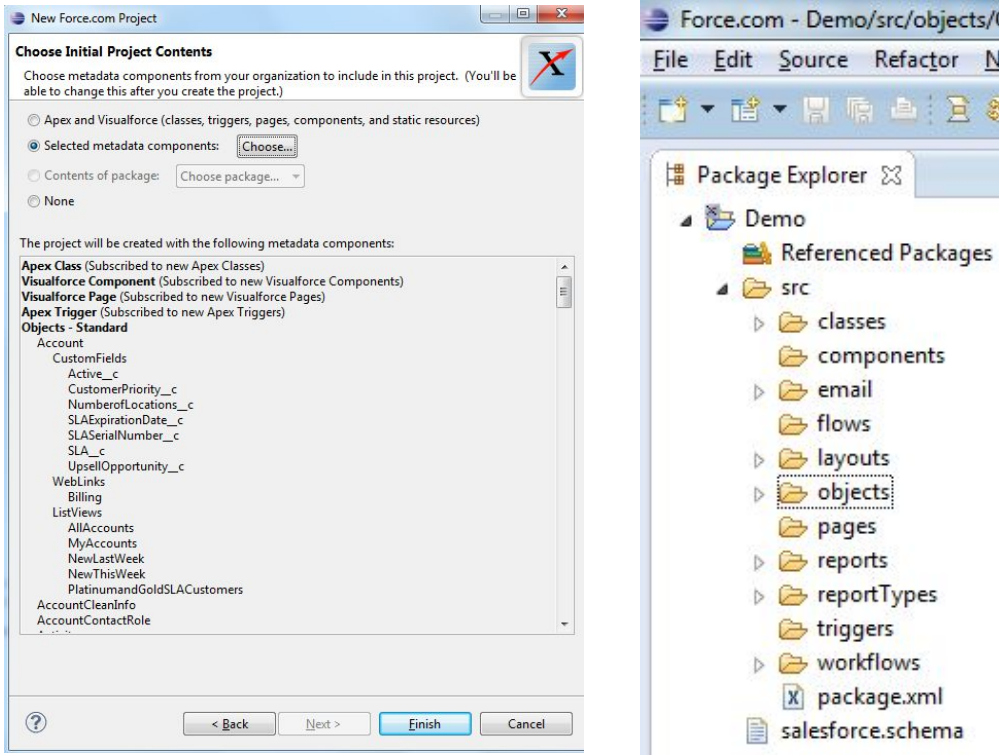


On the Choose Metadata Components screen, select the following:

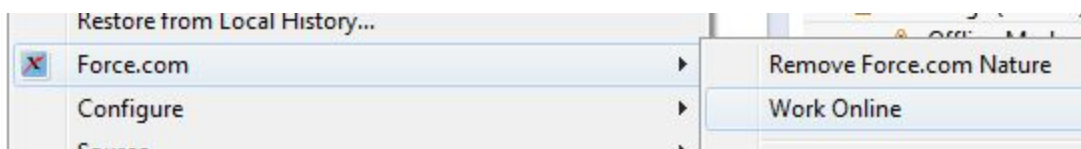
- components
- classes
- email
- flows
- layouts
- objects - standard
- objects - custom
- pages
- reports
- report types
- triggers
- workflows



Click the Finish button to continue. Then you'll see the Salesforce items in the Eclipse Package Explorer.



Your project may default to Work Offline status. To switch to Work Online status, right-click on the project name and choose Force.com > Work Online.



You can expand the folders to see the contents. For example: here are the custom reports that reside in the report folders. Also displayed are the custom report types that have been created in this instance.



Here are the reports and report type as they appear in Salesforce:

Reports & Dashboards

[New Report...](#) [New Dashboard...](#)

Folders

Find a folder...

All Folders

- Unfiled Public Reports
- My Personal Custom Reports
- My Personal Dashboards
- Demo Reports

Unfiled Public Reports

Find reports and dashboards...

Action	Name ↑
▼	Demo Accounts
▼	demo opp report

Reports & Dashboards

[New Report...](#) [New Dashboard...](#)

Folders

Find a folder...

All Folders

- Unfiled Public Reports
- My Personal Custom Reports
- My Personal Dashboards
- Demo Reports

Demo Reports

Find reports and dashboards...

Action	Name ↑
▼	Opps with Contact Roles
▼	Opps with or without contact roles

All Custom Report Types

With custom report types, you can enable users to create reports from the predefined

View: [All Custom Report Types](#) ▼ [Edit](#) | [Create New View](#)

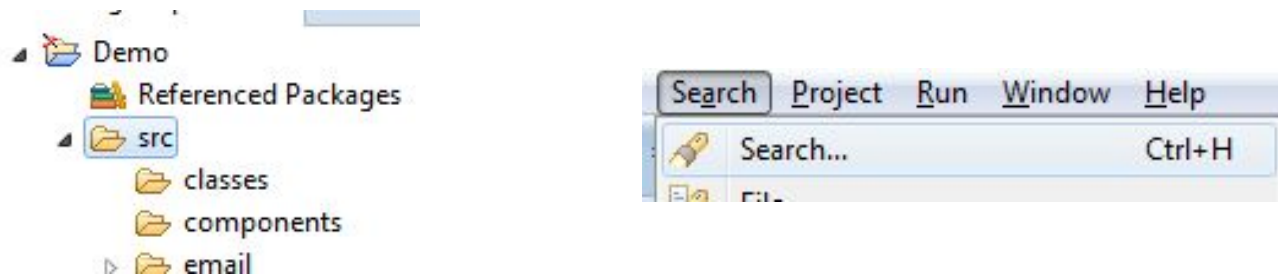
[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#)

New Custom Report Type		
Action	Label ↑	Description
Edit Del	Opps with or without contact roles	Opps with or without contact roles

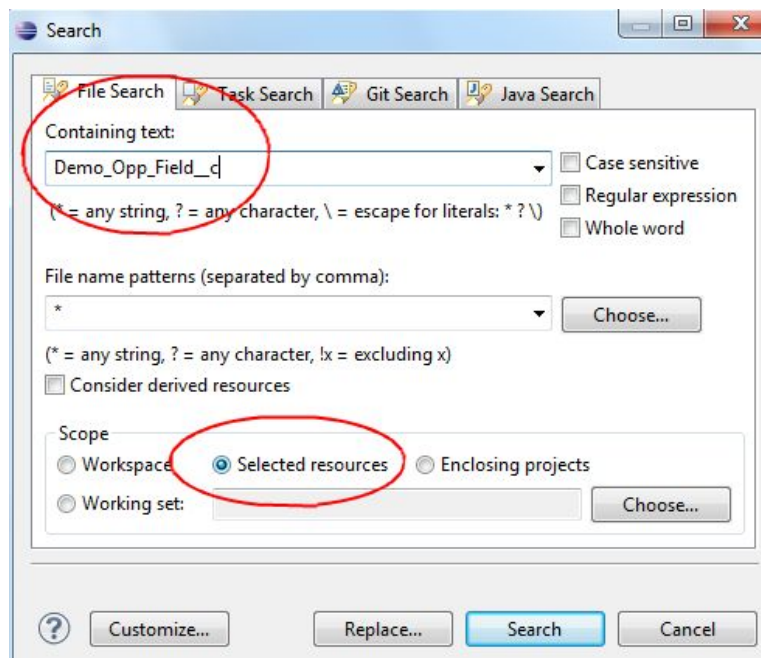
4. Searching the Project

Eclipse is most useful for searching for the name of a custom object, field, report, or report type to see where it is referenced. It is best to use the API name of the object / field / report etc when searching. See Appendix A on where to find API names.

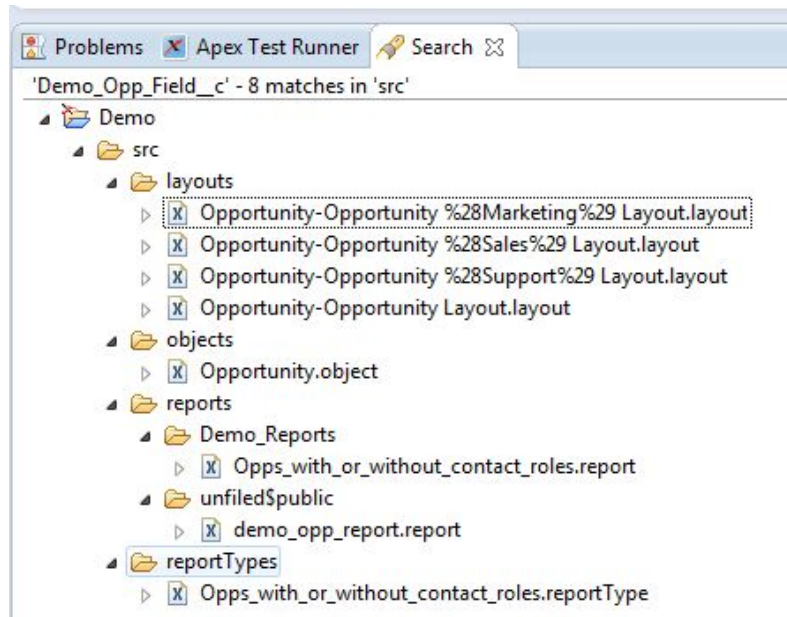
First, select the src folder in the Package Explorer, then choose Search > Search ... from the main menu. (Hint: you can also press Ctrl-H).



In the search window, make sure the File Search tab is selected. Enter the item name in the Containing text field. Make sure the Scope is "selected resources" (i.e., the src folder you just selected), then click the Search button.



The Search tab will display the results. Here you can see that the field is used in several layouts, reports and report types.



If you double-click one of the items, you'll see the XML metadata. Here is a report metadata file that contains the field. The field is displayed as a column on the report.



a. Use Case: Where is a custom field used?

If you have a custom field and you're not sure where it's used, you can search in Eclipse to see if it appears in any reports, workflows, layouts, Apex code, etc.

Here are some examples of how a custom field `Demo_Opp_Field__c` appears in various areas.

Within the object:

```
Opportunity.object
108     <label>Demo Formula Field</label>
109     <required>false</required>
110     <trackTrending>false</trackTrending>
111     <type>Text</type>
112     <unique>false</unique>
113     </fields>
114 </fields>
115 <fullName>Demo Opp Field c</fullName>
116 <externalId>false</externalId>
117 <label>Demo Opp Field</label>
118 <length>255</length>
119 <required>false</required>
120 <trackFeedHistory>false</trackFeedHistory>
121 <trackTrending>false</trackTrending>
122 <type>Text</type>
123 <unique>false</unique>
124 </fields>
```

In a formula field named "Demo Formula Field":

```
Opportunity.object
99     <trackTrending>false</trackTrending>
100     <type>Number</type>
101     <unique>false</unique>
102     </fields>
103 </fields>
104 <fullName>Demo Formula Field c</fullName>
105 <externalId>false</externalId>
106 <formula>LEFT(Demo Opp Field c, 10)</formula>
107 <formulaTreatBlanksAs>BlankAsZero</formulaTreatBlanksAs>
108 <label>Demo Formula Field</label>
109 <required>false</required>
110 <trackTrending>false</trackTrending>
111 <type>Text</type>
112 <unique>false</unique>
113 </fields>
```

On a page layout:

```
Opportunity-Opportunity Layout.layout
31     </layoutItems>
32 </layoutItems>
33     <behavior>Edit</behavior>
34     <field>LeadSource</field>
35 </layoutItems>
36 </layoutItems>
37     <behavior>Edit</behavior>
38     <field>Demo_Opp_Field__c</field>
39 </layoutItems>
```

In a listview (the "All Opportunities" list view for Opportunity):

```
Opportunity.object
323     </fields>
324     <listViews>
325         <fullName>AllOpportunities</fullName>
326         <columns>OPPORTUNITY.NAME</columns>
327         <columns>OPPORTUNITY.TYPE</columns>
328         <columns>OPPORTUNITY.STAGE_NAME</columns>
329         <columns>OPPORTUNITY.CLOSE_DATE</columns>
330         <columns>Demo Opp Field c</columns>
331         <filterScope>Everything</filterScope>
332         <label>All Opportunities</label>
333     </listViews>
```

A column in a report:

```
Opps_with_Contact_Roles.report
37     <field>ADDRESS2_ZIP</field>
38 </columns>
39 <columns>
40     <field>ADDRESS2_COUNTRY</field>
41 </columns>
42 <filter>
43 <criteriaItems>
44     <column>Opportunity.Demo Opp Field c</column>
45     <operator>equals</operator>
46     <value>Renewal</value>
47 </criteriaItems>
```

A filter condition in a report. The condition is "Demo Opp Field equals Renewal:"

```
Opps_with_or_without_contact_roles.report
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Report xmlns="http://soap.sforce.com/2006/04/metadata">
3     <columns>
4         <field>Opportunity$Name</field>
5     </columns>
6     <columns>
7         <field>Opportunity.OpportunityContactRoles$Name</field>
8     </columns>
9     <columns>
10        <field>Opportunity.OpportunityContactRoles$RoleName</field>
11    </columns>
12    <columns>
13        <field>Opportunity$Demo Opp Field c</field>
14    </columns>
```

The report type (standard or custom) used by a report. In this screenshot, the report type is a custom report type:

```
Demo_Jobs_Report.report
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Report xmlns="http://soap.sforce.com/2006/04/metadata">
3   <columns>
4     <field>Job_c$Name</field>
5   </columns>
6   <columns>
7     <field>Job_c.Job_Applications_r$Name</field>
8   </columns>
9   <columns>
10    <field>Job_c$Department__c</field>
11  </columns>
12  <columns>
13    <field>Job_c.Job_Applications_r$Status__c</field>
14  </columns>
15  <format>Tabular</format>
16  <name>Demo Jobs Report</name>
17  <params>
18    <name>co</name>
19    <value>1</value>
20  </params>
21  <reportType>Jobs_with_Job_Applications__c</reportType>
22  <scope>organization</scope>
23  <showDetails>true</showDetails>
24  <timeFrameFilter>
25    <dateColumn>Job_c$CreatedDate</dateColumn>
26    <interval>INTERVAL_CUSTOM</interval>
27  </timeFrameFilter>
28 </Report>
```

A criterion in an workflow rule.
The criterion is "Demo Opp Field equals Renewal:"

```
Opportunity.workflow
6   <formula>&quot;Continuing&quot;</formula>
7   <name>Demo Opp Field Update</name>
8   <notifyAssignee>false</notifyAssignee>
9   <operation>Formula</operation>
10  <protected>false</protected>
11 </fieldUpdates>
12 <rules>
13   <fullName>Demo Opp rule</fullName>
14   <actions>
15     <name>Demo_Opp_Field_Update</name>
16     <type>FieldUpdate</type>
17   </actions>
18   <active>true</active>
19   <criteriaItems>
20     <field>Opportunity.Demo Opp Field c</field>
21     <operation>equals</operation>
22     <value>Renewal</value>
23   </criteriaItems>
24   <triggerType>onCreateOrTriggeringUpdate</triggerType>
25 </rules>
```

In a workflow field update action. In this action the Demo Opp Field is being set to "Continuing:"

```
Opportunity.workflow
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Workflow xmlns="http://soap.sforce.com/2006/04/metadata">
3   <fieldUpdates>
4     <fullName>Demo Opp Field Update</fullName>
5     <field>Demo Opp Field_c</field>
6     <formula>&quot;Continuing&quot;</formula>
7     <name>Demo Opp Field Update</name>
8     <notifyAssignee>false</notifyAssignee>
9     <operation>Formula</operation>
10    <protected>false</protected>
11  </fieldUpdates>
12 </Workflow>
```

In an email template:

```
SalesNewCustomerEmail.email
1 Dear {!Contact.FirstName},
2
3 All of us at GenWatt are glad to have {!Account.Name} as a customer.
4
5 I would like to introduce myself as your Account Manager. Should you have any questions, please contact me.
6
7 Here is the opportunity field: {!Opportunity.Demo_Opp_Field_c}
8
9 You can also contact GenWatt on the following numbers:
10
```

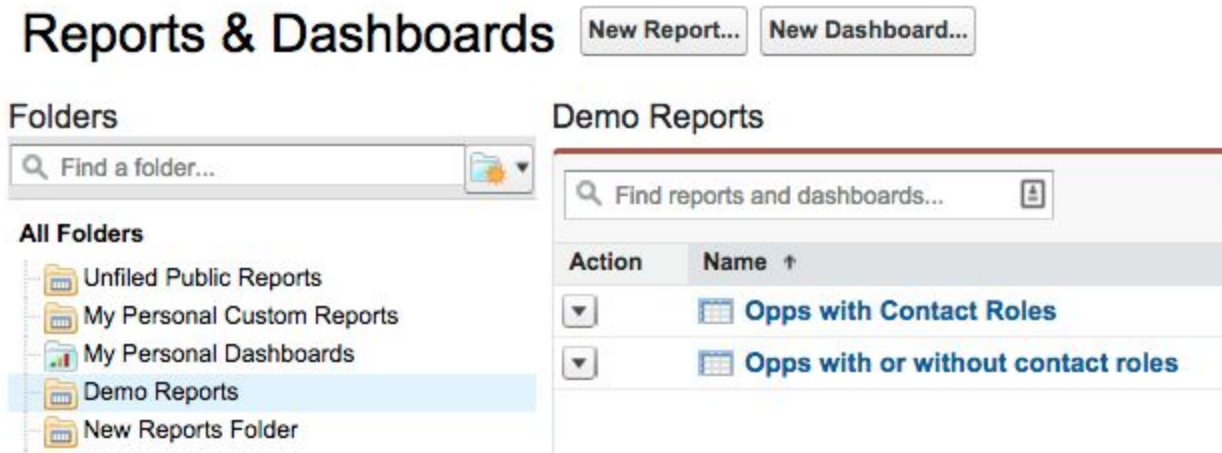
In an Apex class:

```
DemoClass.cls
1 public with sharing class DemoClass {
2   public static void process() {
3     Opportunity opp = [SELECT Demo_Opp_Field_c FROM Opportunity LIMIT 1];
4   }
5 }
```

5. Move Reports to Another Folder

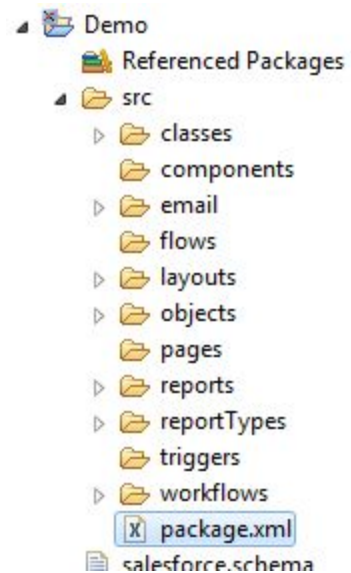
You can use Eclipse to move multiple reports from one folder to another. This is a two-step process that involves editing the package.xml file and also moving the report XML files on your computer's local drive.

For this example, we'll move all reports from the Demo Reports Folder to the New Reports folder.



First, refresh the project's metadata components so you get the complete list of reports and report folders. See the Refreshing the Project section for details on how to do this.

Next, open the package.xml file. This file is an index of all the Salesforce object metadata in the project.



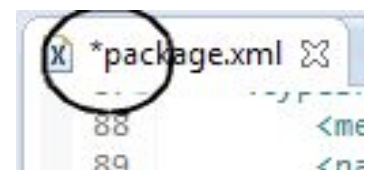
Press Ctrl-F or choose Find to open the Find window. Enter "<name>Report</name>" to find the report folder section.

```
package.xml
89     <name>Layout</name>
90   </types>
91   <types>
92     <members>Demo_Reports</members>
93     <members>Demo_Reports/Opps_with_Contact_Roles</members>
94     <members>Demo_Reports/Opps_with_or_without_contact_roles</members>
95     <members>New_Reports_Folder</members>
96     <members>unfiled$public</members>
97     <members>unfiled$public/Demo_Accounts</members>
98     <members>unfiled$public/demo_opp_report</members>
99     <name>Report</name>
100  </types>
```

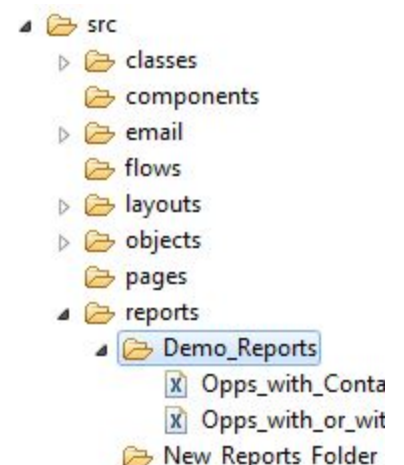
Move the reports to below New_Reports_Folder, then change the prefix from Demo_Reports to New_Reports_Folder for each report.

```
91   <types>
92     <members>Demo_Reports</members>
93     <members>New_Reports_Folder</members>
94     <members>New_Reports_Folder/Opps_with_Contact_Roles</members>
95     <members>New_Reports_Folder/Opps_with_or_without_contact_roles</members>
96     <members>unfiled$public</members>
97     <members>unfiled$public/Demo_Accounts</members>
98     <members>unfiled$public/demo_opp_report</members>
99     <name>Report</name>
100  </types>
```

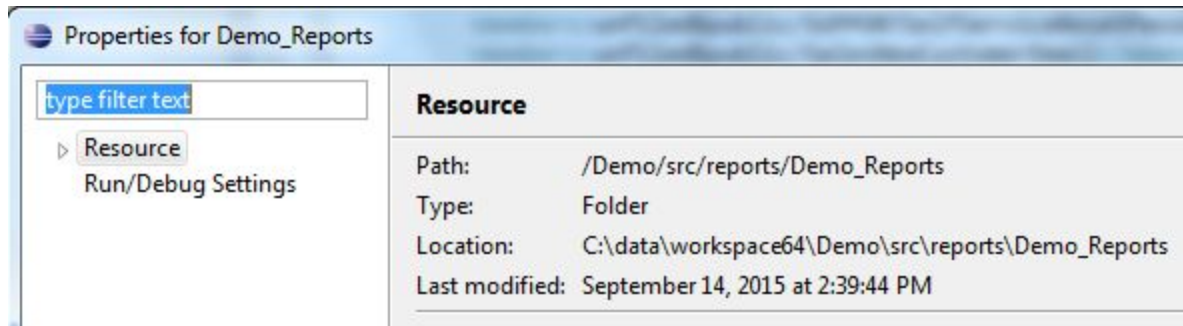
The file window title will indicate that a change has been made, so press Ctrl-S (or choose File > Save) to save.



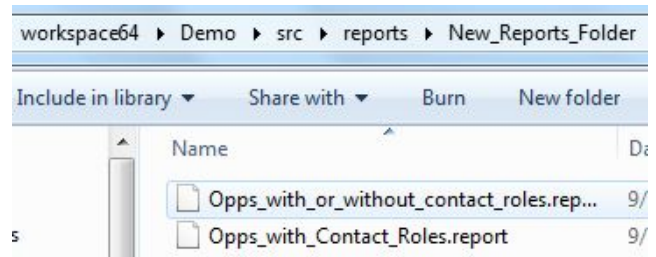
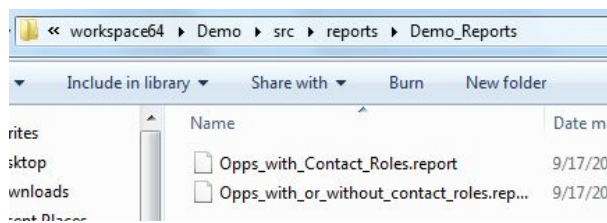
Now you must move the report XML files on your computer's local drive. In the Package Explorer, right-click the Demo Reports folder.



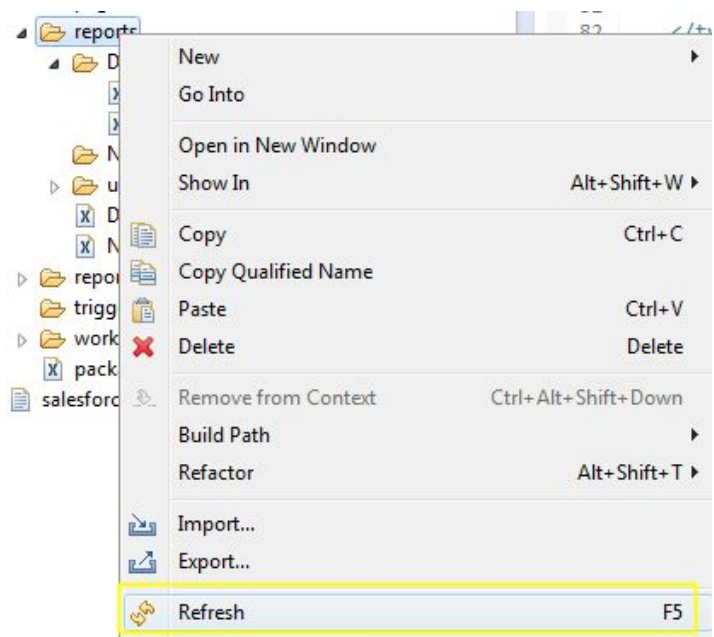
In the Properties window, make a note of the Location.



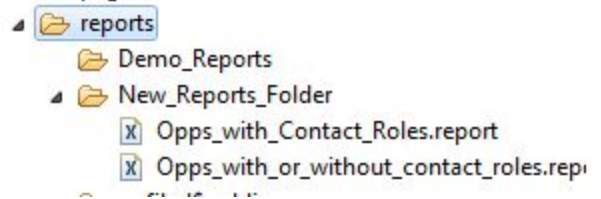
In Windows Explorer (or Mac Finder), navigate to that folder, then move the report files to the new folder. In this example, that is New_Report_Folder.



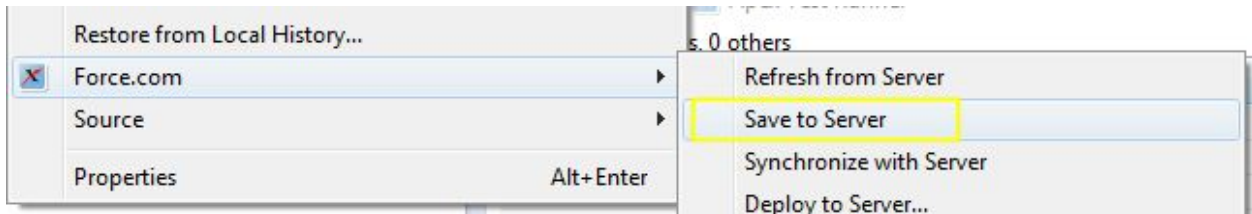
Next, you must refresh the reports folder in Package Builder. Right click on reports and select Refresh. [Note: do not choose Force.com > Refresh from Server.]



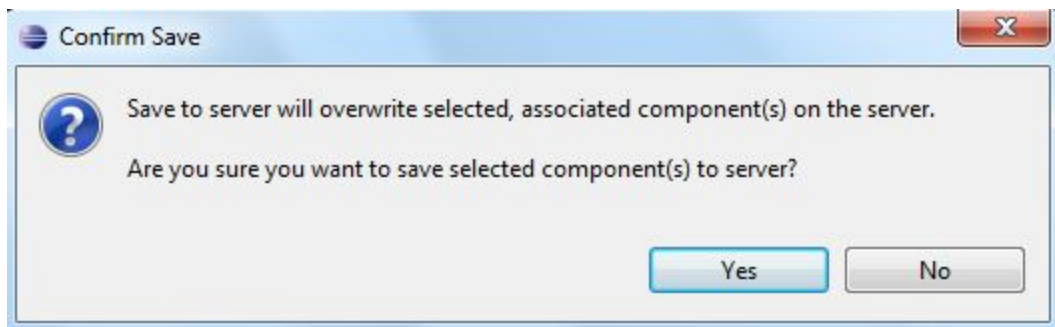
You'll see the report folders' contents updated in the Package Explorer.



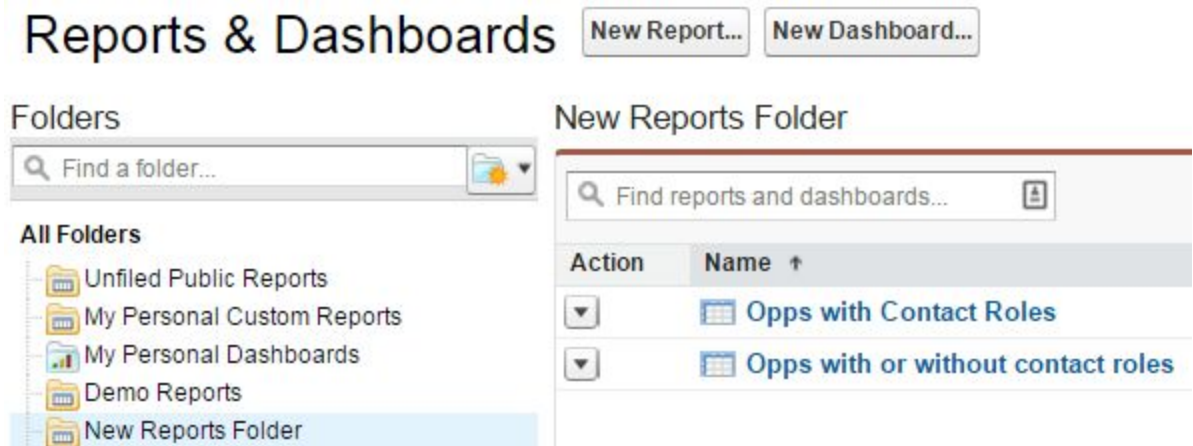
Finally, right-click on the reports folder and choose Force.com > Save to Server.



Click Yes to confirm.



In Salesforce, the reports are now in the New Reports Folder.



6. Copying Fields to Another Salesforce Instance

If you have a large number of fields that you want to copy to another organization, you can't use Change Sets because they can only be used among production and sandbox instances within the same organization.

However, you can copy the fields' XML metadata from your instance and paste them into the other organization's sandbox instance.

For this example, say we want to copy these two opportunity custom fields to a different organization.

Action	Field Label	API Name	Data Type
Edit Del	<u>Current Generator(s)</u>	CurrentGenerators__c	Text(100)
Edit Del	<u>Custom Field 01</u>	Custom_Field_01__c	Number(18, 0)
Edit Del	<u>Custom Field 02</u>	Custom_Field_02__c	Number(18, 0)

First, create a sandbox and Eclipse project for the target organization.

In Eclipse, open the source organization's project and in the Package Explorer, open the Opportunity object XML file.

- objects
 - Account.object
 - AccountCleanInfo.object
 - AccountContactRole.object
 - Activity.object
 - Asset.object
 - Campaign.object
 - CampaignMember.object
 - Case.object
 - CaseContactRole.object
 - Contact.object
 - ContactCleanInfo.object
 - ContentVersion.object
 - Contract.object
 - ContractContactRole.object
 - DandBCompany.object
 - DuplicateRecordItem.object
 - DuplicateRecordSet.object
 - Event.object
 - FeedItem.object
 - ForecastingCategoryMapping.object
 - Idea.object
 - Lead.object
 - LeadCleanInfo.object
 - Macro.object
 - MacroAction.object
 - MacroInstruction.object
 - Opportunity.object**
 - OpportunityCompetitor.object

Scroll through the file until you find the <fields> tag of the first field you wish to copy. Select all the lines for the fields you want to copy. Be sure to include the <fields> and </fields> lines. Copy and paste your selections into another text editor like Notepad, Wordpad, TextEdit, etc.



```
77     </fields>
78     <fields>
79         <fullName>CurrentGenerators__c</fullName>
80         <externalId>>false</externalId>
81         <label>Current Generator(s)</label>
82         <length>100</length>
83         <required>>false</required>
84         <trackFeedHistory>>false</trackFeedHistory>
85         <trackTrending>>false</trackTrending>
86         <type>Text</type>
87         <unique>>false</unique>
88     </fields>
89     <fields>
90         <fullName>Custom_Field_01__c</fullName>
91         <externalId>>false</externalId>
92         <inlineHelpText>Custom field 01</inlineHelpText>
93         <label>Custom Field 01</label>
94         <precision>18</precision>
95         <required>>false</required>
96         <scale>0</scale>
97         <trackFeedHistory>>false</trackFeedHistory>
98         <trackTrending>>false</trackTrending>
99         <type>Number</type>
100        <unique>>false</unique>
101    </fields>
102    <fields>
103        <fullName>Custom_Field_02__c</fullName>
104        <externalId>>false</externalId>
105        <label>Custom Field 02</label>
106        <precision>18</precision>
107        <required>>false</required>
108        <scale>0</scale>
109        <trackFeedHistory>>false</trackFeedHistory>
110        <trackTrending>>false</trackTrending>
111        <type>Number</type>
112        <unique>>false</unique>
113    </fields>
114    <fields>
115        <fullName>DeliveryInstallationStatus__c</fullName>
116        <externalId>>false</externalId>
```

Repeat this process until you've collected all the fields' XML data into your text editor. Now open the target organization's project, and open the target object's XML file.

Press Ctrl-F and find the first <fields> tag, then paste the new field XML data at this point. You do not have to insert the new fields in alphabetical order -- Salesforce will take care of that after the save.

```
46     </actionOverrides>
47     <compactLayoutAssignment>SYSTEM</compactLayoutAssignment>
48     <enableFeeds>true</enableFeeds>
49     <enableHistory>>false</enableHistory>
50     <fields>
51         <fullName>Custom_Field_01_c</fullName>
52         <externalId>>false</externalId>
53         <inlineHelpText>Custom field 01</inlineHelpText>
54         <label>Custom Field 01</label>
55         <precision>18</precision>
56         <required>>false</required>
57         <scale>0</scale>
58         <trackFeedHistory>>false</trackFeedHistory>
59         <trackTrending>>false</trackTrending>
60         <type>Number</type>
61         <unique>>false</unique>
62     </fields>
63     <fields>
64         <fullName>Custom_Field_02_c</fullName>
65         <externalId>>false</externalId>
66         <label>Custom Field 02</label>
67         <precision>18</precision>
68         <required>>false</required>
69         <scale>0</scale>
70         <trackFeedHistory>>false</trackFeedHistory>
71         <trackTrending>>false</trackTrending>
72         <type>Number</type>
73         <unique>>false</unique>
74     </fields>
75     <fields>
76         <fullName>AccountId</fullName>
77         <trackFeedHistory>>false</trackFeedHistory>
78         <trackTrending>>false</trackTrending>
79         <type>Lookup</type>
80     </fields>
81     <fields>
82         <fullName>Amount</fullName>
83         <trackFeedHistory>>false</trackFeedHistory>
```

Save the file. This should save to Salesforce and create the field(s). The XML file will automatically refresh and the fields will be in alphabetical order.

```
73     <fullName>ContractId</fullName>
74     <trackFeedHistory>>false</trackFeedHistory>
75     <trackTrending>>false</trackTrending>
76     <type>Lookup</type>
77 </fields>
78 <fields>
79     <fullName>Custom_Field_01__c</fullName>
80     <externalId>>false</externalId>
81     <inlineHelpText>Custom field 01</inlineHelpText>
82     <label>Custom Field 01</label>
83     <precision>18</precision>
84     <required>>false</required>
85     <scale>0</scale>
86     <trackFeedHistory>>false</trackFeedHistory>
87     <trackTrending>>false</trackTrending>
88     <type>Number</type>
89     <unique>>false</unique>
90 </fields>
91 <fields>
92     <fullName>Custom_Field_02__c</fullName>
93     <externalId>>false</externalId>
94     <label>Custom Field 02</label>
95     <precision>18</precision>
96     <required>>false</required>
97     <scale>0</scale>
98     <trackFeedHistory>>false</trackFeedHistory>
99     <trackTrending>>false</trackTrending>
100    <type>Number</type>
101    <unique>>false</unique>
102 </fields>
103 <fields>
104     <fullName>Demo_Opp_Field__c</fullName>
105     <externalId>>false</externalId>
```

Now you must set the Field Level Security for each field. You can do this for each field or for multiple fields in the user profile.

Opportunity Custom Field
Custom Field 01
[Back to Opportunity Fields](#) [Validation Rules \[](#)

Custom Field Definition Detail

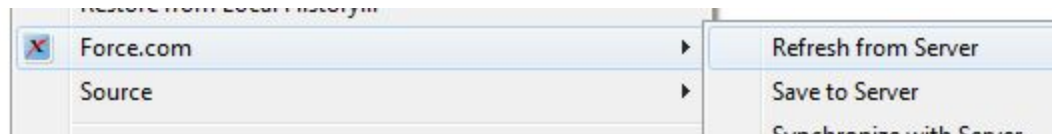
Field Information

Field Label	Custom Field 01
Field Name	Custom_Field_01
API Name	Custom_Field_01__c

7. Refreshing the Project

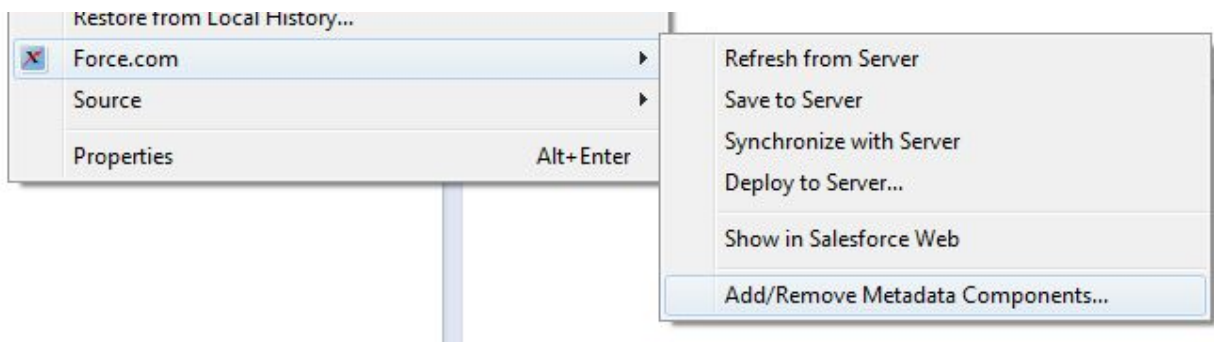
If you made a change to an existing Salesforce item -- changing a report filter, modifying picklist values, etc -- then you need to refresh from the server.

In the Package Explorer, right-click the src folder and choose Force.com > Refresh from Server.

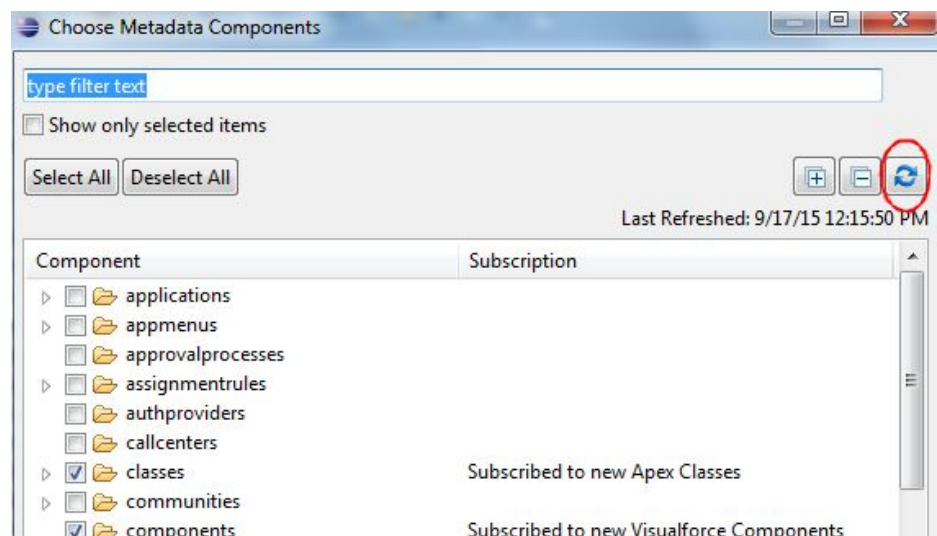


If you create a new item in the Salesforce sandbox -- custom object, custom field, layout, report, etc -- then you must refresh the project's metadata components and add the items to the project.

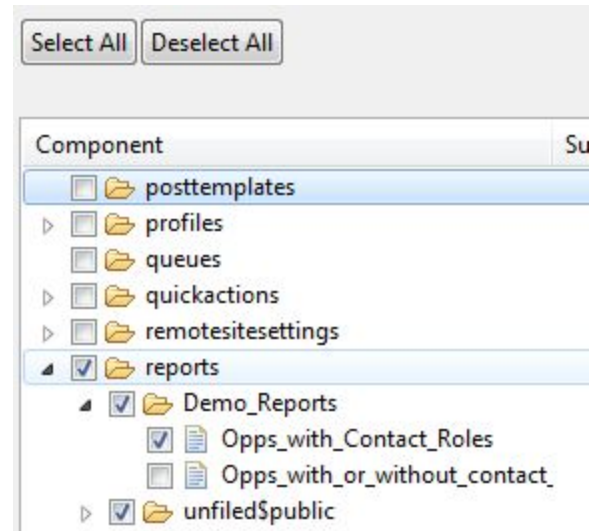
Right-click the src folder, then select Force.com > Add/Remove Metadata components.



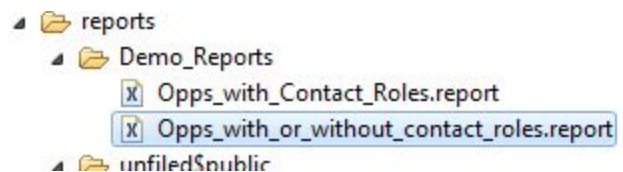
You'll see the Choose Metadata Components screen. Click the refresh icon in the top right.



After refreshing, expand the component where you added the item in Salesforce. In this example, the Opps_with_or_without_contact report was created in Salesforce. Tick the checkbox to add it to the project, then click the Finish button.



You should see the new item in the Package Explorer.



8. What Eclipse Cannot Do

a. Cannot Delete Items

You cannot use Eclipse to delete fields, objects, reports, or report types. You must use the Salesforce Setup menu to delete these items.

b. Cannot Change API Names

You cannot use Eclipse to "rename" an API name. If you change an API name and save the XML, Eclipse will create a new item with that API name in Salesforce, and the original item will remain in Salesforce.

You can, however, change non-API names or non-unique names such as field labels, report display names, etc.

9. Appendix A: Identifying Salesforce API Names

a. Custom Object

Go to Setup > Create > Objects and click on the object name.

Custom Object
Job Application

[Standard Fields \[4\]](#) | [Custom Fields & Relationships \[8\]](#) | [Buttons, Links, and Actions \[8\]](#) | [Records](#)

Custom Object Definition Detail

Singular Label	Job Application
Plural Label	Job Applications
Object Name	Job_Application
API Name	Job_Application__c

b. Custom Field

Opportunity Custom Fields & Relationships

[New](#) [Field Dependencies](#)

Action	Field Label	API Name	Data Type
Edit Del	Current Generator(s)	CurrentGenerators__c	Text(100)
Edit Del Replace	Delivery/Installation Status	DeliveryInstallationStatus__c	Picklist
Edit Del	Demo Opp Field	Demo_Opp_Field__c	Text(255)
Edit Del	Main Competitor(s)	MainCompetitors__c	Text(100)
Edit Del	Order Number	OrderNumber__c	Text(8)
Edit Del	Tracking Number	TrackingNumber__c	Text(12)

c. Report

Open the report, click Customize > Report Properties and view the Report Unique Name field.

Report Properties

[Help for this](#)

Report Type: Opportunities with Contact Roles

Report Name: Opps with Contact Roles

Report Unique Name: Opps_with_Contact_Roles

Report Description:

d. Custom Report Type

Go to Setup > Create > Report Types and click the custom report type name.

Custom Report Type

Opps with or without contact roles

[« Back to List: Custom Report Types](#)

Below is the information for this custom report type. You can click

Custom Report Type Definition

[Edit](#)

[Delete](#)

Report Type Label	Opps with or without contact roles
Report Type Name	Opps_with_or_without_contact_roles
Description	Opps with or without contact roles

e. Workflow

Workflow rules do not have an API name. However, workflow field update actions and email actions do have a unique name that is equivalent to an API name.

Field Update

Demo Opp Field Update

[« Back to List: Workflow Rules](#)

[Rules Using This F](#)

Field Update Detail

[Edit](#)

Name	Demo Opp Field Update
Unique Name	Demo_Opp_Field_Update
Description	

f. Email Template

Go to Setup > Communication Templates > Email Templates and open an email template. The unique name is the equivalent of an API name.

Text Email Template

Marketing: Product Inquiry Response

[« Back to List: Email Templates](#)

Preview your email template below.

Email Template Detail

[Edit](#)

[Delete](#)

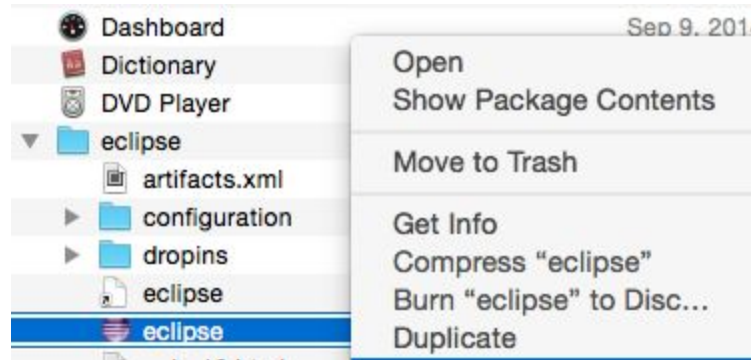
Folder	Unfiled Public Email Templates
Email Template Name	Marketing: Product Inquiry Response
Template Unique Name	MarketingProductInquiryResponse

g. Page Layout

Page layouts do not have an API name.

10. Appendix B: Editing the Eclipse.ini File

If you are running Eclipse on Mac and you need to edit the Eclipse.ini file, you must open the package contents of the application file. In the Applications folder, right-click on the eclipse application file and choose Show Package Contents.



This will reveal a Contents folder which you can then navigate through to get to the eclipse.ini file.

