ISVforce Workbook

Last updated: June 14, 2014

Table of Contents

ISVforce Workbook

The ISVforce workbook is a quick introduction to developing and distributing apps on the Force.com platform. The workbook takes you through the process of signing up for organizations, developing a simple application, and packaging it for distribution. You'll then install and test a beta version of the app, and finally you'll create a released version of your app, so that anyone can install it in their Salesforce organization.

Note: This workbook streamlines the ISVforce lifecycle so that you can complete it in an hour or two. Therefore, some real-life steps are missing, such as signing up for the Salesforce Partner Program, licensing your app, and additional options such as creating a free trial. However, the basic application lifecycle remains the same. When you've completed this workbook and ready to create your killer app, see the Quick Start in the ISVforce Guide, which takes you through the full process.

Supported Browsers

Browser	Comments
Microsoft® Internet Explorer® versions 7, 8, 9, 10, and 11	If you use Internet Explorer, we recommend using the latest version that Salesforce supports. Apply all Microsoft software updates. Note these restrictions. • The full Salesforce site is not supported in Internet Explorer on touch-enabled devices for Windows. Use the Salesforce1 mobile browser app instead. • The HTML solution editor in Internet Explorer 11 is not supported in Salesforce Knowledge. • The Compatibility View feature in Internet Explorer isn't supported. • The Metro version of Internet Explorer 10 isn't supported. • Internet Explorer 6 and 7 aren't supported for login hints for multiple accounts. • Internet Explorer 7 and 8 aren't supported for the Developer Console or the Data Import Wizard. • Internet Explorer 7 isn't supported for Open CTI. • Internet Explorer 7 and 11 aren't supported for Salesforce CRM Call Center built with CTI Toolkit version 4.0 or higher. • Internet Explorer 7 isn't supported for Force.com Canvas. • Internet Explorer 7 isn't supported for Salesforce console features that require more advanced browser performance and recent Web technologies. The console features not available in Internet Explorer 7 include: ◊ The Most Recent Tabs component ◊ Multiple custom console components on sidebars ◊ Multi-monitor components ◊ The resizable highlights panel ◊ The full-width feed option on feed-based page layouts • Internet Explorer 7 and 8 aren't supported for Community Templates for Self-Service. • Community Templates for Self-Service supports Internet Explorer 9 and above for desktop users and Internet Explorer 11 and above for mobile users.

Browser	Comments
	For configuration recommendations, see “Configuring Internet Explorer” in the Salesforce Help.
Mozilla® Firefox®, most recent stable version	Salesforce.com makes every effort to test and support the most recent version of Firefox. • Mozilla Firefox is supported for desktop users only for Community Templates for Self-Service. For configuration recommendations, see “Configuring Firefox” in the Salesforce Help.
Google Chrome™, most recent stable version	Google Chrome applies updates automatically; salesforce.com makes every effort to test and support the most recent version. There are no configuration recommendations for Chrome. Chrome isn’t supported for the Console tab or the **Add Google Doc to Salesforce** browser button.
Apple® Safari® versions 5.x and 6.x on Mac OS X	There are no configuration recommendations for Safari. Apple Safari on iOS isn’t supported for the full Salesforce site. • Safari isn’t supported for the Salesforce console. • Safari isn’t supported for Salesforce CRM Call Center built with CTI Toolkit versions below 4.0.

Tell Me More....

At the end of each step, there is an optional Tell Me More section. If you like to do things quickly, move on to the next step. However, if you're a smell-the-roses type, there's a lot of useful information here.

- For an introduction to developing on Force.com, see http://developer.force.com/workbook.
- To learn more about Force.com and to access a rich set of resources, visit Developer Force at http://developer.force.com.

Tutorial #1: Develop an App

For this workbook you need two organizations, one for development, and one for testing. In a real-world scenario, you would have additional organizations for sales (publishing and licensing), public demos, and possibly many more for development and testing. But let's keep it simple for now, two organizations are enough to show the typical lifecycle. If you're familiar with Salesforce, you know that an organization, or *org* for short, is a cloud unto itself. If you're new to Salesforce, then you can think of an organization as a separate environment where you might develop, test, or publish your app.

After you get your organizations, you'll create a simple Warehouse application to keep track of merchandise. This is a simplified version of the Warehouse application created in the Force.com Workbook.

Step 1: Sign up for Developer Edition

The first Developer Edition organization (or *DE org*, for short) you'll sign up for will be used for developing and packaging your app. Let's call this your "dev org".

1. In your browser go to http://developer.force.com/join.
2. Fill in the fields about you and your company.
3. In the `Email Address` field, make sure to enter a public email address you can access easily from a Web browser.
4. In the `Username` field, enter `dev_` and then the email address you used in the previous step. In this workbook you'll have two DE orgs, one for development and one for testing, and you want to be able to easily distinguish between them by name.

Note: The `Username` field is in the *form* of an email address, but it does not have to be a *valid* email address. In most cases, it's best to change the username to something descriptive, as you just did. How you name your orgs can help you distinguish them later.

5. Read and then select the checkbox for the `Master Subscription Agreement`.
6. Enter the Captcha words shown and click **Submit Registration**.
7. In a moment you'll receive an email with a login link. Click the link and change your password.

After you develop and package your app, you'll need another org for testing purposes, your "test org". You'll use this org to install the app and use it, just as a customer would.

1. Repeat the previous steps to sign up for another DE org.
2. In the `Username` field, enter `test_` and then your email address.
3. In a moment you should receive two emails welcoming you to Developer Force, and two emails with login links. Click the links to login and then change your password.
4. In your test org, click **`Your Name`** > **Logout**.

Tell Me More....

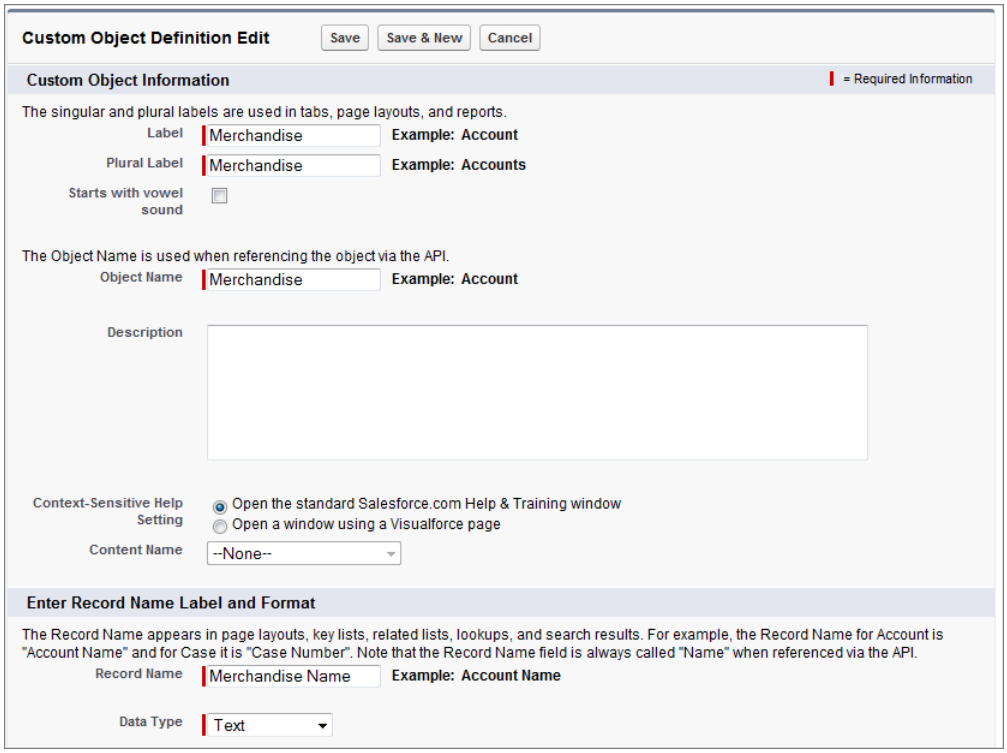
When you sign up for a DE org, you also become part of the Force.com community, which includes forums, webinars, and other exclusive content. `http://developer.force.com`

Step 2: Create an App

In this step you create a Merchandise object and a tab to display it. You then create an app that contains the tab.

1. Your browser should already be open, log into your dev org (this is the first DE org you created).

2. From Setup, click **Create** > **Objects**.
3. Click **New Custom Object** to display the New Custom Object wizard.
4. For the `Label` and `Plural Label` enter `Merchandise`, leave all other values as they are.

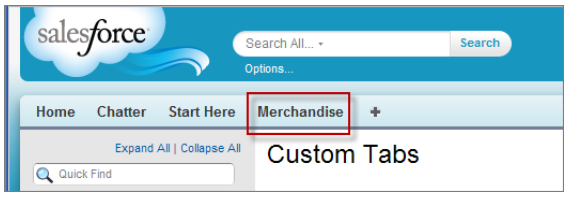


5. Click **Save** to finish creating your new object.

To display information about the Merchandise object, you associate the object with a tab.

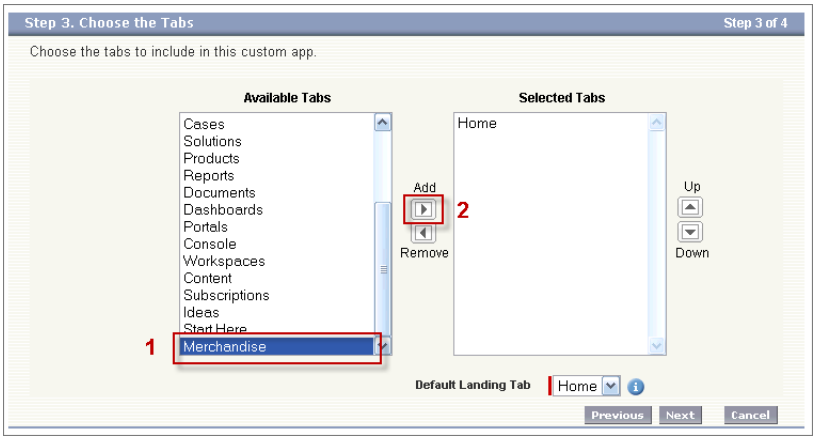
1. From Setup, click **Create** > **Tabs**.
2. Select **Custom app** and click **Next**.
3. In the Custom Objects Tabs related list, click **New** to launch the New Custom Tab wizard.
4. From the `Object` drop-down list, select Merchandise.
5. For the `Tab Style`, click the lookup icon and choose an icon.
6. Accept the remaining defaults, and click **Next**, **Next**, and then **Save**.

As soon as you create the tab, you can see it at the top of the screen.

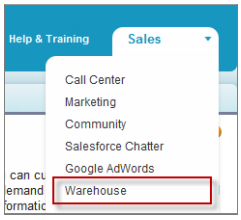


Next create an app that contains the tab.

1. From Setup, click **Create** > **Apps**.
2. Click **New** to launch the New Custom App wizard.
3. For the `App Label` and `App Name` enter `Warehouse`.
4. Click **Next** and **Next** again.
5. In the Available Tabs list, locate the Merchandise tab and click **Add** to add it to the list of selected tabs.



6. Leave the `Default Landing Tab` set to the Home tab, and click **Next**.
7. Select the `Visible` checkbox to make the application available to all user profiles.
8. Click **Save** to create the Warehouse application.
9. The app now appears in the Force.com app menu in the upper right corner of the page. Click it.



Tell Me More....

If it seems like you just created an object within a container, within another container, you did. And you're about to put all of that into another container! What's with all these containers and what do they do?

- A *tab* is a container for things you want to display on the same page, such as a chart, a table, or the Merchandise object you created.
- An *app* is a container for tabs that appear next to each other. Currently your app has only two tabs, Home and Merchandise, but it could have many more.
- A *package* is a container for components. Usually a package contains an app that can be installed in an org. You haven't created a package yet, you'll do that in the next tutorial.

Development Summary

In this tutorial you signed up for two Developer Edition (DE) organizations, for development and testing. You used your development organization to create a simple Warehouse app that contains a Merchandise tab. The Merchandise tab displays information about the Merchandise custom object.

Tutorial #2: Package and Upload Your App

In this tutorial you package the Warehouse app so that other people (your customers) can install the app in their org. You'll also choose a namespace, which is a unique identifier used for the components (objects, tabs, apps, etc) you create in your dev org. The namespace allows you to identify your components in your customers' orgs and to upgrade those components in the future.

Step 1: Package Your App

In this step you'll package the app so other people can download it. A package is simply a container for components, in this case it's your Mechandise object, tab, and Warehouse app.

1. From Setup, click **Create** > **Packages**, and then click **New**.
2. In the Package Name field enter `Warehouse Components` and then click **Save**.
3. On the Package Detail page click **Add**.
4. Select the checkbox next to your Warehouse app and click **Add to Package**.

Tell Me More....

The Components tab displays the components in your package. When you clicked **Add to Package** for your app, did you notice that your Merchandise object and tab were automatically added? Other dependent files, such as the page layout are also added. The framework automatically detects dependent components and adds then to the package.

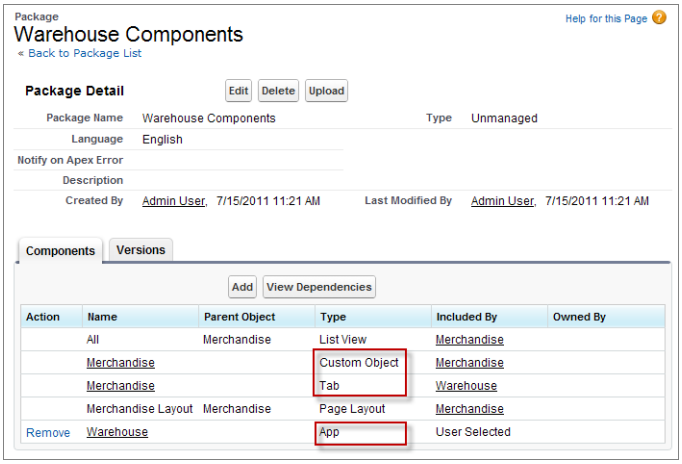


Figure 1: Components in a Package

Step 2: Assign a Namespace

In this step you choose a unique identifier called a namespace. A namespace differentiates your packaged components from other components in your customers' orgs.

1. In the sidebar menu, click **Create** > **Packages**.
2. In the Developer Settings list, click **Edit** and on the following page click **Continue**.
3. In the Namespace Prefix field, enter a 1-15 character alphanumeric ID and then click **Check Availability**. Repeat this step until you have a unique namespace.

Note: You might be wondering if the characters that make up your namespace are important. Not really. Try a shorthand for your company name or something easy to remember. The only people that see the namespace are developers.

4. In the `Package to be managed` field choose your Warehouse Components package and then click **Review Your Selections**.
5. Review the information on the page and then click **Save**.

Tell Me More....

Within the underlying code, your namespace is prepended to all components that are packaged from your dev org. This allows your package and its contents to be distinguished from those of other developers, and ensures your exclusive control of all packaged components.

Step 3: Upload a Beta Version

Before you upload a released version of your app, it's a common practice to upload a beta version for testing.

1. On the Packages detail page, click your **Warehouse Components** package.
2. Click **Upload**.
3. On the Upload Package page, enter a version name and number.
4. For the Release Type, make sure to choose **Managed — Beta**.
5. Click **Upload**. It may take a moment for the upload to complete, so you'll be notified by email just as soon as the package is ready.

Congratulations, you've uploaded an app! Your app isn't available to the general public, but it can be accessed through an install link. You'll install the app in the next step.

Tell Me More....

The purpose of a beta is for testing, and so it can only be installed in a test org, Developer Edition, or sandbox. A sandbox is a replica of your customer's org that allows them to test things before they commit to using them. DE orgs don't have a sandbox, but if you have a sandbox in another org and want to install your app in it, you must replace the initial portion of the **Installation URL** with `http://test.salesforce.com`.

Step 4: Install and Test the Beta Version

Installing the beta is easy, just click the email link and provide the username and password you use for your test org.

1. Log out of your dev org by clicking **`Your Name`** > **Logout**.
2. Click the Installation URL link you received in your email.
3. On the login page, enter the Username and Password of your test org.
4. On the Package Installation Details page, click **Continue**.

Note: If you get a warning that you can't install this package because it was created in this org, you accidentally developed in your test org. It's OK, just remember from now on that your test org is really your dev org and vice versa.

5. Click **Next**.
6. On the Security Level page, **Grant access to all users** and click **Next**.
7. Click **Install**.

8. Click **Deploy Now** and then **Deploy**.
9. Once the installation completes, you can select your app from the app picker in the upper right corner.

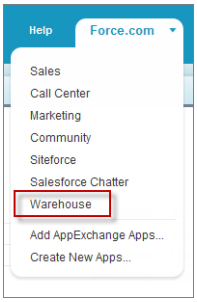


Figure 2: Warehouse App

10. Let's test the app and see if it works. Click the **Merchandise** tab and add then click **New** to create a new record.
11. In the `Merchandise Name` field enter `Wee Jet` and then click **Save**.

Your app installs easily and you can add the name for Merchandise records, your beta test has passed! Next you'll upload a released version of this package.

Tell Me More....

Your app doesn't do much, in fact it only has a single field. The name field is called a *standard field* because it is included automatically with every custom object you create. You can also create *custom fields* to further define your object.

Package and Upload Summary

Congratulations, you just completed a software development lifecycle! Further changes to your app will follow the same procedure.

1. Modify the existing app in your dev org.
2. Package and upload a beta.
3. Install and test the beta.
4. Repeat this process until you have a version you want to release.

Tutorial #3: Upload a Released Version

Imagine you've been through a few development cycles with your beta and you're ready to publish a public app. The next step is to upload a production app, or what we call a *managed released* version of your app.

Step 1: Upload a Managed Released Version

This step will seem familiar, it's similar to uploading a beta.

1. If you've been following along non-stop, you're probably still logged into your test org. Log out and then log into your dev org.
2. Notice in the upper right corner there's a link that says **Developing Warehouse Components, version 1.0**. Click that link to go directly to the Package Detail page.

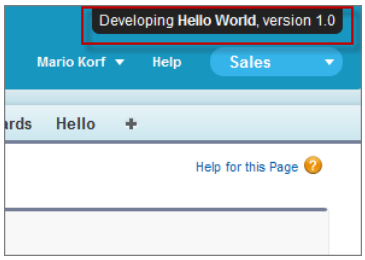


Figure 3: Developing Warehouse Components, version 1.0

3. On the Package Detail page, click **Upload**.
4. Enter a version name.
5. For the Release Type, choose **Managed — Released**.
6. Scroll to the bottom and click **Upload** and **OK** on the popup.

Tell Me More....

Just as before, you'll receive an email with a login link when the package is ready.

Step 2: Uninstall the Beta Version

Before you can install the managed released version of your app, you first need to uninstall the beta.

1. Log into your test org.
2. From Setup, click **Installed Packages**.
3. Click **Uninstall** next to the Warehouse Components package.
4. Select the checkbox and click **Uninstall**.

Tell Me More....

If you created some Merchandise items, the data is preserved in a .zip file so you can import it later. Importing that data is beyond the scope of this workbook, but your customers will be happy to know their data is saved.

Step 3: Install the Released App

Now you can install the released app.

1. Click the login link you received for the released version.
2. Install the released app in the same way you installed the beta in Step 4: Install and Test the Beta Version.
3. Once the app installs, choose the Warehouse app from the drop-down menu and then click the Merchandise tab.
4. Click Add to make sure it works.

Released App Summary

In this tutorial, you uploaded your managed-released app. A released app can be installed in a production org, so that customers around the world can use install your app in their org. Congratulations are in order, well done!

What Next?

If you'd like more of an introduction to development on Force.com, try the Force.com Workbook. The Merchandise object in this tutorial is a stripped down version of the Merchandise object in the Force.com Workbook, so you already have a head start. As you complete the tutorials in the Force.com Workbook, periodically package and upload components to create new versions of the app.

If you're ready to create your killer app, see the Quick Start in the ISVforce Guide, which takes you through the process of signing up for the Salesforce Partner Program, setting up licensing for your app, and registering for the AppExchange.