



# Salesforce Flow Introduction

Leeds Developer User Group

April 2014



# About FinancialForce.com

- Strong ties to Salesforce
  - Joint venture of Salesforce.com and Unit4
  - Executive involvement
  - Native on the Salesforce platform
- HQ San Francisco; Development teams in US, UK and Spain
- 260 Employees
- Customers in 27 Countries with Users in 45 Countries
- 3 Force.com MVP's!

# Session Overview

**Understand what Flow is, who uses it,  
what it can do, where it can be used  
and future roadmap**

- What is Flow?
- What can I do with it?
- Where can it be used?
- Empower your Admins with Flow Plugins!
- Can I Package it?
- Spring'14 and Flow

# What is Flow?

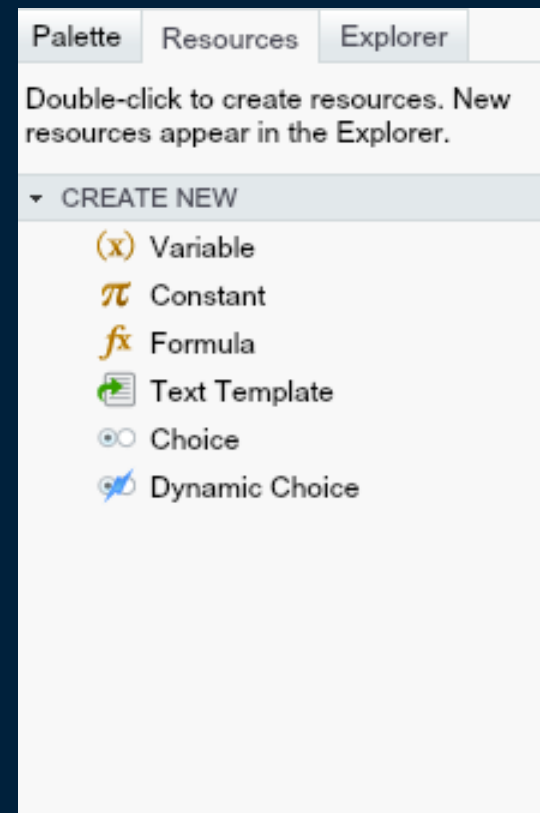
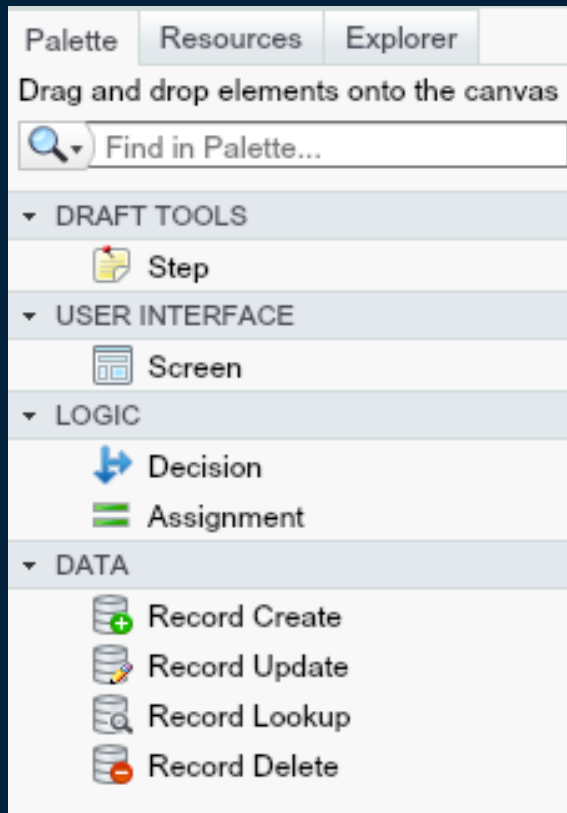
- Also known as...
  - Visual Workflow
  - Visual Flow
  - #clicksnotcode
- What can I do with it?
  - Create Wizard based UI's
  - Manipulate records on Standard or Custom Objects
  - Callout to Apex code (Advanced)
- How is it different from Workflow?
  - Designed to build UI's that are process or task orientated
  - Workflow is event driven based on database events

# Where can Flow be used?

- URL Entry
- Custom Tabs
- Custom Buttons
- Visualforce Pages
  - Anywhere a Visualforce page can be used!
    - Embed in Layouts, Sidebar Component, Salesforce1
- Workflow Actions
  - Pilot for Spring'14, watch this space!

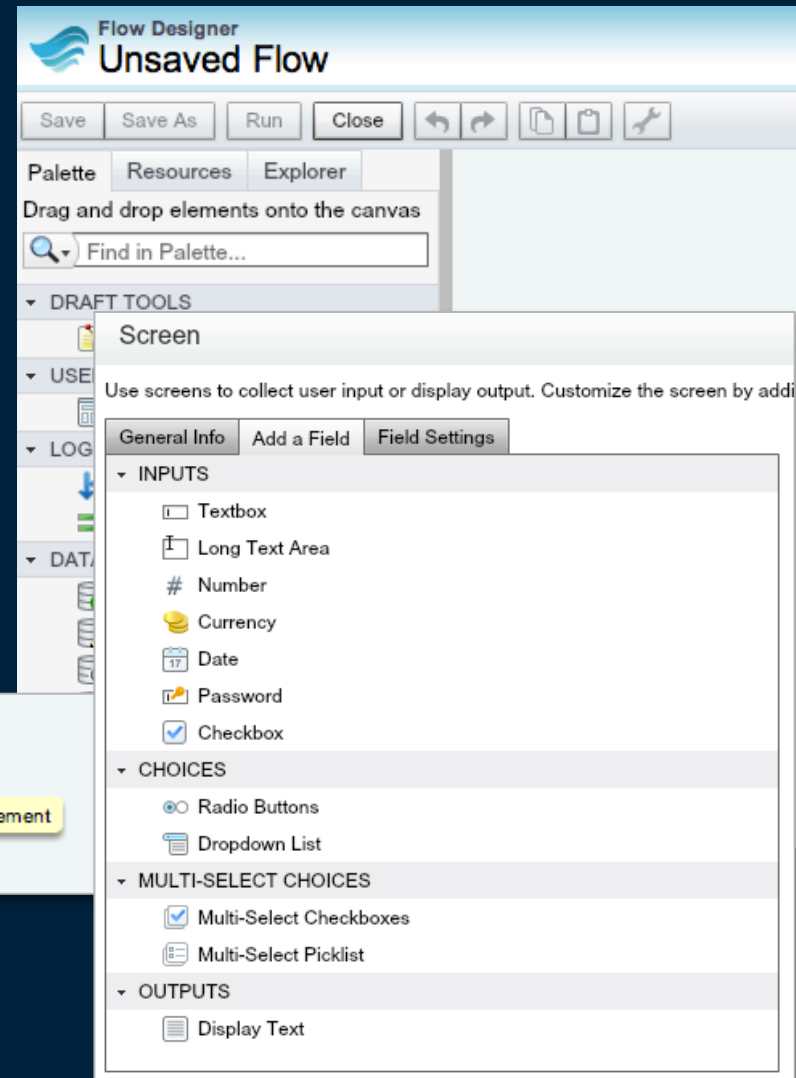
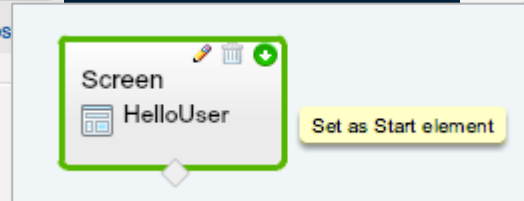
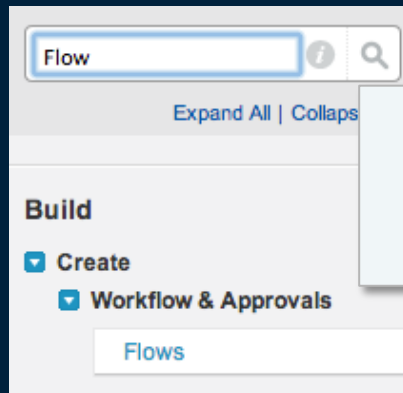
# The Flow Toolbox!

- Palette and Resources



# Demo : Basic Flow to Say Hello!

1. Creating a Flow
2. Add Screen element
3. Add Screen output field
4. Marking the Start element
5. Running the Flow
  - `/flow/HelloFlow`



# Demo : Basic Flow to Say Hello!

**Flow Detail** Edit Open Run Delete

Flow Name	Hello Flow	Unique Name	HelloFlow
Description		Namespace Prefix	
		URL	/flow/HelloFlow
Active Version		Activated/Deactivated	

Hello Flow

<https://emea.salesforce.com/flow/HelloFlow>

**Hello Flow**

Finish

Hello Flow!

Finish



# Demo : Connecting to a Tab

1. Create a Web Tab
2. Utilize the Flow URL

**Flow Detail**

EditOpenRunDelete

Flow Name	Hello Flow	Unique Name	HelloFlow
Description		Namespace Prefix	
		URL	/flow/HelloFlow
Active Version		Activated/Deactivated By	
Created By			04/2014 12:43

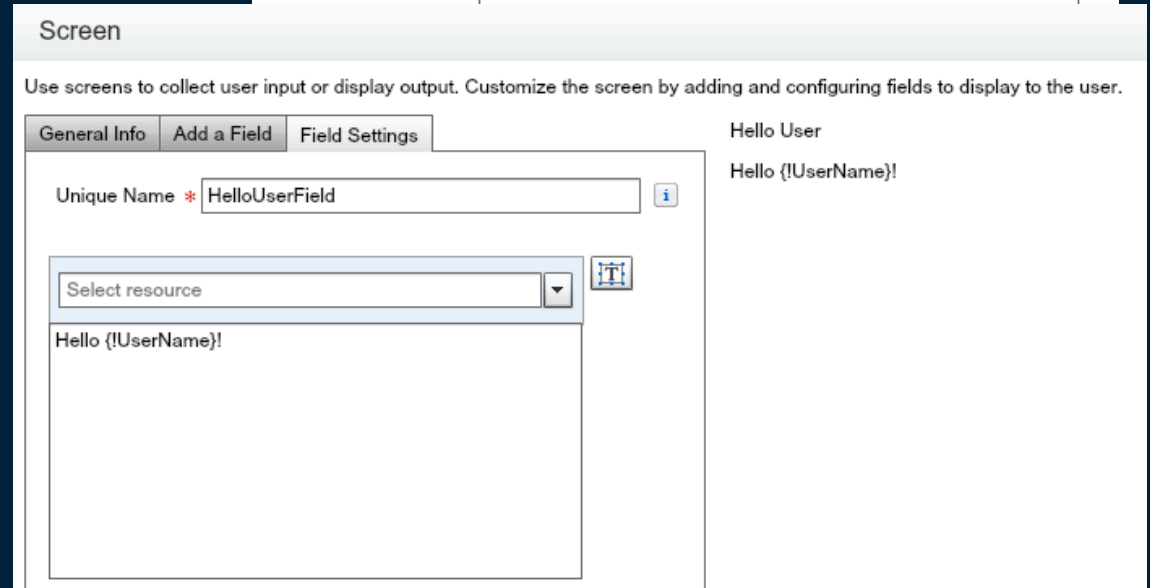
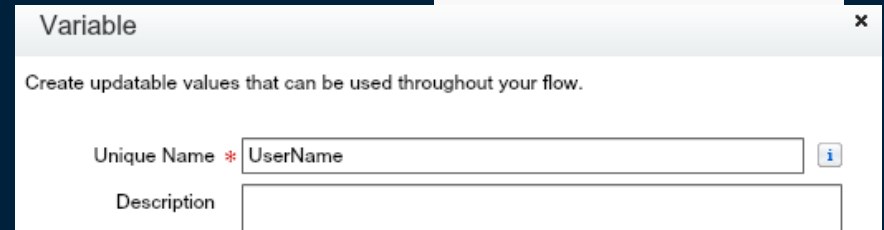
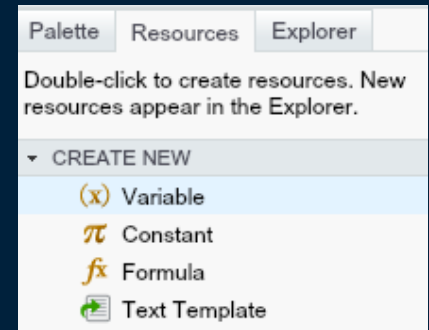
HomeHello Flow+

## Hello Flow

Hello Flow!

# Demo : Passing in Parameters

1. Create new UserName Variable
2. Add Screen element referencing the `{!UserName}` variable
3. Pass via URL  
or  
Visualforce Page



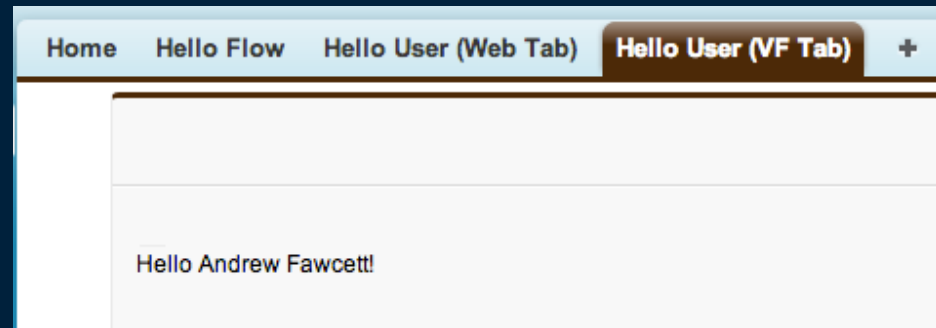
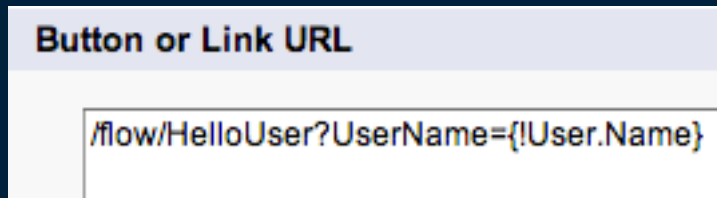
# Demo : Passing in Parameters

- Passing via URL

`/flow/HelloUser?UserName=Andrew%20Fawcett`

- Passing via URL

(Web Tab Example)



- Passing via Visualforce page (Recommended)

(Visualforce Tab Example)

```
<apex:page >
  <flow:interview name="HelloUser">
    <apex:param name="UserName"
      value="{!$User.FirstName} {!$User.LastName}" />
  </flow:interview>
</apex:page>
```

# Demo : Flow in Salesforce1

**Mobile Navigation**

Add Salesforce productivity items, smart search items, Visualforce pages, and apps to the mobile menu.

When organizing the menu items, put the items that users see on the Salesforce1 landing page for users. The Smart Search item will appear in the Apps section of the menu, and it may push other elements below the section.

The Today menu item is only available for the Salesforce1 mobile app.

**Navigation Menu Items** [Save] [Cancel]

Available
Hello User (VF Tab)

**Visualforce Page**  
**HelloUserFlow**

**Page Edit** [Save] [Quick Save]

**Page Information**

Label: Hello User Flow

Name: HelloUserFlow

Description:

Available for Salesforce mobile apps ☒

Search Salesforce

- Hello User (VF Tab)
- Feed
- People

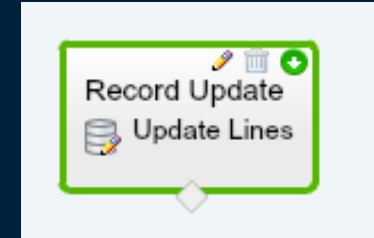
Finish

Hello Andrew Fawcett!

Finish

# Demo : Updating the Database

1. Add Variable `OpportunityId`
2. Add Record Update element
  - Reference `OpportunityId` in filter
  - Reference fields to update



Record Update

Use filters to find a specific record, then select fields to update.

▼ General Settings

Name \*

Unique Name \*  [Add Description](#)

▼ Filters and Assignments

Update \*  that meet the following criteria:

Field	Operator	Value
<input type="text" value="OpportunityId"/>	<input type="text" value="equals"/>	<input type="text" value="{!OpportunityId}"/>

[Add Row](#)

Update record fields with variable, constant, input, or other values.

Field	Value
<input type="text" value="Quantity"/>	<input type="text" value="1"/>

# Demo : Updating the Database

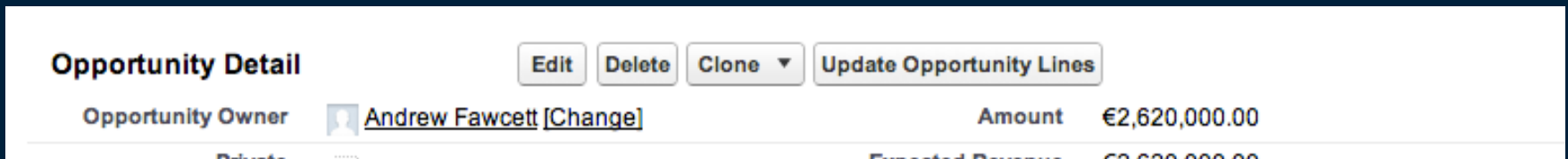
## 1. Create Visualforce Page

- Using Standard Controller

```
<apex:page standardController="Opportunity">
  <flow:interview name="UpdateOpportunityLines"
    finishLocation="/{!Opportunity.Id}">
    <apex:param name="OpportunityId"
      value="{!Opportunity.Id}" />
  </flow:interview>
</apex:page>
```

## 2. Create Custom Button

- Using above Page

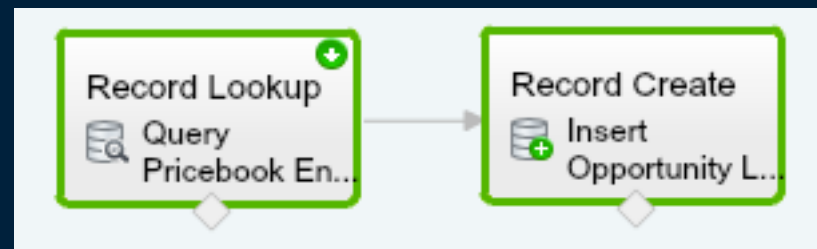


The screenshot shows the 'Opportunity Detail' page in Salesforce. At the top, there are buttons for 'Edit', 'Delete', 'Clone' (with a dropdown arrow), and a custom button 'Update Opportunity Lines'. Below the buttons, the 'Opportunity Owner' is listed as 'Andrew Fawcett' with a '[Change]' link. The 'Amount' is displayed as '€2,620,000.00'. The page is partially obscured by a dark overlay at the bottom.

# Demo : Creating Sub Flows

*“Create a Sub Flow that adds a Opportunity Line Item when given a Product Code, Quantity, Total Price and Opportunity Id”*

1. Add Variables for Input Only
2. Add Record Lookup element
  - Looks up the PriceBookEntryId by Product Code
3. Add Record Create element
  - Inserts the OpportunityLineItem record
4. Save the Flow



# Demo : Creating Sub Flows

- Record Lookup element

Look up \*  that meets the following criteria:

Field	Operator	Value
<input type="text" value="ProductCode"/>	<input type="text" value="equals"/>	<input type="text" value="{!ProductCode}"/>

[Add Row](#)

---

☐ Sort results by:

---

Assign the record's fields to variables to reference them in your flow.

Field	Variable
<input type="text" value="Id"/>	<input type="text" value="{!PricebookEntryId}"/>

[Add Row](#)



# Demo : Creating Sub Flows

- Record Create element

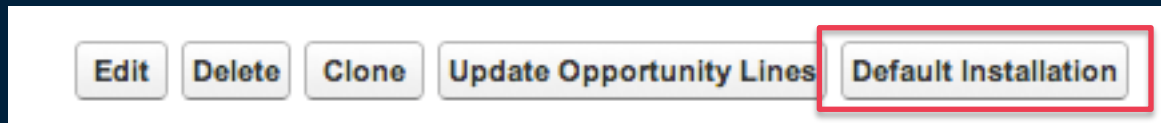
Create \*  ▼ with the following field values:

Field	Value
<input type="text" value="OpportunityId"/> ▼	<input type="text" value="{!OpportunityId}"/> ▼
<input type="text" value="PricebookEntryId"/> ▼	<input type="text" value="{!PricebookEntryId}"/> ▼
<input type="text" value="Quantity"/> ▼	<input type="text" value="{!Quantity}"/> ▼
<input type="text" value="TotalPrice"/> ▼	<input type="text" value="{!TotalPrice}"/> ▼

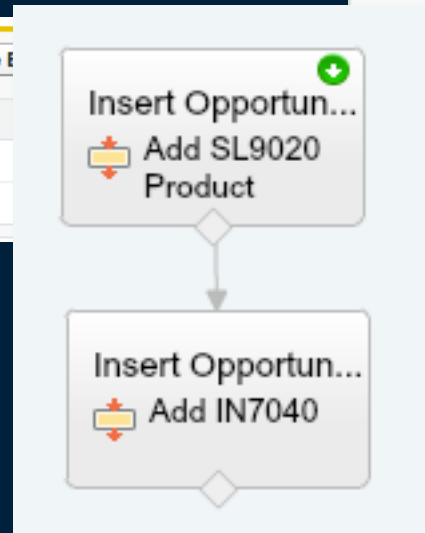
[Add Row](#)

# Demo : Inserting Opportunity Lines

1. Create Variable `OpportunityId`
2. Add `InsertOpportunityLine` elements for each Product required
3. Create a Visualforce page and Custom Button



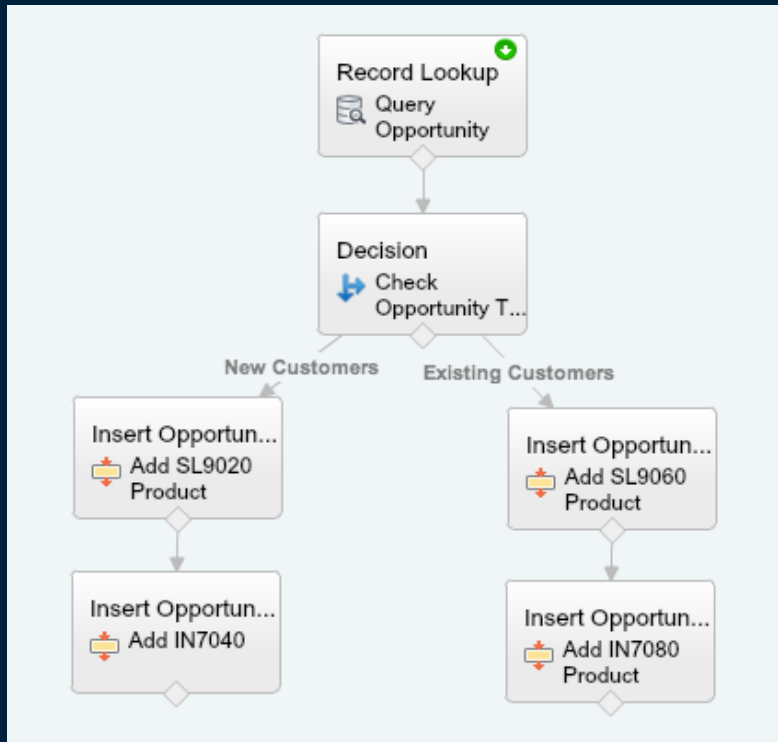
Products (Standard)			
			<a href="#">Add Product</a> <a href="#">Edit All</a> <a href="#">Choose Price E</a>
Action	Product	Quantity	
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Installation: Industrial - Low</a>	1.00	
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">SLA: Bronze</a>	2.00	



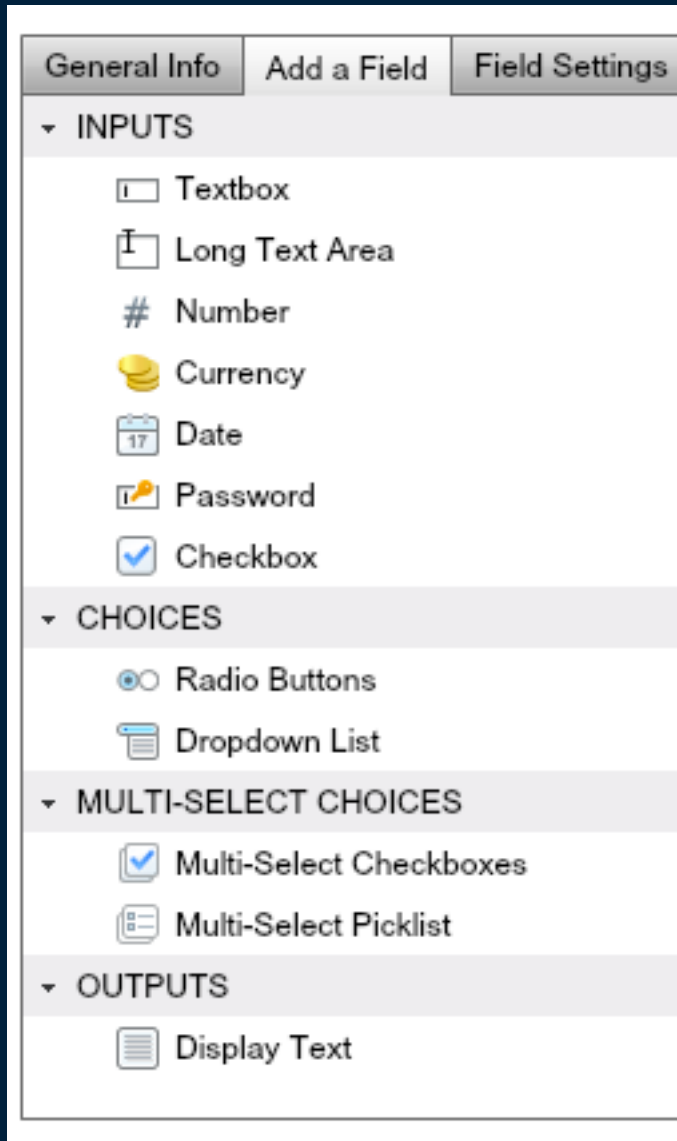
A screenshot of the Salesforce Flow Builder interface. The 'FLOWS' section on the left shows a list of flows: 'Hello Flow', 'Hello User', 'Insert Opportunity Line' (highlighted with a red box), and 'Update Opportunity Lines'. The 'Description' section on the right shows the details for the 'Insert Opportunity Line' flow. It includes the flow name 'InsertOpportunityLine', version '1', inputs 'OpportunityId (Text)', 'ProductCode (Text)', 'Quantity (Number)', and 'TotalPrice (Number)', and outputs 'None'.

# Demo : Conditional Flows

- Add Record Lookup element
  - Query for the Opportunity Type
- Add Decision element
  - Add default Products based on Type



# Screen Element “more to see”...



- Many UI elements
  - Radio Buttons
  - Dropdown Lists
  - Multi-Selects
  - Text
  - Number
  - Currency
  - Password
- Wizard style decision UI

# Empower ClicksNotCode with Plugins!

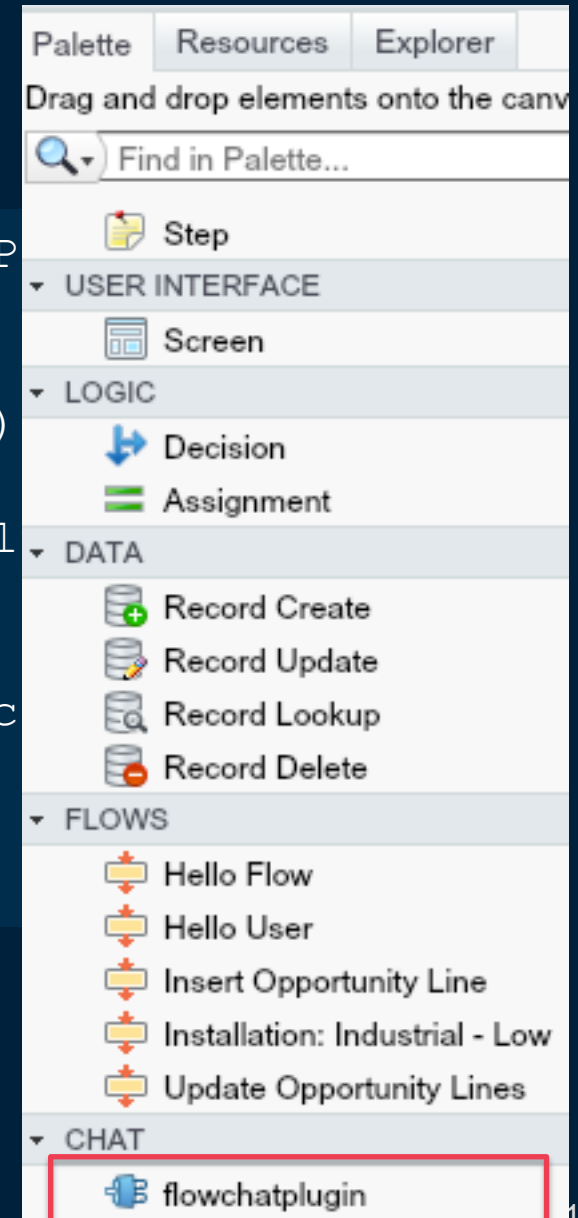
- Plugs can extend the Flow Toolbox!

```
global class MyPlugin implements Process.P
{
  global Process.PluginResult
    invoke(Process.PluginRequest request)
  {
    return new Process.PluginResult(resul
  }

  global Process.PluginDescribeResult desc
  {
  }
}
```

## Visual Workflow

- Passing Data to a Flow Using the Process.Plugin Interface
  - Implementing the Process.Plugin Interface
  - Using the Process.PluginRequest Class
  - Using the Process.PluginResult Class
  - Using the Process.PluginDescribeResult Class
  - Process.Plugin Data Type Conversions
  - Sample Process.Plugin Implementation for Lead Conversion



# Packaging, Change Sets and Metadata API

- Packaging
  - Supported
  - Upgradable
  - Subscriber Editable
    - Only Name, Description and Status
  - Namespace aware
- Change Sets
  - Supported
- Metadata API
  - Supported
    - Dynamically create Flows!?! 😊

# Spring'14 and “Headless” Flow!

- Launch Flows from Workflow Rules—Pilot
- *“A trigger-ready flow is a flow that can be launched from a **flow trigger workflow action**. Because trigger-ready flows must be able to run in bulk and without user interaction, they can’t contain Step, Screen, or Apex Plug-in elements in any flow version.”*
  1. Create and activate the trigger-ready flow.
  2. Create the workflow rule.
  3. Create the flow trigger workflow action. Make sure you select the same object for the workflow rule and the flow trigger.
  4. Associate the flow trigger to the workflow rule.
  5. Activate the workflow rule.

# Spring'14 and “Headless” Flow!

- Possibilities!?!
  - Enhanced Field Defaulting Logic
    - Default Lookup Fields on Create
  - Creating Default Related Records
    - On Create create related records or children
    - Default Product Lines on an Opportunity?
  - Calculated Fields
    - Perform calculations not possible in Formula Fields
  - Other ideas...?



# Spring'14 and “Headless” Flow!

- *“With great power comes great responsibility!”*
- Dangers and Limitations
  - Performance!?!?!?
  - *“In a transaction, flow triggers are executed after all workflow field updates, including any Apex triggers and standard validations that are executed as a result of those workflow field updates.”*
    - Hmmmm bypass Apex Trigger validations?!?!?
  - Not available on Approval Processes
  - Not yet available in Change Sets or Packages

# Questions?

- Contact Info and Social Channels
  - **Twitter:** andyinthecloud
  - **Blog:** andyinthecloud.com
  - Also on Salesforce StackExchange  
<http://salesforce.stackexchange.com/>
  - GitHub Open Source Projects  
<https://github.com/afawcett>  
<https://github.com/financialforcedev>